



**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES  
"CUAUTITLAN"**

**INSTALACION Y ADMINISTRACION DE UN  
SERVIDOR WORLD WIDE WEB.**

**T E S I S**  
**QUE PARA OBTENER EL TITULO DE:**  
**LICENCIADO EN INFORMATICA**  
**P R E S E N T A:**  
**ARTURO BAUTISTA SOTO**

**ASESOR: LIC. CARLOS PINEDA MUÑOZ**

**CUAUTITLAN IZCALLI, EDO. DE MEX.**  
**TESIS CON**  
**FALLA DE ORIGEN**

**1997**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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



INSTITUCIÓN NACIONAL  
AUTÓNOMA DE  
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN  
UNIDAD DE LA ADMINISTRACIÓN ESCOLAR  
DEPARTAMENTO DE EXÁMENES PROFESIONALES

U. N. A. M.  
FACULTAD DE ESTUDIOS  
SUPERIORES CUAUTITLAN

ASUNTO: VOTOS APROBATORIOS



DR. JAIME KELLER TORRES  
DIRECTOR DE LA FES-CUAUTITLAN  
P R E S E N T E .

AT'N: Ing. Raísel Rodríguez Ceballos  
Jefe del Departamento de Exámenes  
Profesionales de la F.E.B. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS:

"Instalación y Administración de un servidor World Wide Web".

que presenta el pasante: Arturo Bautista Soto.  
con número de cuenta: 8808775-9 para obtener el TÍTULO de:  
Licenciado en Informática.

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .

"POR MI RAZA HABLARA EL ESPIRITU"  
Cuautitlan Izcalli, Edo. de Mex.. a 14 de Agosto de 1997

PRESIDENTE	Fis. Jesús Cruz Guzmán.	
VOCAL	Lic. Carlos Pineda Muñoz.	
SECRETARIO	Lic. Valentín Roldán Vázquez.	
PRIMER SUPLENTE	Ing. Moisés Hernández Duarte.	
SEGUNDO SUPLENTE	Lic. Araceli Nivón Zaghi.	

## *Dedicatorias.*

### **A Dios.**

Por cuidarme y permitirme vivir todos estos momentos que para mí son de gran importancia y satisfacción.

### **A Mis Padres.**

Por haberme dado un gran ejemplo de honradez y admirable lucha por conseguir lo deseado, por haberme dado todo lo que he necesitado sin perder de vista lo que cuesta obtener las cosas. Por lo mucho que me han enseñado de la vida. Por el gran esfuerzo que han realizado para que yo logre concluir mis estudios de la manera más satisfactoria. Por haberme enseñado su gran concepto de la familia, por su amor recibido, por ser unas personas admirables.

Ni todas las palabras del mundo son suficientes, para agradecerles todo lo que han hecho por mí, sólo quiero decirles que hoy soy lo que soy gracias a ustedes. Los quiero mucho.

### **A Mis Abuelitos.**

Por los bellos momentos que hemos vivido juntos. Por enseñarme a ser una persona de provecho. Por haberme dado todos sus consejos y cariño que siempre recibí y he recibido. Y a pesar de que algunos de ellos ya no están físicamente con nosotros, yo sigo sintiendo sus cuidados y su cariño. Sirva este trabajo para reiterarles mi eterno cariño.

### **A Mis Hermanos.**

Por su compañerismo y unión que me hacen sentir su compañía, para que nuestros anhelos se vean realizados.

**A Mis Tíos y Primos.**

Porque en ellos siempre he encontrado los consejos y apoyos necesarios cuando los he necesitado. Por todas sus palabras de aliento y porque siempre han estado a mi lado.

**A Mi Novia.**

Por ese cariño siempre incondicional que he recibido por parte de ella. Por todas sus palabras de aliento, su motivación y su comprensión.

**A Mi Asesor.**

Por dedicarme su valioso tiempo y atención para orientarme en la realización de esta tesis.

**Al Fis. José de Jesús  
Cruz Guzmán.**

Por haberme brindado todo su apoyo y confianza durante mi estancia dentro de la Facultad.

**Al Dr. Hiram Gutiérrez  
Renovato.**

Por brindarme todo su apoyo y sus consejos, que para mí son más que valiosos para ser mejor como ser humano y profesionista.

**A Mis Amigos.**

Por haberme brindado su confianza y amistad, así como por haberme alentado durante toda mi carrera. Gracias muchachos, nunca lo olvidaré.

## Indice.

<b>Introducción.</b>	<b>1.</b>
<b>Capítulo 1. Antecedentes Históricos.</b>	
1.1 ¿ Qué es Internet. ?	3.
1.2 ¿ Qué es World Wide Web. ¿	5.
<b>Capítulo 2. Fundamentos y Servicios de Internet.</b>	
2.1 ¿ Quién gobierna a Internet. ?	6.
2.2 ¿ Quién paga lo que cuesta la red. ?	7.
2.3 ¿ Qué puedo obtener en Internet. ?	8.
2.4 Servicios de Internet.	9.
2.5 El futuro de los derechos electrónicos.	9.
2.6 Perspectivas de Internet.	12.
2.7 Internet en México.	14.
<b>Capítulo 3. Tecnologías y Tipos de Conexiones de Red.</b>	
3.1 Tecnologías subyacentes de red.	15.
3.2 Dos enfoques de la comunicación por red.	15.
3.3 Redes de área amplia y redes de área local.	17.
3.4 Ancho de Banda.	19.
3.5 Direcciones de hardware de red.	20.
3.6 Tecnología Ethernet.	21.
3.7 Ethernet de cable delgado.	25.
3.8 Ethernet de par trenzado.	26.
3.9 Esquemas de cableado múltiple y adaptadores.	27.
3.10 Capacidades de las redes Ethernet.	28.
3.11 Interconexión de datos distribuida por fibra (FDDI).	29.
3.12 Propiedades de una red FDDI.	29.
3.13 Anillos dobles de rotación contraria.	30.
3.14 Microondas.	32.
3.15 Tecnología ATM.	33.

#### Capítulo 4. Protocolos de Comunicación.

4.1 TCP/IP.	35.
4.2 El modelo de estratificación por capas de TCP/IP.	36.
4.3 Tendencias de Internet: Nuevos estándares de protocolos.	39.
4.4 Jerarquías de los Protocolos.	39.
4.5 Modelo de referencia OSI.	40.
4.6 Normalización de Redes.	43.
4.7 ¿ Quién determina el mundo de las normas. ?	44.

#### Capítulo 5. Enlace de Redes y Direcciones Internet.

5.1 Interconexión de nivel de aplicación.	47.
5.2 Interconexión de nivel de red.	47.
5.3 Arquitectura de Internet.	48.
5.4 El datagrama de Internet.	49.
5.5 Ruteo en una red de redes.	49.
5.6 La tabla de ruteo.	52.
5.7 Identificadores Universales.	54.
5.8 Direcciones IP.	55.
5.9 Subredes.	57.
5.10 Debilidades del direccionamiento de Internet.	58.
5.11 Notación decimal con puntos.	58.
5.12 Dirección Loopback.	59.
5.13 Orden de octetos de red.	59.
5.14 Autoridad de direccionamiento Internet.	60.

#### Capítulo 6. Configuración del host y Sistema de Nombre de Dominio.

6.1 Host y nombres de dominios.	62.
6.2 Nombres de host.	62.
6.3 Nombres de Dominio.	63.
6.4 Tipos de nombres de dominios.	63.
6.5 Registrando su nombre de dominio.	65.
6.6 La tabla de hosts.	69.
6.7 BIND y DNS.	70.
6.8 Configuración del nombre del servidor.	71.
6.9 Mapeo de Archivos.	74.
6.10 CERN y el Web.	81.
6.11 NCSA y Mosaic.	81.

6.12 El protocolo.	82.
6.13 ¿ Qué es World Wide Web Consortium. ?	82.
6.14 Anatomía de un URL.	83.
6.15 Operaciones básicas del protocolo.	83.
6.16 Aspectos especiales de un servidor.	86.

### Capítulo 7. Instalando el servidor CERN HTTP.

7.1 Instalando el servidor CERN HTTP.	88.
7.2 Descomprimiendo el archivo.	90.
7.3 Uncompress y tar.	91.
7.4 Compilando el servidor.	93.
7.5 Preparándose para instalar el software.	93.
7.6 ¿ Dónde poner el software. ?	94.
7.7 ¿ Cómo se efectuará el servidor. ?	94.
7.8 ¿ Quién podría poseer los procesos del servidor. ?	96.
7.9 Instalando el software.	97.
7.10 Creando la arborescencia del directorio.	98.

### Capítulo 8. Instalando el servidor NCSA HTTP.

8.1 Instalando el servidor NCSA HTTP.	100.
8.2 Obteniendo el software.	100.
8.3 Descomprimiendo el archivo.	103.
8.4 Uncompress y tar.	104.
8.5 Compilando el servidor.	105.
8.6 Preparándose para instalar el software.	107.
8.7 ¿ Dónde poner el software. ?	108.
8.8 ¿ Cómo se ejecutará el servidor. ?	108.
8.9 ¿ Quién podría poseer los procesos del servidor. ?	110.
8.10 Instalando el software.	111.
8.11 Creando la arborescencia del directorio.	111.

### Capítulo 9. Tendencias y seguridad de Internet.

9.1 Visualizadores para Web.	113.
9.2 Organización del trabajo.	114.
9.3 El contenido, la clave del éxito.	115.
9.4 Seguridad en las comunicaciones.	117.
9.5 Secure Sockets Layer.	118.

<b>9.6 Firewalls.</b>	119.
<b>9.7 ¿ Cómo hacer rentables las páginas Web. ?</b>	120.
<b>Conclusiones.</b>	122.
<b>Anexo 1.</b>	126.
<b>Anexo 2.</b>	131.
<b>Anexo 3.</b>	138.
<b>Anexo 4.</b>	144.
<b>Anexo 5.</b>	148.
<b>Bibliografía.</b>	150.

## **Introducción.**

Cada uno de los tres siglos pasados ha estado dominado por una sola tecnología. El siglo XVIII fue la etapa de los grandes sistemas mecánicos que acompañaron a la Revolución Industrial. El siglo XIX fue la etapa de la máquina de vapor. Durante el siglo XX, la tecnología clave ha sido la recolección, procesamiento y distribución de información. Entre otros desarrollos, hemos asistido a la instalación de redes telefónicas en todo el mundo, a la invención de la radio y la televisión, al nacimiento y crecimiento sin precedente de la industria de las computadoras, así como a la puesta en órbita de los satélites de comunicación.

A medida que avanzamos hacia los últimos años de este siglo, se ha dado una rápida convergencia de estas áreas y también las diferencias entre la captura, transporte, almacenamiento y procesamiento de información están desapareciendo con rapidez. Organizaciones con centenares de oficinas dispersas en una amplia área geográfica esperan tener la posibilidad de examinar en forma habitual el estado actual de todas ellas, incluso la más alejada, simplemente oprimiendo una tecla. A medida que crece nuestra habilidad para recolectar, procesar y distribuir información, la demanda de los más sofisticados procedimientos de información crece todavía con mayor rapidez.

La fusión de los ordenadores y las comunicaciones ha tenido una profunda influencia en la forma en que estos sistemas están organizados. El concepto de "centro de cálculo" como un cuarto con un ordenador grande, al cual sus usuarios traían su trabajo para procesamiento, ha llegado a ser obsoleto. Este modelo no tiene uno, sino al menos dos aspectos deficientes: primero, el concepto de un solo ordenador grande haciendo todo el trabajo y segundo, la idea de que los usuarios traigan su trabajo al ordenador en lugar de llevar el ordenador a donde se encuentran los usuarios.

El viejo modelo de tener un solo ordenador para satisfacer todas las necesidades de cálculo de una organización, se está reemplazando con rapidez por otro que considera un número grande de ordenadores separados, pero interconectados que efectúan el mismo trabajo. Estos sistemas, se conocen como redes de ordenadores.

Se dice que dos ordenadores están interconectados, si estos son capaces de intercambiar información. La conexión no necesita hacerse a través de un hilo de cobre, puede hacerse mediante el uso de fibra óptica, microondas y satélites de comunicaciones.

Con Internet nos encontramos ante el nacimiento de un nuevo medio de comunicación; con características y potencialidades propias, que le hacen ocupar un nicho muy particular e incluso le permiten complementar y hasta suplantar algunas funciones de los medios tradicionales.

En el próximo siglo, seguramente se consumará la transición de las formas de poder basadas en la producción, a las del poder sustentado en la información y las comunicaciones.

Como efecto de la intensificación del proceso de globalización de la economía mundial, hoy en día, las empresas que pretenden acceder favorablemente al nuevo milenio ya enfrentan nuevas y agresivas formas de competencia. Para poder subsistir, las empresas del futuro tendrán que aprovechar nuevos medios de transmisión de la información que les permita dar a conocer de manera más rápida y efectiva sus productos y servicios.

Si los corporativos tuvieran acceso a su propia infraestructura de comunicaciones, esta fácilmente les aseguraría el acceso inmediato al gran mercado global, pero con un alto costo. Sin embargo, no es necesario que las empresas inviertan en algo tan complejo y costoso. Esa infraestructura de comunicaciones ya existe y se conoce como Internet.

## **Capítulo 1.**

### **1.1 ¿ Qué es Internet. ?**

Internet es el nombre de un grupo de recursos de información mundial. Por la forma como se habla de Internet, parece que es algo completamente nuevo, pero lo cierto es que la red de redes existe hace más de 20 años. A mediados de la década de 1960, temiendo que la guerra fría derivara en un ataque nuclear, el Departamento de Defensa de Estados Unidos comenzó a buscar la manera de construir una infraestructura de telecomunicaciones, capaz de soportar la pérdida de una de sus partes sin que esto afectara a las demás.

Ese proyecto fue conocido como ARPANET, ya que la dependencia encargada de realizarlo fue la Advanced Research Projects Agency (ARPA). En 1960, la ARPA formó una red experimental cuyo objetivo era el de enlazar a los centros de cómputo de varias universidades de la Unión Americana. En 1970, se lograron enlazar entre sí cuatro universidades: Stanford, UCLA, UCSB y la universidad de UTAH.

Posteriormente la red de ARPA se fraccionó en dos distintas cadenas, conocidas como MILNET y NSFNET. MILNET básicamente servía al gobierno de Estados Unidos, mientras que la NSFNET (National Science Foundation), era utilizada como soporte de instituciones educativas y centros de investigación en general.

Como MILNET estaba dedicada exclusivamente al servicio del gobierno estadounidense y NSFNET funcionaba como una organización no lucrativa para fines educativos, el desarrollo comercial de la red se tuvo que conformar a través de cadenas privadas.

Una de las más relevantes contribuciones del proyecto ARPA, fue el desarrollo de protocolos de comunicación que permitieron que todas las computadoras pertenecientes a la red se mantuvieran al mismo nivel, evitando que un punto central tuviera a su cargo la administración de la información. Esos protocolos son conocidos como TCP/IP (Transmission Control Protocol / Internet Protocol).

Los protocolos incorporaron una nueva técnica que permite transmitir la información por partes. Así, cada mensaje es dividido en pequeños paquetes de datos, lo que impide que una transmisión, por grande que esta sea, monopolice los servicios de la red. De esa manera se aseguró la supervivencia de la red en caso de un ataque nuclear, o cualquier otro tipo de desastre, y al mismo tiempo se dieron las condiciones para estimular el crecimiento de Internet.

Las posibilidades de intercambio de información y de comunicación remota, no pasaron desapercibidas para los investigadores académicos, quienes comenzaron a pedir a sus instituciones que se conectaran a la naciente red. Para 1972, aproximadamente 40 universidades ya formaban parte de la ARPANET, y sus computadoras tenían la capacidad de intercambiar mensajes y archivos, además de controlar a otras computadoras a distancia.

Hasta ese momento, el manejo de redes únicamente era posible en las grandes instituciones que contaban con computadoras de alto nivel. Sin embargo, a mediados de los setentas, la introducción de las minicomputadoras y la creación de Unix permitió el ingreso a Internet a instituciones medianas y pequeñas. De esa manera, por todas partes empezaron a multiplicarse las redes.

En 1979, se creó la Computer Science Research Network (CSNET), auspiciada por la National Science Foundation (NSF). El principal objetivo de esa red, era el de promover el intercambio de información científica y la integración de nuevos centros de educación e investigación.

No pasó mucho tiempo para que surgiera la idea de conectar entre sí a ARPANET y CSNET. Ese fue un paso decisivo en la historia de Internet.

Posteriormente surgieron nuevas redes, como Usenet, Bitnet y Fidonet, las cuales hicieron posible que más computadoras se conectaran a Internet. Pero lo que verdaderamente dio un impulso definitivo al crecimiento de Internet, fue la creación, por parte de la NSF, de una red de alta tecnología y velocidad para supercomputadoras, la cual conectaba más de una docena de centros de supercomputación de Estados Unidos. Esa red es el vehículo que garantiza mayor seguridad y rapidez en la transmisión de información.

Cuando en 1990 se decidió eliminar la obligación de contar con apoyo gubernamental para poder conectarse a Internet, comenzó un periodo extraordinario de crecimiento de la red, gracias al inicio de las actividades comerciales a través de esta. De 159,000 computadoras que en 1990 estaban conectadas a Internet, ascendieron a un total de 5,000,000 a mediados de 1995.

Debido al crecimiento que en los últimos años ha registrado Internet, hay quienes piensan que, en un futuro cercano todos los habitantes del planeta podrán tener acceso ilimitado a la gran supercarretera de información. Sin embargo, los 20 o 30 millones de usuarios de Internet que se prevén para el año 2000, apenas representan un 0.005% de los habitantes del planeta. Algunas personas sugieren que ese porcentaje no variará en un futuro inmediato, y que Internet no será más que una forma de extender el dominio de las clases privilegiadas sobre las masas.

## ***1.2 ¿ Qué es World Wide Web. ?***

World Wide Web es un sistema de información global, interactivo, dinámico, distribuido, gráfico, basado en hipertexto, con plataforma de enlaces cruzados, que se ejecuta en Internet. El Modelo World Wide Web hace uso de la naturaleza distribuida de los hipertextos y mediante un esquema cliente/servidor, proporciona un servicio de transferencia de hipertextos basado en el protocolo HyperText Transfer Protocol (HTTP).

Por tanto HTTP, es un protocolo de nivel de aplicación con la agilidad y velocidad necesaria para sistemas de información hipermedia distribuidos. HTTP es un protocolo orientado a objetos; de carácter genérico, que puede ser usado para muchas tareas, tales como, nombrar servidores y sistemas de manejo de objetos distribuidos, a través de la utilización de las extensiones de sus comandos.

Una característica del protocolo HTTP es la negociación de la representación de los datos, permitiendo que los sistemas sean construidos independientemente de los datos que sean transferidos.

HyperText Markup Lenguaje (HTML), es una aplicación de SGML, la cual está basada en la norma internacional ISO 8879 para marcado de textos. Hipertexto es básicamente lo mismo que un texto regular, puede ser almacenado, leído, buscado o editado; con una importante diferencia: Hipertexto contiene enlaces dentro del texto a otros documentos; estos enlaces son llamadas Hiperligas o simplemente ligas.

## **Capítulo 2.**

### **2.1 ¿ Quién gobierna a Internet. ?**

Internet no tiene presidente, director ejecutivo o mandatario. Las redes que componen a Internet pueden tener presidentes o directores ejecutivos; pero en Internet eso es distinto, no existe la figura de autoridad máxima como un todo.

La máxima autoridad sobre la cual descansa Internet es la Internet Society (ISOC). La ISOC es una organización de membresía voluntaria, cuyo propósito es promover el intercambio de información a nivel global mediante el uso de la tecnología de Internet. Esta designa a una especie de Consejo de Ancianos cuya responsabilidad consiste en la administración técnica y la dirección de Internet.

Quienes integran la ISOC son un grupo de voluntarios invitados llamado Consejo de Arquitectura de Internet -Internet Architecture Board, IAB-. El IAB se reúne con regularidad para "benedicir" estándares y asignar recursos, como los domicilios. Internet funciona porque existen formas estándar para que las computadoras y las aplicaciones de software se comuniquen entre sí. Esto permite que las computadoras de diferentes fabricantes puedan comunicarse sin ningún problema. Internet no es una red de equipos IBM, SUN o Macintosh. El IAB es el responsable de estos estándares: decide cuándo es necesario un estándar y cómo debe ser. Cuando se requiere un estándar, se considera el problema, se adopta el estándar y se anuncia a través de la red. El IAB lleva también un registro de algunos números que deben ser únicos.

Los usuarios de Internet expresan sus opiniones a través de la reuniones del Grupo de Trabajo de Ingeniería de Internet -Internet Engineering Task Force, IETF-. El IETF, es otra organización voluntaria que se reúne con regularidad para discutir problemas operacionales y problemas técnicos a corto plazo. Cuando considera que un problema amerita su atención, el IETF define un grupo de trabajo para realizar una investigación a fondo. Cualquiera puede asistir a las reuniones del IETF y participar en los grupos de trabajo; lo importante es que realmente se trabaje. Los grupos de trabajo tienen funciones diversas; que van desde la producción de documentación, hasta la decisión de cómo deben cooperar las redes cuando se presentan problemas y el cambio de significado de los bits que componen un paquete.

Dependiendo del tipo de recomendación que se haga, este puede consistir solamente en documentación que se pone a disposición de cualquier persona que la necesite, o en una buena idea aceptada por la gente en forma voluntaria o enviada al IAB para ser declarada un estándar.

## **2.2 ¿ Quién paga lo que cuesta la red. ?**

Nadie paga por el servicio; no existe la empresa Internet Inc., que cobre cuotas a las redes Internet o a los usuarios. En lugar de ello todos pagan su parte. La NSF paga por la NSFNET; la NASA paga por la NASA Science Internet. Las redes se reúnen para decidir cómo conectarse y cómo pagar la interconexión. Una escuela u organización paga por su conexión a una red regional, la cual a su vez paga por el acceso a un proveedor de servicios a nivel nacional.

Existe el mito de que Internet es gratis. No lo es, alguien paga por cada conexión de Internet. Muchas veces estas cuotas no llegan hasta el usuario final, lo que da la ilusión de ser de "acceso gratuito", pero existen muchos usuarios que saben que Internet no es gratis, pues pagan mensualmente o por hora los cargos por el acceso desde su hogar, a velocidades superiores a los 56 Kb por segundo (lo mismo que los enlaces originales de la red principal).

El concepto de que Internet no es una red, sino una red de redes significa poco para el usuario final. Lo que usted desea es hacer algo útil: correr un programa o tener acceso a algún tipo de información. No tendrá que preocuparse por cómo está conectado todo. Considere que el sistema telefónico es también una Internet. Pacific Bell, AT&T, MCI, British Telecom, Teléfonos de México, entre otras, son empresas distintas que operan partes de un sistema telefónico.

Los usuarios generalmente se acuerdan de quién transmite sus llamadas cuando ocurre algún problema. Si un conmutador se descompone sólo la compañía a la que pertenece puede componerlo. Las compañías telefónicas pueden hablar entre sí para resolver algún problema pero cada una deberá resolver los propios. Lo mismo se aplica para Internet. Cada red tiene su propio Centro de Operaciones de Red -Network Operation Center, NOC-. Los centros de operaciones se comunican entre sí y saben como resolver sus problemas.

### **2.3 ¿ Qué puedo obtener en Internet. ?**

Una vez conectado a Internet, un usuario tiene acceso a una gran cantidad de servicios y aplicaciones. Usted obtendrá beneficios y satisfacciones de Internet, si cualquiera de las siguientes posibilidades le parece útil, interesante o divertida.

- ✦ Comunicarse con amistades en diversas partes del mundo, sin tener que pagar cargos de larga distancia.
- ✦ Tener acceso a miles de bases de datos con información sobre temas diversos, desde la literatura hasta la ciencia, desde los negocios hasta el entretenimiento.
- ✦ Obtener demostraciones de los juegos más recientes.
- ✦ Comprar artículos de todo tipo, en tiendas repartidas por todo el mundo.
- ✦ Conocer gente de diversos países y culturas.
- ✦ Establecer un sistema de comunicación y comercialización para las sucursales de su empresa.
- ✦ Charlar con un experto o un grupo de expertos sobre algún tema en específico.
- ✦ Leer interesantes artículos en publicaciones electrónicas de todo tipo.
- ✦ Tener acceso a cientos de miles de programas para su computadora.
- ✦ Conocer las noticias más importantes de diversos países, mediante el acceso a las versiones electrónicas de los principales diarios del mundo.
- ✦ Conocer los movimientos de la bolsa.
- ✦ Participar en juegos interactivos, desde ajedrez hasta aventuras en mundos virtuales con personas de otros países.
- ✦ Intercambiar puntos de vista con otras personas en el mundo que compartan intereses similares.
- ✦ Buscar información de investigación sobre algún tema en específico.
- ✦ Dar a conocer nuestras opiniones sobre diversos asuntos de importancia mundial.
- ✦ Encontrar contactos para realizar nuevos negocios.
- ✦ Crear nuestros propios servicios de información y comercialización.
- ✦ Leer las principales obras clásicas de todos los tiempos, relatos o escritos de todo tipo desde novelas y cuentos, hasta ensayos científicos.
- ✦ Publicar nuestros propios conocimientos, investigaciones, relatos o creaciones artísticas.
- ✦ Dar a conocer nuestro negocio al mundo.
- ✦ Conocer detalles importantes sobre cualquier tema que nos interese.
- ✦ Simplemente pasar un rato entretenido curioseando en un mundo de información.

La lista anterior de ninguna manera cubre todas las posibilidades de Internet, pero debe ser suficiente para dar una idea de su utilidad al usuario potencial.

## **2.4 Servicios de Internet.**

- ☛ Archie.
- ☛ Conexión remota.
- ☛ Correo electrónico.
- ☛ Directorio de Páginas Blancas.
- ☛ Finger.
- ☛ F.T.P.
- ☛ Gopher.
- ☛ Internet Bulletin Board System.
- ☛ Internet Relay Chart.

- ☛ Jughead.
- ☛ Knowbots.
- ☛ Multiple User Dimension.
- ☛ Ping.
- ☛ Servidores Wais.
- ☛ Usenet.
- ☛ Utilidad Talk.
- ☛ Veronica.
- ☛ World Wide Web.

## **2.5 El futuro de los derechos electrónicos.**

En el centro de la historia de nuestra civilización, se encuentra la gran necesidad de comunicarse entre sí que tienen los seres humanos. De aquí la importancia de cada nueva herramienta de comunicación.

Las redes de computadoras fueron inmediatamente adoptadas por la gente para la comunicación humana, convirtiéndose en mundos virtuales -nuevos espacios para la interacción social, educativa y laboral-. Estos mundos virtuales ofrecen un nuevo y prometedor ambiente para la convivencia entre las personas.

Las promesas y los retos que plantean las comunidades virtuales son diversos y apenas comienzan a ser estudiados. La manera en que estos retos se resuelvan, definirá la naturaleza y calidad de estos nuevos espacios sociales, así como su papel y sus implicaciones. Nuestra sociedad debe estar consciente de la necesidad de participar en el proceso de formación de los mundos virtuales, de manera que las redes de comunicación permitan a la humanidad expresarse así misma como debe hacerlo.

Los sistemas legales son una pieza clave en la definición de este proceso de formación. Fueron creados para atender las necesidades específicas de comunidades cerradas, de manera que es natural que enfrenten severas dificultades para regular los nuevos medios. No es ningún secreto que el mundo se enfrenta al surgimiento de una nueva era, donde la palabra globalización es la clave.

Los paradigmas del poder y dominio en las sociedades modernas, están dirigiéndose hacia la información como herramienta principal del siglo XXI, y las sociedades deberán formarse alrededor de este nuevo principio.

Esta evolución social podría incluso llevar a la creación de una civilización que trascienda a la que tenemos ahora. Cada vez existe mayor preocupación por la situación del medio ambiente, la sobrepoblación y el hambre en todo el mundo. Es necesario encontrar nuevas formas de organización que puedan ayudarnos a enfrentar mejor estos problemas.

Una nueva civilización basada en la información y la globalización, pondría un gran valor en la colaboración y en la armonía con el medio ambiente. La orientación de esta civilización sería hacia la estabilidad y la continuidad, en lugar de hacia el progreso y desarrollo. En esta nueva civilización, es clave la colaboración entre grupos de personas a nivel local, nacional y global.

Internet y otras redes similares serán efectivamente la base de estas nuevas sociedades, las cuales no dependerán de su ubicación geográfica, sino de la interacción entre sus miembros desde cualquier parte del mundo.

Esta clara tendencia hacia la globalización y la eliminación de las fronteras, pone en evidencia el anacronismo de los actuales aparatos legales. De hecho, desde el final de la primera guerra mundial se ha buscado la creación de una liga de naciones que determine la justicia en el mundo.

No es la infabilidad de los sistemas actuales de gobierno, justicia y toma de decisiones lo que los mantiene vivos, sino el conflicto de intereses que se oponen a la idea de perder el dominio de su parte del mundo. Muchas naciones temen perder su soberanía al perder el control total del intercambio de información a través de sus fronteras. Otras temen a la pérdida de su identidad cultural e integridad étnica, o a la dificultad que podría representar un sistema global para el control de su futuro económico.

A pesar de estos temores, la presión para formar una comunidad global integrada es muy poderosa. Muchas naciones, grandes y pequeñas, desarrolladas y subdesarrolladas, están reestructurando sus sistemas de telecomunicaciones para adaptarse a un ambiente internacional global. El hecho es que una sociedad requiere de aparatos legales globales y no locales.

La creación de distintos códigos legales para la regulación de los nuevos medios electrónicos de comunicación, por bien logrados que resulten, no es sino una solución a corto plazo para una serie de problemas que muy pronto rebasarán la competencia de un solo estado o nación. Es necesario que los distintos países se den cuenta de la dirección que lleva la humanidad y actúen en conjunto para resolver este problema.

La creación de un tratado internacional o una carta de derechos para determinar y proteger los derechos electrónicos en beneficio de la humanidad, sería un paso importante para entrar al nuevo siglo con el pie derecho. Todos los países del mundo deberían participar en esta iniciativa para proteger los derechos de sus ciudadanos. El tipo de sociedad, basado en la cooperación, que el modelo de redes pronostica, es el ambiente cultural ideal para la realización de un esfuerzo como este.

Gracias a las comunidades electrónicas, muchas personas están adquiriendo una consciencia de ciudadanos del mundo, e incluso la gente más apática resulta participativa al interactuar con los nuevos medios. Internet es un medio donde la gente que se quiere hacer escuchar puede hacerlo, y donde la solidaridad puede crecer entre habitantes de distintos países en cuestión de minutos. Cada vez un mayor número de personas utilizan estas nuevas tecnologías y aprecian las facilidades y las oportunidades de comunicación y cooperación que ofrecen como medio complementario de expresión humana.

Cualquier intento por legislar los nuevos medios deberá tomar en cuenta estas características. No solo se trata de regular el uso de una tecnología, sino de establecer normas para la comunicación humana y formar las bases de las futuras comunidades electrónicas.

Existen determinados derechos humanos que no necesitan ser contemplados específicamente en las constituciones de los distintos países para que se espere que sean respetados por todos. Hay una carta de derechos humanos e incluso una comisión creada específicamente para proteger los mismos.

En la nueva era de la información que se acerca, los derechos electrónicos pasarán a ser parte importante de estos derechos básicos, porque afectarán a todo el que quiera participar en una sociedad global.

Las siguientes tres categorías de los derechos de procesamiento y transmisión de la información, formarían la base de esta carta de derechos electrónicos:

- ✦ *Derecho a la autonomía y seguridad en la información.* Tanto los individuos, como miembros de un grupo, necesitan el derecho de autonomía en el procesamiento de la información. El individuo debe tener el derecho de rechazar los intentos de terceros por intervenir o influenciar, sin su autorización, su proceso de toma de decisiones. Esto incluye, pero no se limita, a actos de interferencia como la censura, el correo electrónico "en masa", los virus computacionales y la entrada no autorizada a su sistema.
  
