



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE INFORMACION DE LOS DAÑOS
AMBIENTALES PRODUCIDOS POR LAS
ACTIVIDADES PETROLERAS EN LA REGION SUR
DEL ESTADO DE TABASCO.

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N
REBECA CHAVEZ GALINDO
ALBA I. MARQUEZ VELASCO
MA. ELIZABETH MARTINEZ GONZALEZ

DIRECTOR DE TESIS: ING. GABRIEL CASTILLO HERNANDEZ.

MEXICO, D. F.

1998.



TESIS CON
FALLA DE ORIGEN

267418



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatoria:

Dedico esta tesis a mis padres David Chávez Fragoso y Araceli Galindo Cabrera, que con su cariño y apoyo incondicional me indicaron el camino del estudio, logrando llegar hasta éste momento juntos.

A mi hermana Nydia por su apoyo y comprensión en los momentos en que necesite de ellos.

A mi hermano David por ser el motivo de mi constante superación, con el deseo de inspirar en ti el ser el mejor en todo lo que te propongas.

Agradecimientos:

Al Ing. Gabriel Castillo Hernández por su amistad y apoyo al aceptar dirigirnos en esta tesis.

A mis maestros, compañeros y amigos que a lo largo de mi vida me han brindado sus conocimientos y sincera amistad.

Rebeca Chávez Galindo

Agradecimientos

Es difícil poder agradecerle a todas las personas que de alguna manera me han apoyado con sus consejos, paciencia, tolerancia...

A mis padres porque gracias a ellos estoy aquí.

A mis hermanos, porque siempre estuvieron dispuestos a ayudarme en los momentos difíciles.

Ing. Castillo... qué más se le puede decir a una persona como usted? GRACIAS no es suficiente por todo el tiempo que nos aguantó, la dedicación que nos prestó.

David, Perla, Antonio, Estelia, Rebeca y demás... gracias por todo el cariño demostrado a través de estos años, por estar a mi lado para escucharme, soportar los momentos de "locura"... en fin, gracias por compartir conmigo una de las mejores cosas de la vida: La amistad.

Alba I. Marquez Velasco

Agradecimientos:

A Dios y a ME por estar junto a mí en todo momento.

A Silvia mi mami querida y a mis hermanos Paty y José Luis que me apoyaron y dieron lo mejor de sí para ayudarme a lograr dar un paso más.

Al Ing. Gabriel Castillo por su GRAN paciencia y consejos... Gracias

A Germán que me apoyó y dió ánimos para no claudicar

A la Facultad de Ingeniería por abrirme sus puertas como una segunda casa.

A todos aquellos que de manera directa o indirecta contribuyeron a finalizar esta etapa.

GRACIAS

María Elizabeth Martínez González

Indice

Capítulo 1		Introducción	
		Objetivo General	1
		Objetivos Específicos	2
		Desarrollo	2
Capítulo 2		Internet	
2.1		Antecedentes	4
2.1.1		Definición de Red	4
2.1.2		Breviario de Arquitectura Cliente/Servidor	5
2.1.3		Definición de Internet	5
2.1.4		Historia y Evolución	7
2.2		Organización	9
2.2.1		Direcciones	9
2.2.2		Dominios	11
2.2.3		Protocolos	13
2.3		Aplicaciones/Servicios	15
2.3.1		Correo Electrónico	15
2.3.2		Transferencia de Archivos	15
2.3.3		Sesiones Remotas	16
2.3.4		GOPHER	16
Capítulo 3		WWW	
3.1		Antecedentes	18
3.1.1		Definición	18
3.1.2		Historia y Evolución	18
3.2		Navegadores	19
3.2.1		Definición	19
3.2.2		Historia y Evolución	19
3.3		Servidores	22
3.3.1		Definición	22
3.3.2		Historia y Evolución	22
3.3.3		Interacción con el navegador	24
3.4		Seguridad	27
3.4.1		Generalidades	27
3.4.2		Encriptamiento	27
3.4.3		Autenticación y Certificados	28

Capítulo 4 HTML		
4.1	Antecedentes	32
4.1.1	Ventajas de HTML	33
4.1.2	Requisitos	33
4.1.3	Estructura de un documento escrito con HTML	34
4.2	Elementos	35
4.2.1	Descripción de las etiquetas HTML	38
4.3	Listas	43
4.3.1	Menú Interactivo	44
4.3.2	Lista de Directorio	45
4.4	Mapas de Imágenes	45
4.5	Formas HTML	45
4.5.1	Elementos de una forma	46
4.5.2	Forma <FORM>	47
4.6	Imágenes	51
4.7	Tablas <TABLE>	53
4.8	Editores y Convertidores	55
4.8.1	Editores	56
4.8.2	Convertidores	58
4.9	CGI (Common Gateway Interface)	58
4.9.1	Requerimientos	59
4.9.2	Procesamiento de Formularios (Formas)	61
4.9.3	Datos de Entrada al programa CGI	61
Capítulo 5 Proyecto Tabasco		
5.1	Elección de Herramientas	63
5.1.1	Servidor de WWW	63
5.1.2	Creación de Grupos de Usuarios	64
5.1.3	Operación del Servidor	65
5.1.4	Diseño de las páginas de acceso	67
5.1.5	Mecanismos de Seguridad	70
5.1.6	Empleo de la plantilla ANT_HTML.DOT	71
5.2	SOMOD	73
Comentarios Finales		81
Bibliografía		82
Glosario		84
Apéndice A		88
Apéndice B		99

Capítulo 1

Introducción

El efectuar un diagnóstico ambiental en las instalaciones de PEMEX en el estado de Tabasco, con el propósito de evaluar impactos ambientales asociados a emisiones contaminantes líquidas, sólidas, gaseosas sobre los sistemas naturales y los inducidos por el hombre, nos generará una cantidad considerable de información y con ello la necesidad de establecer una comunicación permanente y rápida entre los diversos grupos participantes del proyecto, por lo anterior se desarrolló un sistema computarizado que proporciona un control y almacenamiento de los datos generados por los grupos de estudio, así como la implementación de mecanismos de seguridad, para evitar la ruptura de su confidencialidad y por ende el uso inapropiado de los resultados y sus documentos, además de proporcionar consultas temáticas a los funcionarios de las instituciones participantes (PEMEX, IMP, UNAM) de manera fácil y segura.

Objetivo general

Desarrollar un sistema que permita almacenar, procesar y comunicar los datos y análisis, a los grupos participantes en los diferentes trabajos involucrados para ayudar al desarrollo del proyecto Tabasco, cumpliendo con los siguientes requisitos:

- **Seguridad.** El sistema desarrollado deberá de contar con los esquemas necesarios para garantizar la confidencialidad de la información, evitando en todo momento que la información pueda ser vista o utilizada por personas no autorizadas.
- **Oportunidad de la información.** La información recibida deberá ser puesta a disposición de los demás grupos de la manera más rápida posible. Esto significa que la información deberá ser incorporada al SIPT¹ pocas horas después de que ha sido recibida
- **Flexibilidad.** Debido a la gran variedad de la información manejada (fotos, imágenes, gráficas, tablas, etc.) el sistema deberá permitir la rápida incorporación de ésta, para su posterior consulta
- **Complejidad en la administración del sistema.** La complejidad en la administración del sistema no debe ser excesivamente alta. Tabasco deberá de tener un sistema similar (espejo) al que se maneje en el D.F., o al menos el acceso desde Tabasco al que se encuentre instalado en la ciudad de México

¹SIPT (Sistema de Información del Proyecto Tabasco), se utilizará como abreviatura del Sistema de Información de los Daños Ambientales Producidos por las Actividades Petroleras en la Región Sur del Estado de Tabasco

- **Tiempo de capacitación.** Tanto la capacitación requerida para el desarrollo de aplicaciones, como la operación del sistema como administrador y como usuario deberá ser mínima

Objetivos Específicos

- Diseño de una base de datos que permita almacenar y actualizar la información generada por los grupos de estudio
- Consulta y transferencia de la información generada por los grupos de estudio
- Acceso al sistema en diferentes niveles de consulta
- Recepción electrónica de la información de los grupos generadores
- Ubicación dentro del sistema de las normas, procedimientos, memorias y metodología en el sistema
- Mecanismos de seguridad y confidencialidad para el manejo de información

Desarrollo

El proyecto contempla la instalación de tres Sistemas de Información idénticos: Uno en Tabasco y dos en la Ciudad de México (uno en el IMP² y otro en la UNAM), como se muestra en la figura 1-1.

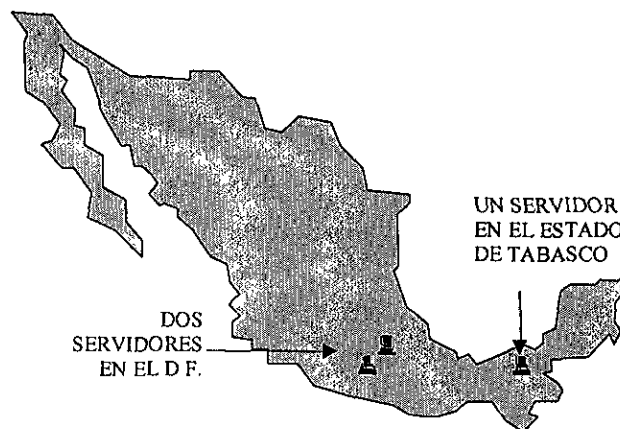


Figura 1-1

Los servidores con sede en el IMP y Tabasco son administrados por el IMP, el servidor localizado en la UNAM es administrado por el Instituto de Ingeniería y este es el tema de nuestra tesis.

La gran cantidad de información que genera el Proyecto Tabasco requiere un control adecuado de la misma, para permitir la consulta de los diversos grupos, su administración, almacenamiento y mantenimiento. Todo esto obligó a buscar una herramienta adecuada para administrar y proteger la información recibida, considerando siempre el cumplimiento de los objetivos propuestos para el Sistema de Información del Proyecto Tabasco.

² Instituto Mexicano del Petróleo

El tránsito de la información por el sistema de comunicaciones deberá de garantizar la seguridad tanto de los mensajes (que los mensajes enviados y recibidos sean los mismos) como el canal de comunicación (que no sea posible que alguien ajeno al sistema tenga el acceso a la información que se está enviando).

Para entender el desarrollo de este proyecto, iniciaremos en el capítulo 2 con algunas definiciones básicas de Internet, de igual forma veremos cómo está organizada y algunos servicios que presta a los usuarios.

Posteriormente en el capítulo 3 se definirá el WWW, los navegadores, servidores y algunos protocolos de seguridad.

En el capítulo 4 se explican algunas etiquetas de HTML, así como la creación de formas o formularios, además del funcionamiento de los programas denominados CGI's.

Finalmente, en el capítulo 5 se mostrarán las herramientas empleadas en el SIPT así como los resultados obtenidos.

Capítulo 2

Internet

2.1 Antecedentes

2.1.1 Definición de Red

Es una conexión de varias computadoras a través de un cableado especial, con el fin de compartir datos y recursos como impresoras, modems, discos de almacenamiento entre las computadoras conectados a la red, etc. Si la red está integrada por computadoras ubicadas en la misma área, se llama *red de área local*, en general estas computadoras están localizadas en el mismo departamento, con el mismo grupo de trabajo o en la misma compañía, como lo muestra el ejemplo de la figura 2-1.

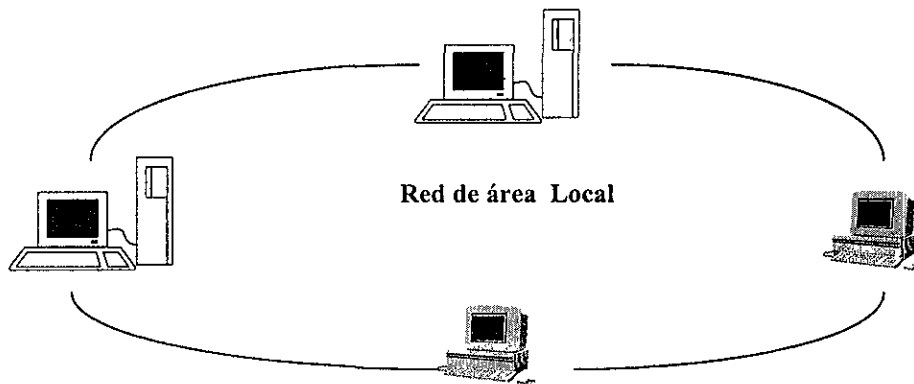


Figura 2-1

Dos *redes aisladas* no se comunican ni pueden intercambiar información entre sí. Una red computacional de comunicación requiere de los siguientes elementos:

1. **Hardware de conectividad.** Dispositivos físicos que se instalan en la computadora para que éstas estén preparadas para establecer comunicación con otros equipos, por ejemplo, tarjetas de red, conectores, etc.
2. **Software de conectividad.** Programas que se instalan en la computadora los cuales están diseñados para la ejecución de los procesos de comunicación
3. **Dispositivos de comunicación o enlace.** Medios físicos necesarios para lograr la conectividad entre las diferentes partes involucradas en la red, estos pueden ser cableado, microondas, satélites, etc.

2.1.2 Breviario de arquitectura Cliente / Servidor

Los términos Cliente y Servidor¹ son empleados en cualquier información que haga referencia a Internet; por eso, es importante definir algunos conceptos para poder comprenderla mejor.

El servidor, es el encargado de proporcionar un recurso en particular, mientras el cliente es el responsable de emplear ese recurso. En Internet, normalmente el hardware no se "ve", y el término "cliente" y "servidor" hacen referencia a los programas que solicitan y proporcionan los servicios.

Muchos nodos de Internet proporcionan un servicio llamado GOPHER, el cual permite seleccionar opciones de una serie de menús. Cada vez que se selecciona una opción, GOPHER ejecuta la tarea indicada. Por ejemplo, si la opción del menú describe una determinada información (como "noticias del día"), GOPHER recuperará esta información y la mostrará en pantalla.

Cuando se emplea GOPHER, dos programas interactúan entre sí. Primero, hay un programa que proporciona una interfaz de usuario, o sea, el programa que interpreta las teclas que se pulsan, muestra los menús y asegura que las peticiones se llevan a cabo. Este programa se denomina cliente GOPHER.

El otro programa necesario es el que suministra cualquier cosa que el cliente GOPHER solicita. Este programa se denomina servidor GOPHER.

Una ventaja de este sistema es que los programas cliente y servidor no deben ejecutarse necesariamente en la misma computadora. Al contrario, es más usual, que los programas cliente y servidor se encuentren en diferentes computadoras.

Todos los servicios de Internet hacen uso de esta relación Cliente / Servidor. Navegar por Internet, implica aprender a usar cada uno de los programas cliente disponibles. Por esta razón, para utilizar un servicio Internet, hay que entender:

- Cómo ejecutar un programa cliente para ese servicio
- Cómo decirle al programa cliente qué servidor se quiere utilizar
- Qué instrucciones se pueden utilizar con cada tipo de cliente

2.1.3 Definición de Internet

Es el resultado de comunicar miles de redes de computadoras entre sí, utilizando para este propósito, el protocolo TCP/IP, que identifica los datos aunque procedan de diferentes equipos (PC's, Macintosh, Amiga, etc.) y diferentes sistemas operativos (por ejemplo MS-DOS y UNIX).

¹ Se le conoce también con el nombre de host, server

En la figura 2-2 (de 1997) se muestra el gran impacto que ha tenido Internet:

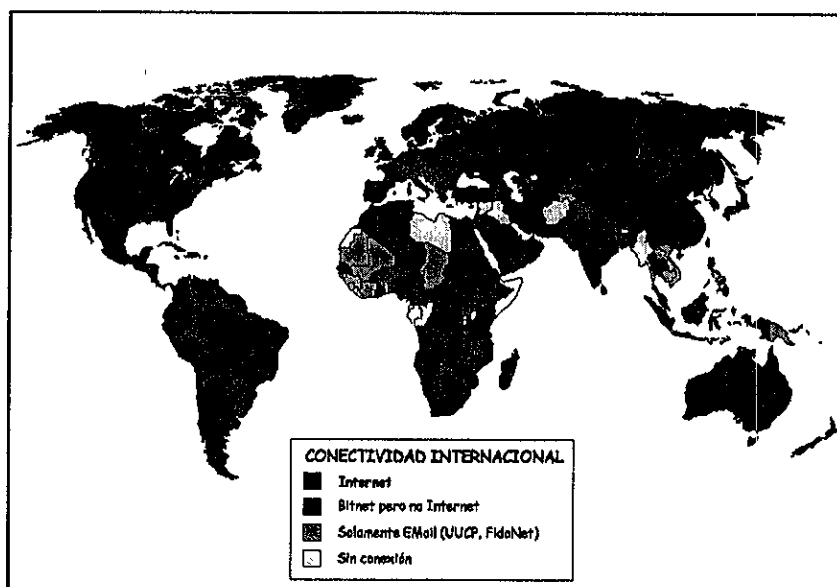


Figura 2-2
(Fuente: FTP.CS.WISC.EDU)

El crecimiento que ha tenido Internet a nivel mundial lo podemos observar en la tabla 2-1

Número de usuarios de Internet a nivel mundial:

Año	Número
1992	600,000
1993	1,200,000
1994	2,400,000
1995	4,800,000
1996	8,600,000
1997	16,400,000
1998	30,000,000*
1999	60,000,000*

Tabla 2-1

*Estimado (Fuente: Yahoo Yellow Pages)

2.1.4 Historia y Evolución

La red Internet tiene una historia muy corta. Fue propuesta por ARPA (*Advanced Research Projects Agency*) y se desarrolló a partir de un proyecto impulsado a principios de los años 70's por el Departamento de Defensa de los Estados Unidos. Entre los objetivos de este proyecto se encontraban:

- Realizar un sistema de comunicación que se pudiera utilizar para manejar documentos secretos
- La creación de una red informática que pudiera seguir funcionando en caso de un desastre, como por ejemplo una guerra nuclear
- Diseñarla de tal forma que si parte de la red era dañada o destruida, el resto del sistema debía de seguir en funcionamiento

Esta red fue llamada ARPANET, y puso por primera vez en contacto a los investigadores científicos y académicos estadounidenses. Con el paso de los años, ARPANET fue abriendo sus estándares de comunicación a las universidades, centros de investigación y dependencias de gobierno, así como a instituciones extranjeras, convirtiéndose en una red llamada Internet.

En 1985, la NSF (*National Science Foundation*) creó NSFNET, la cual es una serie de redes informáticas dedicadas a la difusión de los nuevos descubrimientos y la educación. Basada en los protocolos de comunicación de ARPANET, la NSFNET creó un esqueleto de red o "backbone" nacional, ofrecido gratuitamente a cualquier institución americana de investigación o educación. Al mismo tiempo, otras redes regionales fueron apareciendo con el fin de poder enlazar el tráfico electrónico de instituciones individuales con el esqueleto de red nacional. Geográficamente la red estaba compuesta por los puntos mostrados en la figura 2-3:

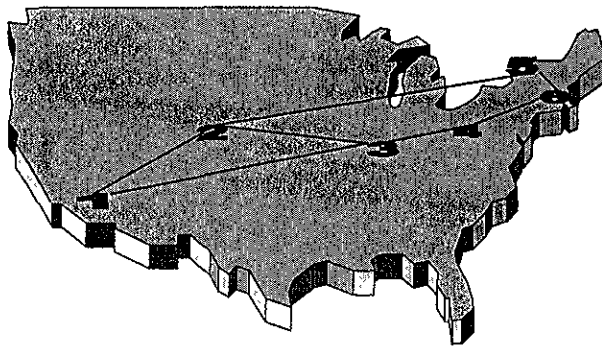


Figura 2-3

Las localidades estaban en:

- (1) San Diego CA, (2) Boulder CO, (3) Champaign IL
(4) Pittsburg PA, (5) Ithaca NY, (6) Princeton NJ

En ese entonces, Internet era de uso exclusivo de los funcionarios de las pocas empresas con gran cantidad de recursos y tecnología y de la comunidad universitaria directamente enlazadas a la red. Esto significaba, que la mayoría de las personas que no tenían acceso a estas instituciones de gran infraestructura, en ningún momento iban a conocer la existencia de la red. Aunándole que el uso de este servicio era realmente complicado y sólo gente con altos conocimientos computacionales se atrevía a poner en práctica su destreza en el ramo.

La NSFNET creció rápidamente a la par con el descubrimiento por parte del público de su potencial y con la creación de nuevas aplicaciones que permitían un acceso más fácil. Corporaciones como Sprint y MCI empezaron a construir sus propias redes, que enlazaron con NSFNET.

NSF también coordinó en un principio un servicio llamado InterNIC (*Centro de Información de la Red Internet*), que registraba todas y cada una de las direcciones de Internet para que los datos pudieran ser dirigidos al sistema correcto. De este servicio se han hecho ahora cargo AT&T y Network Solutions Inc., en cooperación con NSF.

Cada nodo de la red, recibió una identificación numérica, conocida como dirección IP, lo cual permitía que las computadoras y los dispositivos se diferenciaron.

El número de máquinas conectadas se ha duplicado cada año desde 1988, y se espera que su mayor expansión se dará todavía en un futuro cercano.

Dos de los factores más importantes que han influido directamente para el gran crecimiento de Internet en los últimos 5 años son:

1. **El surgimiento de los proveedores de acceso a Internet.** Son empresas que permiten a sus clientes utilizar su infraestructura de Internet. Este servicio es generalmente cobrado basándose en el tiempo de uso y se hace a través de la línea telefónica. A diferencia de una conexión directa a Internet, este tipo de enlace es de menor velocidad, generalmente limitada por el ancho de banda de la línea telefónica. Para poder usar este tipo de servicio se requiere de un módem. Las principales ventajas de esta clase de acceso son:
 - **Facilidad de conexión.** Porque provee todo el software necesario para realizarla de una manera sencilla
 - **Requerimientos mínimos de hardware y software.** Es decir, con computadoras de mediana capacidad y software de fácil instalación es suficiente
 - **Costo / Beneficio.** Este es uno de los principales atractivos de usar Internet, ya que por una razonable cuota se obtienen grandes beneficios como son las consultas a importantes bases de datos, contacto rápido con clientes y proveedores, entre muchos otros

En el mundo existen muchas redes de computadoras, no sólo Internet (ésta puede ser la más grande, pero definitivamente no es la única). Hay servicios tales como CompuServe, Prodigy, Genie, America On Line y otros que tienen miles de usuarios

2. **El esquema del World Wide Web o WWW² como normalmente es conocido**

² Este tema se estudia más detalladamente en el capítulo 3

2.2 Organización

2.2.1 Direcciones

Todos los dispositivos que están conectadas a Internet requieren de una dirección IP (*Internet Protocol*) para poder hacer uso de la red. El número IP o dirección IP es empleado principalmente, para distinguir de manera única a una computadora en Internet (excepto gateways³ que comúnmente cuentan con más de una dirección IP).

Las direcciones IP están formadas por 32 bits que normalmente se escriben como cuatro números enteros entre 0 y 255 separados por puntos (132.248.156. 186) , representación conocida como “notación de punto decimal”. Las direcciones están divididas en dos partes:

- Bits de Red
- Bits de Host

Se emplean cuatro clases para las direcciones IP, de acuerdo al tamaño de la red, estas clases son: Clase A, Clase B, Clase C y Clase D, como se aprecia en la figura 2-4. La clase se puede determinar desde los dos primeros bits (de orden superior), debido a la poca existencia de redes de Clase D.

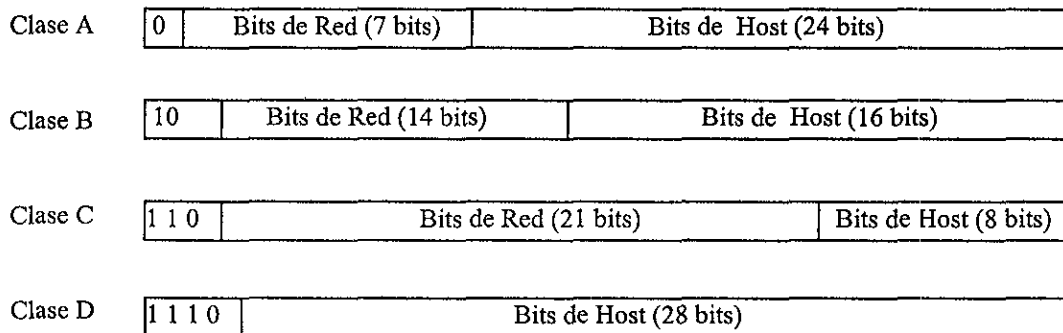


Figura 2-4

- **Clase A.** Las direcciones de clase A corresponden a redes grandes que tienen muchas máquinas y pueden direccionar hasta 16,777,214 máquinas cada una. Las direcciones de red de clase A tienen el primer bit a 0.

Ejemplo:

00001010	00001111	00010000	00001011
10	15	16	11

10.15.16.11 (Notación decimal)

- **Clase B.** Las direcciones de red de clase B son para redes intermedias, permitiendo direccionar 65,534 máquinas cada una. Los primeros bits de una dirección de red de clase B son 10.

Ejemplo:

³ También conocido como ruteadores o routers

10000001 00001010 00000010 00000011
 129 10 2 3

129.10.2.3 (Notación decimal)

- **Clase C.** Las direcciones de clase C permiten direccionar 256 dispositivos, empiezan con los bits 110.

Ejemplo:

11001010 00001111 00010111 00001011
 202 15 23 11

202.15.23.11 (Notación decimal)

- **Clase D.** Las direcciones de clase D son un grupo especial que se utiliza para dirigirse a grupos de máquinas. Estas direcciones son muy poco utilizadas. Los primeros bits de una dirección de clase D son 1110.

Ejemplo:

1110 Dirección múltiple

En la tabla 2-2 podemos ver los rangos que abarca cada una de las clases:

Clase	Rango	No. Redes	No. Máquinas /Red
A	001.xxx.xxx.xxx - 126.xxx.xxx.xxx	126	16,777,214
B	128.001.xxx.xxx - 191.254.xxx.xxx	16,382	65534
C	192.000.001.xxx - 223.255.254.xxx	2,097,152	254
D	224.000.000.000-239.255.255.255.254	-	-

Tabla 2-2

Algunas direcciones están reservadas para propósitos especiales. Estas son:

- **Direcciones de red reservadas**
 - 000.xxx.xxx.xxx (1)
 - 127.xxx.xxx.xxx (reservada como la propia máquina)
 - 128.000.xxx.xxx (1)
 - 191.255.xxx.xxx (2)
 - 192.000.000.xxx (1)
 - 223.255.255.xxx (2)
- **Direcciones de máquinas reservadas**
 - xxx.000.000.000 (1)
 - xxx.255.255.255 (2)
 - xxx.xxx.000.000 (1)
 - xxx.xxx.255.255 (2)
 - xxx.xxx.xxx.000 (1)
 - xxx.xxx.xxx.255 (2)

- (1) Se utilizan para identificar a la red propia
- (2) Se usa para enmascarar

Como los nombres son más fáciles de recordar y usar que las direcciones IP, existe un mecanismo que asigna un nombre a cada máquina. El mecanismo que realiza esta transformación es DNS (*Domain Name System* o Sistema de Nombres por Dominio), capaz de pasar de un nombre a una dirección IP y viceversa. Por ejemplo:

132.248.156. 186 puede estar registrado en el DNS como: *mtf186.iingen.unam.mx*

En este caso *mtf186.iingen.unam.mx* se le conoce con el nombre de dominio.

2.2.2 Dominios

En una computadora multiusuario cada usuario debe disponer de una cuenta para poder trabajar en ella. Se denomina cuenta al espacio que el usuario tiene disponible en una computadora y al derecho a utilizar sus recursos. Las cuentas se identifican por un nombre que se le denomina nombre de usuario o identificación del usuario.

La identificación completa de un usuario en una computadora se suele escribir anteponiendo el nombre de la cuenta, al nombre de dominio del anfitrión, separados por el símbolo @. El formato es el siguiente:

id_usuario@dominio

Esta forma de identificación de un usuario a menudo coincide con su dirección de correo electrónico. Si tenemos la siguiente dirección de correo electrónico:

velmar@hotmail.com

La primera parte (*velmar*) se le conoce como el identificador de usuario y *hotmail.com* es el dominio. Cada parte de dominio se denomina subdominio.

Como se puede observar, los subdominios están separados por puntos. En nuestro ejemplo, hay dos subdominios, *hotmail* y *com*. La forma de entender el nombre de un dominio es analizar los subdominios de derecha a izquierda. El nombre está construido de manera que cada subdominio determina algo sobre la computadora. El subdominio situado más a la derecha, llamado subdominio de primer nivel, es el más general. Los subdominios que se encuentran hacia la izquierda proporcionan información más específicos. En nuestro ejemplo, el dominio de primer nivel *com* nos indica que la computadora pertenece a una empresa comercial. El siguiente subdominio, *hotmail*, por estar más a la izquierda, nos indica el nombre específico de la computadora.

Sin embargo, es posible encontrar algunas variaciones. El ejemplo que se ha manejado hasta ahora, tiene dos subdominios (que es el mínimo de subdominios que se emplean), pero en muchas ocasiones se encontrarán direcciones que tienen más subdominios para que la dirección sea más específica. Por ejemplo:

mva@cronos.fi-b.unam.mx

En este caso, el identificador de usuario es *mva*. El dominio *cronos* se refiere a una computadora llamada cronos, *fi-b* nos indica que forma parte de la Facultad de Ingeniería, la cual pertenece a la UNAM (*unam*), y se encuentra en México (*mx*).

Dominios de primer nivel

En general hay dos tipos de dominios de primer nivel, los cuales se encuentran controlados por la InterNIC:

- **Dominios de organizaciones.** Los dominios de organizaciones están basados en un esquema de direcciones que se desarrolló antes de que aparecieran las redes internacionales. Fue creado principalmente para utilizarse dentro de los Estados Unidos. La idea era que el dominio de primer nivel debería indicar el tipo de organización que era responsable de la computadora.

Los dominios de organizaciones son 7, como se aprecia en la tabla 2-3:

Dominio	Utilización
com	Para organizaciones comerciales (negocios)
edu	Organizaciones educativas (universidades, escuelas secundarias, etc.)
gov	Organizaciones gubernamentales sin incluir a la milicia
mil	La milicia (el ejército, la marina, etc.)
org	Otras organizaciones
net	Recursos de la Red
Arpa	Una identificación ARPANET - Internet

Tabla 2-3

- **Dominios geográficos.** Una vez que Internet se extendió internacionalmente, se hizo necesario crear nuevos dominios de primer nivel que fueran más específicos. Para enfrentarse a esta necesidad, se desarrolló un nuevo sistema de dominios geográficos, en el que una abreviación de dos letras representaría un país entero. Algunos ejemplos de dominios geográficos se encuentran en la tabla 2-4:

Dominio	País
mx	México
pe	Perú
jp	Japón
de	Alemania
au	Australia
ca	Canadá
ch	Suiza
nz	Nueva Zelanda
pl	Polonia
dk	Dinamarca
ru	Federación Rusa
us	Estados Unidos
va	Vaticano
it	Italia
il	Israel
fr	Francia
tr	Turquía
es	España

Tabla 2-4

2.2.3 Protocolos

Un protocolo es un conjunto de convenciones que determinan cómo se realiza el intercambio de datos entre dos computadoras o programas. El conjunto de protocolos usados por todas las redes que forman parte de Internet se denomina de forma abreviada TCP/IP.

TCP/IP es un conjunto de protocolos, los cuales proporcionan las reglas para la comunicación. Contienen los detalles referentes a los formatos de los mensajes, describen cómo responde una computadora cuando llega un mensaje y especifican de qué manera una computadora maneja los errores.

El nombre TCP (*Transfer Control Protocol*) significa Protocolo de Control de Transmisión, mientras que IP (*Internet Protocol*) significa Protocolo de Internet. El IP se encarga de la transmisión de datos, mientras que TCP asegura que la transmisión funcione correctamente.

Por ejemplo supongamos que enviamos un mensaje de correo electrónico muy extenso, TCP dividirá este mensaje en paquetes. Cada paquete se marca con un número de secuencia y con la dirección del destinatario, además inserta información para controlar los 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 no hay errores. Si existe un error TCP pide que el paquete en cuestión le sea reenviado. En sí, TCP/IP se encarga de buscar la mejor ruta y asegurarse de que la información llegue en buen estado.

Modelo de capas

En la actualidad, las funciones propias de una red de computadoras pueden ser divididas en las siete capas (ver la tabla 2-5) propuestas por ISO (*International Organization for Standardization*) para su modelo de sistemas abiertos OSI (*Open Systems Interconnection*). Sin embargo la implantación real de una arquitectura puede diferir de este modelo. Las arquitecturas basadas en TCP/IP proponen cuatro capas como se observa en la tabla 2-6:

Modelo OSI
7 Aplicación
6 Presentación
5 Sesión
4 Transporte
3 Red
2 Enlace
1 Física

Tabla 2-5

Arquitectura TCP/IP
4 Aplicación
3 Transporte
2 Internet
1 Interfaz de red

Tabla 2-6

Las capas de Sesión y Presentación son responsabilidad de la capa de Aplicación y las capas de Enlace y Física son vistas como la capa de Interfaz de Red. Por tal motivo para TCP/IP sólo existen las capas Interfaz de Red, Internet, Transporte y Aplicación.

La arquitectura en capas de los protocolos está diseñada como una pila en la que los protocolos de más alto nivel interactúan con protocolos de niveles más bajos. A continuación se dará una breve explicación de cada una de las capas que conforman al protocolo TCP/IP:

- **Capa de Aplicación.** Es la capa más alta de la pila; ésta provee servicios de alto nivel a los usuarios como transferencia de archivos, entrega de correo electrónico, y acceso a terminales remotas. Los programas de aplicación escogen entre diferentes protocolos de transporte dependiendo del tipo de servicio de transporte que requieran.
- **Capa de Transporte.** Provee comunicación punto a punto entre las aplicaciones. Regula el flujo de información. Puede proveer un transporte confiable asegurándose que los datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente de tal manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota esto lo hace añadiendo identificadores de cada una de las aplicaciones. Realiza además una verificación por suma, para asegurar que la información no sufrió alteraciones durante su transmisión.
- **Capa Internet.** Controla la comunicación entre un equipo y otro, decide qué rutas deben seguir los paquetes de información para alcanzar su destino. Conformar los paquetes IP que serán enviados por la capa inferior. Desencapsula los paquetes recibidos pasando a la capa superior la información dirigida a una aplicación. La integridad de los datos no se verifica en este nivel, por lo que el mecanismo de verificación es implementado en capas superiores (Transporte o Aplicación).
- **Capa de Interfaz de Red.** La capa de interfaz de red acepta datagramas de la capa de Internet y los envía físicamente. Consiste en los manejadores de los dispositivos (device driver) que se conectan al medio de transmisión para una pieza particular de hardware, y la interfaz de red puede consistir de múltiples módulos.

Para que la información fluya a través de las capas, ésta pasa por un proceso de encapsulamiento. Los mensajes o información recibida por la capa de TCP es encapsulada con un encabezado de TCP en un paquete llamado "Segmento de TCP", este segmento de TCP es entregado a la capa de IP, en el que se le agrega un encabezado de IP y el paquete llamado "Datagrama de IP" es creado. El paso final incluye el encapsulamiento del datagrama de IP en paquetes creados para la capa de acceso al medio (ver figura 2-5)

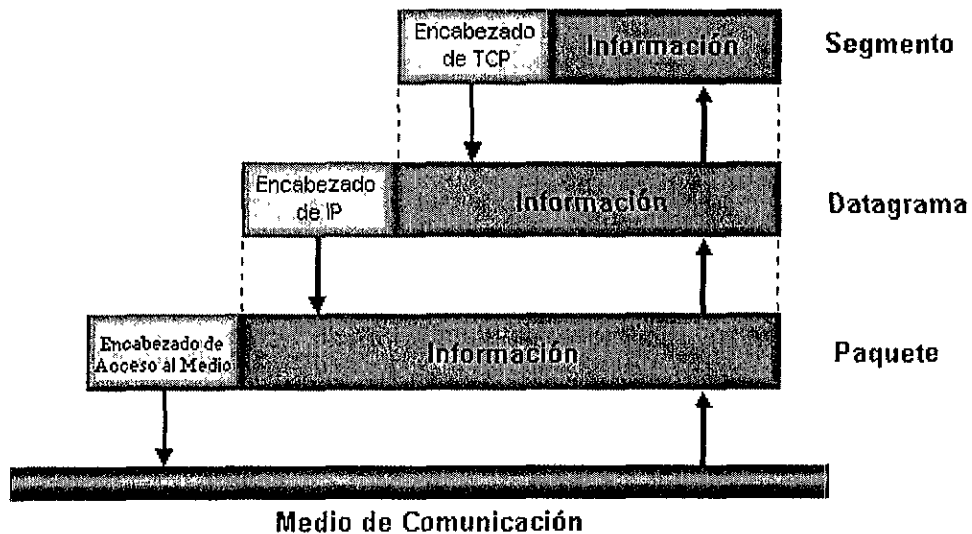


Figura 2-5

2.3 Aplicaciones/Servicios

2.3.1 Correo Electrónico

El servicio básico de cualquier tipo de red es el correo electrónico, el cual se diferencia de las otras aplicaciones porque no es un servicio de usuario a usuario: no es necesario que las máquinas emisora y receptora del correo electrónico se comuniquen entre sí. Al correo electrónico se le conoce como un servicio de "almacenaje y reenvío".

