



TS
005.74
V393b

F07073

TS
005.74
V393b

F07073



BIBLIOTECA

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA



UNIVERSIDAD AUTONOMA DE QUERETARO

Facultad de Informática

Opción de Titulación: "Curso de Actualización".

Curso: "Bases de Datos en Internet".

Asesor de tesina
I.S.C. Pablo Gutierrez Lara

Alumno
Jose Luis Vázquez Angeles.

Generación 1996 – 2001

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

No. Adq. F07073
Clasif. TS605.74
Cutter V3936

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA



CARTA DE ACEPTACIÓN

Por este medio, se otorga constancia de aceptación de tesina para obtener el título de Licenciado en Informática, que presenta el pasante **JOSÉ LUIS VAZQUEZ ANGELES** con el tema denominado “Bases de datos en Internet”.

Este trabajo fue desarrollado como una investigación derivada del curso de titulación “**ADMINISTRACIÓN DE REDES CON WINDOWS NT**”, dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los catorce días del mes de mayo del dos mil tres.

ATENTAMENTE


I.S.C. PABLO GUTIÉRREZ LARA
PROFR. CURSO DE TITULACIÓN



AGRADECIMIENTOS

A Mis Padres, que en todo momento y lugar fueron un apoyo indispensable durante esta tan difícil etapa en mi vida.

A Mis Abuelitos, que siempre estuvieron ahí, cuando más los necesite.

A Mis Amigos, Compañeros de Clase y Maestros, que con regaños, alientos y buenos deseos me incentivaron a salir adelante.

INDICE

	Pag.
1.- Introducción.....	1
2.- La World Wide Web, una historia reciente.....	2
2.1 ¿Qué es Internet?.....	2
2.2 Un poco de historia... ..	3
2.3 Algunos datos estadísticos... ..	5
3.- ¿Qué se necesita para navegar Internet?.....	7
3.1 La importancia de Internet.....	7
3.2 Internet y Organizaciones Involucradas.....	8
3.3 Internet y los protocolos TCP/IP.....	9
4.- El Modelo Cliente - Servidor.....	11
4.1 Alternativas de procesamiento de Base de Datos Multiusuario.....	14
4.2 Estructura de Internet y direcciones IP.....	20
4.3 Dirección IP.....	21
4.4 Nombres de dominio TCP/IP en Internet.....	22
4.5 Sistema de nombres de dominios (DNS).....	23
4.6 Nuevos dominios.....	24
4.7 Localizadores uniformes de recursos (URL).....	26
5.- Servicios de Internet.....	27
5.1 El servicio de http y la world wide web.....	27
5.2 El servicio de ftp.....	27
5.3 El Servicio de Telnet.....	29
5.4 El servicio de correo electrónico.....	31
5.5 El servicio de IRC.....	31
6.- Bases de Datos Vs. Internet.....	32
6.1 ¿Porqué utilizar bases de datos en la Web?.....	32
6.2 Seguridad.....	33

	Pag.
7.- ¿Qué se necesita para la implimentación de una Base de Datos con acceso a Internet?	34
7.1 Panorama General.....	34
7.2 Los Programas <u>CGI</u> como interfaces para acceder a bases de datos.....	37
7.3 La Interfaz.....	40
7.4 Métodos para transferir información de un navegador web a un programa CGI.....	41
7.5 Interfaces para bases de datos recuperación de información de una base de datos.....	47
7.6 Caso Práctico.....	49
8.- ¿ Que es ASP?.....	50
8.1 Active Server Pages (Páginas Activas de Servidor).....	50
8.2 ¿Por qué las bases de datos?.....	52
8.3 ¿Cómo leer una base de datos mediante ASP?.....	62
8.4 Ejemplos de programas utilizando código ASP....	62
9. ¿El acceso remoto a las bases de datos a través de Internet, beneficiará a alguien?.....	70
Bibliografía.....	73
Páginas Web de Referencia Bibliográfica.....	73

1.- Introducción

La evolución de los sistemas de información y el crecimiento no planeado de la información dentro de las organizaciones, ha traído dispersión de los datos en sitios local o geográficamente dispersos. La necesidad de integrar y compartir dicha información, implica el nacimiento de una nueva tecnología capaz de conformar de manera consistente la información de las organizaciones. Una de las tecnologías que trabaja en el problema de integración de información, es la de Bases de Datos Distribuidas (BDD) a través de la Web.

Los Sistemas de Multibase de Datos, como un tipo especial de BDD, trabajan en el proceso de integración de componentes distribuidos, con el fin de ofrecer un esquema integrado (esquema global); esquema capaz de ofrecer transparencia de acceso a las Bases de Datos Componentes (BDC's) y garantizar la autonomía de operación de cada una de ellas.

Dentro de un mundo "globalizado" la meta para un negocio corporativo es comunicarse con todas sus sucursales, tiendas y almacenes propiedades de la empresa, esto puede lograrlo mediante la combinación de varias bases de datos y la Internet, ya que ambos recursos, proporcionan en "mecanismo" o "herramienta" eficaz de comunicación. De lo contrario, si el negocio no invierte en implementar un mecanismo con el antes descrito, estará condenado al fracaso y no podrá participar dentro de su mercado.

En el presente trabajo se proporcionará un panorama amplio sobre el funcionamiento e implementación de bases de datos en Internet, accediendo a sus datos a través de la Web,.

Otra de las finalidades de este trabajo es destacar la importancia que tienen las bases de datos dentro de las actividades "cotidianas" que desarrollan las empresas. Dicha importancia abarca puntos de vista desde lo económico, operacional, laboral y toma de decisiones hasta los puntos que hablan de seguridad, manejo, almacenamiento de la información que las organizaciones manejan. Ya que en los últimos años sea convertido en un "activo" mas de la empresa.

2.- La World Wide Web, una historia reciente.

2.1 ¿Qué es Internet?

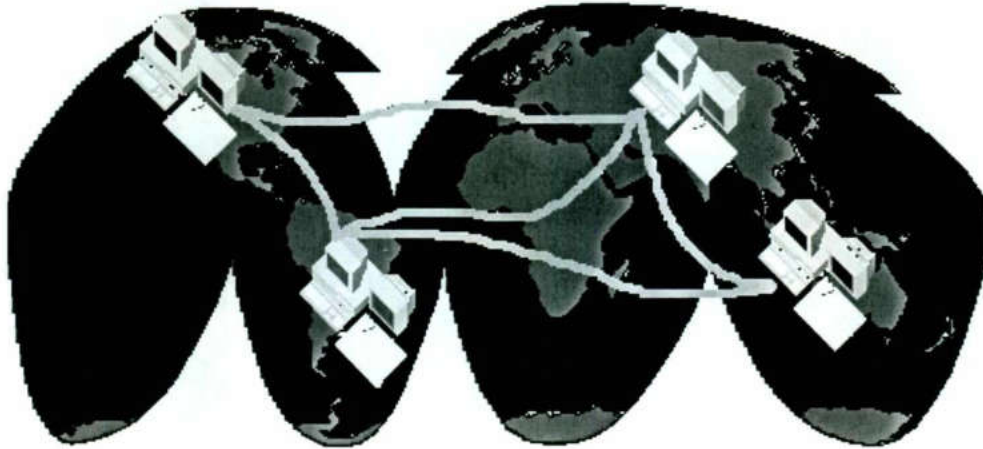


Figura 1. Red de Redes: **Internet**

Las redes que se encuentran interconectadas abarcan desde grandes instalaciones gubernamentales, como la NASA, hasta redes universitarias o mantenidas por organizaciones sin fines de lucro, como la Cruz Roja y corporaciones como IBM, Microsoft, etc. Maestros, estudiantes y cualquier persona en particular utilizan Internet cualquiera sea el fin. Pero a pesar de esta diversidad de personas, empresas e instituciones que la utilizan, nadie es su *dueño*. Simplemente cada usuario paga su conexión para acceder a ella.

Imaginemos a un estudiante que desea investigar sobre la cosecha de arroz en diferentes países. Rastreado a través de la red los sitios donde puede recolectar la información, en pocos minutos obtendrá los datos que de otra manera le hubiera llevado mucho tiempo. Si este mismo estudiante deseara adquirir el último CD de los Rolling Stones que aún no se ha editado en Argentina, simplemente deberá encargarlo por Internet.

La escala de Internet es realmente abrumadora: 40 millones de usuarios, 6 millones de computadoras conectadas... alrededor de 250 Mb de información se agregan a los grupos de noticias cada día... Pero todo esto tuvo un comienzo...

2.2 Un poco de historia...

Corría el año 1969 cuando un grupo de personas da a luz Internet. Nace como un proyecto de la *Agencia De Proyectos De Investigación De Defensa Avanzados de EE.UU.* Surge de la necesidad de intercambiar información entre los investigadores y científicos militares, ubicados en sitios apartados. Esta red, que en principios se suponía debía soportar hasta un ataque nuclear sin perder la conexión, originariamente recibió el nombre de DARPANET y estaba conformada por cuatro computadoras interconectadas.

En 1972 y luego de que comenzara a utilizarse el protocolo NCP y se haya escrito el primer programa de correo electrónico para una red distribuida, ya se habían conectada 37 computadoras a la red que pasó a denominarse ARPANET debido a un cambio en el nombre de la agencia que la administraba. En los años sucesivos el servicio se extendió hasta Inglaterra y Noruega pero como siempre ocurre en estos casos la aplicación más utilizada no fue la que se había definido originariamente (Investigación) sino un servicio subvalorado en un principio: el correo electrónico o E-Mail.

En 1974 se presenta un proyecto de la red ETHERNET en donde se publica el protocolo TCP usado actualmente en Internet y con diversos programas de comunicación. Así se creó una rama comercial de ARPANET denominada TELENET.

En 1976, en los laboratorios de BELL, de AT&T, se desarrolla el UUCP (Unix to Unix Communication Program), usado en sistema de E-Mail por modem y se crea USENET empleando dicho sistema.

En 1982, ARPANET adopta TCP/IP (IP: Internet protocol) y se funda la red europea EUNET.

En 1984 se crea JUNET que es la NET de Japón. Este mismo año, la Fundación Nacional De Ciencias De EE.UU. (NSF) establece la NSFNET (basado en la tecnología de ARPANET), que se origina a partir de la necesidad de compartir la potencia de grandes Super-Computadoras (eran tan grandes y caros que sólo se construyeron cinco centros), instaladas por dicha fundación, con la comunidad académica. En principio se había previsto utilizar ARPANET pero a esta altura ya se encontraba saturada. NSFNET estaba dividida en regiones y cada una de ellas tenía acceso a por lo menos una Super-Computadora, y el resto conectado en forma directa a través de distribuidores.

En 1987 ya había demasiada gente utilizándola, y no solo para la investigación académica, resultando así insuficiente. En el transcurso de este año se redimensionó totalmente incorporando formas de acceso más rápidas gracias al aumento de la velocidad de modems y computadoras, lo que permitió extender su utilización por parte de investigadores universitarios, agencias gubernamentales e incluso a organizaciones internacionales dedicadas a la investigación, con la única condición de pertenecer a países aliados a EE.UU.

En 1989, COMPUSERVE, uno de los mayores sistemas de datos en línea se vincula a la red brindando sus servicios.

Algunos investigadores del Laboratorio Europeo de Física de Partículas (CERN) deseaban desarrollar un método mejorado para que el grupo de investigadores distribuidos alrededor del mundo pudiera compartir información.

Debido a que los investigadores realizaban diferentes actividades (como leer un *paper*, visualizar imágenes, usar correo, etc.), esto implicaba la utilización de varios programas (FTP, Gopher, Archie, etc.), por lo que le objetivo era brindar una única interface, eliminando la necesidad de realizar varios accesos en distintos programas.

Luego de un año, a fines de 1990, del proyecto nace un *Browser* (buscador, navegador) *gráfico* para las computadoras NeXT.

En 1990 la red ARPANET es discontinuada y mediante la caída del Muro de Berlín, la red (o Internet, como se la conoce actualmente) se abre para todo aquel que pudiera conectarse.

En 1991 se crea WAIS (Wide Area Information Service) que facilita el acceso a datos en Internet; junto a este hace su aparición el sistema *Gopher* de menús.

En 1992 se crea la WWW (World Wide Web) siendo recién en 1993, con el uso masivo de esta herramienta, cuando se descubren las fabulosas capacidades de la red.

Luego, en ese mismo año el proyecto se publica para el uso masivo por el CERN. Inicialmente eran solo *aplicaciones hipertextuales* y artículos dentro de noticias, luego se incorporaron otros servicios. El resto lo hicieron cada uno de los "habitantes" de Internet que se percataron que publicar en la WWW era una gran idea y un gran negocio.

Como se mencionó, para acceder a la WWW, es necesario correr en la computadora un *Browser*: Este es una aplicación que "sabe" como interpretar y mostrar los documentos hipertextuales que se encuentran en la WWW.

Estos documentos no contienen solamente texto, sino también comandos que estructuran el contenido en función de cada ítem, permitiendo distintos formatos para los encabezados, párrafos, consultas, gráficos, sonido, videos, etc., según el diseño del proveedor; para lo cual el receptor debe poseer el hardware adecuado para reproducir dichos recursos.

Cuando se accede a un documento de la WWW, el texto visualizado generalmente es de muy buen gusto (Figura 2). Para que esto sea así es necesario aplicar un método que permita explicar formatos e *hiperlinks* en forma sencilla pero poderosa. Este formato de datos o generalmente llamado lenguaje es el HTML.



Figura 2. Ejemplo de página Web

En Argentina se abre el acceso comercial a Internet en 1995, 26 años después de su aparición.

2.3 Algunos datos estadísticos...

Con el correr de los años fue incrementándose notoriamente el uso de la red en el mundo. Como vemos en estas gráficas, entre 1969 y 1983 (figura 3) el total de máquinas conectadas en el mundo era de 500 hacia fines del 83. En 1989 (figura 4) el total de máquinas aumentó notoriamente llegando a 100.000 máquinas conectadas. En 1998 (figura 5) se cuenta con 8 millones de máquinas conectadas (aproximadamente) estimándose un aumento del el doble por año.