- ✦ *Derecho a la propiedad y prioridad en la información.* Los individuos y los grupos deben tener la seguridad de que cualquier información nueva, creada o descubierta por ellos en el proceso de manejo de información, pertenezca a ellos por naturaleza. Deben ser considerados como los originadores de dicha información y tener el derecho de exigir que otros den el crédito debido o soliciten el permiso necesario, si es que quieren transmitir esa información a un tercero.
  
- ✦ *Derecho al control y privacidad de la información.* Los individuos y los grupos deben tener ciertos privilegios de control sobre la recopilación, procesamiento y uso de toda información relacionada con ellos mismos, y por lo tanto el derecho de exigir que otros les informen que clase de información guardan sobre ellos y de evitar que esta sea usada de formas que no aprueben.

En la futura sociedad de la información, estos derechos deben estar propiamente establecidos y limitados, de manera que sea posible acelerar el proceso global de intercambio de información y conocimiento. Es el papel de todas las personas interesadas en la información de este tipo de sociedad, luchar por una pronta definición y limitación de los derechos de la información.

## **2.6 Perspectivas de Internet.**

Hoy en día, algunos de los corporativos mundiales ya consideran a Internet como la llave del éxito para los negocios del futuro. Además, algunas de esas empresas afirman que aquellos corporativos que hoy en día no aprovechen la red, podrían llegar a enfrentar agudos problemas de supervivencia a largo plazo.

Los posibles usos de Internet para las empresas son varios: es utilizada en sus actividades de logística corporativa, comunicaciones internas y externas, planeación estratégica, reducción de costos, colaboración con otras compañías en determinados proyectos, desarrollo de productos, obtención y utilización de información, mercadotecnia, ventas, transmisión de información y presencia corporativa.

Es importante destacar que para que una empresa efectivamente logre mejores ventajas de Internet, es necesario que adopte una actitud innovadora en la solución de sus problemas de servicio, ventas y mercadotecnia. Esa actitud deberá sustentarse en la creación de valores orientados hacia el desarrollo de la información. Tal actitud además supone que la empresa no limitará el uso de Internet a sus clientes; sino que compartirá con los empleados información útil, detallada, actualizada y constante. Esa información deberá presentarse conforme a las principales características de Internet, es decir, interacción y profundidad.

Las grandes compañías no son las únicas que aprovechan los beneficios de Internet. De hecho, la pequeña y mediana empresa son las que pueden resultar más favorecidas por el uso de la red. Internet y su alcance global les permiten, por primera vez, tener una efectiva presencia global y una mayor participación en el mercado, sin la necesidad de realizar cuantiosos gastos publicitarios. Internet está haciendo posible no solamente el intercambio de información o de programas, sino la creación de nuevos conceptos capaces de transformar todos los ámbitos de nuestra vida diaria, desde la educación y la ciencia, hasta la diversión y el entretenimiento, y por supuesto, los negocios.

El dinero electrónico es ya una realidad, mediante lo que es posible realizar transacciones a distancia fácilmente y con poco riesgo. Pronto habrá docenas de "corporaciones virtuales", empresas que físicamente no existen, sino que se hacen presentes en el mundo mediante Internet. Estas empresas pueden formarse a través de la cooperación de diversas empresas o individuos y funcionar constantemente o sólo por la duración de un proyecto.

## **2.7 Internet en México.**

Actualmente más de 8,000 computadoras están conectadas directamente a Internet en el territorio nacional. Esta cifra representa apenas el 0.16 por ciento del total de computadoras que tiene Estados Unidos. Sin embargo, si se compara a México con el resto de América Latina, nuestra nación es líder en la región en materia de tráfico de información, inclusive por encima de Brasil, que es el país que actualmente cuenta con el mayor número de redes de computadoras conectadas a Internet en Centro y Sudamérica.

Las instituciones educativas son los principales usuarios de Internet en México, pero poco a poco se incrementa el número de usuarios individuales y comerciales que tienen acceso a la información contenida en Internet, y que aprovechan sus ventajas.

De acuerdo a los principales expertos de la publicidad, en tiempos de recesión no es conveniente retirar la promoción de los productos. En esas situaciones y considerando las principales características de los artículos, resulta indispensable mantener una presencia positiva en la mente del consumidor.

En México, el progresivo encarecimiento que año con año registran las tarifas de los medios publicitarios convencionales, y principalmente las de la televisión abierta, restringe las posibilidades promocionales de un considerable número de productos. Sin embargo, el alentador desarrollo de algunas de las nuevas tecnologías en materia de comunicaciones -como precisamente el caso de Internet- permite considerar el acceso a una promoción de carácter mundial, mediante la más avanzada tecnología de punta y a bajos costos para los anunciantes.

Por medio de Internet ya es posible dar a conocer los productos y servicios de empresarios mexicanos, a millones de personas en más de 160 países. Internet, más que la mayor red de cómputo del mundo, es en realidad un nuevo medio de comunicación, el cual, en un futuro no muy distante seguramente será capaz de asumir algunas de las principales funciones que hoy en día desempeñan algunos de los medios de difusión colectiva convencionales.

Una de las principales ventajas de Internet radica en su formidable capacidad para proyectar una imagen definida y uniforme de los productos que se publican en sus "aparadores virtuales".

## **Capítulo 3.**

### **3.1 Tecnologías subyacentes de red.**

Es importante entender que Internet no es un nuevo tipo de red física, sino un método de interconexión de redes físicas y un conjunto de convenciones para el uso de redes, que permite a las computadoras conectadas a estas interactuar unas con otras. Si bien el hardware de las redes desempeña un papel de menor importancia en el diseño total, entender la tecnología utilizada para enlazar redes de redes requiere de la distinción, entre sus mecanismos de bajo nivel proporcionados por el hardware mismo y la infraestructura de alto nivel proporcionada por el software de protocolo de TCP/IP.

También es importante entender cómo la infraestructura proporcionada por la tecnología de conmutación de paquetes afecta nuestra selección de las abstracciones de alto nivel.

### **3.2 Dos enfoques de la comunicación por red.**

Si se realiza una conexión entre una computadora y otra, o entre terminales y computadoras, la comunicación entre redes puede dividirse en dos tipos básicos: de circuitos conmutados (a veces llamada orientada a la conexión) y por conmutación de paquetes (a veces llamada sin conexión).

Las redes de circuitos conmutados operan formando una conexión delicada (circuito) entre dos puntos. El sistema telefónico de Estados Unidos utiliza tecnología de circuitos conmutados -una llamada telefónica establece un circuito desde el teléfono que la origina a través de la oficina local de conmutación-, a través de las líneas troncales, hacia la oficina remota de conmutación y finalmente hasta el teléfono destino. Mientras este circuito se mantenga, el equipo telefónico tomará muestras del micrófono continuamente, codificará las muestras en forma digital y las transmitirá a través del circuito hasta el receptor. El emisor garantiza que las muestras pueden ser enviadas y reproducidas dado que los circuitos proporcionan una trayectoria de envío de datos de 64 Kbps (miles de bits por segundo), esta es la cifra necesaria para garantizar el envío de la voz digitalizada.

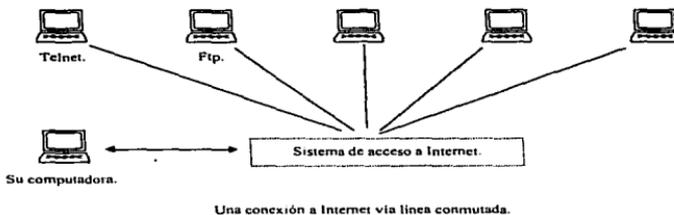
La ventaja de los circuitos conmutados reside en su capacidad garantizada: una vez que un circuito se establece, ninguna otra actividad de la red se verá disminuida en su capacidad.

Una desventaja de los circuitos conmutados es el costo: el costo de un circuito es fijo, independientemente del tráfico. Por ejemplo, se debe pagar una cuota fija por una llamada telefónica, sin importar si las dos partes que se comunicaron hablaron o no en todo momento.

Las redes de conmutación de paquetes, normalmente utilizadas para conectar computadoras funcionan de una manera por completo diferente. En una red de conmutación de paquetes, la información es transferida a través de la red dividida en pequeñas unidades llamadas paquetes, que son multiplexadas en conexiones entre máquinas de alta capacidad. Un paquete por lo general contiene sólo unos cuantos cientos de octetos de datos y transporta información de identificación, que permite al hardware de la red saber cómo enviar el paquete hacia un destino específico.

El hardware de red envía los paquetes al destino especificado, donde el software los reensamblará de nuevo en un solo archivo. Una gran desventaja de la conmutación de paquetes, es que comunicaciones múltiples entre computadoras pueden procesarse de manera concurrente, con conexiones entre máquinas compartidas por todos los pares de máquinas que se están comunicando. La desventaja, por supuesto, es que si la actividad se incrementa, un par de computadoras que se estén comunicando en un momento dado dispondrán de una menor capacidad de la red. Esto significa, que cada vez que una red de conmutación de paquetes se sobrecarga, las computadoras que están utilizando la red deberán esperar para poder continuar enviando paquetes.

A pesar de las dificultades potenciales que no garantizan la capacidad de la red, las redes de conmutación de paquetes se han vuelto muy populares. Los motivos para adoptar la conmutación de paquetes son el costo y el desempeño. Dado que las múltiples máquinas pueden compartir el hardware de red, se requiere de pocas conexiones y el costo se reduce.



### 3.3 Redes de área amplia y redes de área local.

Las redes de conmutación de paquetes que deben recorrer distancias geográficas grandes, son fundamentalmente diferentes de las que deben recorrer distancias cortas. Para ayudar a caracterizar las diferencias en la capacidad y las proyecciones de uso, la tecnología de conmutación de paquetes se divide con frecuencia en dos grandes categorías: Wide Area Networks (redes de área amplia) y Local Area Networks (redes de área local).

La tecnología WAN, a veces llamada *long haul networks* (redes de gran alcance), proporcionan comunicación que cubre grandes distancias. Muchas tecnologías WAN no tienen un límite de distancia de recorrido; una WAN puede permitir que dos puntos inmediatamente lejanos se comuniquen. Comúnmente, las WAN operan más lentamente que las LAN y tienen tiempos de retraso mucho mayores entre las conexiones. La velocidad normal para una WAN llega a un rango que va de los 56 Kbps a 155 Mbps (millones de bits por segundo). Los retardos para una WAN pueden variar de unos cuantos milisegundos a varias decenas de segundos (estos retardos se deben a que las WAN se comunican por medio de envío de señales a los satélites en órbita alrededor de la Tierra).

Las tecnologías LAN proporcionan las velocidades de conexión más altas entre computadoras, pero sacrifican la capacidad de recorrer largas distancias. Por ejemplo, una LAN común recorre una área pequeña, como un edificio o un pequeño campus, y opera dentro de un rango que va de los 10 Mbps a los 2 Gbps (billones de bits por segundo).

Debido a que la tecnología LAN cubre distancias cortas, ofrece tiempos de retraso mucho menores que las WAN. Los tiempos de retardo en una LAN pueden ser más cortos, como unas cuantas decenas de milisegundos, o largos, 10 milisegundos.

Ahora podemos mencionar un principio general de la relación entre la velocidad y la distancia: las tecnologías que proporcionan altas velocidades de comunicación operan en distancias cortas. Existen otras diferencias entre las tecnologías de las categorías señaladas. En la tecnología LAN cada computadora por lo general contiene un dispositivo de interfaz de red que conecta la máquina directamente con el medio de la red (por ejemplo, un alambre de cobre o cable coaxial). Con frecuencia la red es por sí misma es pasiva, depende de los dispositivos electrónicos conectados a las computadoras para generar y recibir las señales eléctricas necesarias. En la tecnología WAN, una red por lo común consiste en una serie de computadoras complejas, llamadas conmutadoras de paquetes, interconectadas por líneas de comunicación y módems. El tamaño de la red puede extenderse si se le añade un nuevo conmutador y otras líneas de comunicación. La conexión de una computadora de usuario a una WAN significa conectarla a uno de los conmutadores de paquetes. Cada conmutador extiende la ruta de la WAN e introduce un retardo cuando recibe un paquete y lo envía al siguiente conmutador. De esta manera, la extensión de una WAN hace que la ruta del tráfico que pasa a través de ella se extienda.

### 3.4 Ancho de Banda.

Una característica muy importante de un enlace a Internet es la velocidad de la transferencia de información. Un ancho de banda es la capacidad de transmisión de una computadora o de un canal de comunicaciones. La transmisión digital pura se mide en bits o bytes por segundo. Cuando los datos digitales son convertidos a frecuencias para transmisión por redes basadas en ondas portadoras, el ancho de banda todavía se expresa en bits o bytes por segundo; sin embargo, cuando se utilizan frecuencias como la señal de transmisión en vez de pulsos ON/OFF, el ancho de banda puede también medirse en ciclos por segundo o Hertz.

El ancho de banda es la diferencia existente entre la mínima y la máxima frecuencia transmitida. Un enlace satelital tiene un ancho de banda de 64,000 bps., o 64 Kbps.; un enlace a través de fibra óptica, puede sostener velocidades desde 64 Kbps., hasta 2,000 Kbps., de manera que proporciona el mayor ancho de banda.

El problema de ancho de banda es uno de los principales obstáculos para el desarrollo de las tecnologías como videoconferencias y servicios de video por Internet. Aún las conexiones a través de fibra óptica no bastan para transmitir toda la información que requiere un video de pantalla completa en un tiempo razonable.

La siguiente tabla muestra las opciones más comunes de anchos de banda disponibles para proveedores de servicios alrededor de Estados Unidos.

Nombre.	Bits por segundo.	Bytes por segundo.
9.6 Kbps.	9600	1,067
14.4 Kbps.	14,400	1,600
19.2 Kbps.	19,200	2,134
28.8 Kbps.	28,800	3,200
38.4 Kbps.	38,400	4,267
56 Kbps.	57,344	7,168
64 Kbps.	65,535	8,192
128 Kbps.	131,072	16,384
T1 (1.5 Mbps.).	1,536,000	192,000
T3/DS3 (45 Mbps.).	46,080,000	5,760,000

Todas las líneas de gran capacidad tienen nombres. Aquí están unos de los más populares:

- ↪ *T1, o DS-1.*- Este es el equivalente de 24 líneas telefónicas normales de multiplexaje o canales. La capacidad es de 1.5 millones de bits por segundo (1.5 Mbps.).
- ↪ *T2, o DS-2.*- Este es el equivalente de 4 multiplexajes, canales T1. La capacidad es de 6.3 millones de bits por segundo (6.3 Mbps.).
- ↪ *T3, o DS-3.*- Este es el equivalente de 28 multiplexajes, canales T1. La capacidad es de 45 millones de bits por segundo (45 Mbps.).
- ↪ *T4, DS-4.*- Este es el equivalente de 6 multiplexajes, canales T3. La capacidad es de 274 millones de bits por segundo (274 Mbps.).

Aquí están otros protocolos que ahora están disponibles:

- ↪ *Frame Relay.*- Este protocolo es usado para la transmisión de datos sobre WAN's. Los paquetes son enviados en tramas de datos que varían en tamaño desde 7 a 1024 bytes. Frame Relay es principalmente un protocolo de Ancho de Banda.
- ↪ *ISDN.*- ISDN, propuesto por la Red Digital de Servicios Integrados -Integrated Services Digital Network. ISDN-, no es un protocolo como un estándar. Básicamente, una línea ISDN es un canal de 128 kbps., que está dividido en tres partes: dos canales portadores (o "B") y un canal delta (o "D"). Este es considerado una interfaz de velocidad básica -Basic Rate Interface, BRI-. Los canales B, son usados para voz o datos (arriba de 64 kbps. por canal) y el canal D es usado para transmisión e intercambio de paquetes. ISDN es el nuevo medio de comunicación.

### **3.5 Direcciones de hardware de red.**

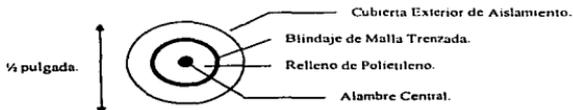
Cada tecnología de hardware de red define un mecanismo de direccionamiento que las computadoras utilizan para especificar el destino de cada paquete. A cada computadora conectada a una red se le asigna una dirección única, por lo general un número entero.

Un paquete enviado a través de una red incluye un campo de dirección de destino que contiene la dirección del recipiente del destino. La dirección del destino aparece en el mismo lugar en todos los paquetes haciendo posible que el hardware de red localice la dirección del destino fácilmente. Cuando se envía información se debe conocer la dirección del recipiente de destino y colocar la dirección del recipiente en el campo de dirección de destino del paquete, antes de transmitir el paquete.

Cada tecnología de hardware específica cómo las computadoras son asignadas a una dirección. El hardware específica, por ejemplo, el número de bits en la dirección así como la localización del campo de dirección de destino de un paquete. Aun cuando algunas tecnologías utilizan esquemas de direccionamiento compatibles, muchas no lo hacen así.

### 3.6 Tecnología Ethernet.

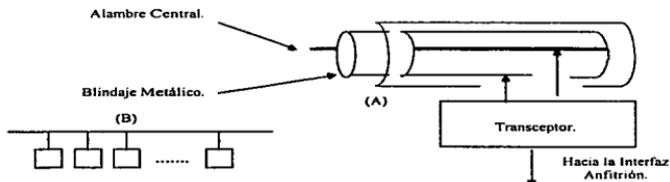
Ethernet, es el nombre que se le ha dado a una popular tecnología LAN de conmutación de paquetes inventada por Xerox PARC a principios de los años setenta. Xerox Corporation, Intel Corporation y Digital Equipment Corporation estandarizaron Ethernet en 1978; IEEE liberó una versión compatible del estándar utilizando el número 802.3. Ethernet se ha vuelto una tecnología LAN popular; muchas compañías, medianas o grandes, utilizan Ethernet. Dado que Ethernet es muy popular existen muchas variantes, analizaremos el diseño original primero y después cubriremos algunas variantes. Cada cable Ethernet tiene aproximadamente  $\frac{1}{2}$  pulgada de diámetro y mide hasta 500 metros de largo. Se añade una resistencia entre el centro del cable y el blindaje en cada extremo del cable para prevenir la reflexión de señales eléctricas.



Sección transversal del cable coaxial utilizado en la red Ethernet original.

El diseño original de Ethernet utilizaba un cable coaxial como el mostrado en la figura. Llamado ether, el cable por sí mismo es completamente pasivo; todos los componentes electrónicos activos que hacen que la red funcione están asociados con las computadoras que se comunican a la red.

La conexión entre una computadora y un cable coaxial Ethernet requiere de un dispositivo de hardware llamado transceptor. Físicamente, la conexión entre un transceptor y el cable Ethernet requiere de una pequeña perforación en la capa exterior del cable como se muestra en la siguiente figura.



(A) Vista recortada de un cable de red Ethernet en la que se muestran los detalles de las conexiones eléctricas entre un transceptor y el cable.

(B) Diagrama esquemático de una red Ethernet con varias computadoras conectadas.

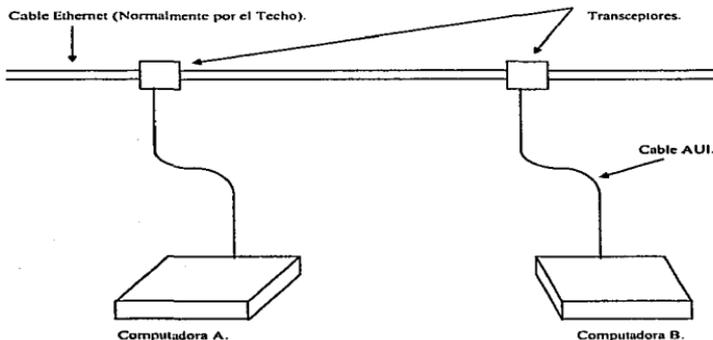
Cada conexión a una red Ethernet tiene dos componentes electrónicos mayores. Un transceptor es conectado al centro del cable y al blindaje trenzado del cable, por medio del cual recibe y envía señales por el cable ether. Una interfaz anfitrión o adaptador anfitrión se conecta dentro del bus de la computadora (por ejemplo, en una tarjeta madre) y se conecta con el transceptor.

Un transceptor, es una pequeña pieza de hardware que por lo común se encuentra físicamente junto al cable ether. Además del hardware análogo que envía y controla las señales eléctricas en el cable ether, un transceptor contiene circuitería digital que permite la comunicación con una computadora digital. El transceptor, cuando el cable ether está en uso; puede recibir y traducir señales eléctricas analógicas, hacia o desde un formato digital en el cable ether.



Aún cuando el transceptor es un simple dispositivo de hardware, la interfaz de anfitrión puede ser compleja (por ejemplo, puede contener un microprocesador utilizado para controlar la transferencia entre la memoria de la computadora y el cable ether).

En la práctica las organizaciones que utilizan el Ethernet original en el ambiente de una oficina convencional extienden el cable Ethernet por el techo de las habitaciones e instalan una conexión para cada oficina conectándola de este modo con el cable. La siguiente figura ilustra el esquema de cableado físico resultante.



**Conexión física de dos computadoras a una red Ethernet mediante el uso del esquema de cableado original. En el ambiente de una oficina, el cable Ethernet por lo general se coloca formando una trayectoria en el techo; cada oficina tiene un cable AUI que conecta una computadora en la oficina con el transceptor conectado al cable Ethernet.**

### 3.7 Ethernet de cable delgado.

Varios componentes de la tecnología Ethernet original tenían propiedades indeseables. Por ejemplo, un transeceptor contenía componentes electrónicos, su costo no era insignificante. Además, ya que el transeceptor estaba localizado en el cable y no en la computadora, estos podían ser difíciles de acceder o reemplazar. El cable coaxial que forma el ether puede ser difícil de instalar. En particular, para proporcionar la máxima protección contra la interferencia eléctrica, el cable contiene un blindaje pesado que hace que el cable sea difícil de doblar. Por último un cable AUJ también es grueso y difícil de doblar.

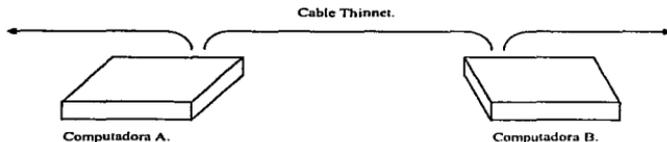
Para reducir costos en el caso de ambientes como el de las oficinas, en donde existe mucha interferencia eléctrica, los ingenieros desarrollaron una alternativa de esquema de cableado Ethernet.

Llamada *thin wire Ethernet* o *thinnet*, el cable coaxial alternativo es más delgado, menos caro y más flexible. Sin embargo, un cable delgado Ethernet tiene algunas desventajas. Dado que no proporciona mucha protección contra la interferencia eléctrica, el cable delgado Ethernet no puede ser colocado junto a equipo eléctrico potente, como podía suceder en el caso de una fábrica. Además, el cable delgado Ethernet cubre distancias algo más cortas y soporta un menor número de conexiones de computadoras por red que el cable grueso Ethernet.

Para reducir aún más los costos con el cable delgado Ethernet, los ingenieros reemplazaron el costoso transeceptor con circuitería digital de alta velocidad especial y proporcionaron una conexión directa desde una computadora hasta el cable ether. De esta forma, en el esquema de cable delgado, una computadora contiene tanto la interfaz de antirtrión como la circuitería necesaria para conectar la computadora con el cable. Los fabricantes de pequeñas computadoras y estaciones de trabajo encontraron el esquema del cable delgado Ethernet especialmente atractivo, debido a que podían integrar el hardware de Ethernet en una sola tarjeta de computadora y hacer las conexiones necesarias de manera directa en la parte posterior de la computadora.

Como el cable delgado Ethernet conecta directamente una computadora con otra, el esquema de cableado trabaja bien cuando muchas computadoras ocupan una sola habitación. El cable delgado conecta en forma directa una computadora con otra. Para añadir una nueva computadora, sólo es necesario enlazarla con la cadena.

En la siguiente figura se ilustra la conexión utilizada en el esquema de cable delgado de Ethernet.



Conexión física de dos computadoras que se valen del esquema de cableado thinnet (cable de red delgado). El cable ether pasa directamente de una computadora a otra, no requiere del hardware de transceptores externos.

El esquema de cable delgado de Ethernet está diseñado para conectarse y desconectarse fácilmente. El esquema de cable delgado utiliza conectores BNC, los cuales no requieren de herramientas para conectar una computadora con el cable. Así, un usuario puede conectar una computadora al cable delgado Ethernet sin ayuda de un técnico. Por supuesto, permitir que el usuario manipule el cable ether tiene sus desventajas: si un usuario desconecta el cable ether, provocará que todas las máquinas en el ether queden incommunicadas. En muchos casos, sin embargo, las ventajas superan a las desventajas.

### 3.8 Ethernet de par trenzado.

Los avances en la tecnología han hecho posible construir redes Ethernet que no necesitan del blindaje eléctrico de un cable coaxial. Llamada *twisted pair Ethernet* (Ethernet de par trenzado), esta tecnología permite que una computadora accese una red Ethernet mediante un par de cables de cobre convencionales sin blindaje. La ventaja de usar cables de par trenzado, es que reducen mucho los costos y protegen a otras computadoras conectadas a la red de los riesgos que se podrían derivar de que un usuario desconecte una computadora. En algunos casos la tecnología de par trenzado hace posible que una organización instale una red Ethernet a partir del cableado telefónico existente sin tener que añadir cables nuevos.

Conocido con el nombre técnico de *10Base-T*, el esquema de cableado de par trenzado conecta cada computadora con un *hub* (concentrador), Ethernet como se muestra en la siguiente figura.

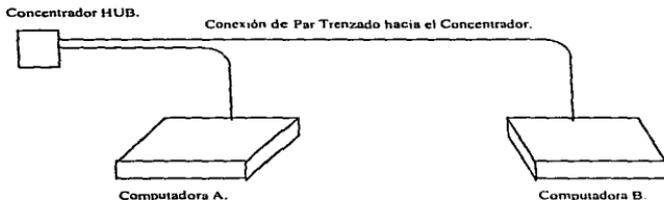


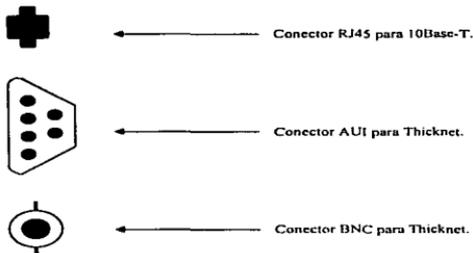
Ilustración de una red Ethernet que emplea cableado de par trenzado. Cada computadora se conecta a un concentrador mediante un par de cables convencionales.

El concentrador, es un dispositivo electrónico que simula la señal en un cable Ethernet. Físicamente, un concentrador consiste en una pequeña caja que por lo general se aloja en un gabinete para cableado; la conexión entre un concentrador y una computadora debe tener una longitud menor a 100 metros. Un concentrador requiere de alimentación eléctrica y puede permitir que el personal autorizado monitoree y controle la operación de la red. Para la interfaz anfitrión, en una computadora, la conexión hacia un concentrador parece operar de la misma forma que la conexión hacia un transceptor. Esto es, un concentrador Ethernet proporciona la misma capacidad de comunicación que un Ethernet delgado o grueso; los concentradores sólo ofrecen una alternativa al esquema de cableado.

### 3.9 Esquemas de cableado múltiple y adaptadores.

Una conexión Ethernet de cable grueso requiere de un conector *AUI*, una conexión para Ethernet de cable delgado requiere de un conector *BNC* y un conector para 10Base-T requiere de un conector *RJ45*. Muchos productos Ethernet, permiten que cada usuario seleccione el esquema de cableado.

Por ejemplo, las tarjetas adaptadoras para computadoras personales con frecuencia cuentan con los 3 conectores como se muestra en la siguiente figura.



Una tarjeta adaptadora Ethernet común con los tres conectores para los tres esquemas de cableado Ethernet. Aun cuando el adaptador contiene tres tipos de conector, sólo se puede utilizar un esquema de cableado a la vez.

Dado que un solo conector puede usarse a la vez, una computadora que cuenta con un adaptador determinado puede cambiarse de un esquema de cableado a otro con facilidad.

### **3.10 Capacidades de las redes Ethernet.**

El estándar Ethernet se define en 10 Mbps, lo cual significa que los datos pueden transmitirse por el cable a razón de 10 millones de bits por segundo. A pesar de que una computadora puede generar datos a la velocidad de la red Ethernet, la velocidad de la red no debe pensarse como la velocidad a la que dos computadoras pueden intercambiar datos. La velocidad de la red debe ser pensada, como una medida de la capacidad del tráfico total de la red. Una red Ethernet a 10 Mbps, por ejemplo, puede soportar unas cuantas computadoras que generan cargas pesadas o muchas computadoras que generan cargas ligeras.

### ***3.11 Interconexión de datos distribuida por fibra (FDDI).***

FDDI, es una tecnología de red de área local muy popular que proporciona un ancho de banda mayor que las redes Ethernet. A diferencia de las redes Ethernet y otras tecnologías LAN que utilizan cables para transportar las señales eléctricas, en la tecnología FDDI se utilizan fibras de vidrio y se transfiere la información codificada en pulsos de luz.

La fibra óptica tiene dos ventajas con respecto a los cables de cobre. En primer lugar, como el ruido eléctrico no interfiere con una conexión óptica, la fibra se puede colocar junto a los dispositivos eléctricos de potencia. En segundo lugar, dado que las fibras ópticas utilizan luz; la cantidad de datos que pueden enviarse por unidad de tiempo, es mucho mayor que en los cables que transportan señales eléctricas.

Podría parecer que las fibras de vidrio son difíciles de instalar y se rompen fácilmente. Sin embargo, un cable óptico posee una flexibilidad sorprendente. La fibra de vidrio, por sí misma tiene un diámetro muy pequeño y el cable incluye una cubierta plástica que protege a la fibra de las rupturas. El cable no se puede doblar en un ángulo de 90° pero se puede doblar en un arco con un diámetro de unas cuantas pulgadas.

### ***3.12 Propiedades de una red FDDI.***

Una red FDDI es una tecnología token ring a 100 Mbps con una capacidad de auto reparación. Una red FDDI es un ring (anillo), dado que la forma un ciclo que comienza desde una computadora, pasa a través de todas las demás computadoras y terminan en el mismo punto en que se inició. La FDDI es una tecnología token ring, porque utiliza un token (o prenda) para controlar la transmisión. Cuando la red está desocupada, una trama especial llamada token pasa de una estación a otra.

Cuando una estación tiene un paquete para enviar, espera a que llegue el token, envía el paquete y entonces, transfiere el token a la siguiente estación. La circulación del token garantiza la equidad: asegura que todas las estaciones tengan una oportunidad para enviar un paquete, antes de que cualquier estación envíe un segundo paquete.

Tal vez la propiedad más interesante de una FDDI reside en su capacidad para detectar y corregir problemas. La red se conoce como red con capacidad de autoreparación ya que el hardware puede adaptarse de manera automática a las fallas.

### 3.13 Anillos dobles de rotación contraria.

Para proporcionar una reparación automática de fallas, el hardware FDDI utiliza dos anillos independientes estando ambos conectados a cada computadora.

La siguiente figura ilustra lo dicho:



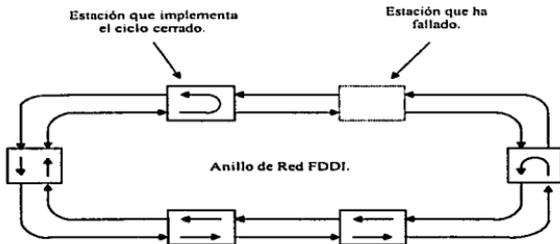
Red FDDI que interconecta seis computadoras con fibras ópticas. Las flechas muestran la dirección del tráfico en las fibras y a través de las computadoras conectadas.