El correo pasa de una máquina a otra hasta que llega a su destino final. Si el emisor y el receptor no están conectados en la misma red, es necesario colocar el mensaje en algún lugar en donde se concentre todo el correo que vaya a una determinada red.

Un mensaje de correo electrónico puede contener cualquier tipo de información que en cuestión de segundos puede llegar a su destinatario en cualquier parte del mundo, ya que la infraestructura de Internet transporta el mensaje a través de medios digitales tan diversos como la fibra óptica, redes de alta velocidad, microondas, etc.

Cuando el usuario desea revisar su correo lo que hace es consultar desde su computadora al servidor de correo y de esta manera podrá leer sus mensajes. Esto significa que los mensajes no llegan directamente a la computadora del usuario, sino al servidor de correo del proveedor. Una vez que el usuario lee sus mensajes, éstos permanecerán almacenados hasta que el usuario decida eliminarlos de su cuenta o almacenarlos en algún medio.

2.3.2 Transferencia de archivos

Permite realizar diversas tareas tales como:

- Conectarse con equipos remotos
- Llevar a cabo operaciones limitadas sobre los directorios dentro de equipos remotos
- Transferir archivos de un equipo remoto a un local

Como se indica, su función es mover archivos de una computadora a otra. No importa donde se localicen estas dos computadoras, cómo estén conectadas o si tienen o no el mismo sistema operativo. Dado que ambas computadoras "hablan" el protocolo FTP y tienen acceso a Internet, es posible utilizar el comando FTP para transferir archivos a través de un servidor. Algunas de las características de su uso cambian con cada sistema operativo, pero la estructura básica de comandos es la misma en cualquier máquina.

Existen dos maneras de realizar FTP's a sistemas remotos, una es conectándose a servidores de FTP privados y otra a servidores de FTP anónimos. Para poder conectarse a un servidor de FTP privado, el usuario necesita tener una cuenta de acceso y una contraseña las cuales son asignadas por el administrador del servidor del sistema remoto al cual se desea conectar.

Los servidores de FTP anónimos permiten el acceso a bases de datos públicas sin necesidad de obtener una cuenta. La mayor parte de los directorios públicos son consultados vía FTP anónimo.

Para poder hacer una conexión a un servidor de FTP es necesario tener en el equipo del sistema local un cliente de FTP, este puede ser incluido en el navegador⁴ que esté utilizando (el Netscape es uno de los navegadores que soporta el protocolo de FTP), así como conocer la dirección del servidor de FTP al cual se desee conectar (sistema remoto) y por último si el servidor al cual se desea conectar no es un servidor de FTP anónimo es necesario tener una cuenta y contraseña de acceso.

2.3.3 Sesiones Remotas

Hace diez años, una de las formas más comunes para obtener información de otras computadoras en Internet era enlazarse a ellas y revisar sus archivos. El método más común para hacerlo era con el comando TELNET. El comando TELNET debe ser empleado por las dos computadoras para que funcione el programa TELNET. Algunos de los muchos usos en Internet son:

- Cientos de catálogos de bibliotecas están disponibles sólo a través de un enlace directo a las computadoras de cada biblioteca
- Si se tienen varias cuentas en más de una computadora en Internet, puede enlazarse a la más cercana y utilizar TELNET para tener acceso a las otras
- Los investigadores de todo el país que colaboren en un proyecto pueden entrar a una sola computadora para intercambiar información

TELNET utiliza Internet para conectarse a la computadora especificada. Una vez que se hace la conexión, TELNET actúa como un intermediario entre nosotros y la computadora remota. Cada vez que la computadora remota produce una respuesta, aparece en la pantalla de nuestra computadora. El resultado final es que nuestro teclado y nuestra pantalla parecen estar conectados directamente a la otra computadora.

2.3.4 GOPHER

GOPHER es un potente sistema que permite consultar muchos de los recursos de Internet de una forma simple y consistente. Para usar GOPHER, todo lo que se necesita es seleccionar en un menú.

Cada vez que se hace una selección, GOPHER realiza lo necesario para llevar a cabo alguna petición. La potencia de GOPHER viene de que los recursos enumerados en un menú pueden estar en cualquier parte de Internet. Cuando se selecciona un elemento, GOPHER lo traerá o hará lo que sea necesario para atender nuestra petición. La mayor parte del tiempo, GOPHER tendrá que conectarse a otra computadora, pero todo será transparente para el usuario, lo que notaremos es que nuestra petición ha sido cumplida simple y fácilmente.

La ventaja que ofrece GOPHER consiste en permitir observar a través de los recursos de Internet, sin importar su tipo.

GOPHER comenzó a funcionar como un servicio de distribución interna en el campus de la Universidad de Minnesota, hogar de los "Golden GOPHERs".

⁴ Ver capítulo 3, en donde se habla más de este tema

El servicio fue diseñado para mantener un control sobre su propio servidor y sus propios datos. Para lograr esto, organizaron el sistema por temas, de manera que fuera posible verlo como un enorme base de datos, en lugar de cientos de bases de datos pequeñas interconectadas. Se puede tener acceso a archivos FTP, números telefónicos de servidores "white pages", catálogos de bibliotecas y otros servidores de bases de datos con propósitos especiales (basados en TELNET). Pero sólo GOPHER sabe dónde se localizan realmente los datos, la forma de tener acceso a ellos y que existen muchos servidores que los proporcionan.

No tomó mucho tiempo darse cuenta de que el sistema podía funcionar para servidores colocados en diferentes partes del mundo. Todo lo que se requería era que Internet enlazara a todos. En cuatro años, más o menos, la utilización del sistema GOPHER creció de un servidor a más de 1300 servidores.

Para tener acceso al sistema GOPHER, se necesita un programa cliente. El programa cliente especial debe estar instalado en una computadora que se encuentre conectada a Internet. Existe software gratuito de clientes GOPHER para UNIX, Macintosh, IBM/PC, X Window, VAX/VMS, y probablemente más en la actualidad.

Lo importante de este sistema es que, en su mayor parte, todos los servidores GOPHER de Internet son públicos y la información que se encuentra es de interés general.

En el siguiente capítulo se presentará el servicio que quizá es el más empleado en la actualidad: WWW.

Capítulo 3

World Wide Web

3.1 Antecedentes

3.1.1 Definición

El World Wide Web también conocido como WWW o simplemente Web, es considerado como la forma más sencilla de acceder a los documentos disponibles en Internet, lo que implica que no importa la ubicación física o geográfica en la que se encuentren éstos.

Se considera como un sistema de distribución de la información contenida en Internet, está basado en un esquema cliente/servidor, lo integran software, protocolos y medios de comunicación que permiten el acceso a la información en diversos formatos como texto, gráficos, audio y video.

3.1.2 Historia y evolución

Como consecuencia del crecimiento de Internet, se generaron grandes cantidades de información surgiendo así la necesidad de generar una herramienta que permitiera poner a disposición de los usuarios la información contenida en Internet.

Aunque existían herramientas como GOPHER y FTP que permitían el acceso a tal información, éstas, resultaban ser un tanto imprácticas debido a que el acceso a ellas no era lo suficientemente “amigable” para aquellos usuarios que tan sólo requerían la información sin tener que aprender la utilización de las mismas, aunado a esto se presentaba una limitante: el manejo de imágenes.

Fue entonces cuando en 1989, Tim Berners-Lee del CERN (*Conseil Européen pour la Recherche Nucléaire o European Laboratory for Particle Physics*) propuso la elaboración de un “sistema de hipertexto”, con el propósito de tener en éste un modo de comunicación para la “Comunidad de Física de Alta Energía” (*High Energy Physics Community*) de la propia CERN; este sistema trabajaría inicialmente en la red de la comunidad antes mencionada y por medio del cual se obtendría acceso a diversos tipos de documentos distribuidos en lugares geográficamente separados.

Se utiliza el término de “hipertexto”, debido a que a éste se le considera como una técnica de organización de documentos por medio de ligas o enlaces, es decir, que un documento contiene asociaciones a otros documentos situados en lugares diferentes, entonces, por medio de un sólo documento se tendría acceso a una gran cantidad de documentos asociados al

primero, el hipertexto con sus enlaces se podía encontrar en diferentes documentos de manera que permitía asociar diferentes máquinas o inclusive diferentes redes.

A este sistema de hipertexto también se le denominó “World Wide Web” (Telaraña de Cobertura Mundial) o simplemente WWW, tomó este nombre debido a la representación gráfica del conjunto de redes que conforman Internet y la semejanza de ésta con una telaraña de redes.

Los componentes básicos que éste debía cubrir eran los siguientes:

- Interfaz consistente para el usuario
- Posibilidad de incorporar diversos tipos de documentos
- Permitir que cualquier persona y desde cualquier tipo de computadora tuviera acceso a los documentos de la manera más sencilla posible

3.2 Navegadores

3.2.1 Definición

Un “navegador” es la herramienta que permite poner a disposición de los usuarios la información disponible en el WWW, por lo que es capaz de interpretar tal información y hacerla presente mediante texto con formato, imágenes, video y sonido.

El hipertexto es la característica más importante del WWW, ya que como se mencionó anteriormente, permite que los archivos en WWW sean relacionados de manera que sea posible trasladarnos fácilmente de unos a otros documentos, esto se conoce comúnmente como “explorar”, es por esto que a los sistemas de hipertexto también se les conoce como *browsers*, navegadores o exploradores.

3.2.2 Historia y evolución

El programa navegador como intermediario entre un usuario y un servidor WWW resultó ser el medio más accesible para los usuarios, ya que por medio de éste se puede acceder a documentos que el servidor WWW tenga disponibles además de facilitar el manejo de los mismos mediante el uso de una interfaz gráfica.

Conforme se fue desarrollando el proyecto iniciado en el CERN, se obtuvieron diversos navegadores para el WWW de entre los cuales podemos mencionar Viola, NexStep y Mosaic, de éstos sobresale Mosaic, que a mediados de 1993 dio a conocer su primera versión y debido al gran auge que tomó por sus servicios provistos de gráficos, llegó a ser la interfaz más popular para el WWW durante esa época.

Los documentos que se utilizan en un navegador están realizados en HTML, éste es un lenguaje estándar que todos los navegadores usan, con respecto a este lenguaje se ampliará su concepto en el siguiente capítulo.

El acceso a la información mediante un programa navegador facilita al usuario el manejo de información desde cualquier parte del mundo; han surgido navegadores para diversas plataformas.

Los navegadores que han sobresalido son Netscape e Internet Explorer. A continuación se dará una breve explicación general de los elementos provistos en un navegador, como ejemplo se tomará la versión de *Netscape Navigator Gold* que se presenta en la figura 3-1.

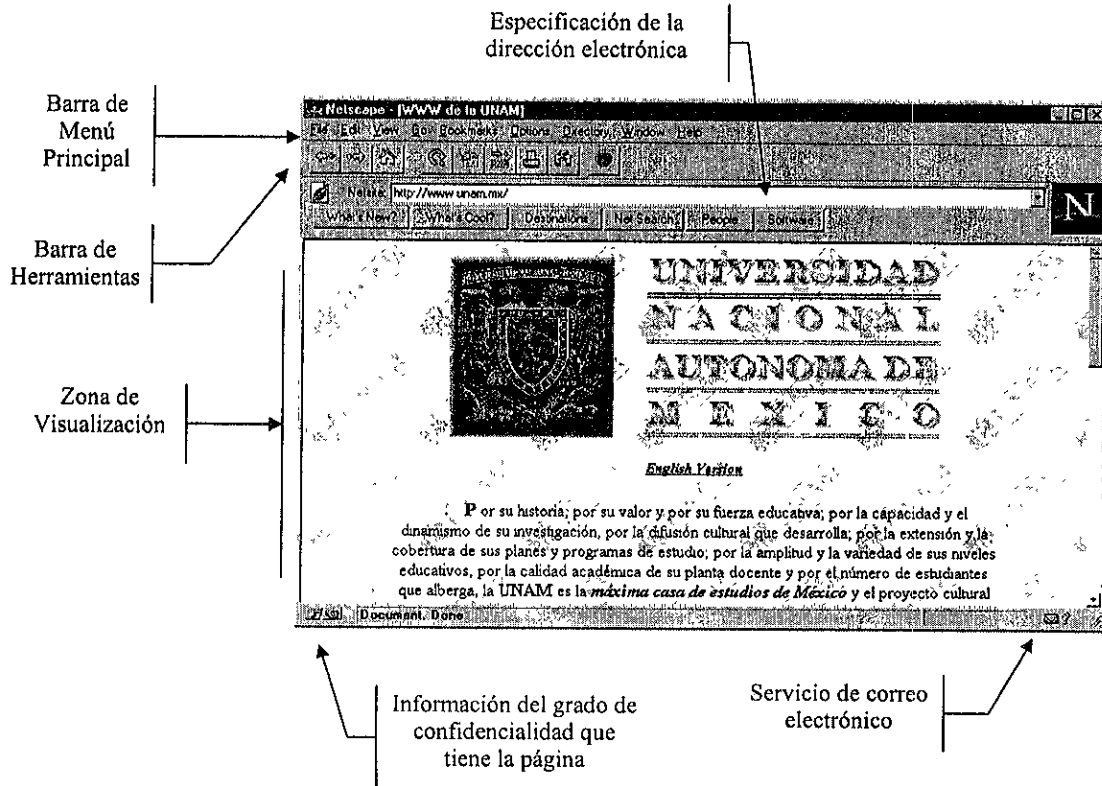


Figura 3-1

En la *Barra del menú principal*, tenemos los siguientes temas para los cuales se mencionan sus características principales:

- **File** Permite abrir: archivos, una nueva ventana de navegador, además de abrir y cerrar la aplicación
- **Edit** Mantiene el estándar de otras aplicaciones en cuanto a la edición de texto, es decir, permite copiar, editar, cortar, etc.
- **View** Brinda información general acerca de la página que se está visualizando (código, nombre de archivos de los que hace uso, etc.)
- **Go** En un navegador, se puede cambiar hacia páginas que se han visto con anterioridad y en este tema se presenta una opción para mantener una lista de las páginas que se han visitado durante la sesión actual, además de poder indicar la dirección de las páginas a visitar
- **Bookmarks** Debido a la facilidad de poder cambiar de página en página, se hace presente esta herramienta que permite “marcar” las direcciones

- **Options** electrónicas, para así tener un acceso directo a determinada página sin tener que seguir cierta secuencia de búsqueda. Permite cambiar configuraciones de inicio del navegador, especificar datos acerca del correo electrónico, así como asociar las aplicaciones que se encontrarán relacionadas con el navegador
- **Directory** Generalmente contiene ligas de información acerca de novedades y búsquedas en el WWW
- **Window** Permite mediante ligas, cambiar de ventana de visualización e ir por ejemplo hacia la ventana de servicio de correo electrónico o la de "bookmarks", además de que se puede ver una lista de las últimas n páginas a las que se ha accedido durante la sesión actual (donde n es un número que se especifica en la configuración del navegador)
- **Help** En general contiene información acerca de la versión del navegador y de las utilerías para este mismo

En la *Barra de Herramientas*, se encuentra un conjunto de botones establecidos para simplificar algunas ejecuciones que se llevan a cabo desde el Menú Principal.

En la *Zona de Visualización* se pueden ver los datos que se reciben de la página a la que se realizó el acceso.

La *Información respecto al grado de confidencialidad que tiene la página* se puede ver desde el ícono señalado, generalmente se muestra como una llave segmentada, sin embargo cuando la página hace uso de algún protocolo de seguridad, este ícono será una llave no segmentada o un candado, la liga que mantienen estos íconos envían hacia una nueva ventana en la que se proporciona información general acerca del grado de seguridad que mantiene la página.

El *Servicio de correo electrónico* es una de las aplicaciones que se ha anexado y generalizado en los navegadores, comúnmente se lleva a cabo una ejecución que abre una ventana que contiene utilerías propias del correo electrónico.

Finalmente en la *Especificación de la dirección electrónica*, se anota la dirección de la página hacia la que se desea realizar una conexión.

A medida que avanza el desarrollo de los navegadores, se adicionan herramientas a éste para proveer otro tipo de servicios integrados al mismo como servicio de correo electrónico, herramientas de desarrollo de páginas en HTML, servicios gráficos de FTP y Gopher, además de ofrecer interacción con diversas aplicaciones.

Entre los problemas que surgen cuando se utiliza una base de datos, y se desea mantener disponible a un conjunto de usuarios, se encuentra la diversidad de plataformas en las que trabaja cada uno de ellos, por esto, se elaboran diferentes programas para cada plataforma o se estandariza a una sola. Esta no es la solución óptima; con el surgimiento de los navegadores se crearon nuevos lenguajes de programación que facilitan el acceso a la información de una base de datos por medio de cualquier navegador y bajo cualquier plataforma, siendo ésta la mayor ventaja que presentan los navegadores.

Surgió además un enfoque al WWW de tal manera que sea posible comprar, hacer pedidos y pagar productos por el WWW, de hecho existen compañías comerciales (CyberCash Inc.,

Checkfree Corporation y VerySign entre otras) para desarrollar sistemas de transacción seguros en el WWW.

Algunos de los avances más importantes en el WWW ocurren en el área de multimedia con el desarrollo de sistemas de reproducción de sonidos e imágenes en “tiempo real”, un ejemplo proviene de la Universidad de Cornell, que desarrolló el primer programa para videoconferencias (CU-Seeme) a través del WWW.

El WWW actualmente (1998) está dirigido por *The World Wide Web Consortium*, éste se encuentra financiado por un gran número de miembros (AT&T, Adobe Systems Incorporated, Microsoft Corporation y Sun Microsystems Inc., entre otros), su propósito es promover el crecimiento del WWW desarrollando especificaciones y programas de referencia que se encuentren disponibles de forma gratuita a todo el mundo. El Consorcio está administrado por el Instituto Tecnológico de Massachusetts (MIT) junto con el Instituto Nacional Francés para la Investigación Informática (INRIA), como servidor en Europa, en colaboración con el CERN.

3.3 Servidores

3.3.1 Definición

Como se mencionó en el capítulo 1, un servidor es un “programa” encargado de llevar a cabo las peticiones de un “programa cliente”, en nuestro caso específico, un “servidor de WWW” es un programa diseñado para poner a disposición de los usuarios la información almacenada en él vía Internet utilizando un “cliente navegador” y el protocolo HTTP (*HiperText Transfer Protocol*, se proporciona información en las siguientes páginas).

3.3.2 Historia y evolución

Conforme surgieron diversos programas navegadores, se fueron desarrollando programas servidores de WWW.

De entre las tareas básicas que se llevan a cabo en un servidor WWW podemos mencionar:

- Administración del contenido de información
- Control de la información disponible a los usuarios
- Control de acceso a usuarios
- Mantenimiento del servidor

Anteriormente existían programas servidores de WWW sólo para plataformas UNIX, en éstos las tareas del administrador de WWW iniciaban desde el proceso de instalación y configuración del servidor hasta las tareas básicas mencionadas anteriormente, esto sólo se podía realizar por personal capacitado en el manejo administrativo de UNIX, además se tenía la desventaja de que únicamente se podían llevar a cabo modificaciones mediante la consola de la propia máquina servidor; todo esto obligó a generar nuevos proyectos de servidores que facilitaran el manejo de un servidor WWW.

Es así que surge una nueva generación de servidores WWW para diversas plataformas que presentan facilidad en el manejo administrativo de los mismos, además de incluir la utilización de la administración remota. Entre ellos podemos mencionar a NetScape Commerce.

Los proyectos de servidores WWW hasta 1998, se mantienen en desarrollo además de que se les han ido integrando herramientas adicionales de búsqueda, utilización de diversos tipos de manejadores de bases de datos y recientemente protocolos de seguridad, entre otros.

A continuación se presentan tablas comparativas de algunos servidores WWW.

* Gratuito para instituciones educativas
 Información obtenida al mes de enero de 1997.

Plataforma	APACHE Stronghold	Netscape Enterprise Server	Zeus
UNIX	✓	✓	✓
Windows 95			
Windows NT		✓	

Tabla 3-1

Elementos	APACHE Stronghold	Netscape Enterprise Server	Zeus
Manejo de generación de "documentos dinámicos"	✓	✓	
Acceso al servidor desde CGI o Scripts	✓	✓	✓
Servicio de búsqueda integrado		✓	

Tabla 3-2

Elementos de seguridad	APACHE Stronghold	Netscape Enterprise Server	Zeus
Acceso a datos de manera jerárquica, basado en dir IP	✓	✓	✓
Modificación en el control de acceso sin restaurar el servicio del servidor	✓	✓	
Manejo de control de autorizaciones por contraseñas	✓	✓	✓
Control de acceso por grupos	✓	✓	✓
Nivel de seguridad por archivos	✓	✓	✓
No permite el acceso a menos que se indique lo contrario en el archivo de accesos		✓	✓
Restricción por dominio	✓	✓	✓
Restricción por dirección IP	✓	✓	✓
Soporta SSL	✓	✓	✓

Tabla 3-3

Registro de accesos			
	APACHE Stronghold	Netscape Enterprise Server	Zeus
Puede escribir en múltiples servicios de accesos	✓	✓	✓
El archivo de accesos se genera automáticamente	✓	✓	
Los accesos se registran en el sistema		✓	
Control del archivo que registra los eventos		✓	

Tabla 3-4

Elementos adicionales			
	APACHE Stronghold	Netscape Enterprise Server	Zeus
Generación automática de directorio	✓	✓	
Instalación en formas GUI		✓	✓
Servicio de mantenimiento en formas GUI		✓	✓
Posibilidad de mantenimiento remoto mientras el servidor presta servicios		✓	✓

Tabla 3-5

Costos			
	APACHE Stronghold	Netscape Enterprise Server	Zeus
Comercial		✓ (*)	✓ (*)
Gratuito	✓		

Tabla 3-6

3.3.3 Interacción con el navegador

La operación de los servidores WWW emplea el hipertexto como pieza fundamental para interactuar con los navegadores.

Esta interacción se lleva a cabo a través del protocolo HTTP (*HyperText Transfer Protocol*), HTTP es un protocolo que permite la "comunicación" de un servidor de WWW y un navegador mediante cualquier texto realizado en HTML. Inicialmente éste era sólo un protocolo para el intercambio de texto en el lenguaje HTML, posteriormente, NCSA extendió la transmisión y despliegue de datos y gráficos.

El protocolo HTTP está basado en un modelo solicitud/respuesta, este proceso se lleva a cabo básicamente en 4 pasos:

- 1.- **Establecimiento de la conexión.** El cliente establece una conexión al sistema servidor de WWW. Esta conexión opera en el puerto que tenga asignado el servidor para el uso del protocolo TCP/IP (comúnmente puerto 80). El servidor WWW se encuentra en espera permanente en dicho puerto y en cuanto establece la conexión espera una petición

del cliente. Cabe mencionar que se establece una conexión por cada archivo que se lea, llámese texto, imagen, audio y video.

2.- **Petición.** La petición de un cliente a un servidor incluye la siguiente información:

Método

La información que solicite el cliente la puede realizar utilizando determinados métodos, entre los más comunes se encuentran:

- **GET.** Solicita un documento específico
- **HEAD.** Solicita únicamente el encabezado del documento
- **POST.** Solicita que el servidor acepte la información enviada como un archivo ejecutable y utilice cierta información
- **PUT.** Reemplaza el contenido de un archivo en el servidor con los datos enviados

Encabezado

El cliente puede enviar “campos de encabezado” al servidor. La mayoría son opcionales, los más comunes se muestran en la tabla 3-:

Encabezado	Descripción
Accept (Aceptación)	El tipo de archivo que el cliente aceptará.
Authorization (Autorización)	Se utiliza si el cliente desea identificarse con el servidor, proporciona información tal como nombre de usuario y contraseña.
User-agent (Agente)	El nombre y la versión del navegador.
Reference(Referencia)	Indica la dirección electrónica del último documento que el usuario accesó.

Tabla 3-7

Datos

Si el cliente ha realizado el método POST, puede enviar los datos después del encabezado y de una línea en blanco. Si el cliente envía un método GET o HEAD no se envían datos y el cliente sólo espera la respuesta del servidor.

3.- **Respuesta.** Las respuestas a requisiciones sencillas son archivos en formato HTML. Las respuestas del servidor incluyen la siguiente información:

- **Código de estado.** Cuando un cliente realiza una conexión, el servidor envía un Código de Estado, éste es un número de tres dígitos que se divide en cuatro categorías mismas que se presentan en la tabla 3-8.

Rango	Descripción	Ejemplo
200-299	Envío correcto.	200 La transmisión de datos ha sido correcta.
300-399	El documento ya no se encuentra ahí	302 La dirección electrónica deseada no se encuentra, sin embargo se realiza una redirección. La mayoría de los navegadores reconocen esta categoría. 304 Utilización de una copia local. Si el navegador tiene una página en caché, y es la que se requiere, el navegador indicará al servidor WWW, que se encuentra en caché. Si la copia del servidor no es más reciente que la copia, el servidor enviará un 304 en lugar de enviar toda la página, reduciendo así el tráfico de la red.
400-499	Error en URL	401 Este código numérico indica un acceso no autorizado. El usuario realizó la petición a determinado documento que requiere un nombre de usuario y contraseña válidos para el servidor WWW y los datos que proporcionó no son válidos. 403 El acceso a la dirección electrónica está "prohibido". Algunos servidores WWW tienen ciertas políticas de acceso a sus páginas. 404 Cuando se envía este código significa que la dirección electrónica que se requirió no se encuentra en el servidor, sin embargo, algunos servidores lo envían cuando una persona no autorizada realiza la petición y la página se encuentra restringida .
500 en adelante	El servidor no puede ejecutar la petición o ha ocurrido algún error	500 Se envía este código cuando se ha generado algún error de configuración del servidor.

Tabla 3-8

- **Encabezado.** La respuesta del encabezado contiene información acerca del servidor y del documento al que se hace acceso, en la siguiente tabla se muestran los encabezados más comunes:

Encabezado	Descripción
Servidor	Indica el nombre y versión del servidor WWW.
Fecha	Indica la fecha actual(GMT).
Última modificación	Apunta a la fecha en la que se realizó alguna modificación en la página.
Expiración	Se refiere a la fecha en que deja de ser válido el documento.
Contenido	Indica el tamaño en bytes de la página.
Tipo de contenido	Se refiere al tipo MIME de los datos .
Identificación WWW	Indica si requiere nombre de usuario y contraseña.

Tabla 3-9

- **Datos.** El servidor agrega una línea en blanco después de la última línea del encabezado y enseguida envía los datos requeridos.

4.- **Cerrar la conexión.** Las conexiones las cierra automáticamente el servidor hasta el término del envío de la petición.

3.4 Seguridad

3.4.1 Generalidades

En el WWW, se presentan diversos tipos de problemas de seguridad durante el envío de datos, éstos son principalmente:

- Modificación de datos
- Falsificación
 - de personas
 - de sitios

Una de las soluciones más comunes es el encriptamiento de la información, proceso que a continuación se revisa de manera general.

3.4.2 Encriptamiento

La encriptación es el proceso de transformar datos, de manera que no puedan ser interpretados por cualquier persona. Esta información procesada es llamada *texto criptográfico*, por lo que, si se envía información (datos) a través de Internet y se hace uso de algún proceso de encriptación, la información es encriptada y se convierte en un *texto criptográfico*, este proceso se muestra en la figura 3-2.

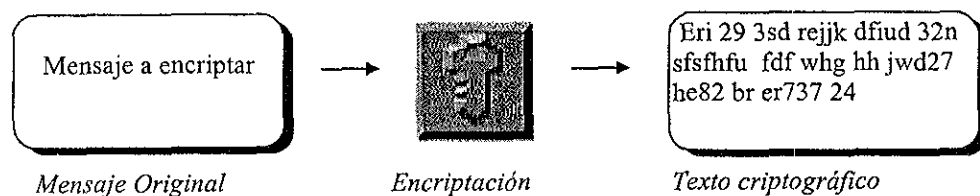


Figura 3-2

El proceso entonces, se realiza encriptando el cuerpo (texto) del mensaje con una "llave", ésta es una secuencia de bits que representa la correspondencia de valores que se utilizarán para la encriptación y/o desencriptación.

En los sistemas de encriptamiento tradicionales, se utilizaba sólo una "llave" para encriptar y desencriptar el texto, sin embargo, éste método resultó ser inseguro debido a que cualquiera que obtuviese la llave podía encriptar y desencriptar el texto fácilmente, a este método se le llamó *encriptamiento asimétrico*.

Debido a los riesgos que presentaba el método de *encriptamiento asimétrico*, surgió un sistema de *encriptamiento simétrico*, en este último se generan dos llaves, una para encriptar y otra para desencriptar los datos. Una de las llaves que se utilizan en este método es la “llave pública” utilizada para encriptar y la otra es la “llave privada”, esta última es utilizada para desencriptar.

Para el proceso de encriptación y/o desencriptación, se utilizan “llaves” además de *algoritmos de encriptación* (funciones matemáticas), de manera que, la llave proporciona valores de correspondencia y el algoritmo se encarga de utilizar esos valores para realizar los procesos establecidos por él mismo y así conformar una encriptación y/o desencriptación.

Algoritmos de encriptación

A continuación se mencionan algunos de los algoritmos de encriptación que se utilizan comúnmente:

- **Funciones de Hash para criptografía.** Una función de hash H es una transformación que toma como entrada una cadena m de longitud variable y regresa un resultado de longitud fija, llamado el valor de hash h ($h=H(m)$). Las funciones de hash son utilizadas extensamente en computación
- **Algoritmo Rivest-Shamir-Adleman (RSA).** RSA es un algoritmo que se fundamenta en el hecho de que la factorización de números primos es un problema “difícil” (en el sentido computacional)
- **RC-2, RC-4.** RC-2 es un algoritmo desarrollado por RSA que utiliza llaves de 40 o 56 bits. Además de la llave, se utiliza una cadena llamada “sal” que se concatena a la llave y se envía sin encriptar junto con la llave
- **MD2, MD4, MD5.** Son funciones de hash para uso en criptografía. Son empleadas principalmente para generar huellas digitales de documentos. Los tres algoritmos generan salidas de 128 bits

3.4.3 Autenticación y Certificados

En Internet, una identificación toma la forma de un “certificado de autenticación” o simplemente “certificado”, éste comprueba que quien lo representa es verdaderamente quien dice ser, por lo que no es transferible. Existe además la Autoridad Certificadora (*Certification Authority*), o CA, compañía que supervisa la autenticidad de estos certificados.

Tanto servidores como clientes pueden tener certificados, inclusive los clientes pueden tener varios certificados.

Cuando el servidor de WWW envía un certificado a su cliente, se realiza una “autenticación del servidor” y cuando el cliente envía un certificado al servidor se lleva a cabo una “autenticación del cliente”.

Cuando se encuentra habilitado un protocolo de seguridad en el servidor, se llevan a cabo los siguientes pasos:

1. Inicialmente el cliente realiza una conexión con el servidor.
2. El servidor envía información de su certificado pero éste no es encriptado.

3. El cliente utiliza la llave pública, ésta es enviada con el certificado del servidor como verificación de que la llave es del certificado que se envía.
4. El cliente decidirá si acepta el certificado o no.

Contenido del certificado

Un certificado es un archivo que contiene información acerca del propietario, este archivo está respaldado por la llave privada de CA's como garantía. Generalmente, se utiliza un estándar del certificado conocido como x.509v3 el cual contiene la siguiente información:

- El apartado de información del propietario: nombre, organización, dirección y demás información concerniente al mismo
- El apartado de la llave pública contiene la fecha de inicio y terminación de la validez del certificado, así como su número de serie

Algunos estándares permiten mostrar información más detallada, por ejemplo el área en la que se encuentra relacionada la compañía que lo presenta así como el tiempo que ésta lleva, todo esto para mayor seguridad de la autenticidad del certificado.

Existen sistemas de encriptación que utilizan algoritmos y certificados para instalar seguridad en el WWW, de los cuales se mencionan:

- **Kerberos.** Kerberos es un sistema de autenticación para ser utilizado en redes potencialmente inseguras (como es el caso de Internet), su objetivo es permitir que un usuario certifique su identidad con un servidor determinado. Dicha identificación se realiza a través de "boletos", éstos son válidos por tiempo definido, y son sólo válidos dentro de un *realm*, éste determina un conjunto de computadoras y usuarios, normalmente, una organización. Cada usuario es conocido como principal (un principal puede ser también un proceso). El principal contacta al Centro de distribución de Llaves (*Key Distribution Centre*, KDC por sus siglas en inglés) del *realm* para solicitar el "boleto" que utilizará para hablar con el Servidor de Derechos (*Ticket Granting Server*, TGS) del *realm*. El KDC verifica la identidad del principal y otorga el "boleto".
- **SSL.** SSL (*Secure Sockets Layer*) es un esquema de seguridad diseñado por Netscape Communications Corporation, el protocolo SSL incluye revisión para autenticación del servidor (verificación del servidor al cliente), encriptamiento de datos y opcionalmente verificación del cliente (verificación de la identidad del cliente ante el servidor). SSL (en 1998) se encuentra habilitado para navegadores tales como: Netscape Navigator, Secure Mosaic y Microsoft Internet Explorer y para diferentes tipos de servidores WWW de: Netscape, Microsoft, IBM, Quarterdeck, OpenMarket and O'Reilly and Associates.

Este protocolo proporciona:

- Confidencialidad. SSL lleva a cabo un proceso de encriptación de manera que se garantiza que los datos no puedan ser interceptados
- Integridad. Asegura que los datos se mantendrán correctamente hasta su destino, y si acaso trataran de descifrar el mensaje, se envía al destino un mensaje anexo a los datos enviados, con una señal de aviso

SSL permite a las aplicaciones cliente/servidor comunicarse de manera que los datos no puedan ser interceptados.

La ventaja del protocolo SSL es que se considera un protocolo “independiente”, es decir que aplicaciones tales como HTTP, FTP y TELNET pueden utilizarlo sin tener problemas. SSL establece un algoritmo de encriptamiento y una “llave de sesión” (llave pública) además de que autentifica el servidor antes de que el protocolo de la aplicación transmita o reciba el primer byte de datos, de manera que cualquier tipo de aplicación del protocolo envía los datos encriptados asegurando su privacidad.

SSL proporciona un “canal de seguridad” el cual tiene básicamente tres componentes:

- Privacidad. El canal de privacidad es reservado. El proceso de encriptamiento se utiliza para todo tipo de mensajes después de una “negociación” para definir una “llave secreta” (llave pública)
- Autenticación. El canal es autenticado. Al punto final de llegada (cliente), se le proporciona una “descripción” del servidor y opcionalmente se presenta una autenticación del cliente
- Confiabilidad. El canal es confiable. El transporte del mensaje incluye chequeo de la integridad del mensaje (se utiliza una *MAC*¹)

Los niveles de seguridad que se manejen dependerán del sistema de encriptamiento a utilizar.

El proceso que se realiza entre el cliente y el servidor que hace uso de protocolos de seguridad, diferirá dependiendo del sistema de encriptación que se utilice.