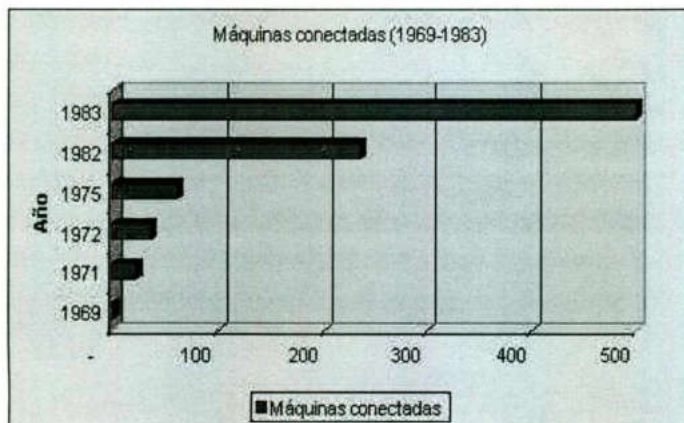


Figura 3

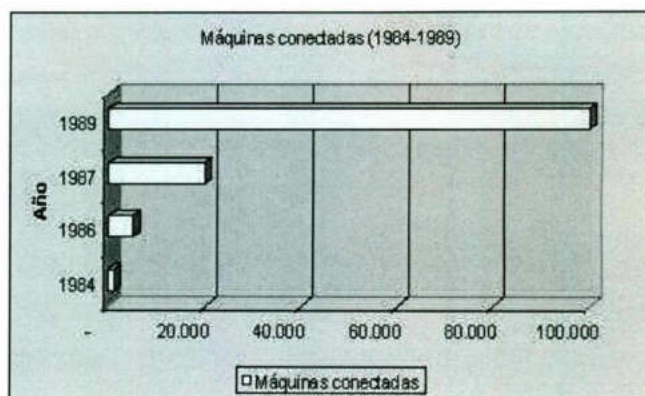


Figura 4

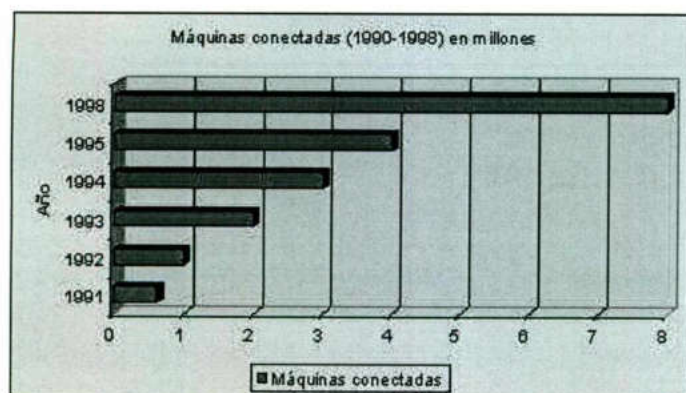


Figura 5

3.- ¿Qué se necesita para navegar Internet?

Hardware: Sobre este punto el mayor problema reside en el vínculo de conexión: ya que lo demás forma parte de un PC estándar de hoy: 486 o superior con 8 (o 16 Mb) de memoria, monitor VGA, disco con capacidad suficiente para contener Windows, el navegador, alguna aplicación y para "bajar" páginas que nos interesen. Si se trata de un usuario *Stand-Alone* se necesita un Módem. En este caso la velocidad es esencial: 28.800 bps o superior es recomendado para una navegación aceptable, aunque si solo se tendrá correo electrónico con uno de 2.400 bps es suficiente (si lo consigue en el mercado).

Software: Es esencial el soft de TCP/IP. Dichos paquetes proveen drivers y aplicaciones necesarias para efectuar la conexión TCP/IP con Internet. Actualmente Windows 95, OS/2, NT, MacOS proveen dichos paquetes. Si no se cuenta con dichos Sistemas Operativos se deberá conseguir programas como PC/TCP, Trumpet, Lan WorkPlace, Chamaleon, etc. Si se trata de un usuario de LAN, deberá verificar que el soft sea compatible con la aplicación de LAN que está utilizando: Odi (Novell), o NDIS (Windows).

Además debe poseer un Browser, que generalmente es provisto por el proveedor del servicio. Los más populares fueron y son el *Mosaic*, *Cello*, *Emissary*, *Netscape Navigator*, *Explorer*, y muchos más que pueden conseguirse gratuitamente en algún sitio de la WEB.

Los Browser utilizan o proveen una biblioteca de windows (*WINSOCK*) la cual provee las herramientas de TCP ya mencionadas.

3.1 La importancia de Internet

El fenómeno Internet ha venido a alterar la forma de vida de millones de personas que hoy en día utilizan la red en diferentes actividades de su vida cotidiana. El correo electrónico es de las aplicaciones más utilizadas y de una gran utilidad para la comunicación entre amigos, investigadores y colegas. Sin embargo la World Wide Web es el servicio al que todo usuario puede acceder, buscar información en bibliotecas virtuales, universidades, centros de investigación o en páginas desarrolladas por personas aficionadas al tema.

Existen otros servicios, como el utilizado para la transferencia de archivos, llamado FTP (File Transfer Protocol). Este está dirigido a quienes necesitan enviar grandes volúmenes de información a lugares distantes y que demandan seguridad y confiabilidad. Otro servicio es el de terminal

virtual (TELNET) con el cual se pueden realizar tareas desde terminales remotas como si físicamente se estuviera trabajando en el equipo al cual se conecta. Existen otros servicios menos utilizados como Gopher (información de tipo texto) y su servicio de búsqueda llamado Veronica.

Sin duda alguna la aparición de la Web, y la implementación de navegadores o browsers con los cuales se podía desplazar de un sitio a otro a través de forma gráfica (iconos) utilizando como interfaz un ratón, dio el puntapié al fenómeno Internet.

3.2 Internet y organizaciones involucradas

Durante el desarrollo del protocolo TCP/IP se incrementó notablemente el número de redes locales de agencias gubernamentales y de universidades, dando origen así a la red de redes más grande del mundo; las funciones militares que se pensaron en un inicio se separaron y se permitió el acceso a la red a todo aquel que lo requiriera sin importar de que país provenía la solicitud siempre y cuando fuera para fines académicos o de investigación.

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 la National Science Foundation quien promovió el uso de la red ya que se encargó de conectar cinco centros de supercómputo en todo Estados Unidos que podían ser accedidos desde cualquier nodo de la red. El grupo de mayor autoridad sobre el desarrollo de la red es la Internet Society, creado en 1992 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 que tiene la responsabilidad de la administración técnica y dirección de Internet, aunque no es el único.

Existen otros tres grupos que tienen un rol significativo, el Internet Architecture Board (IAB), esta organización 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), administrado por el departamento de defensa de los Estados Unidos. En 1989 se reorganiza la estructura de la IAB, un grupo subsidiario e importante de la IAB es el Internet Engineering Task Force (IETF) *Fuerza de Tarea de Ingeniería*, esta se concentra en problemas

de ingeniería a corto y mediano plazo y la IRTF *Fuerza de Tarea de Investigación Internet*, la cual coordina las actividades de investigación relacionadas con los protocolos TCP/IP y con la arquitectura de la red de redes en general.

3.3 Internet y los protocolos TCP/IP

La estandarización tecnológica de Internet dio origen a un nuevo conjunto de protocolos llamados 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, el cual no fue diseñado para ignorar o evadir estándares ya existentes. Estos surgen solamente porque ninguno de los protocolos existentes satisfacía la necesidad de un sistema de comunicación interoperable para el enlace de redes. Los protocolos TCP/IP, cuyo nombre proviene de sus dos protocolos principales fue desarrollado en la agencia de proyectos Avanzados de Investigación.

TCP/IP puede utilizarse para comunicarse a través de cualquier grupo de redes interconectadas. Muchas empresa utilizan TCP/IP para interconectar todas las redes dentro de una corporación, aún cuando las empresas no tengan una conexión hacia redes externas. Otros lo utilizan para comunicarse entre sitios geográficamente alejados unos de otros.

El diseño de los protocolos TCP/IP proporciona una conexión universal entre máquinas independientes a las redes en particular a las que están conectadas, por este motivo TCP/IP realiza las conexiones de forma transparente al usuario sin importar la ruta que la conexión establezca, de esta forma se oculta al usuario la arquitectura de la red global TCP/IP es un conjunto o suite de protocolos diseñados con una arquitectura en capas.

Las capas permiten a los diseñadores del protocolo dividir en módulos las tareas y servicios que realizará el mismo. El diseño también especifica la manera en que un módulo interactúa con los otros. La arquitectura en capas de los protocolos está diseñada como una pila en la que los protocolos de más alto nivel interactúan con los protocolos de niveles más bajos.

El modelo TCP/IP está formado por cuatro capas:

1. La capa de aplicaciones es la capa más alta de la pila, ésta provee servicios de alto nivel a los usuarios como transferencia de archivos, entrega de correo electrónico y acceso a terminales remotas, entre

otros. Los programas de aplicación escogen entre diferentes protocolos de transporte dependiendo del tipo de servicio que requieran.

2. La principal tarea de la capa de transporte es proveer comunicación punto a punto entre las aplicaciones. Los protocolos de transporte (TCP y UDP) usan el servicio de entrega de paquetes que provee la capa de Internet .
3. La capa de Internet provee el servicio de entrega de paquetes de una máquina a otra, por medio del protocolo de Internet (IP). La integridad de los datos no se verifica en este nivel, por lo que el mecanismo de verificación es implementado en capas superiores (Transporte o Aplicación).
4. La capa de acceso al medio acepta datagramas de la capa de Internet y los envía físicamente. El modulo para el acceso al medio es con frecuencia un controlador de dispositivo (device driver) para una pieza particular de hardware, y la capa de acceso al medio puede consistir de múltiples módulos.

Figura. 6 Representación gráfica de las cuatro capas de TCP/IP



Para que la información fluya a través de las capas, esta pasa por un proceso de encapsulamiento. Los mensajes e información recibida por la capa de TCP es encapsulada con un encabezado de TCP en un paquete llamado "Segmento de TCP", este es entregado a la capa de IP, en el que se le agrega un encabezado de IP y se crea el paquete llamado "Datagrama de IP". El paso final incluye el encapsulamiento del datagrama de IP en paquetes creados para la capa de acceso al medio.

4.- El Modelo Cliente - Servidor

Cuando se utiliza un servicio de Internet, se establece un proceso, en el cual entran en juego dos partes. Por un lado, el usuario ejecuta una aplicación en la computadora local, que es el denominado programa *cliente*, este programa se pone en contacto con el host remoto para solicitar la información deseada. El host remoto a su vez, responderá al pedido realizado por el programa cliente, a través de otro programa de aplicación, el programa *servidor*. (figura 7).

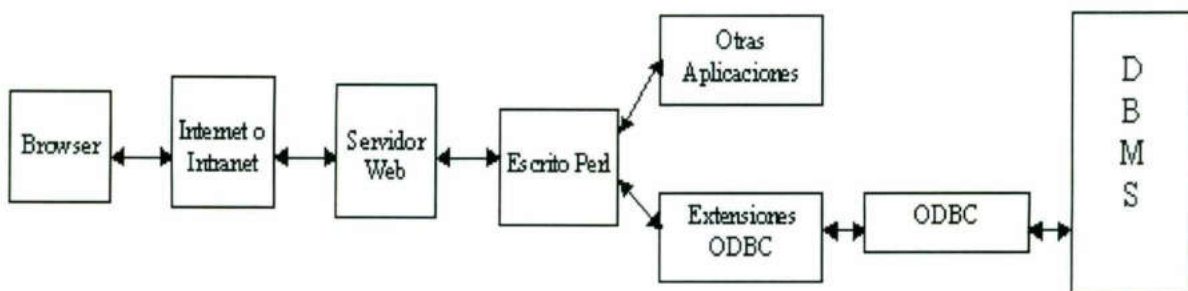


Figura 7. Modelo Cliente - Servidor utilizando Internet o Intranet.

Los términos *cliente* y *servidor* se usan tanto para referirse a los programas que cumplen estas funciones, como a las computadoras donde son ejecutados dichos programas.

El programa cliente cumple dos funciones distintas, por un lado gestiona la comunicación con el servidor, solicita un servicio y recibe los datos enviados por *servidor*; por otro maneja la interfaz con el usuario, presenta los datos en el formato adecuado y brinda las herramientas y comandos para que el usuario pueda utilizar las prestaciones del servidor de forma sencilla.

El programa *servidor*, en cambio, acepta atenciones recibidas a través de la red, realiza el servicio y regresa el resultado al cliente, de esta forma un mismo *servidor* puede atender a varios *clientes* al mismo tiempo, sin embargo el servidor procesa una petición a la vez, después de aceptar una petición, el servidor forma una respuesta y la manda antes de volver a ver si ha llegado otra petición, se asume que el sistema operativo hace una cola de peticiones que llegan al servidor mientras este está ocupado, los servidores tienen dos partes importantes, un programa maestro sencillo,

el cual es responsable de aceptar nuevas peticiones, y un conjunto de esclavos, los cuales son responsables de manejar las peticiones individuales, como el maestro ejecuta un esclavo para cada nueva petición el procesamiento procede de manera concurrente, de este modo las peticiones que requieren de poco tiempo para completarse pueden terminar antes que las peticiones que llevan más tiempo, independientemente del orden en que hayan comenzado. (Tabla no.1)

Funciones del Cliente	Funciones del Servidor
Administrar la interfaz de usuario.	Aceptar las solicitudes de la base de datos de los clientes.
Aceptar datos usuario.	Procesar las solicitudes de la base de datos.
Procesar la lógica de la aplicación.	Dar formato a los resultados y transmitirlos al cliente.
Generar las solicitudes para la base de datos.	Llevar a cabo la verificación de integridad.
Transmitir las solicitudes de la base de datos al servidor.	Mantener los datos generales de la base de datos.
	Proporcionar control de acceso a concurrente.
Recibir los resultados.	Llevar a cabo recuperación

Tabla 1 Funciones de las computadoras cliente y servidor

Sin embargo esto que parece sencillo es aún más complejo, ya que necesita acomodar varias peticiones concurrentes, con esto los servidores deben de reforzar las reglas de autorización y protección, los programas servidor suelen requerir una ejecución de alta prioridad, pues tienen que leer archivos, mantenerse en línea y tener acceso a los datos, cada servidor toma la responsabilidad para reforzar el acceso al sistema y las políticas de protección. (figura 8).