Los anillos FDDI son conocidos como anillos de rotación contraria, dado que el tráfico circula en direcciones opuestas en cada anillo. La razón para utilizar la rotación en sentidos opuestos se hace clara si consideramos como el FDDI maneja las fallas.

A menos que se presente un error, el hardware FDDI no necesita ambos anillos. De hecho, una interfaz FDDI se comporta como cualquier interfaz de red que transfiere un token hasta que se presenta un error. La interfaz examina todos los paquetes que circulan en el anillo, comparando la dirección de destino y la dirección de la computadora en cada paquete. La interfaz toma una copia de cualquier paquete destinado a la computadora local, pero también envía el paquete hacia el anillo.

Cuando una computadora necesita transmitir un paquete espera la llegada del token, temporalmente deja de enviar bits, y envía el paquete. Luego de enviar un paquete, la interfaz transmite el token y comienza a enviar bits de nuevo. Si una estación tiene más de un paquete listo para ser enviado cuando recibe el token, la estación sólo enviará un paquete antes de pasar el token. Por lo tanto, el esquema de token circulante garantiza que todas las estaciones tengan un acceso franco hacia la red.

El hardware FDDI comienza a ser más interesante cuando ocurre un error en hardware. Cuando una interfaz detecta que no se puede comunicar con la computadora adyacente, la interfaz utiliza el anillo de respaldo para derivar la transmisión y evitar la falla. Por ejemplo, la siguiente figura muestra un anillo FDDI en el cual una interfaz ha fallado y las dos interfaces adyacentes la han suprimido del anillo.



Anillo FDDI después de una falla. Cuando el hardware FDDI detecta una falla, utiliza el segundo anillo para derivar el tráfico y permitir que las estaciones restantes se comuniquen.

El propósito del segundo anillo y la razón por la que la información fluye en dirección opuesta se aclara ahora: una falla puede significar que la fibra ha sido desconectada (por ejemplo, si se cortó accidentalmente). Si la fibra en ambos anillos sigue el mismo trayecto físico, la posibilidad de que la segunda fibra también haya sido desconectada será muy alta.

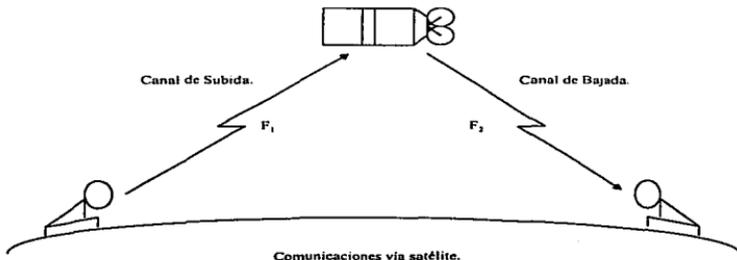
El hardware FDDI de manera automática utiliza el anillo en rotación opuesta para formar un ciclo cerrado en la dirección que aun se mantiene trabajando. Esto permite que las otras computadoras continúen comunicándose a pesar de la falla.

*Cuando el hardware FDDI detecta una falla en la red, automáticamente dirige la información hacia el anillo de respaldo para permitir la comunicación entre las estaciones restantes.*

### 3.14 Microondas.

La transmisión por microondas, transporta la información transmitida en forma de ondas de radio a través de la atmósfera, por lo que no necesita de cable. Puesto que este sistema no necesita cable, es muy adecuado donde el tendido de cables es muy difícil o caro. Sin embargo, como las señales viajan a través de la atmósfera, estas resultan afectadas por las condiciones meteorológicas y los edificios; y por tanto, necesitan espacio libre en línea recta entre las torres de repetición.

En comunicaciones vía satélite, se emplean antenas de microondas para recibir las señales de radio procedentes de las estaciones emisoras en la Tierra y para devolver estas señales a otras estaciones terrenas. La capacidad que posee el satélite de recibir y retransmitir se debe a un dispositivo conocido como transpondedor. Los transpondedores de satélite trabajan a frecuencias muy elevadas, generalmente en la banda de gigahertzios.



Como se observa en la figura anterior, la señal que transmite la estación terrestre tiene distinta frecuencia que la que devuelve el satélite. De esta manera se impide que los canales de subida y bajada se interfieran, ya que trabajan en bandas de frecuencia diferentes.

La característica de transmisión por microondas plantea serios problemas de seguridad, ya que cualquier estación puede captar las transmisiones de una empresa con sólo sintonizar la frecuencia del satélite. Para evitarlo, muchas compañías de comunicaciones por satélite añaden a sus sistemas medidas adicionales de seguridad, como el cifrado de sus transmisiones.

### ***3.15 Tecnología ATM.***

Modo de Transferencia Asíncrono -Asynchronous Transfer Mode, ATM-, es el nombre dado a una tecnología de red orientada a la conexión de alta velocidad; que ha sido utilizada tanto en redes de área local como en redes de área amplia. Para los estándares actuales, alta velocidad se refiere a las redes que operan a 100 Mbps o más; el ATM puede conmutar datos a velocidades en gigabits (Un gigabit es igual a 1,000 millones de bits por segundo). Por supuesto, cada red de alta velocidad requiere de equipo complejo y de vanguardia. Como resultado de ello, las redes ATM son más caras que las de otras tecnologías.

Para obtener una velocidad de transferencia alta, una red ATM utiliza técnicas de software y hardware de propósito especial. Primero, una red ATM consiste en uno o más conmutadores de alta velocidad que se conectan con cada computadora anfitrión y con los otros conmutadores ATM.

Segundo, la tecnología ATM utiliza fibra óptica para las conexiones, incluyendo las conexiones de computadoras anfitrión hacia los conmutadores ATM. La fibra óptica proporciona una razón de transferencia alta, mayor que la de los alambres de cobre; por lo común, la conexión entre un anfitrión y un conmutador ATM opera entre los 100 y los 155 Mbps.

Tercero, las capas más bajas de una red ATM utilizan tramas de tamaño fijo llamadas celdas. Dado que cada celda es exactamente del mismo tamaño, el hardware de conmutador ATM puede procesar las celdas con rapidez.

ATM difiere de las redes de conmutación de paquetes debido a que ofrece un servicio orientado a la conexión -connection oriented-. Antes de que una computadora anfitrión conectada a un ATM pueda enviar celdas, el anfitrión debe interactuar primero con el conmutador para especificar un destino. La interacción es análoga a la que se realiza en una llamada telefónica (Debido a que ATM fue diseñada para transportar voz, así como datos, existe una relación muy fuerte entre ATM y los conmutadores telefónicos). El anfitrión especifica la dirección de la computadora remota, y espera a que el conmutador ATM contacte el sistema remoto y establezca una ruta. Si la computadora remota rechaza la solicitud, no responde o el conmutador ATM no puede llegar a la computadora remota, la solicitud para establecer la comunicación no tendrá éxito.

Cuando una conexión se establece con éxito, el conmutador ATM local selecciona un identificador para la conexión y transfiere el identificador de conexión al anfitrión, con un mensaje que informa al anfitrión del éxito de la comunicación. El anfitrión utiliza el identificador de conexión cuando envía o recibe celdas.

Cuando se termina de usar la conexión, el anfitrión se comunica nuevamente con el conmutador ATM para solicitar que la conexión se interrumpa. El conmutador desconecta las dos computadoras. La desconexión es equivalente a "colgar" en una llamada telefónica al terminar la llamada; después de la desconexión, el conmutador puede reutilizar el identificador de conexión.

## **Capítulo 4.**

### **4.1 TCP/IP.**

Internet está constituida sobre una colección de redes que recorren el mundo. Estas redes conectan diferentes tipos de computadoras y de alguna manera, algo debe mantenerlas a todas unidas. Ese algo es TCP/IP.

Para garantizar que los diferentes tipos de computadoras puedan trabajar juntas, los programadores crean sus programas utilizando protocolos estándar. Un protocolo es una serie de reglas que describen técnicamente, cómo deben hacerse determinadas tareas.

TCP/IP es el nombre común de colección de más de 100 protocolos que nos permiten conectar computadoras y redes. El nombre de TCP/IP proviene de los dos protocolos más importantes TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*).

Dentro de Internet, la información no se transmite como una cadena continua de caracteres de host a host. Mejor que esto, los datos se transmiten en pequeños trozos de información llamados paquetes. TCP divide los mensajes en paquetes. Cada paquete se marca con un número de secuencia y con la dirección del destinatario. Además, TCP inserta determinada información de control de errores.

Estos paquetes se envían a la red, donde el trabajo de IP es transportarlos hasta el host remoto. En el otro extremo, TCP recibe los paquetes y comprueba si hay errores. Si encuentra algún error, TCP solicita que el paquete en cuestión le sea retransmitido. Una vez que todos los paquetes se han recibido de forma correcta, TCP utilizará los números de secuencia para reconstruir el mensaje original.

Partir los datos en paquetes tiene varios beneficios importantes. Primero, permite utilizar en Internet las mismas líneas de comunicación a varios usuarios diferentes al mismo tiempo. Puesto que los paquetes no tienen que viajar juntos, una línea de comunicación puede transportar tantos tipos de paquetes como ella pueda de un lugar a otro.

En su camino, los paquetes son dirigidos de host a host hasta que encuentra su último destino. Esto significa que Internet tiene una gran flexibilidad.

Si una conexión en particular está fuera de servicio, las computadoras que controlan el flujo de datos, pueden encontrar normalmente una ruta alternativa. De hecho, es posible que dentro de una misma transferencia de datos, varios paquetes sigan rutas distintas.

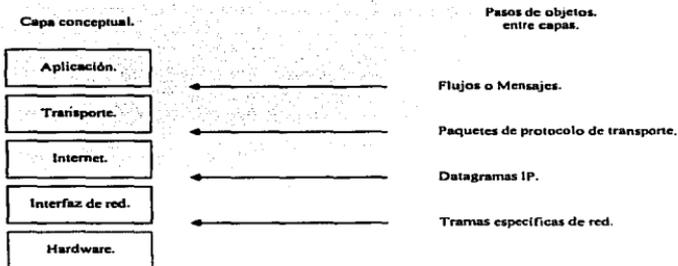
Esto también significa que, cuando las condiciones cambian, la red puede usar la mejor vía disponible en ese momento. Otra ventaja de utilizar paquetes, es que cuando algo va mal, sólo tiene que ser retransmitido un paquete, en lugar del mensaje completo. Esto incrementa de forma importante la velocidad en Internet. Toda esta flexibilidad redundante en una gran fiabilidad. De una forma o de otra, TCP/IP asegura que entrega los datos en forma correcta. En realidad, Internet funciona tan bien que pueden pasar sólo unos segundos en enviar un archivo desde un host a otro, aunque estén a miles de kilómetros de distancia y que todos los paquetes deban pasar a través de múltiples computadoras. En cierto sentido, los protocolos son para las comunicaciones lo que los algoritmos para la computación.

Desde el punto de vista de un usuario, una red de redes TCP/IP aparece como un grupo de programas de aplicación que utilizan la red para llevar a cabo tareas útiles de comunicación. Utilizamos el término de interoperabilidad, para referirnos a la habilidad que tienen diversos sistemas de computación para cooperar en la resolución de problemas computacionales. Los programas de aplicación de Internet muestran un alto grado de interoperabilidad.

La mayoría de los usuarios que acceden a Internet lo hacen al correr programas de aplicación sin entender la tecnología TCP/IP, la estructura de la red de redes subyacente o incluso sin entender el camino que siguen los datos hacia su destino; los usuarios confían en los programas de aplicación y en el software subyacente de red para manejar esos detalles. Sólo los programadores que crean los programas de aplicación de red necesitan ver a la red de redes como una red, así como entender parte de la tecnología.

#### ***4.2 El modelo de estratificación por capas de TCP/IP.***

En términos generales, el software de TCP/IP está organizado en cuatro capas conceptuales que se construyen sobre una quinta capa de hardware. La siguiente figura muestra las capas conceptuales, así como la forma en que los datos pasan entre ellas.



Las cuatro capas conceptuales del software TCP/IP y la forma en que los objetos pasan entre capas. La capa con el nombre interfaz de red se conoce con frecuencia con el nombre de capa de enlace de datos.

### **Capa de Aplicación.**

En el nivel más alto, los usuarios llaman a una aplicación que accesa servicios disponibles a través de la red de redes TCP/IP. Una aplicación interactúa con uno de los protocolos de nivel de transporte para enviar o recibir datos. Cada programa de aplicación selecciona el tipo de transporte necesario, el cual puede ser una secuencia de mensajes individuales o un flujo continuo de octetos. El programa de aplicación, pasa los datos en la forma requerida hacia el nivel de transporte para su entrega.

### **Capa de Transporte.**

La principal tarea de la capa de transporte, es proporcionar la comunicación entre un programa de aplicación y otro. Este tipo de comunicación se conoce frecuentemente como comunicación punto a punto. La capa de transporte regula el flujo de información. Puede también proporcionar un transporte confiable, asegurando que los datos lleguen sin errores y en secuencia.

Para hacer esto, el software de protocolo de transporte tiene el lado de recepción enviando acuses de recibo de retorno y la parte de envío transmitiendo los paquetes perdidos. El software de transporte divide el flujo de datos que se está enviando en pequeños fragmentos (comúnmente denominados paquetes), y pasa cada paquete, con una dirección de destino, hacia la siguiente capa de transmisión.

Aún cuando en la figura anterior, se utiliza un solo bloque para representar la capa de aplicación, una computadora de propósito general puede tener varios programas de aplicación accedendo la red de redes al mismo tiempo. La capa de transporte, debe aceptar datos desde varios programas de usuario y enviarlos a la capa del siguiente nivel. Para hacer esto, se añade información adicional a cada paquete, incluyendo códigos que identifican qué programa de aplicación envía y qué programa de aplicación debe recibir, así como una suma de verificación. La máquina de recepción utiliza la suma de verificación para comprobar que el paquete ha llegado intacto y utiliza el código destino para identificar el programa de aplicación en el que se debe entregar.

### ***Capa Internet.***

La capa Internet maneja la comunicación de una máquina a otra. Esta acepta una solicitud para enviar un paquete desde la capa de transporte, junto con una identificación de la máquina, hacia la que se debe enviar el paquete. Encapsula el paquete en un datagrama IP; llena el encabezado del datagrama, utiliza un algoritmo de ruteo para determinar si puede entregar el datagrama directamente o si debe enviarlo a un ruteador y pasar el datagrama hacia la interfaz de red apropiada para su transmisión. La capa Internet también maneja la entrada de datagramas, verifica su validez y utiliza un algoritmo de ruteo para decidir si el datagrama debe procesarse de manera local o debe ser transmitido.

Para el caso de los datagramas direccionados hacia la máquina local, el software de la capa de red de redes borra el encabezado del datagrama y selecciona de entre varios protocolos de transporte, un protocolo con el que manejará el paquete. Por último la capa Internet envía los mensajes ICMP de error y control necesarios y maneja todos los mensajes ICMP entrantes.

### ***Capa de Interfaz de Red.***

El software TCP/IP de nivel inferior consta de una capa de interfaz de red responsable de aceptar los datagramas IP y transmitirlos hacia una red específica.

### ***4.3 Tendencias de Internet: Nuevos estándares de protocolos.***

Cuando se mencionó cómo se inició Internet se habló de la Organización Internacional para la Estandarización y su conjunto de protocolos. Bien, finalmente se terminó de diseñar. Ahora es un estándar internacional, al que se le conoce comúnmente como conjunto de protocolos de Interconexión de Sistemas Abiertos -*Open System Interconnection, OSI*-. Hoy en día muchos de los componentes de Internet permiten el uso de los protocolos OSI. Todavía no existe mucha demanda. El gobierno estadounidense ha tomado una postura en la que define que todas las computadoras deben ser capaces de manejar estos protocolos. Muchos cuentan con el software, pero muy pocos lo utilizan.

Aún no está claro que tanta demanda tendrá OSI ahora que cuenta con el respaldo del gobierno de Estados Unidos. Actualmente no hay verdaderas ventajas para cambiarse a OSI. Es más complejo y menos maduro que el IP y no es tan eficaz. Ofrece algunas esperanzas en cuanto a características adicionales, pero cuando las redes crecen en velocidad y tamaño también adolecen de los mismos problemas que el IP.

### ***4.4 Jerarquías de los Protocolos.***

La mayoría de las redes se organizan en una serie de capas o niveles, con objeto de reducir la complejidad de su diseño. Cada una de ellas se construye sobre su predecesora.

El número de capas, el nombre, contenido y función de cada una varían de una red a otra. Sin embargo, en cualquier red, el propósito de cada capa es ofrecer ciertos servicios a las capas superiores, liberándolas del conocimiento detallado sobre cómo se realizan dichos servicios.

La capa *n* es una máquina que conversa con la capa *n* de otra máquina. Las reglas y convenciones utilizadas en esta conversación, se conocen conjuntamente como protocolo de la capa *n*.

En realidad no existe una transferencia directa de datos desde la capa *n* de una máquina, a la capa *n* de otra; sino, más bien, cada capa pasa la información de datos y control a la capa inmediatamente inferior, y así sucesivamente hasta que se alcanza la capa localizada en la parte más baja de la estructura. Debajo de la capa 1 está el medio físico, a través del cual se realiza la comunicación real.

Entre cada par de capas adyacentes hay una interfase, la cual define los servicios y operaciones primitivas que la capa inferior ofrece a la superior. Al conjunto de capas y protocolos se le denomina arquitectura de red.

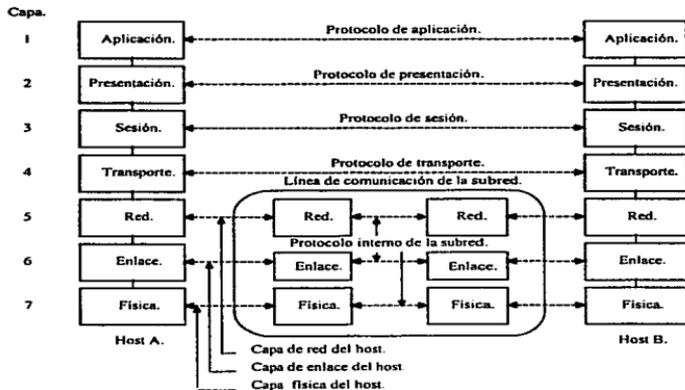
#### ***4.5 Modelo de referencia OSI.***

El modelo de referencia OSI, se refiere a la conexión de sistemas heterogéneos -es decir, a sistemas dispuestos a establecer comunicación con otros distintos-. En forma abreviada, lo llamaremos sencillamente modelo OSI.

El modelo OSI tiene siete capas. Los principios aplicados para el establecimiento de siete capas fueron los siguientes:

- ↖ Una capa se creará en situaciones donde se necesita un nivel diferente de abstracción.
- ↖ Cada capa deberá efectuar una función bien definida.
- ↖ La función que realizará cada capa, deberá seleccionarse con la intención de definir protocolos normalizados internacionalmente.
- ↖ Los límites de las capas, deberán seleccionarse tomando en cuenta la minimización del flujo de información a través de las interfaces.
- ↖ El número de capas deberá ser lo suficientemente grande para que funciones diferentes no tengan que ponerse juntas en la misma capa y, por otra parte, también deberá ser lo suficientemente pequeño para que su arquitectura no llegue a ser difícil de manejar.

El modelo OSI contiene 7 capas conceptuales organizadas de la siguiente manera:



### Capa Física.

La capa física, se ocupa de la transmisión de bits a lo largo de un canal de comunicación. Su diseño debe asegurar que cuando un extremo envía un bit con valor de 1, este se reciba exactamente como un bit con ese valor en el otro extremo, y no como un bit de valor de 0.

### ***Capa de Enlace.***

La tarea primordial de la capa de enlace consiste en, a partir de un medio de transmisión común y corriente, transformarlo en una línea sin errores de transmisión para la capa de red. Esta tarea se realiza al hacer que el emisor trocee la entrada de datos, en tramas de datos (típicamente constituidas por algunos cientos de octetos), y las transmita en forma secuencial y procese las tramas de asentimiento, devueltas por el receptor. La trama puede destruirse por completo debido a una ráfaga de ruido en la línea, en cuyo caso el software de la capa de enlace perteneciente a la máquina emisora, deberá retransmitir la trama.

### ***Capa de Red.***

La capa de red se ocupa del control de la operación de la subred. Un punto de suma importancia en su diseño, es la determinación sobre cómo encaminar los paquetes del origen al destino.

### ***Capa de Transporte.***

La función principal de la capa de transporte consiste en aceptar los datos de la capa de sesión, dividirlos siempre que sea necesario en unidades más pequeñas, pasarlos a la capa de red y asegurar que todos ellos lleguen correctamente al otro extremo.

### ***Capa de Sesión.***

La capa de sesión permite que los usuarios de diferentes máquinas puedan establecer sesiones entre ellos. A través de una sesión se puede llevar a cabo un transporte de datos ordinario, tal y como lo hace la capa de transporte, pero mejorando los servicios que esta proporciona y que se utilizan en algunas aplicaciones. Una sesión podría permitir al usuario, acceder a un sistema de tiempo compartido a distancia, o transferir un archivo entre dos máquinas.

### **Capa de Presentación.**

La capa de presentación realiza ciertas funciones que se necesitan bastante a menudo como para buscar una solución general para ellas, más que dejar que cada uno de los usuarios resuelva los problemas. En particular, y a diferencia de las capas inferiores que únicamente están interesadas en el movimiento fiable de bits de un lugar a otro, la capa de presentación se ocupa de los aspectos de sintaxis y semántica de la información que se transmite.

La capa de presentación está relacionada también con otros aspectos de representación de la información. Por ejemplo, la compresión de datos se puede utilizar aquí para reducir el número de bits que tienen que transmitirse, y el concepto de criptografía se necesita utilizar frecuentemente por razones de privacidad y de autenticación.

### **Capa de Aplicación.**

La capa de aplicación contiene una variedad de protocolos que se necesitan frecuentemente. Por ejemplo, hay centenares de tipos de terminales incompatibles en el mundo.

El modelo OSI, elaborado para describir protocolos para una sola red, no contiene un nivel específico para el ruteo en el enlace de redes, como sucede con el protocolo TCP/IP.

## **4.6 Normalización de Redes.**

Cuando apareció el concepto de redes, cada compañía fabricante de computadoras tenía sus propios protocolos; IBM por ejemplo, tenía más de una docena. Esto daba como resultado que aquellos usuarios que adquirían computadoras de diferentes compañías, no podían conectarlos y establecer una sola red con ellos. El caos generado por esta incompatibilidad, dio lugar a la exigencia de los usuarios para que se estableciera una normalización al respecto.

Esta normalización no solamente iba a facilitar la comunicación entre computadoras construidas por diferentes compañías, sino también traería, como beneficio, el incremento en el mercado para los productos que se apegarán a dicha norma, que conduciría a una producción masiva, una economía de escala por el incremento de la producción, así como otros tipos de beneficios cuya tendencia sería disminuir su precio y alentar su posterior aceptación.

Las normas se dividen en dos categorías que pueden definirse como: *de facto* y *de jure*. Las normas *De Facto* (derivado del latín, que significa "del hecho"), son aquellas que se han establecido sin ningún planeamiento formal. Las normas IBM PC y sus sucesoras son normas *de facto* para computadoras pequeñas de oficina, porque docenas de fabricantes decidieron copiar fielmente las máquinas que IBM sacó al mercado. Similarmente, Unix es la norma *de facto* para los sistemas operativos de los departamentos de ciencias de la computación en las universidades.

En contraste, *De Jure* (derivado del latín, que significa "por ley"), son normas formales, legales; adoptadas por un organismo que se encarga de su normalización. Las autoridades internacionales encargadas de la normalización se dividen, por lo general, en dos clases: la establecida por convenio entre gobiernos nacionales, y la establecida sin un tratado entre organizaciones voluntariamente. En el área de normas de redes de computadoras, existen dos organizaciones principales, de cada uno de los dos tipos.

#### **4.7 ¿ Quién determina el mundo de las normas. ?**

Las normas internacionales son producidas por la ISO (Organización internacional de la normalización), que es una organización voluntaria, fuera de tratados y fundada en 1946, cuyos miembros son las organizaciones nacionales de normalización correspondientes a los 89 países miembros. Entre sus miembros se incluyen a la ANSI (Estados Unidos), BSI (Gran Bretaña), AFNOR (Francia), DIN (Alemania) y otros 85 organismos.

La ISO emite normas en una gama muy amplia de temas, que cubren desde aquellos referentes a tuercas y tornillos (literalmente hablando), hasta los recubrimientos de los postes telefónicos. La ISO tiene casi 200 comités técnicos, cuyo orden de numeración se basa en el momento de su creación, ocupándose cada uno de ellos de un tema específico.

Por ejemplo, TC1 trata lo referente a tuercas y tornillos (normalizando los pasos de las hélices de los tornillos); el TC97, está relacionado con computadoras y procesamiento de información. Cada uno de los comités técnicos tiene subcomités, los cuales a su vez se dividen en grupos de trabajo.

Los grupos de trabajo, constituidos por casi 100,000 voluntarios, distribuidos en todo el mundo, son los que efectivamente realizan el trabajo. Varios de estos "voluntarios" son por lo general asignados por las propias compañías, cuyos productos están en vías de normalización para trabajar en asuntos de la OSI; otros voluntarios son empleados de gobierno interesados en promover que las cosas que está haciendo su país se adopten como normas internacionales. Expertos académicos también juegan una parte muy importante y activa en la constitución de varios grupos de trabajo.

La ISO y el CCITT algunas veces cooperan con respecto a la emisión de normas sobre telecomunicaciones, con objeto de evitar el absurdo de dos normas internacionales oficiales, mutuamente incompatibles.

El representante de Estados Unidos en la ISO es el ANSI (Instituto Nacional Americano de Normalización), quien a pesar de su nombre es una organización privada, descentralizada y no lucrativa; está constituida por fabricantes, proveedores de servicios portadores comunes y otros grupos interesados. Las normas del ANSI son frecuentemente adoptadas por la ISO, como normas internacionales.

El procedimiento que utiliza la ISO para el establecimiento de normas, está diseñado para conseguir el mayor consenso posible del tema respectivo.

El proceso se inicia cuando una de las organizaciones de normalización de carácter nacional, siente la necesidad del establecimiento de una norma internacional en una área específica; entonces se forma un grupo de trabajo que llega a plantear una propuesta de anteproyecto. Una vez que se genera la propuesta de anteproyecto, se hace circular entre todos los miembros, los cuales cuentan con seis meses, a partir de ese momento, para plantear sus comentarios y críticas. Si una mayoría significativa aprueba la propuesta, se produce un documento revisado denominado Anteproyecto de Norma Internacional, el cual se circula nuevamente con el objeto de tener más comentarios y realizar una votación al respecto. Con base en los resultados de esta votación, se prepara, aprueba y publica el texto final de la Norma Internacional.

En algunas de las áreas, en donde existe una gran polémica, la Propuesta de Anteproyecto o Anteproyecto de Norma Internacional probablemente tenga que pasar por varias versiones, en su planteamiento, antes de adquirir el número de votos necesarios para su aprobación, de tal forma que el proceso completo puede llevar varios años.

La Oficina Nacional de Normas es una agencia del Departamento de Comercio de Estados Unidos que emite normas obligatorias para las adquisiciones que realiza el gobierno de Estados Unidos, con excepción de aquellas que realiza directamente el Departamento de Defensa.

Otro participante importante en el mundo de las normas, es el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), que es la organización profesional más grande del mundo. Esta institución además de publicar numerosas revistas y programar un número muy importante de conferencias anuales, ha establecido un grupo dedicado al desarrollo de normas en el área de ingeniería eléctrica y computación. La norma 802 del IEEE, es para una red de área local, es la norma clave para el desarrollo de las LAN. Posteriormente, fue adoptada por la ISO como base para la norma ISO 8802.

## **Capítulo 5.**

### **5.1 Interconexión de nivel de aplicación.**

Los diseñadores han tomado dos enfoques diferentes para ocultar los detalles de las redes, utilizando programas de aplicación para manejar la heterogeneidad, o bien, ocultando los detalles en el sistema operativo. Las primeras conexiones heterogéneas de red proporcionaban la uniformidad por medio de programas de aplicación. En tales sistemas, un programa de nivel de aplicación que corre en cada máquina de la red, "entiende" los detalles sobre las conexiones de red para esa máquina e interactúa con los programas de aplicación a través de dichas conexiones.

Utilizar programas de aplicación para ocultar los detalles de la red puede parecer natural al principio, pero tal enfoque da como resultado una comunicación limitada e incómoda. Agregar funcionalidad al sistema implicaría diseñar un nuevo programa de aplicación para cada máquina. Agregar nuevo hardware de red implicaría modificar o crear nuevos programas para cada posible aplicación. En una máquina, cada programa de aplicación debe "entender" las conexiones de red para esa máquina, dando como resultado la duplicación de código.

Los usuarios que tienen experiencia con el trabajo con redes entienden que una vez que la interconexión crezca a cientos de miles de redes, nadie podría diseñar todos los programas necesarios de aplicación. Además, el éxito del esquema de comunicación requiere que todos los programas de aplicación que se ejecutan a lo largo del camino funcionen correctamente. Cuando falla un programa intermedio, tanto el origen como el destino se encuentran imposibilitados para detectar o resolver el problema. Por lo tanto, los sistemas que utilizan programas intermedios no pueden garantizar una comunicación confiable.

### **5.2 Interconexión de nivel de red.**

La alternativa para proporcionar la interconexión con programas de nivel de aplicación es un sistema basado en la interconexión de nivel de red. Una interconexión de nivel de red proporciona un mecanismo que entrega en tiempo real los paquetes, desde su fuente hasta su destino.

Comutar pequeñas unidades de datos en vez de archivos o grandes mensajes tiene muchas ventajas. Primero, el esquema se proyecta directamente hacia el hardware subyacente de red haciéndolo extremadamente eficiente. Segundo, la interconexión de nivel de red separa de los programas de aplicación, las actividades de comunicación de datos, permitiendo que computadoras intermedias manejen el tráfico de red sin "entender" las aplicaciones que lo utilizan. Tercero, utilizar conexiones de red mantiene flexible a todo el sistema, haciendo posible la construcción de instalaciones de comunicación con propósitos generales. Cuarto, el esquema permite que los administradores de red agreguen nuevas tecnologías de red al modificar o agregar una pieza sencilla de software nuevo a nivel de red, mientras los programas de aplicación permanecen sin cambios.

La clave para diseñar una interconexión universal de nivel de red, se encuentra en un concepto abstracto sobre sistemas de comunicación conocido como enlace de redes (internetworking). El concepto de red de redes o Internet es muy poderoso. Elimina la noción sobre comunicaciones de los detalles de las tecnologías de red y oculta los detalles de bajo nivel al usuario. De manera más importante, controla todas las decisiones sobre diseño de software y explica cómo manejar las direcciones físicas y las rutas.

### 5.3 Arquitectura de Internet.

Físicamente, dos redes sólo se pueden conectar por medio de una computadora en medio de los dos nodos. Sin embargo, una conexión física no proporciona la interconexión que tenemos en mente, debido a que dicha conexión no garantiza que la computadora cooperará con otras máquinas que se desean comunicar. Para obtener una red de redes viable, necesitamos computadoras que estén dispuestas a intercambiar paquetes de una red a otra. Las computadoras que interconectan dos redes y transfieren paquetes de una a otra, se conocen como pasarelas o computas de red de redes o ruteadores de red de redes.



*Das redes físicas interconectadas por un Ruteador.*

## 5.4 El datagrama de Internet.

La analogía entre una red física y una red de redes TCP/IP es muy fuerte. En una red física la unidad de transferencia es una trama que contiene un encabezado y datos, donde el encabezado contiene información sobre la dirección de la fuente y la del destino.