A continuación se mostrará, de manera general, el esquema de encriptación simétrica:

1. Cuando el cliente se conecta a un servidor WWW habilitado con algún protocolo de seguridad, se le notifica (comúnmente mediante cuadros de diálogo) el tipo de certificado que éste utiliza así como la información general acerca del mismo. Si el cliente acepta la conexión, el servidor envía su certificado y una *llave pública*, todo el proceso anterior se puede ver en la figura 3-3.

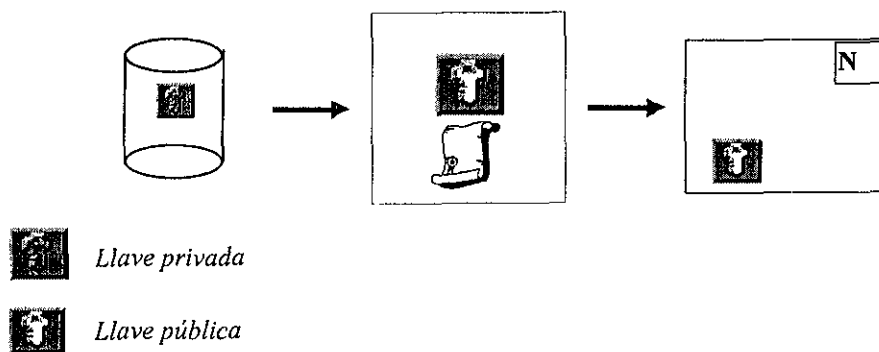


Figura 3-3

¹ MAC *Message Authentication Codes*, etiquetas de autenticación de mensajes

2. Cuando el cliente lleve a cabo una petición al servidor ésta será encriptada por el cliente (con la llave que le proporcionó el servidor al iniciar la conexión), este proceso de encriptación es automático. El servidor lo recibirá y lo desencriptará con una llave privada. Una vez que el servidor mande algún mensaje al cliente, éste se enviará encriptado por la llave privada, de manera que el cliente tendrá una llave pública que le servirá para desencriptar el mensaje enviado por el servidor (proceso mostrado en la figura 3-4).

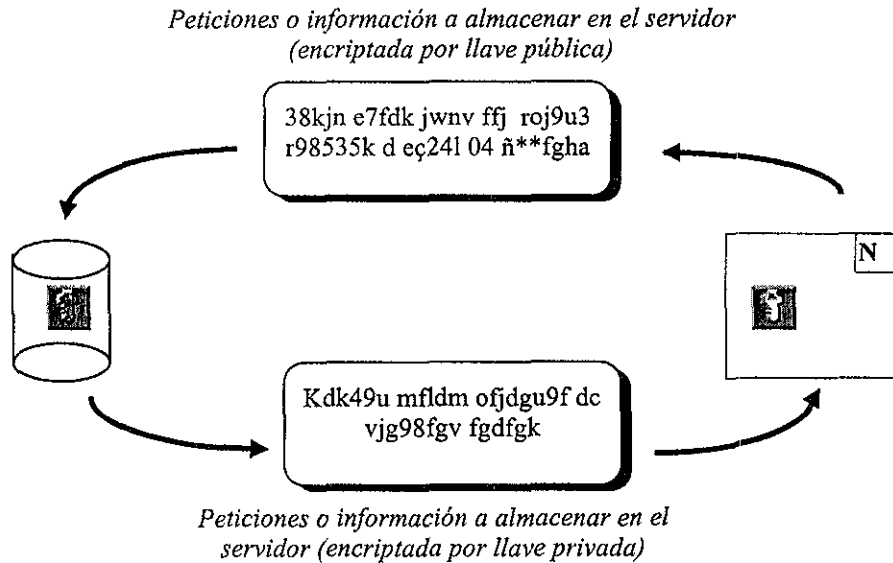


Figura 3-4

De esta manera, una llave pública sólo podrá encriptar los datos al ser enviados al servidor y podrá desencriptar los mismos si provienen de una llave privada (que le corresponda a la llave pública) y una llave privada podrá encriptar y desencriptarlos datos de sí misma o de una llave pública.

Para hacer uso de un sistema de encriptación, generalmente se requiere un certificado de autenticación autorizado por la CA, éstos representan un costo significativo, sin embargo, existen programas que se utilizan para generar certificados aunque algunos navegadores no permiten la conexión a éstos debido a que no son reconocidos por la CA, es decir, que no pueden afirmar que quien dice ser, es al que pertenece el certificado.

Enseguida se procederá a especificar el formato HTML así como algunos programas que pueden ser utilizados en el WWW, para así ser complementados con los navegadores y obtener los resultados que se deseen.

Capítulo 4

HTML

4.1 Antecedentes

La primera especificación del lenguaje HTML fue realizada en el año de 1991 por Tim Berners-Lee perteneciente al CERN como parte de la iniciativa del WWW para facilitar la comunicación entre diferentes tipos de computadoras, lográndose un rápido intercambio de información y conocimientos dentro de la comunidad del WWW.

El lenguaje HTML es considerado como una DTD (*Definición de Tipo de Documento*) del Estándar Generalizado para Lenguajes de Marcación (SGML, ISO 8879:1986), por lo tanto se dice que HTML es una colección de estilos independientes, definidos por etiquetas de marcación. Estos estilos son independientes de los tipos de plataformas existentes. Donde el objetivo principal de HTML es el contenido del documento, y no su apariencia.

HTML por ser parte del SGML es un lenguaje para describir documentos estructurados, debido a que la mayoría de los documentos cuentan con elementos comunes como: títulos, párrafos, encabezados, etc., por ello, si antes de dar inicio a la escritura del documento se definen un conjunto de elementos que lo conformen entonces se podrán etiquetar con los nombres adecuados.

Un documento WWW está formado por elementos, los cuales están representados a través de etiquetas (*del inglés "tags"*) de HTML. Estas etiquetas son las que describen al documento en sí, donde cualquier cosa que esté representada por su etiqueta correspondiente se considera como parte del mismo documento.

La especificación de HTML no permite determinar de manera exacta la ubicación o la apariencia de cada uno de los elementos de una página WWW, debido a que se desconocen desde un principio las distintas capacidades de la plataforma en que será visualizada la página como: el tamaño de la pantalla, los tipos de letras que se encuentren instalados, etc.

Por lo tanto, al separar la estructura de un documento y su apariencia, un programa que "lee" y "entiende" el lenguaje HTML puede tomar decisiones de formato del documento basándose en las capacidades de la plataforma en que se visualice.

Los navegadores de WWW, además de proporcionar las funciones de red para recuperar documentos de Internet, también pueden tener propiedades de formato de documentos escritos en lenguaje HTML, esto se presenta en el momento de la "carga" del documento en la pantalla, siendo el navegador el que "lee" o "analiza" la información, dando formato al texto y a las imágenes que componen al documento sobre la pantalla.

Si se utilizan distintos navegadores, se podrán observar diferencias en la presentación de un mismo documento, por ejemplo, los encabezados podrán ser visualizados centrados con un navegador y en otro el texto puede aparecer con un tipo de letra diferente al primero.

Por lo anterior se puede decir que el lenguaje HTML no cuenta con la propiedad de ser considerado como un WYSIWYG (del inglés "*What You See Is What You Get*"), cuya traducción al español significa "Lo que ve es lo que obtiene". Ésta característica se aplica más frecuentemente a editores para HTML.

4.1.1 Ventajas de HTML

Los documentos escritos en HTML ocupan poco espacio en disco, logrando que la transferencia a través de Internet así como su despliegue en pantalla sea de manera rápida.

Los documentos escritos con HTML son compatibles para cada uno de los diferentes tipos de plataforma e independientes de cada uno de los dispositivos con que cuenten, lo que hace posible que se trabaje el mismo documento en distintas plataformas, siempre y cuando ésta cuente con un navegador de WWW que sea capaz de "leer" y "entender" HTML.

A diferencia de otros lenguajes HTML se considera pequeño en comparación de otros como: PostScript o Troff en el sistema UNIX, es muy sencillo de entender, pues no hay que memorizar demasiadas etiquetas y desde el primer contacto con HTML se pueden crear documentos básicos, es decir, no se necesita de un aprendizaje prolongado para poder llevar a cabo la primera creación de un documento.

La mayoría de los navegadores soportan HTML 3.0 que es una revisión importante del HTML estándar, conteniendo la mayoría de las estructuras que dan forma a las páginas como son: los encabezados, párrafos, listas, imágenes, formularios y tablas.

4.1.2 Requisitos

Para poder crear un documento HTML se requiere:

- Un programa editor de texto. El documento que contendrá las etiquetas de HTML deberá ser escrito en un archivo de tipo texto, que al guardarlo en disco deberá tener la extensión htm o html, dependiendo del tipo de plataforma en que se trabaje, esto se debe a que algunas no admiten más de tres caracteres en la extensión del nombre del archivo.

Ejemplos de estos editores de texto son: qedit, aurora, vi, joe, pico, MS Internet, edit, notepad, etc.

- Experiencia en el manejo de algún navegador HTML como: Netscape, Mosaic, MS Internet Edit, además de entender la filosofía de WWW y URL. /

4.1.3 Estructura de un Documento escrito con HTML

Algunas características que deben ser consideradas con respecto a HTML son:

- No toma en cuenta los espacios en blanco, los cambios de línea, ni las tabulaciones propias de un documento normal, a menos que sean especificadas con sus etiquetas correspondientes como elementos del documento
- No hace diferencia entre mayúsculas y minúsculas al escribir las etiquetas HTML, lo anterior permite obtener documentos con una buena presentación
- Las etiquetas HTML permiten establecer las características del texto a presentar, definen atributos como: color, tamaño, forma, etc.
- La ventaja principal de HTML es su habilidad de ligar texto y/o una imagen a otro documento
- Las etiquetas HTML se encierran entre "picoparéntesis" (<>) dentro de los cuales va el nombre de la etiqueta. Las etiquetas de un elemento que forma parte de un documento, comúnmente se presentan por pares, para darle inicio y fin a la instrucción que definirá al elemento correspondiente. La etiqueta final es igual a la inicial sólo que contiene una diagonal (/) que es la que indica que ésta es la etiqueta de fin de instrucción </>

Cabe mencionar que algunas veces se encuentran etiquetas que no necesariamente tienen su par final, como por ejemplo:

- La etiqueta de inicio de párrafo <P> la que al finalizar un párrafo no necesita que se escriba su par final </P> basta con indicar sólo la primera etiqueta
- La etiqueta de ruptura de línea

- La etiqueta de elemento de lista
- La etiqueta de término de definición <TD>
- La etiqueta de descripción de definición <DD>

Cada documento HTML debe contener ciertas etiquetas HTML estándar, si se considera que por lo general un documento consiste de un encabezado el que a su vez debe contener el título, y el texto real que da cuerpo al documento conteniendo párrafos, listas, tablas, imágenes etc.

4.2 Elementos

Un documento HTML está compuesto en su estructura general por cada uno de los siguientes pares de etiquetas también conocidos como elementos:

- **<HTML> ...</HTML>**

El elemento <HTML> le dice al navegador que el archivo contiene información codificada en lenguaje HTML, lo mismo ocurre con la extensión del nombre del archivo a visualizar en el navegador.

- **<HEAD> ...</HEAD> (Cabecera)**

El elemento HEAD identifica la primera parte del documento HTML, que contiene el título del documento.

Ejemplo de Cabecera:

```
<HEAD>
<TITLE>Ejemplo de Cabecera</TITLE>
</HEAD>
```

Los elementos permitidos dentro del elemento de cabecera son:

- **<ISINDEX>**

El elemento ISINDEX indica que el navegador debería permitir al usuario buscar un índice mediante claves.

- **<TITLE> </TITLE> (Título)**

Cada documento HTML debe tener un elemento de título, donde el título debe identificar al contenido del documento en un contexto global.

- **<NEXTID> (Siguiente Identificador)**

El elemento NEXTID está incluido por razones históricas solamente, la etiqueta <NEXTID> pretendía proporcionar el siguiente identificador disponible para los editores automáticos de hipertexto. Ha sido superada por la etiqueta <LINK>.

- **<LINK> (Liga)**

El elemento LINK representa una hiperliga, es típicamente usado para indicar autoría, índices y glosarios relacionados, versiones más antiguas o más recientes, jerarquía de documentos, etc.

- **<BASE> (Dirección Base)**

El elemento opcional BASE provee una dirección base para interpretar URL's relativas, cuando el documento es leído fuera de contexto.

- **<BODY> (Cuerpo)**

La segunda parte del documento HTML es el cuerpo, el cual comprende el contenido del documento (desplegado dentro del área de texto en la ventana del navegador).

Los elementos más usuales en el cuerpo del documento son:

- **Elementos de Texto**
 - <P>
 - <PRE> ...</PRE>
 - <LISTING> ...</LISTING>
 - <BLOCKQUOTE>...</BLOCKQUOTE>
- **Encabezados**
 - <H1>...</H1>
 - <H2>...</H2>
 - <H3>...</H3>
 - <H4>...</H4>
 - <H5>...</H5>
 - <H6>...</H6>
- **Estilos Lógicos**
 - ...
 - ...
 - <CODE>...</CODE>
 - <SAMP>...</SAMP>
 - <KBD>...</KBD>
 - <VAR>...</VAR>
 - <DFN>...</DFN>
 - <CITE>...</CITE>
- **Estilos Físicos**
 - ...
 - <I>...</I>
 - <TT>...</TT>
- **Lista de Definición/Glosario <DL>**
- **Lista no ordenada **
- **Lista ordenada **
- **Menú interactivo <MENU>**
- **Lista de directorio <DIR>**

Algunos elementos pueden incluir un atributo, donde se entiende por atributo a la información adicional incluida en la etiqueta de inicio.

Una especificación de atributo consiste de un nombre que corresponde al atributo, un signo igual y un valor, aunque algunas especificaciones de atributos pueden incluir solamente al nombre. Permitiendo un espacio en blanco a ambos lados del signo igual, donde el valor de un atributo puede ser una cadena, delimitada por comillas sencillas o dobles.

Ejemplo:

```
<A NAME="Arriba" HREF="../../../menu.html" TITLE="Menú Principal">
```

Los atributos adicionales para este comando son: NAME, HREF y TITLE.

Dentro del elemento BODY se pueden definir el color del fondo del texto (background), de las ligas no visitadas, de las ya visitadas así como de las ligas activas.

El comando que realiza lo descrito es:

```
<body background="#hexcol" text="#hexcol" link="#hexcol" vlink="#hexcol" alink="#hexcol">
```

Donde **background** (bgcolor) es utilizado para realizar cambios al color del fondo del documento, **text** es utilizado para modificar el color del texto, **link** para las ligas no visitadas, **vlink** para las ligas visitadas y **alink** para las ligas activas. La parte de #hexcol indica el color en sus componentes **RGB** (red-green-blue) en notación hexadecimal. Los colores se componen de seis letras y/o números, donde los dos primeros expresan la cantidad de rojo, los dos intermedios la cantidad de verde y los dos últimos la cantidad de azul.

Ejemplos de colores en sus componentes RGB se muestran en la tabla 4-1.

Color	Hexcol
Blanco	#FFFFFF
Negro	#000000
Rojo	#FF0000
Verde	#00FF00
Azul	#0000FF
Amarillo	#FFFF00
Cyan	#00FFFF
Magenta	#FF00FF
Gris	#C0C0C0
Azul Cielo	#3299CC
Rosa	#BC8F8F
Azul Marino	#23238E
Anaranjado	#FF7F00
Crema	#FFFFD8

Tabla 4-1

También es posible cambiar el color de una parte del texto (sólo el texto sin ligas) por medio del comando , el cual también es utilizado para cambiar el tamaño de letra.

El comando para cambiar el color de las letras es el siguiente:

```
<font color="#hexcol"></font>
```

El texto que se encuentre entre estos comandos tendrá el color que se le indique por medio de hexcol.

El color del texto y de los vínculos también llamados ligas o hiperligas, puede cambiarse sólo una vez por página.

Para dibujar líneas horizontales dentro de la página que la divida se utiliza el comando:

Línea <HR>

El formato completo del comando es:

```
<HR Align=X size=Y width=Z noshade>
```

Donde **X** indica la alineación de la línea, sus posibles valores pueden ser left, right o center, la **Y** indica el tamaño vertical (grueso) de la línea en pixeles, la **Z** indica el tamaño horizontal de la línea, expresado en pixeles o en un porcentaje.

Ejemplo:

Línea centrada de tamaño 25 y un ancho de 150

```
<hr align=center size=25 width=150> .
```

Línea centrada, de grueso por default, ancho de 75% sin sombra

```
<hr align=center width=75% noshade>
```

Línea con parámetros por default: `<hr>`

Todos los elementos antes mencionados son sólo algunos de los más utilizados para dar forma a un documento escrito en lenguaje HTML.

4.2.1 Descripción de algunas etiquetas

Párrafo <P>

Los párrafos son rodeados por un espacio horizontal de una línea o media línea. A diferencia de los documentos generados dentro de la mayoría de los procesadores de palabras, los retornos de carro en los archivos HTML no son relevantes. El ajuste de línea puede ocurrir en cualquier punto dentro del archivo fuente y los espacios múltiples son agrupados en un solo espacio por el navegador. Un navegador ignora las rupturas de línea e inicia un nuevo párrafo, sólo cuando encuentra otra etiqueta <P>. La etiqueta de fin </P> puede ser omitida, porque los navegadores entienden que al encontrar una nueva etiqueta <P> implica que hay un fin para el párrafo previo, inserta automáticamente una ruptura de línea al final del párrafo, y produce un espaciado conveniente entre los diferentes párrafos del documento, adicionalmente permite alinear el texto al centro, a la derecha o a la izquierda.

El formato completo de esta etiqueta es: `<P align="X">`, donde **X** puede ser left, right o center.

Ejemplo: `<P align="center">`Esta línea aparecerá al centro.

La alineación no se cambiará mientras no se utilice otro comando <P> con la opción de align.

La sangría exacta, y los espacios en blanco de un párrafo no se especifican y pueden ser función de otras etiquetas.

**Break
**

Cuando no se desea iniciar un nuevo párrafo sino sólo cambiar de línea, lo que se utiliza es el comando
 (del inglés "break") o ruptura en este caso de línea.

Texto Preformateado <PRE>

El elemento PRE representa a un bloque de texto, y es apropiado para texto que ha sido formateado para un tipo de letra que presente espacios definidos. Esta etiqueta también considera a los espacios en blanco, cambios de línea, tabuladores, así como a los espacios múltiples, que aparecen dentro del archivo HTML original.

La etiqueta PRE puede ser utilizada con el atributo opcional WIDTH, el atributo WIDTH especifica el número máximo de caracteres para una línea y permite al navegador HTML seleccionar una fuente e indentación apropiadas.

Listado <XMP>, <LISTING>

En estos elementos no se reconoce otra marca que el fin de etiqueta, debido a que este tipo de elementos tienen un número de técnicas de procesamiento no muy recomendables. Entonces se recomienda que los documentos HTML no contengan los elementos XMP y LISTING (la etiqueta PRE es más expresiva y más consistente).

Bloque Entrecomillado <BLOCKQUOTE>

El elemento BLOCKQUOTE se utiliza para crear una cita (A diferencia de la etiqueta <CITE>, que realiza citas breves <BLOCKQUOTE> se utiliza para citas más largas, que no deben anidarse dentro de otros párrafos). las citas extensas por lo general se separan del texto mediante una sangría o de alguna otra forma.

**Énfasis **

El elemento EM indica una frase enfatizada, normalmente presentada en letra cursiva.

**Énfasis Fuerte **

El elemento STRONG indica un énfasis fuerte, normalmente presentado en negrillas.

Código <CODE>

El elemento CODE indica un ejemplo de código. El elemento CODE esta pensado para palabras o frases cortas de código.

Muestra <SAMP>

El elemento SAMP indica una secuencia de caracteres, típicamente mostrados en un tipo de letra con un solo espacio.

Teclado <KBD>

El elemento KBD indica texto tecleado por un usuario, normalmente desplegado en un tipo de letra de un solo espacio. Esto es común en manuales de instrucciones.

Variable <VAR>

Se utiliza para representar una variable en un programa de computadora, donde el usuario puede reemplazar la variable con información específica desplegada en letra cursiva.

Definición <DFN>

Identifica la *instancia de definición* de una palabra o término. La instancia definición se refiere a la primera vez que se utiliza dicha palabra o término. Internet Explorer exhibe el contenido de la etiqueta <DFN> en letras itálicas y Netscape Navigator no hace ningún énfasis.

Cita <CITE>

El elemento CITE es utilizado para indicar el título de un libro u otra cita. Es generalmente desplegado en cursivas.

**Negrilla **

El elemento B indica texto en negrillas.

Cursivas <I>

El elemento I indica texto en cursivas.

Blink <Blink>

El texto que este incluido dentro de este comando se verá flasheando, esto es, apareciendo y desapareciendo continuamente.

Centrado <Center>

Este comando permite otra forma de centrar algún elemento perteneciente al documento, a diferencia del comando <p align="center">, el comando <center> permite centrar texto, tablas, listas o gráficos.

Para modificar el tamaño del texto se utilizan los siguientes comandos:

<BaseFont Size=Z>

Donde la Z representa un número entero del 1 al 7. La letra más grande es representada por el número 7 y la más pequeña por el número 1.

** y **

Donde la Z puede tomar los siguientes valores: un valor absoluto del 1 al 7, ó un valor relativo de aumento o disminución al valor definido con el comando <basefont>, con el formato +Z o -Z.

Ejemplo:

```
<BaseFont Size=4>Esta línea tendrá un tamaño 4 <BR>
<Font Size=-1>Esta línea tendrá un tamaño 3 (4-3)</Font><BR>
<Font Size=+2>Esta línea tendrá un tamaño 6 (4+2=6)</Font>
<Font Size=7>Esta línea tendrá el tamaño más grande</Font><BR>
<Font Size=1>Esta línea tendrá el tamaño más pequeño</Font><BR>
```

Teletipo <TT>

La etiqueta TT exhibe su contenido en una fuente con ancho fijo. El texto dentro de la etiqueta TT compacta en uno los espacios e ignora los caracteres de línea nueva.

Ancla <A>

El elemento ancla A (*del inglés "Anchor"*) sirve para establecer el principio y fin de una liga de hipertexto o hiperliga, así como definir el destino de una liga. Tiene dos formatos básicamente, el primer formato permite definir una liga hacia otro documento o hacia otra parte del mismo documento. La forma de realizar esto es por medio de en donde "localización" representa la dirección del documento con el cual se desea establecer la liga, el texto que se coloque entre las etiquetas <a> y estará en otro color y subrayado, indicando que si se oprime el botón del ratón (del inglés "*mouse*") en ese texto, habrá un desplazamiento definido en el href.

Atributos del elemento A:

HREF: Especifica hacia donde se dirige el navegador cuando se oprime el botón correspondiente del ratón, puede utilizar cualquier URL valido como destino

NAME: Proporciona el nombre del ancla y la habilita como la cabeza de una hiperliga
TITLE: Especifica un título para la página WWW
REL: El atributo REL brinda la(s) relación(es) descritas por la hiperliga. El valor es una lista separada por espacios en blanco de nombres de relaciones
REV: Igual que el atributo REL, sólo que la semántica de la relación es en orden Inverso

Ejemplo:

Una liga de A a B con REL="X" expresa la misma relación que un liga de B a A con REV="X"

Las anclas son identificadas por una dirección de ancla, compuesta por un URL absoluto, opcionalmente seguido por el caracter '#' y una secuencia de caracteres llamados identificadores de fragmento.

Para incluir un ancla en el documento:

- Empezar el ancla con <A>
- Especificar el documento al que se hace la liga mediante el parámetro REF="filename" seguido por el delimitador '>'
- Escribir el texto que servirá como la liga de hipertexto en el documento actual
- Terminar con la etiqueta de fin

Se pueden hacer ligas a documentos en otros directorios especificando la ruta relativa desde el documento actual al documento ligado. Los nombres de ruta usan la sintaxis estándar de UNIX.

Cada una de las siguientes etiquetas indican la cola de una hiperliga o conjunto de hiperligas:

Elementos A con HREF presente.
Elementos LINK.
Elementos IMG.
Elementos INPUT con el atributo SRC presente.
Elementos ISINDEX
Elementos FORM con 'METHOD=GET'.

Estas etiquetas refieren a cabezas de ancla mediante un URL, ya sea absoluto o relativo, o un identificador de fragmento, o ambos.

Si se desea colocar una liga donde sea posible recibir correo electrónico, solo basta con colocar en el localizador la palabra "mailto" y la dirección electrónica correspondiente.

Ejemplo: Envíe un correo electrónico

Con esto se indica al navegador que debe abrir la ventana de correo electrónico.

Si lo que se desea es colocar una liga a alguna parte dentro del mismo documento, lo que se debe hacer es: poner como localizador la expresión #nombre, donde nombre es la palabra con la cual se define la parte específica a la cual se quiere que exista la liga. Para definir el nombre en la parte a la cual se desea que se conecte se utiliza la otra forma del comando: .

Ejemplo:

```
Indice
<a href="#capitulo1">Capítulo 1 : Introducción </a><br>
<a href="#capitulo2">Capítulo 2 : Comandos Principales</a>
<a name="capitulo1">
<h1>Capítulo 1 </h1>

<a name="capitulo2">
<h1>Capítulo 2 </h1>
```

- **Ligas**

Una hiperliga es una relación entre dos anclas, llamadas la cabeza y la cola de una hiperliga respectivamente.

Un navegador resalta el texto identificado o la imagen con color y/o subrayas para indicar que se trata de un texto de hiperliga.

- **Ligas a Secciones Específicas**

Las anclas pueden también ser usadas para mover a un lector a una sección particular dentro de un documento. Este tipo de ancla es comúnmente llamado un *ancla nombrada*, porque para crear las ligas se necesitan insertar nombres HTML en el documento. Las hiperligas internas son usadas para crear una *tabla de contenido* al principio del documento.

- **Ligas entre Secciones de Diferentes Documentos**

Si se desea establecer una liga de un documento A (documentoA.html) a una sección específica en otro documento (DocumentoX.html).El comando para realizar esta liga es:

```
<A HREF="DocumentoX.html#DOC">Ejemplo de Liga</A>
```

El identificador de fragmento después del signo '#' es un tabulador dentro del archivo DocumentoX.html. Éste tabulador dice al navegador lo que se debe desplegar al principio de la ventana cuando la liga es activada. Después se crea el ancla nombrada (en este ejemplo "DOC") en el archivo DocumentoX.html :

```
<H2><A NAME="DOC">Ejemplo de Liga</A></H2>
```

Con ambos elementos en su lugar, se puede llevar a un lector directamente a la referencia DOC dentro de DocumentoX.html.

- **Ligas a Secciones Específicas dentro del Documento Actual**

La técnica es la misma excepto porque el nombre del archivo es omitido. Por ejemplo, para ligar a la ancla DOC dentro del archivo DocumentoX, solo hay que escribir :

```
<A HREF="#DOC">Ejemplo de Liga</A>
```

- **Encabezados:** <H1> ...<H6>

Los seis elementos de encabezado, H1 a H6, denotan encabezados de sección, el valor 1 se utiliza para el encabezado más importante y el 6 para el de menor importancia. Aunque el orden y ocurrencia de los encabezados no está restringido por el DTD de HTML, se recomienda que dentro de los documentos no existan saltos entre los niveles de encabezados (por ejemplo, de H1 a H3).

Ejemplo:

```
<H1>Esta es la línea principal de Encabezado</H1>  
  A continuación va texto normal  
<H2>Este es un encabezado de Nivel 2 más pequeño</H2>
```

4.3 Listas

Hay tres tipos de listas :

1. **Desordenadas:** Son aquellas que no llevan un orden específico, por lo que es posible listar elementos con marcas (*del inglés "bullets"*) o viñetas. Esto se puede lograr por medio de los comandos (*del inglés "unordered list"*) o lista desordenada, su par final y éste comando no necesariamente debe ir acompañado de su etiqueta final.
2. **Lista ordenada:** Sus elementos están ordenados secuencialmente, los comandos que se utilizan son :

 (de ordered list o lista ordenada), su etiqueta final y

El comando completo es: <ol type=a start=b>, donde **a** nos indica el tipo de secuencia, 1 (1,2,3....),A(A,B,C,....),a(a,b,c,....),I(I,II,III,....),i(i,ii,iii,....),.la **b** nos indica el valor inicial (optativo) de la lista.

Los dos parámetros son optativos y de no incluirse, la lista será del tipo 1 con valor inicial 1.El comando se utilizará de la misma forma, que con las listas desordenadas.

Ejemplo:

Instrucciones para llevar a cabo una conexión a Internet:

```
<ol type=1>  
<li>Entre a Windows  
<li>Abra el ícono de Internet  
<li>Seleccione el programa TCP/IP  
</ol>
```

3. **Listas de definiciones:** Es una lista de términos (<DT>) y definiciones correspondientes (<DD>).

Ejemplo de una Estructura General de una Lista Definida:

```
<dl>  
<dt>Término 1
```

```
<dd>Definición del Término 1
<dt>Término 2
<dd>Definición del Término 2
</dl>
```

Los navegadores de WWW generalmente dan formato a la definición en una nueva línea.

Este tipo de lista se utiliza cuando se necesitan definir diferentes conceptos. Para realizar esto se utilizan los comandos <dl> (del inglés "*definition list*") que le indica al navegador que se va a iniciar este tipo de lista, <dt> (del inglés "*definition term*") para indicar el término a definir, <dd> (del inglés "*definition definition*") con el que indicamos la definición propiamente dicha, y el par </dl> para indicar el fin de la lista. El comando <dl> puede tomar la forma <dl compact> cuando la definición va a ser corta y se desea que vaya en la misma línea del término.

Ejemplo de una Estructura General de una Lista Definida:

```
<dl>
<dt>Término 1
<dd>Definición del Término 1
<dt>Término 2
</dl>
```

Para definir una lista primero se utilizará el comando <ul type=a> donde el parámetro type es opcional y sirve para definir el tipo de figura que se desea como marca; la que puede ser disc (disco), circle (círculo), o square (cuadro).

Para cada uno de los elementos que van a conformar la lista se debe utilizar el comando .

Se cuenta con la opción de que una lista este anidada o contenida dentro de otra lista, en este caso, si no se define el tipo de marca, el navegador automáticamente utilizará otro tipo para la lista anidada. Se debe tener mucho cuidado con el cierre de las listas , pues de otra forma el texto que continúe después de esto, se verá indentado con margen derecho.

Ejemplo:

```
Algunos estados de la República Mexicana son :
<ul>
<li>Baja California Sur
<li>Baja California Norte
<li>Sonora
<li>Coahuila
</ul>
```

4.3.1 Menú interactivo <MENU>

Son listas de elementos o párrafos cortos, sin viñetas ni números, ni ningún otro tipo de marca.

Las listas de menú están rodeadas por las etiquetas <MENU> y </MENU> y cada elemento que formará parte de él tendrá que ir acompañado con la etiqueta . El estilo de la lista de menú es más compacto que el de una lista no ordenada.

4.3.2 Lista de Directorio <DIR>

El elemento DIR es similar al elemento . Representa una lista de términos cortos, generalmente hasta de 20 caracteres cada uno. Los términos en una lista de directorio pueden ser arreglados en columnas.

4.4. Mapas de Imágenes

Por medio de los mapas de imágenes, se pueden realizar figuras con varias hiperligas, pues cada conjunto de coordenadas (x,y) representa una hiperliga distinta.

Los mapas de imágenes son una forma especial de los CGI (ver el punto 4.10). Cuando el navegador activa el vínculo de imágenes, se llama a un programa especial de mapa de imágenes como lo haría cualquier otro programa CGI, pero a su vez incluye coordenadas X e Y del punto en que se indique con el ratón. El programa especial de mapa de imágenes revisa un archivo de mapa que iguala las regiones de la imagen con archivos, hace algunos cálculos para averiguar que página debe de "cargar".

Los elementos fundamentales para trabajar con mapas de imágenes son:

- Código especial HTML para indicar que una imagen es un mapa
- Un archivo de mapa en el servidor, que indica las regiones de la imagen y las páginas WWW a las que apuntan
- Un programa CGI de mapa de imágenes que vincule todo lo anterior

4.5 Formas HTML

Las **Formas** son la más reciente innovación adicionada en el HTML. En ellas el autor del documento WWW presenta al lector una serie de campos de captura, donde se pide información. Además las formas contienen otros tipos de captura de información, como son los radio buttons (botones circulares de elección única), los check buttons (botones cuadrados de elección múltiple) y los menús de lista hacia abajo. Permitiendo a través de éstas la implementación de nuevas aplicaciones dentro del WWW.

El rango de usos de las formas proveen sofisticadas interfaces para la búsqueda o solicitud a través de la máquina, como el registro en línea y aplicaciones interactivas donde la entrada proporcionada por la forma determinará la naturaleza y el contenido de lo desplegado en el documento. Pueden ser utilizadas para cuestionarios, reservaciones de hotel, formas de pedidos, ingreso de datos y una amplia variedad de aplicaciones.

Este proceso de ingresar datos es usualmente manejado por un script o programa escrito en Perl o en cualquier otro lenguaje que maneje texto y archivos de información.

Un conjunto de datos pertenecientes a una forma están representados por una secuencia de pares de campos conocidos éstos como: pares nombre/valor.

Los nombres son especificados en los atributos NAME de los elementos de entrada de la forma, y los valores son valores iniciales otorgados por varias formas de marcación y editados por el usuario.

Las formas WWW permiten al usuario intercambiar información con el servidor de WWW para realizar alguna acción.

Las formas por sí solas no son difíciles de codificar, siguen las mismas reglas de construcción que otras etiquetas HTML, lo que posiblemente resulte complicado es el programa o script que toma la información recibida de la forma y la procese.

Los elementos de las formas pueden ser mezclados con elementos estructurales del documento, por ejemplo, un elemento PRE puede contener un elemento FORM, o un elemento FORM puede contener listas, las cuales contienen elementos INPUT. Esto brinda una flexibilidad considerable en el diseño de los distintos esquemas de las formas.

4.5.1 Elementos de una Forma

Una forma consta principalmente de:

1. El comando `<FORM ACTION=" ... ">`
2. El cuerpo de la forma o formulario, consistente en texto y una serie de campos de ingreso
3. Los botones para envío del formulario y el correspondiente al de limpieza de forma
4. El comando `</FORM>`

HTML 3.0 soporta los siguientes tipos de campos:

- Campos de texto simple
- Campos de texto multilíneas
- Radio buttons
- Checkboxes
- Menús de opción sencilla/múltiple
- Botones de orden para enviar el contenido de una forma
- Botones de reset para restaurar los campos a sus valores iniciales
- Campos ocultos

Los cuatro primeros tipos de ingreso de información, tienen asociado un NOMBRE, que es obligatorio asignar, pues si no se hace de esta manera no habrá modo de referirse a la información que el usuario ha ingresado.

Se espera que futuras versiones de HTML añadan soporte para campos de audio, entradas para tablas de bases de datos y campos más extendidos para texto multilineas para soportar un rango de tipos de datos más amplio.

Cabe mencionar que cada forma debe ser contenida de un solo elemento <FORM>. Existe la posibilidad de que haya varias formas en un documento, pero el elemento <FORM> no puede estar anidado. El navegador es responsable de manejar el foco de entrada, esto es, cuál campo obtendrá la entrada del teclado.

4.5.2 Forma <FORM>

El elemento FORM contiene una secuencia de elementos de entrada, junto con los elementos estructurales del documento.

Los atributos son :

- **ACTION**
El atributo ACTION es un URL que especifica el lugar al cual el contenido de la forma es enviado para recibir una respuesta, si ACTION no esta presente, el URL del documento es asumido. La forma en que los datos son enviados varía con el protocolo de acceso del URL y con los valores de los atributos METHOD y ENCTYPE.
- **METHOD**
Este atributo especifica variaciones en el protocolo utilizado para enviar el contenido de la forma, esta restringido a 2 valores GET (el valor por default) o POST.
- **ENCTYPE**
Especifica el medio utilizado para codificar los pares nombre/valor para su transporte.
- **SCRIPT**
El lenguaje del script y la interfaz con el navegador no es parte de la especificación HTML 3.0. El navegador carga el script y lo interpreta localmente. Los scripts manejan una variedad de mensajes para campos individuales y la forma como un todo. Estos mensajes corresponden a eventos tales como:
 - Forma de Entrada/Salida
 - Cuando un campo gana o pierde el foco de entrada
 - Clicks del mouse y arrastres sobre un campo
 - Eventos de teclado
- **Campo de entrada <INPUT>**
El elemento INPUT representa un campo de entrada cuyo contenido el usuario modificará.