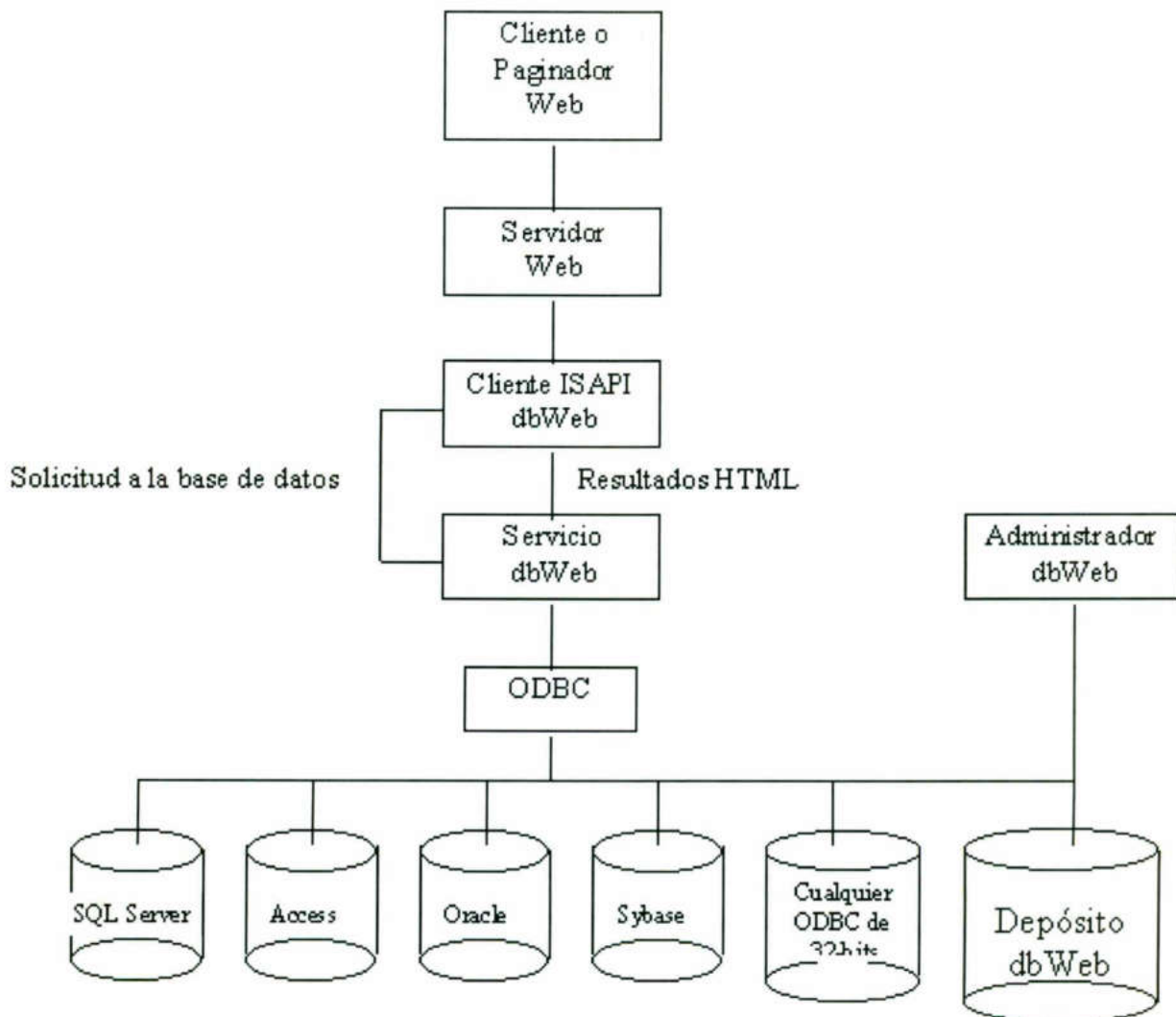


Figura 8. Esquema de solicitudes Cliente - Servidor.

En las aplicaciones cliente-servidor es importante minimizar las interacciones entre un cliente y la información, es por eso que las máquinas se valen del protocolo ARP de TCP/IP para mantener una memoria intermedia (caché) de respuestas para mejorar la eficiencia de las búsquedas redundantes que surjan después. El proceso de memoria intermedia (caching) mejora el desempeño del modelo, un ejemplo real de esta aplicación se da en el caché del browser utilizado para navegar (figura 8), en donde se alojan todos los gráficos y de más archivos de las páginas que han sido visitadas, de forma que si en una sesión se vuelve a visitar el sitio no es necesario que el servidor vuelva a enviar esa información pues esta se tiene en el caché y puede ser recuperada de forma inmediata, el browser es un programa cliente. (figura 8).

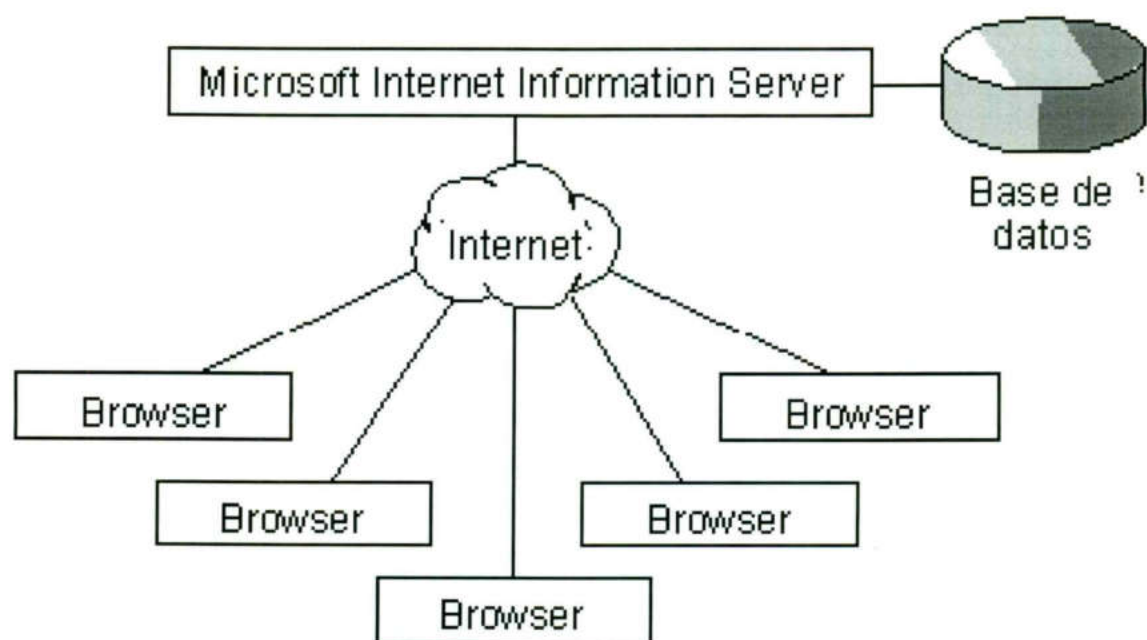


Figura 9. Sistema Browser

4.1 Alternativas de procesamiento de Base de Datos Multiusuario

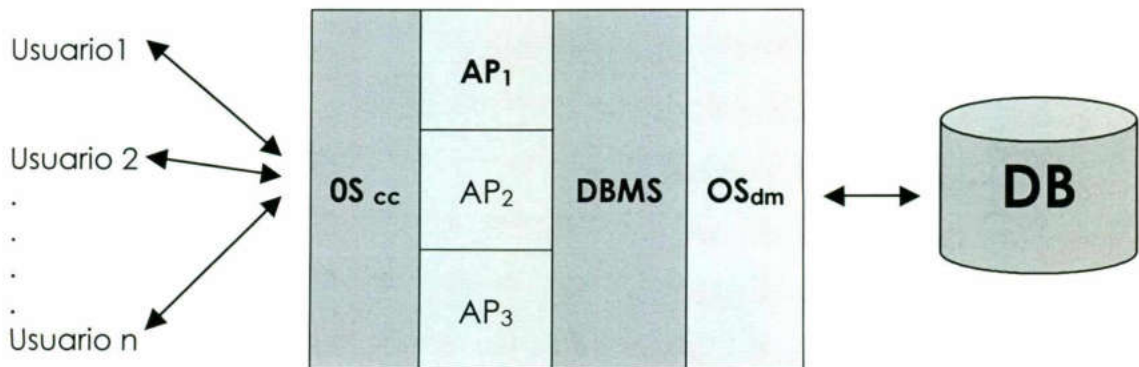
Sistema de teleprocesamiento

Este sistema utiliza una computadora y una CPU o servidor. Todo el procesamiento es efectuado por esta única computadora, el CPU. En la figura 10, se muestra un sistema de teleprocesamiento típico. Los usuarios operan terminales no inteligentes: microcomputadoras que emulan a terminales no inteligentes, y que transmiten a la macrocomputadora mensajes de transacciones y datos.

La porción de control de las comunicaciones del sistema operativo recibe los mensajes y los datos y los envía al programa de aplicación apropiado. El programa llama al DBMS solicitando servicios, y el DBMS utiliza la porción de administración de datos del sistema operativo y procesa la base de datos. Terminada la transacción, los resultados son devueltos a los

usuarios en las terminales no inteligentes vía la porción de controles de comunicaciones del sistema operativo.

La figura 10 muestra n usuarios solicitando transacciones procesadas por tres distintos programas de aplicación. Dado que existe poca inteligencia en el extremo usuario (las terminales no son inteligentes), todo los comandos para el formato de la pantalla deben ser generados por la CPU o servidor y transmitidos por las líneas de comunicación. Por lo general, la interfaz de usuario está orientada a caracteres y es primitiva.



- OS_{cc}** = Porción de control de comunicaciones del sistema operativo.
- OS_{dm}** = Porción de administración de datos del sistema operativo.
- AP** = Aplicación.

Figura 10. Afinidad de programas en su sistema de teleprocesamiento

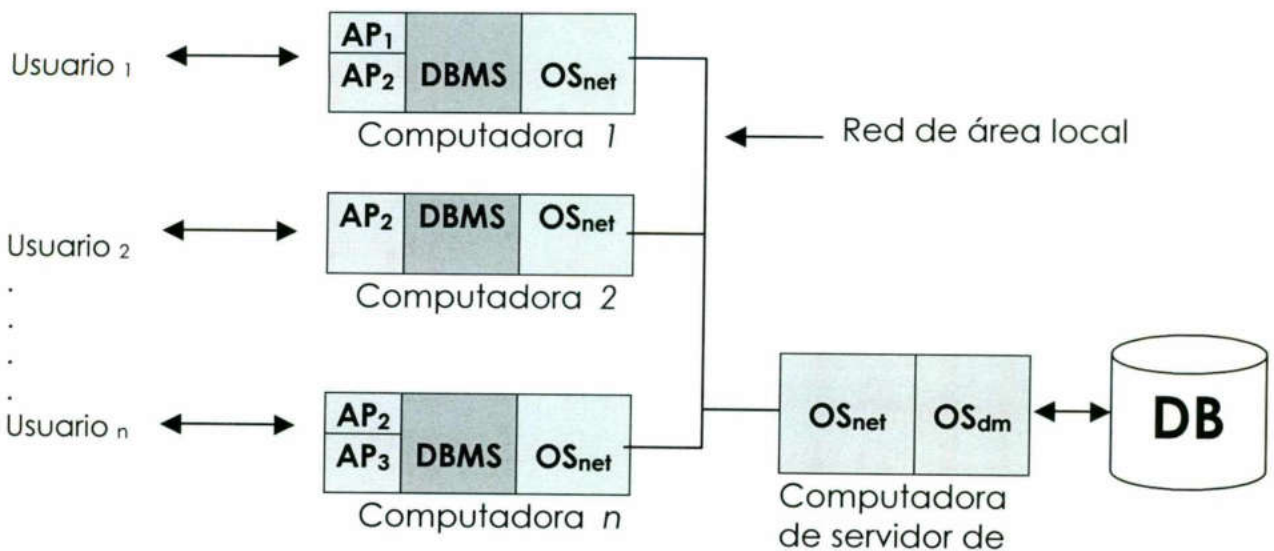
Estos sistemas se conocen como sistemas de teleprocesamiento, y todas las entradas y todas las salidas son comunicadas a la macrocomputadora para su procesamiento a distancia.

Los sistemas de teleprocesamiento han sido la alternativa más común para sistemas de base de datos multiusuario. Conforme se ha ido reduciendo la afinidad precio-rendimiento de las computadoras, y con el advenimiento de las microcomputadoras, han empezado a ser utilizadas otras alternativas que requieren varias computadoras.

Sistemas de recursos compartidos

Una segunda arquitectura de multicomputadoras (figura 11), se conoce como de recursos compartidos y distribuye las computadoras de procesamiento no sólo a los programas de aplicación sino también al DBMS. El servidor es un servidor de archivos, y no un servidor de base de datos. Todos los sistemas de recursos compartidos emplean redes de área local de microcomputadoras.

La arquitectura de recursos compartidos fue desarrollada antes de la arquitectura cliente - servidor , y en muchas maneras, es más primitiva que la de cliente - servidor. En el caso de los recursos compartidos, el DBMS existente en la computadora de cada usuario envía solicitudes a la porción de administración de datos del sistema operativo en el servidor de archivos, para procesamiento a nivel de archivos. Esto significa que cruza por la LAN mucho más tráfico en comparación con la arquitectura cliente servidor.



OS_{net} = Porción de comunicaciones de red del sistema operativo.
OS_{dm} = Porción de administración de datos del sistema operativo.
AP = Aplicaciones.

Figura 11. Arquitectura de recursos compartidos

Como ejemplo para tratar de entender este tipo de sistema, se realizará una consulta para obtener Name y Address de todas las hileras de la tabla CUSTOMER donde Zip sea igual a 98033. En un sistema cliente - servidor, el programa de aplicación enviaría el siguiente comando SQL:

```
SELECT NAME, ADDRESS  
FROM CUSTOMER  
WHERE ZIP=98033
```

El servidor responderá con todos los Name y Address que calificarán.

En un sistema de recursos compartidos, el DBMS está en la computadora local, y ningún programa del servidor de archivos es capaz de procesar SQL o cualquier otro lenguaje similar.

Todo procesamiento como éste deberá ser llevado a cabo en la computadora del usuario, y el DBMS deberá solicitar al servidor de archivos que trasmita la totalidad de la tabla CUSTOMER. Y si dicha tabla tiene índices, o cualquier otra información general asociada, las estructuras de la información general también deberán ser transmitidas. En el caso de los recursos compartidos, se deben transmitir muchos más datos a través de la LAN. Y además, mientras una computadora de un usuario está procesando una solicitud, ocupa una porción grande de la base de datos. Puesto de otra forma, cuando se comparten recursos, se requiere de un alto nivel de bloqueo, y como consecuencia, se reducen los resultados del procesamiento.

Por estos problemas, los sistemas de recursos compartidos se usan poco para procesar bases de datos de multiusuario orientadas a transacciones. En cada una de las transacciones son demasiados los datos que necesitan bloquearse y transmitirse, por lo que daría como resultado un rendimiento muy pobre tratar de utilizar esta arquitectura para el procesamiento de transacciones.

Sin embargo, hay una aplicación de base de datos para la cual esta arquitectura tiene sentido: **el procesamiento de consultas de datos descargados, y extraídos**. Si uno o más usuarios necesitan tener acceso a grandes porciones de la base de datos con la finalidad de producir informes o responder a consultas, puede tener sentido usar un servidor que descargue grandes secciones de datos. Los datos descargados no son actualizados y no son devueltos a la base de datos.

Los sistemas de recursos compartidos también son utilizados en aplicaciones que no son de base de datos. Se utilizan frecuentemente LAN de recursos compartidos para aplicaciones que requieren de discos grandes y rápidos y así almacenar archivos de un solo usuario de gran tamaño, hojas de cálculo extensas, y similares. También se utilizan para compartir impresoras, graficadoras y otro equipo periférico costoso.