La red de redes llama a esta unidad de transferencia básica datagrama Internet, a veces datagrama IP o simplemente datagrama. Como una trama común de red física, un datagrama se divide en áreas de encabezado y datos. También el encabezado del datagrama contiene la dirección de la fuente y del destino; contiene también un campo de tipo que identifica el contenido del datagrama. La diferencia, por supuesto, es que el encabezado del datagrama contiene direcciones IP, en tanto que el encabezado de la trama contiene direcciones físicas.

La siguiente figura muestra la forma general de un datagrama:



Forma general de un datagrama IP, la estructura análoga de TCP/IP con respecto a las tramas de una red. El IP especifica el formato de un encabezado incluyendo, las direcciones de fuente y destino. El IP no especifica el formato del área de datos; el datagrama se puede utilizar para transportar datos arbitrarios.

## 5.5 Ruteo en una red de redes.

Todos los servicios de la red de redes, utilizan un sistema sin conexión de entrega de paquetes y la unidad básica de transferencia en una red de redes TCP/IP es el datagrama IP.

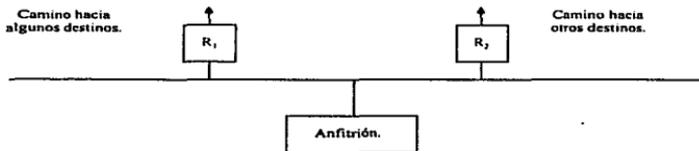
En un sistema de conmutación de paquetes, el ruteo es el proceso de selección de un camino sobre el que se mandarán paquetes y el ruteador es la computadora que hace la selección.

El ruteo en una red de redes puede ser difícil, en especial entre computadoras que tienen muchas conexiones físicas de red. De forma ideal, el software de ruteo examinaría aspectos como la carga de la red, la longitud del datagrama o el tipo de servicio que se especifica en el encabezado del datagrama, para seleccionar el mejor camino.

Para entender completamente el ruteo IP, debemos regresar y recordar la arquitectura de una red de redes TCP/IP. Primero, recuerde que una red de redes se compone de muchas redes físicas, interconectadas por computadoras conocidas como routers. Cada router tiene conexiones directas hacia dos o más redes.

Tanto los anfitriones como los routers, participan en el ruteo de datagramas IP que viajan a su destino. Cuando un programa de aplicación en un anfitrión intenta comunicarse, los protocolos TCP/IP eventualmente generan uno o más datagramas IP. El anfitrión debe tomar una decisión de ruteo cuando elige a dónde enviar los datagramas.

Como se muestra en la siguiente figura, los anfitriones deben tomar decisiones de ruteo, inclusive si sólo tienen una conexión de red.



Ejemplo de un anfitrión singly-homed que debe rutear datagramas. El anfitrión debe enviar un datagrama al router R<sub>1</sub> o al router R<sub>2</sub>, ya que cada uno proporciona el mejor camino hacia algunos destinos.

Cualquier computadora con muchas conexiones de red puede actuar como router y los anfitriones multi-homed que ejecutan el TCP/IP tienen todo el software necesario para el ruteo.

Además, los sitios que no pueden adquirir ruteadores por separado a veces utilizan máquinas de tiempo compartido y propósito general como anfitriones y ruteadores.

Sin embargo, los estándares TCP/IP hacen una gran diferenciación entre las funciones de un anfitrión y las de un ruteador, además los sitios que intentan mezclar funciones de anfitrión con funciones de ruteador en una sola máquina, a veces, encuentran que sus funciones multi-homed llevan a cabo interacciones inesperadas.

La transmisión de un datagrama entre dos máquinas dentro de una sola red física no involucra ruteadores. El transmisor encapsula el datagrama dentro de una trama física, transforma la dirección IP de destino en una dirección física de hardware y envía la trama resultante directamente a su destino.

Se puede pensar que los ruteadores, que saben como rutear paquetes hacia su destino, son grandes máquinas con suficiente memoria primaria o secundaria para guardar información sobre cada máquina dentro de la red de redes a la que se conectan. Sin embargo, los ruteadores utilizados en las redes de redes TCP/IP son por lo general computadoras pequeñas. A menudo tienen muy poco o nada de almacenamiento en disco y memorias principales limitadas. El truco para construir un ruteador pequeño para una red de redes reside en el siguiente concepto:

*Los ruteadores utilizan la red de destino, no el anfitrión de destino, cuando rutean un paquete.*

Si el ruteo está basado en redes, la cantidad de información que necesita guardar un ruteador es proporcional al número de redes dentro de otra red, no al número de computadoras.

TCP/IP está diseñado para proporcionar interconexión universal entre máquinas, independientemente de las redes en particular a las que están conectadas. Además de los ruteadores que interconectan redes físicas, se necesita software en cada anfitrión para permitir que los programas de aplicación utilicen la red de redes como si esta fuera una sola red física real.

### **5.6 La tabla de ruteo.**

El algoritmo usual de ruteo IP emplea una tabla de ruteo Internet, también conocida como tabla de ruteo IP, en cada máquina que almacena información sobre posibles destinos y sobre cómo alcanzarlos. Debido a que tanto los ruteadores como los anfitriones rutean datagramas, ambos tienen tablas de ruteo IP. Siempre que el software de ruteo IP en un anfitrión necesita transmitir un datagrama, consulta la tabla de ruteo para decidir a dónde enviarlo.

Todos los sistemas que utilizan TCP/IP utilizan una tabla de ruteo. Esa tabla es usada por el sistema para el tráfico directo. Por omisión, existe un mínimo de dos rutas en cada tabla de ruteo. La primera es la dirección de Loopback, 127.0.0.1. La segunda entrada informa a el sistema que el puerto de interfase de red es la ruta sobre nuestra red. Este argumento es vital si se desea comunicarse con otros nodos sobre nuestra propia red local. Cuando su sistema se inicialice y configure su(s) puerto(s) de interfase(s) de red, no olvide especificar estas dos rutas.

Sobre sistemas Unix, dos programas pueden acceder y modificar las tablas de ruteo; ellos son *netstat* y *route*, respectivamente. Existen equivalencias para otros sistemas por esta razón la mayor parte de estas implementaciones del programa *route*, no pueden desplegar la lista completa de la tabla de ruteo. Esta es la causa por la cual *netstat* es utilizado.

La opción *-r* de *netstat*, indica a *netstat* que imprima la tabla de ruteo. Puede poner una *n* después de *-r* si desea detener *netstat* para resolver problemas con las direcciones IP de los hosts.

La tabla de ruteo define el camino necesario para destino sobre la red y despliega la interfaz a utilizar. En adición a esta información, existe un campo de banderas que define el estado del ruteo. Diferentes banderas pueden aparecer en este campo. El más común es la bandera *U*, la cual significa "activo". La bandera *U* asegura que el ruteador este activo, funcionando y disponible.

La siguiente tabla muestra todas las banderas disponibles de ruteo.

Bandera.	Significado.
1	Bandera de protocolo específico de ruteo, número 1.
2	Bandera de protocolo específico de ruteo, número 2.
B	Orificio negro, todos los paquetes durante la actualización son descartados.
C	Generar una nueva ruta para uso.
D	Crear un ruteador dinámicamente.
G	Gateway.
H	Host.
L	Protocolo válido para ligar traslado de direcciones.
M	Dinámicamente modificado.
R	Destino inalcanzable.
S	Ruteo estático, adicionado manualmente.
U	Ruteo activo y disponible.
X	Protocolo de traducción de demonios externos para vínculos de direcciones.

### ***Ruteadores Estáticos.***

Los ruteadores estáticos son creados, ya sea para el programa de inicialización automático o manual, con el comando *route*. Redes pequeñas y desconectadas utilizan ruteadores estáticos solamente para comunicarse. Los ruteadores estáticos también son usados cuando sólo existe una ruta para un destino.

### ***Ruteadores Dinámicos.***

Los ruteadores dinámicos son creados para mandar paquetes conforme a sus necesidades. Usualmente, redes que poseen múltiples rutas para un destino, utilizan el ruteo dinámico. Estas rutas son transmisiones de paquetes a la red, con el protocolo de ruteo. Algún sistema recibe esa transmisión ajustando su tabla de ruteo por consiguiente. Esto provee no únicamente la mejor ruta para el destino, pero automáticamente respalda rutas de destino, si la principal ruta está fuera de servicio. El camino apropiado para leer la tabla de ruteo es entender que el Gateway es la ruta para el destino.

### **Gateway por Omisión.**

El ruteador por omisión es la ruta que se toma cuando todo ha fracasado. La ruta por omisión generalmente está asociada con el Gateway por omisión. El Gateway es la ruta para el resto del mundo, tan lejos como su máquina sea reconocida. Si los paquetes son direccionados a un destino fuera de su red local, estos tomarán el Gateway por omisión.

Usando el comando *route*, podemos adicionar una nueva ruta estática:

```
# route add default 170.137.254.1
```

### **5.7 Identificadores Universales.**

Se dice que un sistema de comunicaciones proporciona servicio universal de comunicaciones, si permite que cualquier computadora anfitrión se comunique con cualquier otro anfitrión. Para que un sistema de comunicaciones sea universal, necesita un método aceptado de manera global para identificar cada computadora que se conecta a él.

A menudo, los identificadores de anfitrión se clasifican como nombres, direcciones o rutas. Un nombre identifica lo que un objeto es, una dirección identifica dónde está y una ruta indica cómo llegar hasta ahí. Aunque estas definiciones son intuitivas, pueden ser confusas. Los nombres, direcciones y rutas se refieren a las representaciones sucesivas de bajo nivel de identificadores de anfitrión. En general, las personas prefieren nombres pronunciables para identificar máquinas, mientras que el software trabaja más eficiente con representaciones compactas de los identificadores que nosotros conocemos como direcciones. Este término también se pudo haber llamado identificadores universales de anfitrión TCP/IP. Se tomó la decisión de llamarlos así para estandarizarlos en direcciones compactas y binarias que hacen que cómputos tales como la selección de una ruta sean eficientes.

## 5.8 Direcciones IP.

Todas las tarjetas de red tienen un número de identificación único. Cada tecnología posee un rango de números que pueden ser usados; pero dos tarjetas de red no pueden tener la misma dirección física. Tanto las tarjetas de red Token Ring como Ethernet tienen este tipo de direcciones.

Antes de que usted utilice algunos de los protocolos, cada extremo de la conexión posee una única dirección lógica de red, la cual es llamada dirección IP. Estas direcciones lógicas son mapeadas a la tabla de Protocolo de Resolución de Direcciones (-Address Resolution Protocol, ARP-) a direcciones físicas. La tabla ARP habilita a la computadora para enviar y distribuir paquetes a la correcta dirección física. Las redes transfieren paquetes de información. Dentro de cada paquete es incluida la dirección IP del destino, sin esta información, la entrega de paquetes no sería posible. Esta dirección, es un número único de 32 bits que identifica un host dentro de Internet, pero también identifica al host dentro de la red de área local.

Para complementar esta información, la dirección está compuesta de dos partes: número de red y número de host. La división entre las partes, sin embargo, varía dependiendo de la clase de direcciones IP. Actualmente, existen 5 clases de direcciones IP en uso, las cuales son llamadas simplemente: clase A, clase B, clase C, clase D, clase E. La clase D, es usada para consideraciones especiales de red y la clase E, es reservada para usos futuros, por eso no son usados comúnmente para comunicaciones sobre Internet.

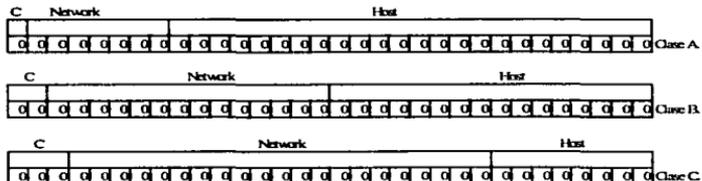
Una clase es identificada por los primeros bits en la dirección. Cada clase reserva un cierto número de bits para la parte de la red y los bits restantes para la identificación del host. La siguiente tabla muestra el número de bytes disponibles para redes e identificación de host para clases A, B y C.

Clase.	Bytes de Red.	Bytes del Host.
A	1	3
B	2	2
C	3	1

La siguiente tabla muestra el número de direcciones usadas por cada clase:

Clases de Hosts.	Rango de Red.	Números de redes usadas.	Números usados.
A	1 - 126	126	16,387,064
B	128-191	63	64,516
C	192-223	31	254

La siguiente tabla muestra la posición de los bits de las direcciones en las clases A, B y C.



Como el número de tamaño de bits de red es variable, puede observar que el número total de nodos y redes es diferente entre todas las clases de direcciones. Como usted puede observar, cada clase A, B o C poseen 1, 2 o 3 bytes para la definición del host. Es común utilizar el punto entre cada byte cuando nos referimos a una dirección IP. Esta representación es llamada notación cuadrática por puntos -Dotted quad notation-.

### 5.9 Subredes.

Si bien, el formato de las direcciones IP es estándar, sitios individuales pueden implementarse como una subred. Una subred es una dirección estándar IP, que es modificada para crear redes adicionales disponibles para incremento de anfitriones. Por subredes su dirección IP ganará diferentes redes, pero perderá direcciones para anfitriones.

Las subredes son especificadas por una máscara de subred, la cual define los bits que serán usados por ambos, la red y la repartición de direcciones para los hosts.

Por ejemplo, la siguiente tabla muestra las máscaras de subredes estándar para tres clases de direcciones IP.

Clase.	Máscara.
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

Como se puede observar, las máscaras de subredes estándar para cada clase señalan la proporción del host para cada clase de direcciones. Las clases de direcciones A y B proveen de múltiples redes y una red de clase C, es justamente una sola red. Sobre una clase C de direcciones IP, obtendrá tres nuevas redes. Instantáneamente, tendrá una gran red dando un total de cuatro redes. Por consecuencia para cada subred de clase C perderá dos host. Esto es por una pequeña regla: ninguna dirección de host puede ser puesta en cero, ya que este pertenece a la red; y tampoco puede tener todos los bits de dirección de transmisión en unos, ya que está reservado para la difusión. Por tanto, perderá un total de ocho hosts.

### ***5.10 Debilidades del direccionamiento de Internet.***

La codificación de información de red en una dirección de red de redes tiene algunas desventajas. La desventaja más obvia es que las direcciones se refieren a las conexiones de red, no a la computadora anfitrión.

Otra debilidad del esquema de direccionamiento en una red de redes es que cuando una red tipo C crece hasta tener más de 255 anfitriones, tiene que cambiar su dirección a una tipo B. Aunque esto puede ser un problema menor, el cambio de direcciones de red puede tomar demasiado tiempo y ser muy difícil de depurar. Debido a que la mayor parte del software no está diseñado para manejar muchas direcciones para la misma red física, los administradores no pueden planificar una transición suave en la que se introduzcan lentamente las nuevas direcciones. En vez de eso, tienen que dejar de utilizar abruptamente la dirección de red, cambiar las direcciones de todas las máquinas y reiniciar la comunicación utilizando la nueva dirección de red.

Las implicaciones son sorprendentes. Los humanos piensan en cada anfitrión como en una sola entidad y quieren utilizar un solo nombre. A veces, se sorprenden al encontrar que deben aprender más de un nombre y se sorprenden aún más, cuando encuentran que los paquetes enviados en los que utilizan muchos nombres pueden comportarse de manera diferente.

### ***5.11 Notación decimal con puntos.***

Cuando se envían a los usuarios, ya sea en documentos técnicos o a través de programas de aplicación, las direcciones IP se escriben como cuatro enteros decimales separados por puntos, en donde cada entero proporciona el valor de un octeto de la dirección IP.

Por lo tanto, sería útil entender la relación entre los tipos de direcciones IP y los números decimales con puntos.

En la siguiente figura se resumen el rango de valores para cada tipo:

<i>Tipo.</i>	<i>Dirección más baja.</i>	<i>Dirección más alta.</i>
<b>A</b>	0.1.0.0	126.0.0.0
<b>B</b>	128.0.0.0	191.255.0.0
<b>C</b>	192.0.1.0	223.255.255.0
<b>D</b>	224.0.0.0	239.255.255.255
<b>E</b>	240.0.0.0	247.255.255.255

Rango de valores decimales con punto que corresponde a cada tipo de direcciones IP.

### **5.12 Dirección Loopback.**

En la tabla anterior, se muestra que no todas las posibles direcciones se asignaron a algún tipo. Por ejemplo, la dirección 127.0.0.0, valor del rango tipo A, se reserva para loopback; y está diseñada para utilizarse en las pruebas de TCP/IP y para la comunicación de los procesos internos en la máquina local. Cuando algún programa utiliza la dirección loopback como destino, el software de protocolo en una computadora regresa los datos sin generar tráfico a través de alguna red.

Además, un anfitrión o un ruteador nunca debe difundir información de ruteo ni de accesabilidad para el número de red 127, pues no es una dirección de red.

### **5.13 Orden de octetos de red.**

Para crear una red de redes que sea independiente de cualquier arquitectura de máquina de una marca en especial o de cualquier hardware de red, el software debe definir una representación de datos estándar. Por ejemplo, considere lo que sucede cuando el software de una computadora envía un entero binario de 32 bits a otra computadora. El hardware físico de transporte mueve la secuencia de bits desde la primera máquina hasta la segunda sin cambiar el orden. Sin embargo, no todas las máquinas almacenan de la misma forma los enteros de 32 bits.

En algunas (llamadas *Little Endians*), la dirección más baja de memoria contiene el octeto de orden bajo del entero. En otras (llamadas *Big Endians*), la dirección más baja de memoria guarda el octeto de orden alto del entero. Otras almacenan los enteros en grupos de palabras de 16 bits, con las direcciones más bajas guardando la palabra de orden inferior, pero con los octetos desordenados. Por lo tanto, la copia directa de octetos de una máquina a otra puede cambiar el valor del número.

La estandarización del orden de octetos para los enteros es muy importante en una red de redes, ya que los paquetes llevan números binarios que especifican información como las direcciones de destino y la longitud de los paquetes. Tales cantidades deben entenderlas tanto el receptor como el transmisor.

Los protocolos TCP/IP resuelven el problema del orden de octetos, al definir un orden de octetos estándar de red que todas las máquinas utilizan para los campos binarios en los paquetes en la red de redes. Cada anfitrión o ruteador convierte los artículos binarios de la representación local al orden de octetos estándar de red antes de enviar un paquete y los convierte del orden de octeto estándar de red al orden específico del anfitrión cuando llega un paquete.

El estándar de la red de redes para el orden de los octetos, especifica que la parte de los enteros que se envía primero es el octeto más significativo (por ejemplo, el tipo *Big Endians*). Si se consideran los octetos en un paquete mientras viajan de una máquina a otra, un entero binario en dicho paquete tiene su octeto más significativo cerca del comienzo del paquete y su octeto menos significativo cerca del final.

### **5.14 Autoridad de direccionamiento Internet.**

Para garantizar que el campo de red dentro de una dirección de Internet es único, todas las direcciones de Internet son asignadas por una autoridad central. La Autoridad Internet de Números Asignados (IANA), establece los procedimientos y tiene el control sobre los números asignados.

Sólo se necesita una autoridad central para asignar el campo de red de una dirección; una vez que una organización obtiene un prefijo de red, puede escoger cómo asignar un sufijo único a cada anfitrión de su red sin tener que contactar a la autoridad central.

La Autoridad Internet de Números Asignados, asigna un número tipo C a una red con pocas computadoras conectadas a ella (menos de 255); se reserva los números tipo B para una organización que tiene una red más grande. Por último, una organización debe tener una red con más de 65535 anfitriones conectados antes de que pueda obtener un número tipo A. El espacio del nombre está diferido debido a que la mayoría de las redes son pequeñas, un menor número de ellas son medianas y sólo muy pocas son gigantes.

Solamente es esencial para la autoridad central asignar direcciones IP para redes que están (o estarán) conectadas a la red de redes. Una corporación individual puede tener la responsabilidad de asignar direcciones únicas de red dentro de su red de redes TCP/IP, siempre y cuando nunca conecte esa red de redes al mundo exterior.

## **Capítulo 6.**

### **6.1 Host y nombres de dominios.**

Recuerde que las direcciones IP, pueden ser difíciles de recordar. En adiciones a las direcciones físicas y lógicas de un host de red, existe el último identificador de información: el nombre del host; los nombres de los host son libres para poder formarlos.

De la misma forma que existe dos partes en una dirección IP -la red y la dirección del nodo-, existen dos partes para el formar el nombre de un host. La primera parte es la porción local, la cual es comúnmente referida a el nombre del host. La segunda parte se conoce con el nombre del dominio. El nombre del host y el nombre del dominio son conocidos como nombre de dominio calificado completo -Fully qualified domain name, FQDN-.

### **6.2 Nombres de host.**

El nombre del host puede ser cualquiera. No necesariamente indica alguna cosa en particular, pero no significa que carezca de significado. La mayoría de los nombres, son simplemente apodos puestos por el administrador del sistema.

Escoger un nombre de host, es como ponerle nombre a una mascota. Cuando se asigne nombres de hosts, existen dos reglas por lo general:

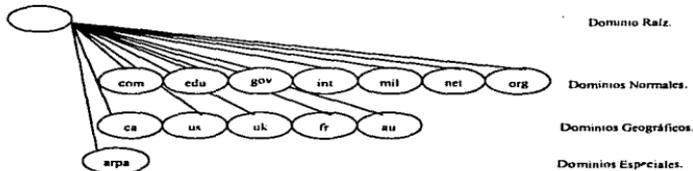
- ⌘ Utilice palabras completas que signifiquen alguna cosa: planetas, mascotas, personajes de caricaturas, etc.; utilice puras letras y números, aunque los números no están completamente fuera de asunto.
- ⌘ Nunca ponga el tipo de máquina con el nombre de *host*.

### 6.3 Nombres de Dominio.

Un nombre de dominio identifica un grupo de hosts. Este es un sinónimo con la parte de la red de una dirección IP. Internet guarda la pista a todos los nombres de dominio a través de un sistema llamado: sistema de nombre de dominio -domain name system, DNS-.

El DNS, es una base de datos distribuida de nombres de host; usada para traducir o resolver, nombres de hosts dentro de direcciones IP. Es distribuida porque no existe un organismo de almacenamiento central para todos los datos. Ningún servidor conoce todo acerca de todos los dominios.

En nombres de dominios existe una jerarquía general que se ilustra en la siguiente figura:



Los nombres de los dominios se escriben de izquierda a derecha. Cada nivel está separado por un punto.

### 6.4 Tipos de nombres de dominios.

Los dominios normales son aquellos que se ajustan a una estructura organizacional, actualmente existen siete tipos de dominios normales disponibles. El siguiente cuadro muestra los tipos de dominios y sus significados:

Domino.	Organizaciones que lo utilizan.
com	Organizaciones con propósitos comerciales (IBM, Dell Computer, etc.).
edu	Instituciones educativas, o Instituciones que ceden derechos (Harvard, Cornell, etc.).
gov	Agencias federales y gubernamentales de Estados Unidos (CIA, FBI, NASA, etc.).
int	Organizaciones internacionales (ONU, Banco Mundial, etc.).
mil	Agencias militares de Estados Unidos (Ejército, Fuerza Aérea, Armada, etc.).
net	Organizaciones de redes (InterNIC, PSINET, etc.).
org	Comúnmente organizaciones no lucrativas.

De la misma manera que existen identificadores de tipos de nombres de dominios, existen dominios geográficos que consisten únicamente de dos caracteres.

El siguiente cuadro ilustra algunos ejemplos de lo anterior:

Domino.	Propietario.
ca	Canadá.
us	Estados Unidos.
uk	Reino Unido.
fr	Francia.
au	Australia.

Los dominios *gov* y *mil* se encuentran exclusivamente reservados por agencias militares y entidades gubernamentales. El dominio *arpa*, existe para resolver direcciones IP, dentro de nombres de hosts. Esto es exactamente lo contrario del propósito de los dominios normales y los dominios geográficos.

El dominio *arpa*, tiene un segundo nivel de dominio el cual es *in-addr*. Los niveles, tercero, cuarto, quinto y sexto debajo del dominio *arpa* son cuadrantes de la dirección IP. Por ejemplo la dirección IP, 170.137.254.221 podría ser debajo de este dominio, de la siguiente manera: 221.254.137.170.in-addr.arpa. Con esta información y configurado adecuadamente el servidor puede encontrar el nombre del host para alguna dirección IP sobre Internet.

### 6.5 Registrando su nombre de dominio.

El nombre del dominio es nuestro y tiene la obligación de registrarlo desde InterNIC. Cuando usted asigne un nombre de dominio seleccione un nombre que refleje claramente el nombre de su organización. Este nombre tendrá un nombre de dominio fácilmente recordable. Aunque es posible que ya esté en uso el nombre que usted eligió. Si es este el caso, podrá seleccionar otro nombre.

Para observar si el nombre del dominio que usted seleccionó está en uso, puede usar el servicio llamado *whois* que pone a su disposición InterNIC.

Para accederlo utilice el servicio *Telnet* y conéctese a la máquina *whois.internic.net* y siga las instrucciones para buscar dominios existentes.

He aquí un ejemplo de una sesión para el nombre de "Snookums".

```
SunOS UNIX 4.1      (rs0) (ttyp8)
*****
* -- InterNIC Registration Services Center --
*
* For was, type:                WAIS <search string> <return>
* For the *original* whois type: WHOIS [search string] <return>
* For referral whois type:     RWHOIS [search string] <return>
*
* For user assistance call (703) 742-4777
# Questions /Updates on the whois database to HOSTMASTER@internic.net
* Please report system problems to ACTION@internic.net
*****
Please be advised that use constitutes consent to monitoring
(Elec Comm Priv Act, 18 USC 2701-2111)
10/8/96
```

We are offering an experimental distributed whois service called referral whois (RWhois). To find out more, look for RWhois documents, a simple client and server under:

```
gopher: (rs.internic.net) InterNIC Registration Services ->
InterNIC Registration Archives -> pub -> rwhois
anonymous ftp: (rs.internic.net) /pub/rwhois
Cmdinter Ver 1.3 Tuesday Oct 8 18:06:13 1996 EST
[vz220] InterNIC > whois
Connecting to the rs Database . . . . .
Connected to the rs Database
InterNIC WHOIS Version: 1.2 Tuesday, 8 Oct 96 18:06:22
```

```
Whois: snookums.com
No math for "SNOOKUMS.COM"
Whois: snookums
No math for "SNOOKUMS"
Whois: quit
```

Si usted carece del servicio *Telnet*, o si usted prefiere usar su visualizador de Web, puede dirigirse a: <http://rs.internic.net/cgi-bin/whois>. Esta es una interfaz del sistema de información *whois*.

### ***La Aplicación.***

El nombre del dominio de la aplicación puede ser llenado utilizando un visualizador de Web, la dirección es: <http://rs.internic.net/cgi-bin/reg/domain-form>.

Si usted no tiene acceso a un visualizador de Web, aquí presento una copia de la forma:

```
[ URL: ftp://rs.internic.net/templates/domain-template.txt ] [ 10/96 ]
***** Please DON'T REMOVE Version Number *****
Domain Version Number: 2.0
***** Please see attached detailed instructions *****
**** Only for registrations under ROOT, COM, ORG, NET, EDU, GOV ****
0. (N)ew (M)odify (D)elete . . . . . :
1 Purpose/Description . . . . . :
Organization Using Domain Name
3a. Organization Name . . . . . :
3b. Street Address . . . . . :
3c. City . . . . . :
3d. State . . . . . :
3e. Postal Code . . . . . :
3f. Country . . . . . :

Administrative Contact
4a. NIC Handle (if known) . . . . . :
4b. Name (Last, First) . . . . . :
4c. Organization Name . . . . . :
4d. Street Address . . . . . :
4e. City . . . . . :
4f. State . . . . . :
4g. Postal Code . . . . . :
4h. Country . . . . . :
4i. Phone Number . . . . . :
4j. E-Mailbox . . . . . :

Technical contact
5a. NIC Handle (if known) . . . . . :
5b. Name (Last, First) . . . . . :
5c. Organization Name . . . . . :
5d. Street Address . . . . . :
5e. City . . . . . :
5f. State . . . . . :
5g. Postal Code . . . . . :
5h. Country . . . . . :
5i. Phone Number . . . . . :
5j. E-Mailbox . . . . . :
```

**Billing Contact**

- 6a. NIC Handle (if known) . . . . . :
- 6b. Name (Last, First) . . . . . :
- 6c. Organization Name . . . . . :
- 6d. Street Address . . . . . :
- 6e. City . . . . . :
- 6f. State . . . . . :
- 6g. Postal Code . . . . . :
- 6h. Country . . . . . :
- 6i. Phone Number . . . . . :
- 6j. E-Mailbox . . . . . :

**Primary Name Server**

- 7a. Primary Server HostName . . . . . :
- 7b. Primary Server Netaddress . . . . . :

**Secondary Name Server**

- 8a. Secondary Server HostName . . . . . :
- 8b. Secondary Server Netaddress . . . . . :

**Invoice Delivery**

- 9. mail (P)ostal . . . . . :

La forma también está disponible desde *rs.internic.net*, a través de *FTP*. Llérrela y envíela por correo electrónico a *hostmaster@internic.net*.

En InterNIC se actualizan los nombres de los dominios comerciales. Hasta el 1 de Octubre de 1995, el registro del dominio era un servicio gratuito por la Fundación Nacional de Ciencia -National Science Foundation, NSF-. Actualmente, el costo por registrar un nombre de dominio es de 100 Dlls. Esto incluye el costo del dominio por un periodo de dos años. El costo por dominios existentes, es de 5 Dlls por año. Para más información consulte la siguiente página:

*ftp://rs.internic.net/policy/internic-domain-3.txt*

Haga un plan acorde a sus necesidades, ya que el registro del nombre del dominio puede tomar hasta ocho semanas para poder realizarse.

## 6.6 La tabla de hosts.

La más básica de todos los mecanismos de resolución de nombres de hosts, es la tabla hosts (o el archivo *hosts*), la cual se encuentra en el directorio */etc*, de los sistemas Unix.

Este archivo no tiene otra cosa, más que una base de datos de todas las direcciones IP y nombres de hosts que pueden ser localizados en cualquier parte de Internet, no únicamente en forma local.

Este es un ejemplo de un archivo *hosts*:

```
# /etc/hosts
#
170.137.254.3      honey.snookums.com
170.137.254.4    pookie.snookums.com
198.41.0.5        rs.internic.net
```

Cuando escribe un comando como *ping honey.snookums.com* o *telnet rs.internic.net*, el sistema revisa la tabla hosts para encontrar la dirección IP correcta. El nombre del hosts es una forma de recordar la dirección IP. La tabla de hosts actúa como un traductor para convertir el nombre del host a una cosa que la computadora pueda entender. Esto es similar a los lenguajes de programación.

Nota.-

Originalmente, antes de que estuviera distribuido el sistema de nombres de dominios, existía una tabla enorme de hosts, la cual era mantenida por InterNIC. Esta contenía información acerca de todas las máquinas registradas en InterNIC. Ahora, únicamente contiene una lista reducida, ya que los nuevos sitios no han sido adicionados.

Dentro de este archivo existen tres tipos de registros: "NET", "HOSTS", "GATEWAY". Los registros NET definen redes (170.137.0.0), los registros HOST definen hosts (170.137.254.221) y los registros GATEWAY definen gateways para otras redes.

## 6.7 BIND y DNS.

Después de que los usuarios comienzan la comunicación hacia Internet, con múltiples hosts, su tablas de hosts, comienzan un largo camino. Un software de empaquetamiento fue desarrollado en la Universidad de Berkeley en California, el cual se conoce ahora como Nombre de Dominio de Internet Berkeley -Berkeley Internet Name Domain, BIND-.

BIND, es un software con dos componentes: un nombre de *server* y un nombre de *resolver*. El *server* es un proceso que está ejecutándose actualmente, y que la parte del *resolver* busca para las resoluciones. El *resolver* no es un proceso, pero establece las llamadas a las librerías que serán utilizadas por programas para llamar a nombres de hosts.

