

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

TECNOLOGÍAS PARA LA CREACIÓN
DE SISTEMAS DE APLICACIÓN EN INTERNET

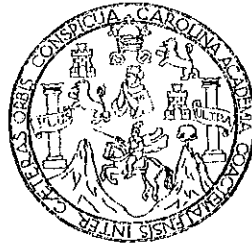
TESIS

PRESENTADA A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MANUEL FERNANDO LÓPEZ FERNÁNDEZ
AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 1,999.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



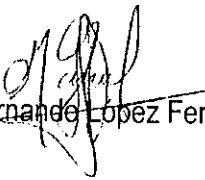
FACULTAD DE INGENIERÍA

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de tesis titulado:

**TECNOLOGÍAS PARA LA CREACIÓN
DE SISTEMAS DE APLICACIÓN EN INTERNET,**

tema que me fuera asignado por la Coordinación de la carrera de Ciencias y Sistemas de la Facultad de Ingeniería, con fecha marzo de 1,998.


Manuel Fernando López Fernández

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Herbert René Miranda Barrios
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Carlos Humberto Pérez Rodríguez
VOCAL III	Ing. Jorge Benjamín Gutiérrez Quintana
VOCAL IV	Br. Oscar Stuardo Chinchilla Guzmán
VOCAL V	Br. Mauricio Alberto Grajeda Mariscal
SECRETARIA	Inga. Gilda Marina Castellanos Baiza de Illescas

TRIBUNAL QUE PRACTICÓ EL EXAMEN

GENERAL PRIVADO

DECANO	Ing. Herbert René Miranda Barrios
EXAMINADORA	Inga. Ligia Maria Pimentel Castañeda de Marroquín
EXAMINADOR	Ing. Ricardo Alfredo Girón Solórzano
EXAMINADOR	Ing. Rolando Haroldo Alonzo Ordoñez
SECRETARIA	Inga. Gilda Marina Castellanos Baiza de Illescas

DEDICATORIA

El presente trabajo se lo dedico a:

- Dios, porque a él le agradezco la oportunidad de vivir y ser feliz.
- Mis padres Víctor Fabricio, Juana Yolanda, mis hermanos: Leonel, Edgar, Víctor y demás familia.
- A Damaris, por toda su paciencia y amor.
- A mi país Guatemala y en especial a mi bella Antigua
- A la gloriosa Universidad de San Carlos de Guatemala

AGRADECIMIENTO

- A Dios, por darme la vida y la oportunidad de completar mis estudios.

Un agradecimiento muy especial a las siguientes personas, quienes invirtieron tiempo y esfuerzo durante la realización de este trabajo:

- A mis padres Víctor Fabricio y Juana Yolanda por todo el apoyo que me brindaron.
- Al Ingeniero Jorge Luis Álvarez Mejía: muchas gracias por su apoyo y orientación.
- A Damaris, por toda su ayuda, amor, apoyo y comprensión.

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
Regional de Post-grado de Ingeniería
Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

Guatemala, octubre de 1999

Ingeniero
Jorge Luis Álvarez Mejía
Coordinador de la carrera,
Ingeniería en Ciencias y Sistemas.

Respetable Ingeniero

Por este medio me permito informarle que asesoré y revisé el trabajo de tesis titulado **TECNOLOGÍAS PARA LA CREACIÓN DE SISTEMAS DE APLICACIÓN EN INTERNET**, elaborado por el estudiante **MANUEL FERNANDO LÓPEZ FERNÁNDEZ** y, a mi juicio, la misma cumple con los objetivos propuestos para su desarrollo.

Sin otro particular, me suscribo atentamente,



Ing. Jorge Luis Álvarez Mejía

Asesor

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



ACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
Regional de Post-grado de Ingeniería
Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

Guatemala, octubre de 1999

Ingeniero
Jorge Luis Álvarez Mejía
Coordinador de la carrera,
Ingeniería en Ciencias y Sistemas.

Ingeniero Álvarez

Por este medio me permito informarle que revisé el trabajo de tesis titulado
TECNOLOGÍAS PARA LA CREACIÓN DE SISTEMAS DE APLICACIÓN EN INTERNET,
elaborado por el estudiante **MANUEL FERNANDO LÓPEZ FERNÁNDEZ** y, a mi juicio, la misma
cumple con los objetivos propuestos para su desarrollo.

Sin otro particular, me suscribo atentamente,

Ing. Ricardo Alfredo Girón Solórzano

Revisor

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
Regional de Post-grado de Ingeniería
Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

Guatemala, octubre de 1999

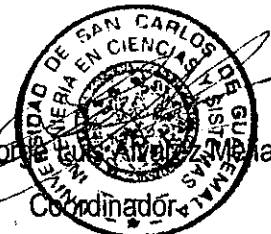
Ingeniero
Herbert René Miranda Barrios
Decano
Facultad de Ingeniería

Señor Decano.

Me dirijo a usted para informarle que después de conocer el dictamen del asesor del trabajo de tesis titulado **TECNOLOGÍAS PARA LA CREACIÓN DE SISTEMAS DE APLICACIÓN EN INTERNET**, elaborado por el estudiante **MANUEL FERNANDO LÓPEZ FERNÁNDEZ**, procedo a la autorización del mismo.

Sin otro particular, me suscribo de usted.

Atentamente,

Ing. 
Coordinador

Ingeniería en Ciencias y Sistemas



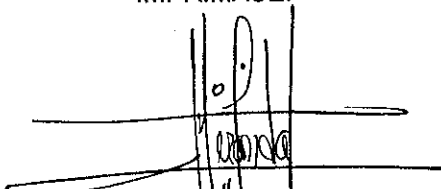
FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
y Regional de Post-grado de Ingeniería
Sanitaria.

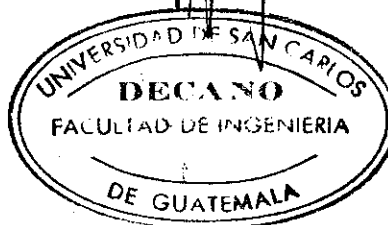
Ciudad Universitaria, zona 12
Guatemala, Centroamérica

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la autorización por parte del Coordinador de la Carrera de Ingeniería en Ciencias y Sistemas, al trabajo de tesis titulado "**TECNOLOGÍAS PARA LA CREACIÓN DE SISTEMAS DE APLICACIÓN EN INTERNET**", presentado por el estudiante universitario **MANUEL FERNANDO LÓPEZ FERNÁNDEZ**, procede a la autorización para la impresión de la misma.

IMPRÍMASE:



Ing. Herbert René Miranda Barrios
DECANO



Guatemala, octubre de 1999

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	XII
GLOSARIO	XII
INTRODUCCIÓN	XXIII
1. CONCEPTOS DE INTERNET	1
1.1 Descripción de Internet	1
1.2 Historia de Internet	3
1.3 Autoridad en Internet	4
1.4 Conexión a Internet	7
1.4.1 "Internet Service Provider"	8
1.4.2 Líneas telefónicas	9
1.4.2.1 Líneas "Dial-Up"	9
1.4.2.2 Líneas dedicadas	10
1.4.3 Servicios de comunicación de datos	10
1.5 Servicios en Internet	11
1.5.1 "World Wide Web" (WWW)	12
1.5.2 Correo electrónico (E-mail)	13
1.5.3 Sesión remota (TelNet)	14
1.5.4 Protocolo de transferencia de archivos (FTP)	15
1.5.5 Charlas a través de Internet (IRC)	16
1.5.6 Motores de búsqueda	17
1.6 Arquitectura cliente / servidor en Internet	20

1.6.1	El servidor: "Web Servers" _____	21
1.6.1.1	"Internet Information Server" _____	23
1.6.1.2	"Oracle Web Server" _____	26
1.6.1.3	"Apache Web Server" _____	28
1.6.2	El cliente: Navegadores _____	29
1.6.2.1	"Internet Explorer" (IE) _____	30
1.6.2.2	"Netscape Navigator" _____	31
1.6.2.3	Otros navegadores _____	34
1.7	Seguridad en Internet _____	34
1.7.1	"Firewall": pared de fuego _____	35
1.7.2	"Proxy Server" _____	36
1.7.3	"Secure Socket Layer" (SSL) _____	37
1.7.4	Firmas digitales _____	39
2.	LENGUAJES PARA LA CREACIÓN DE SISTEMAS EN INTERNET _____	43
2.1	Lenguajes para la "Web" _____	43
2.2	El Lenguaje HTML _____	44
2.2.1	Etiquetas del lenguaje HTML _____	45
2.2.2	Atributos de las etiquetas _____	46
2.2.3	Formato de las URL _____	47
2.2.4	Estructura de un documento HTML _____	50
2.2.5	Cabecera de un documento HTML _____	51
2.2.6	Cuerpo de un documento HTML _____	54
2.3	Lenguajes "Scripts" _____	54
2.3.1	JavaScript _____	56
2.3.1.1	Inclusión de JavaScript _____	56
2.3.1.2	Sintaxis básica _____	57
2.3.1.3	Variables _____	58

2.3.1.4	Instrucciones de control lógico	58
2.3.1.5	Palabras reservadas	59
2.3.1.6	Funciones	60
2.3.1.7	Objetos	60
2.3.1.8	Eventos	61
2.3.2	Visual Basic "Script"	62
2.3.2.1	El Funcionamiento de VBScript	63
2.3.2.2	Aspectos de sintaxis	65
2.3.2.3	Operaciones no permitidas en VBScript, a diferencia de VB	65
2.3.2.4	Recomendaciones para crear páginas con VBScript	67
2.4	HTML Dinámico (DHTML)	68
2.5	Páginas HTML generadas por código ejecutado en el servidor de WWW	69
2.5.1	"Common Gateway Interface" (CGI)	69
2.5.1.1	Fundamentos del CGI	71
2.5.1.2	Variables de entorno	72
2.5.1.3	Variables dependientes del protocolo http	76
2.5.1.4	Resultado de la ejecución de un CGI con salida PH	77
2.5.1.5	Resultado de la ejecución de un CGI con salida NPH	78
2.5.1.6	Procesamiento de formularios	79
2.5.2	Páginas HTML con código de programación inmerso	81
2.5.2.1	"Internet Database Connector" (IDC)	81
2.5.2.2	"Active Server Pages" (ASP)	84
2.5.2.3	PHP/FI de Rasmus Lerdorf	88
2.5.2.4	Servlets	90
2.6	Páginas con código ejecutado en el cliente	91
2.6.1	ActiveX de Microsoft	92
2.6.2	"Java-Applets"	93

3.	TECNOLOGÍAS AVANZADAS PARA INTERNET _____	97
3.1	Tecnologías para bases de datos en proyectos "Web" _____	97
3.1.1	Bases de datos y "Web Servers" _____	97
3.1.2	Gestores de bases de datos _____	98
3.1.3	Interfaz entre las aplicaciones y el gestor _____	98
3.1.3.1	"Open Database Connectivity" (ODBC) _____	99
3.1.3.2	OLE DB _____	99
3.1.3.3	"Java Database Connectivity" (JDBC) _____	100
3.1.3.4	JDBC vs. ODBC _____	101
3.1.4	Acceso a la información en las bases de datos _____	102
3.1.4.1	"Internet Database Connector" (IDC) _____	102
3.1.4.2	"Advanced Data Connector" (ADC) _____	103
3.1.4.3	"ActiveX Data Objects" (ADO) _____	103
3.2	Nuevas tecnologías de desarrollo _____	107
3.2.1	"ComponentWare" _____	109
3.2.1.1	COM: componentes y objetos _____	111
3.2.2	"JavaBean" _____	112
3.2.2.1	Ambientes de uso _____	112
3.2.2.2	Características fundamentales _____	113
3.2.3	"Common Object Request Broker Architecture" (CORBA) _____	114
3.2.3.1	Arquitectura de CORBA _____	114
3.2.3.2	"Interface Definition Language" (IDL) _____	116
3.2.4	Coordinadores / monitores de transacciones _____	117
3.2.4.1	Microsoft "Transaction Server" _____	118
3.2.4.2	Jaguar CTS _____	121
3.3	Herramientas para el desarrollo y diseño de páginas _____	124
3.3.1	Adobe PageMill _____	124
3.3.2	Dreamweaver _____	125

3.3.3	NetObjects Fusion_____	126
3.3.4	"Microsoft Front Page" _____	126
3.3.4	Visual Interdev_____	127
3.3.4.1	Creación de un proyecto que acceda a bases de datos_____	128
3.3.4.2	El entorno de desarrollo y las bases de datos_____	128
3.3.4.3	"Visual Database Tools" _____	129
3.3.4.4	Diseñador de bases de datos_____	129
4.	INTEGRACIÓN DE LAS TECNOLOGÍAS_____	131
4.1	Descripción de la integración_____	131
4.2	Descripción del escenario para el ejemplo_____	133
4.3	Consideraciones de las herramientas y tecnologías para la implementación_____	134
4.3.1	Elementos del cliente: Navegadores y el soporte de tecnologías _____	136
4.3.1.1	HTML _____	137
4.3.1.2	Lenguaje "Script" _____	138
4.3.1.3	Multimedia _____	139
4.3.1.4	"Applets" (JAVA) / Controles ActiveX _____	140
4.3.2	El servidor y herramientas para la implementación_____	140
4.3.2.1	Sistema Operativo _____	141
4.3.2.2	Servidor "Web" _____	142
4.3.2.3	Manejador de base de datos _____	142
4.3.2.4	Interfaz para el acceso a base de datos _____	143
4.3.2.5	Lenguaje "Script" para el lado del servidor _____	144
4.3.2.6	Servidores transaccionales _____	144
4.3.2.7	Lenguaje para la generación de componentes _____	145
4.3.2.8	Herramienta para el diseño y creación de páginas _____	145
4.4	Descripción del ejemplo a implementar_____	146
4.5	Creación del sistema_____	149

4.5.1	Crear las páginas del sistema sin funcionalidad	150
4.5.2	Crear la base de datos	156
4.5.3	Crear un ODBC	159
4.5.4	Hacer un "data connection" en Visual Interdev	160
4.5.5	Crear el componente	161
4.5.6	Registrar el componente en manejador de componentes	170
4.5.7	Incluir las operaciones del componente dentro de las páginas	173
4.6	Finalización del sistema de Internet	174
4.7	Acceso al sistema por los clientes	176
CONCLUSIONES		179
RECOMENDACIONES		181
BIBLIOGRAFÍA		185
APÉNDICE A		187
APÉNDICE B		209

ÍNDICE DE ILUSTRACIONES

FIGURAS

No.	Título	Pág.
1	Jerarquía para el acceso a bases de datos en Microsoft	104
2	Arquitectura de Corba	115
3	Diagrama de bloques del ejemplo	147
4	Dar nombre al nuevo proyecto	150
5	Selección del servidor de "Web"	151
6	Nombre de la nueva aplicación	151
7	El explorador del proyecto	152
8	Agregar una nueva página	153
9	Código inicial de la página	153
10	Página de ingreso al sistema	155
11	Exploración de una tabla	161
12	Ambiente de Visual Basic en la creación de componentes	162
13	Asistente de creación de paquetes	171
14	Asistente de componentes	172
15	MTS con el componente COMMantenimiento	173
16	Página principal del sistema de ejemplo	177

TABLAS

No.	Título	Pág.
I	Dominios superiores en Guatemala	6
II	Servicios de comunicación de datos	11
III	Algunas etiquetas HTML	45
IV	Servicios utilizados en los URLs	48
V	Estructura de una página HTML	50
VI	Etiquetas para el título de una página	52
VII	Forma de inhabilitar JavaScript en Navegadores que no lo soportan	55
VIII	Objetos de JavaScript	61
IX	Ejemplos de uso de los objetos de JavaScript	61
X	Eventos de JavaScript	62
XI	Código fuente de un archivo IDC	82
XII	Código fuente de un archivo HTX	83
XIII	Código fuente de una página con código PHP	88
XIV	Sintaxis para incluir un "Applet" de Java en una página HTML	94
XV	Atributos de la etiqueta "Applet"	94
XVI	Establecimiento de una conexión con ADO	105
XVII	Creación de un ADO "Recordset"	106
XVIII	Forma de agregar un registro a una tabla	107
XIX	Comparación entre ActiveX y "JavaBeans"	111
XX	Estructura del archivo Browsecap.ini	137
XXI	Estructura de la tabla de usuarios	157
XXII	Roles de usuarios del sistema	158
XXIII	"Script" SQL de la tabla de usuarios	158
XXIV	Archivo DSN del ODBC de la base de datos	160
XXV	Constantes del componente COMManejo	164

XXVI	Función que devuelve los datos del usuario	165
XXVII	Procedimiento público que agrega un usuario	166
XXVIII	Procedimiento que elimina un usuario	167
XXIX	Procedimiento que modifica los datos de un usuario	169
XXX	Llamada del componente con VBScript	174

GLOSARIO

- ACID** "Atomicity", "Consistency", "Isolation", "and Durability"; atomicidad, consistencia, aislamiento y durabilidad. Representan las iniciales del conjunto de propiedades que deben cumplir las transacciones.
- ADC** "Advanced data connector"; conector avanzado de datos. Es una nueva tecnología para el acceso sencillo desde aplicaciones Web, a bases de datos heterogéneas. Esta nueva tecnología permite construir aplicaciones Web que accedan a cualquier base de datos a través de un ODBC.
- ADO** "ActiveX Data Objects"; objetos de datos ActiveX. Es la interfaz de Microsoft de alto nivel para el acceso a los datos, que aísla a los desarrolladores de las tecnologías fundamentales OLE DB y ODBC.
- API** "Application program interface"; interface de programa de aplicación. Es el método específico prescrito por una computadora; sistema operativo o programa de aplicación, por medio del cual un desarrollador que escribe un programa de aplicación puede hacer requerimientos del sistema operativo o de otra aplicación.
- ASP** "Active server pages"; páginas de servidor activas. Es una tecnología propuesta por Microsoft para la generación de páginas dinámicamente.

- ATM** "Asynchronous transfer mode"; modo de transferencia asincrónica. Es una implementación avanzada de conmutación de paquetes y que es parte de los servicios de la red digital de banda ancha.
- B-ISDN** "Broadband integrated services digital network"; red digital de servicios integrados de banda ancha. Un nuevo servicio de área amplia, que pretende ofrecer vídeo sobre pedido, televisión en vivo, correo electrónico, multimedia, interconexión con LAN y otros servicios; todo por la línea telefónica.
- CCITT** "Comité consultatif international telegraphique et telephonique"; comité consultivo internacional de telégrafos y teléfonos. Conocido actualmente como ITU-T; es la organización principal internacional para la estandarización en los equipos y sistemas de las telecomunicaciones.
- CERN** "Conseil Europeen pour la Recherche Nucleaire"; centro Europeo de investigación nuclear. Centro de investigación donde surgiera en 1,991 el WWW, debido al investigador Dr. Tim Berners-Lee .
- CGI** "Common gateway interface"; interfaz de pasarela común. Es una interfaz entre programas de aplicación y servicios de información. Hasta el momento, se ha utilizado únicamente como pasarela hacia servidores WWW que utilizan el protocolo http
- CIR** Centro de Información de Redes. La autoridad reconocida en Guatemala para el control de Internet. Tiene como objetivo facilitar y proveer a la comunidad Internet del país, los servicios de registro para la operación de las redes Internet en los diferentes sectores, así como establecer los mecanismos para recoger y difundir la información relevante para la operación óptima de la Internet en Guatemala.

- COM** "Component object model"; modelo de objetos basados en componentes. Es una tecnología que se ha venido imponiendo en los últimos años como una de las tecnológicas de Microsoft. La idea es dotar a los desarrolladores de un ámbito mediante el cual se desarrollen componentes que puedan comunicarse con otros componentes o aplicaciones, proporcionándoles servicios. Es un estándar que define un modo de interacción de aplicaciones.
- CORBA** "Common object request broker architecture"; arquitectura intermediaria de solicitudes de objetos comunes. Es una especificación para una arquitectura estándar orientada a objetos para aplicaciones que puede ser utilizada en las redes por aplicaciones de distintos vendedores, y que fue creada por el OMG("Object Management Group").
- CSS** "Cascading style sheets"; hoja de estilo en cascada. Es una página "Web" estilo hoja, derivada de varias fuentes con un orden definido de precedencia. De esta manera, los desarrolladores del "Web" logran dar nuevos efectos y estilos a las páginas "Web".
- DARPA** "Defense advanced research project agency network"; red de la agencia de defensa de proyectos de investigación avanzada. Proyecto del departamento de defensa de los Estados Unidos que creó Internet; el propósito principal del proyecto era la investigación y desarrollo de protocolos de comunicación para redes de área amplia.
- DBMS** "Database manager system"; sistema administrador de bases de datos. Consiste en una colección de datos interrelacionados y una colección de programas para acceder a esos datos.

- DCOM** "Distributed component object model"; modelo de objetos de componentes distribuido. Utilizando COM, esta tecnología propuesta por Microsoft permite a los desarrolladores implementar soluciones con componentes que residen en servidores remotos y distribuidos.
- DHTML** "Dynamic HTML"; HTML dinámico. Es un conjunto de tecnologías que permiten un contenido "Web" más dinámico; utilizando un lenguaje "script" como JavaScript, es posible utilizar nuevas características de los navegadores, para que sin necesidad de cambiar de página o llamar otra, sea posible realizar cambios de formato en la página actual.
- DNS** "Domain name system"; sistema de nombre de dominio. Resuelve nombres de Internet a direcciones IP.
- DOM** "Document object model"; modelo de objetos de documento . Uno de los componentes del DHTML propuesto por el "World Wide Web Consortium".
- FTP** "File transfer protocol"; protocolo de transferencia de archivos. Un protocolo orientado a la conexión, que se encuentra en la capa de aplicación de los modelos OSI y TCP/IP.
- GNU** GNU's Not UNIX; GNU no es Unix. Es un sistema operativo tipo Unix, que da el código fuente públicamente para ser copiado, modificado y redistribuido. El proyecto fue creado en 1,983 por Richard Stallman.
- "HOST"** Dependiendo del contexto tiene varios significados. En Internet se refiere a cualquier computadora que tenga acceso a la red de redes. En el ambiente IBM, se refiere a las supercomputadoras o "mainframes", que son los servidores de red.

- HTML** "HyperText markup lenguaje"; lenguaje de marcas de hipertexto. Lenguaje basado en etiquetas y diseñado para informar al navegador la manera de presentar los textos, gráficas y archivos multimediales.
- HTTP** "Hypertext transfer protocol"; protocolo de transferencia de hipertexto. Es el encargado de la transferencia de páginas HTML, y es parte de la capa de aplicación en los modelos de referencia TCP/IP.
- IAB** "Internet architecture board". Se encarga de tomar las decisiones acerca de los estándares de comunicaciones entre las diferentes plataformas, para que puedan interactuar máquinas de diferentes fabricantes sin problemas; este grupo es responsable de cómo se deben asignar las direcciones y otros recursos en la red.
- IDC** "Internet database connector"; conector de bases de datos para Internet. Permite a cualquier página "Web" servida por IIS, acceder a bases de datos ODBC.
- IDL** "Interface definition language"; lenguaje de definición de interface. Es un lenguaje declarativo para definir las interfaces que unen a los componentes de la arquitectura CORBA. Estas interfaces son independientes de los lenguajes de programación y de los sistemas operativos e incluyen constantes, declaraciones de tipos, excepciones, atributos y operaciones.
- IEEE** "Institute of electrical and electronics engineers"; instituto de ingenieros eléctricos y electrónicos. Institución que se encarga de velar que los fabricantes de "hardware" cumplan con los estándares, para favorecer al consumidor.

- IETF** "Internet engineering task force"; fuerza de trabajo de Ingeniería del Internet . Grupo encargado de recibir las opiniones de los usuarios de Internet, sobre cómo se deben de implementar soluciones para problemas operacionales, y cómo deben de cooperar las redes para lograrlo.
- IIOB** "Internet Inter-ORB Protocol"; protocolo de Internet entre ORB. Es un protocolo orientado a objetos, que hace posible que programas distribuidos escritos en diferentes lenguajes de programación se comuniquen a través del Internet.
- IIS** "Internet information server"; servidor de información de Internet. "Software" servidor de Microsoft, que es utilizado para prestar los servicios de "Web" y Ftp.
- IRC** "Internet relay chat"; charla a través de Internet. Una forma de comunicación a través del Internet que se realiza en tiempo real y en el cual se trata un tema, normalmente llamado un canal.
- ISP** "Internet service provider"; proveedor de servicios de Internet, es una compañía que provee a otras compañías o usuarios individuales, el acceso o presencia en el Internet.
- JDBC** "Java database connectivity"; conectividad a bases de datos con Java. Es una API de JAVA para permitir ejecutar instrucciones SQL. Para que una aplicación pueda hacer operaciones en una base de datos, ha de tener el correspondiente "driver" que conecte la aplicación con ésta. Así pues la API JDBC es básicamente un paquete de JAVA (java.sql) que contiene un conjunto de clases e interfaces escritas en JAVA.

- JDK** "Java development kit"; herramientas de desarrollo en Java. Un conjunto de herramientas para la programación en Java, incluyendo el compilador.
- LAN** "Local area network"; red de área local. Redes de propiedad privada dentro de un solo edificio, entre 10 m y 1 km.
- MAN** "Metropolitan area network"; red de área metropolitana. Red que podría abarcar un grupo de oficinas cercanas o una ciudad y podría ser privada o pública, de hasta 10 km.
- MIME** "Multi-Purpose Internet mail extensions"; extensiones multipropósito de correo electrónico. Es una extensión del protocolo original de correo electrónico en Internet, que permite a los usuarios intercambiar distintos tipos de archivos de datos.
- MSMQ** "Message Queuing"; Colas de mensaje. Proporciona una manera sencilla para que las aplicaciones envíen y reciban mensajes con seguridad en una red en forma asíncrona.
- MTA** "Message transfer agent"; agente de transferencia de mensajes. Un programa que es responsable del envío de mensajes de correo electrónico localmente o pasarlo a otro MTA en otro "host".
- MUA** "Mail user agent"; agente usuario de correo. Programa especial para escribir mensajes de correo electrónico. También tiene la tarea de mostrar los mensajes que tiene en su correo y que están almacenados en el servidor de correo, posiblemente en alguna máquina remota.

- NCA** "Network computer Architecture"; arquitectura de red computarizada. Es un cuadro de trabajo que puede usarse para crear sistemas abiertos de aplicaciones distribuidas, y en el cual esta basado el "Web Application Server de Oracle".
- NIC** "Network information center"; centro de información de redes. Se encarga de hacer la asignación de las direcciones correspondientes a cada red. Esta organización es administrada por el Departamento de Defensa de los Estados Unidos.
- NPH** "No parse headers"; cabecera sin parsear. Es una forma de indicar a un CGI que no analice ni cambie ninguno de los parámetros enviados en la cabecera de una página.
- ODBC** "Open database connectivity"; conectividad de bases de datos abierta. Es un estándar o API público creado por Microsoft para el acceso a bases de datos de distintas fuentes de datos. Para poder acceder a los datos, es necesario tener un "driver" para esta fuente de datos.
- OLE DB** "Object linking and embedding data base"; incrustación y enlace de objetos de bases de datos. Es una estrategia de Microsoft; es una interface de programación a nivel de sistema para acceder datos a través de una organización. La característica principal es la capacidad de acceder tanto a fuentes de datos a través del ODBC, como a fuentes de datos que no son bases de datos relacionales.
- OLPT** "Online transaction processing"; procesamiento de transacciones en línea. Es un tipo de programa que facilita y maneja aplicaciones orientadas a transacciones, típicamente utilizado para la entrada y recuperación de transacciones.