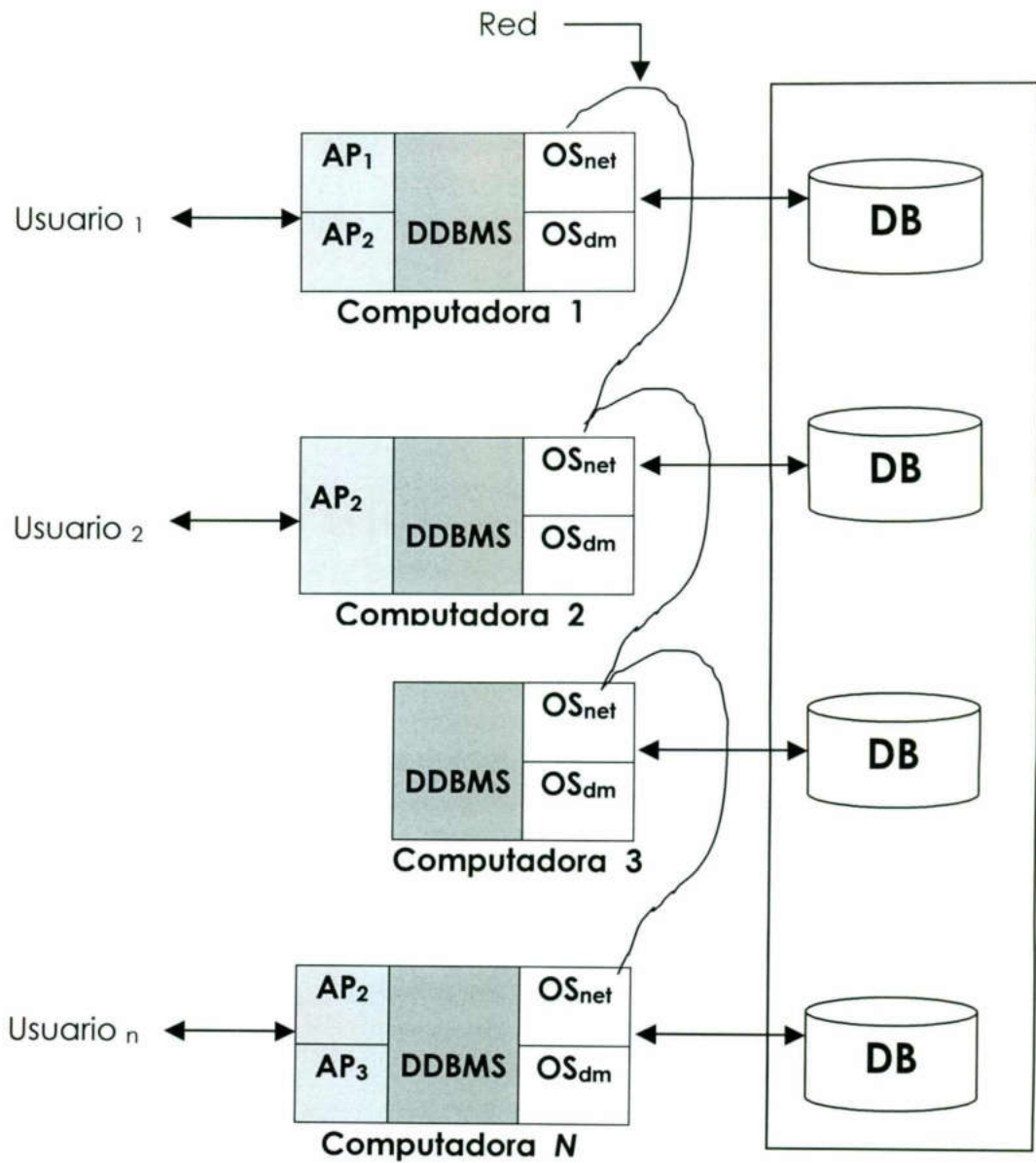
Sistema de base de datos distribuidas

La tercer alternativa (figura 12), es un sistema de base de datos distribuida, en el cual la base de datos misma está repartida en varias computadoras. La base de datos (o una de porción de la misma) está almacenada en todas las computadoras N. Tal y como se muestra, las computadoras 1, 2 y N procesan tanto aplicaciones como la base de dato, y la computadora 3 únicamente procesa la base de datos.

En la figura 12 la línea punteada alrededor de los archivos indica que la base de datos ésta compuesta de todos los segmentos de la base de datos en todas las N computadoras. Tales computadoras pudieran estar localizadas en un mismo lugar o en sitios distintos del mundo.

Haciendo comparación entre las diferentes arquitecturas tal como el sistema de cliente - servidor, el sistema de recursos compartidos y el sistema de teleprocesamiento: Todas ellas utilizan varias computadoras para las aplicaciones o para el procesamiento del DBMS. Por lo tanto, la mayoría de las personas dirían que las tres arquitecturas son ejemplos de sistemas distribuidos, porque el procesamiento de aplicaciones ha sido distribuido entre varias computadoras.

En la figura 12 se observa que la base de datos misma queda distribuida en la arquitectura. Ni las arquitecturas de cliente - servidor ni la de recursos compartidos distribuyen la base de datos en varias computadoras. La mayor parte de las personas jamás se referirán a los recursos compartidos o al cliente - servidor como sistemas de base de datos distribuidas.



OS_{net} = Porción de comunicaciones de red del sistema operativo
 OS_{dm} = Porción de administración de datos del sistema operativo.

DDBMS = DBMS distribuido

Figura 12. Arquitectura de base de datos distribuida

4.2 Estructura de Internet y direcciones IP

Se dice que un sistema de comunicaciones proporciona un servicio universal si permite que cualquier computadora anfitrión se comuniquen con cualquier otro anfitrión o host, para esto se necesita un método aceptado de manera global para identificar cada computadora. Los diseñadores de TCP/IP eligen un esquema análogo al direccionamiento de las redes físicas en el que cada anfitrión de red de redes tiene asignada una dirección de un número entero de 32 bits, llamada *dirección IP*, estos números son seleccionados para hacer eficiente el ruteo.

Conceptualmente, cada dirección IP es un par (*netid*, *hostid*), en donde *netid* identifica una red y *hostid* un anfitrión dentro de la red.

Las direcciones IP se dividen en tres clases A, B y C, cada dirección consiste en cuatro octetos, o conjuntos de ocho dígitos binarios, separados por decimales. Se puede diferenciar una dirección IP según los tres bits del primer octeto, en los que son necesarios sólo dos bits para distinguir entre los tres tipos de clases.

Por ejemplo, el primer octeto de la clase A está en el rango de 1 a 127, la clase B incluye de 128 a 191, y la clase C es de 192 a 223. Las direcciones de la clase A asignan 7 bits al campo *netid* y 24 al campo *hostid*, lo que significa que utilizan los últimos tres octetos para identificar nodos IP específicos, mientras que las direcciones de la clase B asignan 14 bits al campo *netid* y 16 bits al campo *hostid* por eso utilizan los dos últimos octetos para este propósito y las direcciones de clase C asignan 21 bits al campo *netid* y solo 8 bits al campo *hostid*, lo que significa que utiliza el último octeto.

Aunque solo hay 126 direcciones de clase A en el mundo, cada una de ellas puede ser utilizada para crear casi diecisiete millones de direcciones IP anfitrión únicas. Hay mas direcciones de clase B disponibles, sin embargo cada una solo puede ser empleada para utilizar 65, 534 direcciones de anfitrión. Además, aunque las de la clase C son las más abundantes solo pueden ser utilizadas para crear 254 nodos direccionables.

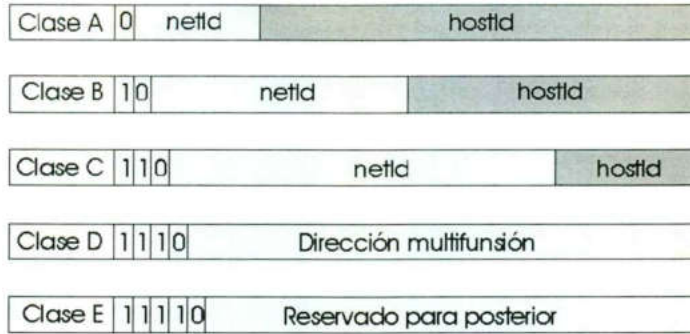


Figura 13. Tipos de direcciones en Internet.

Las direcciones IP se escriben como cuatro enteros decimales, separados por puntos, en los que cada entero proporciona el valor de un octeto de la dirección IP, por lo tanto, la dirección de 32 bits del servidor Guelaguetza del Instituto Tecnológico de Oaxaca sería se presenta en la (Tabla 2).

1er. Octeto	2do. Octeto	3er. Octeto	4º. Octeto
11001000	10111	1011011	10100

Tabla 2. Dirección IP de 32 bits por octetos.

4.3 Dirección IP

La dirección **200.23.91.20** se representa de la siguiente forma.

Clase	Dirección más baja	Dirección más alta
A	0.1.0.0	126.0.0.0
B	128.0.0.0	191.225.0.0
C	192.0.1.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Tabla 3. Clasificación de las clases de direcciones IP.

Existen direcciones IP que no pueden ser utilizadas en una red local, como lo son las direcciones especiales: Dirección Nula (Red o Máquina Actual) y Broadcast (Todos Unos).

Es obvio que resulta casi imposible obtener una dirección de clase A e incluso de la clase B se están volviendo escasas. Por lo tanto, la mayor parte de las redes conectadas a Internet deben utilizar múltiples direcciones de la clase C para solucionar el problema de contar con demasiados dispositivos y direcciones insuficientes.

Un proceso llamado *Protocolo de Configuración Dinámica de Anfitrión* o DHCP (por las siglas en inglés (Dynamic Host Configuration Protocol) está diseñado para solucionar estos dos problemas. En los términos más elementales, DHCP es una aplicación Cliente-Servidor que asigna o presenta una dirección IP a un dispositivo en forma automática cuando se conecta con la red. El dispositivo, como una PC de red ejecuta una pieza de Software llamada cliente DHCP que envía una solicitud cuando se inicia la sesión. El servidor recibe la solicitud y asigna una dirección al cliente. Se puede configurar el servidor DHCP de tal forma que asigne dispositivo a una dirección permanente, por lo general conocida como dirección estática, o una dirección temporal para el periodo determinado, por lo común llamada dirección presentada. Al presentar una dirección, no es necesario tener suficientes direcciones para todos los clientes, ya que es probable que no todos estén conectados al mismo tiempo.

4.4 Nombres de dominio TCP/IP en Internet

El mecanismo que implanta una jerarquía de nombres de máquinas para redes de redes TCP/IP se conoce como Domain Name System (sistema de nombres de dominio o DNS). Un servidor DNS es una base de datos distribuida que mantiene un registro de las direcciones IP de los dispositivos en la red y sus nombres asociados son amigables fáciles de pronunciar y recordar para los usuarios, por ejemplo un servidor DNS utiliza para asociar la dirección 200.23.91.20 con el nombre *guelaguetza.itox.mx* Aunque los servidores DNS facilitan a los usuarios ubicar direcciones IP con nombres de dominios, estos son difíciles de configurar y mantener ya que se tienen que editar configuraciones complicadas

El DNS tiene dos aspectos conceptualmente independientes. El primero es abstracto. Especifica la síntesis del nombre y las reglas para delegar la autoridad respecto a los nombres. Segundo, es concreto: especifica la implantación de un sistema de computación distribuido que transforma eficientemente los nombres en direcciones.

El Sistema de nombres de dominio se vale de un sistema de nombres jerárquico conocido como nombre de dominio, un nombre de dominio consiste en una secuencia de nombres separados por un carácter delimitador (el punto), secciones particulares del nombre representan localidades o grupos del sistema, en teoría, el estándar de nombres de dominio especifica un espacio de nombre jerárquico abstracto con valores arbitrarios para las etiquetas, como el sistema de dominio dicta solo la forma de los nombres y no sus valores actuales, es posible, para cualquier grupo que constituya una instancia de sistema de dominio, seleccionar etiquetas para todas las partes de su jerarquía.

La mayoría de los usuarios de tecnología de dominio sigue la jerarquía de etiquetas utilizada por el sistema de dominios oficial de Internet.

Hay dos razones para ello. En primer lugar el esquema de Internet es completo y flexible, se puede adaptar a una amplia variedad de organizaciones y permite a cada grupo seleccionar entre una jerarquía de nombres asignada geográficamente o en función de la estructura organizativa. En segundo lugar, la mayor parte de las localidades sigue el sistema de Internet por que de esta manera puede conectar sus instalaciones TCP/IP a la red global de Internet sin cambiar nombres.

Dado que el esquema de nombres de Internet predomina en casi todos los usos de sistema de nombres de dominios.

4.5 Sistema de nombres de dominios (DNS)

Es común que un nombre de dominio de una estructura jerárquica para una computadora o un grupo de una organización.

La parte del nombre al extremo derecho, el campo de dominio proporciona la categoría más general. Estados Unidos tiene ocho campos de dominio, los cuales se muestran a continuación.

DOMINIO	DESCRIPCION
Arpa	Miembros de ARPANET (obsoleto)
Com	Organizaciones comerciales e industriales
Edu	Universidades e instituciones educativas
Gov	Organizaciones no militares del gobierno
Mil	Militares
Net	Organizaciones para operar la red
Org	Otras organizaciones
Us	Dominio ISO de Estados Unidos

Tabla 4. Nombres de Dominios en Internet

4.6 Nuevos dominios

Cada dirección se asigna teniendo en cuenta grupos inmensos llamados Top Level Domain (TLD) como por ejemplo .com, .gov, .ar, .uy, etc. Hacia mediados de 1.997 la NSF comenzó a preocuparse por la enorme demanda de dominios **.com**. Por esto, se decidió agregar siete nuevos dominios a los ya existentes (.edu, .org, .com, .mil, gov, .net y 230 más según el país de origen). Ellos son:

DOMINIO	DESCRIPCION
.firm	Firmas de negocios
.store	Negocios y compras
.web	Entidades relacionadas con la WWW
.arts	Cultura y artísticos
.rec	Entidades recreativas
.info	Servicios de información
.nom	Individuos o nomenclaturas de tipo personal

Tabla 5. Dominio y descripción de negocios en Internet.

Con excepción de Estados Unidos, cada nación tiene un dominio de dos letras asignadas con base en su código de país, como se define en el documento 3166 de la Organización de Estándares Internacionales (ISO). Por razones históricas, la inicio no fue muy usado el dominio **us**.

En la *tabla 6* se enuncian los principales Dominios Internacionales para las naciones.

PAIS	DOMINIO
Alemania	de
Argentina	ar
Australia	au
Austria	at
Bélgica	be
Bolivia	bo
Brasil	br
Bulgaria	bg
Canadá	ca
Chile	cl
China	cn
España	es
Francia	fr
Holanda	nl
Italia	it
Japón	Jp
México	mx
Reino Unido	uk
República Checa	cz
Rumania	ro
Tailandia	th
Taiwan	tw
Uruguay	uy

Tabla 6. Principales dominios internacionales para las naciones.

4.7 Localizadores uniformes de recursos (URL)

El URL contiene los segmentos de información que un navegador necesita para localizar una página Web. Este busca en un URL para encontrar una página principal. La página principal es la página Web primaria (index.html), que sirve como punto de partida. La siguiente línea es la dirección de una página inicial de una página específica en el servidor guelaguetza del Instituto Tecnológico de Oaxaca.

Ejemplo: Una dirección URL utilizando el protocolo HTTP

<http://guelaguetza.itox.mx/index.html>

HTTP indica al navegador que utilice el protocolo de transferencia de hipertexto para obtener una página Web. Si la primera parte del URL fuera ftp, iniciales del protocolo de transferencia de archivos, eso indicaría al navegador a que solicite una conexión a un sitio ftp, de tal forma que la primera parte del URL debe especificar el tipo de protocolo o servicio a utilizar.

El nombre guelaguetza.itox.mx corresponde al nombre de la computadora y al dominio correspondiente, para este caso el servidor de HTTP se nombra como guelaguetza que pertenece al dominio itox.mx, este nombre identifica de manera única una máquina en Internet y una vez que el navegador tiene el URL, este sabe en donde se encuentra exactamente la página Web.