El atributo TYPE establece el método que el usuario deberá utilizar para dicha modificación.

- **Campo de Texto <INPUT TYPE=TEXT>**

El valor por default del atributo TYPE es 'TEXT', indicando un campo de entrada de una sola línea, cuyo contenido es modificado a través del teclado.

Ejemplo: <INPUT NAME="Nombre" TYPE="Text">

Los atributos requeridos son:

- **NAME**

Es el nombre del campo de la forma correspondiente a este elemento.

Los atributos opcionales son:

- **MAXLENGTH**

Restringe el número de caracteres que pueden ser ingresados en un campo de entrada de texto. Si el valor de MAXLENGTH es mayor que el valor del atributo SIZE, el campo se recorrerá apropiadamente, el número por default de caracteres es ilimitado.

- **SIZE**

Especifica la cantidad de espacio de despliegue asignado a este campo de entrada de acuerdo a su tipo. El valor por default depende del navegador.

- **VALUE**

Indica el valor inicial del campo.

- **Campo de Contraseña <INPUT TYPE=PASSWORD>**

Es un campo de texto, excepto porque el valor es ocultado conforme es ingresado, normalmente por un asterisco. El usuario puede ver cuántos caracteres han sido tecleados pero no lo que fue tecleado.

- **Check Box <INPUT TYPE=CHECKBOX>**

Representa una elección booleana. Este comando permite seleccionar varias opciones a la vez, por medio de pequeños cuadritos que aparecen al costado del texto, y se pueden seleccionar o deseleccionar de manera independiente, haciendo click sobre ellos.

Un checkbox se define como: <INPUT TYPE="checkbox" NAME="nombre">

Ejemplo :

```
<P>Algunos Deportes : </P>
<INPUT TYPE="checkbox" NAME="deporte_tenis">Tenis
<INPUT TYPE="checkbox" NAME="deporte_golf">Golf
<INPUT TYPE="checkbox" NAME="deporte_natacion">Natación
<INPUT TYPE="checkbox" NAME="deporte_squash">Squash
```

Los atributos requeridos son:

- **NAME**

Representa el nombre simbólico para el campo del elemento o grupo de elementos

– **VALUE**

La porción del valor del campo contribuida por este elemento

Los atributos opcionales son:

– **CHECKED**

Indica que el estado inicial es habilitado

– **Radio Button <INPUT TYPE=RADIO>**

Adecuado para atributos que pueden tomar un valor entre un conjunto de alternativas.

Los botones dentro del mismo grupo deben tener el mismo nombre, solo el botón seleccionado en el grupo genera un par nombre/valor en los datos enviados. Se requieren los atributos NAME y VALUE, para inicializar el botón en su estado de seleccionado se debe incluir el atributo CHECKED.

Este comando nos permite ingresos de una opción de entre varias. Esto se define mediante el comando: `<INPUT TYPE="radio" NAME="nombre" VALUE="valor">`, los cuales comparten el mismo nombre pero tienen distintos valores. Al momento del envío de la forma lo realmente enviado es el nombre, y el valor de la opción que fue elegida.

Ejemplo :

```
<P>Sexo : </P>
<INPUT TYPE="radio" NAME="sexo" VALUE="masc">Masculino <BR>
<INPUT TYPE="radio" NAME="sexo" VALUE="fem">Femenino <BR>
```

La opción que aparecerá seleccionada se marca por default dándole el atributo CHECKED.

Ejemplo:

```
<INPUT TYPE="radio" NAME="sexo" VALUE="fem" CHECKED>Femenino
```

Si no se especifica ninguna entonces aparecerá seleccionada la primera opción de entre las múltiples que compartan el mismo valor del atributo NAME.

– **Campo Oculto <INPUT TYPE=HIDDEN>**

Representa un campo que esta oculto. El usuario no interactúa con este campo, en su lugar, el atributo VALUE especifica el valor del campo. Se requieren los atributos NAME y VALUE, el contenido del campo es enviado junto con la forma. Este valor puede ser usado para transmitir información acerca de la interacción cliente/servidor, por ejemplo un identificador de transacción.

Estos campos son necesarios porque los servidores HTTP no conservan información de estado de una petición a la siguiente.

• **Botón de Envío <INPUT TYPE=SUBMIT>**

Representa una opción de entrada, típicamente un botón que instruye al navegador para enviar la forma.

Se declara utilizando:

```
<INPUT TYPE="submit" VALUE="texto">
```

donde "texto" es lo que aparecerá en la cara del botón.

Ejemplo :

```
<input type="submit" value="Enviar comentarios">
```

Los atributos opcionales son.

- **NAME.** Indica que este elemento contribuye con un campo de forma, cuyo valor es dado por el atributo VALUE. Si el atributo NAME no esta presente, este elemento no contribuye con un campo de forma.
- **VALUE.** Indica una etiqueta para la entrada (botón).
- **Botón de Reset <INPUT TYPE=RESET>.** Representa una opción de entrada, normalmente un botón, que permite al usuario restaurar los campos de la forma a su estado original. El atributo VALUE si esta presente indica una etiqueta para la entrada (botón).

Se declara utilizando el siguiente comando:

```
<INPUT TYPE="reset" VALUE="texto">
```

donde "texto" es lo que aparecerá en la cara del botón.

Ejemplo :

```
<input type="reset" value="Limpiar formulario">
```

- **Selección <SELECT>.** El elemento SELECT restringe al campo de la forma a una lista numerada de valores, el usuario puede escoger uno o varios a la vez. Los valores son dados en los elementos OPTION. Este elemento esta representado por un menú pull-down, el termino desplegado por default es el marcado como Selected si es que hay alguno, si no, entonces se toma el primero.

Un menú desplegable consta de:

1. Una declaración de comienzo del menú: <SELECT NAME="nombre">
2. Una serie de opciones: <OPTION VALUE="valor">
3. Una declaración de termino de menú desplegable: </SELECTED>

Ejemplo :

```
<SELECT NAME="sexo">  
<OPTION VALUE="indefinido">Elija uno de los siguientes:  
<OPTION VALUE="masc">Masculino  
<OPTION VALUE="fem">Femenino  
</SELECT>
```

Los atributos son:

- **MULTIPLE.** Indica que más de una opción puede ser incluida en el valor. En ausencia de este atributo, el elemento define un menú de opción única.
- **NAME.** Especifica el nombre del campo de forma.
- **SIZE.** Especifica el número de términos visibles. Campos SELECT de tamaño unitario son representados por menús pop-down, mientras que los campos SELECT con tamaño mayor que uno son típicamente listas.
- **Opción <OPTION>.** El elemento OPTION puede ocurrir sólo dentro de un elemento SELECT. Este representa una opción y cuenta con los siguientes atributos:
 - **Selección SELECTED**
Indica que esta opción es inicialmente seleccionada.
 - **VALUE**
Indica el valor que debe ser regresado si esta opción es elegida
El contenido del elemento OPTION es presentado al usuario para la opción.
Es usado como un valor de regreso si el atributo VALUE no está presente.

- **Área de Texto <TEXTAREA>**

El elemento TEXTAREA representa un campo de texto multilínea. El texto que está antes de la etiqueta de fin es usado para inicializar el valor del campo. El texto de inicialización puede contener entidades SGML, por ejemplo, caracteres acentuados, pero de otra manera es tratado como texto sin acentuación. Esta etiqueta final es siempre requerida incluso si el campo está inicialmente vacío. Cuando se envía una forma, los terminadores de línea son independientes de la implementación.

En una presentación típica, los atributos ROWS y COLS determinan la dimensión visible del campo en caracteres.

Ejemplo:

```
<TEXTAREA NAME="nombre" ROWS="líneas" COLS="columnas"> ... </TEXTAREA>
```

4.6. Imágenes

Las imágenes que se pueden colocar en un documento WWW pueden ser todas aquellas que cuenten con formatos X-Bitmap (aquellos con extensión .xbm), GIF (con extensión .gif) o del tipo JPEG (con extensión .jpg).

Para crear imágenes que cuenten con los formatos permitidos podemos hacer uso de programas de diseño gráfico, como pueden ser Corel Draw, Aldus Photostyler, Adobe PhotoShop, etc. Se recomienda utilizar imágenes con formato .GIF, ya que ocupan menor cantidad de bytes, haciendo más rápido el despliegue de la imagen. También es importante el definir correctamente el tamaño de la imagen en píxeles. Otra manera de obtener imágenes e

insertarlas en la página WWW es por medio de los scanners, que en su mayoría cuentan con los programas que manejan los formatos permitidos.

En caso de existir algún problema con la imagen se desplegará un cuadro pequeño con un signo de interrogación. Una posible causa de este tipo de problema es que el navegador no encuentre el archivo donde se localiza la imagen.

Otros atributos comúnmente utilizados son: align, height, y width, de acuerdo al formato:

```
<img src=n align=x height=a width=b>
```

donde los nuevos parámetros se refieren a: la **x** indica la alineación deseada para la imagen y puede ser left, o right, la **a** y la **b** indican la altura y el ancho respectivos en pixeles del área donde se desplegará la imagen. Si sólo se especifica el ancho o el alto, el otro parámetro será calculado para mantener el aspecto original de la imagen.

También se puede hacer la imagen más alta, más angosta, más ancha, menos alta, poniendo un ancho más grande que la altura. Si no se especifica el tamaño de la imagen, entonces se utiliza el tamaño con el cual fue creada en el programa de diseño o escaneada.

Si se define la alineación solo se podrán colocar las líneas que alcancen a ser insertadas en el espacio que deje libre la imagen ya sea a su derecha o izquierda.

Si no se define la alineación sólo podrá poner una línea de texto junto a la imagen.

Si se desea centrar una imagen se debe hacer con el comando <center> y </center>, pero si se realiza esto, no se debe poner texto junto a la imagen, ya que se centraría texto e imagen como un todo.

Ejemplo:

Imagen con parámetros por default

Imagen a la derecha y texto a la izquierda

Imagen centrada <center></center>

Imagen deformada

- **Address <ADDRESS>**

El comando Address generalmente se debe colocar al final del documento pero dentro del comando <BODY>.

En el comando address se escribe el nombre del autor del documento, la organización a la que pertenece, su dirección de correo electrónico, y otra información que se considere importante como puede ser la última fecha de actualización del documento.

Lo anterior sirve como referencia para quienes consultan la página HTML.

4.7 Tablas <TABLE>

Una tabla es una forma de mostrar información de manera más compacta. Para HTML una tabla es un conjunto de filas (filas=horizontal), apiladas una sobre otra, donde cada fila contiene a su vez un conjunto de celdas, colocadas una al lado de la otra.

Una tabla se declara usando las etiquetas <Table> ... </Table>. Dentro de la tabla, se utiliza <TR> ... </TR> para delimitar el contenido de una celda, y las etiquetas <TD> ... </TD> definen una celda perteneciente a una tabla.

Los comandos anteriores aceptan varios atributos, se pueden entre otras cosas, unir celdas, especificar la alineación de la celda, insertar imágenes en las celdas, etc.

Ejemplo:

```
<TABLE>
<TR><TD>Celda A1</TD><TD>Celda B1</TD><TD>celda C1</TD></TR>
<TR><TD>Celda A2</TD><TD>Celda B2</TD><TD>celda C2</TD></TR>
</TABLE>
```

Caracteres Especiales de código

Para poder incluir caracteres especiales en un documento HTML, se utiliza el comando: &código, donde código, corresponde al código interno del caracter especial

De los caracteres especiales, generalmente los más utilizados son los que permiten colocar acentos y la letra Ñ.

Algunos ejemplos se muestran en la tabla 4.2:

Código	Caracter
á	á
é	é
í	í
ó	ó
ú	ú
ñ	n
&Nacute	Ñ

Tabla 4-2

Existe una lista muy variada de caracteres especiales, símbolos matemáticos, caracteres extendidos, etc.

- **FRAMES**

<FRAME> ...</FRAME>

La etiqueta FRAME especifica un solo marco (caja) en un conjunto de marcos (cajas).

Antes de llevar la implementación de frames en una página, se debe estudiar la forma de división de la página, decidir que sector (es) contendrá (an) el (los) índice(s) fijo(s) y que sector(es) mostrará(n) la información variable.

Algunos de sus atributos:

- **SCROLLING = yes|no|auto**

El atributo SCROLLING define si el marco tendrá una barra de desplazamiento, por omisión es "auto", es decir, lo tendrá en caso de ser necesaria.

- **NORESIZE**

La presencia del atributo NORESIZE previene que el marco sea cambiado de tamaño por el usuario.

- **NAME=" ... "**

El elemento NAME asigna un nombre al marco para ser usado como objetivo de las ligas.

- **MARGINWIDTH= pixeles**

Ancho del margen.

- **MARGINHEIGHT= pixeles**

Altura del margen.

- **FRAMEBORDER= yes|no**

Denota la presencia o ausencia de un borde alrededor del FRAME.

- **BORDERCOLOR= #rrggbb**

Denota el color del borde de un FRAME.

- **SECTORIZACION**

La sectorización (uso de frames) de la página se realiza con los comandos **FRAMESET** y **FRAME**.

- **FRAMESET**

El elemento FRAMESET se usa en vez del elemento BODY. Se usa en un documento HTML cuyo único propósito es dar definición a las características de los subdocumentos HTML o marcos (FRAME) que formarán la página. Es decir presentar la información ordenada dentro de una página dividida.

Atributos:

- **ROWS = número, número, número, ...**

Es una lista separada por comas que describe la altura de cada fila.

- **COLS = número, número, número, ...**

Es una lista separada por comas que describe el ancho de cada fila.

- **FRAMEBORDER = yes|no**

Indica la presencia o ausencia de un borde alrededor de los diferentes FRAMES. El valor por default es YES.

- **BORDER = pixeles**

Este comando puede configurar globalmente el grosor del borde de todos los FRAMES en un FRAMESET.

- **BORDERCOLOR = #rrggbb**
Denota el color del borde de un FRAME.

El comando FRAMESET acepta varias formas de definir la disposición de los FRAMES en su interior, algunos ejemplos son:

FRAMESET rows="20%,80%" Porcentaje
FRAMESET rows="20%,*" Equivale al anterior, pero el * se utiliza para indicar todo el resto
FRAMESET rows="30,*" Píxeles
FRAMESET rows="30*,30" No necesariamente tienen que ser dos frames solamente

División Típica

```
<FRAMESET cols="30%,70%">  
<FRAME NAME="indice" SRC="temas.f.html">  
<FRAME NAME="informacion" SRC="info.f.html">  
</FRAMESET>
```

División en dos mitades horizontales:

```
<FRAMESET rows="50%,50%">  
<FRAME NAME="superior" SRC="sup.f.html">  
<FRAME NAME="inferio" SRC="inf.f.html">  
</FRAMESET>
```

División Mixta:

```
<FRAMESET rows="20%,80%">  
<FRAME NAME="titulo" SRC="titulo.f.html" SCROLL="no">  
<FRAMESET="30%,70%">  
<FRAME NAME="indice" SRC="temas.f.html">  
</FRAMESET>  
</FRAMESET>
```

En este ejemplo el frame superior carece de barras de desplazamiento, debido al atributo Scroll=no.

4.8 Editores y Convertidores

Los editores y convertidores son programas utilizados como herramientas que permiten convertir a formato HTML los archivos creados en procesadores de texto comerciales o en otros formatos.

La diferencia entre un editor y un convertidor, es que el primero permite escribir los documentos en HTML insertando cada una de las etiquetas del formato HTML, mientras que los segundos proporcionan métodos abreviados para poder crear archivos HTML.

4.8.1 Editores

Los editores de etiquetas están constituidos por una serie de botones y ventanas que permiten la inserción de las etiquetas correspondientes, en el texto del documento.

Una de las características de los editores de etiquetas en su desempeño es más efectivo cuando se tienen los documentos a convertir ya terminados, porque el ir insertando cada una de las etiquetas conforme se va escribiendo el texto normal es un tarea difícil, por ello se recomienda escribir primero el texto y después aplicar las etiquetas para la creación de los estilos deseados dentro del formato HTML.

- **Para Microsoft Windows.** Si se hace uso de Windows existen varios editores de etiquetas, a continuación se mencionan editores más comunes:
 - 1 **HTML Assistant.** Es considerado como uno de los editores de etiquetas más antiguos, permite insertarlas por medio de barras de herramientas, además de permitir observar el resultado de cada una de estas inserciones con alguno de los navegadores existentes. Una de sus principales características es su capacidad de agrupar URL's de listas importantes, además de permitir crear vínculos con estos URL's, por lo que nunca será necesario escribir un URL dentro del documento HTML.
 - 2 **HotDog .** HotDog de Sausage Software, es un editor de etiquetas que cuenta con todo el soporte para casi cada una de las funciones HTML, incluyendo tablas, formularios, las extensiones de Netscape, etc. Una desventaja que presenta HotDog es que trata de dar soporte a todo, por lo que crea confusión en las etiquetas que genera.
 - 3 **HTML Easy .** Como su nombre lo indica es muy sencillo de utilizar, soporta HTML 2.0, las extensiones de Netscape además de tablas y formularios. No soporta la inserción de las etiquetas <HTML> y </HTML> lo que hace necesario que sean insertadas manualmente además de realizar un click por medio del mouse para poder escribir el contenido de la etiqueta.
 - 4 **WWWEdit .** Su objetivo principal de este editor es el de dar soporte a las etiquetas que integran a HTML 2.0 ,HTML 3.0 y Netscape. Por lo tanto se cuenta con un número de etiquetas muy elevado, que para realizar documentos con formato HTML sencillos lo hace inútil, debido a la gama tan variada de etiquetas con que cuenta, provocando que todas aquellas funciones adicionales que presenta estorben. Siempre y cuando se deseen crear documentos sencillos se considera como una herramienta poco útil, pues si son más elaborados entonces si es una herramienta muy versátil.
- **Editores WYSIWYG.** El concepto de editores WYSIWYG (del inglés, "*What You See Is What You Get*") es en realidad poco útil, ya que cada navegador da formato a los documentos HTML de manera distinta, y la mayoría de ellos permite al usuario configurar para usar aquellas fuentes de su predilección, es por esto que no se aplica con la veracidad que corresponde el concepto de editor WYSIWYG.
- **Para Microsoft Windows** Para poder editar sobre Windows HTML casi en un concepto WYSIWYG existen dos opciones: cualquiera de las plantillas para Microsoft Word en sus versiones 2.0 o 6.0 y SoftQuad HotMetal.

Existen 2 paquetes de plantillas conocidos:

- 1 **CU_HTML.** Permite asignar distintos formatos a cada uno de los párrafos con solo hacer una selección en ellos, hacer inserción de imágenes con extensión .GIF en

línea y especificar hiperligas seleccionando el archivo a vincular desde una caja de diálogo, donde las imágenes .GIF pueden ser vistas desde el mismo documento en Word. Utilizando esta plantilla, se pueden crear los documentos en formato HTML sin necesidad de tener amplios conocimientos de este lenguaje, además de permitir utilizar la barra de herramientas que pertenece a Word en lo que se refiere a su menú de edición, es decir poder hacer uso de las herramientas de cortar, pegar, copiar además de realizar Macros.

- 2 **ANT_HTML**. Fue diseñada para facilitar la creación de documentos de Hipertexto, las etiquetas de HTML pueden ser insertadas en cualquier documento nuevo o escrito en el procesador de texto Word, o en formato ASCII. Esta plantilla incluye herramientas para soportar las extensiones de Netscape además de una barra especial, propia de la plantilla para mantener visibles las etiquetas más utilizadas en la generación de documentos con formato HTML. Provee de otra herramienta que facilita la creación de formularios o formas. Convierte caracteres especiales definidos por el procesador. Permite la creación de archivos HTML desde formatos con extensión .DOC, ASCII, RFT, y otros, además de mandar a impresión los documentos HTML generados sin ser un requisito indispensable el contar con HTML. Cuenta con una herramienta que convierte las tablas creadas en Word a tablas en formato HTML, sin necesidad de crearla nuevamente. La generación de los archivos en el concepto WYSIWYG facilita la impresión. En cuanto a la barra de herramientas que se habilita en el momento de ser instalada la plantilla puede soportar modificaciones para establecer las marcas HTML que serán utilizadas con mayor frecuencia en el momento de querer generar un documento HTML. Proporciona herramientas para determinar cada una de las características de forma de una tabla como: ancho de bordes, espacio y tamaño de las celdas, dimensiones de ancho y largo en toda la tabla, al igual las etiquetas correspondientes a la tabla <TABLE> y la etiqueta de fin de tabla, son insertadas automáticamente rodeando a cada tabla durante el proceso de conversión. También ofrece varias opciones para poder seleccionar colores, gráficos que podrán ser utilizados como background (fondo del documento), así mismo la selección de colores se pueden aplicar a ligas y textos, ofrece opciones para controlar la alineación, y otros formatos de texto, aunque respeta, entiende y convierte los que fueron establecidos por Word. Presenta cajas de diálogo para guiar al usuario acerca de como se va llevando el proceso de creación del documento HTML paso a paso.

- **HTML Editor**. Es una aplicación que permite insertar las etiquetas en el archivo y ver el resultado en forma WYSIWYG al mismo tiempo. Una característica es la muestra de las etiquetas en un color más claro que el texto que se encuentra alrededor. Cuenta con la opción de ocultar las etiquetas en el documento para lograr un efecto completo, al igual que permite ver el resultado con algún navegador.
- **Para UNIX**. Existen dos herramientas para UNIX dentro del concepto WYSIWYG : SoftQuad HoTMetaL y tkWWW.
 - **SoftQuad HoTMetaL**. Cuenta con las mismas características que el editor para Windows, y también está disponible para muchas estaciones de trabajo UNIX que corran X11 (Motif).
 - **TkWWW**. tkWWW es un editor además de un navegador World Wide WWW que se ejecuta en UNIX utilizando el lenguaje tcl y el kit de herramientas tk. Es considerado como WYSIWYG por ser editor y navegador a la vez, siempre y

cuando sólo utilice tkWWW como navegador. Dentro de tkWWW se eligen cada uno de los elementos y estilos por medio de menús, y al momento de guardarse el documento se produce un documento en formato HTML. Al utilizar tkWWW se asegura que nunca se verá una etiqueta HTML en el documento.

4.8.2 Convertidores

A continuación se nombran las plantillas más comunes para *Microsoft Word* y *WordPerfect*.

- **Microsoft Word.** Para Microsoft Word, se pueden utilizar las plantillas CU_HTML, ANT_HTML, y GT_HTML, al igual que Microsoft Assistant, también Microsoft Word puede exportar archivos en RTF (Formato de Texto Enriquecido), que posteriormente podrán convertirse en formato HTML a través del filtro RTFTOHTML.
- **WordPerfect.** Un convertidor para documentos escritos en WordPerfect es WPTOHTML, está formado por un conjunto de macros de WordPerfect que realizan la conversión a HTML a los archivos en WordPerfect 5.1 y 6.0 para DOS. También exporta archivos RTF, lo que permite usar el convertidor RTF.

Algunas ventajas de utilizar un convertidor de HTML son :

- Al estar escribiendo un documento el autor, no tiene que memorizar las reglas y las etiquetas del lenguaje HTML, al aplicar el convertidor al documento es tarea del propio convertidor.
- Los documentos HTML presentan un número menor de fallas, pues al ser generado automáticamente el documento HTML se reducen las posibilidades de error del autor.

Algunas desventajas de utilizar un convertidor de HTML son:

- Ninguna herramienta proporciona todas las características de HTML, en especial a lo referente a vínculos con documentos externos. Generalmente es necesaria la intervención manual del autor para dar por terminada la generación del documento HTML.
- Existe el problema de la fuente dividida. Una vez que se ha convertido el archivo a formato HTML, se cuenta con dos fuentes, el original y el ya formateado, así pues para realizar cambios posteriores a la conversión se tendrá que modificar el original y volver a generar el archivo HTML, perdiendo todo trabajo manual realizado, o en su defecto realizar los cambios en ambos archivos, lo anterior es muy problemático en documentos muy grandes.

4.9 CGI (Common Gateway Interface)

Un CGI es un programa que funciona como una interfaz entre un servidor con Protocolo de Transferencia de Hipertexto y los demás recursos del servidor.

En teoría un CGI sirve para poder extender la capacidad de cada uno de los servidores de WWW, y así poder interpretar una entrada desde el navegador y regresar como respuesta información basada en la entrada enviada por el usuario. Un CGI es una interfaz que habilita al programador para crear programas con lo que sea posible mantener una comunicación sencilla con el servidor.

Los CGI no tienen que ser precisamente programas, dependiendo del soporte con que cuenta el servidor de WWW, pueden también ser programas compilados, archivos por lote o cualquier otra entidad ejecutable.

CGI no es en realidad un lenguaje o un protocolo, en el sentido más estricto del término. En realidad sólo es un conjunto de variables y convenciones, para pasar información en ambos sentidos entre el servidor y el cliente.

4.9.1 Requerimientos

Para poder escribir y ejecutar programas CGI, se necesita de un servidor de WWW, el cual debe estar configurado para ejecutar programas CGI.

Generalmente todos los programas CGI están almacenados en un directorio especial llamado *CGI-BIN*, y hay que solicitar autorización al administrador del servidor para poder ejecutar programas CGI en el directorio, pues por cuestiones de seguridad este servicio debe ser requerido y autorizado por el administrador del mismo servidor.

Para escribir programas CGI se puede utilizar prácticamente cualquier lenguaje de programación, en tanto dicho lenguaje se pueda ejecutar en el sistema en que "corra" el servidor WWW. Sin embargo, algunos servidores sólo tienen soporte para programas escritos en cierto lenguaje determinado.

Los únicos criterios que deben considerarse para el lenguaje son los siguientes:

- El lenguaje debe estar soportado por el sistema operativo en el que se ejecute el servidor.
- El lenguaje debe tener las facilidades necesarias para realizar tareas que se requieran de él
- El programador debe estar lo suficientemente familiarizado con el lenguaje para trabajar de manera eficiente dentro de él.

Cada vez que el cliente hace una petición al URL correspondiente al CGI, el servidor ejecutará el programa, la salida de éste se presentará directamente al cliente.

El CGI define los parámetros abstractos, conocidos como variables de ambiente, las cuales describen la petición del cliente.

Generalmente los programas CGI toman su entrada de variables de ambiente y envían su salida al flujo de salida estándar, comúnmente nombrada como: stdout. Denominando *flujo* a esta salida, debido a que su transmisión hacia el usuario es un flujo de bytes o caracteres.

Esta salida es encontrada en formato HTML estándar, no sin antes el programa determinar el tipo de datos con los que está trabajando. Lo anterior se logra con la siguiente instrucción:

Content-Type: <MIME-type>

Content-type especifica el tipo de MIME (del inglés "*Multipurpose Internet Mail Extensions*") del flujo que se enviará al usuario, la mayoría de las ocasiones el valor que tendrá es "text/html" para documentos HTML, y "text-plain" para documentos de tipo informativo.

Durante el desarrollo de un programa CGI se debe de considerar que los encabezados MIME deben aparecer con un espacio en blanco como separación, del cuerpo del documento.

Las funciones básicas de un programa CGI de entrada/salida son:

1. Reunir la entrada del servidor, en forma de variables estandarizadas, donde los datos pueden provenir de formas o pueden ser datos de consulta.
2. Proporcionar datos de salida al cliente, en este caso el navegador de WWW.
3. Brindar la información de negociación del contenido (el encabezado MIME) tanto para el servidor como para el cliente.

La primera salida de un programa CGI siempre debe ser el encabezado MIME ("Content-type: text/html", por ejemplo). El encabezado MIME debe ser seguido de dos líneas nuevas (en blanco).

La forma de insertar las dos líneas nuevas es enviar dos veces la cadena "\n" dentro del código correspondiente.

Para ejecutar un programa CGI se necesita colocar el programa dentro de un directorio y en algunas ocasiones colocarle una extensión especial de nombre de archivo. Al igual se deberá de asegurar de que el programa pueda ser leído por el servidor WWW. Esto significa que se deben de ajustar los permisos de archivo de modo que los usuarios puedan leer y/o ejecutar el archivo.

En los programas CGI se pasa la mayor parte de la entrada en *variables de ambiente* que son establecidas por el servidor de HTTP, algunas de estas variables son estándares de la especificación CGI, otras son específicas de navegadores, servidores y sitios particulares, así como de otros factores.

Cuando se llama a un programa CGI ya están establecidas numerosas variables para dar información acerca del usuario, de la configuración del servidor, y del software instalado.

Las variables más utilizadas para dar información acerca del usuario son:

HTTP_USER_AGENT: Contiene el nombre y la versión del navegador del usuario, en el formato nombre/versión biblioteca/versión.

HTTP_ACCEPT: Proporciona los formatos MIME que puede aceptar el navegador, el formato de la cadena es tipo/subtipo, tipo/subtipo,...etc.

Las siguientes variables de ambiente proporcionan información acerca del software y el servidor: REMOTE_HOST y REMOTE_ADDR.

Las siguientes variables de ambiente proporcionan información acerca del software el servidor:

SERVER_SOFTWARE: Contiene el nombre y la versión del software del servidor, en el formato nombre/versión.

SERVER_NAME: Contiene el nombre del host del servidor, los alias DNS o la dirección IP, que son utilizadas para la construcción de URL's auto-referenciados.

GATEWAY_INTERFACE: Contiene las versiones de la especificación CGI que utiliza el servidor, en el formato CGI/versión.

Las siguientes variables son específicas de solicitud:

QUERY_STRING: Es la variable más utilizada para el paso de información a un programa CGI. Por lo general la solicitud a un programa CGI se realiza por medio de un signo de interrogación "?", seguido de la información adicional en el URL. Esto es todos los caracteres que se encuentren después del signo de interrogación serán colocados en la variable QUERY_STRING.

La variable QUERY_STRING nunca se codifica ni decodifica antes de que la obtenga el programa CGI. Está garantizado que será exactamente como se pasa al servidor HTTP.

SERVER_PORT: Representa el puerto al cual llegó la solicitud. Esto puede ser importante si el programa está dando servicio a solicitudes que lleguen a diferentes puertos, quizá para diferentes dominios o servicios. Este campo casi siempre es el valor "80", el puerto estándar de las solicitudes HTTP.

4.9.2 Procesamiento de Formularios (Formas)

Las formas o formularios son la manera más común de llevar datos del usuario a un programa CGI. Ofreciendo la posibilidad de entrada al usuario mediante diversos tipos de elementos, como cuadros de texto, botones clasificados y gráficos.

Cuando elaboramos formas o formularios dentro de páginas WWW, podemos proporcionarle al usuario una interfaz sencilla para que la llene con cualquier tipo de información que nosotros le solicitemos.

La variable de ambiente QUERY_STRING tiene un formato de flujo de información por pares, donde estos pares están formados por secuencias **name/value** separados por el carácter &. Cada par de name/value está codificado en URL, esto es, los espacios en blanco son cambiados por caracteres "+" y algunos caracteres son codificados hexadecimalmente.

El procedimiento básicamente consiste en separar los datos guiándose por los símbolos &, entonces, para cada par de name/value que se obtiene, se debe codificar el nombre y el valor para poder trabajar sobre ellos.

4.9.3 Datos de Entrada al Programa CGI

Hay dos métodos que pueden ser utilizados para acceder a las formas o formularios de HTML.

Estos métodos son GET y POST, dependiendo del método que se utilice, se recibirán los resultados codificados de manera diferente.

- **El método GET.** Si la forma tiene declarado METHOD=GET en su etiqueta FORM, el programa CGI recibirá la entrada codificada de la forma en la variable de ambiente QUERY_STRING y la variable REQUEST_METHOD contendrá la palabra GET.

- **El método POST.** Si la forma tiene METHOD=POST en su etiqueta FORM, el programa CGI recibirá la entrada codificada de la forma en "entrada estándar".

Habrán al menos CONTENT_LENGTH bytes disponibles para que el programa pueda leer, el programa no está obligado a leer los datos, pero no debe intentar leer más de CONTENT_LENGTH bytes incluso si hay más datos disponibles.

El servidor no está obligado a enviar "fin de archivo" después de que el programa lea CONTENT_LENGTH bytes. Es decir no se garantiza que el flujo de entrada termine con un indicador de fin de archivo.

Esto significa que se debe leer exactamente el número de bytes incluidos en CONTENT_LENGTH y nada más. Si se trata de leer hasta fin de archivo, quizá nunca se termine.

- **REQUEST_METHOD.** Se refiere al método con el cual la petición fue hecha. Para HTTP ésta es GET, HEAD, POST. El método no es sensitivo de mayúsculas y minúsculas, es decir, no hace diferencia entre ellas.
- **QUERY_STRING.** Es la información que sigue al símbolo "?" en el URL que referencia al programa. Esta es la información de consulta, es decir, lo que el usuario quiere buscar en la base de datos o quizás los resultados codificados de la forma GET de retroalimentación, ésta información puede ser añadida ya sea por un documento HTML ISINDEX o por una forma HTML (con la acción GET). También puede ser incrustada manualmente en un ancla HTML que referencie al programa CGI. Esta cadena es codificada en el formato estándar URL, se cambian los espacios por "+" y caracteres especiales con codificación hexadecimal %xx. Se necesitará decodificar la variable para poder ser usada, si el programa CGI no está decodificando los resultados de una forma HTML, también se podrá obtener la cadena de consulta decodificada en la línea de comandos, esto significa que cada palabra de la cadena de consulta estará en una sección diferente del argumento ARGV. Esta variable siempre debe estar establecida cuando hay información de consulta, sin importar la decodificación de la línea de comandos.

De acuerdo a lo anterior es prácticamente posible codificar todo, donde cada caracter tiene un significado propio para diferentes sistemas, por lo que, aun si un caracter no parece necesitar codificación en el sistema que se utilice, probablemente llegué codificado.

Descifrar los datos del formulario es cuestión de dividir las partes *name* y *value* de las cadenas, y reemplazar después cada uno de los indicadores que haya dentro de la cadena por los datos textuales.

Como se ha visto durante este capítulo el lenguaje HTML es de gran utilidad en el desarrollo de las diferentes aplicaciones dentro del WWW, al trabajar de manera conjunta con un servidor, así como con cada una de las aplicaciones con que disponen estos servidores, lográndose una serie de beneficios en el intercambio y procesamiento de información entre usuarios y servidores, al poder generar interfaces rápidas y hasta cierto punto "sencillas".

En el siguiente capítulo se fundamenta el uso de cada una de las herramientas empleadas para el desarrollo del SIPT, así como los resultados obtenidos.

Capítulo 5

Proyecto Tabasco

5.1 Elección de herramientas

A continuación se describirán las herramientas utilizadas para el desarrollo del proyecto.

5.1.1 Servidor WWW

Instalación y operación del servidor WWW en la UNAM

Netscape Enterprise Server, resultó ser el servidor WWW más adecuado de los que se tenían a disposición de acuerdo a las necesidades del proyecto, por lo que se instaló, configuró y puso en operación. Aunque éste no es un servidor de dominio público, la UNAM, como institución educativa, puede utilizarlo de manera gratuita (para mayor referencia del proceso de instalación, configuración y operación del servidor, ver el Apéndice B).

Una vez que se instaló el servidor, se consulta la página del servicio administrativo por medio de la dirección electrónica:

<http://tabasco.iingen.unam.mx:8100>

Para consultar el servicio administrativo del servidor Enterprise Server, se requiere de un nombre de usuario y contraseña (asignados por el administrador del servidor durante el proceso de instalación), por lo que cuando se accesa a esta página aparecerá un cuadro de diálogo para anotar los datos requeridos como se muestra en la figura 5-1.

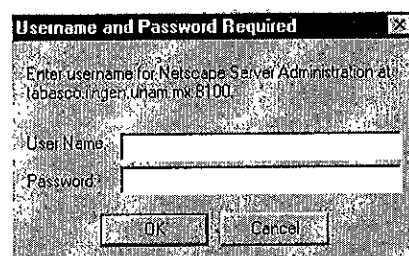


Figura 5-1

Una vez que se han tecleado correctamente el nombre de usuario y la contraseña correspondiente, se mostrará una pantalla inicial de configuración del servidor (figura 5-2).