El servidor BIND es llamado *named*. *Named* tiene tres diferentes modos de operación que son: primario, secundario y virtual. Estos modos son definidos para configurar la base de datos *named* en diferentes caminos. *Named* no está restringido para poderse ejecutar en un solo modo a un mismo tiempo. Puede ser configurado para ser primario y secundario al mismo tiempo.

- ☞ Un servidor primario es un servidor autorizado para aquel dominio. Cada dominio en Internet, requiere tener un dominio sencillo primario.
- ☞ Un servidor secundario es un respaldo del servidor primario. Este considera la autorización para aquel dominio. Parte de la configuración de *named*, involucra establecer los tiempos de expiración. Como el dominio expira, el servidor secundario lee nuevamente la información del dominio desde el servidor primario disponible.
- ☞ El último modo, es decir el servidor virtual, es similar a el servidor secundario, pero no considera autorización. Un servidor virtual, adelanta todas las preguntas para todos los servidores y guarda las respuestas para uso posterior. Los datos almacenados en el *cache* son válidos únicamente por un intervalo de tiempo. Cuando el tiempo expira, la información contenida en el *cache* es eliminada. Entonces el servidor readelanta todas las preguntas. Sin embargo, cuando los datos de los dominios se encuentran dentro del *cache*, el servidor puede responder las preguntas acerca del dominio.

Una nota final: `named` puede ser ejecutado como un servidor primario y secundario al mismo tiempo, pero no, para los mismos dominios. Por ejemplo, el servidor primario puede ser `snookums.com` y el servidor secundario `bunny.com` o viceversa.

### ***6.8 Configuración del nombre del servidor.***

La base de datos `named` es distribuida también por múltiples archivos. Estos archivos son normalmente archivos de texto y pueden ser creados y mantenidos con cualquier editor de texto.

El primer archivo `named.boot`, es un archivo de configuración del servidor. Este nombre de archivo puede ser cambiado y especificado con el comando `named`.

El segundo archivo muestra el mapa de los hosts. Cada línea contiene el nombre del host y su dirección IP asociada.

El tercer archivo muestra el mapa de las direcciones IP para cada nombre de hosts (para invertir los lookups).

El cuarto y último es el archivo `cache`, el cual llama a `named` para las direcciones de los servidores raíz. Los servidores raíz son los servidores para el completo sistema de nombre de dominio.

Los archivos de bases de datos son comúnmente almacenados en el directorio `/etc`, pero conforme su red crezca puede tener más de cuatro archivos.

#### ***named.boot.***

`named.boot` es el archivo de configuración del servidor. Este indica a el servidor en que modo puede operar, a cuál dominio y redes sirve, y dónde observar para los otros archivos de bases de datos.

Cada línea de archivo contiene un comando y la información necesaria para complementar el comando. El siguiente cuadro muestra los comandos que son soportados en `named.boot`.

Comando.	Acción.
directory	Especifica la ruta del directorio donde están almacenados los archivos de bases de datos.
cache	Especifica el nombre del archivo y la ruta del archivo <i>cache</i> .
forwarders	Provee a named una lista de servidores con los cuales podrá realizar búsquedas.
primary	Configura a named para ejecutarse como un servidor primario.
secondary	Configura a named para ejecutarse como un servidor secundario.
slave ;	Configura a named para utilizar únicamente extremos. Comentario. Cualquier cosa a continuación del punto y coma será ignorado.

Para configurar un servidor como un servidor primario, el archivo *named.boot* puede quedar de la siguiente manera:

```

;
; File : /etc/named.boot
; Comment: DNS Configuration for nsl.snookums.com
;
directory /etc/namedb
primary snookums.com snookums.hosts
primary 254.137.170.in-addr.arpa db.170.137.254
primary 0.0.127.in-addr.arpa db.127.0.0
cache named.cache

```

La especificación *directory* indica a el servidor que considere el archivo *namedb* dentro de */etc*.

La primera asignación *primary*, indica a el servidor que el archivo *snookums.hosts* es el archivo de mapeo para *snookums.com*. Este archivo contiene los nombres de los servidores y sus direcciones IP correspondientes. Todas las asignaciones primarias tienen el mismo formato: el comando *primary*, el nombre del dominio y el nombre del archivo de base de datos.

La segunda asignación `primary`, define un dominio `in-addr.arpa` para la red `170.137.254.0`. Este dominio está en el orden inverso de su dominio normal. Este archivo habilita a su servidor y a otros, para invertir la ejecución del nombre de host lookups de su dominio. El formato para la parte del dominio de este comando, es siempre la dirección IP de la red en sentido inverso encontrándose `in-addr.arpa` al final.

La tercer asignación indica a el servidor que el archivo `db.127.0.0` es el inverso del archivo de mapeo para la red `127.0.0`. Esta red, es una red especial que existe en todas las máquinas con sistema operativo Unix. Esta contiene un único nodo (`127.0.0.1`) para direccionamiento loopback. Esta es también una entrada en su tabla de hosts, pero si usted lo indica en la configuración de su `named` hará lookups rápidamente.

Configurar su servidor como un servidor virtual, requiere justamente una pequeña modificación en una asignación de su archivo de configuración. Simplemente remueva los comandos `primary` del archivo `snookums.com`. Note que la red `127.0.0` permanece, porque siempre tendrá su direccionamiento loopback.

```
;  
; File : /etc/named.boot  
; Comment: DNS Configuration for ns3.snookums.com  
;  
directory /etc/namedb  
primary 0.0.127.in-addr.arpa db.127.0.0  
cache . named.cache
```

Ahora que el servidor primario `snookums.com` ha sido configurado, procederé a configurar el servidor secundario. El servidor secundario para un dominio es como un respaldo del servidor. Este servidor tiene una copia del disco, dentro de su memoria que es actualizada periódicamente. La configuración es similar a el servidor primario, pero este tiene un parámetro adicional.

El comando `secondary`, requiere el nombre del archivo de los datos y la dirección IP del servidor primario para aquel dominio. Esta es la dirección IP invocada para actualizar el cache de `named`.

```
:
: File: /etc/named.boot
: Comment: DNS Configuration for ns2.snookums.com
:
directory /etc/namedb
secondary snookums.com 170.137.254.3 snookums.hosts
secondary 254.137.170.in-addr.arpa 170.137.254.3 db.170.137.254
primary 0.0.127.in-addr.arpa db.127.0.0
cache . named.cache
```

La primera asignación secondary, indica a el servidor que seremos el servidor secundario para el dominio snookums.com. Toda la información que el dominio puede recuperar desde el servidor cuya dirección IP es 170.137.254.3. Si el servidor está dado de baja, se encuentra una copia de los datos en el archivo */etc/namedb/snookums.hosts*. Lo mismo va en el segundo comando secondary dentro del archivo, pero relacionado a el dominio in-addr.arpa. Observe que se especificó el servidor primario para la red 127.0.0; esto es, porque existe en todas las máquinas.

Si el archivo especificado en la línea secondary no existiese, el servidor respalda una copia del dominio y escribe esto a el archivo. Si el archivo está disponible, el servidor simplemente carga el archivo y no transfiere la información desde el servidor primario. Esto ahorra tiempo y ancho de banda en la red.

### 6.9 Mapeo de Archivos.

Cada configuración de dominio requiere dos archivos de mapeo. Un archivo mapea los nombres de los hosts a la direcciones IP. El segundo archivo mapea las direcciones IP a los nombres de los hosts (lo inverso). Estos podrían combinar los nombres proporcionados en el archivo named.boot. El procedimiento utilizado anteriormente fue con snookums.hosts y db.170.137.254.

Todos los mapas de archivos utilizan el mismo formato. Los registros de formato utilizados son los llamados Registros de Recursos -Resource Records, RRs-. Comúnmente existen siete tipos de RRs.

Tipo.	Descripción.	Función.
A	Dirección.	Mapas de nombre de host o dirección IP.
CNAME	Nombre canónico.	Crea un alias para un nombre de host.
HINFO	Información del host.	Define el hardware del host y el sistema operativo.
MX	Intercambio de correo.	Define dónde entregar correo para el dominio.
NS	Nombre del Servidor.	Define el nombre del dominio del servidor.
PTR	Puntero.	Mapas o direcciones IP a nombres de hosts.
SOA	Comienzo de Autoridad.	Define el principio de los datos del dominio.
WKS	Servicio Conocido.	Define los servicios conocidos.

Cada RRs tiene el mismo formato:

`<object> [<ttl>] [<class>] <type> <data>`

- ✦ **object**, es el nombre del host o dirección IP que aplica este registro. Algunos tipos de registros permiten que esté en blanco.
- ✦ **ttl**, es el tiempo de vida, que define el incremento de tiempo en segundos antes de que el registro expire dentro de la memoria del servidor. El valor por omisión para este campo es especificado en el registro SOA.
- ✦ **class**, es la clase del registro. La única clase usada por sistemas DNS es la clase IN que representa la clase Internet.
- ✦ **type**, es el tipo de registro (ver el cuadro anterior).
- ✦ **data**, representa los datos asociados con el registro. Cada tipo tiene diferentes requerimientos de datos.

El registro de principio de autoridad, o SOA, es usualmente el primer registro en todos los archivos de mapeo. Este designa el principio de un dominio o zona. Esta zona finaliza con el siguiente registro SOA encontrado. Este podría tener únicamente un registro SOA por zona.

El registro SOA tiene el siguiente formato:

```
<name>  [<ttl>]  [<class>]  SOA  <origin>  <person>  (  
        <serial>  
        <refresh>  
        <retry>  
        <expire>  
        <minium> )
```

- ✦ **name**, es el nombre del dominio o zona.
- ✦ **origin**, es el nombre del host de la máquina donde el archivo de dominio primario existe.
- ✦ **person**, es la dirección e-mail de la persona responsable para esta zona. Reemplace la e en la dirección e-mail con un punto (.).
- ✦ **serial**, es el número de versión del archivo. Podrá aumentar este número si alguna vez realiza un cambio a el archivo. Si no ajusta este valor, sus cambios no serán registrados por otros servidores.
- ✦ **refresh**, es el número de segundos entre cada revisión, que el servidor secundario realiza a el servidor primario para observar si necesita actualizarse. Un valor típico de actualización es una hora, o 3600 segundos.
- ✦ **retry**, es el número de segundos que el servidor secundario podría reprocesar antes del movimiento, después de haber fracasado la actualización. Un valor típico de entrada es 10 minutos, o 600 segundos.
- ✦ **expire**, es el número máximo de segundos que indicará la existencia de los datos cuando una actualización no sea posible. Comúnmente, asignará un valor muy alto, ya que no querrá perder los datos contenidos dentro del *cache* de su red. Comience asignando a este 42 días, o 3,600,000 segundos.

El **minium**, es el número mínimo de segundos que pueden ser usados, para todos los valores no especificados de ttl, dentro de los RRs. Un valor típico es un día, o 86,400 segundos.

He aquí un registro SOA que es usado dentro de los archivos para snookums.com.

```
@ IN SOA ns1.snookums.com. root.ns1.snookums.com (
          951106001 ; Serial number
          3600 ; Refresh
          600 ; Retry
          3600000 ; Expire
          86400 ) ; Minium
```

Nota.-

Una buena práctica es usar la fecha como el número de serie. Con esto podrá registrar la última actualización realizada a sus archivos de configuración, utilizando el siguiente formato:

YYMMDDXXX

Donde:

YY es el año.  
MM es el mes.  
DD es el día.  
XXX es el número de actualización.

Ahora procederé a construir nuestros archivos de mapeo. Los dos archivos necesitados son el mapeo de hosts a la dirección IP y el mapeo de la dirección IP a el host. He aquí el archivo de mapeo de host a la dirección IP de snookums.com.

; File: snookums.hosts

; Comment: Host name to IP address mapping file

```
@ IN SOA ns1.snookums.com. root.ns1.snookums.com (
          951106001 ; Serial number
          3600 ; Refresh
```

```
        600                ; Retry
        3600000           ; Expire
        86400 )          ; Minium
;
; Name Servers
;
        IN                NS                ns1.snookums.com
        IN                NS                ns2.snookums.com
        IN                NS                ns3.snookums.com
;
; Hosts
;
localhost IN            A                127.0.0.1
honey     IN            A                170.137.254.3
pookie    IN            A                170.137.254.4
```

He aquí el archivo de mapeo de la dirección IP a el host.

```
; File: db.170.137.254
; Comment: IP address to host name mapping file
@ IN SOA ns1.snookums.com. root.ns1.snookums.com (
        951106001        ; Serial number
        3600             ; Refresh
        600              ; Retry
        3600000          ; Expire
        86400 )          ; Minium
;
; Name Servers
;
        IN                NS                ns1.snookums.com
        IN                NS                ns2.snookums.com
        IN                NS                ns3.snookums.com
;
; Hosts
;
        IN                PTR             honey.snookums.com.
        IN                PTR             pookie.snookums.com.
```

### El Archivo Cache.

El archivo cache es una lista de todo lo que conoce la raíz del servidor. El servidor necesita tener esta lista para funcionar adecuadamente. Al arranque, el servidor lee el contenido del archivo *cache*. Este comienza a contactar cada servidor en el orden de la lista, solicitando la lista completa de servidores raíz. Después de que un servidor ha respondido, la referencia del archivo *cache* es reemplazada con la lista obtenida desde el servidor raíz. He aquí el contenido de un archivo cache. La primera mitad consiste de registros de servidor de nombres -Name Server, NS-. El dominio raíz es interpretado por un punto (.). Cada registro del servidor de nombres indica que para el dominio ., este es un servidor de nombres.

El 99999999 representa el número de segundos hasta donde cada registro expirará. La segunda parte del archivo consiste en los registros de direcciones; así mismo, puede observar que los ttl tienen asignado 999999999 para estos registros.

```

;
; Generic DNS Cache file
;
.
999999999 IN NS A.ROOT-SERVERS.NET.
999999999 IN NS B.ROOT-SERVERS.NET.
999999999 IN NS C.ROOT-SERVERS.NET.
999999999 IN NS D.ROOT-SERVERS.NET.
999999999 IN NS E.ROOT-SERVERS.NET.
999999999 IN NS F.ROOT-SERVERS.NET.
999999999 IN NS G.ROOT-SERVERS.NET.
999999999 IN NS H.ROOT-SERVERS.NET.
999999999 IN NS I.ROOT-SERVERS.NET.
;
A.ROOT-SERVERS.NET. 999999999 IN A 198.41.0.4
B.ROOT-SERVERS.NET. 999999999 IN A 128.9.0.127
C.ROOT-SERVERS.NET. 999999999 IN A 192.33.4.12
D.ROOT-SERVERS.NET. 999999999 IN A 128.8.10.90
E.ROOT-SERVERS.NET. 999999999 IN A 192.203.230.10
F.ROOT-SERVERS.NET. 999999999 IN A 39.13.229.241
G.ROOT-SERVERS.NET. 999999999 IN A 192.112.36.4
H.ROOT-SERVERS.NET. 999999999 IN A 128.63.2.53
I.ROOT-SERVERS.NET. 999999999 IN A 192.36.148.17
```

### Configuración del Resolver.

La segunda parte del sistema BIND, es la configuración de su resolver. El resolver es vinculado con todos los archivos ejecutables que lo necesitan. Este, usualmente, viene como parte de algún lenguaje desarrollado en la forma de una biblioteca de código. El archivo llamado *resolv.conf* se encuentra alojado dentro de */etc*. Este archivo indica las librerías en donde buscar información. El formato del archivo es simple. La primer línea es el nombre del dominio por omisión para esta máquina. El comando es:

```
domain <nombre del dominio>
```

La segunda y subsiguientes líneas son los nombres de los servidores. La palabra *nameserver* es seguida por la dirección IP de cada servidor de nombres en el orden que usted elija para ser localizados como se muestra a continuación:

```
nameserver <dirección IP>
```

Nota.-

Usar direcciones IP dentro de */etc/resolv.conf* no es obligatorio. Sin embargo, si utiliza nombres de hosts asegura que los nombres de los hosts que utiliza se encuentran dentro de su tabla de hosts.

He aquí el archivo *resolv.conf* para *snookums.com*:

```
domain snookums.com  
nameserver 170.137.254.3  
nameserver 170.137.254.4
```

Observe que la primer línea de *nameserver* apunta a nuestro servidor de nombres primario, y la segunda línea de *nameserver* apunta a nuestro servidor de nombres secundario.

## **6.10 CERN y el Web.**

La mayor parte de los protocolos sobre Internet fueron desarrollados fuera de las necesidades. Los protocolos de World Wide Web no son la excepción. En 1989, dos investigadores del CERN (Laboratorio Europeo de Partículas Físicas -European Laboratory for Particle Physics- en Ginebra, Suiza; construyeron y distribuyeron mecanismos de intercambio de información que podrían ser usados internacionalmente.

Estos dos físicos, Tim Berners-Lee y Robert Cailliau, tuvieron esa visión. Ellos buscaron entrelazar la facilidad del Hipertexto con el alcance global de Internet. Propusieron un protocolo de lenguaje simple que pudiera ser manejado por un servidor. Este servidor, podría ejecutarse sobre una computadora anfitriona y ser capaz de proveer información y formas de negociación para los clientes.

Berners-Lee y Cailliau buscaron un cliente con una interfaz común para visualizar y hojear información. Los físicos no quisieron dictaminar los puntos que debería tener el visualizador; ellos querían reducir las características del sistema de operación de estos hosts.

## **6.11 NCSA Y Mosaic.**

El Centro Nacional para Aplicaciones de Supercomputación -National Center for Supercomputing Applications (NCSA), fue abierto en 1986 como un centro en donde investigadores de algún campo podrían colaborar y usar supercomputadoras para desarrollar sus investigaciones. NCSA también ayuda a grandes compañías beneficiadas de base competitiva para capacitación acerca de la última tecnología en hardware y software.

Como el proyecto World Wide Web comenzó a tomar forma, un visualizador de modo lineal fue desarrollado para examinar el servidor y el protocolo; y únicamente la limitación gráfica del visualizador estaba disponible. En Febrero de 1993, un estudiante que trabajaba en NCSA llamado Marc Andreessen liberó la primera versión alfa de Mosaic para XWindow. Mosaic fue el visualizador gráfico que puso al mundo en acción.

NCSA también se vio frustrado con el desarrollo del servidor HTTP. Este servidor CERN, ahora sabe que el servidor NCSA HTTP, es uno de los más populares servidores hoy en día.

### ***6.12 El protocolo.***

El protocolo HTTP, no es más que un protocolo de solicitud y respuesta. El cliente, o visualizador se conecta a el servidor. Entonces el cliente emite una solicitud. Las solicitudes son simples comandos, como el GET y PUT. El servidor interpreta la solicitud basada en su configuración y responde. La respuesta enviada a el cliente, es un archivo localizado sobre el servidor. Este documento puede estar en algún formato, pero este es usualmente en el formato HTML o Lenguaje Marcador de Hipertexto -HyperText Markup Language-.

Existe también un HTTP-NG (Next Generation) en proceso de elaboración. Este nuevo protocolo usará el ancho de banda más eficientemente y mejorará completamente el protocolo HTTP. La principal diferencia entre HTTP y HTTP-NG es que HTTP-NG es un protocolo binario. Para mayor información sobre HTTP-NG, consulte el servidor Web de Consortium.

### ***6.13 ¿ Qué es World Wide Web Consortium. ?***

El World Wide Web Consortium, o W3C, es la asociación constituida por el Laboratorio de Ciencias para Computadoras en el Instituto Tecnológico de Massachusetts. CERN, el inventor de Web, tiene oficialmente registrada toda la información del Web con respecto al W3C. Toda la información con respecto al World Wide Web está localizada en:

*<http://www.w3.org>*

### **6.14 Anatomía de un URL.**

URL significa Localizador de Recursos Universales -Uniform Resource Locator-. Un URL se compone de tres partes:

- ⌘ El protocolo.
- ⌘ El nombre del anfitrión.
- ⌘ El directorio.

El protocolo representa la manera en que el documento será accedido; esto es, el tipo de protocolo o programa que manejará su visualizador para obtener el archivo.

El nombre del anfitrión (*host name*), es el sistema en Internet en donde se almacena la información. La parte del URL correspondiente al nombre del anfitrión, puede incluir un número de puerto. El número de puerto indica a el visualizador, que abra una conexión para el protocolo apropiado de un puerto específico de red, diferente al puerto por omisión para cada protocolo. Si es necesario que escriba un número de puerto, este se ubica después del nombre del anfitrión.

El directorio representa la localización del archivo o cualquier otro tipo de información en el anfitrión.

### **6.15 Operaciones básicas del protocolo.**

Una transacción HTTP consiste de cuatro operaciones distintas.

- ⌘ Conexión.
- ⌘ Solicitud.
- ⌘ Respuesta.
- ⌘ Desconexión.

Un cliente se conecta a un servidor y emite una solicitud. Este espera para responder y entonces se desconecta.

### ***La Conexión.***

A este simple nivel, el software del cliente (visualizador -browser-) abre una conexión TCP/IP con el servidor Web sobre un puerto designado. Cuando abre una conexión con otra máquina, esta opera como cuando llama desde su casa a una operadora telefónica. Después de que la operadora recibe su llamada, ella introduce el número al cual quiere llamar, dentro de su tablero de conmutadores para completar la llamada. Cuando utiliza TCP/IP para realizar llamadas o conexiones a otras máquinas puede elegir entre 30,000 puertos.

### ***La Solicitud.***

Después de haberse establecido la conexión, el cliente necesita realizar una solicitud a el servidor. Esta solicitud debe estar en ASCII y debe terminar con un par de retorno de carro y alimentador de línea (Carriage-return/Line-feed).

El formato de una simple solicitud es el siguiente:

method <space> URI <space> HTTP-Version CRLF.

El método puede ser uno de los siguientes:

- ✎ GET.- Este método dirige a el servidor, para regresar todos los datos identificados por la solicitud URI. Si el URI representa un proceso, la salida del proceso es regresada. Por ejemplo:  
GET /~munster/index.html HTTP/1.0[CRLF].
- ✎ HEAD.- Este método es idéntico a el método GET, pero el servidor no regresa la parte de los datos. Este método puede ser utilizado para verificar la validez del enlace y su reciente modificación. Por ejemplo:  
HEAD /~munster/index.html HTTP/1.0[CRLF].
- ✎ POST.- Este método habilita a un cliente para enviar datos a un servidor y tener los datos almacenados en el servidor. Por ejemplo:  
POST /~munster/index.html HTTP/1.0[CRLF].

El campo HTTP-Version es llenado usualmente con la palabra HTTP, una diagonal y el número de versión.

#### ***La Respuesta.***

Después de que el servidor ha recibido la solicitud, la respuesta entra en el formato de un flujo de datos HTML.

HTML utiliza etiquetas para identificar partes especiales de un documento de texto. Las partes especiales pueden ser imágenes, vínculos a otros documentos, sonidos, tablas, etc.

Todos los documentos HTML tienen dos diferentes mitades: un encabezado y un cuerpo. Esto es, también el servidor puede enviar la parte del encabezado en respuesta a una solicitud de encabezado. La parte del cuerpo es la esencia de un documento. Estas dos partes son identificadas por las etiquetas *HEAD* y *BODY*.

#### ***La Desconexión.***

La desconexión es la última, pero no menos importante. Después de que la respuesta ha sido recibida, el cliente cierra la conexión. Es importante hacer notar que el servidor no cierra la conexión. Nadie está satisfecho con ninguna versión de software, especialmente en Internet. Esto es normal para programadores que rehacen cosas de mejor forma. La competencia natural es común entre la industria de software comercial.

Diseñar software para uso de Internet es grandioso, encontrará retroalimentación en muchos tipos de personas. Algunas de estas experiencias son positivas, algunas son negativas; pero todo esto es constructivo. Los programadores luchan por hacer que sus versiones de software sean mejores, rápidas y más eficientes.

## 6.16 Aspectos especiales de un servidor.

- ✦ **Capacidad para enviar documentos dinámicos o imágenes.** Un servidor avanzado, es un servidor especial HTTP, desarrollado para ser usado principalmente para animaciones para el cliente. Normalmente, al fin de una operación HTTP, el cliente podría cerrar la conexión. Dentro de un servidor avanzado, el cliente puede dejar la conexión abierta y cuando el servidor percibe este ajuste envía más datos a el cliente. Entonces el cliente puede desplegar los datos que obtiene cada que lo desea.
- ✦ **Acceso a el núcleo del servidor desde CGI u otros programas.** Cuando el servidor ejecuta un programa, este accesa información, ya sea desde la línea de comandos del programa o dentro de las variables de entorno. Si el servidor tiene esta característica, podrá acceder a la información por sí mismo, ya sea desde la línea de comando o desde las variables de entorno.
- ✦ **Respuesta automática para If-Modified-Since.** Los requerimientos de los servidores pueden ser más complejos que un simple *GET*, *HEAD* o *POST*. En adición a estos métodos básicos, cada uno puede tener un método adicional de datos. Uno de los datos adicionados al método *GET* es llamado el encabezado *If-Modified-Since*. Este es un requerimiento que viene desde un cliente a el servidor con esta directiva, el servidor no enviará el documento a el cliente, a menos que el archivo haya sido modificado desde la fecha y hora especificada por el cliente. Esto es diseñado para reducir el uso de la red y para asistir a el modo virtual del lado del cliente. La respuesta automática para este requerimiento es hecho con la fecha y hora del archivo.
- ✦ **Puede servir diferentes raíces basadas sobre direcciones IP.** Cuando instala su servidor Web, lo configura para ser establecido dentro de algún directorio como */Web*. Si el servidor puede servir diferentes raíces basadas sobre direcciones IP, puede tener diferentes directorios base para cada tarjeta de red o dirección IP que utilice.
- ✦ **Acciones basadas para cambio sobre el encabezado User-Agent.** Otras de las directivas de requerimiento es el campo *User-Agent*, el cual indica el software del cliente a el servidor para propósitos estadísticos de análisis. La gran mayoría de los servidores pueden actuar de diferente manera sobre esta información.

- *Respuesta con diferentes documentos basados sobre encabezados de clientes.* Este es el caso más general de un futuro antecedente. El servidor puede enviar múltiples documentos a el cliente basados sobre el encabezado recibido en la solicitud.
- *Acceso a las jerarquías de datos basadas sobre direcciones IP.* Esta adición garantiza el acceso o recepción de datos basados sobre direcciones IP.
- *Puede requerir un password.* El servidor puede requerir autorización antes de proveer información a el cliente.
- *Actuar como un servidor HTTP Proxy.* Un servidor Proxy, es un proceso que actúa como un tablero de conmutadores para viajar a través de la red. Este servidor habilita a los usuarios para establecer contacto con Internet. Típicamente, el servidor toma su solicitud y la envía a el servidor real.

## **Capítulo 7.**

### **7.1 Instalando el servidor CERN HTTP.**

El software que requiere es el servidor HTTP o httpd. Httpd posee el demonio HTTP y es pronunciado "HTTP-di". Justamente como cualquier otro programa Unix que termina con una d, es un proceso permanentemente activo. Este escucha una conexión TCP/IP que entra, acepta la solicitud y responde. El primer paso para establecer nuestro servidor, es obtener el software.

Nota.-

Usted podría obtener su software httpd desde el home page de CERN. Esto le asegurará que no olvide un archivo que posteriormente requerirá y le permitirá obtener la versión más reciente. La dirección es la siguiente:

*<http://www.w3.org/pub/WWW/Daemon/>*

Dentro de esta dirección podrá encontrar códigos fuentes y binarios, tips para la instalación y preguntas frecuentemente contestadas.

Usted podrá obtener los archivos desde CERN, ya sea como archivos binarios o como archivos fuente.

El archivo binario es un sistema precompilado para varias versiones de Unix. Después de haber obtenido el archivo, simplemente descomprima y desempaquete el archivo y el servidor quedará instalado. Esta opción se adapta para aquellos administradores que no están familiarizados con la programación y para aquellos sistemas que carecen de un completo sistema de desarrollo.

Utilizar el código fuente es más complejo y no le agrada a toda la gente. El archivo contiene el código fuente completo para el servidor y archivos de soporte. Descomprima y desempaquete el archivo y compile el servidor para soportar los archivos. Necesitará ciertas cosas sobre su sistema para realizar esto.

Esto puede involucrar algunas ediciones y definitivamente involucra compilación; lo cual no es apropiado para administradores novatos de sistemas. Como mínimo, el ambiente debe incluir un compilador, librerías de ejecución y el programa *make*.

#### ¿ Qué es make.?

Make es un programa que simplifica la construcción de aplicaciones y sistemas. El programa toma un archivo *makefile* como entrada. Este archivo contiene las instrucciones necesarias para compilar una aplicación o sistema. Si usted posee un programa que consista de tres módulos fuente, deberá compilar y ligar sus módulos cada vez que realice un cambio.

Antes de obtener el archivo binario debemos observar el tipo y la versión de sistema operativo Unix existente sobre nuestro sistema. Para observar esto utilice el comando *uname -a* desde el prompt.

La salida de este comando indica el tipo y la versión de nuestro sistema operativo. El comando *uname* obtiene seis segmentos de información: el sistema operativo utilizado, el nombre del host, la versión del kernel, el número de revisión, la fecha y la hora, y el tipo de microprocesador utilizado dentro del host. La siguiente tabla muestra los diferentes tipos de sistemas operativos Unix y las versiones para las cuales CERN provee sistemas precompilados.

UNIX.	Versión.
AIX.	3.2.x
HP-UX.	9.x
IRIX.	5.2
Linux.	1.2.10
NextStep.	3.3
OSF/1.	3.0
SCO.	3.0/3.2
Solaris.	2.3
Solaris.	2.4
SunOS.	4.1.3
Ultrix.	4.3

**Nota.-**

La lista de sistemas soportados cambia mes con mes. Consulte la siguiente página para obtener la actualización de los sistemas soportados.

<http://www.w3.org/pub/WWW/Library/User/Platform/>

Sin embargo, si su sistema no aparece en la lista deberá compilar el código fuente. Esto no es difícil, pero si usted no es un programador esto será una ardua tarea.

En cualquiera de los dos casos, necesita obtener el software. Primero determine cual tipo quiere -precompilado o fuente-, y posteriormente utilice su visualizador de Web y obtenga el archivo de alguno de los dos servidores Web.

⚡ Distribución binaria.

<http://www.w3.org/pub/Dist/httpd>

⚡ Código fuente para distribución (dos archivos).

[http://www.w3.org/pub/Dist/libwww/libwww\\_2.17\\_src.tar.Z](http://www.w3.org/pub/Dist/libwww/libwww_2.17_src.tar.Z)  
[http://www.w3.org/pub/Dist/httpd/httpd\\_src.tar.Z](http://www.w3.org/pub/Dist/httpd/httpd_src.tar.Z)

## **7.2 Descomprimiendo el archivo.**

El software viene dentro de un archivo llamado *tar*. *Tar* es un programa de Unix diseñado para comprimir archivos de datos a cintas. De hecho *tar* viene de "tape-archiver".

El programa *compress* es un programa estándar para la compresión de programas en Unix, que está disponible para todos los sistemas Unix. La contraparte de *compress* es *uncompress*. De la misma manera, *gzip* tiene el comando *gunzip* para descompresión, sin embargo, puede utilizar *gzip* para descomprimir también.

Los archivos producidos por estos dos programas pueden ser identificados por sus extensiones: los archivos *compress* terminan con *.Z*, mientras que los archivos *gzip* terminan con *.gz*.

Ahora seleccione un lugar para almacenar la arborescencia de su código fuente. Para el siguiente ejemplo utilizaré un sistema Linux, guardaré el código fuente en la arborescencia de código fuente para todos los otros procesos, siendo esta arborescencia */usr/src*. Los administradores de sistemas Linux generalmente guardan su código fuente debajo de este directorio, y *htpd* no es la excepción. El archivo binario contiene el código fuente y el código ejecutable. Por tanto, el archivo podría ser descomprimido dentro del directorio */usr/src*.