Existen direcciones que incluyen la tilde (~) la cual indica que es el directorio base del usuario.

El index.html indica al navegador el nombre del archivo a buscar, la extensión .html indica que es un documento de hipertexto. En muchos de los servidores Web, index.html es el nombre del archivo predeterminado a buscar, incluso si el URL no tiene el nombre de la página principal, este automáticamente busca y visualiza el archivo index.html.

En sistemas Unix como en muchos otros, no es necesario que la extensión de las páginas Web sea .html, por el número de caracteres de la extensión, estas pueden ser extensiones .htm

Para que un navegador pueda visualizar una página Web necesita el URL o en su defecto una dirección IP, el URL es la forma de representar direcciones para que un usuario pueda recordar fácilmente una dirección Internet, una dirección Internet es una dirección IP. Ya hemos hablado de la función de los servidores DNS, que prácticamente tienen la función de asociar direcciones URL a direcciones IP.

5.- Servicios de Internet

5.1 El servicio de http y la world wide web

Hasta hace algunos años el uso de Internet estuvo en manos de los expertos, dada la cantidad de conceptos y comandos que el usuario debía conocer para poder entrar al mundo cibernético. En los últimos años, los expertos comenzaron a desarrollar sistemas que pudieran ser usados por personas con pocos conocimientos y experiencia en sistemas de cómputo. Estos sistemas han incorporado el uso del ratón (mouse) y pantallas gráficas, que reducen al mínimo el uso de comandos. De esta manera nació la World Wide Web: un sistema que permite tener acceso a la información de las máquinas conectadas a Internet.

5.2 El servicio de ftp

FTP es una aplicación que permite transferir archivos de una computadora a otra. Es una de las más antiguas herramientas de Internet, pero también una de las más utilizadas. Un servidor de FTP permite a los clientes navegar por la estructura de su directorio y transportar archivos en cualquiera de las dos direcciones. No importa el tipo de máquina ni su sistema operativo, la conexión es "transparente".

Existen dos formas distintas de compartir archivos, el acceso en línea y el copiado de archivo completo, el acceso compartido en línea significa que se permite a varios programas acceder de manera concurrente a un solo archivo, los cambios que se realizan al archivo se efectúan inmediatamente y están disponibles para todos los programas que accedan al archivo, el copiado de archivo completo significa que, cada vez que un programa quiera acceder a un archivo, este obtendrá una copia local, el copiado se utiliza para datos de solo lectura.

Un mecanismo de acceso remoto debe manejar nociones de propiedad, autorización y noción de acceso para seguridad de la información.

La forma de establecer una conexión con un servidor FTP desde la línea de comandos de una sesión FTP, es especificarla escribiendo *open* y el dominio correspondiente del servidor a conectarse o la dirección IP correspondiente, al establecerse la conexión el servidor despliega una pantalla para que el usuario pueda identificarse.

El uso más importante de FTP se conoce como FTP anónimo, y permite acceder bases de información o de software sin tener una cuenta en la computadora remota.

FTP ofrece muchas facilidades que van más allá de la función de transferencia misma como:

- *Acceso interactivo: La mayor parte de las aplicaciones ofrecen al cliente una interfaz interactiva, la cual permite interactuar fácilmente con los servidores remotos.*
- *Especificación de formato: El FTP permite al cliente especificar el tipo y formato de los datos almacenados.*
- *Control de autenticación: el FTP requiere que los clientes se identifiquen ante el servidor con un nombre de conexión y una clave de acceso antes de pedir la transferencia de archivos.*

FTP puede utilizarse para diversos servicios como bases de datos o librerías de programas, transferencia selectiva de archivos o como un canal de información privado.

Comando	Acción	Comando	Acción
APPEND	Agrega un archivo	LITERAL	Envia un comando arbitrario <u>ftp</u>
ASCII	Transfiere archivos de tipo <u>ascii</u>	LS	Lista el contenido del directorio remoto
BELL	Beep cuando el comando es completado	MDELETE	Elimina multiples archivos
BYE	Termina la sesion y finaliza	MGET	Recibe multiples archivos
CD	Cambia el directorio de trabajo remoto	MKDIR	Crea un directorio en la maquina remota
CLOSE	Termina la sesion <u>ftp</u>	MPUT	Envia multiples archivos
DELETE	Elimina un archivo remoto	PWD	Imprime el directorio de trabajo remoto
DIR	Lista el contenido del directorio remoto	QUIT	Termina la sesion <u>ftp</u>
DISCONNECT	Termina una sesion <u>ftp</u>	RENAME	Renombra un archivo
GET	Recive un archivo	RMDIR	Remueve un directorio en la maquina remota
LCD	Cambia el directorio de trabajo local	SEND	Envia un archivo

Tabla 7. Comando de para la transferencia de archivos por FTP.

5.3 El Servicio de Telnet

El conjunto de protocolos TCP/IP incluye un protocolo de terminal remota sencillo, llamado TELNET, este permite al usuario de una localidad establecer una conexión TCP con un servidor de acceso a otro No importa si la computadora está en el mismo cuarto o al otro lado del mundo. TELNET transfiere después las pulsaciones del teclado del usuario a la computadora remota como si hubiesen sido hechos desde un teclado unido a la computadora remota. TELNET transporta la salida de la computadora remota de regreso a la pantalla del usuario en un servicio

transparente y da la impresión de que teclado y monitor están conectados físicamente a la computadora remota. Cada pulso de teclado viaja del teclado del usuario a través del sistema operativo hacia el programa cliente, el programa cliente regresa a través del sistema operativo y a través de la red de redes hacia la máquina servidor. Después de llegar a la computadora destino, los datos deben viajar a través del sistema operativo del servidor al programa de aplicación del servidor en un punto de entrada de pseudoterminal, finalmente el sistema operativo remoto entrega el carácter al programa de aplicación que usuario está corriendo. El software de cliente TELNET permite que el usuario especifique una computadora remota dando su nombre de dominio o la dirección IP, esto permite que se realicen conexiones remotas aunque el software de nombres de dominio no este depurado.

TELNET ofrece tres servicios básicos, el primero define una terminal virtual de red, que proporciona una interfaz estándar para los sistemas remotos. los programas clientes no tienen que comprender los detalles de todos los sistemas remotos, se construyen para ocuparse con la interfaz estándar, el segundo incluye un mecanismo que permite al cliente y al servidor negociar opciones, asimismo proporciona un mecanismo de opciones estándar (una de las opciones controla si los datos que se transfieren se valen del conjunto de caracteres ASCII estándar de siete bits o de un conjunto de caracteres de ocho bits). Por último TELNET trata con ambos extremos de la conexión de manera simétrica. En particular TELNET no fuerza la entrada de cliente para que esta venga de un teclado ni al cliente para que muestre la salida en su pantalla. De esta manera TELNET permite que cualquier programa se convierta en cliente, además de que cualquier extremo puede negociar las opciones.

Cuando un usuario invoca a TELNET, un programa de aplicación en la computadora del usuario se convierte en cliente. el cliente establece una conexión TCP con el servidor por medio de la cual se comunicarán, una vez establecida la conexión el cliente acepta los pulsos de teclado del usuario y los manda al servidor, al tiempo que acepta caracteres de manera concurrente que el servidor regresa y despliega en la pantalla del usuario. El servidor debe aceptar una conexión TCP del cliente y después transmitir los datos entre la conexión TCP y el sistema operativo local. En práctica el servidor es más complejo, pues debe manejar diversas conexiones concurrentes. Normalmente un proceso de servidor maestro espera nuevas conexiones y crea un nuevo esclavo para manejar nuevas conexiones.

5.4 El servicio de correo electrónico

E-mail es el servicio de aplicación utilizado más ampliamente, ofrece un método rápido y conveniente de transferencia de información privada, puede adaptarse al envío de pequeñas notas y grandes y voluminosos documentos mediante un mecanismo sencillo, de hecho hay más usuarios que envían archivos por correo electrónico que por el protocolo de transferencia de archivos. La entrega de correo difiere de otros usos de redes, los protocolos de red envían paquetes directamente a sus destinos utilizando límites de tiempo y retransmisión para los segmentos individuales si no se devuelve un acuse de recibo. Sin embargo, en el caso del correo electrónico el sistema debe proporcionar los medios cuando la máquina remota o las conexiones de la red han fallado. El emisor no desea esperar a que la máquina remota este disponible para continuar trabajando, ni el usuario quiere que se aborte la transmisión sólo por que las comunicaciones con la máquina remota no están disponibles temporalmente.

Para manejar las entregas con retraso, el sistema de correo utiliza una técnica conocida como *spooling*. Cuando el usuario envía un mensaje de correo, el sistema coloca una copia en su área de almacenamiento privado *spool* junto con la identificación del emisor, recipiente, máquina destino y hora de depósito. El sistema indica entonces la transferencia hacia la máquina remota como una actividad subordinada o secundaria, permitiendo al emisor que continúe con otras actividades computacionales.

5.5 El servicio de IRC

El servicio de IRC (Internet Relay Chat) es un lugar virtual, donde las personas pueden reunirse y conversar. Para utilizar este servicio lo único que se debe hacer es conectarse a un servidor IRC, entrar a un canal y conversar. Para esto existen diferentes servidores situados alrededor del mundo que son accedidos por personas diariamente con el único fin de conocerse y entablar conversaciones.

Con el avance de JAVA ahora esos servidores también se encuentran en la red. Por ejemplo, usted puede conocer gente de todas las edades desde www.planetatierra.com y conversar directamente desde un applet JAVA.

6.- Bases de Datos Vs. Internet

Sin duda alguna la infraestructura de Internet a dado cabida a nuevos tipos de aplicaciones y servicio a los usuarios desarrollados por empresas, organizaciones y gobiernos. Parte fundamental de estos servicios son la implementación de bases de datos accedidas a través de la World Wide Web. El alcance y el fácil acceso a ellas, así como la reducción de costos y la popularidad que ha cobrado la Web, son los principales atractivos que ofrece una aplicación de esta naturaleza.

6.1 ¿Porqué utilizar bases de datos en la web?

La Web es un medio para localizar/enviar/recibir información de diversos tipos, aun con las bases de datos. En el ámbito competitivo, Las compañías toman ventajas de esta vía electrónica que proporciona una buena presentación de información para reducir costos, aumentando la rapidez de difusión de la misma.

Internet provee de un formato de presentación dinámico para ofrecer campañas y mejorar negocios, además de que permite acceder a cada sitio alrededor del mundo, con lo cual se incrementa el número de personas a las cuales llega la información.

Millones de personas alrededor del mundo hacen uso de Internet, lo cual demuestra el enorme potencial que esta red ha alcanzado, con lo cual se puede decir que en un futuro no muy lejano, será el principal medio de comunicación utilizado para distintos fines.

Pero, no sólo es una vía para hacer negocios, sino también una gran fuente de información, siendo éste uno de los principales propósitos con que fue creada.

Una gran porción de dicha información requiere de un manejo especial, y puede ser provista por Bases de datos.

En el pasado, las Bases de datos sólo podían utilizarse al interior de las instituciones o en redes locales, pero actualmente la Web permite acceder a bases de datos desde cualquier parte del mundo. Estas ofrecen, a través de la red, un manejo dinámico y una gran flexibilidad de los datos, como ventajas que no podrían obtenerse a través de otro medio informativo.

Con estos propósitos, los usuarios de Internet o Intranet pueden obtener un medio que puede adecuarse a sus necesidades de información, con un costo, inversión de tiempo, y recursos mínimos. Asimismo, las Bases de datos serán usadas para permitir el acceso y manejo de la variada información que se encuentra a lo largo de la red.

6.2 Seguridad

Otro aspecto a considerar de suma importancia es la seguridad en el acceso a la base de datos. Este punto es uno de los más importantes en la interconexión de la Web con Bases de datos. A nivel de red local, se puede permitir o impedir, a diferentes usuarios el acceso a cierta información, pero en la red mundial de Internet se necesita de controles más efectivos en este sentido, ante posible espionaje, copia de datos, manipulación de éstos, etc.

Regla 1.

La identificación del usuario: Es una de las formas de guardar la seguridad. Las identidades y permisos de usuarios están definidas en los Archivos de Control de Acceso.

Pero la seguridad e integridad total de los datos puede conservarse, permitiendo el acceso a distintos campos de una base de datos, solamente a usuarios autorizados para ello.

En este sentido, los datos pueden ser presentados a través de la Web de una forma segura, y con mayor impacto en todos los usuarios de la red mundial.

Regla 2.

Integración. Para la integración de Bases de datos con la Web es necesario contar con una interfaz que realice las conexiones, extraiga la información de la base de datos, le dé un formato adecuado de tal manera que puede ser visualizada desde un browser de la Web, y permita lograr sesiones interactivas entre ambos, dejando que el usuario haga elecciones de la información que requiere.

7.- ¿Qué se necesita para la implementación de una Base de Datos con acceso a Internet ?

- PANORAMA GENERAL
- LOS PROGRAMAS CGI COMO INTERFACES PARA BD.
- LA INTERFAZ
- METODOS DE TRANSFERENCIA A UN PROGRAMA CGI
- INTERFACES PARA RECUPERACION DE INFORMACION DE UNA BD.
- EJEMPLO DE APLICACIÓN - INTERNET INFORMATION SERVER

7.1 Panorama General

La Tecnología

Un servidor Web es un programa que se ejecuta en una computadora conectada a Internet, donde su trabajo es escuchar en un puerto TCP/IP predefinido las solicitudes de cliente y luego responder a navegadores Web con contenido basado en esas solicitudes. Cuando se escribe una dirección URL en el navegador, es proyectado en una dirección y puerto IP correspondiente a un servidor Web específico.

Después de haber establecido una conexión, el cliente y el servidor se comunican con el Protocolo de transferencia de Hipertexto (HTTP). Por lo general el servidor Web envía un bloque de texto HTML en el navegador, mismo que analiza el HTML y puede solicitar contenido adicional como información gráfica. El modelo trabaja bien para la información estática, sin embargo si queremos hacer que nuestra página presente información en base a las peticiones tecleadas por el usuario, CGI (COMMON GATEWAY INTERFACE) es la respuesta.

El servidor ejecuta un programa CGI como un proceso por separado para satisfacer las solicitudes de los usuarios, como puede ser una consulta a una bases de datos. Dado que un programa CGI es externo al navegador Web, puede ser escrito casi en cualquier lenguaje, ya sea compilado o interpretado. Los lenguajes populares de CGI son Perl, C, e incluso el shell de UNIX. Algunos servidores Web ofrecen bibliotecas e intérpretes para JAVA y Visual Basic para ser utilizados por programas CGI.

En la actualidad existen diferentes tipos de herramientas para implantar una base de datos en la World Wide Web. La capacidad y alcance del software disponible depende directamente del hardware con que se cuente para la implantación de la base de datos, del sistema operativo en función y del diseño de la misma. Este puede ser centralizado o distribuido.

El servidor de base de datos puede alojarse junto con el servidor que atiende al navegador o en un servidor o servidores de bases de datos diferentes, en la misma red o en diversas redes. Otro factor importante es el número de usuarios esperados para consultar la base de datos y determinar un rango de usuarios contemplado, así como un crecimiento a futuro de la misma ya que del tamaño de las tablas dependerá también la rapidez de consulta.