- OMG** "Object management group"; grupo de manejo de objetos. Consorcio que agrupa a más de 650 compañías de "hardware", "software", bancos, de telefonía, etc., cuya misión es definir interfaces para "software" interfuncional, usando una tecnología orientada a objetos.
- ORB** "Object request broker"; intermediario de solicitudes de objetos. En CORBA, es la infraestructura de comunicación entre clientes y servidores por el que los objetos hacen peticiones y reciben respuestas. Específicamente, es el responsable de recibir peticiones de objetos, encontrar el objeto (local o remoto) que las implemente, prepararlo y comunicar los datos en la petición.
- OSI** "Open systems interconnection"; interconexión de sistemas abiertos. Un modelo propuesto por la ISO, que define las capas de un sistema de red, con los requerimientos y servicios de cada una de ellas.
- PGP** "Pretty good privacy"; bastante buena confidencialidad. Es un programa muy popular usado para encriptar y desencriptar correo electrónico en Internet. También puede ser usado para encriptar y enviar firmas digitales que permiten al destinatario identificar la identidad del enviador, y así saber que no fue alterado el correo en el camino.
- PHP/FI** "Personal home page/form interpreter"; página personal de inicio / interprete de formas. Es un lenguaje que posibilita el acceso a bases de datos como mSQL, Postgres, Oracle y otras de forma muy sencilla. PHP es muy similar a las páginas ASP de Microsoft.

PPP "Point to point protocol"; protocolo punto a punto. Protocolo que se encargará de realizar la negociación de conexión a un ISP en el caso de una conexión por módem. Es un protocolo de la capa de enlace de datos en el modelo OSI.

PSTN "Public switched telephone networks"; red telefónica pública conmutada. Se refiere a la colección interconectada mundial de redes telefónicas públicas orientadas a voz, tanto en posesión del gobierno como de la empresa privada.

RAD "Rapid application development"; desarrollo rápido de aplicaciones. Es un concepto en el cual los productos de software pueden ser desarrollados en forma rápida y de alta calidad basándose en ciertas técnicas como la reutilización de código.

RFC "Request for comment"; solicitud de comentarios. Es un documento formal en Internet o estándar, que es el resultado de un borrador de un comité, y es revisado posteriormente por las partes interesadas.

"ROUTER" Es un dispositivo, que puede ser "software", que determina el próximo punto de red, al cual un paquete debería seguir para poder llegar a su destino.

RSA "Rivest-Shamir-Adleman". Es un sistema de encriptamiento y autenticación en Internet. Fue desarrollado en 1977 por las tres personas con apellidos del nombre del algoritmo. Es el algoritmo más utilizado y está incluido en los navegadores de Netscape y Microsoft.

SDH "Synchronous digital hierarchy"; jerarquía digital sincrónica. Otro nombre que se le da por las recomendaciones de la CCITT, al sistema SONET.

- SMTP** "Simple mail transfer protocol"; protocolo simple de transferencia de correo. Protocolo del TCP/IP empleado en Internet para dar el servicio de envío de correo electrónico.
- SONET** "Synchronous optical network"; red óptica sincrónica. Este servicio de comunicación de datos, consiste en conmutadores, multiplexores y repetidores, conectados todos por fibra.
- SQL** "Structured query language"; lenguaje estructurado de consultas. Es un lenguaje estándar de alto nivel para crear, manipular, examinar y gestionar bases de datos relacionales.
- SSL** "Secure socket layer"; nivel de socket de seguridad. Es el protocolo para implementar un servidor seguro, el cual garantiza la privacidad de datos que se transmiten por la red a través de encriptamiento.
- TCP/IP** "Transmission control protocol/ Internet protocol"; protocolo de control de transmisión / protocolo de Internet. Un protocolo de transmisión que asigna a cada máquina que se conecta un número específico, llamado "número IP" (que actúa a modo de "número teléfono único"). El protocolo TCP/IP sirve para establecer una comunicación entre dos puntos remotos mediante el envío de información en paquetes.
- TLD** "Top level domains"; dominios de más alto nivel. Es uno por país (dominios de dos letras correspondientes al código ISO-3166 de cada territorio) más los dominios especiales de 3 letras tales como "edu", "com", "gov", "mil", "org", "int" y "net".

- URL** "Uniform resource locator"; localizador uniforme de recursos. Es una especie de dirección en Internet de una de las páginas. El URL especifica el protocolo de transporte a utilizar (HTTP, FTP, etc.), el servidor de "Web" en que se hospeda, el directorio y el nombre del archivo.
- WAN** "Wide area network"; red de área amplia. Red de computadoras que se extiende sobre un área geográfica extensa, a veces un país o un continente, entre 100km. Y 1000 km.
- WEB** Ver Www.
- WWW** "World wide web". Es una red basada en el protocolo HTTP y el lenguaje HTML, diseñados con el propósito de crear una red donde se pueda en una forma sencilla, presentar información tanto tipo texto, como incrustación de multimedia (imágenes, sonido, video).
- WWWC Ó W3C** "World wide web consortium". Fundada en Octubre de 1,994 para dirigir el "World Wide Web" a su máximo potencial, a través del desarrollo de protocolos comunes que promuevan su evolución y que aseguren la interoperabilidad.

INTRODUCCIÓN

Internet es la interconexión de muchas redes de computadoras, a través del protocolo TCP/IP, el cual provee varios servicios como el correo electrónico, la transferencia de archivos y por supuesto el Web, los cuales permiten con las nuevas tecnologías de desarrollo, implementar sistemas de aplicación que dan al usuario la funcionalidad que dan los sistemas de aplicación convencional.

La popularidad de Internet, en los últimos años, se debe en gran parte a que es el medio más novedoso de comunicación a grandes distancias y con costo razonable. En un principio su función era enviar información a través de mensajes y posteriormente presentar información de la empresa en una página de Web. Desde el crecimiento del Web, para la mayoría de las personas, la percepción de la palabra Internet ha sido un sinónimo de la visualización de páginas estáticas HTML en un navegador de Web.

La necesidad de presentar información que responda a ciertos requerimientos y que sea en tiempo real, llevó a los investigadores a crear tecnologías que permitan cumplir con estos requerimientos, y que hoy en día son utilizados para implementar sistemas completos sin mucho esfuerzo.

Los primeros servidores Web no tenían ningún medio de acceder a bases de datos y de devolver los resultados de una consulta en forma de HTML. Este problema se resolvió poco después con la aparición de CGI ("Common Gateway Interface"), una especificación algo tosca,

que define un medio para que los servidores Web intercambien información con software externo. Sin embargo, tiene grandes problemas de rendimiento.

La investigación ha continuado, y han surgido varias tecnologías que tratan de resolver principalmente el acceso a las bases de datos y la presentación de la información en tiempo real, en forma más dinámica. Algunas de las primeras tecnologías no han proporcionado la funcionalidad ni la facilidad al programador para implementar sistemas completos de aplicación a través del navegador, y son muy poco utilizadas.

Tecnologías más versátiles basadas en el concepto de “**ComponentWare**”, que proporcionan una solución para el acceso a más manejadores de bases de datos, han surgido en los últimos años. Sin embargo, para los creadores de sitios “Web” el panorama con el tiempo se oscurece más, al tener que tomar la decisión entre varias tecnologías y varios elementos que son necesarios considerar al implementar un sistema a través del Internet.

La presente investigación describe los elementos que se pueden aplicar en el Internet, y presenta a los desarrolladores de sistemas de aplicación para Internet, las tecnologías que han surgido desde la aparición del Web, que permiten la creación de sistemas más dinámicos y con más funcionalidad. Por último, se presenta un ejemplo utilizando páginas Asp y componentes con las funciones principales en los sistemas de aplicación, utilizando las tecnologías más recientes, más robustas y más estables, que permitan al lector conocer cuál es el proceso para la creación de sistemas de aplicación completos a través del Internet.

1. CONCEPTOS DE INTERNET

1.1 Descripción de Internet

Internet es un conjunto de redes de computadoras y equipos físicamente unidos mediante cables que conectan puntos de todo el mundo. Estos cables se presentan en muchas formas: desde cables de red local (varias máquinas conectadas en una oficina) a cables telefónicos convencionales, digitales y canales de fibra óptica que forman las carreteras principales. Esta gigantesca red se difumina en ocasiones, porque los datos pueden transmitirse vía satélite, o a través de servicios como la telefonía celular, o porque a veces no se sabe muy bien dónde está conectada.

En Internet, las comunicaciones concretas se establecen entre dos puntos: uno es la computadora personal desde el que se accede a la información, llamado **cliente**; el otro es cualquiera de los **servidores** que hay en la red y se encargan de proveer la información.

El fundamento de Internet es el TCP/IP, un protocolo de transmisión que asigna a cada máquina que se conecta un número específico, llamado número IP (que actúa a modo de número telefónico único); por ejemplo, 192.100.26.11. TCP/IP sirve para establecer una comunicación entre dos puntos remotos mediante el envío de información en paquetes. Al transmitir un mensaje o una página con imágenes por ejemplo, el bloque completo de datos se divide en pequeños bloques que viajan de un punto a otro de la red, entre dos números IP determinados, siguiendo cualquiera de las posibles rutas.

La información viaja por muchas computadoras intermedias a modo de repetidores hasta alcanzar su destino, lugar en el que todos los paquetes se reúnen, reordenan y convierten en la información original. Millones de comunicaciones se establecen entre puntos distintos cada día, pasando por cientos de computadoras intermedias.

La gran ventaja del TCP/IP es que es inteligente. Como cada intercambio de datos está marcado con números IP determinados, las comunicaciones no tienen por qué cruzarse. Y si los paquetes no encuentran una ruta directa, las computadoras intermedias prueban vías alternas. Se realizan comprobaciones en cada bloque para que la información llegue intacta y en caso de que se pierda alguno, el protocolo lo solicita de nuevo hasta que se obtiene la información completa.

TCP/IP es la base de todas las máquinas y software sobre el que funciona Internet: los programas de correo electrónico, transferencia de archivos y transmisión de páginas con texto e imágenes y enlaces de hipertexto. Cuando es necesario, un servicio automático llamado Servicio de Nombres de Dominio (DNS), convierte automáticamente esos crípticos números IP a palabras (como www.universidad.edu) para que sean fáciles de recordar.

La finalidad del DNS es permitir la coordinación, tanto desde el punto de vista administrativo como desde el punto de vista técnico, del sistema de nombres de Internet mediante una distribución jerárquica de dominios delegados. Los dominios son entidades administrativas, cuyo propósito es subdividir la carga de gestión de un administrador central y repartirla entre los sub-administradores. A su vez, éstos pueden repetir el proceso, si así lo aconseja el volumen del dominio que se va a administrar.

En el primer nivel de la jerarquía, se encuentran los "Top Level Domains" o TLDs que son uno por país (dominios de dos letras correspondientes al código ISO-3166 de cada territorio) más los dominios especiales de 3 letras: "edu", "com", "gov", "mil", "org", "int" y "net".

En general, la estructura jerárquica del DNS trata de reflejar dependencias administrativas bajo un primer componente geográfico (o por actividad en el caso de los TLDs "especiales"); por ejemplo, corresponderá a una máquina o servicio, que pertenece a un departamento o sucursal de una organización en Francia, un nombre del tipo: máquina.departamento.organización.fr.

Cada TLD dispone de sus propias normas para decir quién puede y quién no puede registrar un dominio de segundo nivel, qué dominios están permitidos, qué procedimiento es preciso seguir, etcétera. El hecho de que alguien cumpla con los requisitos para registrar un dominio bajo un TLD, no implica que cumpla con los requisitos para registrar éste u otro dominio bajo otro TLD.

1.2 Historia de Internet

Internet fue creada a partir de un proyecto del departamento de defensa de los Estados Unidos llamado DARPA (Defense Advanced Research Project Network, red de la agencia de defensa de proyectos de investigación avanzada) iniciado en 1969, cuyo propósito principal era la investigación y desarrollo de protocolos de comunicación para redes de área amplia, para ligar redes de transmisión de paquetes de diferentes tipos, capaces de resistir las condiciones de operación más difíciles y continuar funcionando aún con la pérdida de una parte de la red.

Estas investigaciones dieron como resultado el protocolo TCP/IP ("Transmission Control Protocol/Internet Protocol"), un sistema de comunicaciones muy sólido y robusto, bajo el cual se integran todas las redes que conforman lo que se conoce actualmente como Internet. Durante el desarrollo de este protocolo, se incrementó notablemente el número de redes locales de agencias gubernamentales y de universidades que participaban en el proyecto, dando origen así a la red de redes más grande del mundo.

Las funciones militares se separaron y se permitió el acceso a la red a todo aquel que lo requiriera, sin importar de qué país provenía la solicitud, siempre y cuando fuera para fines académicos o de investigación (y por supuesto que pagara sus propios gastos de conexión); los usuarios pronto encontraron que la información que había en la red era por demás útil, y si cada cual aportaba algo se enriquecería aún más la cantidad de información existente.

1.3 Autoridad en Internet

Por extraño que parezca, no existe una autoridad central que controle el funcionamiento de la red, aunque existen grupos y organizaciones que se dedican a organizar de alguna forma el tráfico en ella. Después de que las funciones militares de la red se separaron en una subred de Internet, la tarea de coordinar el desarrollo de la red recayó en varios grupos, uno de ellos, fue la "National Science Foundation", que promovió bastante el uso de la red, y se encargó de conectar cinco centros de supercómputo que podían ser accesados desde cualquier nodo de la red. Eso funcionó bien al principio, pero pronto fueron superadas las cargas de tráfico previstas, entonces se dio la concesión a "Merit Network Inc.", para que administrara y actualizara la red; se mejoraron las líneas de comunicación y se dio un servicio mucho más rápido, sin embargo, este proceso de mejora nunca termina, debido a la creciente demanda de los servicios que se encuentran en la red.

El grupo de mayor autoridad sobre el desarrollo de la red es la "Internet Society", creado en 1990 y formado por miembros voluntarios, cuyo propósito principal es promover el intercambio de información global a través de la tecnología Internet; puede decirse que esta sociedad es como un consejo de ancianos que tiene la responsabilidad de la administración técnica y dirección de Internet.

Éste no es el único grupo que puede tomar decisiones importantes, pues existen otros tres grupos que tienen un rol significativo; la "Internet Architecture Board" toma las decisiones acerca de los estándares de comunicaciones entre las diferentes plataformas, para que puedan interactuar máquinas de diferentes fabricantes sin problemas; este grupo es responsable de cómo se deben asignar las direcciones y otros recursos en la red, aunque no son ellos quienes se encargan de hacer estas asignaciones, para eso existe otra organización llamada NIC ("Network Information Center", centro de información de redes) administrado por el Departamento de Defensa de los Estados Unidos.

El otro grupo importante es el IETF ("Internet Engineering Task Force", fuerza de trabajo de Ingeniería del Internet), en el cuál los usuarios de Internet expresan sus opiniones sobre cómo se deben de implementar soluciones para problemas operacionales y cómo deben de cooperar las redes para lograrlo.

El enorme crecimiento de Internet se debe en parte a que es una red basada en fondos gubernamentales de cada país que forma parte de Internet, lo que proporciona un servicio prácticamente gratuito. A principios de 1994, comenzó a darse un crecimiento explosivo de las compañías con propósitos comerciales en Internet, dando así origen a una nueva etapa en el desarrollo de la red.

En Guatemala, la autoridad reconocida para el control de Internet es CiR (Centro de Información de Redes, cuya dirección en Internet es www.gt). El objetivo de esta institución según lo describen es: *"CIR- INTERNet Guatemala tiene como objetivo facilitar y proveer a la comunidad Internet del país, los servicios de registro para la operación de las redes Internet en los diferentes sectores (Educativo, Comercial, Gobierno, ONG ...), así como establecer los mecanismos para recoger y difundir la información relevante para la operación óptima de la Internet en Guatemala"*.

Internet funciona a través de nombres de dominios, en una forma jerárquica que facilita la localización y organización de los servidores y las redes. En Guatemala, CIR es el encargado del dominio, en la clase por países de Guatemala; dicho dominio es: .gt y bajo este dominio CIR es el responsable de crear y administrar esos nombres. La información para el dominio de Guatemala es el siguiente:

Nombre del Dominio: (.gt), **Dominio Superior de Guatemala:** ns 168.234.32.2 ns.gt.

Tabla I. Dominios superiores en Guatemala

CANTIDAD DE DOMINIOS	DESCRIPCIÓN DEL DOMINIO	DOMINIO
1	Dominio superior	gt
24	Instituciones educativas	edu.gt
61	Organizaciones no-gubernamentales	org.gt
27	Dependencias de gobierno	gob.gt
395	Entidades comerciales	com.gt
32	Entidades que proveen infraestructura a la red nacional	net.gt
1	Entidades militares	mil.gt

Para poder ver los dominios registrados bajo ".gt", es posible visitar la página de CIR Guatemala, www.gt, y si se desea información sobre otros países, en el apéndice de enlaces se encuentra una dirección Ftp en la cual es posible encontrar los códigos por país.

1.4 Conexión a Internet

Para poder conectarse a Internet como usuario final, son necesarios ciertos dispositivos de hardware y software.

Los dispositivos físicos o de hardware, necesarios para conectarse a Internet, son:

- Una línea de teléfono y un módem interno o externo,
- En el caso de estar en una red LAN con acceso a Internet, una conexión a la red y una tarjeta de red (NIC; "Network Interface card", Tarjeta Interfaz de red).

El software necesario para conectarse a Internet es:

- **Software de "Dial-Up"**, para realizar una conexión telefónica, en el caso que se requiera conectar a través de un módem.
- **El protocolo PPP**, el cual se encarga de realizar la negociación de conexión a un ISP en el caso de una conexión por módem.
- **El protocolo TCP/IP**, para conectarse a través de un ISP ya sea por medio de un módem, un enlace directo o como cliente de una red local a través de un "Proxy server".

1.4.1 "Internet Service Provider"

ISP; **Proveedor de servicios de Internet:** es una compañía que provee a otras compañías o usuarios individuales con acceso, o presencia en el Internet. La mayoría de los ISP dan servicios de ayuda en el diseño, creación y administración de sitios "Web", entrenamiento y administración de intranets.

Para proveer el acceso, estas compañías están provistas de equipo especial; esto incluye un servidor de "Web", un servidor de correo electrónico, un servidor Ftp y otros servicios que desee prestar la empresa. Estos servicios pueden estar en un equipo que tenga suficiente capacidad en disco duro para almacenar el software y el espacio para los servicios de páginas "Web", Ftp. Además, deberá poseer la cantidad necesaria de memoria RAM y capacidad de procesamiento, para poder atender las solicitudes de clientes concurrentes.

Para que los clientes puedan conectarse, es necesario que el proveedor del servicio de Internet cuente con suficientes líneas telefónicas; esto exige tener además un módem por línea telefónica. Estos modems generalmente están conectados a un dispositivo especial llamado **pool de modems**, el cual proporciona los buses para poder conectar los modems a la computadora, a través de un puerto.

Otros dispositivos, para un proveedor de servicio de Internet (ISP), son las conexiones para los clientes que desean conectarse con un enlace dedicado. Esto significa que el ISP provee de una conexión directa desde su servidor a la computadora del cliente. Esto implica que el cliente tendrá un ancho de banda preestablecido para la comunicación y además por lo general, la conexión es sin límite de tiempo, a diferencia de los clientes que usan las líneas telefónicas para

conectarse. Para poder realizar estas conexiones de enlace dedicado es necesario que tanto el ISP como el cliente utilicen dispositivos para redes WAN. Este dispositivo, llamado "Router", es el encargado de realizar la conexión directa entre el cliente y el ISP.

Por último, el dispositivo fundamental para un ISP es su conexión de salida a Internet. Este generalmente involucra el uso de tecnologías de comunicación avanzada, como podría ser microondas, o comunicación satelital, fibra óptica, o alguna otra tecnología que le dé la velocidad y ancho de banda necesario al proveedor del servicio de Internet. Esta conexión es un enlace dedicado del ISP a un "backbone", por medio de alguna empresa que brinde este servicio. Estas compañías son empresas grandes que implementan una infraestructura de comunicación de alta velocidad, que es la base de Internet.

1.4.2 Líneas telefónicas

Son utilizadas para poder establecer la comunicación a través de largas distancias. Existen dos tipos de conexión y la selección de un tipo depende de la distancia, el rendimiento y el costo.

1.4.2.1 Líneas "Dial-Up"

Para que un usuario final de Internet pueda conectarse a un ISP, en el caso de un usuario temporal, es necesario establecer una conexión con el proveedor de servicio de Internet. En este caso, se utilizan las líneas de marcado de teléfono o líneas Dial-up, que son conexiones establecidas temporalmente entre máquinas a través de líneas de teléfono, y utilizan modems para la comunicación. Debido a que los datos digitales que genera una computadora deben viajar

a través de la red telefónica pública de conmutación, en la cual viajan como señales analógicas, es necesario una transformación de señal (modulación / demodulación), de lo cual se encargan los modems. En los últimos años, los servicios han sido mejorados. Algunas líneas digitales pueden transmitir hasta 56 Kbps usando corrección de errores, compresión de datos y modems sincronizados.

1.4.2.2 Líneas dedicadas

En el caso de los clientes que tienen un acceso permanente, se utilizan líneas dedicadas para la conexión. Estas líneas son "voice grade", las cuales proveen el ancho de banda y señalización ordinarios en los PSTN ("public switched telephone networks") y son diseñados de conformidad con la señalización y codificación digital, con lo cual se consiguen mayores velocidades.

Estos proveen conexiones de tiempo completo dedicadas y no usan una serie de conmutadores para completar la conexión. La calidad de estas líneas es generalmente más alta que la calidad de una línea telefónica, la cual fue diseñada sólo para transmisión de voz. El rango típico de velocidad oscila entre 56 Kbps a 45 Mbps o más.

1.4.3 Servicios de comunicación de datos

Las redes públicas telefónicas conmutadas fueron diseñadas y creadas para dar servicios de transmisión de voz. En la actualidad, para la transmisión de datos, esta infraestructura no da la capacidad necesaria para los requerimientos de las redes de comunicación. Por eso, han

surgido nuevos servicios que están diseñados para la transmisión de datos a altas velocidades, y son parte del operador de red que presta el servicio de comunicación a los "host" y terminales del cliente. Este sistema es llamado **red pública** y es análogo al sistema telefónico público, y algunas veces puede ser parte de él.

Cada uno de estos servicios tiene características importantes como la velocidad de transmisión, que normalmente se expresa en Mega bits por segundo (Mbps). La forma de conexión es importante, ya que indica el orden en que llegan los paquetes de información al destino, y si son confiables o no. La **conmutación** de paquetes indica si los paquetes pueden tomar distintas rutas en la subred para llegar al destino. Y la **multidifusión** indica si cada mensaje enviado llega a todas las máquinas en la red o llega directamente al destino. Algunos de los servicios más comunes se muestran en la siguiente tabla.

Tabla II. Servicios de comunicación de datos

SERVICIO	Velocidad (Mbps)	Orientado a la conexión	Conmutación de paquetes	Multidifusión
X.25	.064	Si	Si	No
Frame Relay	1.5	Si	No	No
ATM	155	Si	Si	Si

1.5 Servicios en Internet

Los servicios básicos proveídos por Internet se basan en los protocolos de la capa de aplicación del mismo. Los servicios más antiguos, como el envío de mensajes y la transferencia de archivos, siguen siendo muy populares, aunque han ido mejorando con el tiempo.

Uno de los servicios más recientes y que ha dado la mayor popularización al Internet es el protocolo http, por el cual es posible ver las páginas a través de los navegadores de "Web". A continuación, se describen los servicios más comunes y que normalmente deben de integrarse a los sistemas de aplicación para Internet.

1.5.1 "World Wide Web" (WWW)

Es una red basada en el lenguaje HTML ("Hypertext Markup language"; lenguaje de marcas de hipertexto) y el protocolo http ("Hypertext Transport Protocol"; protocolo de transporte de hipertexto), que es parte de la capa de aplicación en los modelos de referencia TCP/IP y OSI, diseñados con el propósito de crear una red donde se pueda, en una forma sencilla, presentar información tanto del tipo texto, como de la incrustación de multimedia (imágenes, sonido, video). Comenzó en 1,989 en el CERN (Centro Europeo de Investigación nuclear) y su creador fue Tim Berners Lee. La primera interfaz gráfica fue Mosaic, y fue presentado en febrero de 1,993 por su creador Marc Andreessen.

La "Web" sigue un modelo cliente / servidor, en la que los documentos residen en unos computadores que ofrecen la información en forma de páginas a través del software servidor de "Web". Para poder solicitar alguna página, se necesita un programa cliente o navegador, tales como Mosaic, Netscape o "Internet Explorer". Para visualizar una página es necesario tener un URL ("Uniform Resource Locator"; Localizador Uniforme de Recursos), que es una especie de dirección en Internet de una de las páginas. El URL especifica el protocolo de transporte que se va a utilizar (HTTP, FTP, etcétera), el servidor de "Web" en que se hospeda, el directorio y el nombre del archivo. Si no se sabe el URL, se puede utilizar un URL de alguno de los índices de búsqueda para el Web, como Altavista, Yahoo, OLE y seguir los enlaces que allí se encuentran.

1.5.2 Correo Electrónico (E-mail)

Es uno de los servicios más utilizados y antiguos en Internet. Se basa en el envío de mensajes que siguen un formato dado por el RFC 822 ("Request For Comment"; documentos publicados para recibir comentarios o proposiciones por la W3) o MIME ("Extension Multipurpose Internet Mail"; extensiones multipropósito de correo electrónico), que permite enviar texto plano y algunos otros tipos de datos. Los mensajes son escritos por el enviador con la ayuda de un programa especial MUA ("Mail User Agent"; agente usuario de correo).

Generalmente este programa también tiene la tarea de mostrar los mensajes que tiene en su correo y que están almacenados en el servidor de correo, posiblemente en alguna máquina remota. Los mensajes son pasados a algún MTA ("Message Transfer Agent"; Agente de transferencia de mensajes); un programa que es responsable del envío de mensajes localmente o pasarlo a otro MTA en otro "host". Estos MTA se comunican usando el protocolo SMTP ("Simple Mail Transfer Protocol"; protocolo sencillo de transporte de correo). El mensaje es entregado en el recipiente del destinatario (normalmente un archivo en esa computadora), en el cual el destinatario pueda leer su correo.

Los mensajes están compuestos por una envoltura primitiva (descrita en el RFC 821), algunos campos de cabecera, una línea en blanco y el cuerpo del mensaje. Cada campo de cabecera consiste en una sola línea de texto ASCII, que contiene el nombre del campo, dos puntos y para la mayoría de los campos, un valor.

El encabezado contiene entre otras, la dirección de correo electrónico destino, así como la de otras posibles direcciones donde se quiera enviar una copia del mensaje. Además, puede llevar información sobre el tema del mensaje, normalmente llamado el "subject" del mensaje. Luego, el cuerpo del mensaje constituye la información que se va a transmitir, la cual puede contener un archivo, o más incrustados al documento.

Los agentes de usuario esperan direcciones DNS de la forma buzón@Servidor.Correo, donde la palabra buzón es la cuenta en el servidor, Servidor.Correo es el nombre de dominio del servidor de correo, aunque pueden tener direcciones en un formato X.400, que se componen de partes atributo = valor.

1.5.3 Sesión remota (TelNet)

Es un protocolo estándar de Internet, diseñado para el acceso remoto y basado en el modelo cliente/servidor. La idea es que un programa cliente emule una terminal del servidor o "host" y pueda actuar la terminal como si estuviera enfrente del "host".