### 7.3 *Uncompress* y *tar*.

El primer paso para descomprimir el archivo fuente es ejecutar el programa *uncompress*. El archivo fuente obtenido fue almacenado dentro de */tmp*, aunque el lugar de almacenamiento puede variar. He aquí la línea de comando para *uncompress* y *tar*:

```
root@honey:/usr/src# uncompress /tmp/htpd_src.tar.Z
root@honey:/usr/src# uncompress /tmp/libwww_2.17_src.tar.Z
```

El programa *uncompress* descomprime el archivo y le quita la extensión *.Z*. El siguiente paso es utilizar *tar* sobre el archivo:

```
root@honey:/usr/src# tar -xvf /tmp/htpd_src.tar
```

☞ Para consultar la salida del comando remítase al anexo número 1.

```
root@honey:/usr/src# tar -xvf /tmp/libwww_2.17_src.tar
```

☞ Para consultar la salida del comando remítase al anexo número 2.

```
root@honey:/usr/src#
```

*Tar* creó un nuevo directorio debajo de */usr/src* llamado *WWW*. Este directorio contiene los siguientes elementos:

```
root@honey:/usr/src/WWW# ls -l
total 22
drwxrwxr-x  27 13449   69 1024   Sep 24 1995 All
-rwxr-xr-x   1 13449   69 3111   Sep 24 1995 BUILD
-rwxr-xr-x   1 13449   69 3076   Sep 23 1995 BUILD.SH
drwxrwxr-x   3 13449   69 1024   Nov 21 13:26 Daemon
drwxrwxr-x   4 13449   69 1024   Sep 26 1995 ICE
-rw-r--r--   1 13449   69  580   Jun 11 1995 Makefile
-rw-r--r--   1 13449   69 2839   Jun 11 1995 README
-rw-r--r--   1 13449   69 5218   Jun 11 1995 README-SOCKS
drwxrwxr-x   4 13449   69 1024   Jul  8 1995 server_root
```

Los archivos de este directorio son los siguientes:

- ✦ **All**. Este directorio contiene todos los archivos *makefiles* para cada uno de los sistemas soportados.
- ✦ **BUILD** y **BUILD.SH**. Estos archivos son utilizados por el programa *make* para construir los componentes.
- ✦ **Daemon/**. Este directorio aloja todo el código fuente para el programa *httpd*.
- ✦ **ICE/**. Este directorio aloja un software llamado *ICE* que provee la capacidad de indexación para su sitio *Web*.
- ✦ **Makefile**. Este es el archivo *makefile* para el directorio completo.
- ✦ **README** y **README-SOCKS**. Estos archivos contienen información con respecto a la versión y notas generales de compilación.
- ✦ **server\_root/**. Este directorio tiene una muestra de archivos de configuración e iconos por omisión para el servidor.

## 7.4 Compilando el servidor.

El archivo *makefile* del servidor CERN determina automáticamente el sistema operativo que se está utilizando, por lo que la selección del servidor no es una opción. Para compilar el servidor simplemente ejecute el programa *make*. El programa *BUILD* decidirá cuales componentes necesitarán estar creados y cuales creará.

He aquí la salida de *make*:

☞ Para consultar la salida del comando remítase al anexo número 3.

Nota.-

Cuando compile el software, pueden existir varios errores que son ignorados por *make*, esto es normal. Como los componentes son construidos, el *makefile* comienza a borrar la copias anteriores de aquel componente. Si el componente no existe (esto es, la primera vez que se copia) ocurre un error. Como puede observar en el listado, los errores son ignorados.

Si todas las cosas fueron satisfactorias, tendrá un archivo llamado *httpd* es su directorio */usr/src/www/Daemon/linux*. Si no está ejecutando Linux, el nombre del directorio que contiene su código compilado aparecerá al final, siendo este la plataforma donde se ejecutó el proceso.

## 7.5 Preparándose para instalar el software.

Antes de instalar el software, necesitará completar los siguientes pasos:

- ☞ Determinar dónde se encontrarán los archivos de soporte del servidor.
- ☞ Determinar cómo se ejecutará el servidor.
- ☞ Determinar cuáles usuarios y grupos serán propios para los procesos del servidor.

## 7.6 ¿ Dónde poner el software. ?

El primer paso para instalar el software es determinar dónde estarán alojados los archivos. Necesita seleccionar un directorio, que será el directorio raíz para la arborescencia completa del servidor. Esto es similar al lugar para la arborescencia del código fuente, pero ahora para los archivos binarios.

La selección de este directorio es completamente del administrador del sistema, aunque una buena opción para establecer este directorio es */usr/local/etc/httpd*. Aunque habría que considerar los siguientes aspectos: espacio del disco, acceso de usuarios, etc.

### **Espacio del disco.**

El servidor por sí mismo no ocupa mucho espacio del disco. Sin embargo, el directorio donde colocó el software puede contener todos los documentos e imágenes que el servidor provee; si este es el caso desarrolle un plan acorde, la imágenes pueden incrementar de manera significativa el espacio ocupado de su disco.

### **Acceso de usuarios.**

Algunos usuarios requerirán acceder estos directorios (Usted, el Webmaster, el personal del Departamento de Mercadotecnia, etc.). Por tanto, deberá asegurarse que los archivos del sistema que seleccionó para este directorio estén disponibles para todos los usuarios que puedan acceder estos archivos.

## 7.7 ¿ Cómo se ejecutará el servidor. ?

Existen dos opciones para ejecutar su servidor: *standalone* o *spawned*. El servidor *standalone* se ejecuta como un proceso escuchando para conexiones. Cuando una conexión entra, este crea una copia de sí mismo para manipular la solicitud. Este es un proceso rápido y eficiente.

Un servidor *spawned* es "*spawned*" por un proceso llamado *inetd*. *inetd* es un demonio de Internet; este pequeño individuo es la espina dorsal del ruteo de TCP/IP en un servidor Unix. Fuera de esto, muchos procesos típicos que son dados como seguros (como telnet, ftp y finger), simplemente no trabajan adecuadamente.

Nota.-

*Inetd* es la madre de todos los servidores. Este escucha las solicitudes que llegan de los clientes. Cuando una solicitud llega a cierto puerto, este engendra un programa para manipular aquella solicitud. La configuración de este programa es hecha en un archivo llamado *inetd.conf*, el cual se encuentra alojado generalmente dentro del directorio */etc*.

Para ejecutar su servidor fuera de *inetd*, necesita adicionar una línea para el protocolo http. Este protocolo es un cruce referencial con el archivo llamado *services*, que también se encuentra dentro del directorio */etc*. *Services* debe incluir una línea para el protocolo HTTP que incluye el nombre (*http*), el tipo de conexión (*tcp*), y el puerto sobre el cual escucha (generalmente 80).

He aquí la línea que podría adicionar a su archivo */etc/services*, si es que este no la tiene:

```
http      tcp/80      # Hypertext Transfer Protocol.
```

Nota.-

Usted podría adicionar esta línea a su archivo */etc/services* sin considerar el modo de ejecución que haya seleccionado. Por adicionar la línea, estará habilitando la referencia de su puerto por nombre, en lugar de por número con programas como *telnet*, *ping*, *ftp*.

He aquí una línea de su archivo *inetd.conf*, si usted selecciona este modo:

```
http      stream    tcp        nowait    www       /usr/local/etc/httpd/httpd httpd
```

Después de haber realizado algunos cambios a el archivo */etc/inetd.conf*, puede restaurar el proceso *inetd* para enviar la señal de colgar; esto lo realiza con el comando *kill*, puede enviar la señal de colgar con el parámetro *-HUP*.

Este método tiene un inconveniente. Primero es más lento que un servidor *standalone*. Segundo, cada conexión que es recibida resulta en un nuevo proceso *httpd* estando inicializado. Por instancia, asumimos que una página típica de Web tiene cinco imágenes sobre esta.

Un cliente, conectándose a el servidor que se está ejecutando fuera de *inetd*, realizará un total de seis conexiones (una por el documento y cinco más por las imágenes). Cada una de estas conexiones podría resultar en un proceso activo.

Este método también afecta a el desarrollo desfavorablemente. Si su sitio está ocupado podría recibir muchas solicitudes por segundo. Esto puede traer a el funcionamiento de su sistema una pausa.

¿Porqué entonces, podría querer ejecutar su servidor por este medio ?. Porque utilizando *inetd* en algunas instancias puede ser muy seguro. Este puede ser configurado para ejecutar programas antes de ejecutar *httpd*. Puede reservar un nivel adicional de seguridad dentro de este, para asegurarse de que únicamente acepta solicitudes desde ciertos dominios o direcciones IP.

### **7.8 ¿ Quién podría poseer los procesos del servidor. ?**

La última pieza de configuración que necesita antes de instalar su software, es determinar que usuarios tendrán derechos para utilizar los procesos del servidor. Cuando se encuentra dentro de una máquina Unix y ejecuta un proceso, este es poseído por usted. De la misma manera, tiene la opción de establecer su servidor *httpd* para ser ejecutado por cualquier persona sobre su sistema. Esta característica lo habilita para restringir el acceso a ciertos tipos de archivos y proteger la seguridad del servidor. Existen dos opciones que necesita conocer: *nobody* y *el usuario común*.

La gran parte de los sistemas Unix vienen con un usuario y un grupo registrado llamado *nobody*, los cuales son típicamente usados para conexiones entre máquinas que son consideradas como seguras (esto es, montando NFS). Una de las opciones que se recomiendan es utilizar el usuario *nobody* para ejecutar su servidor.

La segunda sugerencia es crear un usuario y un grupo para tener los procesos del servidor. Esto tiene dos beneficios:

- ⚡ Algunos defectos que existen en otro software para el usuario *nobody* son omitidos. El software que actualmente utiliza el usuario *nobody* puede tener orificios de seguridad, acerca de los cuales están desatendidos. Un *hacker* de Web puede tomar ventaja de semejantes orificios y tener el control de la operación de su sistema. Puede eliminar la posibilidad de una violación de seguridad por no usar *nobody*.
- ⚡ Esto fuerza a que verifique su instalación completa. Pero si está usando una nueva cuenta, no existirán muchos archivos propios para este usuario.

Una recomendación es crear un nuevo usuario llamado *www* y un nuevo grupo llamado *www*. Siendo este el usuario y el grupo que se utilizará para ejecutar el servidor.

### 7.9 Instalando el software.

Ahora que seleccionamos nuestro directorio principal, el modo de ejecución, el usuario y el grupo, nos encontramos listos para instalar nuestro servidor. A manera de recapitulación, he aquí los elementos que seleccionamos:

- ⚡ **Directorio:** Vamos a estar utilizando el directorio */usr/local/etc/httpd*.
- ⚡ **Método de Ejecución:** Vamos a estar ejecutando nuestro servidor como un servidor *standalone*.
- ⚡ **Propietario del Proceso Usuario/Grupo:** Se creó un usuario *www* y un grupo *www* que serán los propietarios de los procesos del servidor.

Con estos elementos seleccionados, ahora instalaremos nuestros archivos binarios.

## **7.10 Creando la arborescencia del directorio.**

Lo primero que necesitamos es crear la arborescencia donde se alojarán estos archivos. Estos son los directorios que necesitamos crear:

**⚡ /usr/local/etc/httpd.** Esta es la raíz de nuestro servidor. Todos los archivos estarán alojados bajo este directorio.

**⚡ /usr/local/etc/httpd/cgi-bin.** Es aquí donde estarán alojados todos nuestros programas *CGI*. Estos programas habilitan a el servidor para ejecutar programas externos.

**⚡ /usr/local/etc/httpd/conf.** Este directorio alojará todos los archivos de configuración de nuestro servidor.

**⚡ /usr/local/etc/httpd/logs.** Este directorio alojará todos los archivos de registro de nuestro servidor.

**⚡ /usr/local/etc/httpd/htdocs.** Este directorio alojará todos los documentos de información que proporcionará nuestro servidor.

**⚡ /usr/local/etc/httpd/icons.** Este directorio alojará todos los iconos y gráficos del servidor.

Por lo tanto, con los siguientes comandos crearemos los subdirectorios que necesitamos:

```
root@honey:~# cd /usr/local/etc
root@honey:~/usr/local/etc# mkdir httpd
root@honey:~/usr/local/etc# cd httpd
root@honey:~/usr/local/etc/httpd# mkdir cgi-bin
root@honey:~/usr/local/etc/httpd# mkdir conf
root@honey:~/usr/local/etc/httpd# mkdir logs
root@honey:~/usr/local/etc/httpd# mkdir htdocs
root@honey:~/usr/local/etc/httpd# mkdir icons
```

El siguiente paso es copiar los archivos dentro de los directorios:

```
root@honey:# cd /usr/local/etc/httpd
root@honey:/usr/local/etc/httpd# cp /usr/src/WWW/Daemon/httpd .
root@honey:/usr/local/etc/httpd# cp /usr/src/WWW/Daemon/htadm .
root@honey:/usr/local/etc/httpd# cp /usr/src/WWW/Daemon/htimage .
root@honey:/usr/local/etc/httpd# cp /usr/src/WWW/Daemon/cgiparse .
root@honey:/usr/local/etc/httpd# cp /usr/src/WWW/Daemon/cgiutils .
root@honey:/usr/local/etc/httpd# cp /usr/src/WWW/server_root/config/* ./conf
root@honey:/usr/local/etc/httpd# cp /usr/src/WWW/server_root/icons/* ./icons
```

Si usted no compila el código fuente, la instalación de los archivos binarios es similar. La diferencia es que primero debe descomprimir el archivo con los comandos *uncompress* y *tar* dentro del directorio principal de su servidor, como lo ilustran los siguientes comandos:

```
root@honey:# cd /usr/local/etc/httpd
root@honey:/usr/local/etc/httpd# gunzip /tmp/httpd_3.0_bin.linux_aout.tar.gz
root@honey:/usr/local/etc/httpd# uncompress /tmp/server_root.tar.Z
root@honey:/usr/local/etc/httpd# tar -xvf /tmp/httpd_3.0_bin.linux_aout.tar
root@honey:/usr/local/etc/httpd# tar -xvf /tmp/server_root.tar
root@honey:/usr/local/etc/httpd# mv ./config/* ./conf
root@honey:/usr/local/etc/httpd# rm -rf ./config
```

## **Capítulo 8.**

### **8.1 Instalando el servidor NCSA HTTP.**

Este seguro de decir que el servidor NCSA HTTP es el más popular hoy en día. Esto puede ser atribuido a muchas cosas, pero el factor más importante son sus características. El servidor NCSA está aprobado, ya que cuenta con el software con mayor antigüedad por lo que adolece de las mismas deficiencias presentadas con anterioridad.

### **8.2 Obteniendo el software.**

El software que requiere es el servidor HTTP o httpd. Httpd posee el demonio HTTP y es pronunciado "HTTP-di". Justamente como cualquier otro programa Unix que termina con una d, es un proceso permanentemente activo. Este escucha una conexión TCP/IP que entra, acepta la solicitud y responde. El primer paso para establecer nuestro servidor, es obtener el software.

Nota.-

Usted podría obtener su software httpd desde el home page de NCSA. Esto le asegurará que no olvide un archivo que posteriormente requerirá y le permitirá obtener la versión más reciente. La dirección es la siguiente:

*<http://hoohoo.ncsa.uiuc.edu>*

Dentro de esta dirección podrá encontrar códigos fuentes y binarios, tips para la instalación y preguntas frecuentemente contestadas.

Usted podrá obtener los archivos desde NCSA, ya sea como archivos binarios o como archivos fuente.

El archivo binario es un sistema precompilado para varias versiones de Unix. Después de haber obtenido el archivo, simplemente descomprima y desempaquete el archivo y el servidor quedará instalado.

Esta opción se adapta para aquellos administradores que no están familiarizados con la programación y para aquellos sistemas que carecen de un completo sistema de desarrollo.

Utilizar el código fuente es más complejo y no le agrada a toda la gente. El archivo contiene el código fuente completo para el servidor y archivos de soporte. Descomprima y desempaquete el archivo y compile el servidor para soportar los archivos. Esto puede involucrar algunas ediciones y definitivamente involucra compilación; lo cual no es apropiado para administradores novatos de sistemas. Como mínimo, el ambiente debe incluir un compilador, librerías de ejecución y el programa *make*.

#### *¿Qué es make.?*

*Make* es un programa que simplifica la construcción de aplicaciones y sistemas. El programa toma un archivo *makefile* como entrada. Este archivo contiene las instrucciones necesarias para compilar una aplicación o sistema. Si usted posee un programa que consista de tres módulos fuente, deberá compilar y ligar sus módulos cada vez que realice un cambio.

Antes de obtener el archivo binario debemos observar el tipo y la versión de sistema operativo Unix existente sobre nuestro sistema. Para observar esto utilice el comando *uname -a* desde el prompt.

La salida de este comando indica el tipo y la versión de nuestro sistema operativo. El comando *uname* obtiene seis segmentos de información: el sistema operativo utilizado, el nombre del host, la versión del kernel, el número de revisión, la fecha y la hora, y el tipo de microprocesador utilizado dentro del host.

La siguiente tabla muestra los diferentes tipos de sistemas operativos Unix y las versiones para las cuales NCSA provee sistemas precompilados.

UNIX.	Versión.
AIX.	3.2.5
HP-UX.	9.05
IRIX.	4.0.5
IRIX.	5.3
Linux.	1.2.13
OSF/1.	3.0
Solaris.	1.x
Solaris.	2.4 SPARC
Solaris.	2.4 X86
SunOS.	4.1.3
SunOS.	5.4
SunOS.	5.3
Ultrix.	4.0

**Nota.-**

La lista de sistemas soportados cambia mes con mes. Consulte la siguiente página para obtener la actualización de los sistemas soportados.

<http://hoo.hoo.ncsa.uiuc.edu/docs/setup/PreExec.html>

Puede observar que un archivo binario compilado de una versión anterior, puede ser ejecutado sobre la siguiente versión utilizada. Por ejemplo, los archivos binarios compilados de Solaris 2.4 pueden ser ejecutados en Solaris 2.5.

Sin embargo, si su sistema no aparece en la lista deberá compilar el código fuente. Esto no es difícil, pero si usted no es un programador esto será una ardua tarea.

En cualquiera de los dos casos, necesita obtener el software. Primero determine cual tipo quiere -precompilado o fuente-, y posteriormente utilice su visualizador de Web y obtenga el archivo de alguno de los dos servidores Web.

☛ Distribución binaria.

<http://hoohoo.ncsa.uiuc.edu/docs/setup/PreCompiled.html>

☛ Código fuente para distribución.

<http://hoohoo.ncsa.uiuc.edu/docs/setup/Compilation.html>

### 8.3 Descomprimiendo el archivo.

El software viene dentro de un archivo llamado *tar*. *Tar* es un programa de Unix diseñado para comprimir archivos de datos a cintas. De hecho *tar* viene de "tape-archiver".

El programa *compress* es un programa estándar para la compresión de programas en Unix, que está disponible para todos los sistemas Unix. La contraparte de *compress* es *uncompress*. De la misma manera, *gzip* tiene el comando *gunzip* para descompresión, sin embargo, puede utilizar *gzip* para descomprimir también.

Los archivos producidos por estos dos programas pueden ser identificados por sus extensiones: los archivos *compress* terminan con *.Z*, mientras que los archivos *gzip* terminan con *.gz*.

Ahora seleccione un lugar para almacenar la arborescencia de su código fuente. Para el siguiente ejemplo utilizaré un sistema Linux, guardaré el código fuente en la arborescencia de código fuente para todos los otros procesos, siendo esta arborescencia */usr/src*. Los administradores de sistemas Linux generalmente guardan su código fuente debajo de este directorio, y *httpd* no es la excepción. El archivo binario contiene el código fuente y el código ejecutable. Por tanto, el archivo podría ser descomprimido dentro del directorio */usr/src*.

### 8.4 Uncompress y tar.

El primer paso para descomprimir el archivo fuente es ejecutar el programa *uncompress*. El archivo fuente obtenido fue almacenado dentro de */tmp*, aunque el lugar de almacenamiento puede variar. He aquí la línea de comando para *uncompress* y *tar*:

```
root@honey:/usr/src# uncompress /tmp/httpd_1.5a-export_source.tar.Z
```

El programa *uncompress* descomprime el archivo y le quita la extensión *.Z*. El siguiente paso es utilizar *tar* sobre el archivo:

```
root@honey:/usr/src# tar -xvf /tmp/httpd_1.5a-export_source.tar
```

☛ Para consultar la salida del comando remítase al anexo número 4.

```
root@honey:/usr/src#
```

*Tar* creó un nuevo directorio debajo de */usr/src* llamado *httpd\_1.5a-export*.

Este directorio contiene los siguientes elementos:

```
root@honey:/usr/src/httpd_1.5a-export# ls -l
total 34
-rw-r--r-- 1 root admin 145 Nov 14 10:38 BUGS
-rw-r--r-- 1 root admin 9137 Nov 14 10:38 CHANGES
-rw-r--r-- 1 root admin 3251 Nov 14 10:38 COPYRIGHT
-rw-r--r-- 1 root admin 4880 Nov 14 10:38 CREDITS
-rw-r--r-- 1 root admin 1838 Nov 14 10:38 Makefile
-rw-r--r-- 1 root admin 3341 Nov 14 10:38 README
drwxr-xr-x 2 root admin 1024 Nov 14 10:38 cgi-bin
drwxr-xr-x 2 root admin 1024 Nov 14 10:38 cgi-src
drwxr-xr-x 2 root admin 1024 Nov 14 10:38 conf
drwxr-xr-x 2 root admin 1024 Nov 14 10:38 icons
drwxr-xr-x 2 root admin 1024 Nov 14 10:38 src
drwxr-xr-x 2 root admin 1024 Nov 14 10:38 support
```

Los archivos de este directorio son los siguientes:

- ✎ **BUGS.** Este archivo contiene una lista de defectos conocidos en el software.
- ✎ **CHANGES.** Este archivo lista los cambios hechos entre cada versión o actualización del software.
- ✎ **COPYRIGHT.** Este archivo contiene los derechos reservados que se aplican a el archivo que usted obtuvo.
- ✎ **CREDITS.** Este archivo contiene una lista de gente que contribuyó en el proyecto NCSA HTTP.
- ✎ **README.** Este archivo describe que es lo que contiene la distribución.
- ✎ **cgi-bin/.** Este directorio contiene ejemplos de programas binarios de *CGI*.
- ✎ **cgi-src/.** Este directorio contiene el código fuente de los programas de *CGI*.
- ✎ **conf/.** Este directorio contiene ejemplos de archivos de configuración.
- ✎ **icons/.** Este directorio contiene iconos e imágenes disponibles, algunos de los cuales son utilizados en la configuración de ejemplo.
- ✎ **src/.** Este directorio contiene el código fuente para el servidor.
- ✎ **support/.** Este directorio contiene el código fuente para soporte de varios programas relacionados.

### **8.5 Compilando el servidor.**

Si usted obtuvo el código fuente, ahora podría examinarlo dentro del directorio */usr/src/httpd\_1.5a-export*. Si usted obtuvo el código binario puede pasar a la sección denominada "Preparándose para instalar el software", porque el paso de la compilación ya ha sido realizada.

Compilar el servidor NCSA es una simple tarea. El primer paso es identificar el tipo de su sistema dentro de la lista de sistemas soportados. Después de haber identificado su sistema, ejecute el programa *make*.

El *makefile* que se encuentra dentro del servidor NCSA soporta 16 diferentes implementaciones de Unix. Cada uno de estos sistemas tiene un destino *make*. La siguiente tabla indica los destinos para cada uno de estos sistemas:

System.	Target.
AIX 3.2.x	aix3
AIX 4.1.x	aix4
A/UX (Apple Unix).	aux
BSD/OS 2.x	bsdi
HP-UX 9.x using HP cc	hp-cc
HP-UX 9.x using gcc	hp-gcc
Linux	linux
NetBSD	netbsd
NeXTStep 3.x	next
OSF/1 (DEC Alpha)	osf1
IRIX 4.x	sgi4
IRIX 5.x	sgi5
Sun Solaris 2.x (a.k.a. SunOS 5.x)	solaris
Sun Solaris 2.x (a.k.a. Solaris 1.x)	sunos
System 5 Release 4 (Generic)	svr4
Ultrix (DEC MIPS)	ultrix

Estableciendo el servidor es tan simple como seleccionar el sistema destino y ejecutar el programa *make*.

Nota.-

Si usted omite el destino y ejecuta el programa *make* la tabla de destinos será mostrada en pantalla.

Si su sistema no se encuentra dentro de la lista y desea cambiar la configuración por omisión, debe realizar algunos cambios a la distribución por omisión. Existen muchos lugares donde necesitará realizar cambios; NCSA tiene incluidas diferentes modificaciones para diferentes tipos de sistemas dentro de sus archivos *makefile*.

Como el sistema utilizado es un sistema Linux, deberá usar como destino Linux para el programa *make*.

A continuación se muestra la salida del programa *make* para el servidor Linux.

☞ Para consultar la salida del comando remítase al anexo número 5.

```
root@honey:/usr/src/ httpd_1.5a-export#
```

Si todas las cosas fueron satisfactorias, tendrá un archivo llamado *httpd* en su directorio */usr/src/httpd\_1.5a-export*. En su defecto asegúrese de haber seleccionado el sistema correcto y hágalo nuevamente.

## **8.6 Preparándose para instalar el software.**

Antes de instalar el software, necesitará completar los siguientes pasos:

- ☞ Determinar dónde se encontrarán los archivos de soporte del servidor.
- ☞ Determinar cómo se ejecutará el servidor.
- ☞ Determinar cuáles usuarios y grupos serán propios para los procesos del servidor.

### 8.7 ¿ Dónde poner el software. ?

El primer paso para instalar el software es determinar dónde estarán alojados los archivos. Necesita seleccionar un directorio, que será el directorio raíz para la arborescencia completa del servidor. Esto es similar al lugar para la arborescencia del código fuente, pero ahora para los archivos binarios.

La selección de este directorio es completamente del administrador del sistema, aunque una buena opción para establecer este directorio es */usr/local/etc/httpd*. Aunque habría que considerar los siguientes aspectos: espacio del disco, acceso de usuarios, etc.

#### **Espacio del Disco.**

El servidor por sí mismo no ocupa mucho espacio del disco. Sin embargo, el directorio donde colocó el software puede contener todos los documentos e imágenes que el servidor provee; si este es el caso desarrolle un plan acorde, la imágenes pueden incrementar de manera significativa el espacio ocupado de su disco.

#### **Acceso de Usuarios.**

Algunos usuarios requerirán acceder estos directorios (Usted, el Webmaster, el personal del Departamento de Mercadotecnia, etc.). Por tanto, deberá asegurarse que los archivos del sistema que seleccionó para este directorio estén disponibles para todos los usuarios que puedan acceder estos archivos.

### 8.8 ¿ Cómo se ejecutará el servidor. ?

Existen dos opciones para ejecutar su servidor: *standalone* o *spawned*. El servidor *standalone* se ejecuta como un proceso escuchando para conexiones. Cuando una conexión entra, este crea una copia de sí mismo para manipular la solicitud. Este es un proceso rápido y eficiente.

Un servidor *spawned* es "*spawned*" por un proceso llamado *inetd*. *Inetd* es un demonio de Internet; este pequeño individuo es la espina dorsal del ruteo de TCP/IP en un servidor Unix. Fuera de esto, muchos procesos típicos que son dados como seguros (como telnet, ftp y finger), simplemente no trabajan adecuadamente.

Nota.-

*Inetd* es la madre de todos los servidores. Este escucha las solicitudes que llegan de los clientes. Cuando una solicitud llega a cierto puerto, este engendra un programa para manipular aquella solicitud. La configuración de este programa es hecha en un archivo llamado *inetd.conf*, el cual se encuentra alojado generalmente dentro del directorio */etc*.

Para ejecutar su servidor fuera de *inetd*, necesita adicionar una línea para el protocolo *http*. Este protocolo es un cruce referencial con el archivo llamado *services*, que también se encuentra dentro del directorio */etc*. *Services* debe incluir una línea para el protocolo *HTTP* que incluye el nombre (*http*), el tipo de conexión (*tcp*), y el puerto sobre el cual escucha (generalmente 80).

He aquí la línea que podría adicionar a su archivo */etc/services*, si es que este no la tiene:

```
http          tcp/80          # Hypertext Transfer Protocol.
```

Nota.-

Usted podría adicionar esta línea a su archivo */etc/services* sin considerar el modo de ejecución que haya seleccionado. Por adicionar la línea, estará habilitando la referencia de su puerto por nombre, en lugar de por número con programas como *telnet*, *ping*, *ftp*.

He aquí una línea de su archivo *inetd.conf*, si usted selecciona este modo:

```
http stream tcp nowait www /usr/local/etc/httpd/httpd httpd
```

Después de haber realizado algunos cambios a el archivo */etc/inetd.conf*, puede restaurar el proceso *inetd* para enviar la señal de colgar; esto lo realiza con el comando *kill*, puede enviar la señal de colgar con el parámetro *-HUP*.

Este método tiene un inconveniente. Primero es más lento que un servidor *standalone*. Segundo, cada conexión que es recibida resulta en un nuevo proceso *httpd* estando inicializado.

Por instancia, asumimos que una página típica de Web tiene cinco imágenes sobre esta. Un cliente, conectándose a el servidor que se está ejecutando fuera de *inetd*, realizará un total de seis conexiones (una por el documento y cinco más por las imágenes). Cada una de estas conexiones podría resultar en un proceso activo.

Este método también afecta a el desarrollo desfavorablemente. Si su sitio está ocupado podría recibir muchas solicitudes por segundo. Esto puede traer a el funcionamiento de su sistema una pausa.

¿Porqué entonces, podría querer ejecutar su servidor por este medio? Porque utilizando *inetd* en algunas instancias puede ser muy seguro. Este puede ser configurado para ejecutar programas antes de ejecutar *httpd*. Puede reservar un nivel adicional de seguridad dentro de este, para asegurarse de que únicamente acepta solicitudes desde ciertos dominios o direcciones IP.

### **8.9 ¿ Quién podría poseer los procesos del servidor. ?**

La última pieza de configuración que necesita antes de instalar su software, es determinar que usuarios tendrán derechos para utilizar los procesos del servidor. Cuando se encuentra dentro de una máquina Unix y ejecuta un proceso, este es poseído por usted. De la misma manera, tiene la opción de establecer su servidor *httpd* para ser ejecutado por cualquier persona sobre su sistema. Esta característica lo habilita para restringir el acceso a ciertos tipos de archivos y proteger la seguridad del servidor. Existen dos opciones que necesita conocer: *nobody* y *el usuario común*.

La gran parte de los sistemas Unix vienen con un usuario y un grupo registrado llamado *nobody*, los cuales son típicamente usados para conexiones entre máquinas que son consideradas como seguras (esto es, montando NFS). Una de las opciones que se recomiendan es utilizar el usuario *nobody* para ejecutar su servidor.

La segunda sugerencia es crear un usuario y un grupo para tener los procesos del servidor. Esto tiene dos beneficios:

- ⚡ Algunos defectos que existen en otro software para el usuario *nobody* son omitidos. El software que actualmente utiliza el usuario *nobody* puede tener orificios de seguridad, acerca de los cuales están desatendidos. Un *hacker* de Web puede tomar ventaja de semejantes orificios y tener el control de la operación de su sistema. Puede eliminar la posibilidad de una violación de seguridad por no usar *nobody*.
- ⚡ Esto fuerza a que verifique su instalación completa. Pero si está usando una nueva cuenta, no existirán muchos archivos propios para este usuario.

Una recomendación es crear un nuevo usuario llamado *www* y un nuevo grupo llamado *www*. Siendo este el usuario y el grupo que se utilizará para ejecutar el servidor.