La calidad de los dispositivos de comunicación y la capacidad del servidor tienen gran relevancia para ofrecer un buen servicio a los usuarios.

El sistema operativo

Sin duda alguna el factor más importante para un buen desempeño en el rendimiento, es el sistema operativo.

Esto determina la posible potencia de la base de datos. Los sistemas operativos han desarrollado bases sólidas para este tipo de aplicaciones. Existen en el mercado sistemas UNIX, Windows, OS/2 con grandes capacidades y con un alto grado de estabilidad. En los últimos años se han desarrollado sistemas operativos experimentales con herramientas de comunicación. Un ejemplo de ellos es Linux que se ha convertido en el sistema operativo para PC con capacidades Internet, pues configura todos los servicios en una computadora sin exigir gran requerimiento de hardware.

Las máquinas UNIX que corren SunOS, Solaris, HP-UX o cualquier otra variedad de UNIX son las plataformas preferidas para bases de datos de gran tamaño. No olvidemos también la aceptación que Windows NT ha tenido en el mercado de los sistemas operativos de red y la madurez que ha alcanzando en los últimos Resource Kit.

Tecnología de resguardo de información

Al elegir una base de datos se debe considerar no solo el tipo de dato, si no la cantidad de datos que va a almacenar. Si la cantidad de registros está en la escala de los cientos de millones el espacio de almacenamiento se vuelve crucial. Para esto se deben considerar los controladores nativos del disco duro, controladores SCSI (Small Computer Systemes Interface) y reflejo de datos. Los controladores SCSI son la norma en equipos Macintosh y UNIX. Estos pueden manejar hasta siete dispositivos cada uno y es posible utilizar varios controladores simultáneamente.

Los discos duros también pueden utilizarse para reflejar datos (el reflejo es una técnica de resguardo de información) en más de una unidad. Si el motor de la base de datos acepta el reflejo de datos, estos pueden escribirse en dos discos diferentes para mayor seguridad. Si un disco falla, la base de datos se cambia automáticamente al otro, cuando la unidad original se devuelve al servicio, la base actualizará los datos en el disco reparado, o en uno nuevo, para que concuerde con el disco reflejado. Cuando el disco nuevo está completamente actualizado, la base conmuta al disco original y continúa reflejando la información en el disco alternativo.

Hoy en día existen otros métodos de resguardo de información. La tecnología Raid 5 (Independent Disk Array) es una de ellas. Este es un sistema de discos independiente con integración de códigos de error mediante una paridad. Los datos y la paridad son guardados en los mismos discos, por lo que se consigue aumentar la velocidad de demanda, ya que cada disco puede satisfacer una demanda independientemente de los demás. En el Raid 5 guarda la paridad del dato dentro de los discos y no hace falta un disco para guardar dichas paridades. La paridad se genera haciendo un XOR de los datos A0,B0,C0,D0 creando la zona de paridad PAR0. La paridad nunca se guarda en los discos que contienen los datos que han generado dicha paridad, ya que en el caso en que uno de ellos se estropeará como por ejemplo el dato A0, bastaría con regenerar la banda B0, C0,DO,PAR0 para que el dato vuelva a reestablecerse.

7.2 Los Programas C G I como interfaces para acceder a bases de datos

Un gateway es una conexión con el sistema operativo externo, la acción de llamar un programa CGI desde un navegador Web es muy sencilla para el usuario, lo cual es uno de los principales atractivos del Common Gateway Interface, el tener acceso a una base de datos a través de un programa CGI tiene una metodología propia, comúnmente el usuario hace clic sobre un botón predeterminado o sobre un vínculo, en este momento el navegador envía una solicitud de ejecutar el programa CGI al servidor Web, el servidor Web revisa la configuración y los archivos de acceso para asegurarse que se cuenta con el permiso de ejecución del programa CGI y se asegura de que este exista, cualquier resultado producido por el programa CGI se devuelve al navegador Web que despliega el resultado.

Los programas CGI son un gateway de doble sentido. Los datos pueden transferirse al programa CGI para su procesamiento, al igual los programas CGI pueden devolver información al servidor Web. Con esto la información introducida por el usuario puede afectar el comportamiento del programa CGI, y los resultados devueltos por el programa son resultado directo de lo que introduce el usuario. En algunos casos los programas CGI se ejecutan al cargar la página y los resultados se despliegan como parte de la misma (figura 11).

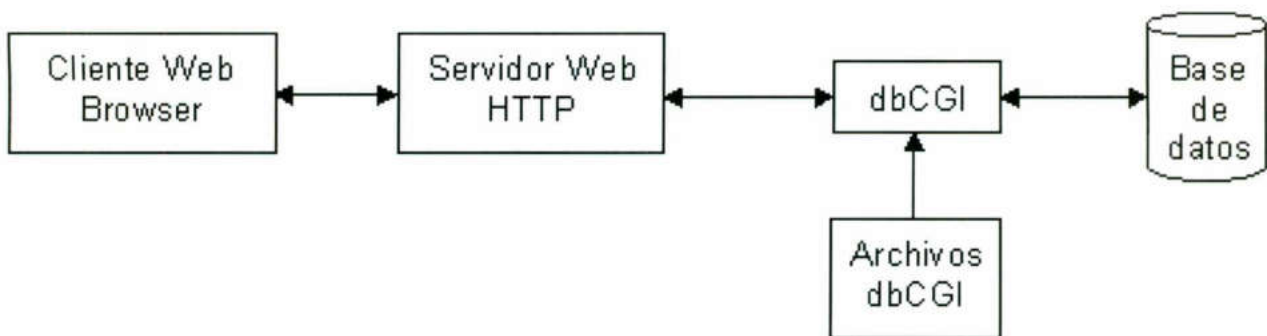


Figura 14. Esquema de aplicación utilizando programas CGI

Algunos lenguajes que ofrecen un manejo de las Bases de Datos por Internet

Los programas que maneja el CGI pueden estar compilados en diferentes lenguajes de programación. El más popular para el desarrollo de contenidos Web es el lenguaje Perl de distribución gratuita, aunque también podemos mencionar:

1. C
2. C++
3. Java.

El funcionamiento de esta tecnología es muy sencillo. Los scripts residen en el servidor, donde son llamados, ejecutados y regresando información de vuelta al usuario.

Para una mejor aclaración presentamos en la Figura 15:

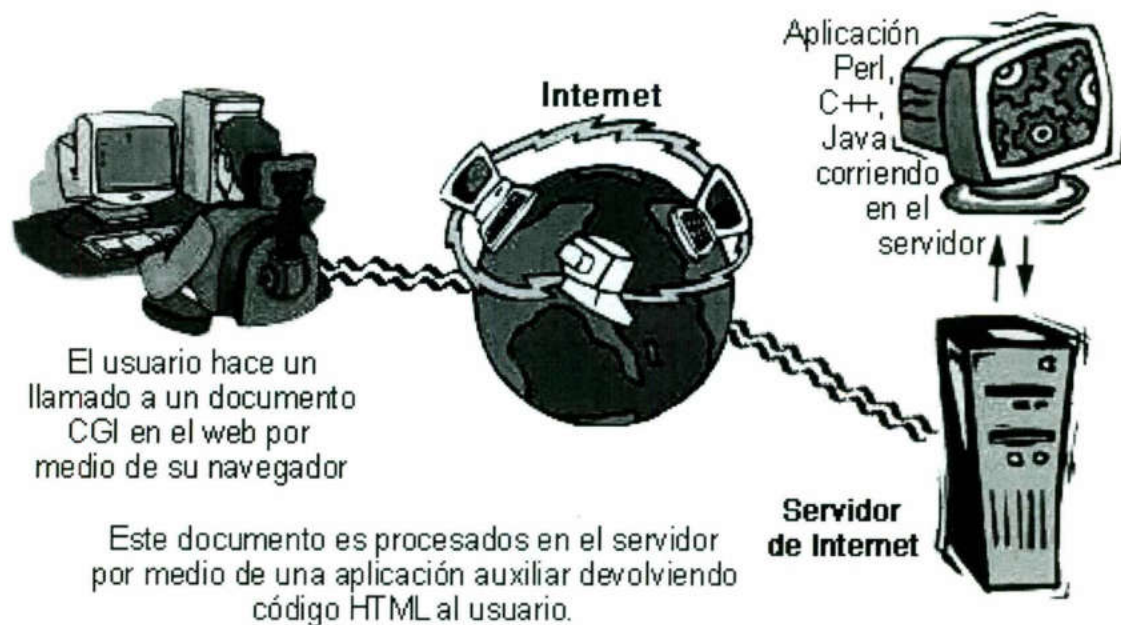


Figura 15. Tipos de Lenguajes

Un ejemplo de un llamado a un script CGI puede ser:
<http://www.maestrosdelweb.com/cgi-bin/ejemplo.cgi>

Hay que tomar muy en cuenta que existen dos formas de enviar información; siendo estas por medio de **GET** y **POST**. El **GET** envía la información dentro del URL separando los campos con signos de interrogación '?' antes de la información, como por ejemplo:

<http://www.maestrosdelweb.com/cgi-bin/ejemplo.cgi?nombre=Christian&cargo=Webmaster>

Esta forma presenta varias desventajas, ya que no permite manejar grandes cantidades de información y por ser una dirección, los Caches de los navegadores pueden guardar los resultados, lo que dificulta el acceso nuevamente a la aplicación, pero simplifica el uso de los scripts con solo utilizar una dirección URL.

La otra forma de enviar la información es por medio de **POST**, el cual envía la información directamente al servidor, haciendolo invisible en el URL y permitiendo cualquier cantidad de información, como el envío de archivos.

Las diferencias entre GET y POST son primordiales para adaptar una aplicación y hacerla funcionar.

Así, el siguiente paso luego de manejar el diseño de páginas web es aventurarse a estas tecnologías que darán un nuevo concepto de interacción en un sitio web.

Existe una gran cantidad de código gratuito disponible para iniciar, pero es necesario contar con un servidor adecuado y que soporte esta tecnología, así como la modificación de permisos dentro de sus directorios. La forma más fácil de hacer pruebas es con el compilador Perl que viene incluido en cualquier plataforma Unix o derivada como es el caso de Linux.

En otras plataformas de 32 bits como Windows puede ser instalado el Perl32.

7.3 La Interfaz

La interfaz proporciona un método para interactuar con una base de datos. Los proveedores de bases de datos pueden proporcionar una interfaz de varios niveles de complejidad dependiendo de las necesidades.

Los gateway reciben datos transferidos desde un navegador Web mediante un servidor de HTTP y los convierten a un formato que la base de datos pueda entender. La información convertida se transfiere a la interfaz de la base de datos y esta la ejecuta. Los resultados se devuelven al programa de gateway, el cual los convierte a un formato de manera que el navegador los pueda desplegar.

Un gateway no puede trabajar solo, requiere de un tipo de canal para hacer contacto con la base de datos, este canal lo proporciona la interfaz de la base de datos, el cual es un software especial que suministra el proveedor.

El gateway se comunica con la interfaz, la cual se pone en contacto con la base de datos.

Dependiendo de las necesidades de la aplicación, el diseño de base de datos puede o no tener capacidades de redes, esto lo determina el software del proveedor, el método más sencillo se alcanza con una base de datos sin capacidad de red, pues el servidor de HTTP, los programas CGI y el servidor de la base de datos están ubicados en la misma máquina, de esta forma los programas CGI tienen acceso a cualquier programa de interfaz que deseen utilizar, las consultas y los resultados devueltos no van y vienen a los programas CGI a través de la red, de esta manera se optimiza el tiempo de respuesta.

Sin embargo se supone que los navegadores están accediendo a la información de manera remota, si la base de datos es de mediano o gran tamaño, y hay muchos usuarios, no es buena idea tener el servidor de base de datos, el servidor de HTTP en la misma computadora.

Si la base de datos incluye capacidades interconstruidas de redes tiene la opción de correr el servidor de HTTP en una máquina remota y acceder a la base de datos utilizando el software de bases para red, muchas veces es preferible ubicar en el servidor de HTTP los programas CGI

y tener una máquina dedicada al servidor de la base de datos para evitar problemas potenciales y proteger los datos tanto como sea posible, este método funciona bien si se tiene más de un servidor de base de datos, un solo servidor de HTTP puede acceder a varias máquinas con bases de datos mediante programas CGI.

Este método aligera la carga de una base de datos y se refleja un notable incremento en el rendimiento, sin embargo esto sólo vale la pena en el caso de bases de datos verdaderamente grandes.

7.4 Métodos para transferir información de un navegador web a un programa CGI

Existen diferentes métodos para transferir información de un navegador Web a un programa CGI, esto depende de cómo se llame al programa en el documento HTML. Estos pueden ser:

Método 1. Por transferencia de parámetros en la línea de comandos:

El lenguaje HTML proporciona un método sencillo en la línea de comandos llamado ISINDEX, el método ISINDEX es la única manera de enviar parámetros de esta forma a un programa CGI. <ISINDEX> es una etiqueta HTML que debe escribirse dentro de la sección <HEAD> del documento. En la utilización de ISINDEX el documento HTML y el programa CGI están escritos en el mismo código.

La razón de ello es que cuando una página Web utiliza el método ISINDEX, no hay forma de indicarle a dónde enviar los parámetros en línea de comandos, entonces los parámetros son enviados asimismo.

Método 2. Por transferencia de variables de entorno:

Las variables de entorno permanecen desde que el Shell de comandos se activa hasta el momento en que se desconecta de la máquina. Las variables pueden establecerse mediante algunos programas, algunos buscan en su ambiente para saber cuáles son los valores de ciertas variables, si la variable no existe, el programa puede usar un valor predeterminado o puede fallar desplegando un mensaje de error.

Un servidor Web utiliza las variables de ambiente y establece los valores de distintas variables cada vez que se llama un programa CGI. El programa CGI puede acceder a los contenidos de esas variables si las necesita y las usa dentro del programa.

Las variables de entorno se usan para alimentar información a un programa CGI. Un programa CGI se llama en forma indirecta a través del servidor por el navegador, ya sea con un método de solicitud HTTP GET o POST.

El código CGI debe examinar una variable de entorno REQUEST_METHOD para determinar si la solicitud fue un GET o un POST, y así, la manera de obtener la información viene de la interactividad del usuario con una página Web. La variable de entorno es creada por el servidor Web antes de que se ejecute el programa CGI. Para el método POST la información llegará a través de la entrada normal, mientras que con el método GET, la información residirá en la variable de entorno QUERY_STRING.

La diferencia radica en que una solicitud que utiliza el método GET pasa variables al incrustarlas en el URL, esta técnica se conoce como *codificación URL*, y la porción del URL que codifica las variables siempre comienza con un signo de interrogación. Esta porción del URL se pasa al programa CGI como variable de entorno QUERY_STRING. Cada variable se expresa como un par solicitado de la forma *nombre=valor*.

Por ejemplo si el valor de mi nombre es una búsqueda yahoo, la solicitud se traduce en: **HTTP://av.yahoo.com/bin/search?p=carlos+jimenez** El URL informa al programa CGI receptor que la variable *p* debe recibir el valor "carlos jimenez. Asimismo parte de la codificación URL está compuesta por una variable de caracteres especiales.