Telnet corre sobre el protocolo TCP/IP, y muchos "host", especialmente los que utilizan el sistema operativo UNIX, ya implementan este protocolo. Otros "host" o servidores necesitan de un software especial para poder ofrecer este servicio, como es el caso de servidores con el sistema operativo Windows NT. Estos programas son productos de terceros como los de Georgia SoftWorks.

Para poder conectarse a un "host", es necesario saber la dirección. Se carga el programa cliente de telnet seguido de dicha dirección. Debe tomarse en cuenta que como se está emulando una terminal (teclado y pantalla principalmente), existen diferentes tipos de emulación, de los cuales el más común es el de la empresa DEC llamado VT-100, que era la terminal tonta para acceder a los "mainframe" de IBM. Luego de haberse conectado, a continuación se solicita un "login" y "password" al usuario. Si está registrado en el "host" (o este permite realizar sesiones como visitante "guest"), se mostrará una pantalla de información sobre el "host", y a continuación, será posible utilizar los servicios que presta el "host" como si se estuviera ante él.

1.5.4 Protocolo de transferencia de archivos (FTP)

El "file transfer protocol" es uno de los protocolos más antiguos en Internet. Es un protocolo que en forma sencilla y basado en el modelo cliente / servidor, da la posibilidad de bajar o copiar archivos en cualquier formato desde una máquina remota (servidor de Ftp), a nuestra computadora (cliente Ftp) o viceversa.

El Web ha dado una nueva presentación a los servidores FTP, ya que ahora es posible mediante un documento en HTML presentar todos los archivos disponibles en el servidor FTP, incluso los enlaces a los archivos pueden darse en forma gráfica. Han surgido discusiones sobre la eficiencia entre un servidor de FTP y uno que utilice el HTTP para la transferencia de archivos, pero el HTTP da muchas más funciones y ventajas en la parte de presentación al cliente. El cliente puede ser un software o aplicación especial para acceder el servidor ftp, tal como cuteFTP, que es un programa "shareware"; también es posible utilizarlo, a través de comandos en el navegador, incluso a través de comandos en el D.O.S.

1.5.5 Charlas a través de Internet (IRC)

Internet Relay Chat. Las charlas en tiempo real son una de las novedades, que han dado auge al crecimiento de Internet. Su funcionamiento general es el siguiente: una persona que quiera charlar debe saber una dirección URL, la cual puede incluir una página con una aplicación cliente en java, para poder conectarse al servidor de "chat" o puede utilizar el telnet; la más común y más antigua es tener una aplicación que pueda conectarse a un servidor IRC. Los IRC son servidores que hablan cierto lenguaje, para poder prestar los servicios de "chat" a los usuarios.

Sea cual sea la forma utilizada para conectarse, al iniciar una sección lo primero que se solicita es un tema, mas conocido como canal, al cual desea conectarse el cliente; éste podría ser por ejemplo un canal sobre la **paz**, o sobre **redes**, por algún país en específico, o algún tema de actualidad.

Al haber seleccionado el canal aparece una nueva ventana, en la cual se cuenta con una ventana principal, donde aparecen todos los mensajes de las personas en línea y que van dirigidos a la general. Puede existir otra ventana donde aparecen los alias de las personas que están en dicho canal. Debe existir algún espacio en la aplicación, donde es posible ingresar los mensajes que se quiere, lleguen a los demás usuarios del canal.

En algunas aplicaciones, es posible enviar mensajes a personas en específico, a lo que se le llama **hablar en privado**, ya que los mensajes sólo pueden ser visto por el que lo envía y la persona a la que se envía el mensaje.

En algunos de estos servidores de "chat", se implementan robots que pueden ser programas, encargados de monitorear el lenguaje utilizado por los usuarios. Si se utilizan palabras no adecuadas, se le da un aviso informando que las palabras utilizadas no son apropiadas, y si el usuario insiste en usarlas puede ser expulsado del canal. Algunas veces, en lugar de robots, pueden existir administradores o dueños del canal, que son los que se dedican a orientar la conversación en el canal.

Otro tipo de aplicaciones orientadas a la comunicación en tiempo real y que han evolucionado y crecido grandemente, son aquellas en las cuales es posible registrarse en un servidor y registrar a personas en nuestra aplicación cliente para poder saber en tiempo real, si están conectados o no a Internet.

El programa da un aviso o muestra en la aplicación qué personas que uno haya registrado se encuentran en línea. La aplicación cliente da la posibilidad de enviar mensajes, archivos, iniciar sesiones de "chat", utilizar un juego para red, etcétera. Dos de estos ejemplos son Miralbis ICQ y AOL de America Online Inc.

1.5.6 Motores de búsqueda

Debido al crecimiento exponencial de Internet, la búsqueda de la información se ha vuelto cada vez más difícil, debido a la gran cantidad de información que día con día se publica. Desde sus inicios, se ha pensado en dos formas para solucionar este problema.

La primera es la creación de índices jerárquicos, en los cuales se establece un nivel superior con los temas más generales de la información, como podría ser el caso de Educación, Medicina, Negocios, etcétera. Todo esto presentado a través de una página de "Web". El usuario que busca la información puede pulsar con el "mouse" sobre alguno de estos temas; a continuación, se presentan más temas específicos sobre el tema seleccionado. En esta forma, se puede continuar hasta llegar a temas específicos, que por medio del link, se muestra la información solicitada y que puede residir en un servidor en cualquier parte del mundo.

La segunda forma para la búsqueda de información, y la más común, es la utilización de motores de búsqueda, que son bases de datos que contiene la información de páginas y las palabras claves de cada una. De esta forma, el motor de búsqueda provee una página donde es posible ingresar palabras claves de la información buscada. El motor de búsqueda se encarga de realizar una búsqueda en la base de datos donde reside la información de las páginas, registradas por la gente que realiza las páginas.

Cada motor de búsqueda provee de ciertos mecanismos para facilitar la búsqueda. Por ejemplo, en www.infoseek.com es posible realizar la búsqueda de un tema general. Luego de obtener un resultado es posible realizar una búsqueda sobre los resultados obtenidos, y de esta forma poder llegar a obtener los "links" (enlaces a otras páginas), a las páginas de la información buscada. Otros motores de búsqueda como www.altavista.com proveen de operadores del álgebra booleana como el "and" y el "or" para realizar las búsquedas.

Otro tipo de motor de búsqueda, importante de conocer, es el buscador de buscadores, el cual provee la facilidad de hacer una búsqueda de buscadores, que puedan especializarse en ciertos temas, como pueden ser buscadores en inglés, en español o por algún tema específico. A continuación, se muestra una lista de motores e índices de Internet.

- **<http://www.buscopio.com/>:** es una evolución de GOBIB, guía de índices y motores de búsqueda; una página creada en la primera quincena de diciembre de 1.996 y mantenida desde entonces por Ricardo Fornas Carrasco. Gobib, hoy BUSCOPIO, gozó desde el principio del favor de los internautas, tanto los veteranos como los de reciente incorporación.
- **[http://www.yahoo.com.](http://www.yahoo.com/)** apareció en 1994 de la mano de dos estudiantes de Ingeniería Electrónica de la Universidad de Stanford, con la intención de recopilar y clasificar los recursos de la "Web". Yahoo es considerado el más popular de este tipo de "Web sites"; al realizar una búsqueda, es posible utilizar los operadores lógicos AND y OR. Los resultados se presentan en orden alfabético. Este popular motor de búsqueda incluye un completo catálogo dividido por temas.
- **<http://www.hotbot.com/>:** este impresionante motor de búsqueda trabaja a partir de la tecnología suministrada por Inktomi, y cuenta con una base de datos de 54 millones de documentos. Ofrece una impresionante cantidad de opciones al definir la búsqueda; así es posible utilizar los operadores lógicos AND, OR y NOT.
- **<http://www.altavista.com/>:** éste es uno de los buscadores más rápidos y utilizados en estos momentos, y se constituye como el heredero idóneo de Yahoo!. Al realizar una búsqueda, es posible utilizar los operadores lógicos AND, OR, NOT y NEAR. Sus resultados al realizar la búsqueda son de buena calidad.
- **<http://www.lycos.com/>:** la página principal de este buscador, aunque incluye bastante publicidad, ofrece uno de los mejores diseños. Aparte del sistema de búsqueda propiamente dicho, ofrece una amplia variedad de servicios, acceso al servidor de noticias "topnews",

realizar el "download" gratuito de software, entre otras cosas. Al realizar la búsqueda, se pueden utilizar los operadores lógicos AND y OR; presenta los resultados de más alta calidad y tiene buena velocidad.

- <http://www.excite.com/>: este sencillo y potente motor de búsqueda puede resultar una opción ideal para realizar la búsqueda de información sobre temas específicos. En su página principal, aparece un directorio dividido en categorías que se expandirán al pulsar sobre ellas hasta llegar al tema que interese. Al realizar la búsqueda, se pueden utilizar los operadores lógicos AND, OR y NOT.

Existe una gran cantidad de motores de búsqueda y algunos son especializados. En los apéndices, se puede consultar una lista más completa.

1.6 Arquitectura Cliente / Servidor en Internet

Internet es la interconexión de redes de computadoras más grande en el mundo. La interconexión está hecha a partir de una infraestructura de redes Wan, lo que implica la utilización de una subred de comunicación.

Cada una de las redes en Internet está compuesta generalmente por uno o más servidores de varios servicios. Estos servidores son computadoras generalmente con grandes capacidades de procesamiento y almacenamiento, tanto primario (RAM) como secundario (HDD, discos duros). A estas computadoras físicas y que tiene algún sistema operativo y servicios especiales, se les llama **servidores de red**.

Por ejemplo, una computadora "alpha server" con 12 Terabytes en disco duro y 128 de memoria RAM, con un sistema operativo Windows NT, es un caso común de servidores de red. Estos servidores normalmente no trabajan solos, ya que los servicios que deben prestar son varios.

Por ejemplo puede existir un servidor de red, en conjunto con un servidor de archivos, uno de impresión y otro de "Web". Estos otros servidores, generalmente también son computadoras con cierto hardware que está corriendo, cierto software para prestar los servicios, o pueden residir en la misma computadora que el servidor de red. En el caso de un servidor de "Web", en Windows NT, puede estar corriendo el programa IIS ("Internet Information Server", Servidor de información de Internet) para prestar los servicios de "Web".

Los clientes, en el modelo cliente / servidor en Internet, normalmente se refieren a todos los usuarios de los servicios que presta Internet, es decir los usuarios de los servidores de "Web", de FTP, etcétera. Estos clientes pueden ser usuarios de una red LAN, la cual tiene acceso a Internet o pueden ser usuarios que utilizan líneas "Dial-Up", con modems para conectarse a un ISP, para entrar a la Internet. Estos usuarios necesitan como mínimo el protocolo TCP/IP y una configuración de "Dial-Up" para conectarse.

1.6.1 El servidor: "Web Servers"

Un servidor de "Web" es un servicio prestado por una computadora a través de un software especial, el cual hace uso del protocolo HTTP para responder a los requerimientos de los navegadores remotos, los cuales son utilizados por los clientes del "Web". Este software especial, llamado "Web Server", hace uso del concepto de "Web Site", que es una forma de organizar la

información bajo un nombre o directorio en el "Web Server". Además, el "Web server" puede tener otras sesiones de "Web", los cuales deben prestar los servicios en otro puerto. EL protocolo HTTP utiliza por "default" el puerto 80. Si un "Web server" hace uso de varios nombres para el mismo "host", entonces estos otros nombres dados al mismo servidor se le llaman **servidores virtuales**.

Entre el software de servidores de "Web" más populares se encuentran:

- "Apache Web Server" y "Web" StrongHold, el cual esta basado en el apache para Unix y Linux
- NCSA y Netscape Enterprise Server de Netscape Corp
- IIS de Microsoft para Windows
- Sun "Web Server"
- Oracle "Application Web Server"

Todos los servidores tienen diferencias principalmente en el lado del servidor, pero los servicios prestados al cliente deben ser los mismos. Como mínimo, aparte de prestar los servicios de "Web" y FTP, estos servidores deberían poder soportar CGI o "Server Scripts", que son tecnologías descritas a detalle en la sección 2.5.1 y 2.5.2.2.

En el mercado, existe gran cantidad de productos de "Web Server"; para ver una lista más detallada de los mismos, consúltese el apéndice B, "Web Servers".

1.6.1.1 "Internet Information Server"

El servidor "Web", incorporado en Microsoft Windows NT Server, "Internet Information Server" (IIS), facilita compartir documentos e información en una Intranet o en Internet. IIS es el servidor "Web" más rápido para Windows NT Server y está integrado completamente a sus Servicios de Directorio. Esta combinación, servicios "Web" y sistema operativo, hace posible desplegar aplicaciones escalables y confiables basadas en el "Web".

Windows NT Server se integra en forma natural al "Web". En Windows NT Server, el "Web" se convierte simplemente en otra parte del sistema operativo. Las "Active Server Pages" (ASP) del IIS ofrecen un ambiente de aplicación avanzado, abierto y libre de compilación, en el cual se puede combinar HTML, "scripts" y componentes de servidor ActiveX reusables, para crear soluciones de negocios dinámicas y poderosas basadas en el "Web". En el sistema operativo, están completamente integradas las herramientas como Microsoft FrontPage y la búsqueda de texto dinámica y total.

"Crystal Reports", incluida en IIS, es una herramienta para reportes visuales que permite crear presentaciones de calidad e integrarlas en aplicaciones de bases de datos. Esta integración "Web" permite que se construyan sitios "Web" con calidad profesional, que administre una red, e incremente la flexibilidad del sistema y su escalabilidad, además de que mantenga una seguridad total y centralizada.

La integración de IIS, "Microsoft Transaction Server" (que es el servidor de transacciones) y "Microsoft Message Queue" (que es un servicio para el envío de mensajes en forma asíncrona) en "Windows NT Option Pack" (es un grupo de aplicaciones, que se incluyen en el paquete para

empresas de Microsoft, llamado "Back Office"), hace que Windows NT Server sea una plataforma muy completa para desarrollar y desplegar fácilmente aplicaciones confiables, escalables y de usuario múltiple, basadas en el "Web". En específico, esto se logra a través de:

- **"Crash Protection"**: permite que los usuarios ejecuten confiadamente varias aplicaciones en el servidor "Web". Si una de las aplicaciones se detiene, el servidor "Web" y las otras aplicaciones continuarán en ejecución, mientras la aplicación con la falla se reinicia con una solicitud posterior.
- **"Transactional Active Server Pages" - ASP**: IIS introduce ASP, lo que permite a las aplicaciones con "scripts" y componentes efectuar acciones múltiples, con todas o ninguna acción relacionada. Este aspecto es crítico en las aplicaciones orientadas a bases de datos.
- **Depuración de "script"**: IIS proporciona una solución integrada para depurar "scripts" y aplicaciones de ASP, por lo que facilita el desarrollo de aplicaciones de ASP.
- **Colas de mensaje integradas**: MSMQ proporciona una manera sencilla para que las aplicaciones envíen y reciban mensajes con seguridad en una red (incluso en el caso en el cual parte de la aplicación o de la red no estén disponibles).
- **Soporte para Java**: IIS incluye un conjunto de clases de Java para construir componentes del lado del servidor y la Java 1.1 "Virtual Machine" de Microsoft. Al usar páginas ASP y MTS ("Microsoft Transaction Server"; Servidor de Transacción de Microsoft), es posible construir

fácilmente aplicaciones de servidor empleando Java y ejecutarlas con eficacia en un entorno de servidor complicado.

IIS fue diseñado para soportar varios escenarios de servidor "Web", que abarcan desde los sitios "Web" simples en una intranet corporativa, hasta grandes granjas "Web" de ISP ("Internet Service Provider"; proveedores de servicios de Internet).

- **Control total del contenido:** ahora los administradores pueden configurar el servidor "Web" dependiendo del servidor, sitio, directorio virtual, subdirectorio o bien, tomando como base el archivo, con lo que se ofrece una flexibilidad completa en la mayoría de las configuraciones de servidor.
- **Sitios "Web" múltiples:** al contar con soporte para varios sitios "Web" en una sola dirección de IP, operadores de sitio "Web", reducción de ancho de banda por sitio "Web" y administración mejorada y personalizable, IIS permite que las organizaciones hospeden fácilmente varios sitios departamentales en una intranet, o que proporcionen servicios de hospedaje a varios sitios públicos en Internet.
- **Estándares de Internet:** IIS incorpora los estándares industriales anteriores y más modernos, incluidos: HTTP 1.1 para desempeño mejorado en Internet, emisión y manejo de certificados digitales X.509 (que se pueden mapear hacia cuentas de usuarios de Windows NT Server), además del envío de mensajes hacia servidores estándares de correo SMTP y de noticias NNTP.

- **Administración:** IIS ofrece las herramientas de administración más flexibles para manejar el servidor "Web". Entre ellas figuran una consola de administración integrada, basada en Windows; una administración enriquecida basada en el explorador; elaboración de "scripts" en la línea de comando; así como interfaces programables para construir herramientas de administración personalizables. Esta versión también ofrece una nueva funcionalidad para restaurar la configuración anterior, por lo que permite que se devuelva al servidor las configuraciones guardadas previamente.
- **Seguridad:** IIS incluye un servidor de certificación integrado que se acopla firmemente al modelo de seguridad de Windows NT Server, y permite que las organizaciones emitan y administren certificados digitales estándares X.509 para Internet. Esta versión también ofrece mejoras de seguridad para los bancos internacionales. Al emplear la tecnología "Server Gated Crypto", los bancos internacionales pueden ofrecer el encriptamiento más fuerte (de 128 bits) en sus transacciones en línea.

1.6.1.2 "Oracle Web Server"

El "Oracle Web Application Server" es el producto de Oracle para implementar aplicaciones distribuidas dentro del Internet, así como también dentro de intranets organizacionales. La arquitectura de la red computarizada (NCA) es un cuadro de trabajo que puede usarse para crear sistemas abiertos de aplicaciones distribuidas y en el cual está basado el "Web Application Server" de Oracle; éste juega un papel primordial dentro del NCA, y actúa como el punto de enfoque para el manejo de "cartridges" (cápsula de código, que forma parte del concepto "ComponentWare" que se describirá en el capítulo 3) de servicios y la comunicación entre "cartridges".

Los "cartridges" son unidades autocontenidas de funcionalidad, que pueden actuar como componentes cliente, componentes de lógica de aplicación o componentes de datos. El NCA también incluye un protocolo de comunicación estandarizado, para que los diferentes "cartridges" en una aplicación NCA puedan comunicarse entre ellos

Un "cartridge" es un módulo de código que interactúa con el "Oracle Web Application Server", a través de una interface estándar. La función básica del "Oracle Web Application Server", en el NCA, es manejar la interacción de los "cartridges". Para lograrlo, el "Oracle Web Application Server" administra la creación de los "cartridges" de lógica de la aplicación, la comunicación entre el "cartridge" de la aplicación, y los de lógica de aplicación y la comunicación entre "cartridges" de lógica de aplicación. El "Oracle Web Application Server" también provee los servicios básicos para los "cartridges" que son necesarios para implementar una aplicación robusta usando el NCA.

El "Oracle Web Application Server" está compuesto de los siguientes componentes básicos:

- **"Web Listener"**: el cual maneja la comunicación entre los clientes y el servidor de la aplicación de "Web", a través de los protocolos de Internet estándares.
- **"Web Request Broker"**: éste maneja la creación de procesos de "cartridges"; balancea la carga entre las múltiples instancias de un "cartridge" individual y los servicios a los "cartridges" como los servicios de transacciones, servicios de comunicación entre "cartridge", servicios de almacenaje persistentes y servicios de autenticación.

1.6.1.3 "Apache Web Server"

En el área de UNIX, el servidor más popular actualmente reconocido es el "Apache Web server". Este "software" tiene varias ventajas y una de las principales razones por la que es el más popular entre estos servidores, es que este software es gratis.

El software es distribuido en forma de código, fuente bajo la licencia de GNU; de esta forma, es posible compilarlo en plataformas UNIX, incluyendo Linux y FreeBSD (es un sistema operativo UNIX para i386 de Berkeley). El software ha sido distribuido gratuitamente desde sus inicios, por el proyecto Apache, localizado en <http://www.apache.org>. También existen binarios precompilados para las variantes de UNIX, incluyendo AIX, Digital UNIX, HP-UX y solaris.

La funcionalidad al "Apache Web Server" puede ser agregada a través de módulos. Apache es aún para programadores y si se desea hacer cambios al núcleo del servidor Apache; se necesita agregar y remover módulos. Básicamente, un módulo es un código en lenguaje C, que está compilado como un ejecutable dentro del servidor. El Apache contiene archivos precompilados binarios, los cuales tienen los módulos básicos del Apache (la mayoría relacionados con la autenticación del usuario). Ya compilados los archivos, el servidor Apache es configurado como un proceso demonio y corre indefinidamente.

El apache tiene tres archivos para la configuración de los servicios: el `conf/httpd.conf`, `conf/srm.conf` y `conf/access.conf`. Es necesario editar estos archivos para hacer que corra el apache. La configuración es a través de texto y se provee una interfaz por medio del navegador para su configuración. Otra de las grandes ventajas de este servidor es la facilidad para poder actuar como un servidor "proxy", o como un servidor de autenticación.

Apache ha comenzado a incursionar en el área de Microsoft Windows; actualmente hay versiones para Windows NT. Como en un principio no fue diseñado para un sistema operativo como éste, aun sus competidores como IIS de Microsoft y "Site Server" de Netscape Communication's, le llevan muchas ventajas en este ambiente.

El apache está diseñado principalmente para prestar servicios de "Web", imágenes gráficas y presta servicios de aplicación a través del "common gateway interface" (CGI), creados en C/C++, perl o TCL; Java es soportado en sus últimas versiones. En el área del comercio electrónico, el Apache no es de los mejores servidores, ya que soporta sólo algunas de las formas de SSL 3.0, que es el protocolo para las transacciones seguras. El Apache también soporta el acceso a bases de datos por medio de ODBC, pero la competencia tiene un mejor soporte en esta área de las bases de datos.

1.6.2 El cliente: navegadores

Un navegador, explorador o buscador ("browser"), es un programa cliente capaz de leer las etiquetas que contienen los documentos WWW y convertir esta información en formato gráfico. Además de permitir ver hipertexto, conecta el cliente a los servidores "Web", para pedirles los documentos a los que apuntan los "links".

Todos los navegadores presentan algunas características comunes: los botones para desplazarse hacia adelante o hacia atrás por las páginas ya visualizadas, el cuadro de texto donde se indica la dirección donde se está y en la que se puede escribir una nueva dirección a la que se desea viajar, la caché, la lista de lugares favoritos o visitados, etcétera.

1.6.2.1 "Internet Explorer" (IE)

"Internet Explorer" es un software cliente de la "Web". Esta aplicación es desarrollada por una de las empresas más grandes de software de computación, Microsoft.

Muchas de las más espectaculares características multimedia y de automatización del IE ("Internet Explorer") se encuentran disponibles a través de los "plug-ins" para "Navigator", pero la ventaja es que Microsoft las ha integrado en IE. Las nuevas posibilidades incluyen soporte para evaluación de los contenidos de páginas "Web", autenticación de archivos que garantiza la fidelidad de los archivos cargados, y una amplia gama de instrucciones abreviadas vía teclado.

Con su compilador JIT y soporte para "applets" y "scripts" en Java, da mucha funcionalidad a las páginas, ya no simplemente es código HTML inmerso en los documentos. IE añade mayor automatización e interactividad a través de controles y programación ActiveX (formalmente .OCX). Ambos navegadores hacen uso de un compilador JIT, de similares prestaciones, que acelera la ejecución de "applets" en Java.

El centro de atención es la llamativa interfaz de IE, que incluye una barra de herramientas que puede reducirse a sólo-imágenes (pero no sólo-texto como en el "Navigator"), una barra de direcciones y una barra de botones de enlace que puede personalizarse para incluir los sitios favoritos. El usuario de IE puede acelerar la navegación desactivando la carga de gráficos. Por otra parte, IE permite acceder primero al texto seguido de los gráficos, seleccionando la opción disponible en el teclado. "Navigator" sólo puede cargar el texto y los gráficos en el orden en que aparece el fichero original.

Otra de las mejoras de la interfaz de IE consiste en poder personalizar el tamaño de fuente, además de una interfaz de teclado para etiquetar enlaces y campos de formulario; "Navigator" sólo puede desplazarse a través de campos de formulario.

Ya se encuentran disponibles versiones de IE para Windows 3.x, Windows 9x, Windows NT y Macintosh, pero Microsoft sigue por detrás de la completa cobertura de sistemas operativos de la que dispone Netscape. A diferencia de "Navigator", que usa una lista de "bookmarks" en un único fichero; IE almacena las direcciones favoritas individualmente, lo que le integra mejor en la dinámica de Windows 9x.

La tecnología ActiveX de Microsoft proporciona a IE una ventaja potencial sobre "Navigator"; gracias a ella, IE se puede integrar con las aplicaciones del PC y se aprovecha la amplia gama de controles. OCX; el problema radica en que ActiveX se ajusta tanto a Windows, que muchos desarrolladores ignorarán esta tecnología en favor de Java o cualquier otra plataforma independiente de herramientas.

1.6.2.2 "Netscape Navigator"

La primera interfaz gráfica en navegadores fue Mosaic, en febrero de 1,993. Mosaic tuvo tanto éxito que, un año después, su autor Marc Andreessen, dejó el Centro Nacional de Aplicaciones de Supercómputo, donde se desarrolló Mosaic, para formar una compañía, Netscape Communications Corp., cuya meta fue desarrollar clientes, servidores y otros tipos de software de la "Web". Netscape se volvió pública en 1,995.

Actualmente Netscape Communicator incluye un navegador, una interfaz de desarrollo, de "chat" y algunas otras herramientas novedosas como "smartDownload" para bajar archivos de servidores "Web" o ftp, además el desarrollador incluye el uso de un compilador de Borland JIT Java, soporte para la construcción de una gran variedad de archivos multimedia y extensiones significativas a HTML que permiten a los diseñadores un mayor control sobre composición y apariencia. Adicionalmente, Netscape ha extendido JavaScript y su tecnología "LiveConnect", lo cual posibilita a los "applets" de Java, "plug-ins" y a los "scripts" de JavaScript, comunicarse con un control de uno sobre los otros. Netscape también ha mejorado las facilidades auxiliares de navegación, proporcionando conversaciones de voz en tiempo real y la posibilidad de compartir datos a través de su "CoolTalk applet".

Navigator actualmente (Netscape Communicator 4.6) está disponible para Windows 95/98/NT, MacPPC 7.6.1, Unix. La relativa apertura de Netscape a los sistemas operativos, al menos en comparación con Microsoft, ha ayudado a "Navigator" a mantener el liderato en lo relativo al soporte de otras plataformas. Netscape fue pionero del almacenamiento de páginas "Web" con una riqueza asombrosa de contenidos, gracias a su arquitectura "Plug-in", la temprana adopción del Java y el desarrollo del "JavaScript". El compilador JIT Java consigue drásticas mejoras equiparables a las de su homónimo de IE.

La extensión del "JavaScript" de "Navigator" incluye soporte para su evaluación. Esta mejora hace más fácil la creación de páginas HTML personales o intercambiables y deja que el autor "Web" introduzca expresiones de "JavaScript" en textos HTML, (una notación más intuitiva y compacta que la anteriormente disponible). Además, "JavaScript" puede ahora controlar "applets" de Java y "plug-ins", y permitir que los "plug-ins" puedan recompilarse por medio del "Live-Connect" SDK.