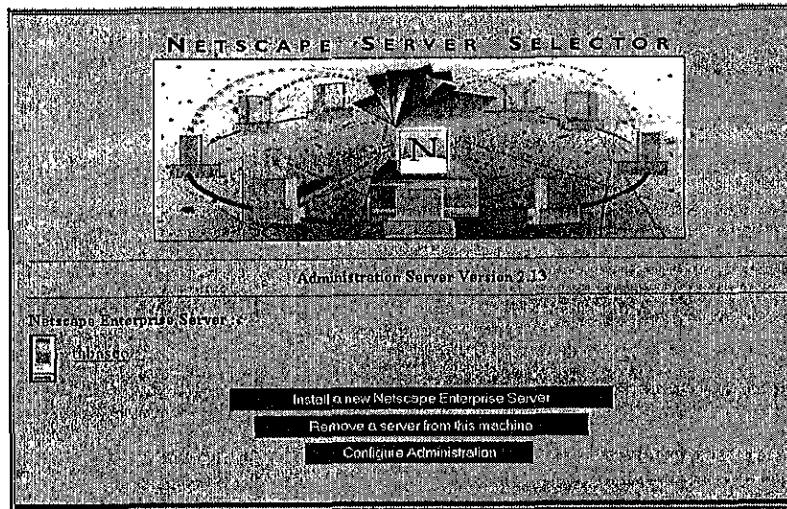


Figura 5-2

La administración del sistema WWW se lleva a cabo por medio de un servidor de administración que es proporcionado en conjunto con el servidor WWW Enterprise Server (el proceso de administración del servidor se ve con mayor detalle en el Apéndice B).

Entre los requerimientos del proyecto se encontraba el permitir un acceso limitado a la información disponible en el servidor, por lo que se habilitó el servidor Enterprise Server para la utilización del protocolo SSL (protocolo que utiliza Enterprise Server para hacer uso de la seguridad de la información). Para habilitar el protocolo se requirió la generación de un certificado para el servidor, mismo que se generó con un programa disponible en Internet llamado *cm10nti.exe*.

Cabe mencionar que para iniciar con la seguridad requerida para la información almacenada, se recomienda dirigirse al botón Configuración de Administración (*Configure Administration*) de la pantalla del servicio de administración (figura 5-2) y una vez que se ha elegido este botón, se elige la opción Usuario de Administración (*User to run as*), en este apartado, se muestra la cuenta de usuario con la que se está ejecutando el servicio administrativo del servidor WWW (Enterprise Server), por omisión no se encuentra activado SSL (por esto se accesa como *http*) pero por seguridad se recomienda activarlo mediante la elección de la opción Activar SSL (*Activate SSL*).

La utilización de SSL es de importancia ya que asegurará el envío de los datos para la configuración del servidor, por esto se anotará *https* en lugar de *http*, por lo que la nueva dirección electrónica será:

<https://tabasco.iingen.unam.mx:8100>

5.1.2 Creación de Grupos de Usuarios

Entre las opciones que conforman Enterprise Server, existe una para el manejo y control de los usuarios, de manera que, habilitado el servidor para uso del protocolo, se procedió a crear los grupos en *Control de Acceso*¹. A continuación se enumeran los grupos de usuarios creados:

1. AIRE_Y MATERIALES
2. GENERAL
3. SISTEMAS_ACUATICOS
4. SISTEMAS_BIOLOGICOS
5. SISTEMAS_TERRESTRES
6. SISTEMA_DE_INFORMACION

Posteriormente, se llevó a cabo el dar de alta a los usuarios pertenecientes a cada grupo, cabe mencionar que se creó un grupo *GENERAL* debido a que a él pertenecería el personal que se encontraba realizando pruebas del proyecto y que debían tener autorización de acceso a todas las páginas.

5.1.3 Operación del Servidor

Enterprise Server es el servidor WWW que actualmente se encuentra en operación en la UNAM y se encuentra habilitado para hacer uso del protocolo SSL, puede consultarse en la siguiente dirección electrónica:

<https://tabasco.iingen.unam.mx:9000>

Cuando se accesa a la página, se despliegan en pantalla cuadros de diálogo (figuras 5-3 y 5-4) que informan acerca del certificado que presenta el servidor, éstos presentan información tal como fecha de creación y de expiración, organización a la que pertenece y tipo de encriptación de la que hace uso.

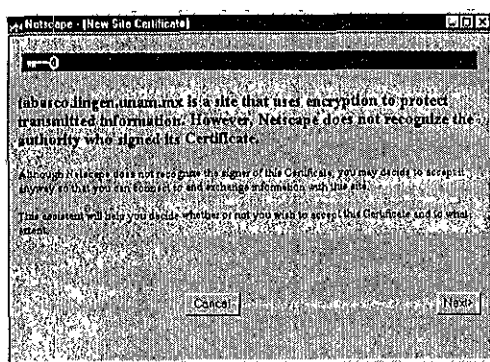


Figura 5-3

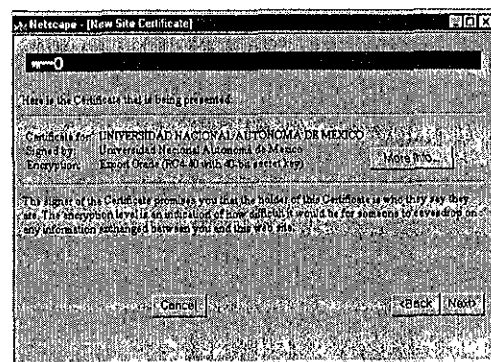


Figura 5-4

En el cuadro de diálogo (figura 5-5) existe una opción en la que podemos indicar que <Acepte este certificado> (*Accept this ID forever (until expires)*) para todas las sesiones posteriores que se establezcan al servidor desde la computadora en uso, en caso de ser

¹ Herramientas generales de Enterprise Server, se manejarán con letra negrita cursiva

aceptado, no se mostrarán más estos cuadros de diálogo en sesiones de conexión posteriores dando por hecho que se aceptó el certificado presentado.

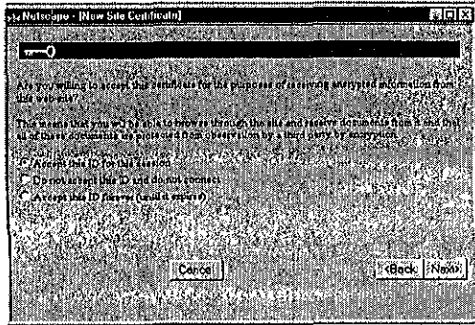


Figura 5-5

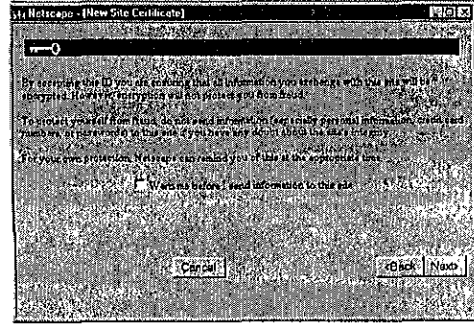


Figura 5-6

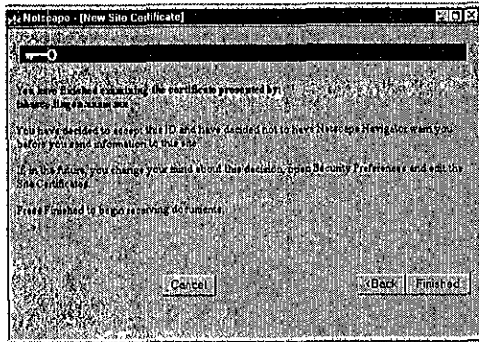


Figura 5-7

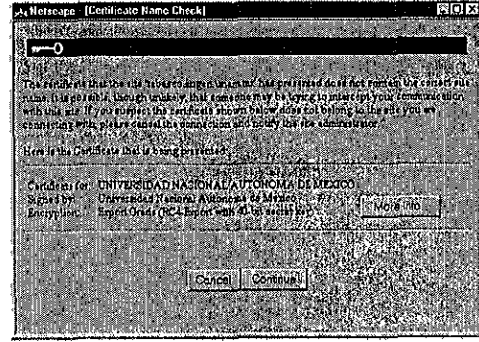


Figura 5-8

Además de presentarse estos cuadros informativos (figuras 5-6, 5-7 y 5-8), dependiendo del navegador utilizado, se presenta un cuadro en el que se informa que se encuentra consultando una página con seguridad, éste se presenta en la figura 5-9.

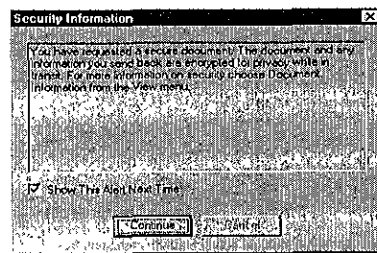


Figura 5-9

5.1.4 Diseño de las páginas de acceso

Finalmente se unificaron las páginas elaboradas por el IMP y la UNAM (figura 5-10), para mostrar los resultados se presentó una página en la que se presentan las ligas de ARCHIVOS (correspondiente al IMP) e HIPERTEXTO (correspondiente a la UNAM).



Figura 5-10

Una vez que se consulte la sección correspondiente a la UNAM, se presenta una página de inicio, como podemos ver en la figura 5-11, se presenta un índice general y una sección de despliegue principal.

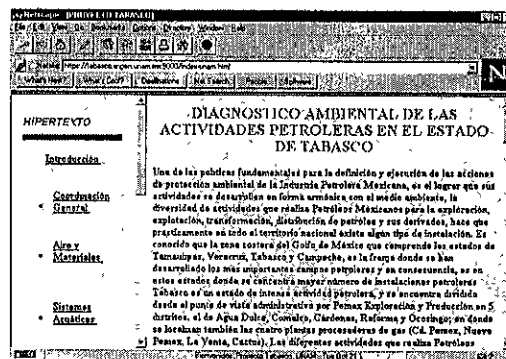


Figura 5-11

El servidor implantado permite consultar de dos maneras posibles la documentación proporcionada por los grupos:

- Por medio de acceso directo al archivo fuente. Para ello el usuario debe tener instalado Word, Excel y en algunos casos Power Point. El navegador, de manera automática abrirá la aplicación de acuerdo con la extensión del documento consultado (.doc, .xls o .ppt) y cargará el documento en cuestión. La principal desventaja de este mecanismo es que quien desee saber de qué trata el documento tendrá que recibir todo el documento y hasta entonces podrá consultarlo. Además, se requiere tener instalada la aplicación con que se creó el documento.
- El otro mecanismo de acceso a la información es a través de páginas HTML, las cuales pueden ser consultadas únicamente por medio de un navegador, Netscape o similar. Este método permite consultar los documentos de manera directa sin requerir la aplicación con que se generó el documento original

Una vez creada la página de inicio del servidor, se elaboraron diversas páginas en HTML para la presentación de los documentos, mismas que se crearon con las siguientes ligas de acuerdo a los grupos que conforman el proyecto:

1. Introducción
2. Coordinación General
3. Aire y Materiales
4. Sistemas Acuáticos
5. Sistemas Terrestres
6. Sistemas Biológicos
7. Sistema de Información
8. Búsquedas
9. SOMOD
10. Mail

En la liga de *Introducción* se muestra en el panel principal una breve explicación del proyecto.

En *Coordinación General*, se muestra una página con una lista del personal que conformó el proyecto.

Para las 6 siguientes ligas, se procedió a generar una página de información general para el tema correspondiente, sin embargo, de entre los requisitos del proyecto (restricción de acceso a la información), se configuró el servidor para que a partir de estas ligas, se hiciera presente un cuadro de diálogo en el que pide un nombre de usuario (*login*) y una contraseña (*password*), mismo que se presenta en al figura 5-12.

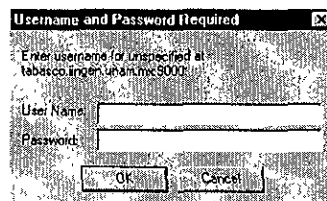


Figura 5-12

Una vez que se proporcionan los datos correctos, se presenta una página principal del tema correspondiente a la liga, en la figura 5-13 se muestra un ejemplo.

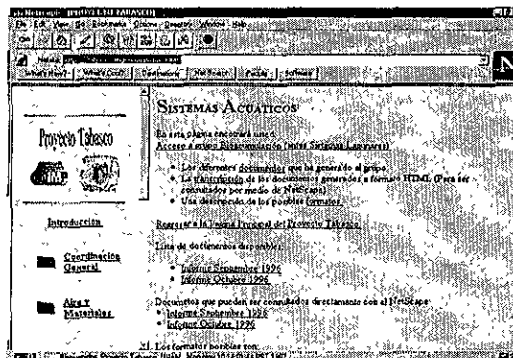


Figura 5-13

La restricción de la información de acceso a los usuarios, se relacionó conforme al grupo de información del que provenían.

Finalmente, se presenta una sección con el “Sistema de Búsquedas”, esta herramienta se desarrolló con *Búsquedas*.

Para inicializar este servicio, desde la página de administración del servidor, se declaró el directorio:

unidad_local : \directorio_de_servidor\plugins\search\vsearch

que en nuestro caso es:

c : \enterp\plugins\search\vsearch

como “directorio CGI”, con esto inicializamos el servicio de búsquedas, posteriormente se crearon las *Colecciones* que conjuntan los archivos en los que se llevarían a cabo las búsquedas. Las colecciones que se formaron para el sistema de búsqueda se realizaron en base a los grupos de usuarios existentes, por lo que se generó una colección por grupo, misma que se presenta en la tabla 5-1.

Colección	Descripción
Aire y Materiales	Contiene todos los archivos de información del grupo de Aire y Materiales.
Sistema de Información	Contiene todos los archivos de información del grupo “Sistema de Información”
Sistemas Acuáticos	Contiene todos los archivos de información del grupo “Sistemas Acuáticos” y “Bioacumulación”
Sistemas Terrestres	Contiene todos los archivos de información del grupo “Sistemas Terrestres” y “Monitoreo Edafocológico”
Sistemas Biológicos	Contiene todos los archivos de información del grupo “Sistemas Biológicos”
General	Contiene todos los archivos de los grupos.

Tabla 5-1

El sistema está enfocado a realizar búsquedas en archivos con formato *html*, esto de acuerdo a los requerimientos del proyecto.

Una vez que se llevaron a cabo los procedimientos de instalación y configuración de éste, se diseñó una página principal para que los usuarios tuviesen derecho a su uso (figura 5-14).

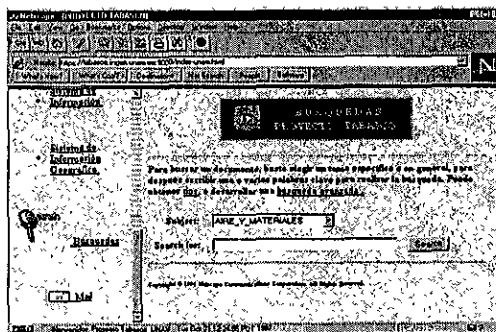


Figura 5-14

En esta página se presenta adicionalmente una ayuda para el manejo del mismo además del uso de esta misma con las utilerías que ofrece para realizar una búsqueda con mayor precisión.

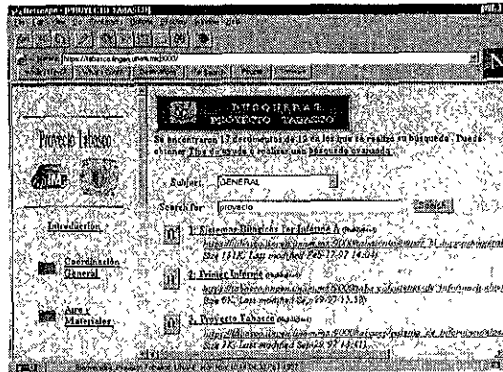


Figura 5-15

Como se puede ver en la figura 5-15, se presenta en el mismo *frame* de la página una lista con los resultados de la búsqueda, ésta contiene un listado con las ligas relacionadas a nuestra búsqueda, así mismo, se hace presente un ícono (a un lado de la liga) en el que se puede obtener el mismo texto de la liga pero con la característica que la palabra tecleada en la búsqueda se encuentra resaltada (negrillas).

5.1.5 Mecanismos de seguridad

Como se mencionó anteriormente, el acceso a la información se lleva a cabo por medio de permisos por lo que, para poder obtener información se requiere un nombre y contraseña de usuario además de tener derechos de lectura sobre la información requerida, todo esto otorgado por el servicio de administración del servidor. Una vez que se cuenta con lo anteriormente mencionado, y se accesa a la página proporcionando los datos correctos, se llevará a cabo la petición, sin embargo cuando se ha proporcionado algún dato incorrectamente se desplegará la página que a continuación se muestra en la figura 5-16.

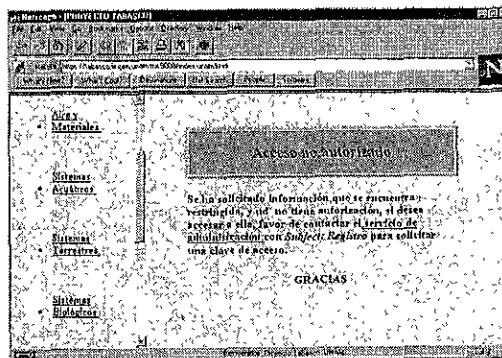


Figura 5-16

Cabe mencionar que, comúnmente se despliega una página en blanco, pero para dar mayor referencia a los usuarios, se optó por configurar el servidor para que enviase páginas con el mayor detalle posible de los resultados a su petición. De manera que cuando se proporcionen

nombre de usuario y contraseña correctos pero no se tenga autorización de lectura en la página requerida se desplegará en la página mostrada en la figura 5-17:

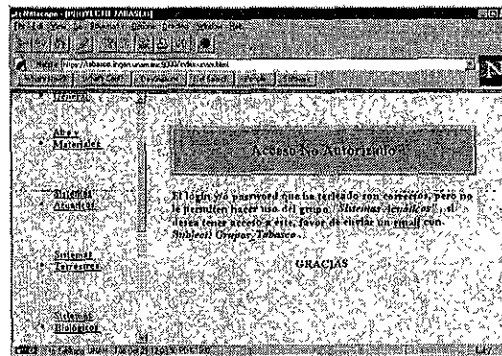


Figura 5-17

Podemos agregar que cuando se realice algún acceso para requerir el índice de algún directorio, se desplegará la siguiente página que se muestra en la figura 5-18.

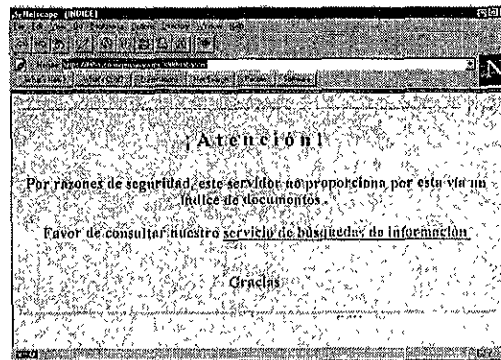


Figura 5-18

Para esto fue necesario elaborar la página e indicar en el archivo de configuración *obj.conf* el nombre de la página que se desea desplegar (Para mayor detalle ver el Apéndice B).

5.1.6 Empleo de la plantilla ANT_HTML.DOT

Se utilizó el lenguaje HTML para el desarrollo del sistema por ser un mecanismo de acceso de información rápido y útil para la consulta de los documentos que conforman al sistema. Inicialmente se consideró tener cada uno de los documentos generados por cada grupo disponible dentro del ambiente de Internet de manera que el usuario pudiera “bajarlos” a espacio en disco, lo anterior representa una alternativa para fines de consulta, sin embargo esto a su vez tiene algunas desventajas, como es, el contar con el software de cada una de las distintas aplicaciones en que se generó el documento necesario dentro de la computadora donde va a ser recibido, para poder consultar parte del documento es necesario como primer paso exportar todo el documento para después llevar a cabo la consulta, y por último el espacio libre en disco con el que se debe contar para recibir al documento de consulta.

Lo anterior funciona para el formato original de la información, pero cabe mencionar que esto es posible siempre y cuando cada uno de los documentos originales sea “convertidos” al formato HTML estándar, a través de convertidores de HTML. Siendo entonces necesario tener

una herramienta que realice esto, de forma rápida, dinámica, y transparente para el usuario final, lo anterior debe de ser posible para cualquier tipo de documento soportado por cualquier servidor WWW y cualquier navegador. Después de llevar a cabo una investigación de cada una de las posibles herramientas para convertir documentos al formato estándar de HTML se decidió utilizar el convertidor Ant_html.Dot (ver capítulo 4) para documentos en Word e Internet Assistant para hojas de cálculo en Excel.

La investigación para establecer la herramienta a utilizar, se realizó en dos etapas:

- Determinación de las herramientas disponibles en el medio, para lo cual se realizó una búsqueda a través de Internet, a la vez que se seleccionaron las opciones más cercanas a las necesidades de conversión establecidas de acuerdo con los formatos manejados por los grupos en sus informes (en general formatos de MS-Office)
- Prueba y Evaluación de las herramientas seleccionadas. Una vez que se obtuvieron los programas de demostración (en algunos casos) y las versiones Beta (en otros casos) de cada herramienta, se instalaron y configuraron, para posteriormente llevar a cabo pruebas de operación

La diferencia principal entre todas las herramientas se midieron en términos de número de formatos que soporta para su conversión. Así mismo, es necesario considerar otros detalles, tales como la facilidad de operación, costo, flexibilidad, etc.

A medida que se fueron realizando las pruebas de conversión con las diferentes aplicaciones, se establecieron algunos criterios de evaluación que ayudaron a identificar las ventajas de cada herramienta, así como sus desventajas. Dentro de estos criterios se consideraron:

- Compatibilidad con Windows 3.X
- Operación desde la aplicación de MS-Office
- Ayudas y Ejemplos
- Integración del navegador a la herramienta
- Proceso de conversión
- Conversión de gráficos

Además de permitir fijar características de color, formato de texto, listas, tablas, gráficos, etc.

Otro de los puntos por los que se decidió utilizar al lenguaje HTML en el desarrollo del sistema de información fue la flexibilidad para realizar solicitudes de información al servidor, donde esto es posible por medio de la creación de formas o formularios donde el usuario llenará y solicitará la información, una vez enviada la forma el servidor la procesa para finalmente enviar información de regreso al navegador, éste último despliega el documento recibido.

La publicación de documentos HTML, puede verse como el proceso en el cual se escribe algún material, se agregan algunos códigos especiales y se proporciona el documento al servidor WWW.

Las actividades que están involucradas en este proceso son:

- Authoring (Autoraje). Crear un archivo que contenga la información que se desea publicar. Esto se obtiene mediante la combinación de texto y código HTML. Actualmente existen una gran variedad de herramientas que permiten la generación de documentos HTML, muchas de ellas incluso, sin necesidad de requerir grandes conocimientos del estándar HTML. Por ejemplo es posible encontrar programas que conviertan documentos en Word a hojas HTML y viceversa, lo mismo con Excel o Power Point
- Scripting. Adicionar alguna funcionalidad para procesar el intercambio de información entre el cliente y el servidor. Estos scripts (o pequeños programas) tienen la capacidad de manejar las solicitudes de información más que simplemente la solicitud de un documento. Por ejemplo son capaces de recibir, procesar y responder a la información que un cliente proporciona al llenar una forma
- Administración. Efectuar actividades relacionadas con la verificación y el mantenimiento que aseguren que el sistema se encuentra funcionando adecuadamente y que no ha sido vulnerada su integridad, al igual se incluyen dentro de estas operaciones la incorporación de nuevos documentos
- Programación. Consiste en extender las funciones del servidor WWW y del navegador para proveer aquellos servicios que no fueron contemplados originalmente

5.2 SOMOD

A petición del IMP se empleó la herramienta SOMOD para el desarrollo del sistema, teniendo la función de almacenar y sistematizar todos los datos que se generaran por los grupos de trabajo.

SOMOD tiene sus ventajas y desventajas, algunas de ellas son las siguientes:

Ventajas:

1. El tamaño del registro es variable, lo cual parte con el esquema tradicional de la base de datos. Esto es con el fin de almacenar solamente los campos que tengan datos registrados, y omitir aquellos que no tengan información
2. En bases de datos de gran tamaño, el factor crítico es el tiempo de búsqueda de los registros. En SOMOD este se minimiza
3. SOMOD es directamente código fuente, y no intérprete. Esto redundará en una operación más precisa en cuanto a objetivos y reducción de ejecución de los programas
4. SOMOD no requiere de grandes requerimientos para incorporarse a cualquier programa de cómputo. Esto se refleja en la rapidez con la cual SOMOD se adapta a la tecnología WWW. Todas las bases de datos tradicionales (como SQL) requieren un "aditamento" adicional para incorporarlo al WWW
5. No hay que saber manejar SQL ni otro manejador de Base de Datos.
6. SOMOD es gratis (a menos que se desee comercializar)
7. SOMOD se puede ejecutar bajo cualquier plataforma

Desventajas:

1. Para poder trabajar con ésta herramienta, es necesario tener experiencia con el lenguaje C
2. No existen manuales de operación de SOMOD
3. No se pueden realizar gráficas directamente
4. No tiene la opción de imprimir reportes

Algunas de estas desventajas no son significativas, puesto que los resultados de SOMOD se pueden enviar a una base de datos u hoja de cálculo tradicional donde se pueden sacar gráficas y reportes.

De acuerdo a las necesidades de los grupos participantes del SIPT, se decidió dividir en grupos y subgrupos al sistema, éstos son:

- **Sistemas Biológicos**
 - Sistemas biológicos acuáticos flora
 - Sistemas biológicos acuáticos fauna
 - Sistemas biológicos terrestres flora
 - Sistemas biológicos terrestres fauna
- **Sistemas Lagunares**
 - Concentración de hidrocarburos aromáticos policíclicos en sedimentos, organismos y en forma disuelta/dispersa (agua)
 - Concentración de hidrocarburos alifáticos o saturados en organismos y sedimentos
 - Concentración y su biodisponibilidad de metales tóxicos en agua, sedimentos y organismos
 - Concentración de plaguicidas organoclorados en sedimentos y organismos
- **Sistemas Terrestres**
 - Agua subterránea
 - Zona vadosa
- **Sistemas Acuáticos**
 - Hidrología e hidrodinámica
 - Descargas residuales
 - Cuerpos receptores
- **Monitoreo Edafo-Ecológico**
 - Análisis de suelos
 - Análisis de TPH'S muestras de suelos
- **Aire y Materiales**
 - Monitoreo de corrosión
 - Monitoreo de fuentes fijas
 - Depositación seca
 - Calidad del aire
 - Monitoreo de lluvia ácida
 - Monitoreo de estaciones meteorológicas
 - Caracterización de corrientes a los quemadores
 - Venteos

Para poder acceder a los grupos y subgrupos antes mencionados, inicialmente se debe "conectar" a la página principal (ver Figura 5-19).

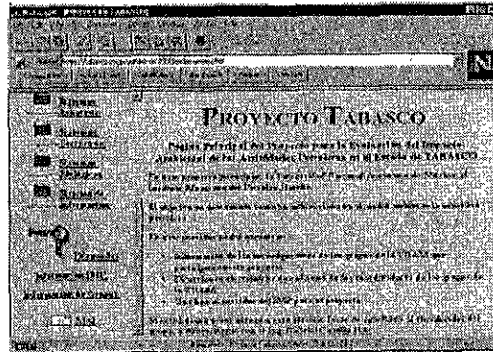


Figura 5-19

Posteriormente se selecciona la liga “*Información de Grupos*”, y la página que muestra es la que se aprecia en la figura 5-20.

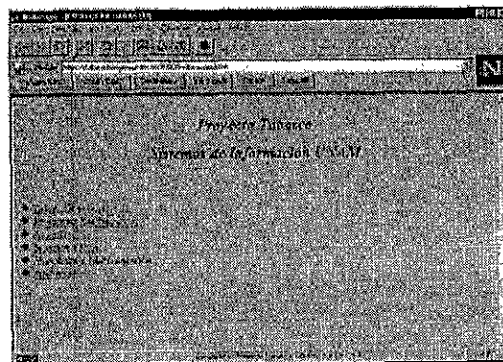


Figura 5-20

Esta pantalla despliega información de cada uno de los grupos que conforman el Proyecto Tabasco. Como los procesos son semejantes (Capturar Información, Consultar Información, Consultar toda la Información sobre...), se describirá al subgrupo “*Hidrología e Hidrodinámica*”.

Hidrología e Hidrodinámica

Al seleccionar el grupo “*Sistemas Acuáticos*”, aparece la forma [menu.htm](#) (figura 5-21). En esta forma se puede elegir una de tres opciones:

- Capturar información
- Consultar Información
- Consultar toda la Información sobre....

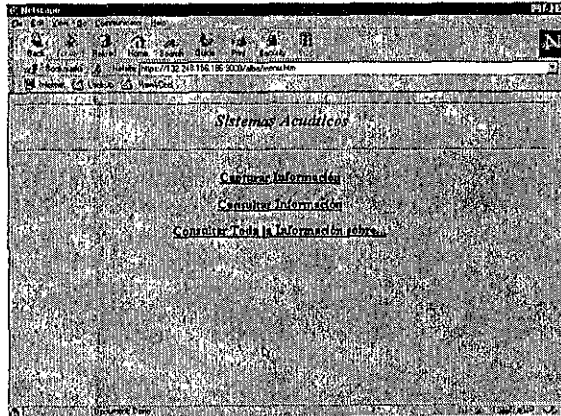


Figura 5-21

Si se elige “*Capturar Información*”, mostrará la forma [acuaticos.htm](#) (Figura 5-22). Esta forma a su vez tiene varias opciones:

- Aforos
- Campo de Hidrología e Hidrodinámica
- Hidrocarburos
- Calidad del agua.
- Campo de Descargas Residuales
- Análisis de Laboratorio de Descargas Residuales
- Campo de Cuerpos Receptores
- Capturar Análisis de Laboratorio de Cuerpos Receptores.

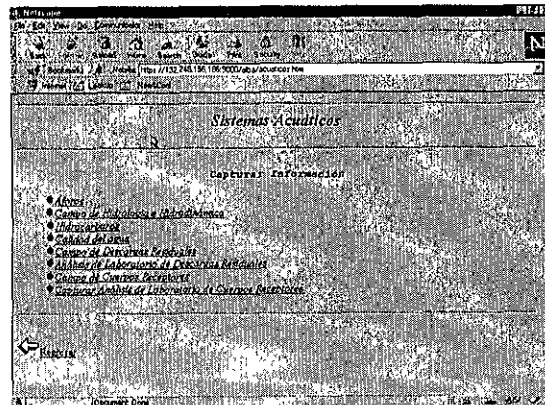


Figura 5-22

Finalmente, al seleccionar “*Campo de Hidrología e Hidrodinámica*”, envía la página [hidro.htm](#) (Figura 5-23), y es en esta forma donde se captura la información obtenida por los investigadores.

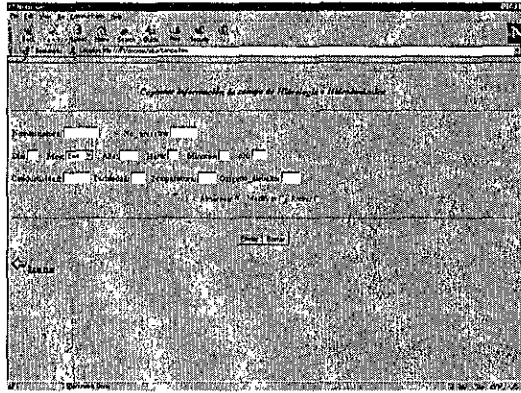


Figura 5-23

Los programas que están ligados a esta forma se aprecian en la Tabla 5-2:

Programa	Breve Explicación	Localización
Campo.htm	Forma que llama al CGI	F:\docroot\alba
*Campo.h	Definición de la estructura	D:\alba\
*Campo.c	Rutinas empleadas	D:\alba\
*Put_camp.c	Programa para registrar información	D:\alba\
Put_camp.exe	Programa ejecutable (CGI)	F:\cgi_bin\tabasco\

Tabla 5-2

Es recomendable que la información la capture el usuario, porque ésta se encuentra dispersa en varios archivos. Algunas cosas que se validan al insertar la información son:

- Que la llave no esté duplicada
- Que la forma tenga todos los campos llenos

En caso de que no cumpla con alguno de estos detalles, se enviarán mensajes de error para que el usuario pueda corregir los datos.

Una vez que se han llenan los campos correctamente, se presiona “Enviar” y los datos serán almacenados en el archivo aforos.dat con ayuda de SOMOD.

Si en la forma menu.htm, selecciona “Consultar Información”, desplegará la forma acuaticos2.htm (ver Apéndice A), en donde se podrá elegir una de las siguientes ligas (ver Figura 5-24):

- Aforos
- Campo de Hidrología e Hidrodinámica
- Hidrocarburos
- Calidad del agua
- Campo de Descargas Residuales
- Análisis de Laboratorio de Descargas Residuales
- Campo de Cuerpos Receptores
- Análisis de Laboratorio de Cuerpos Receptores

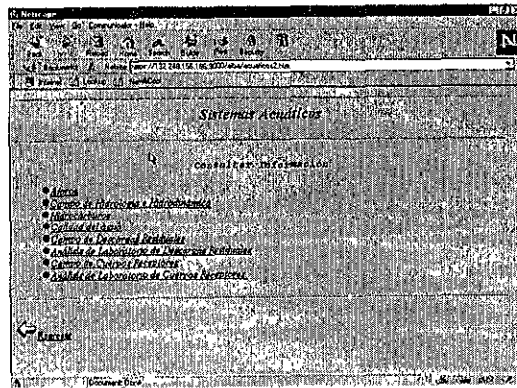


Figura 5-24

Si elige la liga "Campo de Hidrología e Hidrodinámica", presentará la pantalla de la Figura 5-25:

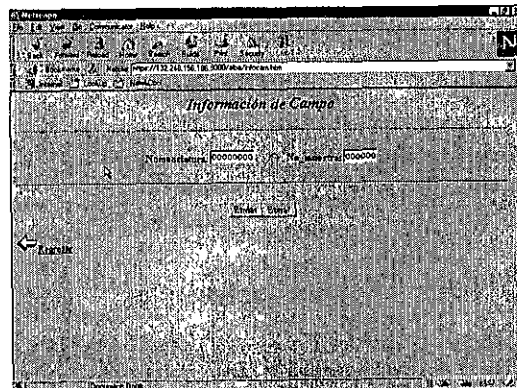


Figura 5-25

Aquí se pedirán dos datos para buscar la información: Nomenclatura y No_muestra (que conforman la llave).

Si la llave se encontró, desplegará la información de esa llave (ver Figura 5-26), en caso contrario enviará un mensaje de que la llave no existe.

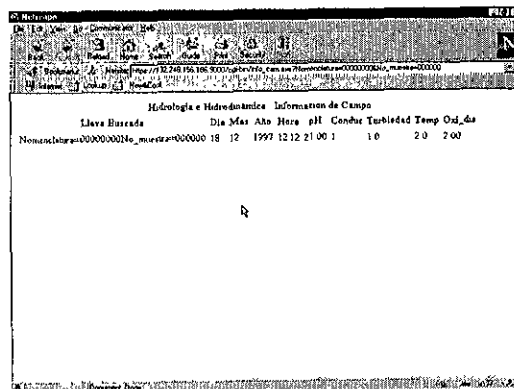


Figura 5-26

Los programas relacionados con esta forma se aprecian en la Tabla 5-3:

Programa	Breve Explicación	Localización
Infocam.htm	Forma que llama al CGI	F:\docroot\alba
*Campo.h	Definición de la estructura	D:\alba\
*Campo.c	Rutinas empleadas	D:\alba\
*Info_cam.c	Programa para buscar la información	D:\alba\
Info_cam.exe	Programa ejecutable (CGI)	F:\cgi-bin\tabasco

Tabla 5-3

Por último está la opción “Consultar Toda la Información sobre...” en la pantalla *menu.htm*. Esta opción nos llevará a la forma *acuáticos3.htm* (figura 5-27) y a su vez tendrá las siguientes opciones:

- Aforos
- Campo de Hidrología e Hidrodinámica
- Hidrocarburos
- Calidad del agua.
- Campo de Descargas Residuales
- Análisis de Laboratorio de Descargas Residuales
- Campo de Cuerpos Receptores
- Análisis de Laboratorio de Cuerpos Receptores

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

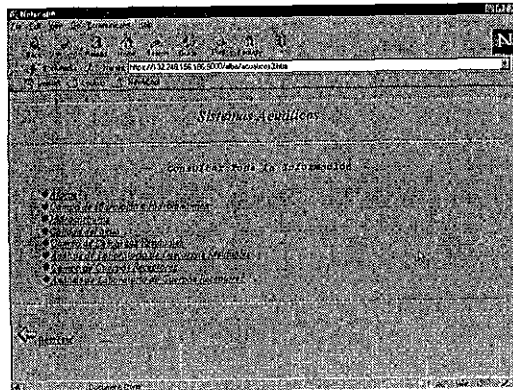


Figura 5-27

Si el archivo existe (por lo menos tiene una llave almacenada), nos desplegará la información (ver figura 5-28), en caso contrario nos enviará un mensaje de error.