SIMBOLO	FUNCIÓN
+	separador de datos o espacio
=	Combina nombres con valores para los pares solicitados
&	Combina pares solicitados
%	Valor hexadecimal sigue en formato %XX

Tabla 8. Símbolos delimitadores del método GET.

El método **POST** puede enviar una cantidad ilimitada de datos por no estar incrustado en el URL. Las variables de entorno pueden tener información sobre la dirección IP del host remoto, nombre del servidor que ejecuta el gateway, nombre del scrip, puerto en que es conectando, software por el cual se está ejecutando y demás tipo de información.

Dentro de un programa CGI, la información POST se recupera de entrada estándar después de leer la variable de entorno `CONTENT_LENGTH` para obtener el tamaño de la información, cuando un programa CGI termina de procesar la información de usuario y está listo para responder solo escribe los resultados en salida estándar. Los resultados deben estar en un formato que el navegador Web comprenda, en C se utiliza la función `printf` para generar HTML al instante.

La especificación CGI define una serie de variables que el servidor debe generar en el entorno de ejecución de los scripts que invocan. Estas variables no son *case sensitive* y un valor NULL equivale a su no definición. Entre ellas me permito hacer una distinción entre *variables primarias* y *variables secundarias*.

Podríamos considerar *primarias* a 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 *secundarias* consideraríamos todas las demás, cuyo contenido es asignado por el servidor de información atendiendo a diversos elementos que definen el estado de la comunicación o al valor de las variables primarias.

Variables de entorno primarias

Por tanto, como variables primarias encontraríamos 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 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 *form*, podrá acceder a esta información la URL-descodificada como si se le hubiera facilitado desde 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.

Variables de entorno secundarias

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 *form*.

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`.

GATEWAY_INTERFACE

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

PATH_TRANSLATED

Indica el valor de `PATH_INFO` una vez el servidor de información ha realizado la traslación 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 *ident* facilita 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

Ésta variable sí es *case sensitive* y facilita 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. 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

Facilita 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.

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 a 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.

Transferencia de datos mediante entrada estándar :

Esta es la manera en como los formularios HTML envían datos capturados por medio de listas de selección, casillas de verificación, cuadros de entrada de texto y otras características interactivas disponibles al servidor de HTTP y al programa CGI.

Generalmente la entrada estándar es la terminal que está utilizando el usuario para correr el programa, si un programa produce algún dato resultante mediante la ejecución del programa, el resultado llega a la pantalla terminal mediante la *salida estándar*. Si el programa produce algún mensaje de error, se envía a la pantalla por medio del *error estándar*. Entrada, salida y error estándares son canales de datos proporcionados por el sistema operativo para conectar un dispositivo de entrada/salida, como su terminal a una pieza de software que sea capaz de enviar o recibir datos. El servidor de HTTP puede recibir información enviada por el usuario y transferirla a un programa CGI mediante la entrada estándar, para después obtener los resultados mediante la salida estándar.

A diferencia de ISINDEX, el servidor HTTP no procesa los argumentos antes de enviarlos al programa CGI. Los argumentos llegan codificados al programa tal y como los envía el navegador.

7.5 Interfaces para bases de datos recuperación de información de una base de datos.

Cuando alguien usa un navegador Web para acceder a una base de datos hay varios componentes que intervienen para transferir la consulta del usuario a la base de datos y devolver los resultados al navegador, la acción se desarrolla de la siguiente manera (figura 12):

1. El usuario llama a un programa gateway que utiliza CGI, haciendo clic en un Hipervínculo u oprimiendo un botón del formulario.
2. El navegador reúne toda la información escrita por el usuario para enviarla al programa CGI.
3. El navegador contacta al servidor de HTTP en la máquina donde reside el programa CGI, pidiéndole que localice a este último y le transfiere la información.
4. El servidor de HTTP corrobora si la máquina solicitante tiene autorización de acceso al programa CGI.

-
5. Si el usuario tiene acceso, el servidor de HTTP localiza el programa gateway y transfiere a este la información del navegador Web.
 6. Se ejecuta el programa Gateway.
 7. El proceso Gateway convierte la información recibida a un formato que la base de datos sea capaz de entender.
 8. El Gateway usa el módulo de la base de datos para transferir la consulta a la interfaz de la base.
 9. La interfaz de la base de datos analiza la sintaxis de la consulta para asegurar que sea precisa.
 10. Si la interfaz encuentra un error de sintaxis en la consulta, se envía un mensaje de error al programa Gateway.
 11. El mensaje de error se envía al servidor de HTTP, el cual lo transfiere al navegador Web para que este lo despliegue al usuario.
 12. Si no hay error, la interfaz envía la consulta a la bases de datos.
 13. La base de datos atiende la consulta y devuelve los resultados al programa gateway a través de la interfaz.
 14. El programa gateway formatea los resultados y los envía al servidor, por medio del CGI, para su envío al navegador Web.
 15. El navegador Web despliega los resultados

Hacer inserciones, actualizaciones y eliminaciones requiere de un procedimiento más complejo que el necesario para hacer simples consultas.

El código de inserción debe verificar si existen los datos en la base de datos, con el fin de evitar duplicidad en la información, el código de actualización debe asegurarse que la información exista, antes de modificarla, en caso contrario debe originarse un error.

Las acciones de eliminación y edición debe de asignarse a un grupo reducido y confiable para evitar la eliminación de datos necesarios.

Para procesar algo mas que una simple consulta en un ambiente de esta naturaleza, es necesario algún tipo de control de acceso como lo pueden ser el comparar el identificador de conexión del usuario contra una lista de usuarios autorizados para determinada acción, asignar a cada tipo usuario un identificador de conexión y asignar a cada tipo nivel un nivel de acceso basándose en las funciones que tienen permitidas o utilizar la autenticación de clave de acceso de la base de datos destino para validar los niveles de acceso a los usuarios. En definitiva, el método que utilice depende de como planea utilizar el programa de gateway.

7.6 Caso Práctico

Sentencia sobre reproducción de bases de datos en Internet

Jurisprudencia : Un caso sobre propiedad intelectual.

El Juzgado de Primera Instancia e Instrucción número 3 de Elda (Alicante) ha dictado una sentencia condenatoria por infracción de los derechos de propiedad intelectual sobre bases de datos y por actos de competencia desleal. Se trata del primer caso en el que una resolución judicial se refiere a la reproducción no autorizada y a la comunicación pública de bases de datos a través de Internet.

La empresa demandante, Editorial Aranzadi, inició el procedimiento a raíz de la existencia de una página web en la que se ofrecía el acceso a jurisprudencia y legislación, y tras la comprobación de que las sentencias y normas que figuraban en dicho servidor, eran las mismas que, tras varios años de recopilación, introducción y sistematización, formaban las bases de datos jurídicas comercializadas por dicha editorial.

La sentencia se basa en dos hechos que resultan concluyentes en el momento de valorar las pruebas recabadas durante las diligencias de comprobación de los hechos. Por un lado, Aranzadi había introducido en el texto de las sentencias una serie de errores intencionados y de identificadores ocultos, que también aparecían en las sentencias de la demandada. Estos identificadores habían sido protocolizados notarialmente con anterioridad a la comercialización de las bases de datos. Por otro lado, la demandada había cambiado la numeración que Aranzadi había dado a las sentencias y a las leyes, pero en vez de asignarle una referencia y un orden distinto, se había limitado a aplicar una macro que multiplicaba el número del documento por 3 y después le sumaba 5. Ello implicaba, por ejemplo que no existiesen las sentencias 1, 2, 3, 4, 5, 6 y 7, ya que la sentencia número 1 de Aranzadi, llevaba el número 8 en la base de datos de la demandada, como resultado de multiplicar 1 por 3 y sumar 5.

Desde el punto de vista jurídico, la novedad de la sentencia es su fundamentación jurídica, que se basa en los artículos del Texto refundido de la Ley de Propiedad Intelectual incorporados a raíz de la transposición de la Directiva comunitaria sobre protección jurídica de bases de datos, y, en especial, por aplicar la protección "sui generis" a las colecciones legislativas y jurisprudenciales.

8.- ¿ Que es ASP?

8.1 Active Server Pages (Páginas Activas de Servidor).

Se trata de una tecnología de Microsoft que mediante varios scripts permite la construcción dinámica de páginas a petición del navegador del usuario. De esta manera el código ASP se encuentra en el servidor y el cliente solo ve el resultado en el HTML de siempre.

Al navegar más de alguna vez nos hemos topado con alguna página que tiene archivos con extensión **".asp"** y nos hemos preguntado que significa éste tipo de archivos.

Microsoft introdujo esta tecnología llamada Active Server Pages en diciembre de 1996, por lo que no es nada nueva. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente.

Traduciendo la definición de Microsoft: "Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web".

El principio de la tecnología ASP es el VBScript, pero existe otra diversidad de lenguajes de programación que pueden ser utilizados como lo es Perl, JScript, etc. El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.

Para explicar mejor el funcionamiento del ASP se presenta en la Figura 16.



Figura 16. Funcionamiento del ASP

Para insertar instrucciones ASP dentro del código HTML se incluye encerrado entre "<% %>". Estos comandos son los que procesa el servidor antes de enviar la página al navegador. A continuación tenemos un ejemplo del código ASP en el servidor y los resultados HTML que serán vistos en el navegador:

Código ASP	Código HTML	Resultado en el Navegador
<pre><P> <% For I = 1 To 5 Step 1 %> <FONT SIZE="<%= I %>">Maestros del Web!
 <% Next %> </P></pre>	<pre><P> Maestros del Web!
 Maestros del Web!
 Maestros del Web!
 Maestros del Web!
 Maestros del Web!
 </P></pre>	<pre>Maestros del Web! Maestros del Web! Maestros del Web! Maestros del Web! Maestros del Web!</pre>

Tabla 9. Vista de diferentes códigos

El ejemplo anterior crea un ciclo que se repite 5 veces y aumenta el tamaño del tipo de letra en una frase establecida.

El desarrollo que se ha venido dando a lo que es ASP ha sido bastante amplio. Entre sus funciones principales del ASP están: el acceso a

base de datos, envío de correo electrónico, creación dinámica de gráficos y otros. Básicamente, muchas cosas que podemos realizar por medio de CGI pueden ser realizadas con esta tecnología. Esto debido a que el ASP es tan eficiente con escribir código directamente a la interfase de aplicación del servidor, con la ventaja de que es más eficiente que el CGI que depende de un compilador ya que el ASP corre como un servicio en el servidor, tomando ventaja de la arquitectura de multitareas.

Para empezar con el desarrollo de las Active Server Pages es necesario un servidor con Windows NT 4.x o mayor y el Internet Information Server. El IIS es una aplicación gratuita que puede conseguirse en el Option Pack del NT. También es posible utilizar ASP en Windows 9x por medio del Personal Web Server junto al Option Pack mencionado anteriormente.

En caso del uso de un servidor Linux, Chilisoft ha desarrollado el Chilisoft ASP que también permite el uso de esta tecnología.

La mayoría de proveedores de Hosting pagado con plataformas NT tienen acceso a esta tecnología en sus servidores. El código puede ser trabajado en cualquier editor HTML o de texto. Existen en el mercado dos herramientas para trabajar profesionalmente el ASP en modo visual como lo son el Drumbeat 2000 y el Visual Interdev de Microsoft.

8.2 ¿Por qué las bases de datos?

Las páginas ASP, proporcionan la posibilidad de trabajar con bases de datos. Esto hace que unas simples páginas Web se puedan convertir en aplicaciones completas de gestión de datos. Con las bases de datos se consigue un dinamismo impresionante, a parte de ahorrar gran cantidad de trabajo.

Por ejemplo, un noticiero que deseara divulgar sus noticias, era necesario hacer una página HTML por cada una de las noticias. Todas las páginas tendrían un diseño similar, lo único que cambiaría sería el contenido de la noticia. Sin embargo, con una base de datos, sólo se tendría que hacer una única página que muestre la base de la noticia que se desea. Pasándole un parámetro con el código de la noticia, la página sabría buscar en la base de datos los textos a mostrar.

Pero no sólo se ahorra en el tiempo de desarrollo, la mayor ventaja que se tendría será la facilidad de realizar cambios. En el ejemplo del noticiero si se quisiera cambiar el formato de las noticias para hacerlo mas atractivo a la vista, habría que recorrerse todas las páginas para cambiar una a una el formato. Con bases de datos, sólo habría que cambiar una página, la de acceso a la base de datos.

Los Objetos ADODB

ASP, a través de los objetos ADODB proporciona un acceso flexible a todo tipo de bases de datos, desde las sencillas como Access hasta otra más potentes como Oracle. Estos objetos permitirán acceder de la misma forma a cualquier tipo de base de datos. De este modo será sencillo cambiar de una plataforma de datos a otra.

La librería ADODB se compone de tres objetos principales, Connection, Record-Set y Command y de otros secundarios que son Field, Parameter, Record, Stream y Error. Con estos objetos se puede gestionar una base de datos mediante Web de igual forma como se haría en una aplicación local.

El Objeto Connection

Este objeto se utilizará para enlazar la aplicación Web con una base de datos. Es decir, en la página Web de la terminal se tendrá un objeto de este tipo, el cual proporcionará un enlace directo a la base de datos que se le indique.

El objeto Connection es el más importante de la librería ADODB, sin él no se podría hacer nada, ya que para trabajar con cualquier elemento de una base de datos, como una tabla o una vista, antes se tendría que tener una referencia a la base de datos.

Además, con el objeto Connection se puede ejecuta sentencias SQL directamente, sin utilizar otro objeto de la librería ADODB.

Métodos y propiedades del objeto Connection, por orden lógico en el que se deberían utilizar en una página.

1. Instanciación: para crear un objeto de este tipo se utilizara la siguiente sentencia:

```
SET Obj_Conn= Server.CreateObject("ADODB.Connection")
```

Se utilizará SET, ya que se asignará a la variable Obj_Conn un objeto. Se conseguirá una instancia del objeto Connection con el método CreateObject del objeto integrado Server. Se le indicará primero la librería, ADODB y luego el objeto en concreto que se desea instanciar, Connection.

2. PropiedadConnectionString: una vez que se ha creado el objeto, se le indicará que base se quiere que ataque. Con esta propiedad, se le dará una cadena de texto al objeto, indicando la base de datos que se quiere utilizar, el tipo de base de datos que es y, si fuese necesario, el usuario y password de la base.

```
Obj_Conn.ConnectionString=  
"driver={Microft Driver Access (*.mdb)};dbp=c:\base\base.mdb"
```

Todo esto no es necesario si se utiliza un DSN. Un DSN es como una agenda donde se guarda las propiedades de una base de datos. Estas propiedades son las mismas que se indicaron a la propiedad ConnectionString: tipo de base, ubicación de la misma, usuario y password. Al DSN se le dará un nombre, de este modo cada vez que se quiera conocer las propiedades de la base, se le llamará al DSN por su nombre. Por ejemplo, se puede crear un DSN con las propiedades de la base de datos a utilizar y llamarlo "Alumnos", con lo que para dar valor a la propiedad ConnectionString sólo se tendría que poner lo siguiente:

```
Obj_Conn.ConnectionString= "Alumnos"
```

3. Propiedad ConnectionTimeout: con esta propiedad se puede configurar otros parámetros como, por ejemplo, el tiempo de espera de la conexión. Este tiempo, por defecto 15 segundos, es el que dedica el objeto Connection a intentar conectarse con la base de datos. Sino lo consigue en este tiempo, se produce un error.
4. Propiedad CommandTimeout: Una vez conectada a la base de datos a la página Web del usuario, se podrá ejecutar sentencias

SQL. Al igual que en la conexión, se tendrá un tiempo de espera máximo para ejecutar una sentencia. Superado ese tiempo, se producirá un error. Por defecto este tiempo es de 30 segundos, pero con esta propiedad se puede cambiar se cree necesario.