"Navigator" cubre la gama del multimedia basado en el "Web": Vídeo en vivo para ficheros AVI, audio en vivo para ficheros de sonido y 3D en vivo para ficheros VRML ("virtual reality modeling language", lenguaje de modelación de realidad virtual), pero desatiende el soporte para el formato MPEG-1. Como estas aplicaciones forman parte del navegador estándar, no hay que entretenerse en localizar, cargar e instalar dichos módulos.

"Navigator" no proporciona soporte original para controles ActiveX ni VBScript de Microsoft. En lo concerniente a las extensiones de HTML, Netscape ha sabido aprovechar todo el volumen que Microsoft ha definido; además, ha introducido algunas de su propia cosecha. "Navigator" soporta "frames" sin limitación, colores de fondo para las celdas de las tablas y permite al autor de HTML especificar una secuencia de tipos preferidos, en vez de simplemente aceptar las preferencias por defecto del usuario.

Las etiquetas HTML de "Navigator" incluyen Multicol, que automáticamente distribuye el texto en múltiple columnas. Una etiqueta espaciadora arbitra un mecanismo para forzar una estipulada cantidad de espacio entre objetos, incluyendo palabras y líneas. Sólo resta que los autores HTML aprendan a utilizar estos recursos, si bien todas estas nuevas etiquetas carecen de la flexibilidad exigida por la especificación CSS ("Cascading Style Sheets"; hojas estilo cascada) de nivel uno de Microsoft.

Microsoft argumenta que sólo usa elementos admitidos por las organizaciones de estandarización del HTML, lo cual se contradice con el hecho de que algunas de las características exclusivas HTML de IE, sean más una propuesta tentadora que estándares aprobados. La clave estriba en que ambas compañías entiendan que a los "Webmasters" y a los usuarios, no les gusta aquellos elementos HTML accesibles únicamente con el navegador apropiado.

1.6.2.3 Otros navegadores

Los dos navegadores, antes descritos (IE de Microsoft y "Netscape Navigator"), son los que las empresas más grandes en Internet promocionan. Aparte de estos dos navegadores, existen algunos, muchas veces llamados navegadores rebeldes, los cuales dan al usuario la posibilidad de interactuar en otras formas con las páginas "Web".

Muchos de estos navegadores no soportan todas las tecnologías existentes, tales como los lenguajes "Script". Algunos otros navegadores menos comunes son el Opera, Amaya, NeoPlanet, etcétera, y otros desarrollados en Java.

1.7 Seguridad en Internet

Uno de los objetivos, al diseñar la Internet, fue que el intercambio de información, datos y archivos fuera lo más libre posible; esta libertad ha implicado un precio: piratas informáticos y diseñadores de virus intentan atacar Internet y los computadores que están conectados a ella; aquellos "hackers" que tratan de invadir la información ajena, ya sean correos electrónicos, bases de datos o computadoras, o "crackers" que se dedican en un nivel de mayor profesionalismo, no sólo a invadir información, sino a robar, modificar y eliminar información que muchas veces implican desastres a las empresas

1.7.1 "Firewall": Pared de fuego

Muchas empresas tienen conectadas su red local a Internet, por lo que son vulnerables a ataques de los "hackers" o "crackers". La solución no es separar la red de Internet, sino que la compañía pueda establecer un "firewall", a fin de proteger la red interna de los ataques. Los "firewalls" permiten a los usuarios internos de la red de la compañía tener acceso al Internet, pero impiden que los piratas informáticos y otros usuarios de Internet obtengan acceso a la misma, y causen daños.

Los "firewalls" son combinaciones de "hardware" y "software" que se construyen utilizando "routers", servidores y diversos programas. Se encuentran en el punto más vulnerable entre la red corporativa e Internet, y pueden ser tan simples o complejos como se desee. Existen varios tipos de "firewalls" y los más simples emplean la técnica del filtrado de paquetes. Mediante esta técnica, un "router" de filtrado examina la cabecera de cada uno de los paquetes que se desplaza entre Internet y la red corporativa.

Las cabeceras de paquetes contienen información como la dirección IP del remitente y del destinatario, el protocolo empleado para enviar el paquete, además de otra información para la comunicación. En función de esta información, el "router" sabe qué tipo de servicio de Internet (como FTP o rlogin) está utilizándose para enviar los datos. Según esto, el "router" puede impedir que se envíen determinados paquetes entre Internet y la red de la compañía. Por ejemplo, el "router" puede bloquear cualquier tráfico, excepto el correo electrónico, o bien bloquear la entrada o salida de datos desde o hacia destinos sospechosos.

Otros componentes habituales de los "firewalls" son los "host" bastión. Se trata de servidores que gestionan todas las solicitudes procedentes de Internet y se dirigen a la red de la compañía. Los "host" bastión están muy bien protegidos contra los ataques. Al tener un único "host" encargado de gestionar la entrada de solicitudes, es más sencillo mantener la seguridad y localizar los ataques y, en el caso de que alguien forzase la entrada, solamente se pondría en compromiso ese "host" y no toda la red.

1.7.2 "Proxy Server"

Compartiendo una puerta de paso única y segura, un "Proxy Server" elimina la necesidad de compartir una máquina dedicada para Internet entre múltiples usuarios, o de operar diferentes líneas de comunicaciones en una empresa para proporcionar acceso a Internet a cada puesto de trabajo con conexiones separadas. Esto permite una máxima accesibilidad, a la vez que ayuda a reducir los costos y simplificar la administración.

Cuando un usuario de la red de la compañía desea acceder un servidor de Internet, se envía una solicitud desde el computador del usuario al servidor "proxy", que se conecta al servidor de Internet y tras lo cual, transmite la información desde éste al computador de la red de la compañía. Al actuar como intermediarios, los servidores "proxy" pueden mantener la seguridad, así como registrar todo el tráfico que se produce entre Internet y la red.

El servidor "proxy" es una máquina intermedia entre el cliente (cualquier navegador como mosaic, netscape, "Internet Explorer", etcétera) y los servidores de la WWW, de manera que las peticiones a páginas "Web" se realizan al "proxy", que se encarga de obtener la página deseada, y se conecta al servidor remoto. Para acelerar los accesos a la "Web", el servidor "proxy" dispone

de una memoria "cache" con los últimos documentos accedidos; de este modo, cuando se quiera obtener un documento, comprueba antes si éste ya se encuentra en la "cache"; en cuyo caso no es necesaria la conexión con el servidor remoto, incrementando de esta forma la velocidad de transferencia.

Tenga unos pocos usuarios o algunos miles, "Microsoft Proxy Server" puede ayudar a asegurar que tienen un acceso rápido a los datos que necesita. Mediante su "cache" inteligente, almacena los datos en Internet que son accedidos con frecuencia en su red de área local, a fin de que los usuarios los puedan utilizar, con la misma eficacia que cualquier otro dato de la red. Esto le ayuda a mejorar la productividad de sus usuarios, y reduce los costos a base de reducir el tráfico de la red y la congestión hacia o desde Internet.

1.7.3 "Secure Socket Layer" (SSL)

Cuando se rellena un formulario en una página de Internet y pulsa el botón **enviar**, se está enviando esos datos a través de la red. Estos datos, son transmitidos de servidor en servidor, hasta llegar a su destinatario. La información podría ser recogida (robada) en cualquiera de los servidores por los cuales pasa hasta llegar a su destino. Un servidor seguro garantiza la privacidad de datos que se transmiten por la red. Dicha privacidad se consigue mediante el protocolo SSL.

Un servidor seguro es un servidor de páginas html, especialmente configurado para establecer una conexión transparente con el cliente, consiguiendo que la información que circule entre ellos (cliente-servidor) viaje a través de Internet, encriptada mediante algoritmos que aseguran que la información transmitida sea comprendida sólo para el servidor y el visualizador

que accede al "Web". SSL es la plataforma necesaria para proteger la información confidencial (por ejemplo números de tarjetas de crédito). Es requisito imprescindible para el establecimiento de servicios de banca electrónica o de comercio electrónico.

Un servidor seguro funciona encriptando los datos que se envían mediante el sistema cifrado RSA, cuando se esté ubicado en una zona segura con el navegador. Su navegador, colaborando con el servidor seguro al que llama, encripta los datos de forma que, si algún individuo en el proceso de transmisión consigue apropiarse de éstos, no podrá leerlos, ya que no dispone de la clave necesaria. Esta encriptación se basa en el "Secure Socket Layer", SSL, estándar desarrollado por "Netscape Communications" para transferir información segura, a través de Internet.

Un servidor seguro certificado por VeriSing cuenta con una clave de 128 bits, con una parte secreta de 40 bits. Esto quiere decir que si algún intruso intenta descifrar los datos transmitidos con este sistema, deberá realizar 240 complicadas operaciones para descifrar estos datos, que en tiempo de computación supone miles de años en una de las máquinas más potentes del mercado.

Para saber si se está en un servidor seguro, se puede visualizar en la parte de dirección URL `https://` en vez de `http://`. Además, la mayoría de los navegadores lo indican: en Netscape se presentan una llave en la parte inferior izquierda, que habitualmente aparece partida y cuando se está en un servidor seguro se ve completa. En Microsoft "Internet Explorer", aparece un candado cerrado; si está configurado puede que el navegador avise que entra en un servidor seguro.

1.7.4 Firmas digitales

Es uno de los mecanismos de seguridad, utilizados para evitar el robo y modificación de la información que viaja a través de Internet.

Una firma digital es un bloque de caracteres que acompaña a un documento (o fichero), acreditando quién es su autor ("autenticación") y que no ha existido ninguna manipulación posterior de los datos ("integridad"). Para firmar un documento digital, su autor utiliza su propia clave secreta, a la que sólo él tiene acceso, lo que impide que pueda después negar su autoría ("no revocación"). De esta forma, el autor queda vinculado al documento que firma. Cualquier persona puede verificar la validez de una firma, si dispone de la clave pública del autor.

Para realizar una firma digital, el software del firmante aplica (de forma transparente al usuario) un algoritmo "hash" sobre el texto que se va a firmar, y se obtiene un extracto de longitud fija y absolutamente específico para ese mensaje (un mínimo cambio en el mensaje produce un extracto completamente diferente). Los algoritmos "hash" más utilizados son MD5 ó SHA-1. Este extracto, cuya longitud oscila entre 128 y 160 bits (en función del algoritmo utilizado), se somete a continuación a cifrado mediante la clave secreta del autor, previa petición de contraseña.

El algoritmo, utilizado para cifrar el extracto, puede ser el mismo RSA o una clave específica para firmar tipo DSS. El extracto cifrado constituye la firma y se añade al final del mensaje (o en un fichero adherido a él).

Para comprobar la validez de una firma digital, se necesita disponer de la clave pública del firmante para poder verificar su firma. El software del receptor descifra el extracto cifrado que constituye la firma digital (de forma transparente al usuario), utilizando para ello la clave pública del remitente. Como resultado, se obtiene un bloque de caracteres. A continuación, calcula el extracto "hash" que corresponde al texto del mensaje. Si el resultado coincide exactamente con el bloque de caracteres obtenido en la operación anterior, la firma se considera válida. Si existe la menor diferencia, la firma se considera no válida.

El sistema tiene un punto débil; la firma digital permite comprobar la relación entre un mensaje y la clave utilizada pero... ¿Cómo se puede estar seguro de que esa clave corresponde realmente a la persona o entidad que dice poseerla?. Este problema requiere la intervención de una tercera parte fiable, en la que confían las dos partes implicadas. Es lo que se llama **Autoridad de Certificación**.

Algunos programas, como PGP, no utilizan autoridades de certificación externas, sino que delegan en el propio usuario la responsabilidad de certificar claves conforme a su criterio, estableciendo lo que se denomina una red de confianza ("Web of Trust") totalmente descentralizada, pero con el apoyo de una red de servidores de claves. "The Global Trust Register" es un directorio que contiene las principales claves públicas del mundo, que permite verificar la validez de certificados X.509 y claves públicas PGP.

Una Autoridad de Certificación es esa tercera parte fiable que acredita, actuando como una especie de notario que extiende un certificado de claves (firmado con su propia clave), la relación entre una determinada clave y su propietario real. Verisign y Thawte son las autoridades de certificación más reconocidas a nivel mundial y puede obtenerse más información en sus sitios de Web (www.verisign.com , www.thawte.com respectivamente).

Los servidores certificados son aplicaciones destinadas a crear, firmar y administrar certificados de claves, y que permiten a una empresa u organización constituirse en autoridad de certificación, para subvenir sus propias necesidades. Uno de los productos más utilizados es "Netscape Certificate Server".

2. LENGUAJES PARA LA CREACIÓN DE SISTEMAS EN INTERNET

2.1 Lenguajes para la "Web"

El servicio más conocido por la comunidad de Internet es el "World Wide Web", el cual permite a los usuarios, a través de un programa cliente, llamado en general **navegador**, acceder a servidores que son la fuente de información, por medio de un protocolo llamado HTTP del TCP/IP. Este protocolo permite hacer solicitudes de documentos llamados páginas, las cuales son la unión de un lenguaje llamado HTML y la información en sí. El lenguaje HTML le indica al navegador en qué forma debe presentar la información y más recientemente el DHTML ayuda a darle más dinamismo a las páginas.

Estas páginas, además del HTML y la información propiamente dicha, pueden llevar inmerso lenguajes "Script", tales como "JavaScript", "PerlScript", "VBScript", los cuales dan funcionalidad extra a los navegadores. Normalmente este código es incluido en una página para hacer ciertas validaciones en los formularios, o realizar ciertas funciones más complejas que no pueden ser realizadas por el HTML.

Además de los lenguajes HTML y los lenguajes "script", existen otros lenguajes encargados de generar la página en el momento en que es solicitada, como lo son las Páginas de Servidor Activas ("Active Server Pages", ASP), que es una tecnología propuesta por Microsoft; PHP es otro lenguaje similar a ASP para la generación de páginas dinámicamente.

Un lenguaje muy importante en Internet y que le da al "Web" muchas funcionalidades es Java, normalmente a través de programas pequeños llamados "Applets". A continuación, se describen los principales lenguajes y tecnologías disponibles. Las tecnologías más populares se describen en detalle, sin embargo, en el apéndice B se encuentra más información sobre cada uno de estos lenguajes.

2.2 El Lenguaje HTML

El lenguaje de marcadores de hipertexto (HTML; "HyperText Markup Language") es un lenguaje basado en etiquetas y diseñado con el fin de implementar un lenguaje sencillo. El lenguaje consiste en etiquetas o comandos que informan al navegador de la manera como debe presentar textos, gráficos y archivos multimedia. Contiene, asimismo etiquetas para vincular las páginas a otras páginas y a otros recursos.

El lenguaje tiene una estructura sencilla y es transmitido a través de la red por medio del protocolo de transporte de Hipertexto (HTTP, "Hypertext Transport Protocol"); el lenguaje puede llevar inmerso los lenguajes "script" y Java, que no son soportados por todos los navegadores. El lenguaje HTML surgió con la creación del "World Wide Web", que en sí es la creación del lenguaje y protocolo antes descritos.

La tecnología fue desarrollada por Tim Berners Lee, actual director del "World Wide Web Consortium" (W3C), la organización que dirige el futuro del WWW. Esta organización es la encargada de dar las recomendaciones para el desarrollo del "Web", actualmente la recomendación es el HTML 4. A cada documento en HTML se le llama página de "Web", y normalmente se le hace sólo referencia como página.

2.2.1 Etiquetas del lenguaje HTML

El lenguaje HTML es un lenguaje de etiquetas; estas etiquetas serán fragmentos de texto destacado de una forma especial, que permiten la definición de las distintas instrucciones de HTML, tanto los efectos que se van a aplicar sobre el texto, como las distintas estructuras del lenguaje. Las etiquetas son la base principal del lenguaje HTML. Un documento HTML es un archivo de texto con la información que se va a mostrar en la página y etiquetas que indican la forma de presentar la información.

Una **etiqueta** es un texto incluido entre los símbolos menor que (<) y mayor que (>). El texto incluido dentro de los símbolos será explicativo de la utilidad de la etiqueta. Por ejemplo:

Tabla III. Algunas etiquetas HTML

	Letra Negrita, del inglés "Bold" (negrita)
<TABLE>	Definirá una tabla
	Inclusión de una imagen

Existe normalmente una etiqueta de inicio y otra de fin; la de fin contiene el mismo texto que la de inicio añadiéndole al principio una barra inclinada (/). El efecto que define la etiqueta tendrá validez para todo lo que esté incluido entre las etiquetas de inicio y fin, ya sea texto plano u otras etiquetas HTML..

<ETIQUETA>Texto Afectado</ETIQUETA>

Por ejemplo, con la etiqueta siguiente: `texto que será en negrita`, se obtiene: **texto que será en negrita**. Algunas etiquetas no necesitan la etiqueta de fin; son aquellas en las que el final está implícito, por ejemplo `<P>` párrafo, `
` salto de línea o `` inclusión de una imagen. Definen un efecto que se producirá en un punto determinado, sin afectar a otros elementos. El uso de mayúsculas o minúsculas en las etiquetas es indiferente; se interpretan del mismo modo en ambos casos, pero lo normal es expresarlas en mayúsculas, para que destaquen con más nitidez del texto normal.

2.2.2 Atributos de las etiquetas

Las etiquetas pueden presentar modificadores que son llamados **atributos**, que permitirán definir diferentes posibilidades de la instrucción HTML. Estos atributos se definirán en la etiqueta de inicio y consistirán normalmente en el nombre del atributo y el valor que toma, separados por un signo de igual (=). El orden en que se incluyan los atributos es indiferente, y no afecta el resultado. Si se incluyen varias veces el mismo atributo con distintos valores, el resultado obtenido será impredecible, según como lo interprete el navegador. Cuando el valor que toma el atributo tiene más de una palabra, debe expresarse entre comillas; en otro caso, no es necesario. Un ejemplo de atributo será:

```
<A HREF="http://www.usac.edu.gt">Pagina principal de la USAC</A>
```

En este caso la etiqueta A, utilizada para enlazar a otra página, presenta un atributo HREF cuyo valor es <http://www.usac.edu.gt> y que da la dirección enlazada para la etiqueta. Igualmente una etiqueta podría presentar varios atributos:

<HR ALIGN=LEFT NOSHADE SIZE=5 WIDTH=50%>

En este caso, la etiqueta HR utilizada para espaciar, presenta cuatro atributos. El primero indica que la alineación es por la izquierda; el segundo atributo NOSHADE es un caso especial que no presenta valor, e indica que no tiene sombra. El tercer atributo indica que el tamaño es de 5 líneas de altura, y el cuarto, que el ancho es el 50 por ciento de la ventana actual. El orden en que se especifiquen los atributos no afectará el resultado final.

2.2.3 Formato de las URL

URL es el acrónimo de ("Uniform Resource Locator", Localizador de recursos uniforme), localizador uniforme de recursos y permite localizar o acceder de forma sencilla, cualquier recurso de la red desde el navegador de la WWW. Con la WWW se pretende unificar el acceso a información de servicios que antes eran incompatibles entre sí, tratando de conseguir que todos los servicios de Internet sean accesibles a través de la WWW; de esta forma desde un mismo programa se puede tener acceso a todos los recursos de una forma uniforme y permite que los documentos HTML incluyan enlaces a otras fuentes de información en servicios como FTP, gopher, WAIS, etcétera.

Las URL se utilizan para definir el documento; destino de los hiperenlaces, para referenciar los gráficos y cualquier otro fichero que se desee incluir dentro de un documento HTML. Cada elemento de Internet tendrá una URL que lo defina, ya sea que se encuentre en un servidor de la WWW, FTP, o las "News". El formato de una URL es:

Servicio://maquina.dominio:puerto/camino/archivo

Tabla IV. Servicios utilizados en los URLs

Http: - "HyperText Transport Protocol" - Protocolo de transporte de hipertexto	Es el protocolo utilizado para transmitir hipertexto. Todas las páginas HTML en servidores WWW deben ser referenciadas mediante este servicio. Indica conexión a un servidor de la WWW.
Https: - "HyperText Transport Protocol Secure" - Protocolo seguro de transporte de hipertexto	Es el protocolo para la conexión a servidores de la WWW seguros. Estos servidores son normalmente de ámbito comercial y utilizan encriptación para evitar la interceptación de los datos enviados, generalmente números de tarjeta de crédito, datos personales, etc ...; realiza una conexión a un servidor de la WWW seguro.
Ftp: - File Transfer Protocol - Protocolo de transferencia de archivos	Utiliza el protocolo FTP de transferencia de archivos. Se utiliza cuando la información que se desea acceder se encuentre en un servidor de ftp. Por defecto, se accesa a un servidor anónimo ("anonymous"); si se desea indicar el nombre de usuario, se usará: ftp://maquina.dominio@usuario y luego pide la clave de acceso.
"Gopher", "Wais":	Cualquiera de estos servicios de localización de información, indicará el directorio para localizar el recurso concreto.
"News":	Accede al servicio de "news", para ello el visualizador o navegador de la WWW debe ser capaz de presentar este servicio (no todos lo son). Se indica el servidor de "news" y como camino, el grupo de noticias al que se desea acceder: news://usac.edu.gt/sistemas.gt.
Telnet:	Emulación de terminal remota, es utilizada para conectarse a una máquina multiusuario, para acceder a cuentas públicas como por ejemplo la de una biblioteca. Lo normal es llamar a una aplicación externa que realice la conexión. En este caso, se indicará la máquina y el "login": telnet://maquina.dominio@login.
Mailto:	Se utiliza para enviar correo electrónico (no todos los navegadores lo soportan). En este caso, sólo se indicará la dirección de correo electrónico del destino: mailto://alias.correo@domino.

La **máquina.dominio** de un URL indicará el servidor que proporciona el recurso; en este caso se utilizará el esquema IP para identificar la máquina o el nombre de la máquina y el dominio. En el caso de nuestra Universidad, el dominio siempre será usac.edu.gt, por tanto un nombre válido de máquina será www.usac.edu.gt.

El puerto en una URL es opcional y generalmente no se pone si es el mismo que utiliza el servicio por omisión. Solo se utilizará cuando el servidor utilice un puerto distinto al puerto por defecto. El camino será la ruta de directorios, que hay que seguir para encontrar el documento que se desea referenciar. Para separar los subdirectorios, se utiliza el símbolo de diagonal utilizado para los caminos en el sistema operativo UNIX (/); se usa por convenio al ser este tipo de máquinas las más usadas como servidores. El nombre de los subdirectorios y del fichero referenciado puede ser de más de ocho caracteres, y se tendrá en cuenta la diferencia entre mayúsculas y minúsculas en el nombre.

El navegador de la WWW realiza una acción para cada tipo de fichero, sólo los que sean del tipo text/html serán mostrados como documentos HTML. En caso que el tipo no sea conocido por el cliente, se considerará por defecto como un documento de texto normal. Si no se indica un fichero y sólo se hace referencia a un directorio, se accede a la página html por defecto de ese directorio.

En el servidor, están definidos nombres de archivos para ser usados, si no se indica uno por defecto. En caso que no exista este archivo, se mostrará un listado de todos los documentos que forman el directorio, si el servidor de "Web" está configurado con ese permiso de visualización. Este archivo es la página inicial ("home page") del servidor o del espacio "Web".

2.2.4 Estructura de un documento HTML

Un documento HTML está definido por una etiqueta de apertura <HTML> y una etiqueta de cierre </HTML>. Dentro de éste, se dividen dos partes fundamentales la cabecera, delimitada por la etiqueta <HEAD> y el cuerpo, delimitado por la etiqueta <BODY>. Por tanto, la estructura de un documento HTML será:

Tabla V. Estructura de una página HTML

<HTML>
<HEAD>
Definiciones de la cabecera
</HEAD>
<BODY>
Instrucciones HTML
</BODY>
</HTML>

Ninguno de estos elementos es obligatorio, por lo que es posible crear documentos HTML, que se muestren sin ningún problema sin incluir estas etiquetas de identificación. Si se utilizan elementos que forzosamente deban ser incluidos en la cabecera (como la etiqueta de título), no serán reconocidos correctamente, si no se incluyen entre las etiquetas de <HEAD>.

Para insertar comentarios dentro de un documento HTML, se utiliza la etiqueta especial <!--, definiéndose un comentario de la forma:

<!--Esto es un comentario y no se despliega en la página -->

Los comentarios son útiles para identificar el documento, pudiendo indicar al comienzo del documento, su título, autor y la fecha en que fue realizado; esto se hace antes de la etiqueta <HTML>. También se emplean pero con menor frecuencia, para comentar cualquier instrucción o circunstancia del documento. Los comentarios no se muestran en el documento HTML y el único modo de verlo, es viendo el código HTML fuente del documento, cosa que permiten algunos navegadores de la WWW.

2.2.5 Cabecera de un documento HTML

La cabecera de un documento HTML está delimitada por las etiquetas <HEAD> y </HEAD>; en ésta, se incluirán las definiciones generales que afectarán a todo el documento. Todas sus etiquetas son opcionales y se utilizarán sólo en casos muy determinados; únicamente la etiqueta TITLE tiene un uso general y aunque es opcional, se recomienda incluirla en todos los documentos creados. A continuación, se citan los distintos componentes que pueden formar la cabecera de un documento HTML.

<TITLE>: Título del documento

Especifica el título del documento HTML; todo documento HTML debe tener un título. El título debe guardar relación con el contenido del documento y definirlo de forma general; su tamaño no está limitado, aunque se recomienda que no sea excesivamente extenso. Dentro de esta etiqueta, no se podrá usar ninguna de las restantes etiquetas HTML. El título no forma parte del

documento en sí, y no se incluye ni se muestra dentro del mismo; generalmente se muestra en la parte superior de la ventana del programa navegador.

Se utiliza principalmente para etiquetar e identificar la página en los "bookmarks" (marcadores) y las "history list" (lista de documentos accedidos), y también es utilizado por algunos servidores de búsqueda, como resultado de una búsqueda para poder intuir el contenido del documento. El título deberá guardar relación con el contenido del documento y definirlo de forma general. La etiqueta <TITLE> debe ser usada dentro de las etiquetas que definen la cabecera de la siguiente forma:

Tabla VI. Etiquetas para el título de una página

```
<HEAD>
  <TITLE>
    Título del documento HTML
  </TITLE>
</HEAD>
```

<BASE>: URL base del documento

Especifica la URL que se toma como base del documento HTML; ésta se utiliza para las referencias a URL relativas, que se encuentren en los hiperenlaces y en las referencias de las imágenes. No tiene mucho uso, y lo normal es utilizarlo en documentos, obtenidos de otros servidores o directorios e incluidos fuera de contexto; de esta forma los enlaces siguen siendo válidos. Presenta un atributo HREF, que indica la URL base del documento; el formato es el siguiente: <BASE HREF="URL">, por ejemplo: <BASE HREF="http ://www.usac.edu.gt/docs/"> .

<ISINDEX>: el documento es un índice

Indica que el documento es un índice y por tanto se debe realizar una búsqueda en él. Presenta un indicador que pregunta la palabra clave de la búsqueda. No se utiliza en documentos normales, sino en aquellos realizados con cgi-bin para indicar que se les debe pasar una palabra clave para realizar la búsqueda. Presenta dos atributos, "PROMPT" que indica el texto que aparecerá como inductor de la búsqueda, y "ACTINO" que especifica el fichero cgi que trata la consulta; por defecto será el documento actual.

<META>: Indica refresco del documento

Aunque tiene más utilidades, la principal es indicar documentos con refresco automático. Se indica un documento que debe sustituir al actual en un determinado número de segundos. El formato es el siguiente: `<META HTTP-EQUIV="REFRESH" CONTENT = "número_segundos ; URL = URL_de_refresco " >`

Se indica el número de segundos que deben pasar antes del refresco y el documento HTML, que sustituye al actual. Si se indica cero segundos, la transición entre uno y otro documento será inmediata. Si no se indica un URL, el documento actual se refresca.