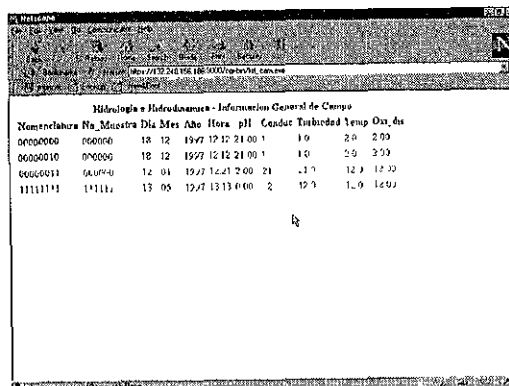


Figura 5-28

Los programas relacionados con esta forma se muestran en la Tabla 5-4:

Programa	Breve Explicación	Localización
Acuaticos3.htm	Forma que llama al CGI	F:\docroot\alba
*Campo.h	Definición de la estructura	D:\alba\
*Campo.c	Rutinas empleadas	D:\alba\
*List cam.c	Programa para listar la información	D:\alba\
List cam.exe	Programa ejecutable (CGI)	F:\cgi-bin\tabasco\

Tabla 5-4

Los programas marcados por un asterisco (*) pueden consultarse en el Apéndice A.

Finalmente se pueden consultar los Comentarios Finales que se derivaron de este proyecto.

Comentarios Finales

A continuación se describen los resultados obtenidos al elaborar este sistema, así como el alcance logrado.

Al desarrollar el sistema sobre una arquitectura cliente/servidor, lenguaje HTML y la implementación de la aplicación SOMOD se cumplió el objetivo principal al permitir almacenar, procesar y transmitir la información en diferentes áreas geográficas, pues se puede consultar este sistema vía WWW, contando con la veracidad de que las medidas de seguridad especificadas se cumplieron gracias a las características de Internet.

Cabe mencionar que si se hubiera utilizado un manejador de Bases de Datos para la manipulación de la información generada en el proyecto, probablemente el tiempo de programación empleado hubiera sido menor.

Manejar el lenguaje HTML permitió que el acceso al sistema se lograra de forma remota, con lo cual se asegura que el sistema en un futuro podrá contar con acciones de consulta más completas por ser un lenguaje de fácil aprendizaje no sin antes olvidar que si se lleva a cabo una modificación de cualquier hoja HTML el programa CGI correspondiente debe reflejar el mismo cambio.

El auge que ha tenido todo lo referente al procesamiento de información y las comunicaciones, así como su explotación y las necesidades de intercambio rápido de información ha llevado al surgimiento de nuevas formas de implementación de sistemas explotando todos estos recursos.

El WWW, ofrece la posibilidad de “comunicarse” mediante el lenguaje común a él, HTML, en cuanto a éste, se han ido desarrollando herramientas que permiten elaborar documentos de una manera fácil motivando así su uso.

Dadas las grandes posibilidades de conexión vía Internet, diversas instituciones educativas, gubernamentales y de comercio, se han ido sumando a su estructura, sin embargo, esa misma facilidad de conexión también la hace propensa a que se pueda interceptar y alterar la información que viaja a través de ella, dados los requerimientos que se deben cumplir ante ciertas compañías por mantener su información confidencial, se ha ido incrementando el número de desarrolladores de software en herramientas que de alguna manera garanticen cierto nivel de seguridad, es por esto que cada día se generan sistemas de encriptación que controlan la seguridad de la información.

Finalmente, se puede decir que para mantener la seguridad de la información se deben combinar las herramientas desarrolladas para seguridad en Internet además de la correcta administración del sistema en el que se encuentre la información.

Bibliografía

Libros:

Aprendiendo TCP/IP en 14 días

Timothy Parker, Ph. D

Ed. Prentice Hall

Segunda edición 1996

TCP/IP running a successful network

Kevin Washburn, Jim Evans

Ed. Addison Wesley

Segunda edición 1996

The World Wide Web Unleashed 2nd. Edition

John December, Neil Randall

Edit. Sams NET

Administrator Survival Guide Web Site

Jerry Ablan, Scott Yanoss

Edit. Sams NET

HTML 3.2 Soluciones Instantáneas

Robert Mullen

Prentice Hall Hispanoamericana, S.A.

El libro de CGI

William E. Weinman

Prentice Hall Hispanoamericana, S.A.

Aprendiendo HTML 3.0 para Web en una semana

Laura Lemay

Prentice Hall Hispanoamericana , S.A.

Completly Revised and Updated Netscape, Java, ActiveX, Internet Explorer HTML 3.2

John December

Prentice Hall Hispanoamericana, S.A.

HTML Source Book Third Edition

A complete guide to HTML 3.2 and Extensions

Ian S. Graham

Prentice Hall Hispanoamericana, S.A.

HTML 3 Manual of Style

Larry Aronson
Prentice Hall Hispanoamericana, S.A.

Master CGI Programming of the Internet

CGI Developer's Guide
Eugene Eric Kim
Sams NET

Special Edition

Using CGI

Jeffrey Dwight and Michael Erwin
Prentice Hall Hispanoamericana, S.A.

Direcciones Electrónicas:

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out.forms>

<http://robot0.ge.uiuc.edu/~carlosp/cs317/ft.1.html>

<http://www.infi.net/wwwimages/colorindex.html>

<http://www.boutell.com/gd>

<http://www.uni-essen.de/www/selfhtml/frames.htm>

<http://www.dic.uchile.cl/~manual/>

<http://www.hoofoo.ncsa.uiuc.edu/cgi>

<http://www.sdsu.edu/cgi-bin/genstats.pl>

<http://www-zeus.co.uk/products/server/intro/>

<http://www.lookup.com/hompages/57205/htmleng.html>

<http://www.microsoft.com>

<http://www.microsoft.com/msword>

<http://www.microsoft.com/msexel>

<http://www.microsoft.com/mspowerpoint>

http://ftp.mcia.com/pub/windows/ant/ant_demo.zip

http://ftp.mcia.com/pub/windows/ant/ant_demo.exe

<http://info.isoc.org/guest/zakon/Internet/History/HIT.html>

<http://www.ocean.ic.net/ftp/doc/nethist.html>

<http://ds.internic.net/ds/dspg0intdoc.html>

<http://www.civila.com/desenredada/ventajas.html>

<http://www.learnthenet.com>

Revistas:

PC-Computing

Los mejores 42 productos del año

Enero 1998

Glosario

ANCHO DE BANDA. Rango de frecuencias que se pueden transmitir en un canal.

ARPA (*Advanced Research Project Agency*). Agencia que fundó el gobierno federal de E.U., inicialmente su labor abarcaba la investigación, posteriormente, además de dedicarse a ésta también lo hizo en el área militar.

AGENTE. Software que se ejecuta para dispositivos de redes como ruteadores.

AUTENTIFICACIÓN. Permite a los clientes asegurarse de que la conexión se realiza a un servidor "seguro", previniendo que otra computadora pretenda ser el servidor.

BACKBONE (*Columna vertebral*). Conjunto de nodos conectados y comprendidos en una red. Algunas veces se refiere al medio físico de la red.

BITNET (*Because It's Time Network*). Red de correo electrónico.

BROWSER. Software utilizado para peticiones de información WWW, también se le conoce como "programa cliente" o cliente.

CERN. Laboratorio de Física de Partículas. Fue el desarrollador del World Wide Web, buscando construir un sistema de hipertexto e hipermedia. Actualmente la iniciativa en el desarrollo, especificaciones y software pertenece al consorcio W3 (W3 Consortium). Consorcio de empresas del Sector Informático y Comunicaciones.

CGI (*Common Gateway Interface*). Programa cuya función es la de servir de interfaz entre un servidor con protocolo de transferencia de hipertexto (HTTP) y los demás recursos del servidor.

CLIENTE. Un programa que intenta conectarse al servidor, el cliente hace peticiones y el servidor se encarga de contestarlas.

DAEMON. Proceso en "background" responsable para una tarea específica en el sistema.

DATAGRAMA. Unidad básica de información empleada por TCP/IP

ENCAPSULACIÓN. Cuando se incluye un mensaje de entrada dentro de otro más extenso, agregando información al frente, atrás, o en ambas partes del mensaje. La encapsulación la utilizan los protocolos para redes en capas. Con cada capa se agregan nuevos encabezados y colas.

FTP (*File Transfer Protocol*). Permite transmitir ficheros sobre Internet entre una máquina local y otra remota. Los comandos básicos de FTP:

open 'nombre de nodo o dirección' Abre una sesión FTP en el ordenador indicado.

dir Lista los ficheros del directorio del ordenador al que nos hemos conectado.

pwd Visualiza el directorio remoto en el que estamos situados.
cd 'nombre de directorio' Cambio al directorio especificado.
lcd 'nombre de directorio' Comando de movimiento para directorios locales.
binary Establece modo binario de transferencia.
ascii Establece modo ascii de transferencia. Sólo para ficheros texto.
get 'nombre archivo' Obtiene un determinado fichero desde el ordenador remoto al local.
put 'nombre archivo' Transmite un determinado fichero desde nuestro directorio local al remoto.
bye Cierra una sesión FTP.

FTP Anonymous. Los servidores FTP anonymous son grandes cajones de ficheros distribuidos y organizados en directorios. Contienen programas (normalmente de dominio público o shareware), ficheros de imágenes, sonido y video.

El medio de acceso y recuperación de la información es FTP (*File Transfer Protocol*). Para entrar en estos servidores, tecleamos FTP y nombre del servidor. El sistema nos pregunta login, a lo que respondemos con la palabra 'anonymous' y en el password le indicaremos nuestra dirección de correo electrónico. Algunos servidores autentifican esta dirección.

Al existir miles de servidores FTP, se hace imprescindible una herramienta de búsqueda. Archie es la solución Cliente/Servidor implementada para este fin.

GATEWAY. Dispositivo que enruta los datagramas. También se utiliza para hacer referencia a cualquier dispositivo de la red que traduce los protocolos de un tipo de red a los de otra red.

GOPHER. Es un sistema de entrega de información distribuido. Utilizando GOPHER podemos acceder a información local o bien a acceder a servidores de información GOPHER de todo el mundo.

GOPHER combina las características de BBS (*Bulletin Board Service*) y bases de datos, permitiendo establecer una jerarquía de documentos, y permitiendo búsquedas en ellos por palabras o frases clave. Concebido y desarrollado en la Universidad de Minnesota en el año 91 es de libre distribución para fines no comerciales.

GOPHER soporta directorios, ficheros de texto, ítem de búsqueda, sesiones telnet y tn3270, multimedia y texto formateado (postscript y otros).

Algunos ejemplos de la información que GOPHER puede ofrecer:

- Pronósticos y mapas del tiempo
- Recetas
- Problemas y respuestas de temas de computación
- Animaciones sobre reacciones químicas
- Acceso a news
- Libros de los clásicos, de Shakespeare, Moby Dick, etc..
- Catálogos de bibliotecas de todo el mundo
- Canciones
- Catálogos de cursos universitarios

GOPHER trabaja en arquitectura Cliente/Servidor, existiendo clientes para: Macintosh, DOS, Microsoft Windows, Unix (Terminales ascii, emacs y X-Windows), VMS, NeXTstep, OS/2, VM/CMS, MVS y servidores para: UNIX, VMS, Macintosh, VM/CMS, DOS, OS/2, MVS

HTML (*HiperText Markup Language*). Lenguaje de Marcación de Hipertexto, empleado para crear documentos(páginas) dentro del World Wide Web.

HTTPS. Versión del HTTP que maneja seguridad con la implementación de SSL.

INTERNET. Es la red de redes. Nacida como experimento del ministerio de defensa americano, conoce su difusión más amplia en el ámbito científico-universitario.

JAVASCRIPT. Es un lenguaje creado por Netscape para desarrollo de páginas de WWW.

LAN (*Red de Área Local*). Conjunto de dispositivos conectados para permitir las comunicaciones entre sí en un solo medio físico.

MAC. Los Message Authentication Codes (MAC's) son etiquetas de autenticación de mensajes. El tipo más común de MAC es el basado en funciones de hash, por ejemplo MD5. En este caso, se calcula MD5 del mensaje y se cifra el resultado con la llave privada del servidor. Podemos decir que las MAC's son firmas electrónicas.

MACINTOSH. Serie de ordenadores de Apple Computer. Posee un sistema operativo basado en ventanas. El entorno es intuitivo, eliminando el teclado de los comandos del sistema. Prácticamente todo puede hacerse a través de menús desplegables y de ratón. A todos los objetos se le asigna una representación gráfica (iconos).

MAIL. El correo electrónico es el servicio más básico, antiguo, y más utilizado dentro de Internet. La mensajería electrónica es el medio más eficaz y más rápido de comunicación, permite intercambiar además de mensajes, programas, audio, vídeo e imágenes.

MD5. Algoritmo de RSA Data Security, Inc.

MILNET (*Military Network*). Una red que inicialmente era parte de ARAPANET, y que actualmente está diseñada para uso exclusivo militar.

NAVEGADOR. También conocidos como visualizadores son programas que permiten acceder al WWW.

NCSA (*National Center for Supercomputing Applications*). De la Universidad de Illinois, desarrolladores de Mosaic, el primer cliente para servidores WWW

NODO. Término empleado para hacer referencia a los dispositivos de la red.

PAQUETE. Se emplea para hacer referencia a los datos transferidos a través de una red.

PROTOCOLO. Reglas que determinan el comportamiento de operación de alguno de los aspectos de la red.

RECURSO. Se emplea para referirse tanto a los programas de aplicación, como a las capacidades como la memoria, redes, etc.

SEGMENTO. Unidad de datos del protocolo (PDU) que consiste en un encabezado TCP y en los datos (opcionales). También empleado para referirse a las partes de una red que está dividida en partes más pequeñas (segmentos).

SERVIDOR. Aplicación que responde las solicitudes del cliente.

SGML (*Standard General Markup Language*). Lenguaje Estándar de Marcación General. Un poderoso lenguaje de marcación que le permite estructurar documentos de manera que puedan exhibirse en cualquier tipo de computadora. El estándar actual de HTML está definido utilizando SGML.

TCP/IP (*Transmission Control Protocol/Internet Protocol*). Familia de protocolos que hacen posible la interconexión y tráfico de red en Internet. A ella pertenecen por ejemplo: FTP, SMTP, NNTP, etc. Los dos protocolos más importantes son los que dan nombre a la familia IP y TCP

URL (*Uniform Resource Locator*). Localizador Uniforme de Recursos. Se emplea para especificar la ubicación y nombre de documentos World Wide Web. También puede especificar otros servicios de Internet disponibles desde navegadores.

WWW (*World Wide Web*). Servidor de información, desarrollado en el CERN (Laboratorio Europeo de Física de Partículas), buscando construir un sistema distribuido hipermedia e hipertexto. También llamado WEB y W3. Existen grandes cantidades de clientes WWW para diferentes plataformas.

Apéndice A

Como ya se mencionó anteriormente, se empleó HTML para la elaboración de la interfaz gráfica, un ejemplo lo tenemos en la pantalla de Sistemas Acuáticos: Consultar Información (*acuaticos2.HTM*), el código es el siguiente:

```
<HTML>
  <HEAD>
    <META NAME="Author" CONTENT="Alba Marquez Velasco">
    <TITLE>Acuaticos2</TITLE>
  </HEAD>
  <BODY BACKGROUND="cloudtile.jpg">
    <CENTER>
      <H2>
        <HR WIDTH="100%"><I><FONT COLOR="#000080">Sistemas
        Acuaticos</FONT></I></H2>
      </CENTER>
      <HR WIDTH="100%">
      <CENTER>
        <H3>
          <I><TT><FONT COLOR="#000080">Consultar
          Informacion</FONT></TT></I></H3>
        </CENTER>
        <UL>
          <DT>
            <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
            WIDTH=14</A><I><A HREF="infoafo.htm">Aforos</A></I></DT>
          <DT>
            <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
            WIDTH=14</A><I><A HREF="infocam.htm">Campo
            de Hidrologia e Hidrodinamica</A></I></DT>
          <DT>
            <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
            WIDTH=14</A><I><A HREF="infohid.htm">Hidrocarburos</A></I></DT>
          <DT>
            <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
            WIDTH=14</A><I><A HREF="infocal.htm">Calidad del agua</A></I></DT>
          <DT>
            <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
            WIDTH=14</A><I><A HREF="infoinde.htm">Campo
            de Descargas Residuales</A></I></DT>
          <DT>
            <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
            WIDTH=14</A><I><A HREF="infofab.htm">Analisis
            de Laboratorio de Descargas Residuales</A></I></DT>
          <DT>
            <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
            WIDTH=14</A><I><A HREF="infoca.htm">Campo
            de Cuerpos Receptores</A></I></DT>
```

```

        <DT>
        <A HREF="infoafo.htm"><IMG SRC="blueball.gif" BORDER=0 HEIGHT=14
        WIDTH=14></A><|><A HREF="infoanl.htm">An&aacute;lisis
        de Laboratorio de Cuerpos Receptores</A></|></DT>
    </UL>
    <HR WIDTH="100%">
    <P><A HREF="menu.htm"><IMG SRC="755.gif" BORDER=0 HEIGHT=32 WIDTH=32></A><A
    HREF="menu.htm">Regresar</A>
</BODY>
</HTML>

```

De igual forma, se verá un ejemplo general de los programas empleados. En este caso será *campo.h*, el cual define la estructura de la llave y de los campos necesarios donde se almacenará información:

```

/* informacion de campo */
/* campo.h */

/* Incluir las librerias necesarias para que el programa corra */
#define xPCTEST
#ifdef PCTEST
#include "d:\clair\smd_3_2.h"
#else
#include "smd_3_2.h"
#endif
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <math.h>
typedef struct
{
    /* Definición de la estructura, de acuerdo a los campos que necesitamos para almacenar la
información */
    unsigned char nomen_pm[8];
    unsigned char no_muestra[6];
    char dia;
    char mes;
    int anio;
    int hora;
    int minutos;
    float ph;
    int condu;
    float turb;
    float tempe;
    float oxdis;
} registro_cam;

int error(place *,place *,int);
void check(double);
char id_mes(char *);
char * mes(char);
void printtail(void);
void printhead(void);

```


campo.c describe las rutinas empleadas:

```

/* campo.c */

void flotante(query)
{
char numero[10];
int n=0,i,bandera=0;
printf("Introduce un numero:");
gets(numero);
while(numero[n]!='\0')
{
    if(((isdigit(numero[n]))||((numero[n]=='.')))
        {}
    else bandera=1;
    n++;
}
if(bandera)
{
    exit(error(NULL,NULL,22));
}
else
{
    return (bandera);
}
return(bandera);
}

char id_mes(char *cual)
{
if (strcmp(cual,"Ene")==0) return 1;
if (strcmp(cual,"Feb")==0) return 2;
if (strcmp(cual,"Mar")==0) return 3;
if (strcmp(cual,"Abr")==0) return 4;
if (strcmp(cual,"May")==0) return 5;
if (strcmp(cual,"Jun")==0) return 6;
if (strcmp(cual,"Jul")==0) return 7;
if (strcmp(cual,"Ago")==0) return 8;
if (strcmp(cual,"Sep")==0) return 9;
if (strcmp(cual,"Oct")==0) return 10;
if (strcmp(cual,"Nov")==0) return 11;
if (strcmp(cual,"Dic")==0) return 12;
return 0;
}

char * mes(char cual)
{
switch (cual)
{
case 1: return("Ene");
case 2: return("Feb");
case 3: return("Mar");
case 4: return("Abr");
case 5: return("May");
case 6: return("Jun");
case 7: return("Jul");
}
}

```

```

case 8: return("Ago");
case 9: return("Sep");
case 10: return("Oct");
case 11: return("Nov");
case 12: return("Dic");
}
return 0;
}

/* En caso de existir algún error, el programa entrará aquí */
int error(place *node,place *nodeP,int code)
{
registro_cam *dat;
if (nodeP!=NULL)dat=(registro_cam *)nodeP->data;
printf("Content-type: text/html\n\n");
/* Para desplegar el mensaje de error en una página HTML */
printf("<HTML><body bgstart bgcolor=\\"#ffffff" text=\\"#0000ff" link=\\"#ff0000" VLINK=\\"#aaaaaa"
bgend><body>");
printf("<p><h2><center>\n");
/* Se seleccionará el tipo de error */
switch (code)
{
case 1: printf("No puedo abrir la base de datos.");break;
case 2: printf("No puedo actualizar la base de datos"); break;
case 3: printf("La planta no esta registrada...");break;
case 4: printf("Revise sus valores numéricos en la forma.");
case 5: printf("Password incorrecto.");
printf("<p><a href=\\"http://ixchel.imp.mx/claire/login.html">Inténtelo nuevamente...</a>");break;
case 6: printf("Uno de los campos es más grande que 16581375 (unsigned char*3)");break;
case 7: printf("Llene todos los campos necesario de la forma.");
printf("<p>Presione BACK para intentar de nuevo" ); break;
case 9: printf("Es necesario proporcionar la clave, tipo, nombre y centro de trabajo");
printf("<p>presione BACK para intentar de nuevo"); break;
case 10: printf("Su registro ha sido aceptado."); break;
case 11: printf("Todos los campos deben de ser positivos.");
printf("Presione BACK para intentar de nuevo "); break;
case 12:printf("No puedo crea el campo de la tabla.");
printf("Presione BACK para intentar de nuevo "); break;
case 14: printf("Por favor checa el orden de tu reporte");
printf("\n<p> Presione BACK para intentar de nuevo ");break;
case 15: printf("No puedo continuar con el proceso");
printf("\n<p> Presione BACK para intentar de nuevo ");break;
case 16: printf("Se ha actualizado la base de datos. Ya puede meter información del monitoreo");
printf("\n ");break;
case 18: printf("El registro solicitado ya esta en la base de datos.");
printf("<p><p>Si desea cambiar, seleccione dicha opción en la forma"); break;
case 19: printf("La suma de los porcentajes de composición de gas natural y combustóleo no suman
100%%.");
break;
case 20: printf("Es necesario llenar la forma.");
printf("<p> Por favor intente de nuevo."); break;
case 21: printf("No hay información de la llave solicitada");
printf("<p> Por favor intente de nuevo."); break;
case 22:printf("Se detectaron errores en los datos");
printf("<p> Por favor revíselos."); break;
}
}

```

```

printf("\n</h2></center></body></html>");
if (node!=NULL) closesomod(node);
if (nodeP!=NULL) closesomod(nodeP);
return 1;
}

void printhead(void) /*Imprime un encabezado para presentar la información en una página HTML */
{
printf("Content-type:text/html\n\n");
printf("\<HTML><body bgstar bgcolor=#ffffff text=#0000ff link=#ff0000 VLINK=#aaaaaa bgend>");
printf("<body><p>");
return;
}

void printtail(void) /* Final de la página HTML */
{
printf("\n</h2></center></body></html>");
}

```

put_camp.c registra la información :

```

/* Almacenará la información en el archivo campohh.dat */
/*put_camp.c*/
#include "campo.h"
#include "campo.c"
#define xPCTEST
#define
"Nomencultura=cactus&No_muestra=000014&Dia=21&Mes=Mar&Año=1997&Hora=13&Minutos=50&pH=1.7&Condu
ctividad=150&Turbiedad=80.7&Temperatura=10.3&Oxigeno_disuelto=12.3"
#ifdef IRIX
#define FILEDIR "f:\cgi-bin\tabasco\
#else
#define FILEDIR ""
#endif

int serial=1;
unsigned char change=0;
registro_cam data;
void process_query(place *,char *);

void main()
{
char *query; void * temp; place *node;
char archivo[128];
#ifdef PCTEST
query=COMMAND;
#else
query=getenv("QUERY_STRING");
#endif
sprintf(archivo,"%scampohh.dat",FILEDIR);
if ((node=opensomod(archivo))==NULL) node=createsomod(archivo,13);
node->operate=NULL;
process_query(node,query);
if ((load(node)!=-1)&&(!change)) exit(error(node,NULL,18));

```

```

temp=node->data;
node->data=&data;
node->bytes_userdata=sizeof(registro_cam);
if (update(node)==-1) {node->data=temp;exit(error(node,NULL,2));}
node->data=temp;
exit(error(node,NULL,10));
}

/* Checará que todos los campos traigan información, en caso de no ser así, enviará un mensaje de error */
void process_query(place *node,char *query)
{
double numero; int numerol; float numeroF;
unsigned char i;
/* process query string */
if (strstr(query,"registro=modreg")) change=1;
if ( (strstr(query,"Nomenclatura=&")!=NULL)||
      (strstr(query,"No_muestra=&")!=NULL)||
      (strstr(query,"Dia=&")!=NULL)||
      (strstr(query,"Mes=&")!=NULL)||
      (strstr(query,"Año=&")!=NULL)||
      (strstr(query,"Hora=&")!=NULL)||
      (strstr(query,"Minutos=&")!=NULL)||
      (strstr(query,"pH=&")!=NULL)||
      (strstr(query,"Conductividad=&")!=NULL)||
      (strstr(query,"Turbiedad=&")!=NULL)||
      (strstr(query,"Temperatura=&")!=NULL)||
      (strstr(query,"Oxigeno_disuelto=&")!=NULL))
    exit(error(NULL,NULL,20));

/* Asigna valores a cada uno de los campos*/
query=strtok(query,"=");
query=strtok(NULL,"&");
for (i=0;i<=7;i++) { node->key[i]=data.nomen_pm[i]=query[i];}

query=strtok(NULL,"=");
query=strtok(NULL,"&");
for(j=0;j<=5;j++)
data.no_muestra[j]=query[j];
node->key[8]=data.no_muestra[0];node->key[9]=data.no_muestra[1];
node->key[10]=data.no_muestra[2];node->key[11]=data.no_muestra[3];
node->key[12]=data.no_muestra[4];node->key[13]=data.no_muestra[5];

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numerol=atoi(query);
data.dia=numerol;

query=strtok(NULL,"=");
query=strtok(NULL,"&");
data.mes=id_mes(query);

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numerol=atoi(query)-1900;
data.anio=numerol;

```

```

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numeroI=atoi(query);
data.hora=numeroI;

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numeroI=atoi(query);
data.minutos=numeroI;

query=strtok(NULL,"=");
query=strtok(NULL,"&"); flotante(query);
if(bandera==0)
{
    numeroF=atof(query);
    data.ph=numeroF;
}

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numeroI=atoi(query);
data.condu=numeroI;

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numeroF=atof(query);
data.turb=numeroF;

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numeroF=atof(query);
if(atof==0) exit(error(NULL,NULL,22));
data.tempe=numeroF;

query=strtok(NULL,"=");
query=strtok(NULL,"&");
numeroF=atof(query);
data.oxdis=numeroF;
}

```

info_cam.c busca la información de una llave determinada, en caso de encontrarla despliega toda la información, en caso contrario, despliega un mensaje indicando que no encontró la llave solicitada:

```

/*info_cam.c*/
#include "campo.h"
#include "campo.c"
#define nPCTEST
#define COMMAND "Nomenclatura=cactus&No_muestra=000014"
#ifdef IRIX
#define FILEDIR "f:\cgi-bin\tabasco"
#else
#define FILEDIR ""
#endif

```

```
int serial=1;
unsigned char change=0;
unsigned char nohay=1;
registro_cam data;
unsigned char * process_query(char *);
void printdatos(place *);

void operate(place *node)
{
static unsigned char first=1;
registro_cam *dat;
dat=(registro_cam *)node->data;
nohay=0;
first=0;
printf("<TR><TD>"); printdatos(node);
printf("</TD><TD>"); printf("%d",dat->dia);
printf("</TD><TD>"); printf("%02d",dat->mes);
printf("</TD><TD>"); printf("%d",dat->anio+1900);
printf("</TD><TD>"); printf("%d:%d",dat->hora, dat->minutos);
printf("</TD><TD>"); printf("%.2f",dat->ph);
printf("</TD><TD>"); printf("%d",dat->condu);
printf("</TD><TD>"); printf("%.1f",dat->turb);
printf("</TD><TD>"); printf("%.1f",dat->tempe);
printf("</TD><TD>"); printf("%.2f",dat->oxdis);
printf("</TD><TD>");
}

void main()
{
char *query;
place *node;
char archivo[128];
int i;
unsigned char *startkey;
registro_cam *dat;

#ifdef PCTEST
query=COMMAND;
#else
query=getenv("QUERY_STRING");
#endif

sprintf(archivo,"%scampohh.dat",FILEDIR);

if ((node=opensomod(archivo))==NULL) exit(error(NULL,NULL,1));

node->operate=operate;
startkey=process_query(query);
for (i=0;i<=13;i++) { node->key[i]=startkey[i]; }

if (load(node)==-1) exit(error(node,NULL,21));
dat=(registro_cam *)node->data;

printhead();
printf("<TABLE>");
```

```

printf("<TR><TH COLSPAN=22>Hidrologia e Hidrodinámica - Informacion de Campo<TH></TR>");
printf("<TR><TH>Llave Buscada</TH><TH>Dia</TH><TH>Mes</TH><TH>Año</TH><TH>Hora </TH>
    <TH>pH</TH> <TH> Conduc</TH>");
printf("<TH>Turbiedad</TH><TH>Temp</TH><TH>Oxi_dis</TH>");
newbarre(node,startkey,NULL,8);
printf("</TABLE>");

if (nohay) printf("\n No hay ninguna informacion para la llave solicitada");
printtail();
closesomod(node);
}

/* Imprimirá los datos de la llave que encontró */
void printdatos(place *node)
{
    int i;
    registro_cam *dat;
    dat=(registro_cam *)node->data;
    printf("Nomenclatura=%c%c%c%c%c%c%c%c", dat->nomen_pm[0], dat->nomen_pm[1],
        dat->nomen_pm[2],dat->nomen_pm[3],dat->nomen_pm[4],dat->nomen_pm[5],
        dat->nomen_pm[6],dat->nomen_pm[7]);
    printf("<br>");
    printf("No_muestra=%c%c%c%c%c%c",dat->no_muestra[0],dat->no_muestra[1],
        dat->no_muestra[2],dat->no_muestra[3],dat->no_muestra[4],
        dat->no_muestra[5]);
}

/* Checará que todos los campos trigan infromación */
unsigned char * process_query(char *query)
{
    unsigned char i;
    static unsigned char startkey[14];
    int numeroI; float numeroF;

    if (    (strstr(query,"Nomenclatura=&")!=NULL)||
        (strstr(query,"No_muestra=&")!=NULL)||
        (strstr(query,"Dia=&")!=NULL)||
        (strstr(query,"Mes=&")!=NULL)||
        (strstr(query,"Año=&")!=NULL)||
        (strstr(query,"Hora=&")!=NULL)||
        (strstr(query,"Minutos=&")!=NULL)||
        (strstr(query,"pH=&")!=NULL)||
        (strstr(query,"Conductividad=&")!=NULL)||
        (strstr(query,"Turbiedad=&")!=NULL)||
        (strstr(query,"Temperatura=&")!=NULL)||
        (strstr(query,"Oxigeno_disuelto=&")!=NULL))
        exit(error(NULL,NULL,20));

    /* process query string */
    /******se carga la clave de consulta *****/
    query=strtok(query,"=");
    query=strtok(NULL,"&");
    for (i=0;i<=7;i++) { startkey[i]=query[i]; }

    query=strtok(NULL,"=");

```

```

query=strok(NULL,"&");
startkey[8]=query[0];startkey[9]=query[1];
startkey[10]=query[2];startkey[11]=query[3];
startkey[12]=query[4];startkey[13]=query[5];

return (startkey);
/* end of query string processing */
}

```

list_cam.c desplegará toda la información que se encuentra almacenada en el archivo *aforos.dat* (por lo menos deberá contener una llave almacenada), en caso de no existir información, enviará un mensaje de error:

```

/*list_cam.c*/

#include "campo.h"
#include "campo.c"

#define nPCTEST

#define COMMAND "modo=sistema"
#ifdef IRIX
#define FILEDIR "f:\cgi-bin\tabasco/"
#else
#define FILEDIR ""
#endif

int sistema=0;
unsigned char change=0;
unsigned char nohay=1;
registro_cam data;

void process_query(char *);

#define DATABDIR "/cgi-bin/tabasco/"

void operate(place *nodeP)
{
registro_cam *dat;
dat=(registro_cam *)nodeP->data;

nohay=0;

printf("<TR><TD>");printf("%c%c%c%c%c%c%c%c",dat->nomen_pm[0],dat->nomen_pm[1],
dat->nomen_pm[2],dat->nomen_pm[3],dat->nomen_pm[4],dat->nomen_pm[5],
dat->nomen_pm[6],dat->nomen_pm[7]);
printf("</TD><TD>");printf("%c%c%c%c%c%c",
dat->no_muestra[0],dat->no_muestra[1],dat->no_muestra[2],
dat->no_muestra[3],dat->no_muestra[4],dat->no_muestra[5]);
printf("</TD><TD>");printf("%02d", dat->dia);
printf("</TD><TD>");printf("%02d",dat->mes);
printf("</TD><TD>");printf("%d",dat->anio+1900);
printf("</TD><TD>");printf("%d:%d",dat->hora,dat->minutos);
printf("</TD><TD>");printf("%.2f",dat->ph);
printf("</TD><TD>");printf("%d",dat->condu);

```



```

printf("</TD><TD>");printf("%.1f",dat->turb);
printf("</TD><TD>");printf("%.1f",dat->tempe);
printf("</TD><TD>");printf("%.2f",dat->oxdis);
printf("</TD><TD>");

}

void main()
{

char *query; place *nodeP;
char archivo[128];

query=COMMAND;
process_query(query);

sprintf(archivo,"%scampohh.dat",FILEDIR);
if ((nodeP=opensomod(archivo))==NULL) exit(error(NULL,NULL,1));
nodeP->operate=operate;

printhead();
printf("<TABLE>");
printf("<TR><TH COLSPAN=22>Hidrologia e Hidrodinamica - Informacion General de Campo<TH></TR>");
printf("<TR><TH>Nomenclatura</TH><TH>No_Muestra</TH><TH>Dia</TH><TH>Mes</TH><TH>Año
    </TH> <TH>Hora</TH><TH>pH</TH><TH>Conduc</TH>");
printf("<TH>Turbiedad</TH><TH>Temp</TH><TH>Oxi_dis</TH>");
newbarre(nodeP,NULL,NULL,NULL);
printf("</TABLE>");
if (nohay) printf("\n<p>No hay ninguna fuente registrada del grupo solicitado.");
printtail();
}

void process_query(char *query)
{
if (strstr(query,"system")) sistema=1;else sistema=0;

}

```

Apéndice B

Netscape Enterprise Server versión 2.13 para Windows NT

Netscape Enterprise Server es un servidor para WWW, desarrollado en la compañía *Netscape Communications Corporations* y elaborado con *Java* y *JavaScripts*.

A partir de estas líneas, se considerará Netscape Enterprise Server sólo como Enterprise Server.

Entre las características principales que surgen en Enterprise Server son:

Control de acceso. Ofrece la posibilidad de “proteger” desde archivos específicos hasta directorios completos, esta restricción de acceso se lleva a cabo utilizando un “nombre de usuario” y su respectiva contraseña (*login* y *password*). Asimismo, permite restringir el acceso por medio de dirección IP o por nombre de dominio.