### **8.10 Instalando el software.**

Ahora que seleccionamos nuestro directorio principal, el modo de ejecución, el usuario y el grupo, nos encontramos listos para instalar nuestro servidor. A manera de recapitulación, he aquí los elementos que seleccionamos:

- ⚡ **Directorio:** Vamos a estar utilizando el directorio */usr/local/etc/httpd*.
- ⚡ **Método de Ejecución:** Vamos a estar ejecutando nuestro servidor como un servidor *standalone*.
- ⚡ **Propietario del Proceso Usuario/Grupo:** Se creó un usuario *www* y un grupo *www* que serán los propietarios de los procesos del servidor.

Con estos elementos seleccionados, ahora instalaremos nuestros archivos binarios.

### **8.11 Creando la arborescencia del directorio.**

Lo primero que necesitamos es crear la arborescencia donde se alojarán estos archivos. Estos son los directorios que necesitamos crear:

- **/usr/local/etc/httpd.** Esta es la raíz de nuestro servidor. Todos los archivos estarán alojados bajo este directorio.
- **/usr/local/etc/httpd/cgi-bin.** Es aquí donde estarán alojados todos nuestros programas CGI. Estos programas habilitan a el servidor para ejecutar programas externos.
- **/usr/local/etc/httpd/conf.** Este directorio alojará todos los archivos de configuración de nuestro servidor.
- **/usr/local/etc/httpd/logs.** Este directorio alojará todos los archivos de registro de nuestro servidor.
- **/usr/local/etc/httpd/htdocs.** Este directorio alojará todos los documentos de información que proporcionará nuestro servidor.
- **/usr/local/etc/httpd/support.** Este directorio alojará todos los archivos de soporte del servidor.

Por lo tanto, con los siguientes comandos crearemos los subdirectorios que necesitamos:

```
root@honey:~# cd /usr/local/etc
root@honey:/usr/local/etc# mkdir httpd
root@honey:/usr/local/etc# cd httpd
root@honey:/usr/local/etc/httpd# mkdir cgi-bin
root@honey:/usr/local/etc/httpd# mkdir conf
root@honey:/usr/local/etc/httpd# mkdir logs
root@honey:/usr/local/etc/httpd# mkdir htdocs
root@honey:/usr/local/etc/httpd# mkdir support
```

El siguiente paso, es copiar los archivos dentro de los directorios:

```
root@honey:~# cd /usr/local/etc/httpd
root@honey:/usr/local/etc/httpd# cp /usr/src/httpd_1.5a-export/httpd .
root@honey:/usr/local/etc/httpd# cp /usr/src/httpd_1.5a-export /conf/* ./conf
root@honey:/usr/local/etc/httpd# cp /usr/src/httpd_1.5a-export /cgi-bin/* ./cgi-bin
root@honey:/usr/local/etc/httpd# cd conf
root@honey:/usr/local/etc/httpd/conf# cp access.conf-dist access.conf
root@honey:/usr/local/etc/httpd/conf# cp httpd.conf-dist httpd.conf
root@honey:/usr/local/etc/httpd/conf# cp srm.conf-dist srm.conf
```

## **Capítulo 9.**

### **9.1 Visualizadores para Web.**

Para acceder a World Wide Web se utiliza lo que se llama *visualizador*. En ocasiones también se les llama *clientes*, ya que obtienen información de algún *servidor*.

El trabajo de un visualizador es doble: fijado un apuntador en cierto fragmento de información dentro de Internet (llamado URL), debe establecer el acceso a dicho fragmento y realizar determinada tarea de acuerdo con el contenido de la posición del apuntador. Para documentos de hipertexto de Web, lo anterior significa la posibilidad de comunicarse con el servidor utilizando el protocolo HTTP.

Como Web también maneja información de servidores FTP y Gopher, dentro de nuevos sitios postales de correo electrónico y demás tareas semejantes, el visualizador también debe tener la capacidad para comunicarse en el lenguaje de tales herramientas.

No obstante, lo que con mayor frecuencia ejecuta un visualizador es lo relacionado con los documentos de Web. Cada una de las páginas que se cargan en Web representan un documento escrito en el lenguaje llamado HTML (HyperText Markup Language), que incluye el texto del documento, su estructura, los vínculos hacia otros documentos, imágenes y otros medios.

El visualizador habla al servidor de Web en Internet y abre documentos desde ese servidor. Si el documento es un archivo HTML, lo interpreta según el código HTML contenido en tal documento, le da el formato y lo despliega.

Si el documento contiene imágenes o vínculos hacia otros documentos, también ejerce control sobre esos elementos.

A continuación mencionaré una lista de visualizadores existentes dentro del mercado, así como la compañía que genera dicho producto de software:

Producto.	Compañía.
Netscape Navigator.	Netscape Communications Corporation.
NCSA Mosaic.	National Center of Supercomputing Applications.
Lynx.	Universidad de Kansas.
MacWeb.	Trade Web Corporation.
WinWeb.	Trade Web Corporation.
Cello.	Microsoft.
Air Mosaic Express.	Spry, Inc.
SlipKnot.	Peter Brooks of Micromind, Inc.
Pythia.	Appian Interactive Corporation.
Explorer.	Microsoft.

## 9.2 Organización del trabajo.

Cuando se escribe un libro, una noticia, un artículo e incluso un memorándum, por lo general no se comienza con la primera frase y se sigue de corrido hasta el final. Lo mismo sucede con la pintura o el dibujo, normalmente no se comienza por la esquina superior izquierda del lienzo o del papel y se va recorriendo todo el camino hasta concluir en el ángulo inferior derecho.

La mejor manera de escribir, dibujar o diseñar es comenzar elaborando un plan, para saber lo que se debe hacer y el resultado por lograr; también es importante tener una idea general o un esquema preliminar de la estructura antes de entrar de lleno al trabajo.

De la misma manera que se hace con los métodos tradicionales de comunicación, escribir y diseñar páginas Web requiere planeación y reflexión antes de insertar textos y gráficos en ella y vincularlos unos con otros.

Tal vez en este caso se requiera más, porque tratar de aplicar las reglas propias de la escritura y el diseño tradicionales al hipertexto en línea suele terminar en documentos difíciles de entender y de navegar, o simplemente no se aprovechan en ellos las ventajas que ofrece el hipertexto. Además, los documentos de Web mal organizados son más difíciles de renovar y de expandir.

A continuación mencionaré algunos principios recomendables para diseñar una página Web:

- ≍ Defina el propósito de su home page.
- ≍ Realice un análisis del sistema de información.
- ≍ Divida su contenido en asuntos principales.
- ≍ Realice un análisis de los hipermedios a utilizar.
- ≍ Cumpla con los fines de la comunicación.

### **9.3 El contenido, la clave del éxito.**

Quienes desean ganar mucho dinero con Internet deberán pensar en el contenido tal como ocurrió antes con la radio y la televisión.

La revolución de la televisión, que comenzó hace medio siglo, generó una serie de industrias, incluida la fabricación de aparatos receptores, pero los ganadores a largo plazo fueron aquellos que emplearon el medio para difundir información.

Cuando hablamos de una red interactiva, tal como Internet, la definición de "contenido" es muy amplia. Pero la mayoría de las oportunidades para una gran cantidad de empresas, radica en la posibilidad de ofrecer información y entretenimiento y ninguna compañía, por pequeña que sea, está exenta de participar.

Internet también facilita la distribución de información a nivel mundial a un costo inferior para el editor, comparándolo con los medios impresos actuales. Las oportunidades son notables y muchas empresas están diseñando planes para crear contenido destinado a Internet.

Por ejemplo, la red de televisión NBC y Microsoft acordaron en fecha reciente, ingresar conjuntamente al negocio de las noticias interactivas. Estas empresas competirán la propiedad de una red de noticias por cable MSNBC y un servicio de noticias interactivo en Internet. NBC mantendrá el control editorial sobre esa nueva compañía.

Las revistas tienen lectores que comparten intereses comunes. Es fácil suponer que esas comunidades de lectores podrán ser abastecidas también con ediciones electrónicas en la red. Pero una revista no puede limitarse a enviar por módem lo que imprime en el papel. No hay la suficiente profundidad o interactividad en el contenido impreso para superar las desventajas del medio.

Las personas que poseen una computadora tienen que ser recompensadas con información de gran profundidad y muy actualizada que puedan explorar a voluntad. Esas personas necesitan tener audio y posiblemente vídeo. Requieren de una oportunidad de participación personal que vaya más allá de las cartas al director publicadas en diarios y revistas.

Una pregunta que se hacen muchas personas, es si una empresa que satisface las necesidades de un grupo de interés a través de una publicación tendrá éxito cuando haga la misma tarea en línea. Inclusive, el futuro de algunas revistas y periódicos se halla en entredicho debido a Internet.

Por ejemplo, Internet está provocando una revolución en el intercambio de información científica. Las revistas científicas son generalmente de circulación restringida, lo cual las hace muy costosas. Las bibliotecas universitarias constituyen una gran parte del mercado. Ha sido una forma lenta y costosa de distribuir información a una audiencia especializada, pero no había existido una alternativa previamente.

Ahora algunos investigadores están comenzando a emplear la red Internet con el fin de publicar sus hallazgos científicos. Esa práctica pone en riesgo el futuro de algunas venerables publicaciones. Con el transcurso del tiempo, el volumen de información que pasará por Internet será enorme, lo cual la hará imprescindible. Aunque la carrera por el filón de oro, que por un tiempo estuvo confinada principalmente a Estados Unidos alcanzó ya un interés mundial, a medida que baje el costo de las comunicaciones y gran cantidad del contenido esté disponible en diferentes países hará a Internet aún más atractivo.

Pero para que Internet prospere, los proveedores de contenido deberán ser recompensados por su trabajo. Las perspectivas a largo plazo son buenas, pero creo que habrá mucha gente decepcionada en el corto plazo; mientras las empresas que proveen de contenido, tratan de ganar dinero mediante avisos y suscripciones.

Todavía no resulta rentable y tal vez no lo sea durante algún tiempo. Por lo menos hasta ahora, la mayoría del dinero y del esfuerzo invertido en la publicación interactiva no es mucho más que una tarea de amor o un esfuerzo destinado a promover productos que se venden en el mundo al margen de la electrónica. Con frecuencia esos ciertos esfuerzos se basan en la creencia de que con el transcurso del tiempo alguien descubrirá cómo obtener ganancias.

A largo plazo, la posibilidad de publicar productos es promisorio. Una ventaja de la publicidad interactiva es que un mensaje inicial sólo necesita atraer la atención sin brindar excesiva información. Pero en la actualidad las ganancias por suscripciones o anuncios en Internet alcanzan entre 20 y 30 millones de dólares en Estados Unidos. Los anunciantes siempre se muestran algo renuentes cuando aparece un nuevo medio e Internet ciertamente lo es y además bastante diferente.

Algunas empresas que venden contenido están experimentando con suscripciones, con frecuencia, ofreciendo algo gratis. Pero eso tiene sus bemoles, pues tan pronto como una comunidad electrónica cobra una suscripción, la cifra de gente que "visita" el sitio se reduce de manera drástica, haciendo disminuir el atractivo que puede tener para los anunciantes. Aquellos que tengan éxito impulsarán a Internet hacia adelante, hacia un mercado de ideas, experiencias y productos; en definitiva, un mercado de contenido.

#### ***9.4 Seguridad en las comunicaciones.***

La seguridad se ha constituido en uno de los tópicos más importantes para las empresas que desean comercializar sus productos y servicios en Internet. Muchas personas se sienten inseguras de realizar transacciones financieras o cualquier otro tipo de transacción confidencial a través de Internet.

Actualmente, existen dos avances para el desarrollo de la seguridad del comercio electrónico y de las transacciones de información.

El primero de ellos son los firewalls, que se basan en la protección de las fuentes de datos por medio del resguardo individual de los servidores y de las redes. El segundo utiliza el Secure Sockets Layer, que es el estándar en la industria para proveer seguridad en las capas de datos entre aplicaciones de protocolos y protocolos de comunicación.

### **9.5 Secure Sockets Layer.**

El medio mediante el cual se puede observar la información dentro de Internet, es una aplicación conocida como visualizador. Los visualizadores de Internet deben compararse a dos niveles: arquitectura y características. La arquitectura es el factor fundamental en cuanto a prestaciones, posibilidad de ampliación y futuras características. El conjunto de características, es la de mayor importancia para garantizar que el explorador aproveche al máximo las posibilidades del Web; tanto en la visualización de contenidos, como en seguridad, comunicaciones y colaboración, o en la personalización de la experiencia en Internet.

Los visualizadores más populares hoy en día, Netscape y Microsoft Explorer, proporcionan un sistema de seguridad completo.

Microsoft Explorer utiliza el Microsoft Internet Security Framework, que permite a los usuarios de Web comunicarse en forma privada, descargar código en el que pueden confiar e identificarse ante otros en Internet. Los usuarios pueden realizar transacciones y participar en servicios de consumidor en Internet con la misma discreción y seguridad existentes en el mundo real.

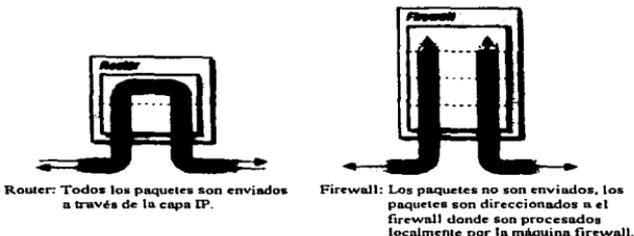
Tanto Microsoft Explorer como Netscape son compatibles con:

- ✦ *Identificación del servidor y cliente.* Utiliza un sistema digital de identificación, un certificado, que identifica al usuario en los servidores de Web. Por su parte, la identificación del servidor garantiza que los usuarios finales hayan establecido contacto con los interlocutores que desean.
- ✦ *Secure Sockets Layer.* Es un protocolo desarrollado por Netscape Communications Corporation que permite a aplicaciones cliente/servidor comunicarse en un sentido que previene escuchar, entrometerse o falsificar mensajes. Secure Sockets Layer provee encriptación de datos, integridad, autenticación del servidor, y autenticación opcional del cliente; además puede soportar una gran variedad de aplicaciones de protocolos (incluyendo HTTP, Telnet, NNTP, SMTP, IMAP, o FTP), y se ejecuta por encima de una gran variedad de protocolos de comunicación (TCP/IP, X.25, IPX/SPX).

## **9.6 Firewalls.**

El término firewall implica protección desde daños. Una computadora con firewall provee un control de acceso estricto entre su sistema y el mundo externo.

El concepto firewall es simple. Un sistema con firewall reemplaza un ruteo IP con un nodo multi-homed que no envía paquetes. Por no enviar paquetes IP entre redes, los firewalls efectivamente sirven a la conexión entre redes.



### **Routers contra Firewalls.**

La figura anterior muestra una comparación entre un ruteador IP y un firewall. Un ruteador maneja paquetes a través de una capa IP. El ruteador envía cada paquete basado sobre la dirección destino de cada uno de ellos y la ruta del destino indicada en la tabla de ruteo. Mientras que en un firewall los paquetes no son enviados; por el contrario, el firewall acepta los paquetes que son direccionados a este y procesa aquellos paquetes a través de la capa de aplicación. El firewall ignora los paquetes que no son enviados a este.

Con un firewall, un intruso no puede meterse dentro del sistema. Los paquetes destinados al host detrás del firewall simplemente no son entregados porque la máquina puede ser el destino de múltiples ataques siendo esto una línea de seguridad muy estricta.

La desventaja de un sistema firewall es obvia. De la misma manera que restringe el acceso desde afuera hacia una red local, este restringe el acceso de una red local hacia afuera del mundo.

### ***9.7 ¿ Cómo hacer rentables las páginas Web. ?***

El gran auge del World Wide Web a traído consigo una encarnizada lucha por la protección de los miembros de Internet. Es por ello que la creación de una buena página puede ser la diferencia entre un servidor lleno de movimiento y uno escasamente visitado. A continuación se presentan una serie de puntos a tomar en cuenta para la construcción de una buena presencia en el Web.

#### ***Cuide el contenido.***

Este es quizás el punto medular para que su home page sea un éxito, los usuarios potenciales buscan acceso a información valiosa 99 de cada 100 veces que accesan un documento. No importa que tan bonita, rápida o avanzada en tecnología sea su página; si se carece de información interesante y útil, no será visitada por los miembros de la red.

#### ***Organización.***

Al igual que con cualquier otro proyecto, haga un análisis cuidadoso y tenga un plan maestro en el que se base para implementar el sistema. Esto le ayudará a presentar la información de una manera clara y concisa, al evitar que los usuarios deban pasar por hojas y hojas con poca o nula utilidad. Recuerde que si el sistema es complicado y difícil de navegar, será poco favorecido por los posibles visitantes. Mantenga constantes ligas hacia los demás lugares del tema en cuestión.

***Cuide la velocidad.***

Resista la tentación de usar todas y cada uno de las características del Web. A menos que los usuarios potenciales tengan una conexión dedicada de 64 Kb será mejor que mantenga un aspecto austero que permita una mayor velocidad de transferencia. Además, diseñe su página en dos versiones: con gráficas y sólo texto, de esta manera los lectores podrán optar por la que sea de su preferencia.

***Mantenimiento.***

Aún el mejor lugar Web del mundo terminará relegado en un "rincón virtual", sino se le da mantenimiento y se actualiza constantemente. De hecho este es uno de los errores más comunes que cometen los novatos, pues la construcción del servidor Web es muy parecida a la de la vialidad en una gran ciudad: nunca termina y siempre hay miles de mejoras posibles para realizar.

***Utilice tecnología de punta.***

Aproveche todos los recursos disponibles en la actualidad. Esto no quiere decir que necesite una Cray y un enlace ATM (Asynchronous Transfer Mode) para lograr un buen sitio Web, lo que significa es que utilice las nuevas tecnologías disponibles para programación de páginas Web, manténgase al tanto por ejemplo de: Java y VRML.

***Cobre cuando sea necesario.***

Si usted desea cobrar por cualquier trozo de información en su página, tenga por seguro que recibirá muy pocas visitas. El perfil del usuario típico de la red y la evolución de la misma, obliga a nuevas maneras de pensar a la hora de hacer negocios a través de World Wide Web. Proporcione buena información de manera gratuita y cobre por la que caiga en la clasificación de excelente.

***Ligas a otros servidores.***

Asegúrese de proporcionar todas las herramientas necesarias para el correcto acceso de su página Web. Por ejemplo, si hace uso del propio Infoseek incluya la liga correspondiente para que los usuarios que aún no conocen dicha herramienta de búsqueda de información a través del Web puedan utilizarla y finalmente, no olvide mencionar las ligas hacia otros lugares de la red con información relacionada a cada tema.

**Conclusiones.**

Internet es la red de computadoras más grande del mundo. Está formada por millones de computadoras de casi todo el mundo conectadas entre sí que intercambian información de todo tipo. En pocas palabras, es un sistema informativo constituido por dos partes: las personas que la utilizan y la información que contiene. Entre ellos no hay nadie: ningún organismo la gobierna, ni la controla; se rige por reglas establecidas por los programadores, en los cuales el respeto a la libertad absoluta es la base.

Internet es un medio que le permite comercializar sus productos y/o servicios. El tema central del enfoque de mercadotecnia en Internet debe basarse en el ofrecimiento de información y ventas directas a personas interesadas. La información que ofrezca puede y debe incluir datos que vayan más allá de lo relacionado con sus productos y/o servicios.

No debe escatimar la información que vaya a ofrecer. Si no ofrece la información suficiente, quizá su producto o servicio no resulte muy atractivo. Cuanto más a menudo actualice la información en su servicio, más personas se interesarán en regresar a visitarlo.

Aún cuando no hay ningún medio o ley que regule el contenido de lo que se publica en la red, existe una cuidadosa vigilancia realizada por los propios usuarios, quienes pueden convertir su página o su buzón de correo electrónico en un verdadero infierno si les desagrada algo.

A casi tres décadas de su nacimiento, la "red de redes" arroja cifras de crecimiento impresionantes: se calcula, conservadoramente, que casi 40 millones de personas de todo el mundo están conectadas a Internet a través de tres millones de computadoras. Además se transfieren alrededor de 150 megabytes de información nueva a través del sistema y esta cantidad aumenta a diario.

Su ritmo de crecimiento es tan acelerado que en realidad nadie sabe que tan grande es; aunque cifras conservadoras calculan su crecimiento en 12% mensual, esto es, entre 200,000 y 300,000 computadoras que se conectan a Internet mensualmente. Siendo aún más interesante, Internet es utilizada por gente de todas las profesiones y condiciones sociales.

De la misma manera, cuanto más bits por segundo pueda ofrecer una conexión, mayor será el ancho de banda. Las líneas telefónicas exclusivas son estupendas porque eliminan, en gran medida, el molesto ruido que es común en las líneas telefónicas. Al encontrar ruido es necesario retransmitir los paquetes de la red y por ende, la velocidad general de la transmisión se reduce. Si hay mucho ruido, se puede perder la señal e interrumpirse la conexión.

Por otra parte, poner una línea telefónica exclusiva resulta costoso, ya que se debe pagar a la compañía telefónica para que tienda una línea entre su compañía y su proveedor de servicios de Internet. El costo de este servicio se calcula de acuerdo con la distancia implicada. Sin embargo, el costo por el uso de líneas exclusivas es menor que en el caso de líneas telefónicas regulares.

En Internet también se encuentran aplicaciones atractivas para las empresas. Dentro de las cuales podemos mencionar: Creación de nuevos clientes, Análisis de productos, Análisis de mercados, Consejos y ayudas de expertos, Ofertas y demandas de empleo, Acceso rápido a la información, Diseminación de productos a gran escala, Comunicaciones rápidas, Coste efectivo en la transferencia de documentos, Comunicación permanente con el mercado destino y Retroalimentación.

Si una persona desea obtener ganancias con Internet, no debe pensar de aquí a 6 o 12 meses, sino tener la mirada puesta en los próximos 10 años; así mismo, debe evaluar que clase de capitales está dispuesta a invertir y el tipo de ingresos que desea obtener.

Una obvia oportunidad es vincular a la gente con Internet. De hecho, las empresas que han tenido la idea de entrar en este negocio han sido bien recompensadas, por lo menos hasta ahora.

Por otra parte, existe una relación entre el protocolo de comunicación y el software de telecomunicaciones, ya que el protocolo marcará las pautas mediante las cuales se realizará la transferencia de información a través del canal de comunicaciones.

Finalmente, cuando se diseñen páginas Web habrá que considerar los diferentes visualizadores existentes dentro del mercado, ya que de lo contrario, habrá la posibilidad de marginar un cierto sector de la población asidua a Internet del contenido de nuestra página.

## Anexo I.

La salida del comando *tar* es la siguiente:

```
WWW/Daemon/Implementation/HTDaemon.c
WWW/Daemon/Implementation/HTDaemon.h
WWW/Daemon/Implementation/HTRequest.c
WWW/Daemon/Implementation/HTRequest.h
WWW/Daemon/Implementation/HTRetrieve.c
WWW/Daemon/Implementation/HTLoad.c
WWW/Daemon/Implementation/HTims.c
WWW/Daemon/Implementation/HTims.h
WWW/Daemon/Implementation/HTCache.c
WWW/Daemon/Implementation/HTCache.h
WWW/Daemon/Implementation/HTCacheInfo.c
WWW/Daemon/Implementation/Htgc.c
WWW/Daemon/Implementation/HTConfig.c
WWW/Daemon/Implementation/HTConfig.h
WWW/Daemon/Implementation/HTWild.c
WWW/Daemon/Implementation/HTWild.h
WWW/Daemon/Implementation/HTScript.c
WWW/Daemon/Implementation/HTScript.h
WWW/Daemon/Implementation/HTPasswd.c
WWW/Daemon/Implementation/HTPasswd.h
WWW/Daemon/Implementation/HTAuth.c
WWW/Daemon/Implementation/HTAuth.h
WWW/Daemon/Implementation/HTLex.c
WWW/Daemon/Implementation/HTLex.h
WWW/Daemon/Implementation/HTGroup.c
WWW/Daemon/Implementation/HTGroup.h
WWW/Daemon/Implementation/HTACL.c
WWW/Daemon/Implementation/HTACL.h
WWW/Daemon/Implementation/HTAAProt.c
WWW/Daemon/Implementation/HTAAProt.h
WWW/Daemon/Implementation/HTAAServ.c
WWW/Daemon/Implementation/HTAAServ.h
WWW/Daemon/Implementation/HTAAFile.c
WWW/Daemon/Implementation/HTAAFile.h
WWW/Daemon/Implementation/HTLog.c
```

WWW/Daemon/Implementation/HTLog.h  
WWW/Daemon/Implementation/HTRFC931.c  
WWW/Daemon/Implementation/HTImage.c  
WWW/Daemon/Implementation/CGIParse.c  
WWW/Daemon/Implementation/cgiutils.c  
WWW/Daemon/Implementation/HTUserInit.c  
WWW/Daemon/Implementation/HTUserInit.h  
WWW/Daemon/Implementation/HTSInit.c  
WWW/Daemon/Implementation/HTSUtils.c  
WWW/Daemon/Implementation/HTSUtils.h  
WWW/Daemon/Implementation/HTAdm.c  
WWW/Daemon/Implementation/CommonMakefile  
WWW/Daemon/Implementation/Version.Make  
WWW/All/  
WWW/All/Implementation/  
WWW/All/Implementation/Makefile  
WWW/All/Implementation/Makefile.product  
WWW/All/Makefile.include  
WWW/All/apollo\_m68k  
WWW/All/apollo\_m68k/Makefile.include  
WWW/All/decstation/  
WWW/All/decstation/Makefile.include  
WWW/All/isc3.0/  
WWW/All/isc3.0/Makefile.include  
WWW/All/linux/  
WWW/All/linux/Makefile.include  
WWW/All/next/  
WWW/All/next/Makefile.include  
WWW/All/next-386/  
WWW/All/next-386/Makefile.include  
WWW/All/osf1/  
WWW/All/osf1/Makefile.include  
WWW/All/rs6000/  
WWW/All/rs6000/Makefile.include  
WWW/All/sco/  
WWW/All/sco/Makefile.include  
WWW/All/sgi/  
WWW/All/sgi/Makefile.include  
WWW/All/snake/  
WWW/All/snake/Makefile.include

WWW/All/src/  
WWW/All/src/Makefile  
WWW/All/src/Makefile.product  
WWW/All/src/  
WWW/All/sun3/  
WWW/All/sun3/Makefile.include  
WWW/All/sun4/  
WWW/All/sun4/Makefile.include  
WWW/All/sun4/Makefile.include.prod  
WWW/All/sun4/Makefile.include~  
WWW/All/sun4-so12/  
WWW/All/sun4-so12/Makefile.include  
WWW/All/sun4-so12/Makefile.include~  
WWW/All/unix/  
WWW/All/unix/Makefile.include  
WWW/All/vax\_ultrix/  
WWW/All/vax\_ultrix/Makefile.include  
WWW/All/aux/  
WWW/All/aux/Makefile.include  
WWW/All/uts4/  
WWW/All/uts4/Makefile.include  
WWW/All/uts2/  
WWW/All/uts2/Makefile.include  
WWW/All/pyramid/  
WWW/All/pyramid/Makefile.include  
WWW/All/ncr/  
WWW/All/ncr/Makefile.include  
WWW/All/dell/  
WWW/All/dell/Makefile.include  
WWW/All/unisys/  
WWW/All/unisys/Makefile.include  
WWW/ICE/  
WWW/ICE/dir.tiff  
WWW/ICE/dir3\_0.wmd  
WWW/ICE/ opendir.tiff  
WWW/ICE/README  
WWW/ICE/doc/  
WWW/ICE/doc/query-syntax.html  
WWW/ICE/scripts/  
WWW/ICE/scripts/dir.tiff

WWW/ICE/scripts/opendir.tiff  
WWW/ICE/scripts/ice-form.pl  
WWW/ICE/scripts/ice-idx.pl  
WWW/ICE/scripts/ice-line.pl  
WWW/ICE/scripts/ice.pl  
WWW/ICE/thesaurus.dat  
WWW/ICE/ICE-1.06a.tar  
WWW/BUILD  
WWW/BUILD.SH  
WWW/Makefile  
WWW/README  
WWW/README-SOCKS  
WWW/server-root/config/  
WWW/server-root/config/httpd.conf  
WWW/server-root/config/proxy.conf  
WWW/server-root/config/caching.conf  
WWW/server-root/config/prot.conf  
WWW/server-root/config/all.conf  
WWW/server-root/icons/  
WWW/server-root/icons/back.xbm  
WWW/server-root/icons/binary.xbm  
WWW/server-root/icons/binhex.xbm  
WWW/server-root/icons/blank.xbm  
WWW/server-root/icons/compressed.xbm  
WWW/server-root/icons/dir.gif  
WWW/server-root/icons/directory.xbm  
WWW/server-root/icons/doc.gif  
WWW/server-root/icons/dummy.xbm  
WWW/server-root/icons/ftp.xbm  
WWW/server-root/icons/image.xbm  
WWW/server-root/icons/index.xbm  
WWW/server-root/icons/index2.xbm  
WWW/server-root/icons/movie.xbm  
WWW/server-root/icons/sound.xbm  
WWW/server-root/icons/tar.xbm  
WWW/server-root/icons/telnet.xbm  
WWW/server-root/icons/text.xbm  
WWW/server-root/icons/unknown.doc.xbm  
WWW/server-root/icons/uu.xbm  
WWW/server-root/icons/unknown.xbm

WWW/server-root/icons/doc.xbm  
WWW/server-root/icons/gzip.xbm  
WWW/server-root/icons/unknown.old.xbm  
WWW/Daemon/Implementation/vms/descrip.mms  
WWW/Daemon/Implementation/vms/build\_multinet.com  
WWW/Daemon/Implementation/vms/setup.com

**Anexo 2.**

La salida del comando *tar* es la siguiente:

```
WWW/Library/Implementation/HTParse.c
WWW/Library/Implementation/HTAccess.c
WWW/Library/Implementation/HTTTP.c
WWW/Library/Implementation/HTFile.c
WWW/Library/Implementation/HTBTree.c
WWW/Library/Implementation/HTMulti.c
WWW/Library/Implementation/HTFTP.c
WWW/Library/Implementation/HTTCP.c
WWW/Library/Implementation/SGML.c
WWW/Library/Implementation/HTML.c
WWW/Library/Implementation/HTMLPDTD.c
WWW/Library/Implementation/HTPlain.c
WWW/Library/Implementation/HTWriter.c
WWW/Library/Implementation/HTFWriter.c
WWW/Library/Implementation/HTMLGen.c
WWW/Library/Implementation/HTTee.c
WWW/Library/Implementation/HTChunk.c
WWW/Library/Implementation/HTAtom.c
WWW/Library/Implementation/HTAnchor.c
WWW/Library/Implementation/HTStyle.c
WWW/Library/Implementation/HTList.c
WWW/Library/Implementation/HTString.c
WWW/Library/Implementation/HTAlert.c
WWW/Library/Implementation/HTRules.c
WWW/Library/Implementation/HTFormat.c
WWW/Library/Implementation/HTInit.c
WWW/Library/Implementation/HTMIME.c
WWW/Library/Implementation/HTHistory.c
WWW/Library/Implementation/HTNews.c
WWW/Library/Implementation/HTGopher.c
WWW/Library/Implementation/HTTelnet.c
WWW/Library/Implementation/HTWAIS.c
WWW/Library/Implementation/HTWSRC.c
WWW/Library/Implementation/HTAAUtil.c
WWW/Library/Implementation/HTAABrow.c
```