La utilidad puede ser para documentos que cambien con una gran frecuencia y transcurrido un tiempo se actualizan, o documentos que deban ser vistos de forma secuencial; se empieza por un documento pasando a los siguientes transcurrido un período de tiempo, sin necesidad de acción alguna por parte del usuario.

2.2.6 Cuerpo de un documento HTML

El cuerpo de un documento HTML estará delimitado por las etiquetas <BODY> y </BODY> y en él se incluirán todas las instrucciones HTML y el texto que forman el documento; es similar al BEGIN ó {} de un lenguaje de programación. Al igual que la cabecera (HEAD), es opcional, pero se recomienda para la buena identificación de las distintas zonas del documento. Si un documento no presenta ninguna de las etiquetas de identificación de sus distintas partes (<HTML>, <HEAD> ó <BODY>), se considerará que todo lo que se defina pertenece al cuerpo del mismo. La lista de etiquetas para cada versión de HTML queda fija; simplemente se agregan etiquetas a las nuevas versiones.

En el apéndice B, se muestra una lista de enlaces a sitios en la "Web", en las cuales se incluyen etiquetas para los siguientes propósitos: espaciados y saltos de línea, cabeceras, tamaño y color de las fuentes de caracteres, definición de listas, hiperenlaces, imágenes, tablas, formularios, documentos con "frames", etcétera.

2.3 Lenguajes "Scripts"

Son lenguajes de programación orientados a documentos, es decir, que tienen una estructura sencilla para operaciones sencillas (como para verificaciones de formato), que generalmente son extracciones de un lenguaje completo de programación, como por ejemplo Java, Perl o Visual Basic. La interpretación de dicho código que se realiza en el momento de solicitar la página, puede ejecutarse en el cliente (navegador) o en el servidor de "Web", y es útil por ejemplo, si se desea mostrar la fecha y hora actual del servidor; entonces al código se le solicita que corra en el servidor. Por ejemplo:

```
<SCRIPT LANGUAGE="JavaScript" RUNAT:"Server">
```

Si fuera el caso de que se desea que se muestre la hora y fecha actual de la computadora del cliente, simplemente se elimina el parámetro RUNAT de la etiqueta SCRIPT. Por default, los lenguajes "script" corren en el cliente. El más antiguo y popular en los clientes es JavaScript, sin embargo, por la cantidad de servidores "Web" de Microsoft y ya que VBScript se desempeña mejor en los servidores de IIS, este lenguaje también se ha hecho más popular.

Debido a que no todos los navegadores soportan los lenguajes "script", se puede colocar todo el código "script" dentro de un comentario de HTML (<!-- el código Script va aquí -->); si el navegador soporta el lenguaje, lo ejecutará; de lo contrario, no sucederá nada en el navegador. Como es muy probable que el navegador lo soporte, es necesario marcar el final del comentario HTML, como comentario del lenguaje "Script". Por lo que el código queda así:

Tabla VII. Forma de inhabilitar JavaScript en Navegadores que no lo soportan

```
<SCRIPT LANGUAGE="JavaScript">
<!-- // Esto es un comentario en el lenguaje script
      document.write( 'Este código se ejecutó porque el navegador soporta' );
      document.write( ' Los lenguajes script ' ); // -->
</SCRIPT>
```

Esta es la forma correcta de escribir cualquier código con un lenguaje "Script", debido a que el desarrollador no puede estar seguro que todos los clientes tienen navegadores que soporten el lenguaje. A continuación, se describen los dos lenguajes "script" más populares en Internet.

2.3.1 JavaScript

El lenguaje JavaScript es el complemento ideal del HTML, al permitir a la página, realizar algunas tareas por sí misma, sin necesidad de sobrecargar el servidor del cual depende. Entre estas tareas, puede estar, por ejemplo, realizar algunos cálculos simples, formatear un texto para que sea leído por distintas personas de manera distinta, proveer de un medio de configurar la visualización de una página, realizar un pre-chequeo de validación en un formulario antes de enviarlo, etcétera. La especificación del grupo que desarrolló el JavaScript se puede encontrar en <http://developer.netscape.com>

2.3.1.1 Inclusión de JavaScript:

Existen principalmente tres lugares de una página donde puede aparecer un trozo de lenguaje JavaScript:

- Usando la directiva (normalmente dentro de las etiquetas <HEAD></HEAD>), <SCRIPT LANGUAGE = "JavaScript"> ... </SCRIPT>. Por ejemplo:

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
  document.write( 'Esto es una prueba' );
  // -->
</SCRIPT>
```

- Utilizado como respuesta a algún evento. Por ejemplo:

```
<INPUT TYPE="checkbox" onClick="window.status='Usted dio click.'; return true;">
```

- Incluido en el atributo HREF del elemento A. Por ejemplo:

```
<A HREF="javascript:window.history.back()">Volver</A>
```

En la práctica, las dos primeras son más usadas que la última.

2.3.1.2 Sintaxis básica

La sintaxis de JavaScript es muy similar a la del lenguaje C. Las instrucciones terminan con un punto y coma (;), se pueden agrupar usando llaves ({}), y un doble "slash" (//) indica que el resto de la línea es un comentario.

Los operadores matemáticos se mantienen: suma (+), resta (-), multiplicación (*), división (/), módulo (%). Además existe el autoincremento (++) y autodecremento (--). Se pueden concatenar strings de manera muy simple usando el operador de suma (+). Las asignaciones de variables se realizan usando el signo igual (=). Adicionalmente existen los siguientes operadores (-=, +=, *=, /=, %=), que funcionan igual que el lenguaje de programación C. Los operadores lógicos ("or" ||, "and" &&, "not" !, xor ^, "shiftright" >>, "shiftright" w/0 >>>) también están implementados, junto con las comparaciones (<, >, ==, !=, <=, >=).

2.3.1.3 Variables

Es necesario declarar todas las variables que se van a utilizar. Si se declara fuera de una función, es una variable global, si no es local. Se declara una variable sin indicar explícitamente su tipo, sino usando la instrucción: `var nombre;` donde `nombre`, es el nombre de la variable. Más adelante en el programa, el mismo intérprete le asigna el tipo apropiado. Esto para seguir la filosofía de diseño de JavaScript que indica que se realizarán programas pequeños; la idea es lograr que el programador realice los "scripts" de la manera más rápida posible.

2.3.1.4 Instrucciones de control lógico

Existen las instrucciones: "if" ... "else", "for", "while" ... "break" ... "continue", "with", "function" ... "return", que funcionan de manera idéntica al lenguaje C y la instrucción "for" ... "in". Algunos ejemplos se muestran a continuación:

Ejemplo 1: Suponiendo que la variable `myvar` existe y que hay un formulario llamado "myform" que tiene un input de tipo text llamado "texto1".

```
if ( myvar == 0 ){
    document.myform.texto1.value = 'Error'; }
else
    document.myform.submit();
```

Ejemplo 2: aquí habrá un error si el formulario "myform" tiene menos de 31 elementos. Los elementos en general en JavaScript se enumeran comenzando por el elemento cero (las opciones de un "select", o de un "radio button"; los elementos de un documento o los elementos de un formulario, o los formularios de un documento).

```
For ( myvar = 0; myvar <= 30; myvar++ )  
    Document.myform.elements[myvar].value = " ";
```

Ejemplo 3: un ciclo while típico que no hace nada excepto hacer position = 30.

```
Var position;  
Position = 0;  
While ( position < 30 )  
    Position++;
```

Ejemplo 4: una vez especificado with, el resto de las variables utilizadas comienzan con el argumento de with se asume que existen dos formularios en el documento; el segundo formulario puede no tener nombre, por lo que se referencia por su número dentro del arreglo especial 'form'.

```
With ( document.form[1] ){  
    nombre.value = "Ingrese aquí su nombre";  
    edad.value = "Puede dejarme en blanco si lo desea";}
```

Ejemplo 5: for ... in ejecuta una acción con cada elemento de un arreglo

```
for contador in Arreglo  
    write( Arreglo[contador] );
```

2.3.1.5 Palabras reservadas

Además de las nombradas antes, existe la instrucción **"alert"**, que muestra al usuario una ventana con el mensaje dado como parámetro en la llamada a esa función. Esta instrucción es muy útil en el momento de programar y depurar código JavaScript. Su uso es muy simple: alert("mensaje");

Otras palabras reservadas son: "abstract, boolean, byte, case, catch, char, class, const, default, do, double, extends, false, final, finally, float, goto, implements, import, instanceof, int, interface, long, native, null, package, private, protected, public, short, static, super, switch, synchronized, throw, throws, transient, try, void". Una descripción más técnica de estas palabras puede ser encontrada en los sitios del Web, que se encuentran en el apéndice B; es la parte de JavaScript.

2.3.1.6 Funciones

Para declarar una función se usa la siguiente sintaxis:

Function NombreFuncion (parametro1, parametro2, ..., parametro N)

No existe un cuerpo principal del programa ("main"), puesto que todo lo que no esté dentro de una función es ejecutado mientras se va cargando la página, conforme va apareciendo. Cuando se declara una función, tampoco es necesario indicar qué tipo de valor retornará, si es que retorna alguno. Si importa el orden en que se declaren las funciones, una función debe ser declarada antes de usarse.

2.1.3.7 Objetos

Los siguientes son algunos de los objetos con los que los "scripts" pueden interactuar:

Tabla VIII. Objetos de JavaScript

Globales	
"Document"	El documento cargado.
"Window"	La ventana activa
"Form"	Un formulario del documento, identificado por su atributo NAME además de todos sus elementos, identificados también por sus nombres.
"History"	La historia de la ventana o de un "frame"
Objetos especiales	
"Date"	Un objeto genérico conteniendo una fecha.
"Math"	Una biblioteca incorporada de funciones matemáticas y constantes.

Tabla IX. Ejemplos de uso de los objetos de JavaScript

window.history.go(0)	Recarga la página actual (método)
document.referrer	Contiene el URL de la página de la cual venimos (no puede modificarse)
window.status	La barra de estado (si puede modificarse)
window.history.back()	Regresa a la página anterior
var pi = Math.pi	Biblioteca de funciones matemáticas
window.defaultStatus	La barra de estado, cuando nada mas aparece en ella
document.form1.submit()	Envía el formulario llamado form1
window.location = 'index.html'	Actúa con la ventana.

2.3.1.8 Eventos

Un evento debe asociarse a un elemento HTML, no a un código JavaScript. Generalmente se usan para invocar a funciones que realizan, alguna operación con el mismo elemento que las invoca; por eso se define la palabra reservada **"this"**, para referirse al objeto, cuyo evento invocó a la función.

Tabla X. Eventos de JavaScript

Evento	Descripción	Elementos que lo admiten	Ejemplo
"OnBlur"	Perder el cursor	"text, select, textarea"	<INPUT TYPE="text" onBlur="chequea_valor(this.value)">
"OnChange"	Cambiar o perder cursor	"text, select, textarea"	<INPUT TYPE="text" onChange="Set1(this.value)">
"OnClick"	Hacer "click" sobre el elemento	"button, checkbox, link, radio"	<INPUT TYPE="button" VALUE="Enviar 1" onClick="document.myform.oculto1 = 1; document.myform.submit();">
"OnFocus"	Ganar el cursor	"text, select, textarea"	<TEXTAREA onFocus="this.value = ""> Esto se borrara</TEXTAREA>
"OnLoad"	Cargarse una página	"body"	<BODY onLoad="preparar_formulario(document.myform)">
"OnUnload"	Descargarse una página	"Body"	<BODY onUnload="document.ocultaform.submit()">
"OnMouseOver"	Pasar el mouse por encima	A	
"OnSelect"	Seleccionar texto	"text, textarea"	<INPUT TYPE="text" onSelect="window.status='Presione espacio ahora para borrar el texto seleccionado'">
"OnSubmit"	Enviar un formulario	"Form"	<FORM onSubmit="window.defaultStatus = 'Espere un momento mientras se procesa el formulario ...'">

2.3.2 Visual Basic "Script"

VBScript fue liberado con el Microsoft "Internet Explorer" 3.0, para toda la comunidad desarrolladora especializada en Visual Basic. El lenguaje de programación Visual Basic permite hacer prácticamente cualquier tipo de aplicación, y ahora en Internet, VBScript permitirá crear "Scripts" a la altura de JavaScript en los visualizadores más comerciales, entre ellos el "Explorer" de Microsoft y las próximas versiones de NetScape.

De esta manera, los desarrolladores se ven beneficiados al encontrar una opción de su herramienta de desarrollo, en casi cada uno de los ámbitos que se manejan en la computación moderna.

VBScript es un subconjunto de Visual Basic, que fue diseñado para integrarse a la tecnología ActiveX presentada por Microsoft y aprovechar sus capacidades para crear contenido activo en la "Web". Incluye la misma ideología orientada a eventos presentada por VB y muchas de las capacidades propias de este lenguaje, con la adición de estrategias de seguridad, que permiten crear scripts confiables, eficientes y compactos. Para poder tener vbscript en el lado del cliente, se requiere tener el "Internet Explorer" 3.0 o posterior, de Microsoft.

2.3.2.1 El Funcionamiento de VBScript

Todas las instrucciones de VBScript funcionan en Visual Basic, sin embargo, esto no ocurre a la inversa. VBScript es un lenguaje creado expresamente para la formulación de "scripts", éstos son una serie de instrucciones destinadas a usarse para hacer más eficiente el intercambio de información en Internet. Como no es posible poner todas las instrucciones de Visual Basic en el "script", debido a que podría realizarse operaciones como formatear el disco, se excluyeron varias instrucciones del VBScript, tal como aquellas que permiten acceder al API de Windows, otras que permiten el acceso a disco y otro gran número de instrucciones potencialmente peligrosas.

De este modo, en VBScript, se ve severamente recortada la cantidad de instrucciones disponibles para el usuario, con el objeto de que sea útil para crear el contenido activo de la página, hacer más eficiente la validación de la información contenida en los formularios, procesar

cierta información dada en los formularios, manipular los objetos intrínsecos de HTML y los de tipo ActiveX, crear juegos, interfaces de bases de datos, etcétera. Todo esto del recorte de instrucciones no es, sino una medida que cualquier lenguaje de formulación de "scripts" (como JavaScript) ha tenido que enfrentar en pro de la seguridad.

Para que un "script" de Visual Basic funcione en un navegador, debe tener las posibilidades de interpretar el código relacionado. Todos los navegadores pueden interpretar HTML en cualquiera de sus versiones, sin embargo, no todos pueden interpretar un "script". Para esto, se necesita que el navegador incluya el archivo en tiempo de ejecución (Run-Time) de VBScript. Por el momento, solamente el "Explorer" de Microsoft (y con agregados, el Netscape) tiene este archivo en tiempo de ejecución incluido, lo que hace que el VBScript sólo funcione en Windows y en Macintosh, pero se espera que otros navegadores incluyan estas posibilidades, de modo que el VBScript sea un lenguaje tan extendido como el JavaScript.

Ahora, no se debe creer que incluir el archivo en tiempo de ejecución de VBScript es costoso, trabajo secreto o algo por el estilo. Si se está desarrollando un navegador (o cualquier programa que requiera del uso de macros) y se quieren añadir las capacidades de VBScript, deberá, entonces, entrar al sitio de Microsoft en <http://www.microsoft.com/vbscript>, para obtener la información necesaria para añadirle tales capacidades a su navegador. Esta información es gratuita y se encuentra disponible para todo aquel que las requiera libre de regalías.

VBScript, al igual que JavaScript, es un lenguaje 100% interpretado. Sin embargo su velocidad de ejecución es lo suficientemente adecuada como para hacer más eficiente un proceso de validación de información directamente en la máquina cliente, y así evitar el tráfico en la Internet tan sólo para hacer la misma validación y devolver un mensaje de error, por encontrar algún dato no válido. Este tipo de cuestiones se realizan con mayor eficiencia del lado del cliente,

y así se usa la línea tan sólo para enviar los datos ya verificados con la consecuente eficiencia en el tiempo utilizado para la conexión.

2.3.2.2 Aspectos de sintaxis

VBScript utiliza únicamente el tipo de datos "Variant" para almacenar la información en variables. Esto hace que los programadores de VB que gustan declarar el tipo de datos de la variable explícitamente, tengan que conformarse con tan sólo un Dim Variable. Al igual que en Visual Basic, existe la opción de forzar la declaración explícita de variables con "Option Explicit".

La declaración de constantes, mediante Const, es imposible en VBScript. Sin embargo, se pueden emular las constantes mediante el uso de variables y con las normas de programación propuestas, donde se distinguirá visualmente una variable de una constante, si esta última está completamente en mayúsculas (IVA, PI, VELOCIDAD_DE_LA_LUZ, MI_CONSTANTE).

Contrario a lo que pasa en Visual Basic, donde es posible declarar una variable llamada "Niños", VBScript no acepta caracteres fuera de los que se incluyen en el abecedario inglés. Así pues, todas las letras acentuadas y la ñ, están descartadas para el uso de identificadores de procedimientos o de variables.

2.3.2.3 Operaciones no permitidas en VBScript a diferencia de VB

La instanciación de objetos no es posible. No se puede crear un objeto a partir de otro.

No se permite manejar arreglos con un límite inferior distinto a cero, acceder al portapapeles, manejar colecciones, crear compilación condicional, usar "DoEvents, Gosub...Return, Goto, On Error GoTo, On...Gosub, On...Goto", Números de línea, etiquetas o "With...End UIT". La conversión de tipos mediante "Ccur, Cvar, CVDate, Format(\$)" y Val" no se puede realizar. Además no existen herramientas de depuración más allá del Script Editor de Microsoft (que viene integrado al "Explorer" 4, ó puede integrarse como un agregado al "Explorer" 3.0x).

No se pueden declarar llamadas al API, DLL, propiedades, variables públicas, ni parámetros opcionales o de arreglo. No hay acceso a la entrada/salida de archivos. No existen las funciones financieras; no existen los métodos gráficos, no se puede usar "Arrange, ZOrder, SetFocus, Drag, Hide, Show, Load, Unload, Move, PrintForm, Refresh, AddItem" ni "RemoveItem" para manejar los objetos a los que se les aplique.

Cabe recordar que las aplicaciones creadas en VBScript corren directamente en el entorno del navegador que se utiliza, tal como ocurre con JavaScript; por eso se debía cuidar mucho, que el lenguaje no permita que algún programador malintencionado, pudiese llevar a cabo alguna tarea desastrosa en la máquina sin darse cuenta. Por añadidura, el código de VBScript se transmite a la par de la página "Web" en modo de texto; no está compilado en ninguna forma y por lo tanto sobrecarga un tanto la página que el usuario está bajando a su máquina. Por esto, el código de VBScript debe ser compacto, para que a la vez que se ejecute con mayor eficiencia, se reduzca el tiempo de transmisión.

2.3.2.4 Recomendaciones para crear páginas con VBScript

Aunque es una buena práctica de programación comentar el código fuente, es una buena idea no sobrecargar de comentarios una página acompañada con VBScript, dado que esas mismas líneas de comentarios también son transmitidas y por lo tanto, tomará un poco más de tiempo lograr la completa transmisión de la página. Así pues, se debe procurar usarlos con moderación y preferentemente en los procesos medulares.

No se debe intentar solucionarlo todo con VBScript (con cualquier lenguaje de formulación de guiones). HTML, por sí mismo, provee muchas herramientas para la solución de problemas (lo que se mejora con HTML Dinámico). VBScript, así como cualquier lenguaje acorde, están hechos para asistir a HTML en sus deficiencias y permitir que el usuario sienta que con la página activa, tiene una mejor respuesta que con una página estática creada sólo con HTML.

Se debe combinar lo mejor de VBScript con lo mejor de HTML, ActiveX y Applets de Java, para poner en manos del usuario (Navegador de Internet o de Intranet) una aplicación robusta y seria. No existe un sólo lenguaje mágico que contenga todas las soluciones sencillas a todo.

Se debe procurar colocar sus guiones o "scripts" entre líneas de comentario HTML, para evitar que aquellos navegadores que no soporten los guiones de VBScript, visualicen en su lugar una derrama incesante de bytes en su pantalla, lo que haría que se perdiera la armonía que tanto trabajo le costó crear en su página "Web".

2.4 HTML Dinámico (DHTML)

A diferencia del HTML, el DHTML no es un estándar sencillo. En lugar de eso, es un conjunto de tecnologías que permiten un contenido "Web" más dinámico, en el sentido de que utilizando un lenguaje "script" como JavaScript, es posible utilizar nuevas características de los navegadores, para que, sin necesidad de cambiar de página o llamar otra, sea posible realizar cambios en la página actual.

Los tres componentes principales del HTML dinámico son: el lenguaje HTML, "Cascading Style Sheets" (CSS) y un lenguaje de programación como JavaScript, que el WWWC estandarizó como EcmaScript. Estos tres componentes están unidos por el DOM, "Document Object Model". Una ventaja del DHTML es que se transfiere parte del procesamiento del servidor "Web" a la PC del cliente. Hasta el momento, el principal problema para el DHTML ha sido la compatibilidad entre visualizadores. El HTML y los lenguajes "Script" ya fueron descritos; se procede a dar una descripción del CSS.

CSS ("Cascading Style Sheets"; Hoja de estilo en cascada). El WWWC da dos recomendaciones para las hojas de estilo "Web", CSS1 y CSS2, que reciben un soporte extenso, aunque no consistente, en las versiones 4.x y posteriores de Microsoft "Internet Explorer" y "Netscape Navigator". La ventaja de CSS es que permite especificar las coordenadas (x, y) para colocar los objetos en una página, al nivel de píxeles. Además, CSS permite colocar capas de texto, imágenes u otros objetos, incluso a un lado de la página, especificando una tercera coordenada z. Utilizando un lenguaje de programación relativamente sencillo, es posible crear muchos efectos como intercambios de imágenes, la transiciones de páginas y las exhibiciones de diapositivas

2.5 Páginas HTML generadas por código ejecutado en el servidor de WWW

El significado de "páginas dinámicas", para Internet, ha tomado distintos significados y existen diversos lenguajes, tecnologías, formas de hacerse, y cada una tiene propósitos distintos. Lo común en estos métodos es que el código de programación que hace las páginas HTML dinámicas, es ejecutado en el servidor WWW. La ventaja es una mayor seguridad para el cliente debido a que al cliente únicamente llegan los resultados en formato HTML, y no sabe qué sucede en el servidor. La desventaja es que la interacción con el usuario es más lenta.

A continuación, se presenta un esquema de esta forma de páginas dinámicas y algunos ejemplos dentro de esta categoría; después del esquema, se describe más a detalle.

- CGI
- Páginas HTML con código de programación inmerso
 - IDC de Microsoft
 - ASP de Microsoft
 - PHP/FI de Rasmus Lerdorf
 - ServLets con Java

2.5.1 "Common Gateway Interface" (CGI)

Interfaz de Pasarela Común. Es una interfaz entre programas de aplicación y servicios de información. Hasta el momento, se ha utilizado únicamente como pasarela hacia servidores WWW que utilizan el protocolo HTTP. Sin embargo, ha sido planteada como pasarela independiente del servicio y de la aplicación.

Un documento HTML es algo estático, permanente, lo que no se adecua a los requisitos de interactividad necesarios en una aplicación del estilo de la tele-educación, como el acceso a información en constante actualización, consulta a bases de datos o seguimiento, control o recuperación de resultados de un determinado proceso. En este tipo de aplicaciones, es necesario acceder a una información que se está generando "en tiempo real". Así pues, es necesario contar con algún tipo de pasarela como la que define CGI.

La interfaz define una forma cómoda y simple de ejecutar programas que se encuentran en la máquina en la que se aloja el servidor. Para el cliente, presenta una ventaja en el aspecto de la seguridad, ya que no tendrá que ejecutar ningún programa de efectos desconocidos en su sistema local. Desde el punto de vista humano, además de eliminar la necesidad de aprendizaje, se resuelven los problemas de mantenimiento, operación y distribución de clientes, ya que el acceso se realizará a través de cualquier cliente estándar de WWW y la comunicación se efectuará según el protocolo HTTP.

Sin embargo, aunque en el servidor ya no será necesario permitir el acceso a extraños, disponer de un programa CGI en un servidor implica que virtualmente todo usuario de Internet será capaz de ejecutar un programa en ese servidor, por lo cual es necesario atender ciertas precauciones concernientes a la seguridad. La primera de ellas suele ser centralizar la localización de todos los CGIs de un servidor en un directorio reservado, efecto al cual se permitirá el acceso únicamente al "Webmaster". Todo servidor de WWW es configurable, de tal forma que al recibir una petición de acceso a un documento alojado en tal directorio, considerará que se trata de un programa ejecutable.

Como su propio nombre indica, CGI es una interfaz entre servidores de información y programas de aplicación. Por tanto, define una serie de reglas que deben cumplir, tanto las

aplicaciones, como los servidores, para hacer posible la presentación de resultados de programas ejecutables en tiempo real, a través de servicios de información estandarizados. Al tratarse de una interfaz, no existe ningún tipo de dependencia con el lenguaje de programación empleado. Así pues, es posible encontrar aplicaciones CGI programadas en lenguajes como en C, C++, fortran, perl, tcl, Visual Basic, AppleScript y cualquier "shell" de UNIX.

2.5.1.1 Fundamentos del CGI

La interfaz CGI es fruto de las conversaciones que se han venido realizando a través de una lista de correo: la www-talk y en su mayor parte, se debe a Rob McCool, John Franks, Ari Luotonen, George Phillips y Tony Sanders. En Enero de 1996, D. Robinson de la Universidad de Cambridge, realizó el primer resumen de RFC ("Request For Comments"; Solicitud de Comentarios) describiendo la interfaz, aunque se puede encontrar documentación desde mucho antes en el servidor del W3 Consortium, <http://www.w3.org/pub/WWW/CGI> . En cualquier caso, este "draft" de RFC aún cuenta con numerosas ambigüedades.

Para invocar un "script" CGI se emplea una URL que identifica al "script", el cual deberá encontrarse en un directorio especial destinado a tal efecto, para lo que debe estar configurado el servidor de información. El formato de esta URL es :