Servicio de Administración Remota. Para el manejo de administración, proporciona una interfaz que puede manejarse desde un navegador de WWW, esta interfaz permite configurar, encender y apagar el servidor además de poder realizar las tareas de administración de éste; debido a que es una página HTML ésta se puede acceder desde cualquier computadora con servicio de WWW, por seguridad se proporciona una sección para restringir el acceso a la página de administración del servidor.

Herramienta de búsqueda. Un elemento principal que viene integrado, es la herramienta de búsqueda, ésta permite realizar búsquedas por palabras en colecciones que se encuentren en el servidor, entendiéndose por colecciones como agrupaciones de archivos que se pueden generar en el servidor.

Instalación y manejo de múltiples servidores. Además de mantener un (1) servidor de WWW, permite instalar múltiples servidores que pueden controlarse de manera independiente o alternar con el existente y todos ellos con la posibilidad de ser instalados en la misma computadora.

Seguridad. La transmisión de la información vía WWW se lleva a cabo de forma confidencial entre el servidor y el cliente utilizando el protocolo SSL. La utilización de este protocolo permite mantener un control de los accesos a los recursos del servidor mediante un “nombre de usuario” y su respectiva contraseña.

REQUERIMIENTOS

Se debe considerar que la computadora en la que se realiza la instalación debe cumplir con los siguientes requerimientos mínimos de hardware:

- Sistema Operativo NT 3.51 o superior
- Procesador 486 o superior
- 32 MB en RAM como mínimo
- 80 MB de espacio libre en disco duro

INSTALACIÓN

Una vez cumplidos los requerimientos de hardware, se recomienda considerar los siguientes datos antes de realizar el proceso de instalación de Enterprise Server:

- Tener disponible en la computadora el archivo *esni20.exe*, (archivo de instalación de Enterprise Server) el cual se encuentra disponible en la dirección electrónica:
http://home.netscape.com/download/server_download.html?enterpriseintl3.0defr
en donde se seleccionará "Netscape Enterprise Server Standard Edition 2.13"
- Haber elegido un nombre para el servidor (por omisión es el nombre de la máquina)
- Tener asignado un nombre de usuario para utilizarlo como clave de entrada y su respectiva contraseña, para tareas de administración
- Tener asignado un número de puerto en el que se ejecutará el servicio de "Netscape Enterprise https-tabasco" para los clientes y el de "Netscape Admin Server" para la administración de el servidor
- Asignar una cuenta de usuario en la que se llevarán a cabo los procesos del servidor WWW (Por omisión es *Local System*)
- Determinar el directorio principal de los archivos de Enterprise Server, es decir, que se debe tener asignada una ruta en la que se instalarán los archivos de configuración del servidor WWW

Cuando se tienen los elementos para la instalación, se procederá a la ejecución del archivo *esni20.exe*, durante este proceso, se requerirán los datos anteriormente listados, además, se debe especificar una ruta en la que se almacenarán los archivos que se pondrán a disposición de los usuarios a través de WWW, el directorio generado en esta ruta se considerará como "directorio público" (cliente WWW).

En la ruta en la que se instalen los archivos de configuración del servidor, se crearán los siguientes directorios, de los cuales se especifica su contenido:

/admserv	Contiene los archivos de administración del servidor, (cuenta de administración y su respectiva contraseña) y algunos archivos binarios de configuración.
/authdb	Se considera un directorio principal de bases de datos, contiene las bases de datos de los grupos y usuarios.
/bin	Incluye algunos archivos de configuración ejecutables y en formato html.
/docs	Por omisión se considera directorio principal para la documentación del servidor.
/extras	Comprende el directorio de bases de datos que contiene programas que permiten crear, cambiar y borrar bases de datos de usuarios y grupos, incluye además un programa en línea que analiza el archivo de accesos de los clientes al servidor, éste se encuentra en el subdirectorio <i>flexanlg</i> (de este mismo directorio).

- /httpacl** Incluye archivos que almacenan la información de configuración del control de accesos, además de almacenar los cambios que se realizan en las formas del Servidor de Administración.
- /https-tabasco** Este directorio, se genera por cada servidor que se cree, es decir que si se crea el servidor "NUEVO", existirá un directorio independiente llamado "https-nuevo". Para el nombre de este directorio se toma en cuenta el tipo de configuración del servidor (refiriéndose a SSL->https, o sin SSL-> http).
- /config** Incluye los archivos de tipo MIME, además de contener la configuración de los certificados de encriptamiento para SSL.
- /logs** Contiene los archivos de registro de "errores de accesos" y de "accesos" al servidor.
- /ns-icons** Este directorio contiene los íconos que se utilizan cuando se generan automáticamente las páginas de WWW del servidor (ejemplos: archivo de directorios generales, lista de resultados de búsquedas).
- /plugins** Este directorio incluye los subdirectorios:
- Java (Contiene herramientas para trabajar con el lenguaje Java)
 - Autocatalog (Se utiliza para la generación de estadísticos de accesos a las páginas)
 - Herramienta de búsqueda (Elemento adicional del servidor)

CONFIGURACIÓN

Server Selector

Una vez que se realizó con éxito la instalación del servidor se procede a llevar a cabo la configuración del mismo mediante un servicio que proporciona *Enterprise Server* llamado "*Server Selector*", éste es la interfaz gráfica que nos permite realizar una serie de configuraciones (básicas) para el servicio de administración de el o los servidor(es) que se encuentren instalados, la utilización de esta herramienta se lleva a cabo desde cualquier navegador de WWW.

Desde "*Server Selector*" se permiten desarrollar las siguientes tareas:

- Instalar otro servidor WWW en la misma computadora
- Llevar a cabo configuraciones básicas de el (los) servidor(es) que se encuentren instalados
- Eliminar algún servidor instalado desde la computadora en uso
- Activar o desactivar algún(os) servidor(es)

Al finalizar la instalación, se crea el grupo de trabajo "Administrador de servidores" (*Netscape Servers Admin*) el cual contiene un ícono en el que se ejecuta el navegador de WWW con acceso directo a la página de "*Server Selector*"; podemos acceder a la página de WWW de "*Server Selector*" desde cualquier navegador en donde se realizan los siguientes pasos:

- a) Utilizar un navegador de WWW que soporte *Frames* y *JavaScript*, teclear el URL del servicio de administración, es decir:

http://tabasco.iingen.unam.mx:8100

el cual tiene el siguiente formato :

http://[nombre del servidor].[subdominio].[dominio]:[número de puerto]

Es importante mencionar que se debe teclear el puerto que se designó para la administración del servidor durante el proceso de instalación.

- b) Enseguida deberá teclear el “nombre de usuario” y la contraseña de administración que se indicó durante el proceso de instalación.

Finalmente, cuando se realicen con éxito los pasos anteriores tendrá acceso a la página “Server Selector”, la cual se muestra en la siguiente figura:

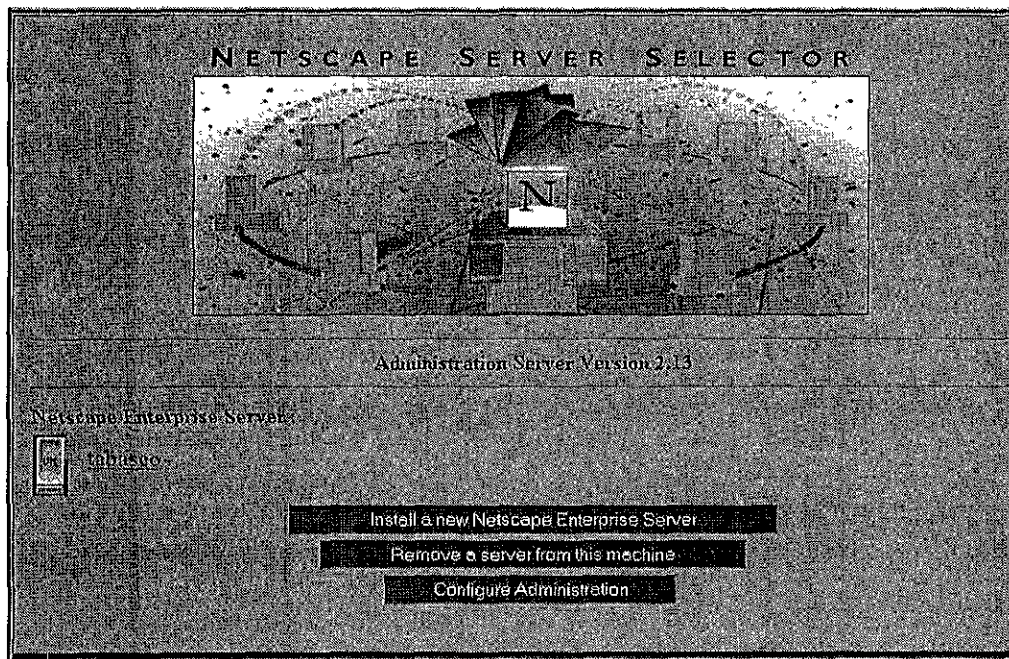


Figura B-1

A continuación se describirán los elementos que lo componen:

Instalación de múltiples servidores (*Install a New Netscape Enterprise Server*)

Enterprise Server brinda la posibilidad de tener instalado más de un servidor de WWW en la misma computadora. El servidor que se instale puede compartir algunas configuraciones de un servidor existente o puede manejarse de manera independiente.

Los servidores que se instalen pueden ejecutarse en cualquier puerto de TCP/IP del sistema de la computadora siempre y cuando no se ejecuten en el mismo número de puerto y al mismo tiempo. En caso de que la computadora tenga más de una dirección IP y éstas se utilicen para los servidores, entonces podrá manejarse el mismo puerto al mismo tiempo para ejecutar los servicios de los servidores.

Cuando se instalan servidores que comparten los archivos de configuración, es indiferente si alguno de ellos hace uso de un protocolo de seguridad y otro no.

Si se desea instalar un servidor de WWW con archivos de configuración independientes, se deben tener direcciones IP diferentes además de seguir los siguientes pasos:

- a) Desde “*Server Selector*”, elegir “*Instalación de un nuevo servidor*” (*Install a new Netscape Enterprise Server*)
- b) Llenar la forma con los siguientes datos:

Nombre del servidor (*Name Server*), en este registro se debe teclear el nombre del servidor con su correspondiente dominio y subdominio.

Dirección IP (*Bind address*), este registro se utilizará cuando se tenga un sistema que maneje múltiples direcciones IP, en caso de serlo así, se designará una dirección IP diferente a la que se tiene en uso.

Número de puerto (*Port Number*), se debe indicar en este registro el número de puerto que se utilizará para el servicio del nuevo servidor WWW.

Identificador del servidor (*Server Identifier*), se refiere al nombre con el que se desee reconocer al servidor.

Eliminar un servidor (*Remove a server from this machine*)

En este apartado, aparece un registro en el que se enlistan los servidores que se encuentran instalados, una vez elegido el servidor a eliminar aparece un registro en el que se indica si se desean eliminar los archivos binarios de configuración, ya que se han hecho las selecciones anteriores, se procede a eliminar el(los) servicio(s) del servidor WWW y/o de todos los archivos de configuración.

Configuración de administración (*Configure Administration*)

Para minimizar los riesgos de seguridad, se recomienda inhabilitar el servicio de administración del servidor WWW en cuanto se termine de utilizar éste. Para realizar lo anterior desde el “*Server Selector*”, deberá oprimir la liga “*Detener el servicio administración WWW*” (*Shut down your admin server*), con esto se detendrá el servicio que ejecuta el servidor WWW para su administración.

En esta misma página se pueden consultar y/o modificar las siguientes configuraciones:

a) Configuración (*Daemon configuration*)

- Usuario de administración (*User to run as*), nombre de la cuenta en la que se está ejecutando el servicio del servidor de administración
- Puerto de administración (*Port number*), número de puerto en el que se ejecuta el servicio de administración del servidor
- Habilitación de seguridad en el servidor (*Activate SSL and specify a key pair and certificate file*), activación/desactivación y configuración del protocolo de seguridad

b) Control de acceso (*Access Control*)

- Usuario de administración (*Authentication Username*), nombre del usuario de administración que se indicó durante el proceso de instalación del servidor
- Contraseña de Administración (*Authentication Password*), indica la contraseña que se proporcionó durante la instalación
- Restricción de lugares de acceso (*Host to allow*), para realizar el proceso de administración, Enterprise Server permite restringir el servicio de acceso de administración vía WWW, esta restricción se puede realizar por dirección IP o por dominio

c) Almacenamiento de archivos (*Options*)

- Almacenamiento de accesos al servidor (*Access loggin*), permite cambiar la ruta en la que se almacenan los archivos de acceso que comúnmente se generan con HTTP. Al dejar este registro en blanco se desactivará el almacenamiento de accesos al servidor
- Detalle de accesos (*Detailed logging*), mediante esta opción, el servidor almacena una lista de las ocasiones en que se han realizado cambios de configuración. Si el registro está en blanco, no se almacenan los detalles de los accesos
- RespalDOS (*Configuration file backups*), Enterprise Server lleva a cabo una serie de almacenamientos de los archivos de configuración del servidor

Activación o Desactivación del servidor de WWW

“*Server Selector*” permite activar o desactivar el servidor, basta con oprimir la liga indicativa de encendido/apagado del servidor correspondiente.

SERVER MANAGER

“*Server Manager*”, es una herramienta gráfica que ofrece Enterprise Server para la administración general, con ésta podemos realizar la administración de la configuración del servidor así como del contenido del mismo.

La utilización de esta herramienta, se realiza desde una página en cualquier navegador de WWW. El acceso a esta página se lleva a cabo desde “*Server Selector*”, en donde se deberá oprimir la liga indicativa del nombre del servidor.

Una vez accesada a esta página, se presentan en pantalla tres secciones, en las que claramente podemos ver que en la sección del encabezado se hacen presentes los siguientes temas:

1. Configuración general (*System Settings*)
2. Control de acceso (*Access Control*)
3. Encriptamiento (*Encryption*)
4. Programas (*Programs*)
5. Estado del Servidor (*Server Status*)
6. Configuraciones de estilos (*Config Style*)
7. Manejo del contenido (*Content Management*)
8. Indexación de documentos (*Index Documents*)

9. Catálogo de Documentos (*Autocatalog*)

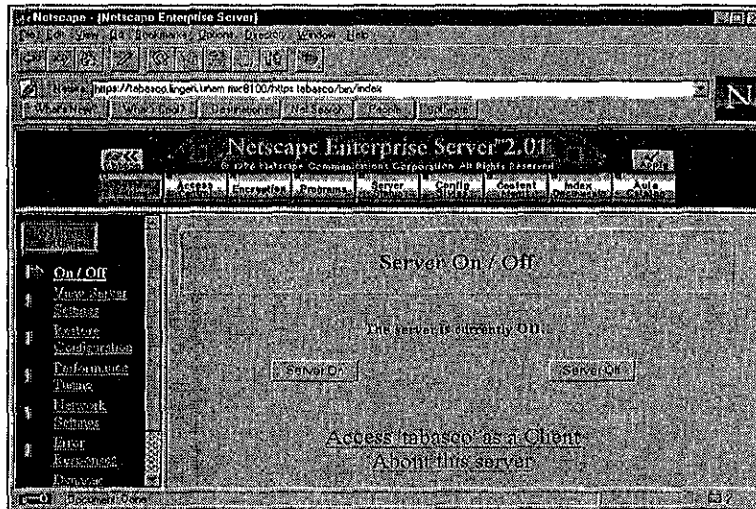


Figura B-2

Cada uno de ellos enlista diversas tareas para la administración y el mantenimiento de la información que contiene el servidor, a continuación se describen los elementos que los componen:

1. CONFIGURACIÓN GENERAL

Esta sección abarca los temas de habilitación y/o deshabilitación del servidor, además de edición, recuperación y modificación de configuraciones del servidor.

1.1 Encendido/apagado (*On/Off*)

En la sección principal de este apartado, se presenta el estado de actividad del servidor, es decir, si éste se encuentra apagado o encendido. Por medio de esta página, podemos habilitar o deshabilitar el servicio con el que se está ejecutando el servidor, con sólo oprimir las ligas **Encender** (*Server on*) o **Apagar** (*Server off*) respectivamente.

1.2 Configuraciones del servidor (*Viewing Server Settings*)

Por medio de esta utilidad, podemos consultar configuraciones generales básicas del servidor, es decir, nombre de la cuenta en la que se ejecuta el servicio de WWW, directorio público, restricciones de recursos del servidor, etc., estos temas se enlistan con los valores correspondientes de cada uno por medio de una liga, de manera que si se desea modificar su valor, al oprimir la liga se podrá acceder desde este apartado a la página en la que se encuentra detallado. La configuración del servidor se encuentra en los archivos localizados en el subdirectorio *admserv/https-tabasco*.

Los parámetros técnicos se encuentran en el archivo *magnus.conf*, enseguida se presenta su código:

```
#ServerRoot C:/enterp/https-tabasco
ServerName tabasco.ingen.unam.mx
Port 9000
LoadObjects obj.conf
RootObject default
```



```
ErrorLog C:/enterp/https-tabasco/logs/errors`
DNS off
Security on
Ciphers +rc4,+rc4export,+rc2,+rc2export,+des,+desede3
SSL3Ciphers+rsa_rc4_128_md5,+rsa_3des_sha,+rsa_des_sha,+rsa_rc4_40_md5,+rsa_rc2_40_md5,-sa_null_md5
ACLFile C:/enterp/httpacl/generated.https-tabasco.acl
Keyfile c:\enterp\Llaves\ServerKey.db
Certfile c:\enterp\Llaves\ServerCert
```

Los parámetros generales están contenidos en el archivo **obj.conf** del cual se presenta su código:

#Código del programa del archivo obj.conf

```
# Netscape Communications Corporation - obj.conf
# You can edit this file, but comments and formatting changes
# might be lost when the admin server makes changes.

# Use only forward slashes in pathnames--backslashes can cause
# problems. See the documentation for more information.

Init format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%" fn="flex-init" access="C:/enterp/https-
tabasco/logs/access"
Init fn="load-types" mime-types="mime.types"

<Object name="default">
NameTrans fn="search-find" from="/ns-search"
NameTrans from="/tabasco" fn="pfx2dir" dir="c:/enterp/docroot/tabasco"
NameTrans from="/ns-icons" fn="pfx2dir" dir="C:/enterp/ns-icons"
NameTrans from="/mc-icons" fn="pfx2dir" dir="C:/enterp/ns-icons"
NameTrans fn="pfx2dir" from="/cgi-bin" dir="c:/enterp/cgi-bin/tabasco" name="cgi"
NameTrans from="/search" fn="pfx2dir" dir="c:/enterp/plugins/search/vsearch" name="cgi"
NameTrans from="/count" fn="pfx2dir" dir="c:/enterp/cgi-bin/counter" name="cgi"
NameTrans fn="pfx2dir" from="/ui" dir="C:/enterp/plugins/search/ui"
NameTrans root="C:/enterp/docroot" fn="document-root"
PathCheck fn="nt-uri-clean"
PathCheck fn="find-pathinfo"
PathCheck index-names="index.html,home.html" fn="find-index"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"
Service fn="imagemap" method="(GET|HEAD)" type="magnus-internal/imagemap"
Service fn="send-error" path="c:/enterp/noIndice.html" method="(GET|HEAD)" type="magnus-
internal/directory"
Service fn="search-service" cgi-path="C:/enterp/plugins/search/vsearch/iarecord.exe" type="magnus-
internal/search"
Service fn="send-file" method="(GET|HEAD)" type="*~magnus-internal/*"
AddLog fn="flex-log" name="access"
</Object>

<Object name="cgi">
ObjectType fn="force-type" type="magnus-internal/cgi"
Service fn="send-cgi"
</Object>

<Object ppath="C:/enterp/docroot/tabasco/aire_y_materiales/*">
PathCheck fn="check-acl" acl="https-tabasco_formgen-READ-ACL_allow-8836" bong-
file="c:/enterp/docroot/mensajes/soloaire.html"
PathCheck fn="check-acl" acl="https-tabasco_formgen-WRITE-ACL_deny-8836" bong-
file="c:/enterp/docroot/mensajes/soloaire.html"
</Object>
```

```

<Object ppath="C:/enterp/docroot/tabasco/sistemas_acuaticos/*">
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-READ-ACL_allow-257"      bong-
file="c:/enterp/docroot/mensajes/soloacua.html"
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-WRITE-ACL_deny-263"      bong-
file="c:/enterp/docroot/mensajes/soloacua.html"
</Object>

<Object ppath="C:/enterp/docroot/tabasco/sistemas_biologicos/*">
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-READ-ACL_allow-861"      bong-
file="c:/enterp/docroot/mensajes/solobiol.html"
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-WRITE-ACL_deny-864"      bong-
file="c:/enterp/docroot/mensajes/solobiol.html"
</Object>

<Object ppath="C:/enterp/docroot/tabasco/sistemas_terrestres/*">
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-READ-ACL_allow-1249"      bong-
file="c:/enterp/docroot/mensajes/soloterr.html"
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-WRITE-ACL_deny-1253"      bong-
file="c:/enterp/docroot/mensajes/soloterr.html"
</Object>

<Object ppath="C:/enterp/docroot/tabasco/sistema_de_informacion/*">
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-READ-ACL_allow-9822"      bong-
file="c:/enterp/docroot/mensajes/soloinfo.html"
PathCheck      fn="check-acl"      acl="https-tabasco_formgen-WRITE-ACL_deny-9822"      bong-
file="c:/enterp/docroot/mensajes/soloinfo.html"
</Object>

```

A continuación se enumeran los parámetros técnicos del servidor

1. Server root, representa el directorio principal del servidor.
2. Hostname, es el URL que los usuarios reconocerán como ubicación del servidor.
3. Puerto, se designa un número de puerto en el que se ejecutará el servicio de WWW para los usuarios.
4. Error log, indica la ruta y nombre del archivo en que se encuentran almacenados los accesos que se han realizado al servidor.
5. User, indica el nombre de la cuenta en la que se está ejecutando el servicio de WWW.
6. Processes, es el número de procesos que el servidor permitirá que se ejecuten.
7. DNS, indica si se encuentra habilitado el servicio del Servidor de Nombres de Dominio (Domain Name System), es importante la habilitación de este servicio si en nuestra restricción a los recursos del servidor se está limitando el acceso por dominios, ya que si no está habilitado este servicio la restricción por dominios será inválida.

1.3 Recuperación de archivos de configuración (*Restore Settings*)

Enterprise Server facilita las tareas administrativas mediante la consulta y/o recuperación de archivos de configuraciones anteriores, (archivos *magnus.conf* y *obj.conf*), esta tarea la podemos realizar mediante la presente liga en la que se puede ver una lista de las últimas modificaciones de configuración del servidor que se llevaron a cabo, en esta lista, aparece la fecha en la que se llevó a cabo la última modificación, se anexan además “botones” por medio de los cuales se tiene la opción de editar y/o restablecer la configuración seleccionada.

1.4 Desarrollo óptimo de DNS (*Performance Tuning*)

Desde esta opción también podemos habilitar y/o deshabilitar la opción de DNS. En caso de habilitarla, desde esta sección se llevará a cabo la configuración de tamaño de caché para el almacenamiento de listas DNS.

1.5 Configuraciones de red (*Network Settings*)

Esta sección se utiliza para realizar cambios de:

- **Cuenta de usuario con la que el servidor se ejecutará (*User to run*)**. Por omisión la cuenta con la que inicia el servicio del servidor es *Local System*, para utilizar una cuenta diferente a esta, se deberá de registrar en la lista de usuarios de la computadora en la que se encuentra instalado el servidor y deberá tener permisos de “Administración” y “Acceso como servicio”
- **Nombre del servidor (*Name Server*)**. Cuando se instaló el servidor se pidió un nombre para éste (el cual se muestra en este apartado), este nombre se puede modificar desde esta sección con sólo anotar el nuevo nombre del servidor en este campo
- **Puerto de servidor**. Este apartado indica el número de puerto que utiliza el servicio de el protocolo TCP/IP designado a el servidor
- **Envío de mensajes de error (*Error messages*)**. El servidor envía mensajes de error cuando el archivo solicitado: no se encuentra, no existe, no se tiene autorización de acceso de lectura y/o escritura del archivo, se ha ejecutado incorrectamente un programa CGI y cuando el servidor tiene errores de configuración. Para tales hechos, el servidor ofrece la posibilidad de cambiar los mensajes respectivos por algún archivo en formato HTML o por un programa CGI, todo esto con la posibilidad de modificar el mensaje

1.6 Configuración de páginas (*Error responses*)

En esta sección se permite llevar a cabo la modificación de la página que se genera automáticamente cuando existen: errores de configuración del servidor, el documento solicitado no se encuentre, o no se tiene autorización de lectura para la información solicitada, entre otros, Específicamente en Enterprise Server, se generan páginas automáticamente debido a:

- **Acceso no autorizado (*Unauthorized*)**. Esto es, cuando se trata de acceder a información del servidor que se encuentra restringida y el cliente que está accediendo no tiene derechos sobre ella
- **Recurso no encontrado (*Not found*)**. Sucede cuando en el servidor no se encuentra el recurso solicitado o cuando el servidor ha sido configurado para que indique que el recurso no se encuentra
- **Error de configuración (*Server Error*)**. Página generada cuando existen errores de configuración del servidor o cuando el sistema se está ejecutando con memoria insuficiente para servir la solicitud del cliente

Una vez que se conocen las razones por las que se pueden generar estas páginas, en esta sección se tiene la posibilidad de elegir algún recurso del servidor y hacer que se despliegue una página diferente a la que se genera automáticamente. La elección del recurso se lleva a cabo mediante los botones **Explorar (*Browse*)** o **Agregar todo (*Wildcard*)**, una vez que se ha seleccionado el recurso, se presentan los 3 puntos anotados con anterioridad, para cada se tiene un campo en el que se especifica la ruta del archivo que se desplegará en lugar de la página por omisión, este archivo puede ser un archivo CGI o en formato HTML.

2. CONTROL DE ACCESO

Enterprise Server ofrece la posibilidad de mantener un control sobre el acceso a los recursos del servidor WWW mediante una lista de usuarios. Este control se basa en una estructura en la que existe una base de datos que almacena la información básica de los usuarios que registre el administrador, dicha estructura está conformada por:

- a) Base de datos. Enterprise Server permite manejar un sinnúmero de usuarios, para ello tiene la opción de generar bases de datos para llevar un control de usuarios, soporta la generación de diferentes bases de datos en un mismo servidor.
- b) Grupos de usuarios. Una vez que se desee trabajar con una base de datos, tenemos la posibilidad de generar grupos de usuarios que pertenezcan a ella.

Para llevar a cabo la utilización de esta herramienta, podemos auxiliarnos de los siguientes temas que se presentan en esta sección:

2.1 Creación de usuarios (*Create user*)

Al oprimir esta liga, se presenta una forma de registro en la que se pide anotar el nombre del usuario con el que se va a acceder además de anotar su nombre completo y contraseña de acceso, como medida de seguridad existe un campo en el que se pide teclear nuevamente la contraseña sólo para asegurar ésta; se incluye además el **Grupo:** (*Group:*), es decir un grupo al que pertenecerá este usuario dentro de la base de datos que se registre en esta sección.

2.2 Eliminar a un usuario (*Remove user*)

Enterprise Server presenta esta liga por medio de la cual se permite eliminar la cuenta de un usuario del servidor WWW, al utilizar esta opción no se podrá recuperar dato alguno de esta cuenta, para llevar a cabo esto, se debe anotar el nombre del usuario a eliminar y además de especificar la base de datos a la que pertenece, y si es el caso anotar la contraseña de uso para la base de datos.

2.3 Listar usuarios (*List users*)

Al iniciar esta sección se presenta automáticamente una lista de los usuarios que se encuentran registrados en el servidor WWW, para realizar búsquedas rápidas de usuarios, se cuenta con un registro en el que se permite limitar la lista que se presente, es decir que podemos utilizar “filtros” de manera que se obtenga una lista con usuarios específicos, por ejemplo, para indicar que se enlisten sólo aquellas cuentas de usuarios que inician con la letra *e*, entonces se anotará: *e**, (para mayor referencia ver tabla de formatos). Se presenta también un registro en el que se especifica la base de datos en la que se desea trabajar y si es el caso se deberá anotar su contraseña.

La lista que se presenta tiene el formato: nombre_de_usuario <opción de editar><opción de eliminar> las últimas opciones se presentan en forma de ligas, es decir, que podemos oprimir cualquiera de éstas para que nos lleve a la sección correspondiente en la que se obtiene la edición de los datos o la eliminación de la cuenta del usuario.

2.4 Editar cuentas de usuarios (*Edit users*)

Mediante esta sección podemos obtener los datos generales como son: nombre completo, nombre de usuario, grupo y base de datos a la que pertenece, inicialmente se presenta el registro **Editar usuario**: (*Edit user:*) en el que se deberá anotar el nombre de usuario de quien se desea obtener la información, una vez que se anota éste se deberá utilizar el botón **Obtener datos** (*Get user data*) para que se lleve a cabo nuestra búsqueda y se realice la edición de los datos. Además de los datos mencionados, se presentan los registros de **Contraseña** y confirmación de la misma, cuando se obtienen los datos del usuario, estos registros permanecerán en blanco de manera que podemos realizar cambios de contraseña desde esta sección, si no se modifican permanecerá la contraseña anterior.

El registro de **Grupos**, despliega la lista de todos los grupos que existen remarcándose el (o los) grupo(s) que pertenece(n) al usuario.

2.5 Control de acceso para usuarios certificados (*List Certificates*)

Cuando se tiene habilitado el servidor con el protocolo SSL, el control de acceso puede utilizar “certificados de usuario”. Cuando el usuario realiza una conexión a un recurso asegurado por el servidor, se puede requerir un certificado de usuario, de manera que el servidor hará que mediante una caja de diálogo el usuario se identifique con su certificado, una vez que el usuario ha proporcionado los datos requeridos, se entiende que ha mostrado un “certificado o identificación” de si mismo, en el que de manera transparente para el usuario se ha reconocido la dirección IP y /o su dominio.

Para conocer los certificados presentados en el servidor, en la sección del encabezado del “Server manager”, se presenta la liga Control de Acceso que contiene la liga **Listar certificados** (*List Certificates*), una vez que se accese a esta sección, podemos conocer una lista general de los certificados que se han presentado a nuestro servidor, como nuestra lista puede ser bastante extensa, se pueden realizar “filtros”, por ejemplo sólo aquellos cuyo inicio de contraseña empiece con la letra *a* de manera que teclearemos *a**, en esta misma sección se presenta un registro para elegir la base de datos de la que deseamos adquirir la información.

2.6 Creación de grupos (*Create group*)

Como se mencionó anteriormente, Enterprise Server permite crear grupos y/o subgrupos en una base de datos de tal manera que, podamos tener mejor organizada nuestra base de datos. Para crear un grupo se debe anotar el nombre del grupo en el registro **Nuevo Grupo** (*New group*), cabe la posibilidad de permitir que ese grupo forme parte de un grupo existente, para esto se presenta una lista en el registro **Este grupo es parte de:** (*This group is part of:*), posteriormente se elegirá la base de datos de la que se desea que forme parte y si es el caso se deberá anotar la contraseña de uso de la base de datos en el registro correspondiente que aparece.

Cuando se requiere que el grupo no forme parte de ninguno otro, en la opción “Este grupo es parte de:” existe la opción **NINGUNO** (*NONE*), de manera que si se elige esta opción, el grupo creado será independiente de cualquier otro.

2.7 Eliminar grupos (*Remove group*)

Para eliminar grupos que se hayan creado, existe esta sección en la que se debe anotar el nombre del grupo a eliminar, la base de datos a la que pertenece y si es el caso la contraseña de uso de la base de datos.

2.8 Edición de grupos (*Edit groups*)

Una vez que tenemos grupos creados y deseamos obtener datos generales de éstos como son: base de datos a la que pertenece, grupos que dependen de él así como los usuarios que lo conforman, en esta sección se permite llevar a cabo un proceso para tal efecto. Se presenta una forma de registro en la que se debe elegir la base de datos a la que pertenece el grupo que se desea editar, y si es el caso la contraseña de uso de la misma, se elige el grupo de la lista que presenta el registro Grupo: y se oprime el botón **Obtener datos:** (*Get group data:*). Cuando se presentan los datos en el registro **Grupos** (*Groups:*), se presenta una lista general de los grupos de la base de datos, los grupos que dependan de aquél que se pidió editar permanecerán resaltados (negrillas). Sucederá lo mismo en el registro **Usuarios:** (*Users members:*), en donde se presenta una lista general de usuarios en la que estarán resaltados aquellos que lo conforman.

2.9 Restricciones de acceso (*Access Restriction*)

Una vez que se mantiene un control de usuarios, Enterprise Server permite restringir el acceso de éstos a recursos del servidor.

En esta opción, se presenta una forma de registro en la que se debe especificar en el registro **Elegir recurso a limitar:** (*Editing*) el recurso al que se restringirá su acceso, en caso de querer realizar una búsqueda en los recursos existentes, se tiene el botón **Explorar** (*Browse*), que al oprimirse presentará un formato en el que se despliega en forma de lista todos los recursos del servidor, o si es el caso podemos utilizar el botón **Agregar todo** (*Wildcard*).

Ya que tenemos elegido el recurso que se restringirá, enseguida se presentará el estado del recurso, es decir, si se encuentra habilitado o deshabilitado su servicio de restricción, aunque por omisión, ningún recurso del servidor se encuentra restringido. Para habilitar el servicio de Control de Accesos, será necesario seleccionar el recurso y oprimir el botón **Habilitar el servicio de restricciones** (*Turn on access control*), una vez que se oprimió éste, se presentará una forma en la que se encuentra una sección como la que se ve a continuación:

Access Type	Action		Edit
Read	<input type="radio"/> Allow	<input checked="" type="radio"/> Deny	Permissions
Write	<input type="radio"/> Allow	<input checked="" type="radio"/> Deny	Permissions

En este esquema podemos ver que se presentan las opciones de restricción de lectura y/o escritura. Lectura se refiere a los métodos de HTTP: GET, HEAD, POST, INDEX, escritura se refiere a: PUT, DELETE, MKDIR, RMDIR, MOVE, al aplicar estas restricciones cualquier acceso a ese recurso será negado.

Una vez que se ha hecho la elección de restricción para escritura y/o lectura, se deben elegir aquéllos que serán la excepción para esta restricción.

Cuando se elige la opción **Permitir** (*Allow*) o **Negar** (*Deny*) y se oprime el botón **Permisos** (*Permissions*), se presenta en pantalla una forma de registro en la que se deberá especificar aquéllos que serán las excepciones, es decir que se ha elegido el permiso ya sea de lectura o escritura y al editar los permisos podremos hacer que para determinados usuarios este restricción no se cumpla.

Por lo tanto si se elige Permitir, en la opción de Permisos se podrán especificar aquéllos que les será negado el recurso, y si se elige Negar se especificarán aquéllos que no entran en esta restricción, es decir, con acceso a los recursos.

Cuando se ha editado la página del botón Permisos, se podrán especificar ciertas restricciones, para negar el acceso a usuarios por dirección IP, se indica la dirección IP y si necesita agregar direcciones se deben separar éstas por "comas" (,).

En caso de ser necesario, Enterprise Server también permite realizar restricciones por nombre de dominio, pero cabe hacer notar que en caso de que no esté disponible el DNS o que se encuentre inhabilitado desde este servidor, no se llevará a cabo tal restricción.

En esta misma página, se incluyen más registros en los que se podrá especificar el acceso por grupo y/o nombre de usuario.

Se incluye también un registro para poder modificar la página que se hace presente cuando se solicita el recurso que se restringió y no se tiene la autorización requerida, en el registro se deberá anotar la ruta de la página que se desplegará, ésta podrá ser en formato HTML o un archivo CGI.

2.10 Manejo de bases de datos (*Manage users databases*)

Esta opción permite crear y eliminar bases de datos del servidor WWW. Para mantener seguridad en el manejo de las mismas, se proporciona la opción de utilizar una contraseña para su administración. Una vez creada la base de datos, podemos realizar cambios en este registro.