WWW/Library/Implementation/HTAssoc.c  
WWW/Library/Implementation/HTUU.c  
WWW/Library/Implementation/HTTeXGen.c  
WWW/Library/Implementation/HTDirBrw.c  
WWW/Library/Implementation/HTDescript.c  
WWW/Library/Implementation/HTGuess.c  
WWW/Library/Implementation/HTIcons.c  
WWW/Library/Implementation/HTError.c  
WWW/Library/Implementation/HTErrorMsg.c  
WWW/Library/Implementation/HTParse.h  
WWW/Library/Implementation/HTAccess.h  
WWW/Library/Implementation/HTTP.h  
WWW/Library/Implementation/HTFile.h  
WWW/Library/Implementation/HTMulti.h  
WWW/Library/Implementation/HTBTree.h  
WWW/Library/Implementation/HTFTP.h  
WWW/Library/Implementation/HTTCP.h  
WWW/Library/Implementation/SGML.h  
WWW/Library/Implementation/HTML.h  
WWW/Library/Implementation/HTMLPDTD.h  
WWW/Library/Implementation/HTPlain.h  
WWW/Library/Implementation/HTWriter.h  
WWW/Library/Implementation/HTFWriter.h  
WWW/Library/Implementation/HTMLGen.h  
WWW/Library/Implementation/HTStream.h  
WWW/Library/Implementation/HTTee.h  
WWW/Library/Implementation/HTAtom.h  
WWW/Library/Implementation/HTAnchor.h  
WWW/Library/Implementation/HTStyle.h  
WWW/Library/Implementation/HTList.h  
WWW/Library/Implementation/HTString.h  
WWW/Library/Implementation/HTAlert.h  
WWW/Library/Implementation/HTRules.h  
WWW/Library/Implementation/HTFormat.h  
WWW/Library/Implementation/HTInit.h  
WWW/Library/Implementation/HTIME.h  
WWW/Library/Implementation/HTHistory.h  
WWW/Library/Implementation/HTNews.h  
WWW/Library/Implementation/HTGopher.h  
WWW/Library/Implementation/HTUtils.h

WWW/Library/Implementation/tcp.h  
WWW/Library/Implementation/HText.h  
WWW/Library/Implementation/HTTlnet.h  
WWW/Library/Implementation/HTWAIS.h  
WWW/Library/Implementation/HTWSRC.h  
WWW/Library/Implementation/HTAAUtil.h  
WWW/Library/Implementation/HTAABrow.h  
WWW/Library/Implementation/HTAssoc.h  
WWW/Library/Implementation/HTUU.h  
WWW/Library/Implementation/HTTeXGen.h  
WWW/Library/Implementation/HTDirBrw.h  
WWW/Library/Implementation/HTDescript.h  
WWW/Library/Implementation/HTGuess.h  
WWW/Library/Implementation/HTIcons.h  
WWW/Library/Implementation/HTError.h  
WWW/Library/Implementation/Version.make  
WWW/Library/Implementation/CommonMakefile  
WWW/Library/Implementation/Makefile  
WWW/README.txt  
WWW/Copyright.txt  
WWW/BUILD  
WWW/Makefile  
WWW/All/apollo\_m68k/Makefile.include  
WWW/All/aux/Makefile.include  
WWW/All/decstation/Makefile.include  
WWW/All/dell/Makefile.include  
WWW/All/isc3.0/Makefile.include  
WWW/All/linux/Makefile.include  
WWW/All/ncr/Makefile.include  
WWW/All/next-386/Makefile.include  
WWW/All/next/Makefile.include  
WWW/All/osf1/Makefile.include  
WWW/All/pyramid/Makefile.include  
WWW/All/rs6000/Makefile.include  
WWW/All/sco/Makefile.include  
WWW/All/sgi/Makefile.include  
WWW/All/snake/Makefile.include  
WWW/All/sun3/Makefile.include  
WWW/All/sun4-sol2/Makefile.include  
WWW/All/sun4/Makefile.include

WWW/All/unisys/Makefile.include  
WWW/All/unix/Makefile.include  
WWW/All/uts2/Makefile.include  
WWW/All/uts4/Makefile.include  
WWW/All/vax\_ultrix/Makefile.include  
WWW/All/Implementation/Makefile  
WWW/All/Implementation/Makefile.product  
WWW/Library/Implementation/vms/descrip.rms  
WWW/Library/Implementation/vms/COPYING.LIB  
WWW/Library/Implementation/vms/build\_multinet.com  
WWW/Library/Implementation/vms/buiid\_multinet.com\_alpha  
WWW/Library/Implementation/vms/multinet.opt  
WWW/Library/Implementation/vms/multinet.opt\_alpha  
WWW/Library/Implementation/vms/build\_ucx.com  
WWW/Library/Implementation/vms/build\_ucx.com\_alpha  
WWW/Library/Implementation/vms/ucx.opt  
WWW/Library/Implementation/vms/ucx.opt\_alpha  
WWW/Library/Implementation/vms/patchlevel.h  
WWW/Library/Implementation/vms/ufc-crypt.h  
WWW/Library/Implementation/vms/crypt.c  
WWW/Library/Implementation/vms/crypt\_util.c  
WWW/Library/Implementation/vms/getline.c  
WWW/Library/Implementation/vms/getpass.c  
WWW/Library/Implementation/vms/HTVMUtils.h  
WWW/Library/Implementation/vms/HTVMUtils.c  
WWW/Library/Implementation/vms/dirent.h  
WWW/Library/Implementation/vms/dirent.c  
WWW/Library/Implementation/vms/sys\_dirent.h  
WWW/All/  
WWW/All/Implementation/  
WWW/All/Implementation/Makefile  
WWW/All/Implementation/Makefile.product  
WWW/All/Makefile.include  
WWW/All/apollo\_m68k/  
WWW/All/apollo\_m68k/Makefile.include  
WWW/All/decstation/  
WWW/All/decstation/Makefile.include  
WWW/All/isc3.0/  
WWW/All/isc3.0/Makefile.include  
WWW/All/linux/

**WWW/All/linux/Makefile.include**  
**WWW/All/next/**  
**WWW/All/next/Makefile.include**  
**WWW/All/next-386/**  
**WWW/All/next-386/Makefile.include**  
**WWW/All/osf1/**  
**WWW/All/osf1/Makefile.include**  
**WWW/All/rs6000/**  
**WWW/All/rs6000/Makefile.include**  
**WWW/All/sco/**  
**WWW/All/sco/Makefile.include**  
**WWW/All/sgi/**  
**WWW/All/sgi/ Makefile.include**  
**WWW/All/snake/**  
**WWW/All/snake/Makefile.include**  
**WWW/All/src/**  
**WWW/All/src/Makefile**  
**WWW/All/src/Makefile.product**  
**WWW/All/src/!**  
**WWW/All/sun3/**  
**WWW/All/sun3/Makefile.include**  
**WWW/All/sun4/**  
**WWW/All/sun4/Makefile.include**  
**WWW/All/sun4/Makefile.include.prod**  
**WWW/All/sun4/ Makefile.include~**  
**WWW/All/sun4-sol2/**  
**WWW/All/sun4-sol2/Makefile.include**  
**WWW/All/sun4-sol2/Makefile.include~**  
**WWW/All/unix/**  
**WWW/All/unix/Makefile.include**  
**WWW/All/vax\_ultrix/**  
**WWW/All/vax\_ultrix/Makefile.include**  
**WWW/All/aux/**  
**WWW/All/aux/Makefile.include**  
**WWW/All/uts4/**  
**WWW/All/uts4/Makefile.include**  
**WWW/All/uts2/**  
**WWW/All/uts2/Makefile.include**  
**WWW/All/pyramid/**  
**WWW/All/pyramid/Makefile.include**

WWW/All/ncr/  
WWW/All/ncr/Makefile.include  
WWW/All/deil/  
WWW/All/deil/Makefile.include  
WWW/All/unisys/  
WWW/All/unisys/Makefile.include  
WWW/Library/Implementation/HTAABrow.html  
WWW/Library/Implementation/HTAAUtl.html  
WWW/Library/Implementation/HTAAccess.html  
WWW/Library/Implementation/HTAlert.html  
WWW/Library/Implementation/HTAnchor.html  
WWW/Library/Implementation/HTAssoc.html  
WWW/Library/Implementation/HTAtom.html  
WWW/Library/Implementation/HTBTree.html  
WWW/Library/Implementation/HTChunk.html  
WWW/Library/Implementation/HTDescript.html  
WWW/Library/Implementation/HTDirBrw.html  
WWW/Library/Implementation/HTError.html  
WWW/Library/Implementation/HTFTP.html  
WWW/Library/Implementation/HTFWriter.html  
WWW/Library/Implementation/HTFile.html  
WWW/Library/Implementation/HTFormat.html  
WWW/Library/Implementation/HTGopher.html  
WWW/Library/Implementation/HTGuess.html  
WWW/Library/Implementation/HTHistory.html  
WWW/Library/Implementation/HTIcons.html  
WWW/Library/Implementation/HTInit.html  
WWW/Library/Implementation/HTList.html  
WWW/Library/Implementation/HTMIME.html  
WWW/Library/Implementation/HTML.html  
WWW/Library/Implementation/HTMLDTD.html  
WWW/Library/Implementation/HTMLGen.html  
WWW/Library/Implementation/HTMLPDTD.html  
WWW/Library/Implementation/HTMulti.html  
WWW/Library/Implementation/HTNews.html  
WWW/Library/Implementation/HTParse.html  
WWW/Library/Implementation/HTPlain.html  
WWW/Library/Implementation/HTRules.html  
WWW/Library/Implementation/HTStream.html  
WWW/Library/Implementation/HTString.html

**WWW/Library/Implementation/HTStyle.html**  
**WWW/Library/Implementation/HTTCP.html**  
**WWW/Library/Implementation/HTTP.html**  
**WWW/Library/Implementation/HTTeXGen.html**  
**WWW/Library/Implementation/HTTeE.html**  
**WWW/Library/Implementation/HTTelnet.html**  
**WWW/Library/Implementation/HTUU.html**  
**WWW/Library/Implementation/HTUtils.html**  
**WWW/Library/Implementation/HTWAIS.html**  
**WWW/Library/Implementation/HTWSRC.html**  
**WWW/Library/Implementation/HTWriter.html**  
**WWW/Library/Implementation/HTText.html**  
**WWW/Library/Implementation/Overview.html**  
**WWW/Library/Implementation/SGML.html**  
**WWW/Library/Implementation/WWW.html**  
**WWW/Library/Implementation/tcp.html**

**Anexo 3.**

```

root@honey:/usr/src/WWW# make
/BUILD
WWW build for machine type:                linux
make[1]: Entering directory '/usr/src/WWW/All/Implementation'
----- Building libwww -----
Object files of libwww will go to directory Library/linux
- creating
make[2]: Entering directory '/usr/src/WWW/Library/linux'
echo Include files generated from hypertext.
Include files generated from hypertext.
if [ ! -r ../. ]; then mkdir ../.; else echo OK ; fi
OK
if [ ! -r ../Library ] ; then mkdir ../Library; else echo OK ; fi
OK
if [ ! -r ../Library/linux ] ; \
then mkdir ../Library/linux; else echo OK ; fi
OK
touch ../Library/linux/created
cc -c -o ../Library/linux/HTParse.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTParse.c
cc -c -o ../Library/linux/HTAccess.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTAccess.c
cc -c -o ../Library/linux/HTTP.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTTP.c
cc -c -o ../Library/linux/HTFile.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTFile.c
cc -c -o ../Library/linux/HTBTree.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTBTree.c
cc -c -o ../Library/linux/HTFTP.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTFTP.c
cc -c -o ../Library/linux/HTTCP.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTTCP.c
cc -c -o ../Library/linux/SGML.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/SGML.c
cc -c -o ../Library/linux/HTML.o -DDEBUG -DPOIXWAIT -I../Library/
Implementation/ ../Library/Implementation/HTML.c
cc -c -o ../Library/linux/HTMLPDTD.o -DDEBUG -DPOIXWAIT -I../Library/

```

⊕ Implementation/ ././Library/Implementation/HTMLPDTD.c  
cc -c -o ././Library/linux/HTChunk.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTChunk.c  
cc -c -o ././Library/linux/HTPlain.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTPlain.c  
cc -c -o ././Library/linux/HTWriter.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTWriter.c  
cc -c -o ././Library/linux/HTFWriter.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTFWriter.c  
cc -c -o ././Library/linux/HTMLGen.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTMLGen.c  
cc -c -o ././Library/linux/HTTee.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTTee.c  
cc -c -o ././Library/linux/HTAtom.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTAtom.c  
cc -c -o ././Library/linux/HTAnchor.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTAnchor.c  
cc -c -o ././Library/linux/HTStyle.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTStyle.c  
cc -c -o ././Library/linux/HTList.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTList.c  
cc -c -o ././Library/linux/HTString.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ -DVC="2.17" ././Library/Implementation/HTString.c  
cc -c -o ././Library/linux/HTAlert.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTAlert.c  
cc -c -o ././Library/linux/HTRules.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTRules.c  
cc -c -o ././Library/linux/HTFormat.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTFormat.c  
cc -c -o ././Library/linux/HTInit.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTInit.c  
cc -c -o ././Library/linux/HTMIME.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTMIME.c  
cc -c -o ././Library/linux/HTHistory.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTHistory.c  
cc -c -o ././Library/linux/HTNews.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTNews.c  
cc -c -o ././Library/linux/HTGopher.o -DDEBUG -DPOSIXWAIT -I././Library/  
⊕ Implementation/ ././Library/Implementation/HTGopher.c  
cc -c -o ././Library/linux/HTTclnet.o -DDEBUG -DPOSIXWAIT -I././Library/

```

$ Implementation/ ././Library/Implementation/HTTNet.c
cc -c -o ././Library/linux/HTWSRC.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTWSRC.c
cc -c -o ././Library/linux/HTAAUtil.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTAAUtil.c
cc -c -o ././Library/linux/HTAABrow.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTAABrow.c
cc -c -o ././Library/linux/HTAssoc.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTAssoc.c
cc -c -o ././Library/linux/HTUU.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTUU.c
cc -c -o ././Library/linux/HTMultic.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTMultic.c
cc -c -o ././Library/linux/HTTeXGen.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTTeXGen.c
cc -c -o ././Library/linux/HTDirBrw.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTDirBrw.c
cc -c -o ././Library/linux/HTDescript.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTDescript.c
cc -c -o ././Library/linux/HTGuess.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTGuess.c
cc -c -o ././Library/linux/HTIcons.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTIcons.c
cc -c -o ././Library/linux/HTError.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTError.c
cc -c -o ././Library/linux/HTErrorMsg.o -DDEBUG -DPOIXWAIT -I././Library/
$ Implementation/ ././Library/Implementation/HTErrorMsg.c
rm ././Library/linux/libwww.a
rm: ././Library/linux/libwww.a: No such file or directory
make[2]: [././Library/linux/libwww.a] Error 1 (ignored)
ar r ././Library/linux/libwww.a ././Library/linux/HTParse.o
././Library/linux/HTAccess.o ././Library/linux/HTTP.o
././Library/linux/HTFile.o ././Library/linux/HTBTree.o
././Library/linux/HTFTP.o ././Library/linux/HTTCP.o
././Library/linux/SGML.o ././Library/linux/HTML.o
././Library/linux/HTMLPDTD.o ././Library/linux/HTChunk.o
././Library/linux/HTPlain.o ././Library/linux/HTWriter.o
././Library/linux/HTFWriter.o ././Library/linux/HTMLGen.o
././Library/linux/HTTee.o ././Library/linux/HTAtom.o
././Library/linux/HTAnchor.o ././Library/HTStyle.o

```

```

../Library/linux/HTList.o ../Library/linux/HTString.o
../Library/linux/HTAlert.o ../Library/linux/HTRules.o
../Library/linux/HTFormat.o ../Library/linux/HTInit.o
../Library/linux/HTMIME.o ../Library/linux/HTHistory.o
../Library/linux/HTNews.o ../Library/linux/HTGopher.o
../Library/linux/HTTelnet.o ../Library/linux/HTWSRC.o
../Library/linux/HTAAUtil.o ../Library/linux/HTAABrow.o
../Library/linux/HTAssoc.o ../Library/linux/HTUU.o
../Library/linux/HTMulti.o ../Library/linux/HTTeXGen.o
../Library/linux/HTDirBrw.o ../Library/linux/HTDescribe.o
../Library/linux/HTGuess.o ../Library/linux/HTIcons.o
../Library/linux/HTError.o ../Library/linux/libwww.a
ranlib ../Library/linux/libwww.a
make [2]: Entering directory '/usr/src/WWW/Daemon/linux'
cp ../Daemon/Implementation/HTDaemon.c HTDaemonDIR.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT -
DRULE_FILE="" "/etc/httpd.conf"" \
-DVD=""3.0"" ../Daemon/Implementation/HTRequest.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT -
DRULE_FILE="" "/etc/httpd.conf"" ../Daemon/Implementation/HTRetrive.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT -
DRULE_FILE="" "/etc/httpd.conf"" \
-DVD=""3.0"" ../Daemon/Implementation/HTScript.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT
../
Ⓜ Daemon/Implementation/HTLoad.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT
../
Ⓜ Daemon/Implementation/HTCache.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT
../
Ⓜ Daemon/Implementation/HTCacheInfo.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT
../
Ⓜ Daemon/Implementation/HTConfig.c
cc -c -I../Library/Implementation -I../Implementation -DDEBUG -DPOSIXWAIT
../
Ⓜ Daemon/Implementation/HTWild.c
cc -c -DDEBUG -DPOSIXWAIT -I../Library/Implementation
Ⓜ ../Daemon/Implementation/HTSInit.c

```

```
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
% -I../Library/Implementation././Daemon/Implementation/HTSUtils.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTIm.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTPasswd.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTAuth.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTLex.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTGroup.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTACL.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTAAProt.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTAAServ.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTAAF.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTLog.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTgc.c
cc -c -DDEBUG -DPOSIXWAIT -I../Library/Implementation
% ../Daemon/Implementation/HTUserInit.c
cc -c -I../Library/Implementation -I./Implementation -DDEBUG -DPOSIXWAIT
../
% Daemon/Implementation/HTRFC931.c
```

```

cc -o httpd_3.0 HTDaemonDIR.o HTRrequest.o HTRetrieve.o HTScript.o HTLoad.o
% HTCACHE.o HTCACHEInfo.o HTConfig.o HTWild.o HTSInit.o HTSUtils.o HTImS.o
% HTPasswd.o HTAuth.o HTLex.o HTGroup.o HTACL.o HTAAProt.o
HTAAAServ.o
% HTAAFile.o HTLog.o HTGc.o HTUserInit.o HTRFC931.o
./../Library/linux/libwww.a
rm httpd
rm: httpd: No such file or directory
make[2]: [httpd] Error 1 (ignored)
if [ linux != isc3.0 +; then ln -s httpd_3.0 httpd; fi
cc -o htadm -I../Library/Implementation -I./Implementation -DDEBUG
-DPOIXWAIT ../Daemon/Implementation/HTAdm.c HTAsswd.o HTAAFile.o \
../Library/linux/libwww.a
cc -o htimage -I../Library/Implementation -I./Implementation -DDEBUG
% -DPOIXWAIT ../Daemon/Implementation/HTImage.c
% ../Library/linux/libwww.a
% cc -o cgparsc -I../Library/Implementation -I./Implementation -DDEBUG
% -DPOIXWAIT ../Daemon/Implementation/CGIParsc.c
% ../Library/linux/libwww.a
cc -o cgiutils -I../Library/Implementation -I./Implementation -DDEBUG
% -DPOIXWAIT ../Daemon/Implementation/cgiutils.c HTSUtils.o \
% ../Library/linux/libwww.a
make[2]: Leaving directory /usr/src/WWW/Daemon/linux'
CERN httpd, htadm, htimage, chiparsc and cgiutils built successfully and can be
found in directory /Daemon/linux.

```

Have fun! If you have any problems with this software feel free to contact [htpd@info.cern.ch](mailto:htpd@info.cern.ch). Online documentation is available via [root@honey:/usr/src/WWW#](mailto:root@honey:/usr/src/WWW#)

**Anexo 4.**

La salida del comando *tar* es la siguiente:

```
httpd_1.5-export/  
httpd_1.5-export/cgi-bin/archie  
httpd_1.5-export/cgi-bin/calendar  
httpd_1.5-export/cgi-bin/date  
httpd_1.5-export/cgi-bin/donothing  
httpd_1.5-export/cgi-bin/finger  
httpd_1.5-export/cgi-bin/fortune  
httpd_1.5-export/cgi-bin/mail  
httpd_1.5-export/cgi-bin/nph-error.pl  
httpd_1.5-export/cgi-bin/nph-test-cgi  
httpd_1.5-export/cgi-bin/redirect  
httpd_1.5-export/cgi-bin/test-cgi  
httpd_1.5-export/cgi-bin/test-cgi.tcl  
httpd_1.5-export/cgi-bin/test-env  
httpd_1.5-export/cgi-bin/uptime  
httpd_1.5-export/cgi-bin/wais.pl  
httpd_1.5-export/cgi-src/  
httpd_1.5-export/cgi-src/Makefile  
httpd_1.5-export/cgi-src/change-passwd.c  
httpd_1.5-export/cgi-src/imagemap.c  
httpd_1.5-export/cgi-src/jj.c  
httpd_1.5-export/cgi-src/phf.c  
httpd_1.5-export/cgi-src/post-query.c  
httpd_1.5-export/cgi-src/query.c  
httpd_1.5-export/cgi-src/util.c  
httpd_1.5-export/cgi-src/util.h  
httpd_1.5-export/BUGS  
httpd_1.5-export/CHANGES  
httpd_1.5-export/COPYRIGHT  
httpd_1.5-export/CREDITS  
httpd_1.5-export/Makefile  
httpd_1.5-export/README  
httpd_1.5-export/conf/  
httpd_1.5-export/conf/access.conf-dist  
httpd_1.5-export/conf/httpd.conf-dist
```

httpd\_1.5-export/conf/localhost\_srm.conf-dist  
httpd\_1.5-export/conf/mime.types  
httpd\_1.5-export/conf/srm.conf-dist  
httpd\_1.5-export/icons/  
httpd\_1.5-export/icons/back.xbm  
httpd\_1.5-export/icons/ball.gif  
httpd\_1.5-export/icons/ball.xbm  
httpd\_1.5-export/icons/binary.gif  
httpd\_1.5-export/icons/binary.xbm  
httpd\_1.5-export/icons/blank.xbm  
httpd\_1.5-export/icons/blue\_ball.gif  
httpd\_1.5-export/icons/ftp.gif  
httpd\_1.5-export/icons/ftp.xbm  
httpd\_1.5-export/icons/green\_ball.gif  
httpd\_1.5-export/icons/image.gif  
httpd\_1.5-export/icons/image.xbm  
httpd\_1.5-export/icons/index.gif  
httpd\_1.5-export/icons/index.xbm  
httpd\_1.5-export/icons/menu.gif  
httpd\_1.5-export/icons/menu.xbm  
httpd\_1.5-export/icons/movie.gif  
httpd\_1.5-export/icons/movie.xbm  
httpd\_1.5-export/icons/red\_ball.gif  
httpd\_1.5-export/icons/sound.gif  
httpd\_1.5-export/icons/sound.xbm  
httpd\_1.5-export/icons/telnet.gif  
httpd\_1.5-export/icons/telnet.xbm  
httpd\_1.5-export/icons/text.gif  
httpd\_1.5-export/icons/text.xbm  
httpd\_1.5-export/icons/unknown.gif  
httpd\_1.5-export/icons/unknown.xbm  
httpd\_1.5-export/src/  
httpd\_1.5-export/src/cgi.c  
httpd\_1.5-export/src/cgi.h  
httpd\_1.5-export/src/config.h  
httpd\_1.5-export/src/constants.h  
httpd\_1.5-export/src/digest.c  
httpd\_1.5-export/src/digest.h  
httpd\_1.5-export/src/env.c  
httpd\_1.5-export/src/env.h

httpd\_1.5-export/src/fdwrap.c  
httpd\_1.5-export/src/fdwrap.h  
httpd\_1.5-export/src/global.h  
httpd\_1.5-export/src/host\_config.c  
httpd\_1.5-export/src/host\_config.h  
httpd\_1.5-export/src/http\_access.c  
httpd\_1.5-export/src/http\_access.h  
httpd\_1.5-export/src/http\_alias.c  
httpd\_1.5-export/src/http\_alias.h  
httpd\_1.5-export/src/http\_auth.c  
httpd\_1.5-export/src/http\_auth.h  
httpd\_1.5-export/src/http\_config.c  
httpd\_1.5-export/src/http\_config.h  
httpd\_1.5-export/src/http\_dir.c  
httpd\_1.5-export/src/http\_dir.h  
httpd\_1.5-export/src/http\_include.c  
httpd\_1.5-export/src/http\_include.h  
httpd\_1.5-export/src/http\_ipc.c  
httpd\_1.5-export/src/http\_ipc.h  
httpd\_1.5-export/src/http\_log.c  
httpd\_1.5-export/src/http\_log.h  
httpd\_1.5-export/src/http\_mime.c  
httpd\_1.5-export/src/http\_mime.h  
httpd\_1.5-export/src/http\_request.c  
httpd\_1.5-export/src/http\_request.h  
httpd\_1.5-export/src/httpd.c  
httpd\_1.5-export/src/http\_send.c  
httpd\_1.5-export/src/http\_send.h  
httpd\_1.5-export/src/httpd.h  
httpd\_1.5-export/src/httpy.h  
httpd\_1.5-export/src/imagemap.c  
httpd\_1.5-export/src/imagemap.h  
httpd\_1.5-export/src/md5.c  
httpd\_1.5-export/src/md5.h  
httpd\_1.5-export/src/md5c.c  
httpd\_1.5-export/src/open\_logfile.c  
httpd\_1.5-export/src/open\_logfile.h  
httpd\_1.5-export/src/portability.h  
httpd\_1.5-export/src/rfc931.c  
httpd\_1.5-export/src/util.c

httpd\_1.5-export/src/util.h  
httpd\_1.5-export/src/Makefile  
httpd\_1.5-export/support/auth/  
httpd\_1.5-export/support/auth/pgp-dec  
httpd\_1.5-export/support/auth/pgp-enc  
httpd\_1.5-export/support/auth/ripem-dec  
httpd\_1.5-export/support/auth/ripem-enc  
httpd\_1.5-export/support/auth/uudecode.c  
httpd\_1.5-export/support/auth/uencode.c  
httpd\_1.5-export/support/Makefile  
httpd\_1.5-export/support/README  
httpd\_1.5-export/support/README.change-passwd  
httpd\_1.5-export/support/change-passwd.reade  
httpd\_1.5-export/support/dbm2std.c  
httpd\_1.5-export/support/dbmdigest.c  
httpd\_1.5-export/support/dbmgroup.c  
httpd\_1.5-export/support/dbmpasswd.c  
httpd\_1.5-export/support/inc2shtml.c  
httpd\_1.5-export/support/std2dbm.c  
httpd\_1.5-export/support/unescape.c

**Anexo 5.**

```

root@honey:/usr/src/httpd_1.5a-export# make linux
cd src ; make linux ; cd ../cgi-src ; make linux ; cd ../support ; make linux
make[1]: Entering directory '/usr/src/httpd_1.5a-export/src'
make tar AUX_CFLAGS=DLINUX CC=gcc CFLAGS=-O2 DBM_LIBS=-lgdbm
make[2]: Entering directory '/usr/src/httpd_1.5a-export/src'
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT httpd.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_config.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_request.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT util.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_dir.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_alias.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_log.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_mime.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_access.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_auth.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_send.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT cgi.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_include.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT rfc931.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT imagemap.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT http_ipc.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT digest.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT md5.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT md5c.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT env.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT host_config.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT fdwrap.c
gcc -c -O2 -DLINUX -DDIGEST_AUTH -DDBM_SUPPORT open_logfile.c
gcc -o ../httpd httpd.o http_config.o http_request.o util.o http_dir.o
httpalias.o http_log.o http_mime.o http_access.o http_auth.o http_send.o
cgi.o http_include.o rfc931.o imagemap.o http_ipc.o digest.o md5.o md5c.o
env.o host_config.o fdwrap.o open_logfile.o -lgdbm
make[2]: Leaving directory '/usr/src/httpd_1.5a-export/src'
make[1]: Leaving directory '/usr/src/httpd_1.5a-export/src'
make[1]: Entering directory '/usr/src/httpd_1.5a-export/cgi-src'
make all CC=gcc
make[2]: Entering directory '/usr/src/httpd_1.5a-export/cgi-src'

```

```
gcc -c -g query.c
gcc -c -g util.c
gcc query.o util.o -o ../cgi-bin/query
gcc -c -g post-query.c
gcc post-query.o util.o -o ../cgi-bin/post-query
gcc -c -g imagemap.c
gcc imagemap.o -o ../cgi-bin/imagemap
gcc -c -g jj.c
gcc jj.o util.o -o ../cgi-bin/jj
gcc -c -g phf.c
gcc phf.o util.o -o ../cgi-bin/phf
make[2]: Leaving directory '/usr/src/httpd_1.5a-export/cgi-src'
make[1]: Leaving directory '/usr/src/httpd_1.5a-export/cgi-src'
make[1]: Entering directory '/usr/src/httpd_1.5a-export/support'
make all CC=gcc CFLAGS="-DLINUX" EXTRA_LIBS=-lgdbm
make[2]: Entering directory '/usr/src/httpd_1.5a-export/support'
gcc -DLINUX -I../src htpasswd.c -o htpasswd -lgdbm
gcc -DLINUX -I../src unescape.c -o unescape
gcc -DLINUX -I../src inc2shtml.c -o inc2shtml
gcc -c -DLINUX -I../src htdigest.c
gcc -DLINUX -o htdigest htdigest.o ../src/md5.o ../src/md5c.o -lgdbm
gcc -DLINUX -I../src dbm2std.c -o dbm2std -lgdbm
gcc -DLINUX -I../src std2dbm.c -o std2dbm -lgdbm
gcc -c -DLINUX -I../src dbmdigest.c
gcc -DLINUX -o dbmdigest dbmdigest.o ../src/md5.o ../src/md5c.o -lgdbm
gcc -DLINUX -I../src dbmgroup.c -o dbmgroup -lgdbm
gcc -DLINUX -I../src dbmpasswd.c -o dbmpasswd -lgdbm
make[2]: Leaving directory '/usr/src/httpd_1.5a-export/support'
make[1]: Leaving directory '/usr/src/httpd_1.5a-export/support'
```

**Bibliografía.**

- ☛ Craig Hunt, TCP/IP Network Administration, O'Reilly & Associates, Inc., U.S.A., 1996.
- ☛ Harley Hann, Internet, Manual de Referencia, Mc. Graw Hill, México, 1994.
- ☛ Jerry Ablan and Scott Yanoff, Web Site Administrators Survival Guide, Sams Net, U.S.A., 1996.
- ☛ Douglas E. Comer, Internetworking with TCP/IP, Vol. 1. Principles, Protocols and Architecture, Prentice Hall Hispanoamericana, S.A., U.S.A., 1996.
- ☛ Ed Krol, The Whole Internet Users Guide and Catalog, O'Reilly & Associates Inc., U.S.A., 1996.
- ☛ Hwang Ka., Briggs Fayé, Arquitectura de Computadoras y Procesamiento Paralelo, Mc. Graw Hill, México, 1994.
- ☛ Stephen Coffin, Unix System V Release 4: The Complete Reference, Mc. Graw Hill, U.S.A., 1995.
- ☛ Kochan and Wood, Exploring the Unix System, Sams Publishing, U.S.A., 1995.
- ☛ Tanenbaum Andrew, Computer Networks, Prentice Hall Hispanoamericana, S.A., U.S.A., 1994.
- ☛ Black Uyless, Computer Networks, Protocols, Standards and Interfaces, Prentice Hall Hispanoamericana, S.A., U.S.A., 1992.
- ☛ Eckel George, Building a Unix Internet Server, New Riders Publishing, U.S.A., 1996.
- ☛ Sidnie Feit, TCP/IP Architecture, Protocols and Implementation, Mc. Graw Hill, U.S.A., 1996.
- ☛ Eileen Frisch, Essential System Administration, O'Reilly & Associates, Inc., U.S.A., 1996.