```
script-url=protocolo:/"SERVER_NAME":SERVER_PORT enc_script enc_path_info"?QUERY_STRING
```

donde enc_script corresponde al nombre del "script", bajo el nombre lógico del directorio asignado a CGIs por el servidor y enc_path_info y QUERY_STRING se explican a continuación.

2.5.1.2 Variables de entorno

La especificación CGI define una serie de variables que el servidor debe generar en el entorno de ejecución de los "scripts" que invoca. Estas variables no son "case sensitive" y un valor NULL indica que no han sido definidas. Entre ellas, se hace una distinción entre variables primarias y variables secundarias.

Se pueden considerar **primarias** aquellas variables cuyo valor puede fijar el usuario directamente en la invocación del "script", y contienen información que pueda definir la función del "script", como es el caso de parámetros.

Como variables secundarias, se consideran todas las demás, cuyo contenido es asignado por el servidor de información atendiendo a diversos elementos que definen el estado de la comunicación o el valor de las variables primarias.

Como variables primarias están, por tanto, únicamente dos:

"QUERY_STRING"

Como se puede deducir de la sintaxis correspondiente a la URL de un "script", esta variable contendrá todo lo que figure a partir del signo "cierre de interrogación" (?) en este URL. La especificación exige que esta información se encuentre de forma URL-codificada, lo que implica que se debe sustituir los espacios en blanco por el signo "más" (+) y los caracteres especiales deben indicarse en hexadecimal precedidos por un signo "tanto por ciento": %xx. Siempre que el "script" no se encuentre invocado por un formulario, podrá acceder a esta información ya URL-decodificada como si se le hubiera facilitado desde la línea de comandos, en forma de

argumentos. Así, una "shell" podrá acceder a ella a través de las variables \$1, \$2 ... y un programa en C, a través de la función argv.

"PATH_INFO"

A través del URL de invocación de un "script", es posible facilitar cierta información adicional al mismo, a través del campo "enc_path_info". El "script" CGI podrá acceder a esta información a través de la variable de entorno "PATH_INFO". El uso habitual es la indicación de "paths" de búsqueda de ficheros al "script", de donde toma el nombre. Sin embargo, la semántica de esta información viene dada por el programador.

Como variables secundarias están:

"AUTH_TYPE"

Es posible establecer protección de acceso a ciertos URLs mediante la adecuada configuración del servidor de información. Si este es el caso de la URL del "script", se emplea esta variable para indicar el método de protección.

"CONTENT_LENGTH"

Indica el tamaño de la información asociada a la invocación ("attachment"), o toma el valor NULL, si no se ha asociado información. Esta información puede provenir, por ejemplo, de un método POST o PUT de HTTP. Un ejemplo sencillo es la información facilitada por un formulario.

"CONTENT_TYPE"

Indica el tipo MIME correspondiente a la información que acompaña a la invocación, si procede. Sólo en caso de que esta variable no tenga valor, el "script" debe tratar de deducir el tipo de la información examinándola. Si aún así no se puede determinar de qué tipo es, se asumirá el tipo "application/octet-stream", que es uno de los posibles valores asignados por el MIME.

"GATEWAY_INTERFACE"

Indica la versión de la especificación CGI que cumple el servidor, de la forma "CGI/versión", por ejemplo "CGI/1.1"

"PATH_TRANSLATED"

Indica el valor de "PATH_INFO" una vez el servidor de información ha realizado la conversión de direcciones lógicas a direcciones físicas. El uso de esta variable es interesante, ya que un URL no siempre designa un objeto físico, como es un fichero. Sin embargo, no es recomendable usarla si se desea hacer portables los "scripts", ya que depende de la implementación del servidor.

"REMOTE_ADDR"

Indica la dirección IP del agente que invocó al CGI.

"REMOTE_HOST"

Indica el nombre completo (o NULL) del agente que invocó al CGI.

"REMOTE_IDENT"

Si el servidor remoto admite identificación del usuario, proporciona el nombre del usuario al que pertenece el cliente. Este valor debe únicamente ser considerado como información complementaria o elaboración de estadísticas, pero nunca como información de autenticación.

"REQUEST_METHOD"

Esta variable sí es "case sensitive" y proporciona el método HTTP empleado para facilitar información al servidor: GET, HEAD, POST o cualquier extensión.

"SCRIPT_NAME"

Proporciona el URL que identifica al "script" CGI. Es útil para referirse a sí mismo.

"SERVER_NAME"

Proporciona el nombre del servidor tal y como se indicó en la parte <host> del URL de invocación del "script". Este nombre puede, por tanto, ser una dirección IP, un alias de DNS o un nombre propiamente dicho.

"SERVER_PORT"

Funciona análogamente a SERVER_NAME indicando el puerto de atención del servidor.

"SERVER_PROTOCOL"

Proporciona la versión y extensión del protocolo empleado por el servidor de información. Por ejemplo: "SERVER_PROTOCOL = HTTP-version|extension-version".

"SERVER_SOFTWARE"

Proporciona el nombre y versión del software servidor, que sirve la petición o ejecuta la pasarela.

2.5.1.3 Variables dependientes del protocolo http

La interfaz define una serie de variables específicas para servidores de información basados en el protocolo HTTP. En general, se define una variable por cada cabecera de mensajes del protocolo y se le da el nombre de HTTP_(nombre_de_cabecera). Al darse valor a las variables se realiza un cambio del carácter guión (-) por el carácter subrayado (_). Se exceptúan aquellas variables, cuyo valor sea recogido en otras genéricas, como "Content-type" y "Content-length" o que haya sido procesado, como "Authorization".

Entre las variables HTTP_* más usadas se encuentran:

"HTTP_ACCEPT"

Indica los tipos MIME aceptados por el servidor separados por comas (,) en la forma habitual: tipo/subtipo, tipo/subtipo.

"HTTP_USER_AGENT"

Indica el "browser" empleado por el cliente de la forma software/versión biblioteca/versión.

2.5.1.4 Resultado de la ejecución de un CGI con salida PH

Según la especificación, todo CGI debe generar algo por su salida estándar. Esta salida queda directamente conectada con el servidor de información que, tras cierto proceso, la hará llegar al cliente.

Se trata del tipo más habitual de "script". Exige que el "script" genere una salida que corresponda a :

***(Cabecera_CGI|Cabecera_HTTP)NL[Cuerpo_Entidad],**

donde NL designa al carácter de avance de línea, Cabecera_HTTP es cualquier cabecera válida del protocolo HTTP y Cabecera_CGI es una de las siguientes:

Content-type

Define el tipo MIME al que corresponde la información que proporciona el "script". Esta cabecera es de uso obligatorio, si la respuesta del CGI incluye un cuerpo de información, el cual no es obligatorio.

Location

Actúa como un puntero devolviendo un URL en lugar de la información. Un cliente, al recibir esta cabecera, debería realizar automáticamente una conexión HTTP para recuperar el URL, indicado de manera transparente al usuario, por lo que lo habitual es que la salida de un CGI que genere una cabecera "Location", no contenga cuerpo de mensaje y, por tanto, no precise de una cabecera "Content-type"; en cualquier caso, dado que los "browsers" más antiguos, no reconocen la cabecera "Location", a veces aparecen ambas juntas con un cuerpo de mensaje destinado a informar al usuario de la localización de la información.

Status

Se emplea para facilitar un valor de status definido en la especificación del protocolo HTTP.

2.5.1.5 Resultado de la ejecución de un CGI con salida NPH

Un "script" NPH debe devolver un mensaje HTTP completo. El servidor enviará este mensaje completo al cliente, sin realizar ningún tipo de análisis de las cabeceras ("No Parse Headers") y el "buffering" intermedio realizado por el servidor, se reduce a la mínima expresión, por lo que en la práctica, es como si el CGI se comunicara directamente con el cliente. En la práctica, y si no se desea descender hasta el nivel del protocolo para conseguir un mayor control, es suficiente con incluir el mensaje HTTP/1.0 200 OK inmediatamente antes de la cabecera "Content-type": del

CGI para convertir un "script" CGI PH en un NPH. Otro dato a tener en cuenta es que algunos servidores deben saber que un "script" es NPH antes de ejecutarlo, por lo que es necesario nombrarlos de manera especial.

2.5.1.6 Procesamiento de formularios

Todo formulario consta de dos partes: una página HTML, (el formulario propiamente dicho); y un "script" CGI encargado de procesar la información que ésta página recoge. El envío de información se sustenta, como todo el sistema, sobre el protocolo http, y se puede realizar a través de dos métodos definidos en el mismo: GET y POST.

El "script" que procesará un documento con formulario se indica en el propio documento mediante la etiqueta <FORM>, según la sintaxis:

```
<FORM METHOD=método ACTION=script>
```

Donde METHOD es uno de los métodos HTTP: GET o POST y "script" es el URL de invocación del CGI.

Si el método elegido es GET, el servidor facilitará toda la información complementada por el usuario en el formulario al CGI, a través de la variable "QUERY_STRING". Si, por el contrario, el método elegido es POST, el servidor proporcionará toda la información al CGI a través de la

entrada estándar de este último. En ambos casos, la información se encontrará codificada según el formato que se describe a continuación.

Decodificación de la información procedente de un formulario

El formato en que se encuentra codificada la información procedente de un formulario es un flujo de pares: nombre=valor separados por el signo "ampersand" (&). Es decir:

Nombre_1=valor_1&nombre_2=valor_2&...&nombre_n=valor_n

En definitiva, se utilice el método que se utilice, habrá que procesar la información recibida según los siguientes pasos:

- a. Extraer "tokens" tomando el signo "ampersand" (&) como delimitador.
- b. En cada "token", realizar una nueva separación tomando el carácter "igual" (=) como delimitador. El primer "token" obtenido ahora corresponderá al nombre de la variable y el segundo, al valor obtenido. El nombre de la variable coincide con el que figura en el documento en el atributo NAME de cada etiqueta <INPUT>.
- c. URL, decodificar ("URL decode") los valores obtenidos.

Uso de GET y POST

No se aconseja el uso de GET para el envío de información de un formulario a un CGI, debido a que se realizará a través de una variable de entorno que puede tener una longitud más o menos limitada según el sistema. Por eso, la información presente en el formulario podría desbordarla. En caso de emplear POST, la única aclaración que se va a realizar para aquellos "scripts" que accedan a su entrada mediante una interfaz de ficheros, es que el servidor no

colocará ningún signo de terminación, como por ejemplo EOF. Por eso, si se accede de esta forma a los datos, será necesario emplear la información presente en "CONTENT_LENGTH".

2.5.2 Páginas HTML con código de programación inmerso

Este método depende del servidor de WWW, ya que éste tiene que saber qué hacer con el código introducido en la página HTML. Al obtener una petición del navegador, el servidor ejecuta el código en la página HTML y envía al navegador el código HTML resultante. Algunas tecnologías para realizar esta tarea son:

2.5.2.1 "Internet Database Conector" (IDC)

Con IDC (Conector de base de datos en Internet, de Microsoft), y el servidor de WWW Microsoft IIS, se pueden incluir resultados y consultas a bases de datos en las páginas HTML, sin necesidad de saber programar. La página creada con IDC contiene código HTML con variables especiales de la forma <%variable%>, que hace referencia a los campos de datos de la base de datos. Los resultados de la consulta a la base de datos se obtienen al ejecutar un fichero que contiene código SQL.

Para poder implementar una página que regrese resultados de una base de datos, es necesario definir en el servidor de "Web" un ODBC para la base de datos a utilizar. Luego se debe crear un archivo con extensión IDC, que tiene una estructura en la cual se define el ODBC de la base de datos donde se realizará la consulta, si es necesario el usuario y contraseña.

El "Template" es una página HTML con etiquetas especiales, que definirán cómo tratar los resultados de la consulta. Y por último, se debe escribir la instrucción SQL para la consulta o la operación que se desee realizar. Las operaciones que se pueden realizar sobre la base de datos corresponden a las instrucciones del lenguaje SQL, el cual es un estándar. Es importante decir que es posible pasar parámetros a la consulta, a través de un formulario.

Tabla XI. Código fuente de un archivo IDC

```
Datasource: ODBCBaseDatos
Username: sa
Template: Consulta.htx
SQLStatement:
+SELECT *
+FROM usuarios
+WHERE nombre like '%vNombre%'
DefaultParameters:vNombre=%
```

En este ejemplo, el nombre del ODBC es **ODBCBaseDatos**, el "Username" sirve para poder acceder a la base de datos, luego el template indica que archivo HTX recibe la información y la muestra; en este caso será Consulta.htx. A continuación, se escribe la instrucción SQL que puede utilizarse con los parámetros, y por último se definen los parámetros que recibe el archivo.

Debe existir un archivo HTX por cada IDC, ya que los resultados de la operación serán enviados a ese archivo. Este archivo es código HTML con etiquetas especiales (<% código especial %>). Entre las instrucciones que pueden colocarse en estas etiquetas, se encuentra las condicionales como el if, para mostrar los resultados se utiliza <%begindetail%> <%enddetail%>; esto realiza un ciclo hasta recorrer todos los resultados de la consulta.

Dentro de estas etiquetas se debe colocar el código para mostrar la información del resultado; esto es a través de la etiqueta <%NombreColumna%>. Un ejemplo de un archivo HTX para el IDC del ejemplo anterior es el siguiente.

Tabla XII. Código fuente de un archivo HTX

```
<html>
<title>RESULTADO DE LA CONSULTA</title>
<BODY>
<h1>RESULTADOS</h1>
<%begindetail%>
  <%if CurrentRecord NOT EQ 0 %>
    AQUI ESTAN LOS DATOS
  <%endif%>
  <%FirstName%><%LastName%><BR>
<%enddetail%>
<%if CurrentRecord EQ 0 %>
  <h2>LA CONSULTA NO REGRESO DATOS</h2>
<%endif%>
</body>
</html>
```

Los archivos IDC pueden contener varias operaciones en un solo archivo, y el archivo HTX que recibe los resultados debe escribirse para poder recibir los resultados en el orden en que se solicitaron las operaciones.

2.5.2.2 "Active Server Pages" (ASP)

Los ASP (Páginas de servidor activas, de Microsoft), se pueden emplear en el servidor de WWW IIS (a partir de la versión 3.0), como una alternativa a CGI. El código es ejecutado en el servidor WWW, sin necesidad de cargar otro programa. Esto es una gran ventaja respecto a CGI.

Las ASP son un tipo especial de páginas "Web" que incorporan "scripts" (JavaScript, VBScript, PerlScript), que se ejecutan en el servidor. Así como las páginas "Web" normales, pueden llevar "scripts" en JavaScript o VBScript para ejecutarse en la máquina del cliente, bajo los auspicios del navegador; las páginas ASP además pueden llevar "scripts" que se ejecutan en el servidor antes de enviar la página "Web".

Las ASP comparten una categoría similar a los CGI, ya que, al igual que ellos, permiten la creación de páginas "Web" dinámicas. Si por ejemplo, se quiere crear una página "Web" que muestre la hora, podría hacerse de la siguiente manera con una página ASP:

- Crear la página "Web" normalmente, y en el lugar donde debe aparecer la hora, poner una variable.
- Crear un "script" de servidor que calcule la hora y se la asigne a la variable anterior.

Cuando un navegador pida la página, el servidor "Web" hace lo siguiente:

- Carga la página y como su extensión es asp, primero se ejecutan los "scripts".
- El "script" hace que la variable tome el valor de la hora en ese momento.

- El servidor "Web" sustituye la variable por su valor y elimina el "script" de servidor de la página.
- El servidor "Web" envía el resultado al navegador como la página "Web" pedida.

De esa forma, el navegador ve siempre la página "Web" con la hora del momento, y no puede ver el "script" que genera ese resultado. Las páginas ASP se distinguen porque su extensión es ASP en lugar de HTML.

Con las ASP, Microsoft trata de que la programación de servidores de Internet sea una prolongación natural de sus otros sistemas de programación, con especial énfasis, en Visual Basic, que es el más popular. Las ventajas de las ASP se pueden resumir en lo siguiente:

Se trata de aplicaciones que van a correr en servidores Windows NT exclusivamente, por lo que pueden ser probadas localmente en un PC con sistemas operativos de Microsoft. Sin embargo estas páginas pueden generar dinámicamente código, que pueda ser leído por cualquier navegador en cualquier plataforma.

Las páginas activas incorporan objetos COM y ActiveX con la sencillez del Visual Basic. Es más que probable que empiece a surgir gran cantidad de objetos Active X de terceros, que permitan aplicaciones muy potentes con un mínimo de codificación.

El acceso a bases de datos se realiza mediante ADO (ActiveX "Data Object"), que es un sistema que resultará familiar a quienes hayan programado en Visual Basic o en Access. El ADO

trata los datos mediante objetos "Recordset". También puede utilizarse un acceso a datos más tradicional basado en ODBC.

Ofrece persistencia de datos entre distintas ejecuciones de una misma página. Esto es, se necesita la utilización de "cookies", que es una forma de almacenar en la computadora del cliente, utilizando algún lenguaje "script", o campos "type=hidden", para mantener variables entre distintas ejecuciones de un CGI. Con las ASP, se dispone de propiedades del objeto "Session", a las que se les pueden asignar valores permanentes.

Para crear páginas ASP basta con un editor de texto normal y corriente. Aún así, Microsoft dispone del Visual InterDev como herramienta de desarrollo para Internet. Las páginas ASP tienen que estar en un directorio con permisos de ejecución. Una aplicación ASP está formada por todas las páginas ASP que están en un mismo directorio y sus subdirectorios, los cuales deben tener activados los permisos de ejecución de "scripts".

Cuando se hacen "scripts" del lado del cliente (los que se ejecutan en el navegador), es mejor utilizar JavaScript que Visual Basic Script, ya que es un lenguaje que entienden los dos navegadores más comunes en Internet, mientras que Visual Basic sólo lo entiende el de Microsoft.

Sin embargo los ASP son "scripts" que se ejecutan en el servidor antes de enviar la página al cliente. Por ello, no están condicionados por el navegador sino por el servidor.

Los usuarios no podrán ver código fuente del ASP. El código ASP se ejecuta en el servidor y posteriormente es eliminado antes de enviar el resultado al cliente. Esa es otra diferencia entre los scripts del lado del cliente y los del servidor. Aún así, se puede asegurar que no vean el código fuente, quitando el permiso de lectura al directorio donde están los "scripts".

JavaScript y VisualBasic Script son versiones restringidas respecto al Java y Visual Basic respectivamente; el PerlScript es Perl 100% operativo.

Tanto JavaScript como VBScript son lenguajes pensados originalmente para ejecutarse en la máquina del cliente (en el navegador), por lo que debían tener grandes restricciones por cuestiones de seguridad, como por ejemplo, no tener funciones de escritura en disco.

Al utilizar dichos lenguajes en el servidor, donde se necesita la máxima potencia, el lenguaje no da mucho de sí y hay que utilizar objetos para casi todo.

Sin embargo el Perl es un lenguaje completo, con una gran cantidad de bibliotecas que pueden usarse en las ASP y que ha sido extensivamente utilizado para programación del lado de servidor a través de los CGI. Como no existe (prácticamente) el PerlScript del lado del cliente, no ha sido limitado en ninguna forma respecto al Perl. Así pues, se puede decir que el PerlScript es Perl dentro de una página "Web".

Ejemplos de las páginas activas se darán en detalle en el capítulo 4, el cual da ejemplo de cómo desarrollar un mantenimiento usando ASP.

2.5.2.3 PHP/FI de Rasmus Lerdorf

PHP/FI, "Personal Home Page/Form Interpreter", fue creado por el danés Rasmus Lerdorf y es gratuito para el uso no comercial. Es un lenguaje muy fácil de aprender, sobre todo para los conocedores del C. PHP/FI posibilita el acceso a bases de datos como mSQL, Oracle y otras de forma muy sencilla. PHP es muy similar a las páginas ASP de Microsoft.

PHP es una herramienta que permite crear páginas "Web" dinámicamente. Las páginas con PHP incorporado, son tratadas como una página HTML normal, las cuales es posible editar en la misma forma que las páginas HTML. El servidor debe soportar PHP y los archivos que contienen el código deben tener una extensión .php3 o la que indique el servidor.

Como ejemplo, clásico se escribe el programa para imprimir **Hola Mundo**, el cual generará una página en la cual se vea la frase "Hola Mundo". El código para dicha página es el siguiente:

Tabla XIII. Código fuente de una página con código PHP

```
<HTML>
  <HEAD>
    <TITLE>
      Prueba de PHP
    </TITLE>
  </HEAD>
  <BODY>
    <?php echo "Hola Mundo<P>";?>
  </BODY>
</HTML>
```

En este ejemplo, es posible ver una página HTML con la instrucción especial de PHP, `<?php echo "Hola Mundo<P>";?>`. La primera parte (`<?php`) indica el inicio del código PHP, luego viene el código PHP, y por último la instrucción que cierra la etiqueta que es (`?>`).

Para interactuar con las variables de ambiente en PHP, simplemente es necesario saber el nombre de la variable, las cuales son enviadas en cada solicitud de una página. Por ejemplo, para imprimir esta variable, se utiliza la instrucción de impresión "echo" dentro de las etiquetas del PHP y luego el nombre de la variable de ambiente, por ejemplo "HTTP_USER_AGENT", precedida de un signo de dólar (\$), la cual indica que es una variable de ambiente. Esta instrucción quedaría así:

```
<?php echo $HTTP_USER_AGENT; ?>
```

Esta instrucción imprime en la página, que navegador realizó la solicitud de la página, la versión y la compatibilidad, si existe la información, por ejemplo Mozilla/4.0 (compatible, MSIE 5.0, Windows NT; DigExt). La instrucción PHP no llega al navegador, únicamente el resultado.

Dentro de las etiquetas de PHP de inicio (`<?php`) y fin (`?>`), pueden estar varias instrucciones de PHP, las cuales tienen la misma sintaxis que el lenguaje C.

Una característica importante, y que le da a PHP muchas ventajas, es la facilidad de interactuar con los formularios. Cuando un formulario es enviado y éste llama la acción de un programa PHP, el programa o página PHP recibe los datos del formulario en formato de variables de ambiente. Es decir que para accederlas, únicamente es necesario colocar el símbolo de dólar

(\$) y el nombre del elemento dentro del formulario. Para poder tener una guía más completa de las instrucciones de PHP, es posible encontrar varios enlaces en el apéndice de enlaces de PHP.

2.5.2.4 ServLets

Los Servlets son módulos que extienden los servidores orientados a petición-respuesta, como los servidores "Web" compatibles con Java. Por ejemplo, un servlet podría ser responsable de tomar los datos de un formulario de entrada de pedidos en HTML, y aplicarle la lógica de negocios utilizada para actualizar la base de datos de pedidos de la compañía.

Los Servlets son para los servidores, lo que los "applets" son para los navegadores. Sin embargo, al contrario de los "applets", que se discutirán posteriormente en este capítulo, los servlets no tienen interface gráfico de usuario. Los servlets pueden ser incluidos en muchos servidores diferentes porque el API Servlet, el que se utiliza para escribir Servlets, no asume nada sobre el entorno o protocolo del servidor. Los servlets se están utilizando ampliamente dentro de servidores HTTP; muchos servidores "Web" soportan el API Servlet.

Los Servlets son un reemplazo efectivo para los scripts CGI. Proporcionan una forma de generar documentos dinámicos que son fáciles de escribir y rápidos en ejecutarse. Los Servlets también solucionan el problema de hacer la programación del lado del servidor con APIs específicos de la plataforma: están desarrollados con el API Java Servlet, una extensión estándar de Java. Por eso, se utilizan los servlets para manejar peticiones de cliente HTTP. Por ejemplo, tener un servlet procesando datos POSTeados sobre HTTP, utilizando un formulario HTML, que incluye datos del pedido o de la tarjeta de crédito. Un servlet como este podría ser parte de un

sistema de procesamiento de pedidos, trabajando con bases de datos de productos e inventarios, y quizás un sistema de pago "on-line".

Los ServLets permiten la colaboración entre la gente. Un Servlet puede manejar múltiples peticiones concurrentes, y puede sincronizarlas. Esto permite a los Servlets soportar sistemas como conferencias "on-line".

Los Servlets pueden reenviar peticiones a otros servidores y servlets. De esta manera, pueden ser utilizados para cargar balances desde varios servidores que reflejan el mismo contenido, y para dividir un único servicio lógico en varios servidores, de acuerdo con los tipos de tareas o la organización compartida.

2.6 Páginas con código ejecutado en el cliente

- Páginas con código ejecutado en el cliente
 - ActiveX de Microsoft
 - Java-Applets

El código de programación es ejecutado en el cliente. Esto lleva a la posibilidad de conectar con la base de datos, sin tener que pasar por el servidor de WWW. La ventaja es una mayor rapidez; la desventaja es la seguridad, ya que el usuario no sabe qué programa se va a ejecutar en su PC.

Existen dos posibilidades de realizar páginas HTML con código ejecutado en el cliente: ActiveX y Java-Applets. Un requisito es que el visor soporte Java o ActiveX.

2.6.1 ActiveX de Microsoft

Los controles ActiveX forman parte de toda una familia tecnológica, y pueden ser definidos como objetos "vivos", que pueden ser insertados tanto en páginas "Web", como en Visual Basic, Visual C++, Visual FoxPro, Microsoft Access, o cualquier otra aplicación que sea diseñada para contenerlos (ActiveX control container). En el pasado, estos mismos objetos eran referidos como controles OLE y de hecho, a pesar del cambio de terminología, no han dejado de serlo. Forman parte también de otro gran conjunto: el de los objetos OLE. En resumen, un control ActiveX es un control OLE y por lo tanto, es también un objeto OLE.

Todos los controles ActiveX pueden ser insertados tanto en páginas "Web" (integrados con "scripts" Visual Basic o Java) como en aplicaciones Visual Basic, Visual C++, etc, pero también existen controles ActiveX, diseñados para agregar capacidades de navegación y comunicaciones sobre Internet a esas aplicaciones. Esto incluye, a modo de ejemplo, visualizadores html, servicios de correo electrónico, acceso a bases de datos, de transferencia ftp, etcétera, que pueden ser fácil y rápidamente integrados en cualquier aplicación, con esfuerzo mínimo.

Para que una página HTML con código ActiveX que acceda a una base de datos pueda ejecutarse, se requiere que el cliente: **Soporte ActiveX** (para los navegadores de Microsoft desde la versión 3.0 de "Internet Explorer"), y que tenga instalado el controlador ODBC de la base de datos correspondiente, lo cual no resulta difícil en una Intranet, pero si en Internet.

Para ver controles activeX en los navegadores de Netscape, se necesita un "plug-in" que se puede bajar de www.ncompass.com.

A nivel básico, los controles ActiveX se utilizan de la siguiente manera: la etiqueta principal se introduce dentro del código HTML y es `<OBJECT ID="xxx" CLASSID="yyy"><OBJECT>`, siendo **xxx**, el nombre del control ActiveX y **yyy** una larga cadena de caracteres alfanuméricos. Al ejecutar el control, el navegador busca en su librería de controles ya preinstalados (archivos con extensión *.ocx en el directorio SYSTEM) los datos del control ActiveX, pues conoce el nombre y su clasificación. Si lo tiene instalado, lo ejecutará con los parámetros que tenga en las etiquetas `<PARAM NAME="xxx" VALUE="yyy">`; si no lo tiene, intentará buscarlo por la red de una manera automática, y tras eso será descargado a la computadora, desde un servidor de Microsoft o desde la propia página "WEB" donde se está utilizando el control ActiveX, sobre todo si se ha diseñado con un programa como VB 5.0.

Algunos controles tienen una certificación de seguridad llamada "**Authenticode**", que garantiza que los códigos de los controles ActiveX descargados, sean seguros y que no perjudiquen al sistema, el cual sólo posee muy pocos controles y normalmente no los que se crean.

2.6.2 "Java-Applets"

La definición general de un "applet" es una pequeña aplicación accesible en un servidor Internet, que se transporta por la red, se instala automáticamente y se ejecuta **in situ** como parte de un documento "Web". En realidad, un "applet" es una aplicación corta (nada impide que ocupe más de un gigabyte, a no ser el pensamiento de que se va a transportar por la red), basada en un

formato gráfico sin representación independiente, es decir, se trata de un elemento a embeber en otras aplicaciones; es un componente en su sentido estricto.

Dado que los "applets" están mayormente destinados a ejecutarse en navegadores "Web", el HTML debe contener una etiqueta para soportar Java, es decir los "applets". El esquema de etiquetas de HTML, y la evolución del estándar marcado por Netscape, hicieron fácil la adición de una nueva etiqueta que permitiera, una vez añadido el correspondiente código gestor en los navegadores, la ejecución de programas Java en ellos.

La sintaxis de las etiquetas <APPLET> y <PARAM> es la que se muestra a continuación:

Tabla XIV. Sintaxis para incluir un "Applet" de Java en una página HTML