Crear y eliminar bases de datos, esta opción permite crear y eliminar bases de datos del servidor WWW, para mantener seguridad en el manejo de las mismas se proporciona la utilización de un nombre de usuario, para el manejo de la misma. Una vez creada la base de datos, podemos realizar cambios en este registro.

2.11 Importación de usuarios (*Import users*)

Enterprise Server permite utilizar una base de datos de usuarios existente, a este proceso lo llamaremos "importación de usuarios". Para poder realizar este proceso, la base de datos existente, debe seguir cualquiera de los siguientes formatos: texto o NCSA. La forma que deben seguir éstos es:

```
usuario1:contraseña1  
usuario2:contraseña2
```

En caso de que en la base de datos a importar se tenga almacenado el nombre completo del usuario, deberá seguir el siguiente formato:

usuario1:contraseña1:x:x:Nombre completo del usuario1

Si la base de datos a importar contiene algún número de usuario y/o número de grupo estos serán reemplazados según el formato anterior por los caracteres x:x respectivamente.

La sección principal de esta liga presenta en pantalla un registro en el que se debe anotar la ruta completa del archivo en el que se encuentre la base de datos a importar.

Cabe destacar que si la base de datos a importar esta en formato texto se debe elegir la opción de **Encriptamiento de contraseña** (*encrypt password*), si se trata de una base de datos de NCSA las contraseñas ya están encriptadas por lo que no será necesario utilizar la opción mencionada, además se tiene la opción de permitir **Almacenar los nombres completos** (*Extract full user names*) de los usuarios (si se tienen). Una opción que finalmente se presenta indica la posibilidad de duplicar o **Renombrar cuentas de usuarios existentes** (*Overwrite existing users*), es decir, que si se encuentra un nombre de usuario idéntico en la base de datos a importar al de la base de datos instalada en el servidor, éste será duplicado, o reemplazado según se haya elegido en el momento de la importación.

3. ENCRIPCIÓN

En esta sección, Enterprise Server, hace uso de SSL, protocolo de seguridad que permite al servidor manejar llaves públicas y privadas, además de incluir herramientas para el mantenimiento de las llaves creadas.

3.1 Encendido/apagado (*On/Off*)

En esta sección aparece una forma de registro en la que se elige si el servidor utilizará el protocolo SSL, es decir, que la información transmitida será encriptada. En esta misma forma de registro, se debe incluir el **Número de puerto** (*Port:*) que se utilizará (puede ser el mismo que se utiliza sin SSL), se incluyen además registros para especificar la ruta en la que se encuentra la **Contraseña de la llave** (*key file:*) y el **Certificado correspondiente** (*Certificate file:*), permite utilizar SSL 2 y SSL 3 .

3.2 Opciones de seguridad (*Security preferences*)

En este apartado se debe especificar si se requiere el certificado del cliente, además de elegir el o los modos de encriptación de SSL2 (RC4 o RC2), también se debe especificar para SSL3 (RC4 y MD5, RC4 y MD5 o sólo MD5)

3.3 Generación de llaves (*Generating key file*)

Desde este módulo se permiten generar llaves para utilizarlas con el protocolo SSL, aparece un registro en el que se indicará el nombre y la ruta en la que se almacenará el “archivo llave” generado.

3.4 Cambiar contraseña (*Change key password*)

Una vez creada la llave, podemos modificar la contraseña que se indicó cuando se generó ésta, para esto, se requiere anotar la **Ruta en la que se encuentra la llave** (*Key File path*), una vez que se indicó la ruta, se requiere la **Contraseña anterior** (*Old password*) y enseguida la

Contraseña que se utilizará (*New password*), se requiere **Teclear nuevamente la contraseña anterior** (*Again*).

3.5 Certificados (*Request Certificate*)

Enterprise Server, desde esta sección permite enviar una solicitud ante la CA (*Certification Authority*), en éste se debe anotar la dirección electrónica de la CA que se haya elegido, además de indicar si se trata de una certificación nueva o si será una renovación. Enseguida habrá que anotar la ruta del “archivo llave” local, se utiliza para que el servidor encripte la información y la envíe a la CA con una “llave pública”. A partir de este registro se deben anotar datos generales como son el número telefónico de la persona que se responsabilizará del uso del certificado, organización que lo representa, ciudad, localidad y estado.

3.6 Instalación de Certificados (*Install Certificates*)

Una vez que se ha generado el certificado, éste debe instalarse en el presente apartado, se permite instalar cualesquiera de los siguientes tipos de certificados: **Certificado propio** (*This server*), **Certificado CA** (*Server Certificate Chain*), **Certificado autorizado** (*Trusted Certificate Authority*). Posteriormente se anotará el nombre del certificado (nombre que aparecerá en la lista de certificados de los navegadores), también se incluye un registro en el que se debe anotar la base de datos del certificado que por omisión es *ServerCert*.

3.7 Manejo de certificados (*Manage Certificates*)

Cuando se han instalado los certificados, Enterprise Server permite editar o borrar los certificados instalados en el servidor, mediante esta sección en la que aparece el nombre, tipo y fecha de expiración del o los certificados instalados.

4. PROGRAMAS

Una herramienta del servidor muy importante es el uso de la interacción con programas CGI's así como manejadores de bases de datos, mismos que contiene este módulo.

4.1 Directorio CGI (*CGI directory*)

Enterprise Server, permite considerar un directorio específico para hacer uso sólo de programas CGI's, es decir, que cualquier archivo que se encuentre en este directorio se considerará como CGI, en la etiqueta **URL** (*URL prefix*) se debe anotar la liga con la que reconocerá el CGI, enseguida será la ruta específica del directorio a considerar como **CGI** (*CGI directory*).

4.2 Archivos CGI (*CGI file type*)

En esta sección tenemos la posibilidad de determinar una extensión específica para considerarla como archivo CGI, independientemente de la ubicación del archivo. En el recuadro que aparece se indica la extensión y se habilita la opción **Archivo CGI: (CGI as a file type)**.

4.3 Java (*Java*)

En el servidor se incluye una herramienta para compilar los programas elaborados con el lenguaje Java, basta con **Habilitar el intérprete** (*Activate Java Interpreter*) y anotar el directorio en el que se encontrarán los *applets*.

4.4 LiveWire (*LiveWire*)

Enterprise Server permite interactuar con manejadores de bases de datos, para esto basta con **Habilitar el servicio** (*Activate the LiveWire application environment*) y si se desea **Anotar contraseña de administración para este servicio** (*Require administration server password for LiveWire Application Manager*).

4.5 ShellCGI (*ShellCGI*)

Este módulo permite ejecutar aplicaciones utilizando archivos que se encuentran asociados en el sistema (Windows NT). Por ejemplo si al servidor se le pide ejecutar un archivo con extensión *.pl*, el servidor utilizará el archivo asociado a esta extensión, por lo que si se encuentra asociado *c:binperl.exe*, el servidor ejecutará el archivo haciendo uso de *binperl.exe*.

Para declarar un archivo, bastará con crear un directorio específico y anotar en **URL** (*URL prefix*) el nombre como se desea que sea reconocido este directorio, posteriormente en **Directorio** (*Directory file*) se anotará la ruta en la que se encontrarán los archivos.

5. SERVER STATUS

En general, esta sección proporciona información a detalle de los accesos que se realizan al servidor, además de indicar el estado de actividad de procesos que se están llevando a cabo en la sección.

5.1 Control de accesos (*View Access Log*)

El servidor hace uso de un archivo llamado **access** en éste, se lleva un registro de los accesos que se han realizado desde que se instala el servidor hasta la fecha en la que se realice el almacenamiento de este archivo, este archivo tiene el siguiente formato:

```
132.248.156.186 - - [01/Jul/1997:13:53:43 -0600] "GET /indice.html HTTP/1.0" 200 6878
```

Este línea de texto indica la dirección IP de la computadora que realizó una conexión con el servidor, enseguida se encuentra registrada la fecha de la misma en el formato día/mes/año:hora:min:seg y podemos ver que también se registra el nombre del recurso del que se hizo uso además de anexar el protocolo utilizado para la acción anterior, para identificar el tipo de acceso que se realizó se encuentran los dos últimos registros de dígitos que indican los procesos que se llevaron a cabo.

Para limitar la lista de registros tenemos la opción **Limitar lista a:** (*Only show entries with*) con esta opción podemos utilizar "filtros" con los que indicaremos que sólo se editarán aquéllos que cumplan con las anotaciones que se hagan aquí.

5.2 Errores de acceso (*View errors log*)

Cuando se registran errores de acceso en el servidor de WWW, éstos se registran automáticamente en el archivo **errors**, que se encuentra en el directorio principal de Enterprise Server, se permite listar el contenido de **errors** mediante una página de WWW, para llevar a cabo esto, se hace presente la liga **Ver errores de acceso** (*View error log*) de esta sección.

En esta sección se presenta una forma con los siguientes registros:

- **Número de errores a ver** (*Number of errors to view*). En este registro como su nombre lo indica, se anota el número de errores que se desean ver (por omisión es 25, aunque podemos modificar esta cantidad)
- **Editar accesos sólo con** (*Only shows entries with*). Al hacer uso de ésta, podemos limitar la lista de accesos que se va a editar en la página de WWW mediante la utilización de "filtros" que indiquen las limitantes de la lista, estos filtros pueden ser: *132.** con la que indicamos que deseamos poder ver sólo aquéllos cuya dirección IP empiece con 132, cuando tenemos el DNS activado podemos realizar la acción anterior con el nombre que le corresponda a 132. Por omisión se presenta en blanco esta opción

Una vez que se ha llenado la forma con las opciones mencionadas anteriormente y han sido aceptadas, podremos ver una lista de los registros que cumplen con las especificaciones anotadas y que han causado un mensaje de error en el servidor. La lista se presenta con el siguiente formato:

```
[30/Jun/1997:15:40:32] warning: for host 132.248.156.230 trying to GET
/testing123/img/trans/fo1_1.gif, send-file reports: can't find
C:/enterp/docroot/testing123/img/trans/fo1_1.gif (ERROR_FILE_NOT_FOUND)
```

es decir:

[Fecha en la que se llevó a cabo la conexión (día /mes/año:hora:min:seg)] tipo de alerta (warning) · dirección IP desde la cual se realizó la conexión (en caso de que se encuentre habilitado el DNS, aparecerá el nombre de la máquina en lugar de su dirección IP) recurso a acceder (trying to GET): /ruta_del_recurso_a_accesar, tipo de error y la ruta que está causando éste (send file reports: can't find) c:/enterp/docroot/testing123/img/trans/fo1_1.gif (ERROR_FILE_NOT_FOUND)

En el archivo errors también se registra el servicio de inicio, reestablecimiento, y de interrupción del servidor.

```
[30/Jun/1997:15:42:14] info: Suspend Httpd Service
[30/Jun/1997:15:42:14] info: successful server startup
```

5.3 Monitoreo de procesos (*Monitor current activity*)

Para informarnos de los procesos que se están ejecutando en el servidor, en la liga **Actividad del servidor** (*Monitor current activity*), podemos hacer uso de un programa que al ejecutarse proporcione datos como: número de procesos que se están llevando a cabo actualmente en el servidor y así determinar si es necesario incrementar el número de procesos que el servidor puede ejecutar.

5.4 Archivo de accesos (*Archiving log files*)

Enterprise Server genera un archivo llamado *access.log* en el que se almacena un registro de las conexiones al servidor vía WWW, lo anterior se realiza de manera automática, sin embargo podemos hacer que este archivo se maneje de manera independiente, es decir, que el contenido de *access.log* se almacene en otro archivo, esta opción se hace presente en la liga **Archivo de accesos** (*Archiving log files*) en ésta se presenta una forma de registro en la que podemos especificarle que se almacene periódicamente además de especificar el nombre del archivo, por omisión se almacenará en el directorio de registros (*logs*).

5.5 Guardar preferencias (*Log preferences*)

Dado que Enterprise Server permite almacenar los accesos que se han realizado al servidor, a su vez puede crear un tipo de almacenamiento de acuerdo a las necesidades requeridas, por lo que es capaz de llevar un control de accesos por archivos, directorios específicos o por todos los recursos del servidor (este último se encuentra por omisión). En esta página se presenta una forma de registro en la que inicialmente, pide elegir el recurso del que se desea llevar el registro de accesos, llámese archivo, directorio o recursos generales del servidor (todo), la elección se puede hacer mediante la elección de los botones **Explorar** (*Browse*) o **Agregar todo** (*Wildcard*), una vez que se ha elegido el recurso, se debe confirmar el registro del mismo.

Una vez que se eligió el recurso, se debe anotar la ruta y el nombre de archivo en el que se almacenarán los accesos, cabe mencionar que se debe anotar la ruta específica del archivo. Posteriormente, se pide elegir, si el registro se realizará por DNS, o por Direcciones IP, ya que se ha seleccionado alguna de estas opciones, se debe elegir el formato del archivo que se almacenará, las opciones de formato que presenta son básicamente dos: **a)** Utilizar un formato común o **b)** Elegir los elementos que conformarán el archivo (los elementos que presenta son: Hostname del cliente (*Client hostname*), Nombre del usuario (*Authenticate username*), Fecha (*System date*), Petición completa (*Full request*), Estado de la petición (*Status*), Tamaño del recurso requerido (*Content length*), Encabezado (*Header refer*), Método, URL mediante el que se accesa (*Query string of the URI*) y Protocolo).

Podemos indicar que no registre el acceso de algunos clientes, ésto se indica por dirección IP o dominio en el registro **No registrar acceso de clientes** (*Do not log client accesses from*).

5.6 Generación de reportes (*Generate Report*)

Una vez que tenemos un archivo con el registro de accesos, Enterprise Server permite en esta sección, generar reportes de lo que se encuentra almacenado en el archivo *access*. Este tipo de informes pueden ser generados para cualquier servidor que se encuentre instalado, en tal caso bastará con anotar el nombre del servidor del que se desea hacer un reporte de accesos en el registro **Nombre del servidor** (*Server Name*); tenemos la opción de generar estos reportes en los siguientes formatos **Formato HTML** (*Output in HTML*) **Formato texto** (*Output in text/plain*), posteriormente se debe **Elegir el archivo** (*Select the log file you want to analyze*) del que se desea generar el reporte así mismo se anotará el **Nombre del archivo** (*Output file name*) con el que se desea almacenar este reporte, si se deja este espacio en blanco (sin nombre), los resultados se presentarán sólo en pantalla. Para complementar el reporte, tenemos opciones de almacenar de acuerdo a el estado de los registros, éstos son: 304(No modificado), 302(Redireccionados), 404(No localizados), 500(Error del servidor), además de incluir el total de número de ligas del servidor que se han accedido, número de

kilobytes que el servidor ha transferido desde la activación del control de accesos y número de kilobytes que se han almacenado en el caché del cliente. Podemos incluir un registro de determinado número de usuarios que han accedido más el servidor, las ligas que más se han frecuentado del servidor. Una vez que se han elegido las opciones anteriores, se debe anotar el orden de generación de los siguientes elementos: **Generación de totales, Estadísticas generales y Generación de listas.**

La generación de reportes se realiza con el archivo *flexanlg*, que se encuentra en el directorio *ruta_de_servidor\extras*, el cual se puede ejecutar desde la línea de comando en el modo MS-DOS con opciones de los siguientes parámetros:

```
flexanlg [-n nombre] [-x] [-r] [-i file] [-o file]
```

en el que:

- n Nombre del servidor
- x Salida en formato HTML
- r Registro por direcciones IP
- i Nombre del archivo a reportar
- o Nombre del archivo en el que se almacenarán los resultados

6. CONFIGURACIÓN DE ESTILOS

En este módulo, se permite aplicar un conjunto de opciones específicas de configuración a determinados archivos o conjunto de ellos.

6.1 Creación de un estilo (*New style*)

Para su creación, se deberá anotar en el registro el **Nombre del Estilo:** (*Name Style:*) y una vez que se oprima el botón <OK> aparecerá una página en la que se ve el nombre del estilo recién creado y las posibles ligas a las que se les puede asignar, éstas son:

- Pie de página
- Control de Accesos
- Ejecuciones de archivos HTML
- Archivos de accesos

Cabe mencionar que se deberá seguir el proceso anterior cada vez que se desee elegir un estilo para cada liga presentada.

6.2 Eliminar un estilo (*Remove style*)

Antes de eliminar un estilo de configuración, se deberá colocar la liga a la que se aplicó en la opción de **NINGUNO** (*NONE*), esto se debe a que si se elimina el estilo y la liga se queda configurada, se enviará un mensaje de error de configuración del servidor cada vez que se accese a ella. Ahora, para eliminar un estilo, desde esta sección se elige el nombre del mismo del registro **Estilo** (*Style*).

6.3 Edición de un estilo (*Edit style*)

Una vez que se ha creado un estilo, la edición del mismo se realiza desde esta sección, en donde se elige el registro a editar y se oprime el botón <Editar> (*Editing style*). Se podrá

realizar la modificación de alguna liga desde esta misma sección, bastará con hacer la selección de la liga de la lista que se hace presente.

6.4 Aplicar un estilo (*Assign style*)

Para especificar un estilo, en esta sección, se anota el **URL** (*URL prefix wildcard*), en donde se aplicará y se selecciona el **Estilo** (*Style*).

6.5 Listar estilos (*List assignments*)

Después de haber creado algún estilo en particular, en esta sección se muestra una lista de los estilos que han sido aplicados a algunas de las ligas que se hacen presentes en el servidor, esta lista se presenta con su correspondiente nombre y una liga de <Edición> (*Edit*) para cada uno de los que ahí se encuentren.

7. MANEJO DEL CONTENIDO DEL SERVIDOR

En las opciones que ofrece esta sección, podemos realizar consultas y modificaciones de algunos directorios de configuración del servidor, además de poder redireccionar documentos, crear mensajes dinámicos en ellos y crear servidores anexos al servidor principal.

7.1 Ubicación del directorio principal del servidor (*Primary Document Directory*).

Enterprise Server permite establecer una ruta específica en el servidor a partir de la cual se declaran los archivos que se encontrarán a disposición en el WWW, esto es, un *directorio público o directorio principal para WWW*.

El establecimiento de un directorio principal resulta ser práctico cuando se requiere cambiar de directorio de uso al servidor, dado que los documentos contienen ligas con referencia a rutas específicas de otros documentos, estas ligas no tendrán que ser modificadas ya que sólo hacen referencia al directorio principal que se tenga activado.

En esta sección podemos realizar las modificaciones necesarias en el registro que se presenta, para indicar la ruta a partir de la cual se desea que los documentos se encuentren disponibles vía WWW.

7.2 Creación de directorios adicionales (*Additional Documents Directories*)

Una vez que se tiene creado un directorio principal, Enterprise Server permite la utilización de directorios que se encuentren en una ruta diferente del directorio principal del servidor, esta sección permite crear este tipo de directorios teniendo la posibilidad de proporcionar un nombre con el cual se identificará la ruta establecida. Se presenta un registro en el que se debe especificar:

Identificador (*URL prefix*), en éste, se indica el nombre con el que se requiera que los usuarios reconozcan el directorio.

Ruta (*Path*), se debe indicar la ruta específica que tiene el directorio en el servidor.

Estilo (*Style*), se refiere al estilo que se desee utilizar para la configuración de este directorio.

7.3 Administración de archivos remota (*Remote File Manipulation*)

Enterprise Server permite que los usuarios modifiquen archivos en el servidor siempre y cuando tengan autorización para ello, para esto, se debe elegir en el registro que se hace presente en esta sección el directorio o archivo al cual se le permitirá realizar cambios. Cabe mencionar que esta opción será inválida si el usuario no tiene autorización de escritura (este permiso se realiza desde "Control de Acceso").

7.4 Indexación de documentos (*Documents Preferences*)

Cuando se accesa a un servidor WWW, se presenta un archivo de inicio del directorio público del servidor, este archivo es designado por el administrador del servidor, aunque por omisión este archivo es: "index.html". Cuando no se encuentra éste, el servidor genera un archivo índice en el cual se hace una descripción general de los archivos que están contenidos en este directorio, Enterprise Server permite modificar el nombre de este archivo, en esta sección se presenta un registro para modificarlo éste es **Archivos índices** (*Index Files*), en éste, se debe especificar el nombre del archivo que se desea indicar como archivo índice. Cuando se requiera incluir otro archivo que considere como índice, bastará con separarlos con una "coma" (,).

Como una herramienta de apoyo durante la generación de este archivo, Enterprise Server permite modificar el tipo de archivo índice que se generará, existiendo para ello, dos tipos de formato:

1. *Fancy*. En este tipo de formato, se genera una lista en forma de árbol, en la cual se describe el nombre y tamaño del archivo así como una breve descripción general de su contenido (opcional).
2. *Simple*. Para este formato, se genera una lista sólo con los nombres de los archivos.

Existe una opción en la que no se permite generar un archivo índice, esta opción es **Ninguno** (*None*), es decir que cuando se accesa a la página de inicio y no existe el archivo de inicio especificado, el servidor enviará un mensaje de error y no generará el listado de archivos.

Cuando tenemos un archivo (sin que éste sea precisamente el archivo índice), que deseemos que se considere "Home Page" (página de inicio general del servidor), podemos establecer su ruta de ubicación en el registro "Home page:". Desde cualquier subdirectorío en que se encuentren el cliente, cuando en el navegador se utilice la liga de "Home"(Casa) de las páginas, nos llevará a este archivo.

Si el servidor no identifica el tipo de archivo que va a utilizar en el formato de índices hace uso del formato: "texto plano", éste es el formato por omisión.

7.5 Redirección de archivos (*URL Forwarding*)

Cuando se realizan modificaciones de ubicación de nuestros archivos, se tienen que modificar las ligas en las páginas del servidor, para evitar la tarea de reubicación de éstas, Enterprise Server permite que cuando se accese a un archivo, éste de inmediato realice la conexión a otro sitio que el administrador le especifique en el registro de esta sección.

Para llevar a cabo este procedimiento, en el registro **URL** (*URL prefix*), se anota la ruta específica del archivo que va a ser direccionado (archivo origen), después será la ruta específica del archivo que lo representará (archivo destino).

7.6 Servidor de software virtual (*Software Virtual Server*)

Enterprise Server permite “generar servidores” a partir del que se tiene instalado y con sólo una dirección IP, esto es, que pueda responder a `http://servidor1.iingen.unam.mx` y `http://servidor2.iingen.unam.mx` de manera independiente. Bastará con agregar el segundo nombre al DNS y crear un directorio propio para el segundo (el primero será el que se encuentra ya instalado). Una vez que se encuentre en esta sección, se deberá anotar el nombre con el que se desea ser reconocido el servidor en el registro **http:** y enseguida se anota la ruta a partir de la cual se encontrarán los archivos para este servidor.

7.7 Servidor de hardware virtual (*Hardware Virtual Server*)

Enterprise Server permite que el servidor responda a diversas direcciones IP (hasta 256), esto será como tener “servidores de hardware virtuales”, éstos compartirán la misma configuración, por lo que si se ha configurado el original con uso del protocolo de seguridad, el servidor que se dé de alta por esta sección también hará uso del protocolo, entonces para tener un “servidor de hardware virtual”, deberá anotarse los datos correspondientes en los registros **Dirección Ip:** (*IP Address*) y **Directorio principal:** (*Document root*).

7.8 Caracteres Internacionales (*International Characters*)

El conjunto de caracteres de un documento se determina en parte por el lenguaje en que este se encuentra escrito, sin embargo, mediante esta sección, se pueden modificar los caracteres a un archivo específico, un conjunto de archivos o hasta un directorio.

El conjunto de caracteres que son reconocidos comúnmente por algunos navegadores son:

- us-ascii
- iso-2022-jp
- iso-8859-1

Algunos alias reconocidos para *us-ascii* son:

- ascii
- us
- ansi_x3.4-1968
- ansi_x3.4-1986

Una vez que se ha determinado cambiar un conjunto de caracteres, en esta sección, se deberá elegir el recurso a cambiar y enseguida se anota el nuevo conjunto de caracteres que se haya elegido de entre los que existen.

7.9 Pie de página (*Document footer*)

Enterprise Server proporciona esta sección en la que podemos anexar líneas de texto a cualquier documento (en formato HTML) del servidor, éstas se ubicarán en el pie de página del documento.

Para llevar a cabo lo anterior, se presenta un registro en el que debemos elegir el archivo que contendrá el pie de página, la elección de el archivo se realiza mediante la selección de una lista de los archivos del servidor que se presenta una vez que se oprima el botón **Explorar** (*Browse*). Se tiene la posibilidad de anexar la fecha en la que se realizó el pie de página, ésta se anexa al texto con sólo teclear : **LASTMOD:** .

8. CATÁLOGO DE DOCUMENTOS

De entre los servicios que ofrece Enterprise Server, se encuentra el “Catálogo de Documentos”, que permite llevar un registro de los archivos que se encuentran en el servidor, éste se puede visualizar mediante una página en el WWW, proporcionándonos además información para realizar estadísticas.

8.1 Encendido/Apagado (*On/Off*)

Para hacer uso de este servicio, basta con oprimir el botón “Encender” (*Server On*) que se presenta en este módulo, una vez realizado lo anterior, se restaurará automáticamente el servicio de administración del servidor y se podrá acceder hacia la dirección electrónica del servidor anexando el directorio /catalog, por lo que se escribirá:

<http://tabasco.iingen.unam.mx:9000/catalog>

8.2 Configuración (*Configure Catalog Agent*)

Una vez que se tiene el servicio de catálogo habilitado, se puede administrar de manera que se indique el directorio a partir del cual se desea ejecutar este servicio, lo anterior podemos anotarlo en el recuadro que aparece después de la dirección electrónica del servidor, en este recuadro se anota un directorio específico (/nombre_dir), en caso de haber dejado el recuadro en blanco, el servicio iniciará a partir de la página de inicio (index.html) y continuará con las ligas relacionadas a esta.

También en este mismo módulo, aparece el recuadro **Velocidad** (*Speed*), cuando tenemos habilitado el servicio de catálogo, éste realiza las búsquedas necesarias en los directorios especificados en la configuración del mismo, la velocidad a la que se realiza esta búsqueda implica el número de peticiones que el servidor enviará, a continuación se muestra una tabla indicativa que se debe considerar de acuerdo a la computadora que se utiliza como servidor para configurar la velocidad para el servicio de catálogo.

Configuración de Velocidad	Número de respuestas	Retardo (segundos)
1	1	60
2	1	30
3	1	15
4	1	5
5	1	1
6	2	0
7	4	0
8	8	0
9	12	0
10	16	0

Tabla B-1

8.3 Control automático (*Schedule*)

Las listas generadas para la publicación de la página de este servicio, requieren de un proceso de indexación (este se realiza automáticamente cuando se inicializa el servicio por primera vez), por lo que si se desea mantener actualizada la página, se deben llevar a cabo indexaciones periódicas. En este módulo se puede configurar la indexación de acuerdo al día y hora que se desee, para esto se presenta el recuadro **Hora: (Time)**, en el cual se presentan listas para elegir <HORA> (se presentan las 24 horas del día) y <MINUTOS> (se presentan intervalos de 10 minutos) y el recuadro **Días: (Days)** en el que tenemos presente todos los días de la semana para que así sea posible realizar la indexación cualquier día de la semana con la opción de poder elegir más de un día y a cualquier hora.

Cabe mencionar que este servicio de control automático inicialmente se debe activar para poder utilizar las configuraciones anteriores, para tal efecto, basta con elegir **Activar (Activate)** del recuadro **Control automático (Schedule)** desde esta misma sección.

8.4 Control manual (*Manually Control*)

Las indexaciones que se utilizan puede ser requeridas de vez en cuando, es decir, sólo si las modificaciones llevadas a cabo en el contenido de los directorios son relevantes, para esto sólo será necesario indexar “manualmente” los archivos, por lo que existe este módulo en el que se hacen presentes los siguientes temas:

- **Inicia (Start)**, inicia el servicio de indexación de acuerdo a las configuraciones establecidas en la sección: Configuración
- **Pausar (Stop)**, detiene el servicio de indexación de archivos pero continúa recopilando información de los documentos (esta información generada puede ser utilizada para cuando se indexe nuevamente la información)
- **Detener (Stop enumeration)**, detiene el servicio y no logra recuperar información para una nueva indexación, alcanza a eliminar los archivos temporales que fueron generados durante la indexación detenida
- **Detener inmediatamente (Kill)**, detiene el servicio rápidamente de manera que no logra recuperar información para utilizar los archivos generados en esta indexación ni elimina los archivos temporales que se generaron durante la misma

9. CREACIÓN DE UNA HERRAMIENTA DE BÚSQUEDA

Enterprise Server contiene una herramienta que permite a los usuarios realizar búsquedas de la información contenida en el servidor WWW, esta herramienta se puede utilizar con las opciones presentadas por omisión o se puede modificar de manera que se adecue a las necesidades de los usuarios.

9.1 Generación de colecciones (*New Collection*)

Para hacer uso de esta herramienta se deben cubrir ciertos requerimientos, inicialmente se debe dar de alta una ruta en la que se indique el directorio en el que se encuentre la información que estará disponible para el servicio de búsquedas, es decir, que se creará un “directorio adicional” desde “Manejo del contenido del servidor”, posteriormente se deben “generar colecciones”, esto es, agrupar archivos contenidos en el servidor para formar así grupos de ellos de manera que podamos manejar toda la información contenida y disponible del servidor en agrupaciones de archivos para el usuario, para esto se anota el **Nombre de la**

colección (*Collection name*) el (o los) nombre(s) que aquí se anoten, aparecerán en la página del servicio de búsquedas del servidor, por lo que se aconseja sea(n éstos) lo mas identificable posible para el usuario), una **Descripción** (*Description*) (ésta servirá como referencia sólo para el servicio de administración ya que no es visualizable en la página de búsquedas), el **Directorio a indexar** (*Directory to index*) (esto es, el directorio del que se tomarán los archivos para formar la colección), además se presenta la opción de incluir los subdirectorios del directorio en curso, se incluye el formato en el que se almacenará la colección de archivos, de entre los cuales se presentan **ASCII** (para archivos de aplicaciones como procesadores de palabras, hojas de cálculo, etc.) y **HTML** (archivos con formato HTML), finalmente se eligen los archivos que se incluirán en la colección de acuerdo al formato elegido.

9.2 Edición de colecciones (*Edit Collections*)

Una vez que se han creado las colecciones, Enterprise Server permite “manejar” el contenido de las mismas, en esta sección se presenta un registro en el que aparecen en forma de lista todas las colecciones creadas. Cuando se realiza la elección de una colección, aparecen en pantalla datos referentes a ésta como lo es: Localización de la colección de archivos del servidor, nombre que se le designó, descripción (información obtenida cuando se generó la colección), además de la fecha de creación y de última modificación, enseguida aparece un registro en el que se pueden ver todos los documentos que forman parte de la colección. En esta sección podemos llevar a cabo: **Actualización** (*Update*) y **Eliminación** (*Remove*) de colecciones, con el botón de Actualización podemos además de actualizar, anexas archivos de otros directorios y/o eliminar determinados archivos de la colección; si se utiliza el botón de Eliminación se borrará toda la colección. Cuando deseemos utilizar el botón de Actualización para anexas documentos aparece en esta misma sección un cuadro de edición para anotar el nombre del **Directorio o archivo** (*Directory or file*) que se incluirá, si se trata de un directorio, se puede indicar que **Indexe subdirectorios** (*Index Directory*) .

9.3 Manejo de colecciones (*Manage Collections*)

En esta sección se presentan los mismos datos que se describieron en “Edición de colecciones” sólo que se diferencian porque aquí se agregan botones para: **Optimizar** (*Optimize*), **Reparar** (*Repair*) y **Eliminar** (*Delete*), en el primero se realiza una verificación para que no se dupliquen los documentos o para confirmar que se encuentren en el directorio que se especificó al generarse la colección; en el segundo, cuando una colección ha sufrido posibles daños, tenemos la opción de “repararla” (por ejemplo si hubo algún error durante la generación de la misma), finalmente como su nombre lo indica, se eliminará por completo la colección, una vez que se ha eliminado no se podrá recuperar ésta, esto no significa que se eliminarán los archivos de origen de la colección.

9.4 Períodos de indexación (*Timing utility*)

Cuando tenemos generadas las colecciones y se lleven a cabo modificaciones en los archivos, es necesario actualizar las colecciones, para ésto existe esta sección en la que podemos “programar” las indexaciones indicándole el **nombre de la colección** (*Choose collection*) y los **Archivos que se indexarán** (*Reindex documents*), se incluye además el registro para indicar la **Hora** (*At*) y el (o los) **Día(s)** (*Days*) en que se llevarán a cabo las indexaciones.

9.5 Ejecuciones de archivos HTML (*Parse HTML*)

Generalmente un archivo HTML es enviado al cliente con el formato que contiene originalmente del servidor, sin embargo, mediante esta liga, podemos indicarle que el recurso que elijamos de la lista que se presenta mediante el botón **Explorar** (*Browse*), sea capaz de ser modificable, debido al proceso anterior, se activará la etiqueta "exec" que permitirá que un archivo en formato html ejecute cualquier programa localizado en el servidor, cabe mencionar que la activación de este módulo repercutirá en la velocidad de la ejecución de las tareas del servidor. Finalmente se debe elegir el tipo de archivos que utilizarán este servicio (comúnmente son aquellos con extensión *.shtml*).

9.6 Control de archivos (*Version control*)

De entre las utilidades que maneja Enterprise Server existe "Control de archivos", mediante la cual podemos verificar los documentos del servidor, de manera que en un momento dado podamos "bloquearlo" para así realizar las verificaciones y/o modificaciones pertinentes para que ningún usuario pueda trabajar con éste hasta que el servicio de administración del servidor lo "desbloquee", este tipo de deshabilitación permite entonces que 2 o más personas no lleven a cabo modificaciones en un documento. Para llevar a cabo esto, bastará con teclear: `;version=head:lock` después de la dirección electrónica del archivo que será bloqueado.

Si sólo se desea verificar alguna versión específica del documento, se anexará al final de la dirección electrónica `;version=head` o `;version=` con las líneas anteriores se logrará ver la última versión, en caso de saber la versión se anotará `;version=num_version`.

Para poder llevar a cabo modificaciones de escritura en un documento, se tiene que anotar al final de la dirección electrónica `;version=head:update`, si se quiere bloquear el documento se anotará `;version=head:lock:update` cabe mencionar que para esto se lleve acabo, se deberán realizar las modificaciones de acceso de escritura pertinentes.

Para "desbloquear" un documento bastará con anotar `;version=head:unlock`. Si se desea ver las versiones de un documento, bastará con anotar `;version=index` para que se genere una lista de las versiones del documento especificado.

TABLA DE FORMATOS

En algunas secciones como lo son: Restricción de accesos, Configuración de páginas, Ver errores de entradas, etc. , se presenta un registro en el que se anota el recurso que se utilizará o que se desea ver, para esto, se hacen presentes los botones **Explorar** (*Browse*) y **Agregar todo** (*Wildcard*), para facilitar el uso de estas herramientas podemos utilizar las siguientes convenciones que se tienen en Enterprise Server :

Notación	Descripción
*	Se refiere a cualquier caracter.
?	Considera sólo 1(un) caracter que puede ser cualesquiera.
	Expresión OR.
\$	Se refiere a la parte final de una cadena.
[a-z]	Hace referencia a cualquier ocurrencia de cualquier caracter entre <i>a</i> y <i>z</i> .
[^ad]	Se refiere a cualquier caracter excepto <i>a</i> y <i>d</i> .
[abc]	Cualquier ocurrencia entre <i>a</i> , <i>b</i> o <i>c</i> .
*~	Seguida de cualquiera de las notaciones anteriores denega lo que le sigue.

Tabla B-2

Ejemplos :

*.unam.mx	Hace referencia a cualquier cadena que finalice con unam.mx
(iingen fi-p).unam.mx	Se refiere ya sea a: <i>iingen.unam mx</i> o <i>fi-p.unam mx</i>
132.248.15[56]. ???	Considera cualquier cadena que inicie con 132.248.155 o 132.248.156 y termine con cualesquiera 3 caracteres.
.	Cualquier caracter entre un punto.
132.248.*~90.155	Considera cualquier cadena que inicie con 132.248 y termine con 155 excepto aquellas que tengan el 90 como intermedio.