```

<APPLET CODE= WIDTH= HEIGHT= [CODEBASE=] [ALT=] [NAME=] [ALIGN=] [VSPACE=]
[HSPACE=]>
  <PARAM NAME= VALUE= >
</APPLET>

```

Tabla XV. Atributos de la etiqueta "Applet"

"CODE":	Nombre de la clase principal
"WIDTH":	Anchura inicial
"HEIGHT":	Altura inicial
"CODEBASE":	URL base del "applet"
ALT :	Texto alternativo
"NAME":	Nombre de la instancia
"ALIGN":	Justificación del applet
VSPACE :	Espaciado vertical
HSPACE :	Espaciado horizontal

Los "applets" se incluyen en las páginas "Web" a través de la marca <APPLET>, que es muy similar a la marca . Ambas necesitan la referencia a un fichero fuente que no forma parte de la página en que se encuentran embebidos. IMG hace esto a través de SRC=parámetro y APPLET lo hace a través CODE=parámetro. El parámetro de CODE indica al navegador donde se encuentra el fichero con el código Java compilado (.class). Es una localización relativa al documento fuente.

Al igual que IMG, APPLET tiene una serie de parámetros que lo posicionan en la página. WIDTH y HEIGHT especifican el tamaño del rectángulo que contendrá al applet, se indican en "pixels". ALIGN funciona igual que con IMG (en los navegadores que lo soportan), y define cómo se posiciona el rectángulo del "applet" respecto a los otros elementos de la página. Los valores posibles a especificar son: "LEFT, RIGHT, TOP, TEXTTOP, MIDDLE, ABSMIDDLE, BASELINE, BOTTOM y ABSBOTTOM". Y, finalmente, lo mismo que con IMG, se puede especificar un HSPACE y un VSPACE en pixels, para indicar la cantidad de espacio vacío que habrá de separación entre el "applet" y el texto que le rodea.

APPLET tiene una marca ALT. La utiliza un navegador que entienda la marca APPLET, pero que por alguna razón, no pueda ejecutarlo. Por ejemplo, si un "applet" necesita escribir en el disco duro de la computadora, pero en las características de seguridad se tiene bloqueada esa posibilidad, entonces el navegador presentaría el texto asociado a ALT.

ALT no es utilizado por los navegadores que no entienden la marca APPLET; por eso se ha definido la marca </APPLET>, que finaliza la descripción del "applet". Un navegador con soporte Java ignorará todo el texto que haya entre las dos marcas <APPLET> y </APPLET>, sin embargo, un navegador que no soporte Java, ignorará las marcas y presentará el texto que se encuentre entre ellas.

BIBLIOGRAFÍA

1. AFERGAN, Michael. **Programación en Web**. México: Editorial Prentice Hall S.A., 1997.
2. BROWN, Incola. **Diseño de animación en Web**. México: Prentice Hall S.A. 1,997.
3. CABALLERO, José. **Redes de banda ancha**. España: Alfaomega grupo editores S.A. 1,998
4. CBA. "Qué es Internet". <http://www.cba.com.ar/intro/que-es.html>. 1,999.
5. FIGUEROA, Pablo. **Qué es un Java Bean** <http://pegasus.uniandes.edu.co/~pfigueroa/CursoJava/JavaBeans/contjavabean.html>. 1,998.
6. GRALLA, Preston. **Cómo funciona Internet**. USA: acmilla Computer Publishing. 1,996.
7. IBÁÑE, Alvaro. "Qué es Internet" <http://bbs.seker.es/~alvy/queesinternet.html> 1,996
8. IEEE. "The Institute of electrical and electronics engineers" <http://www.ieee.org.mx/> 1,997
9. INTERNET SOCIETY. "Sociedad Internet". <http://www.isoc.org/>. 1,999.
10. MCLAUGHLIN, Brenda. **CGI Bible**. United States Of America. IDG Books WorldWide, Inc. 1,996.
11. MICROSOFT. **Info: mdac, da sdk, odbc, ole db, ado, rds y ado/md**. <http://www.microsoft.com/intlkb/SPAIN/E190/4/63.ASP>. 09/03/1999.
12. NAUGHTON, Patrick. **Manual de Java**. España: McGraw-Hill S.A. 1,996.

13. ONINET. **"Introducción a Internet"** <http://www.oninet.es/cursointernet/arpanettcp.htm>
1,998
14. TENENBAUM, Andrew. **Redes de Computadoras**. 3ª. ed., México: Editorial Prentice Hall Hispanoamericana, S.A., 1997.
15. UNIVERSIDAD DE LAS AMERICAS, **Un Marco de Comunicación Inter-Agentes en una Biblioteca Digital**. <http://ict.pue.udlap.mx/people/adolfo/doctesis.html#2.5.1>. 1,998.

APÉNDICE A

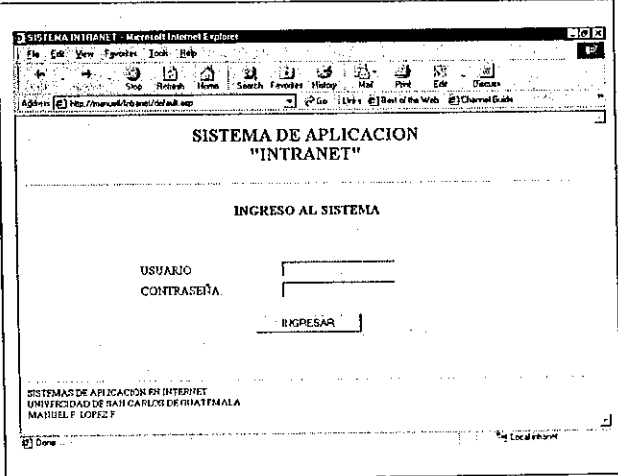
CÓDIGO FUENTE DE LAS PÁGINAS DEL SISTEMA

1. PÁGINA PRINCIPAL DEL SISTEMA INTRANET

Nombre: default.asp

Funcionalidad:

Esta página es llamada default.asp y contiene un formulario el cual es el encargado de enviar el nombre del usuario y la contraseña, a una página que verifica los datos en la tabla. Además, contiene un botón tipo submit, el cual ejecuta el envío de la información.



CÓDIGO DE LA PÁGINA

```
// Define que el lenguaje script a utilizar en el lado del servidor es VBScript
<%@ Language=VBScript %>
//Encabezado de la página
<HTML>
<HEAD>
  <TITLE>SISTEMA INTRANET</TITLE>
</HEAD>

//Cuerpo de la página
<BODY background="">

//Identificación de la página
<P align=center>
  <FONT size=5> <STRONG>SISTEMA DE APLICACION <BR> "INTRANET"</STRONG></FONT>
</P>
<HR>
<P align=center>
  <FONT size=4> <STRONG>INGRESO AL SISTEMA</STRONG></FONT>
</P>

<br><br>

//Formulario encargado de recibir y enviar los datos del usuario a la página verifica.asp
<FORM action="verifica.asp" id=FORM1 method=post name=FORM1>
<TABLE align=center border=0 style="WIDTH: 60%" width="60%">
<TR>
  <TD> USUARIO:</TD>
  <TD><INPUT type=text name="usuario"></TD>
</TR>
<TR>
```

```

<TD>      CONTRASEÑA:</TD>
<TD>      <INPUT type=password name="contras" ></TD>
</TR>
</TABLE>

<P align=center><INPUT name=submit1 type=submit value=" INGRESAR " ></P>
</FORM>

<br><br><HR>

<FONT size=2>
  SISTEMAS DE APLICACION EN INTERNET<BR>UNIVERSIDAD DE SAN CARLOS DE
  GUATEMALA<BR>MANUEL F. LOPEZ F.
</FONT>
</BODY>
</HTML>

```

<p>2. VERIFICA IDENTIFICACIÓN DEL USUARIO</p> <p>Nombre: verifica.asp</p> <p>Funcionalidad: Esta página es llamada verifica.asp y contiene el código para recibir los datos del formulario de la página default.asp, luego llama al componente COMManejo, y utilizando la función de Usuario, se revisa si existe, y si la contraseña ingresada corresponde a la del usuario. Si los datos son correctos, redirecciona a la página del menú principal, en caso contrario redirecciona a una página que informa del error.</p>	<p>Página interna, no muestra resultados</p>
---	---

CÓDIGO DE LA PÁGINA

```

// Define que el lenguaje script a utilizar en el lado del servidor es VBScript
<%@ Language=VBScript %>

// Da parámetros al objeto response, el cual es el utilizado para la comunicación con el cliente
<%
  Response.Expires=0      //Tiempo en segundos cuando expira el cache
  Response.Buffer=true    //Indica que se maneja el buffer
  on error resume next    //Indica que si existe un error debe continuar
%>
<HTML>
// Encabezado de la página
<HEAD>
</HEAD>
//Cuerpo de la página
<BODY>

<!-- Validación de login -->
<%
// Variables a utilizar
Dim vusuario

```

```

Dim vcontras
Dim ObjUsuario
Dim rs
Dim ObjNumAuto
Dim ObjBitacora
Dim NumAuto

//Solicita los parámetros enviados por el formulario en la página default.asp
vusuario=trim(Request.Form("usuario"))
vcontras=trim(Request.Form("contras"))

//Crea un objeto en el servidor, del tipo del componente registrado en el MTS
Set ObjUsuario = Server.CreateObject("COMManejoUsuarios")

//Llama la función usuario del componente, pasando como parámetro el usuario
Set rs = ObjUsuario.Usuario(vusuario)

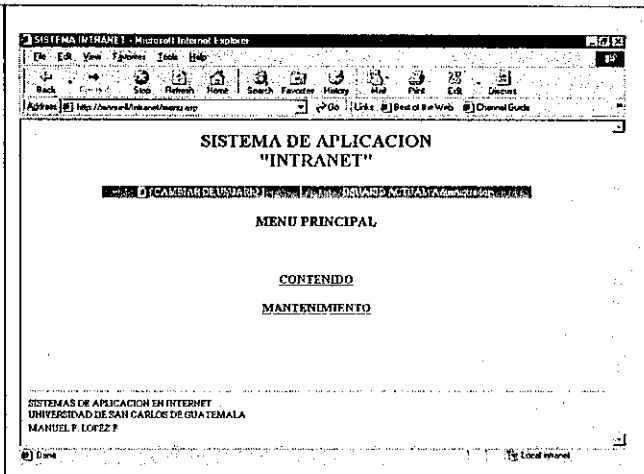
// Elimina el objeto
Set ObjUsuario = Nothing

// Verifica si existió algún error
if err.number<>0 then
    // Si existe algún error informa el número y la descripción y termina la sesión
    Response.Write "<br><br><br><br><br><br><DIV align=center>"
    Response.Write "Error desconocido:" & err.number & "<BR>" & err.description
    Response.Write "<BR>Intentelo de nuevo en unos minutos </DIV>"
    Response.End
else
    // Si no hay error en la ejecución del componente
    // Si la consulta no generó resultados...
    if rs.EOF then
        // No tiene permiso, redirecciona a una página donde se informa el error y termina la sesión
        Response.Redirect "sinacceso.asp"
        Response.End
    Else
        // Si se encontró el usuario, se revisa la contraseña
        if trim(vcontras)=trim(rs.fields("password")) then
            // Si corresponde la contraseña con la ingresada se crean las variables
            // de sesión para la identificación dentro del sistema
            session("usuario")= vusuario
            session("contras")= vcontras
            session("rol")= rs.fields("rol")
            session("acceso")=111

            //Envía a el usuario a la página del menú
            Response.Redirect "menu.asp"
            Response.End
        else
            // La contraseña no corresponde a la del usuario, lo envía a la página de error
            Response.Redirect "sinacceso.asp"
            Response.End
        end if
    end if
    //Elimina la variable del recordset
    set rs=nothing
end if
%>
</BODY>

```

</HTML>

<p>3. MENÚ PRINCIPAL DEL SISTEMA INTRANET</p> <p>Nombre: menu.asp</p> <p>Funcionalidad: Esta página es llamada menu.asp; contiene los links al menú del contenido en sí de la intranet, y al menú de mantenimiento del sistema.</p>	
--	--

CÓDIGO DE LA PÁGINA

```
// Define que el lenguaje script a utilizar en el lado del servidor es VBScript
<%@ Language=VBScript %>
<html>

//Encabezado de la página
<head>
  <title>SISTEMA INTRANET</title>
</head>

//Cuerpo de la página
<body link=#ffffff alink=#ffffff vlink=#ffffff >

//Esta instrucción incluye el código vbscript que verifica los permisos del usuario para acceder esta página
//Si tiene permisos, continua con las siguientes líneas, si no hace el código "else" que se encuentra en este archivo
<!--#include file="chequea.asp"-->

//Identificación y links de la página
<p align="center">
  <font size="5">     <strong>SISTEMA DE APLICACION <br>"INTRANET"</strong></font>
</p>

<table align="center" border="0" cellPadding="0" cellSpacing="2" width="75%">
<tr align="middle" bgcolor="#4169e1">
<td>
  <font size="2" color="#ffffff">
  <A href="default.asp" style="TEXT-DECORATION: none">
  <IMG align=top border=0 height=16 src="images/url.gif" width=13>
  [ CAMBIAR DE USUARIO ] </a></font></td>
</tr>
//Muestra el usuario actual, a través de la variable de sesión usuario
  <font size="2" color="#ffffff">USUARIO ACTUAL: <%=session("usuario")%></font></td>
</tr>
</table>

<p align="center">
```

```

<font size="4"><strong>MENU PRINCIPAL</strong></font>
</p>
<br><br>
<p align="center">

// Link a la página de contenido de la intranet
<A href="contenido.asp">
  <strong><FONT color=mediumbblue>CONTENIDO</FONT></strong></a>
</p>
<p align="center">

// Link a la página del menú del mantenimiento
<A href="mantenimiento/mantenimiento.asp">
  <strong><FONT color=mediumbblue>MANTENIMIENTO</FONT></strong>
  </a>
</p>
<br><br><br><br><br>

<font size="2">

// Información adicional del sistema
SISTEMAS DE APLICACION EN INTERNET<br>UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA<br>MANUEL F. LOPEZ F.</b>
// Esta línea es el final del else de la página incluida al principio
<%end if%>
</body>
</html>

```

<p>4. MENÚ DEL CONTENIDO DEL SISTEMA INTRANET</p> <p>Nombre: contenido.asp</p> <p>Funcionalidad: Esta página es llamada contenido.asp; contiene los links a la información del sistema, que en este caso está bloqueada por la seguridad y es accesible únicamente por los usuarios registrados. Los links de la página son utilizados únicamente como ejemplo.</p> <p>Esta página no contiene nada especial más que el código HTML, y la inclusión del mismo archivo de chequeo del usuario, utilizado por todas las páginas del sistema, por lo que no se muestra el código.</p>	
---	--

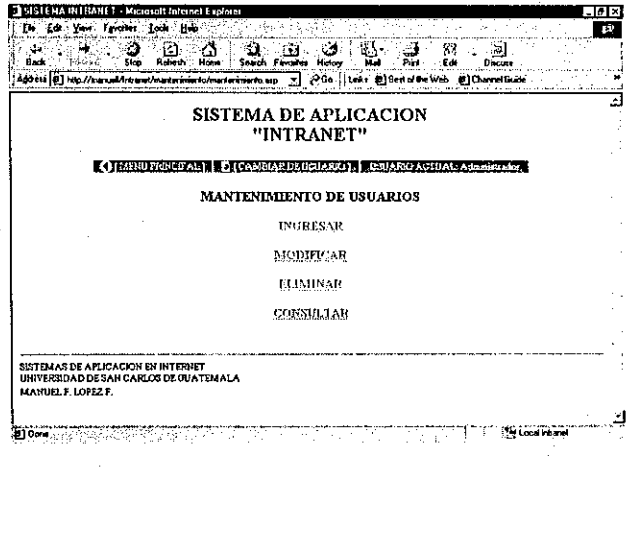
5. MENÚ DE MANTENIMIENTO DEL SISTEMA

Nombre: mantenimiento.asp

Funcionalidad:

Esta página es llamada mantenimiento.asp; contiene los links a las opciones de ingresar, modificar, eliminar y consultar.

Esta página sólo el código HTML y la inclusión del archivo de chequeo del usuario que se encuentra en el directorio de mantenimiento, y que es utilizado por todas las páginas de mantenimiento, este archivo agrega únicamente la verificación que el rol del usuario sea de administrador. Si no tiene ese rol, se manda de regreso a la página del menú del sistema y se informa del error. El código no se muestra por la igualdad con las siguientes páginas.

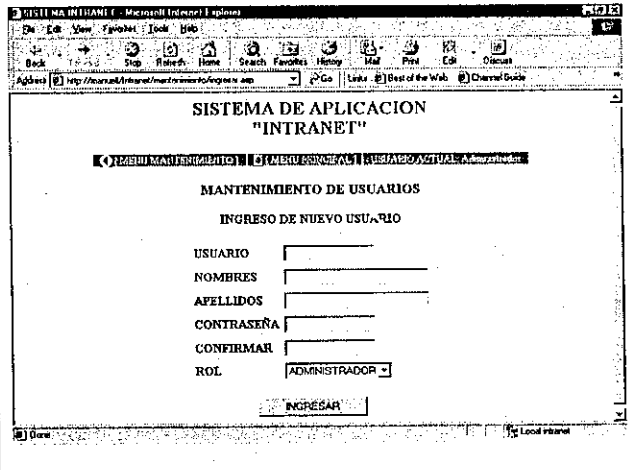


6. INGRESO DE UN NUEVO USUARIO

Nombre: ingreso.asp

Funcionalidad:

Esta página es llamada ingreso.asp; es la encargada de mostrar un formulario con la información necesaria para ingresar un nuevo registro en la tabla de usuario. Los datos solicitados son el usuario, los nombres y apellidos del usuario, una contraseña, luego se vuelve a pedir para poder confirmar que se ingresó correctamente debido a que sólo se muestran asteriscos y para finalizar se debe seleccionar el rol del usuario. Al dar un click sobre el botón de INGRESAR, se llama a la página ingresarr.asp a la cual procesa la solicitud.



CÓDIGO DE LA PÁGINA

// Define que el lenguaje script a utilizar en el lado del servidor es VBScript

```
<%@ Language=VBScript %>
```

```
<html>
```

// Encabezado de la página

```
<head>
```

```
    <title>SISTEMA INTRANET</title>
```

```
</head>
```

//Código script del lado del cliente el cual se encarga de verificar que se ingrese toda la información correctamente.

```
<script language="javascript">
```

//Se ejecuta cuando se cumple con el evento OnSubmit del elemento FORM

```
function Valida(theForm)
```

```
{
```

```

// Las siguientes verificaciones revisan que ningún campo del formulario tenga un valor en blanco
if(theForm.usuario.value=="")
{alert("DEBE INGRESAR UN USUARIO");
theForm.usuario.focus();
return(false);
}
else
{
if(theForm.nombres.value=="")
{alert("DEBE INGRESAR EL NOMBRE DE EL USUARIO");
theForm.nombres.focus();
return(false);
}
else
{
if(theForm.apellidos.value=="")
{alert("DEBE INGRESAR EL APELLIDO DE EL USUARIO");
theForm.apellidos.focus();
return(false);
}
else
{
if(theForm.password1.value=="")
{alert("DEBE INGRESAR LA CONTRASEÑA");
theForm.password1.focus();
return(false);
}
else
{
if(theForm.password2.value=="")
{alert("DEBE INGRESAR LA CONFIRMACION DE CONTRASEÑA");
theForm.password2.focus();
return(false);
} } } } }
// Revisa que la contraseña y la confirmación de la misma sean iguales
if (theForm.password1.value!=theForm.password2.value)
{
alert("LA CONFIRMACION DE CONTRASEÑA ES INCORRECTA");
theForm.password1.value="";
theForm.password2.value="";
theForm.password1.focus();
return(false);
}
}
</script>
//Cuerpo de la página
<body vlink="#fffff" link="#fffff" alink="#fffff">
//Esta instrucción incluye el código vbscript que verifica los permisos del usuario para acceder esta página
//Si tiene permisos continua con las siguientes líneas, si no hace el código "else" que se encuentra en este archivo

<!--#include file="chequea.asp"-->
//Identificación y links de la página
<p align="center">
<font size="5">
<strong>
SISTEMA DE APLICACION <br>&quot;INTRANET&quot;
</strong>
</font>

```

```

</p>

<table align="center" border="0" cellPadding="0" cellSpacing="2" width="75%">
<tr align="center" bgcolor="#4169e1">
<td>
<font size="2" color="#ffffff">
  <a href="mantenimiento.asp" style="TEXT-DECORATION: none">
    <img SRC="..images/r.gif" border="0" align="top" WIDTH="16" HEIGHT="16">
    [ MENU MANTENIMIENTO ]
  </font></a>
</td>

<td>
<font size="2" color="#ffffff">
  <a href="..menu.asp" style="TEXT-DECORATION: none">
    <img SRC="..images/ur.gif" border="0" align="top" WIDTH="13" HEIGHT="16">
    [ MENU PRINCIPAL ]
  </font></a>
</td>
<td>
  <font size="2" color="#ffffff">
    USUARIO ACTUAL: <%=session("usuario")%>
  </font>
</td>
</tr>
</table>

<p align="center">
  <font size="4">
    <strong>
      MANTENIMIENTO DE USUARIOS
    </strong>
  </font>
</p>

<p align="center">
  <strong>
    INGRESO DE NUEVO USUARIO
  </strong>
</p>
//Formulario que recibe toda la información del usuario a ingresar, ejecuta la función Script Valida al enviarse
<form action="ingresarr.asp" method="POST" id="form1" name="form1" onsubmit="return Valida(this)">
<table ALIGN="CENTER" background border="0" cellPadding="2" cellSpacing="3" style="HEIGHT: 187px; WIDTH:
303px" width="75%">
<tr>
<td><strong>USUARIO</strong></td>
<td><input type="text" name="usuario" size="15" maxlength="15"></td></tr>
<tr>
<td><strong>NOMBRES</strong></td>
<td><input type="text" name="nombres" size="25" maxlength="25"></td></tr>
<tr>
<td><strong>APELLIDOS</strong></td>
<td><input type="text" name="apellidos" size="25" maxlength="25"></td></tr>
<tr>
<td><strong>CONTRASEÑA</strong></td>
<td><input id="password1" name="password1" size="15" type="password" maxlength="15"></td></tr>

```

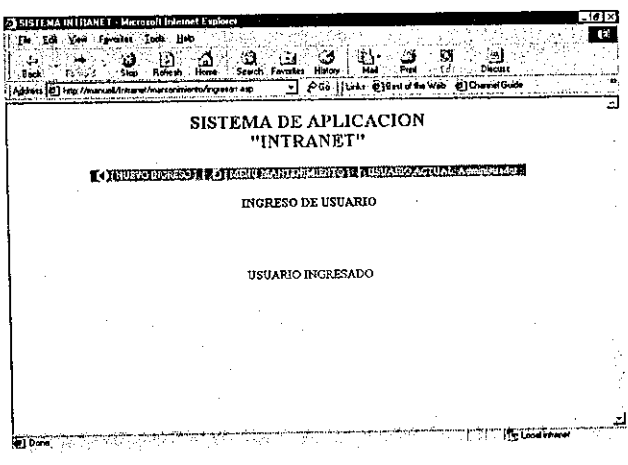


```

<tr>
<td><strong>CONFIRMAR</strong></td>
<td><input id="password2" name="password2" size="15" type="password" maxlength="15"></td></tr>
<tr>
<td><strong>ROL</strong></td>
<td>
<select name="rol">
<option selected value="1">ADMINISTRADOR</option>
<option value="2">VISITANTE</option>
</select>
</td>
</tr>
</table>

<p align="center">
<input name="submit1" type="submit" value=" INGRESAR " >
</p>
</form>
<%end if%>
</body>
</html>

```

<p>7. INGRESO DE UN NUEVO USUARIO</p> <p>Nombre: ingresarr.asp</p> <p>Funcionalidad: Esta página es llamada ingresarr.asp, se encarga de procesar la información enviada para ingresar. Se llama al componente COMMantenimiento y se utiliza la función agregar, enviando como parámetros a la función, los datos recibidos del formulario. Al finalizar la ejecución del componente, se informa del resultado en la página.</p>	
--	---

CÓDIGO DE LA PÁGINA

```

// Define que el lenguaje script a utilizar en el lado del servidor es VBScript, se informa que debe manejar transacciones
<%@ Transaction=required Language=VBScript %>
<%
on error resume next
%>
<html>
//Encabezado de la página
<head>
<title>SISTEMA INTRANET</title>
</head>
//Cuerpo de la página
<body link="#fffff" alink="#fffff" vlink="#fffff">

//Esta instrucción incluye el código vbscript que verifica los permisos del usuario para acceder esta página

```

```
//Si tiene permisos continua con las siguientes líneas, si no hace el código "else" que se encuentra en este archivo
```

```
<!--#include file="chequea.asp"-->
```

```
//Identificación y links de la página
```

```
<div align="center"><font size="5"><strong>SISTEMA DE APLICACION <br>  
&quot;INTRANET&quot;</strong></font><br><br>
```

```
<table align="center" border="0" cellPadding="0" cellSpacing="2" width="75%">
```

```
<tr align="center" bgcolor="#4169e1">
```

```
<td>
```

```
<font size="2" color="ffffff">
```

```
<a href="ingresar.asp" style="TEXT-DECORATION: none">
```

```
<img SRC="..images/r.gif" border="0" align="top" WIDTH="16" HEIGHT="16">  
[ NUEVO INGRESO ]
```

```
</font></a>
```

```
</td>
```

```
<td>
```

```
<font size="2" color="ffffff">
```

```
<a href="mantenimiento.asp" style="TEXT-DECORATION: none">
```

```
<img SRC="..images/url.gif" border="0" align="top" WIDTH="13" HEIGHT="16">  
[ MENU MANTENIMIENTO ]
```

```
</font></a>
```

```
</td>
```

```
<td>
```

```
<font size="2" color="ffffff">
```

```
USUARIO ACTUAL: <%=session("usuario")%>
```

```
</font>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<div align="center"><strong>INGRESO DE USUARIO</strong>
```

```
<%
```

```
Dim ObjUsuario
```

```
Dim rs
```

```
Dim usuario
```

```
Dim nombres
```

```
Dim apellidos
```

```
Dim rol
```

```
Dim contras
```

```
//Asigna a las variables el valor del formulario que llamo la página
```

```
usuario=trim(Request.Form("usuario"))
```

```
nombres=trim(Request.Form("nombres"))
```

```
apellidos=trim(Request.Form("apellidos"))
```

```
contras=trim(request.form("password1"))
```

```
rol=cint(trim(Request.Form("rol")))
```

```
//Crea un objeto en el servidor, del tipo del componente registrado en el MTS
```

```
Set ObjUsuario = Server.CreateObject("COMMantenimiento.Usuarios")
```

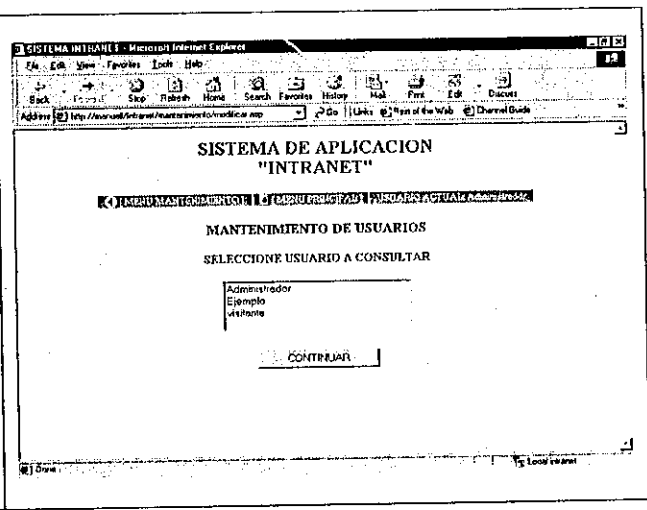
```
//Llama la función agregar del componente, pasando como parámetro los datos el usuario
```

```
rs = ObjUsuario.Agregar(usuario,nombres,apellidos,contras,rol)
```

```

// Elimina el objeto
Set ObjUsuario = Nothing
// Verifica si existió algún error y da el mensaje apropiado
select case err.number
//Si no existió error
case 0
Response.Write "<br><br><br><br><br> USUARIO INGRESADO"
//Si el error es que ya existe el usuario informarlo
case -2147217900
Response.Write "<br><br><br><br><br> ERROR, USUARIO YA EXISTE!"
//Cualquier otro error imprimirlo en pantalla
case else
Response.Write "<br><br><br><br><br> ERROR DESCONOCIDO<BR>"
Response.Write err.number & "<br><BR>" & err.description & "<br><BR>ERROR AL INGRESAR"
end select
%>
</div>
</table>
<%end if%>
</body>
</html>

```

<p>8. MODIFICAR USUARIO</p> <p>Nombre: modificar.asp</p> <p>Funcionalidad: Esta página es llamada modificar.asp; su función es mostrar los usuarios existentes en la tabla de usuarios, para que el usuario Administrador del sistema seleccione el que desea modificar. Este formulario enviará el nombre del usuario a la página modificar.asp.</p>	
---	---

CÓDIGO DE LA PÁGINA

```

// Define que el lenguaje script a utilizar en el lado del servidor es VBScript
<%@ Language=VBScript %>
<html>
// Encabezado de la página
<head>
<title>SISTEMA INTRANET</title>
</head>
//Cuerpo de la página
<body link="#ffffff" alink="#ffffff" vlink="#ffffff">

//Esta instrucción incluye el código vbscript que verifica los permisos del usuario para acceder esta página
//Si tiene permisos continua con las siguientes líneas, si no hace el código "else" que se encuentra en este archivo
<!--#include file="chequea.asp"-->
//Identificación y links de la página
<p align="center">

```

```

<font size="5">
    <strong>
        SISTEMA DE APLICACION <br>&quot;INTRANET&quot;
    </strong>
</font>
</p>

```

```

<table align="center" border="0" cellPadding="0" cellSpacing="2" width="75%">
<tr align="center" bgcolor="#4169e1">
<td>
<font size="2" color="ffffff">
    <a href="mantenimiento.asp" style="TEXT-DECORATION: none">
        <img SRC="..images/r.gif" border="0" align="top" WIDTH="16" HEIGHT="16">
        [ MENU MANTENIMIENTO ]
    </a>
</td>
<td>
<font size="2" color="ffffff">
    <a href="..menu.asp" style="TEXT-DECORATION: none">
        <img SRC="..images/url.gif" border="0" align="top" WIDTH="13" HEIGHT="16">
        [ MENU PRINCIPAL ]
    </a>
</td>
<td>
    <font size="2" color="ffffff">
        USUARIO ACTUAL: <%=session("usuario")%>
    </font>
</td>
</tr>
</table>

```

```

<p align="center">
    <font size="4">
        <strong>
            MANTENIMIENTO DE USUARIOS
        </strong>
    </font>
</p>

```

```

<%
// Variables a utilizar para la utilización del componente
Dim ObjUsuario
Dim rs
Dim todos

//Esta variable define a la consulta que regrese todas las ocurrencias de la tabla
todos=""

//Crea un objeto en el servidor, del tipo del componente registrado en el MTS
Set ObjUsuario = Server.CreateObject("COMMantenimiento.Usuarios")

//Llama la función Usuario del componente, pasando como parámetro %
Set rs = ObjUsuario.Usuario(todos)

// Elimina el objeto

```

```
Set ObjUsuario = Nothing
```

```
%>
```

```
<strong>
```

```
//Formulario que envía el usuario seleccionado por el usuario
```

```
<form action="modificarp.asp" id="FORM1" method="post" name="FORM1">
```

```
<p align="center">
```

```
<strong>
```

```
SELECCIONE&nbsp;USUARIO A CONSULTAR
```

```
</strong>
```

```
</p>
```

```
<p align="center">
```

```
<select id="select1" name="select1" size="2" style="HEIGHT: 74px; WIDTH: 246px">
```

```
<%
```

```
// Este código recorre todo el recordset enviado por el componente y forma un cuadro de selección
```

```
// múltiple con los datos del recordset
```

```
do while not rs.eof
```

```
%>
```

```
// Esta línea se escribirá en la página HTML un número de veces igual a la cantidad de registros del resultado
```

```
// de la consulta cada una de estas líneas crea una línea en el select con los datos de los usuario.
```

```
<option value="<%=rs.fields("usuario")%>"><%=rs.fields("usuario")%> </option>
```

```
<%
```

```
// Esta instrucción del recordset se corre al siguiente registro
```

```
rs.movenext
```

```
loop
```

```
%>
```

```
</select>
```

```
</p>
```

```
<p align="center">
```

```
<input id="submit1" name="submit1" type="submit" value=" CONTINUAR " >
```

```
</p>
```

```
</form>
```

```
</strong>
```

```
<%end if%>
```

```
</body>
```

```
</html>
```

9. MODIFICAR USUARIO

Nombre: modificarp.asp

Funcionalidad:

Esta página es llamada modificarp.asp; esta página recibe el nombre del usuario seleccionado en la página anterior. Realiza una consulta por medio del componente de los datos del usuario y se muestra en un formulario. En estos cuadros de texto, es posible cambiar los datos del usuario. Cuando se modifican los datos se da un click sobre el botón del formulario, para que la página modificarp.asp procese el cambio.

SISTEMA DE APLICACION
"INTRANET"

MANTENIMIENTO DE USUARIOS

MODIFICAR INFORMACION DE USUARIO

USUARIO Ejemplo

NOMBRES Hombres de Ejemplo

APellidos Apellidos de Ejemplo

ROL ADMINISTRADOR

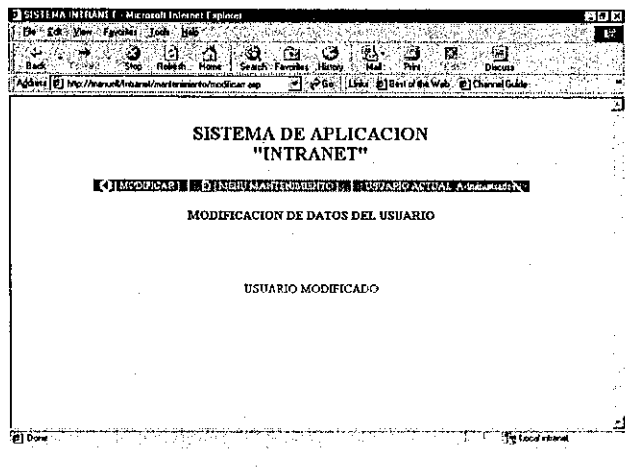
MODIFICAR

10. MODIFICAR USUARIO

Nombre: **modificarr.asp**

Funcionalidad:

Esta página es llamada modificarr.asp y es la encargada de recibir los nuevos datos del usuario y modificarlos en la tabla de usuarios, a través de la función `modificar` del componente `COMManenimiento`. Dependiendo de la salida, se muestra el resultado de la operación.



CÓDIGO DE LA PÁGINA

// Define que el lenguaje script a utilizar en el lado del servidor es VBScript, se informa que debe manejar transacciones

```
<%@ Transaction=required Language=VBScript %>  
<%
```

```
    on error resume next
```

```
%>
```

```
<html>
```

```
//Encabezado de la página
```

```
<head link="#" alink="#" vlink="#">  
    <title>SISTEMA INTRANET</title><br /></HEAD>
```

```
//Cuerpo de la página
```

```
<body link="#" alink="#" vlink="#">
```

```
//Esta instrucción incluye el código vbscript que verifica los permisos del usuario para acceder esta página
```

```
//Si tiene permisos continua con las siguientes líneas, si no hace el código "else" que se encuentra en este archivo  
<!--#include file="chequea.asp"-->
```

```
//Identificación y links de la página
```

```
<p align="center">
```

```
    <font size="5">
```

```
        <strong>
```

```
            SISTEMA DE APLICACION <br>&quot;INTRANET&quot;
```

```
        </strong>
```

```
    </font>
```

```
</p>
```

```
<table align="center" border="0" cellpadding="0" cellspacing="2" width="75%">
```

```
<tr align="center" bgcolor="#4169e1">
```

```
<td>
```

```
<font size="2" color="#">
```

```
    <a href="modificar.asp" style="TEXT-DECORATION: none">
```

```
        <img SRC="..images/r.gif" border="0" align="top" WIDTH="16" HEIGHT="16">
```

```
        [ MODIFICAR ]
```

```
</font></a>
```

```
</td>
```

```
</tr>
```

```

<font size="2" color="#ffffff">
  <a href="mantenimiento.asp" style="TEXT-DECORATION: none">
    <img SRC="..images/url.gif" border="0" align="top" WIDTH="13" HEIGHT="16">
    [ MENU MANTENIMIENTO ]
  </a></font>
</td>
<td>
  <font size="2" color="#ffffff">
    USUARIO ACTUAL: <%=session("usuario")%>
  </font>
</td>
</tr>
</table>

<br>
<div align="center"><strong>MODIFICACION DE DATOS DEL USUARIO</strong>

<%
Dim ObjUsuario
Dim rs
Dim usuario
Dim nombres
Dim apellidos
Dim rol
Dim contras

//Asigna a las variables el valor del formulario que llamo la página
usuario=trim(Request.Form("usuario"))
nombres=trim(Request.Form("nombres"))
apellidos=trim(Request.Form("apellidos"))
rol=cint(trim(Request.Form("rol")))

//Crea un objeto en el servidor, del tipo del componente registrado en el MTS
Set ObjUsuario = Server.CreateObject("COMMantenimiento.Usuarios")

//Llama la función modificar del componente, pasando como parámetro los nuevos datos del usuario
rs = ObjUsuario.Modificar(usuario,nombres,apellidos,rol)

// Elimina el objeto
Set ObjUsuario = Nothing

// Verifica si existió algún error y da el mensaje apropiado
select case err.number
  case 0
    //Si no hay error indicar que fue modificado
    Response.Write "<br><br><br><br><br> USUARIO MODIFICADO"
  case else
    // Informar sobre el error
    Response.Write "<br><br><br><br><br> ERROR DESCONOCIDO<BR>"
    Response.Write err.number & "<br><BR>" & err.description & "<br><BR>ERROR AL MODIFICAR"
end select
%>
</div>
</table>
<%end if%>
</body>
</html>

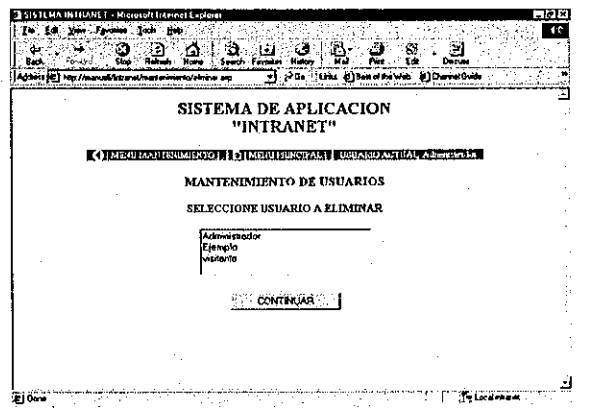
```

11. ELIMINAR USUARIO

Nombre: eliminar.asp

Funcionalidad:

Esta página es llamada eliminar.asp; es la encargada de mostrar al usuario administrador los usuarios del sistema, para que pueda seleccionar el usuario a eliminar. Esta página es muy similar a la página modificar.asp, por lo que no se muestra el código.

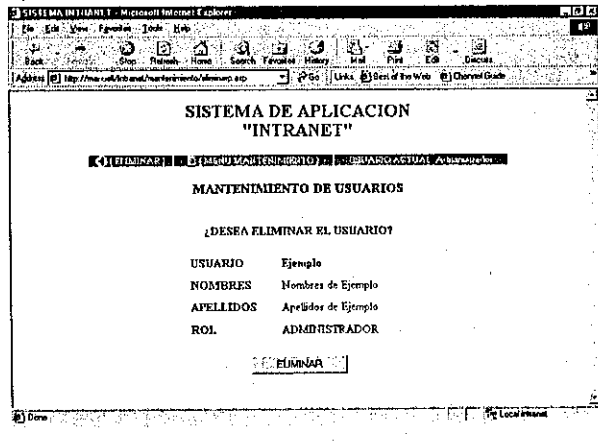


12. ELIMINAR USUARIO

Nombre: eliminarp.asp

Funcionalidad:

Esta página es llamada eliminarp.asp, y es la encargada de recibir la confirmación que se desea eliminar al usuario. En ésta se muestran los datos del usuario a eliminar; dichos datos son obtenidos a través de la función usuario del componente. Al dar un click sobre eliminar, se llama a la página eliminarr.asp la cual procesa la eliminación de la tabla.

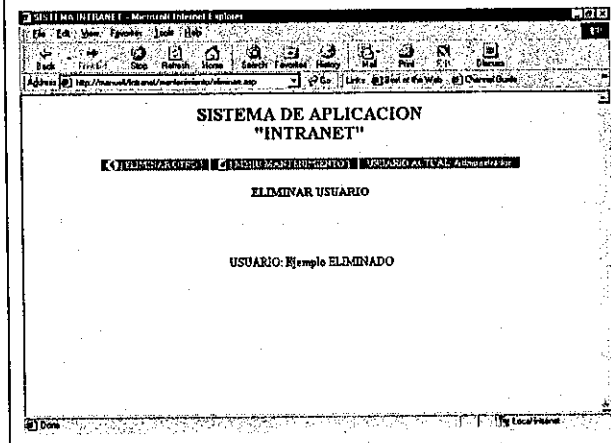


13. ELIMINAR USUARIO

Nombre: eliminarr.asp

Funcionalidad:

Esta página es llamada eliminarr.asp. Recibe el usuario que se desea eliminar, llama a la función eliminar del componente COM Mantenimiento e informa el resultado de la operación.



CÓDIGO DE LA PÁGINA

// Define que el lenguaje script a utilizar en el lado del servidor es VBScript, se informa que debe manejar transacciones

```
<%@ Transaction=required Language=VBScript %>
```

```
<%
```

```
on error resume next
```

```
%>
```

```
<html>
```

```
//Encabezado de la página
```

```
<head>
```

```
<title>SISTEMA INTRANET</title>
```

```
</head>
```

```
//Cuerpo de la página
```

```
<body link="#ffffff" alink="#ffffff" vlink="#ffffff">
```

```
//Esta instrucción incluye el código vbscript que verifica los permisos del usuario para acceder esta página
```

```
//Si tiene permisos continua con las siguientes líneas, si no hace el código "else" que se encuentra en este archivo
```

```
<!--#include file="chequea.asp"-->
```

```
//Identificación y links de la página
```

```
<p align="center">
```

```
<font size="5">
```

```
<strong>
```

```
SISTEMA DE APLICACION <br>&quot;INTRANET&quot;
```

```
</strong>
```

```
</font>
```

```
</p>
```

```
<table align="center" border="0" cellPadding="0" cellSpacing="2" width="75%">
```

```
<tr align="center" bgcolor="#4169e1">
```

```
<td>
```

```
<font size="2" color="#ffffff">
```

```
<a href="eliminar.asp" style="TEXT-DECORATION: none">
```

```
<img SRC="..images/r.gif" border="0" align="top" WIDTH="16" HEIGHT="16">  
[ ELIMINAR OTRO ]
```

```
</font></a>
```

```
</td>
```

```
<td>
```

```
<font size="2" color="#ffffff">
```

```
<a href="mantenimiento.asp" style="TEXT-DECORATION: none">
```

```
<img SRC="..images/url.gif" border="0" align="top" WIDTH="13" HEIGHT="16">  
[ MENU MANTENIMIENTO ]
```

```
</font></a>
```

```
</td>
```

```
<td>
```

```
<font size="2" color="#ffffff">
```

```
USUARIO ACTUAL: <%=session("usuario")%>
```

```
</font>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<div align="center"><strong>ELIMINAR USUARIO</strong>
```

```
<%
```

```
Dim ObjUsuario
```

```
Dim rs
Dim usuario
```

```
// Obtiene el usuario seleccionado en la página eliminar.asp y lo asigna a la variable
usuario=trim(Request.Form("usuario"))
```

```
//Crea un objeto en el servidor, del tipo del componente registrado en el MTS
Set ObjUsuario = Server.CreateObject("COMMantenimiento.Usuarios")
```

```
//Llama la función eliminar del componente, pasando como parámetro el usuario
rs = ObjUsuario.Eliminar(usuario)
```

```
// Elimina el objeto
Set ObjUsuario = Nothing
```

```
// Verifica si existió algún error y da el mensaje apropiado
```

```
select case err.number
```

```
case 0
```

```
//Si no existió error
```

```
Response.Write "<br><br><br><br><br> USUARIO:<strong> " & usuario & "</strong> ELIMINADO"
```

```
case else
```

```
//Cualquier otro error imprimirlo en pantalla
```

```
Response.Write "<br><br><br><br><br> ERROR DESCONOCIDO<BR>"
```

```
Response.Write err.number & "<br><BR>" & err.description & "<br><BR>ERROR AL INGRESAR"
```

```
end select
```

```
%>
```

```
</div>
```

```
</table>
```

```
<%end if%>
```

```
</body>
```

```
</html>
```

14. CONSULTAR USUARIOS

Nombre: consultar.asp

Funcionalidad:

Esta página es llamada consultar.asp; su función es mostrar todos los usuarios y su información. Para esto, se utiliza la función usuario del componente COMMantenimiento, pasando como parámetro el símbolo de porcentaje (%), lo cual le indica a SQL que debe seleccionar a todos los usuarios de la tabla.

USUARIO	NOMBRES	APELLIDOS	ROL
Administrador	Nombres Administrador	Apellidos Administrador	ADMINISTRADOR
Visitante	Nombres Visitante	Apellidos Visitante	VISITANTE

CÓDIGO DE LA PÁGINA

```
// Define que el lenguaje script a utilizar en el lado del servidor es VBScript, se informa que debe manejar transacciones
```

```
<%@ Language=VBScript %>
```

```
<html>
```

```
//Encabezado de la página
```

```

<head>
  <title>SISTEMA INTRANET</title>
</head>

//Cuerpo de la página
<body link="#ffffff" alink="#ffffff" vlink="#ffffff">

//Esta instrucción incluye el código vbscript que verifica los permisos del usuario para acceder esta página
//Si tiene permisos continua con las siguientes líneas, si no hace el código "else" que se encuentra en este archivo
<!--#include file="chequea.asp"-->

//Identificación y links de la página
<p align="center">
  <font size="4">
    <strong>
      SISTEMA DE APLICACION <br>&quot;INTRANET&quot;;
    </strong>
  </font>
</p>

<table align="center" border="0" cellPadding="0" cellSpacing="2" width="75%">
<tr align="center" bgcolor="#4169e1">
<td>
  <font size="2" color="#ffffff">
    <a href="mantenimiento.asp" style="TEXT-DECORATION: none">
      <img SRC="..images/r.gif" border="0" align="top" WIDTH="16" HEIGHT="16">
      [ MENU MANTENIMIENTO ]
    </a>
  </font>
</td>
<td>
  <font size="2" color="#ffffff">
    <a href="..menu.asp" style="TEXT-DECORATION: none">
      <img SRC="..images/url.gif" border="0" align="top" WIDTH="13" HEIGHT="16">
      [ MENU PRINCIPAL ]
    </a>
  </font>
</td>
<td>
  <font size="2" color="#ffffff">
    USUARIO ACTUAL: <%=session("usuario")%>
  </font>
</td>
</tr>
</table>

<p align="center">LISTADO DE USUARIOS DEL SISTEMA</strong>
<%
Dim ObjUsuario
Dim rs
Dim todos

//Le indica a SQL que seleccione todos los usuarios
todos=""

//Crea un objeto en el servidor, del tipo del componente registrado en el MTS
Set ObjUsuario = Server.CreateObject("COMMantenimiento.Usuarios").

```

```

//Llama la función agregar del componente, pasando como parámetro los datos el usuario
set rs = ObjUsuario.Usuario(todos)

// Elimina el objeto
Set ObjUsuario = Nothing
%>
</p>
<table ALIGN="center" WIDTH="75%" BORDER="1" CELSPACING="1" CELLPADDING="1">
<tr>
<td><strong>USUARIO</strong></td>
<td><strong>NOMBRES</strong></td>
<td><strong>APELLIDOS</strong></td>
<td><strong>ROL</strong></td>
</tr>

<%
//Aqui se define el ciclo para recorrer todo el recordset que tiene los datos de los usuario
do while not rs.eof
%>
<tr>
//Las siguientes líneas muestran los datos del registro actual en el recordset, a través de una tabla
<td><%=rs.fields("usuario")%></td>
<td><%=rs.fields("nombres")%></td>
<td><%=rs.fields("apellidos")%></td>
<td><%
//En base al número de rol se muestra si es administrador o visitante
select case rs.fields("rol")
case 1
Response.Write "ADMINISTRADOR"
case 2
Response.Write "VISITANTE"
end select
%>
</td>
</tr>

<%
rs.movenext
loop
%>
</table>
<%end if%>
</body>
</html>

```

<p>15. REVISAR ACCESO A PÁGINAS</p> <p>Nombre: chequea.asp</p> <p>Funcionalidad: Esta página es llamada chequea.asp; su función es incluir un código, que unido con la página que la llama forma la siguiente estructura:</p> <p>Si acceso<>111 o Rol<>Administrador entonces Mostrar el contenido de la página que lo llama Si no tiene permiso Mostrar página de error Fin</p> <p>Esta estructura es formada por la unión de las páginas con esta página especial.</p>	<p>Página interna, no muestra resultados</p>
--	--

CÓDIGO DE LA PÁGINA

```
//Básicamente la página es código VBScript
<%
//Si NO tiene acceso o el rol es distinto del administrador, esto en el caso de las páginas de mantenimiento
If session("acceso")<>111 or session("rol")<>1 Then
  // Muestra el error y lo envía a la página del menú
  Response.Write "<HTML>"
  Response.Write "<HEAD>"
  Response.Write "<META HTTP-EQUIV=REFRESH CONTENT=""2;URL="
    Response.Write ("http://servidor/intranet/menu.asp")
  Response.Write """">"
  Response.Write "</HEAD>"
  Response.Write "<DIV align=center><STRONG><FONT size=4>"
  Response.Write "<BR><BR><BR><BR><BR><BR><BR><BR>"
  Response.Write "NECESITA PERMISO DE ADMINISTRADOR"
  Response.Write "<BR>"
  Response.Write "<BR>"
  Response.Write "ACCESO DENEGADO"
  Response.Write "</FONT></STRONG><BR><BR></DIV>"
  Response.Write "</BODY>"
  Response.Write "</HTML>"
// Este else queda abierto, ya que el else será el contenido de la página que lo llama,
// el fin del if queda dentro de las página que lo llama
Else
%>
```


APÉNDICE B

CÓDIGOS DE DOMINIOS POR PAÍS

<ftp://ftp.ripe.net>

SERVIDORES DE WEB

<http://webcompare.internet.com/cgi-bin/servdir.pl>

NAVEGADORES

Internet Explorer 3.0	http://www.winmag.com/tips/webbrows3.htm
Internet Explorer 4.0.1	http://www.winmag.com/library/1998/0602/sw_year.htm
Internet Explorer 4.0.2	http://www.winmag.com/tips/webbrows5.htm
Internet Explorer 4.0.3	http://www.winmag.com/tips/webbrows8.htm
Internet Explorer 5.0	http://www.winmag.com/tips/webbrows6.htm

LENGUAJES DE PROGRAMACIÓN PARA EL WEB

ASP	http://www.activeserverpages.com
HTML	http://www.geocities.com/Athens/academy/2042/HTML2.html
	http://www.geocities.com/Athens/olympus/7428/primer_1.html
	http://www.pass.es/intro/c_html.htm
	http://www.programacion.net/html/curso/introduccion.html

TECNOLOGÍAS PARA LA IMPLEMENTACIÓN JAVABEANS

Herramientas JavaBeans	java.sun.com/beans/tools.html
Instant Basic for Java	www.halcyonsoft.com/screeshots.html
JavaBeans (JavaSoft)	java.sun.com/beans
JavaBeans Tutorial	java.sun.com/beans/docs
JDK 1.1	java.sun.com/products/jdk/1.1
Tips & Tricks (JavaShareware)	www.javashareware.com/TNT/TNTBeans.html

IDC

<http://www.clari.net.au/~ananda/htx.htm>
<http://home.chattanooga.net/~japanese/testing/page0002.htm>
<http://home.chattanooga.net/~japanese/testing/page0004.htm>
<http://confucius.hhs.se/material/idxsrv/queryhtx.htm>
<http://home.chattanooga.net/~japanese/testing/Default.htm>
<http://home.chattanooga.net/~japanese/testing/page0006.htm#Finding>
<http://www.acer.edu.au/Database%20Examples/page0005.htm>
http://support.microsoft.com/support/frontpage/idc/default.asp#W_HXT
<http://home.chattanooga.net/~japanese/testing/page0008.htm>
<http://home.chattanooga.net/~japanese/testing/page0006.htm>
<http://members.xoom.com/paquinasicd>

HERRAMIENTAS PARA LA IMPLEMENTACIÓN

Adobe PageMill 3.0	www.adobe.com
Atrax the Web Publisher	http://www.winwareinc.com
Drumbeat 2000	http://www.macromedia.com/software/drumbeat/
HomeSite	http://www.allaire.com
HOTDOG	http://www.sausage.com
HoTMetaL	http://www.hotmetalpro.com/products/hm6_eval_signup.htm
Macromedia Dreamweaver	http://www.macromedia.com
Microsoft Front Page 98	http://www.microsoft.com/frontpage/
NetObjects Fusion	http://www.netobjects.com/products/html/nf4.html
QuickSite	http://www.sitetechnology.com
Symantec's Visual Page	http://www.symantec.com/domain/cafe/vpage/
WebEdit PRO	http://www.luckman.com
CoffeeCup HTML Editor++	http://www.coffeecup.com/products.html
Web Media Publisher	http://www.wbmedia.com/publisher/index.shtml
Namo Web Editor 3.0	http://www.namo.com/webeditor/index.htm

MOTORES DE BÚSQUEDA

PRINCIPALES MOTORES DE BÚSQUEDA

<http://www.buscopio.com>
<http://www.yahoo.com>
<http://www.altavista.com>
<http://www.hotbot.com>
<http://www.lycos.com>
<http://www.excite.com>

OTROS MOTORES DE BUSQUEDA

<http://www.infoseek.com>
<http://index.opentext.net>
<http://www.webcrawler.com>

<http://www.mckinley.com>
<http://goto.com>
<http://galaxy.einet.net/galaxy/>
<http://aleph.ac.upc.es/aliweb/>

MOTORES DE BUSQUEDA EN ESPAÑOL

<http://www.elcano.com>
<http://www.telepolis.com>
<http://www.ozu.es>
<http://www.fantastico.com>
<http://www3.uniovi.es/Vicest/MIBI>
<http://biwe.cesat.es>
<http://mexico.web.com.mx/>
<http://www.tarantula.com.mx>
<http://donde.uji.es>
<http://www.dirglobal.net>
<http://www.mexmaster.com>
<http://www.eliana.com>
<http://trovator.combios.es>
<http://www.yupi.com>
<http://www.vindio.com>
<http://www.ole.es>
<http://espanol.yahoo.com>