5. Método Open: una vez se los parámetros se hayan definido con los que se a acceder a la base de datos, se iniciará la conexión. El método Open realizará la conexión real con la base de datos, tomando los parámetros previamente configurados. Este método también permite insertar directamente la propiedad `ConnectionString`, con lo cual se puede ejecutar justo después de crear el objeto `Connection` y olvidarse del resto de propiedades.

Resumiendo, se tienen tres formas de utilizar el método Open. La primera es después de darle un valor a `ConnectionString`:

```
Obj_Conn.Open
```

Si se quiere utilizar `ConnectionString` se puede incluir esta propiedad dentro de `Open`, ya sea mediante la cadena completa:

```
Obj_Conn.Open "driver={Microsoft Driver  
Access (*.mdb)};dbq=c:\base\base.mdb"
```

O mediante un DSN

```
Obj_Conn.Open "Alumnos"
```

Si fuese necesario introducir un usuario y un password, también se podría hacer con éste método:

```
Obj_Conn.Open "Alumnos","usuario","password"
```

6. Propiedad State: esta propiedad dirá si un objeto `Connection` está abierto o cerrado devolviendo las constantes `adStateOpen` (&H00000001) cuando esté abierto o `adStateClose` (&H00000000) cuando esté cerrado.
7. Propiedad Versión: devuelve la versión del ADO que se esta utilizando.
8. Método Execute: permite ejecutar una sentencia SQL. Por ejemplo, se podría borrar un registro de una tabla sin necesidad de cargar dicha tabla.

```
Obj_Conn.Execute "DELETE * FROM Alumnos WHERE  
DNI='02342334-J' "
```

También se puede utilizar este método para realizar una consulta de selección, en cuyo caso devolverá un objeto de tipo RecordSet con el resultado de la consulta. En este caso se tendría que poner la sentencia SQL entre paréntesis y recibir el resultado con SET, ya que será un objeto:

```
SET Obj_RS=Obj_Conn.Execute("SELECT * FROM Alumnos")
```

9. Método BeginTrans: al ejecutar este método se le indicará que va comenzar una transacción. Esto significa que todo lo que se haga en la base de datos, inserciones, modificaciones y borrados, no será permanentes hasta que con el siguiente método se confirme.
10. Método CommitTrans: este método dará por válida una transacción, es decir, ejecutará todo lo que se hizo desde el comienzo de la transacción.
11. Método RollBackTrans: si al comenzar una transacción se produce un fallo, se puede cancelar todo lo que se ha realizado dentro de esa transacción con este método.
12. Método Close: cierra la conexión con la base de datos, con lo que el objeto Connection se queda disponible para ser utilizado con otra base de datos.
13. Destrucción del objeto: servirá para liberar memoria y perder todo vínculo con la base de datos:

```
SET Obj_Conn = Nothing
```

Es importante destruir todo los objetos una vez que no se utilizan, de este modo optimizaremos el rendimiento del servidor.

El Objeto RecordSet

Con este objeto se tiene acceso a las tablas en sí de la base de datos, ya sea a través de las propias tablas, de vistas o de consultas realizadas por el usuario. Para realizar este objeto, se debe crear antes un objeto Connection, ya que este objeto será quien indique al RecordSet de que base de datos tiene que leer la tabla.

Después de seleccionar la tabla, el objeto RecordSet apuntará al primer registro de la misma. A través de sus métodos y propiedades, se podrá ir corriendo el resto de registros y consultando sus campos. Este objeto también permitirá modificar, borrar y añadir registros dentro de la tabla seleccionada.

El objeto RecordSet contiene métodos y propiedades las cuales se explicarán en orden de acuerdo a como se utilizarán en una página Web. También se dará por sentado que ya está creado el objeto Connection para acceder a la base de datos.

1. Instanciación: es muy similar a la del objeto Connection, simplemente se cambia el nombre del objeto a instanciar:

```
SET Obj_RS = Server.CreateObject ("ADODB.RecordSet")
```

2. Propiedad ActiveConnection: a esta propiedad se asignará el objeto Connection que se ha creado con anterioridad. Ésta es la forma de decir al objeto RecordSet, la base de datos con la que va a trabajar.

```
Obj_RS.ActiveConnection = Obj_Conn
```

3. Propiedad CursorType: indica la forma en que se va a recorrer el RecordSet. Al cual se le pueden asignar cuatro valores:
 - adOpenForwardOnly(0): el objeto RecordSet solo se puede mover de forma secuencial. Es decir, al abrir una tabla el RecordSet se situará en el primer registro de esa tabla y sólo se podrá mover de uno en uno al siguiente. Éste es el cursor por defecto.
 - adOpenKeySet (1): se puede mover en todas direcciones por el objeto, es decir, se puede ir del último registro al primero, etc. Además, si otro usuario modifica o borra un registro

-
- durante la lectura del RecordSet, éste se daría cuenta de ese cambio. No pasaría lo mismo si otro usuario añadiese un nuevo registro.
- adOpenDinamic (2): se puede mover en todas direcciones por el RecordSet. Si otro usuario añade, borra o modifica un registro durante la lectura del RecordSet, éste se daría cuenta del cambio.
 - AdOpenStatic (3): se puede mover en todas direcciones, pero RecordSet no detectará ningún cambio ajeno al usuario que lo esté utilizando.
4. Propiedad LockType: indica el nivel de acceso que se tiene sobre los registros, es decir, que se puede hacer con los datos de la base (leer, borrar, actualizar). Se puede utilizar los siguientes cerrojos:
- adLockReadOnly (1): sólo permite leer el registro, no se podrá modificar. Éste es el utilizado por defecto.
 - adLockPessimistic (2): se pueden modificar los registros. Se tendrá acceso a la base desde que se habrá la tabla hasta que se cierre.
 - adLockOptimistic (3): se podrán modificar los registros. Sólo se bloqueará la base de datos en el momento que se realicen los cambios.
 - adLockBatchOptimistic (4): se podrán modificar los registros, pero todos los cambios se harán en modo batch.
5. Propiedad MaxRecords: con esta propiedad se puede limitar el número de registros que se inserten en una consulta. Por ejemplo, se puede tener una tabla con miles de registros pero sólo interesa seleccionar una muestra de diez.
6. Método Open: una vez que se han configurado todas las propiedades que se necesitan del objeto RecordSet lo siguiente será abrirlo, es decir, llenarlo con los datos que se quiera conseguir. Como parámetro tendrá el nombre de la tabla que se quiera seleccionar o una consulta SQL. De momento, el más sencillo llamado será las llamadas directas a tablas. En el ejemplo siguiente se abrirá la tabla Alumnos:

```
Obj_RS.Open "Alumnos"
```

Al igual que con el objeto Connection, en el método Open del objeto RecordSet se pueden pasar otros parámetros para evitar así el tener que configurar las propiedades del objeto. Estos parámetros serán, por orden de inserción, el objeto Connection que se va a utilizar, el cursor y el cerrojo. La sentencia quedaría de la siguiente forma:

Obj_RS.Open "Alumnos", Obj_Conn,adOpenStatic,adLockOptimistic

7. Propiedad State: esta propiedad avisa si se ha abierto o no un objeto RecordSet devolviendo las constantes adStateOpen (&H00000001) cuando esté abierto o adStateClose (&H00000000) cuando esté cerrado.
8. Colección Fields: esta colección contendrá los campos que tienen un registro. Como cosas más destacables se tiene la posibilidad de conocer el valor de cada campo de la consulta:

Obj_RS.Fields (Numero_Campo) o Obj_RS.Fields (Nombre_Campo)

el nombre del campo:

Obj_RS.Fields (Numero_Campo).Name o
Obj_RS.Fields (Nombre_Campo).Name

y el número de campos del registro:

Obj_RS.Fields.Count

9. Propiedad RecordCount: esta propiedad devolverá el número de registros que contiene el RecordSet. Si se utiliza el cursor adOpenForwardOnly no se podrá utilizar esta propiedad ya que devolverá -1.
10. Propiedad Filter: esta propiedad seleccionará sólo un conjunto de registros. Pasándole una condición, se puede determinar los registros que se quieran ver. Por ejemplo, si se tiene una tabla de alumnos, a lo mejor sólo interesa ver a los usuarios con el nombre Pedro.

-
11. Propiedad PageSize: esta propiedad servirá para paginar resultados. Cuando se tienen listados demasiado grandes, interesa que salgan en páginas. Con esta propiedad se determina el número de registros que se mostrarán por cada página.
 12. Propiedad PageCount: una vez que se ha decidido cuantos registros va a tener cada página, esta propiedad dirá el número final de páginas que habrá.
 13. Propiedad AbsolutePage: esta propiedad llevará al usuario al primer registro de cada página. Por ejemplo, si se tienen páginas de cinco registros y asignados a esta propiedad el valor dos, lo llevará al registro número seis.
 14. Propiedad AbsolutePosition: con esta propiedad el usuario se podrá situar en un registro específico. Simplemente se tendrá que asignar a la propiedad la posición del registro que se quiera ver.
 15. Propiedad EOF: devolverá True cuando se haya sobrepasado el último registro del RecordSet.
 16. Propiedad BOF: devuelve True cuando se esté antes del primer registro del RecordSet.
 17. Método Move: con éste método el usuario se podrá mover un número determinado de registros. Se le dará al método un número positivo para moverse hacia delante registros y negativo para moverse hacia atrás. Ejemplo:

Obj_RS.Move 5 "Moverse 5 registros hacia delante".
 18. Método MoveFirst: este método conduce al primer registro. Esto no significa que la propiedad EOF sea True, ya que para esto se tendría que ir un registro más para atrás.
 19. Método MoveLast: este método conduce al último registro. Esto no significa que la propiedad EOF sea True, ya que para esto se tendría que ir un registro más adelante.
 20. Método MoveNext: el usuario se mueve un registro hacia delante.
 21. Método MovePrevious: el usuario se mueve un registro hacia atrás.

-
22. Método GetRows: devuelve un array con todos los registros.
23. Método Clone: devuelve un objeto RecordSet igual que el que se tiene. Se tendría que utilizar SET para recibir el resultado ya que se trata de un objeto.

SET Objt_RSCopia = Obj_RS.Clone

24. Método Delete: Borra el registro en el que el usuario esta posicionado.
25. Método AddNew: añade un nuevo registro al RecordSet. Este registro estará en blanco y a la espera que se le inserten datos.
26. Método Update: actualiza los cambios que se han hecho en el RecordSet, ya sea para añadir un registro o por modificar uno existente.
27. Método CancelUpdate: cancela las modificaciones que se hayan echo en el RecordSet ya sea para añadir un registro o modificar uno existente.
28. Método Close: cierra el RecordSet. Esto hace que desaparezca todos los datos del mismo. Una vez ejecutado este método se podrá volver a utilizar el RecordSet para otra selección.
29. Destrucción del objeto: se utilizará la misma sentencia que en el objeto Connection, es decir:

SET Obj_RS = Nothing

8.3 ¿Cómo se puede leer una base de datos mediante ASP?

El ASP nos ayuda a darle dinamismo a nuestros sitios con la construcción de páginas dinámicas. Una de sus cualidades más importantes, es su interacción con bases de datos.

En el siguiente taller leeremos los contenidos de una base de datos Access que contiene 3 campos: Nombre, Email y URL, con los que alimentaremos unas tablas para nuestras páginas.

Y para facilitar el entendimiento del ejemplo, se comentará directamente en el código ASP para una mejor explicación.

Recuerden que para comentar el código en ASP, se utiliza una comilla simple (').

8.4 Ejemplos de programas utilizando código ASP

El código ASP está denotado en renglones subrayados:

Ejemplo no.1

```
<HTML>
<HEAD><TITLE>Leyendo una base de datos de Access con ASP</TITLE></HEAD>
<BODY>
<TABLE BORDER="1">
<TD><B>Nombre<B></TD>
<TD><B>Email<B></TD>
<TD><B>Url<B></TD>
<%
```

Dim conexion, registros, basedatos

' El primer paso consiste en declarar algunas variables que nos serán de utilidad.

' De estas tres variables, utilizaremos la primera para nuestra conexión a la base de datos, la segunda para seleccionar los registros y la tercera para ingresar la dirección donde se encuentra nuestra base de datos.

```
basedatos = Server.MapPath("ejemplo.mdb")
```

' En la asignación anterior, la instrucción Server.Mappath nos da el Path completo en el servidor hacia el archivo ejemplo.mdb que se encuentra en el mismo directorio que el ejemplo.

' También se puede utilizar el path completo para definir esta variable que quedaría algo así:

```
' basedatos = "c:\inetpub\wwwroot\ejemplo.mdb"
```

```
Set conexion = Server.CreateObject("ADODB.Connection")
```

' Declaramos nuestra conexión a bases de datos.

```
conexion.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & basedatos & ".;"
```

' Abrimos la conexión a través del OLE Driver., indicándole la dirección de la base de datos por medio de la variable que declaramos algunas líneas antes.

' Así como utilizamos el OLE Driver, también pudimos utilizar una conexión por medio del ODBC y un DNS declarado en el sistema, aunque este Driver es el más rápido y recomendado para utilizar.

```
Set registros = conexion.Execute("SELECT * FROM Ejemplo ORDER BY Ejemplo.Nombre")
```

' A continuación, declaramos en registro los datos que estaremos seleccionando en la base de datos.

' Por medio de lenguaje SQL Seleccionamos todos los campos de la tabla Ejemplo, ordenándolos por el campo Nombre

Do While Not registros.EOF

' Para empezar a mostrar los registros, utilizamos un CICLO que se repetirá mientras no lleguemos al final de los registros encontrados.

' Luego de esto, iremos tomando los valores de la base de datos e imprimiéndolos dentro del HTML.

```
%>
<TR><TD><%= registros.Fields("Nombre").Value %></TD>
<TD><A HREF="mailto:<%= registros.Fields("Email").Value %>"><%=
registros.Fields("Email").Value %>
</A></TD>
<TD><A HREF="<%= registros.Fields("URL").Value %>"><%=
registros.Fields("URL").Value %></A></TD> </TR>
<%
registros.MoveNext
Loop
```

' Cada vez que se ejecuta el ciclo, le pedimos por medio de .MoveNext que pase al siguiente registro encontrado

' La instrucción Loop es la que indica hasta donde debe de repetirse el ciclo.

```
                registros.Close
Set registros = Nothing
conexion.Close
Set conexion = Nothing
```

'Algo muy importante que debemos recordar es cerrar nuestras conexiones, para no dejar la base de datos abierta.

' Para esto, primero la cerramos por medio de .Close y a continuación es recomendable declararles valores de "nada", para dejar las variables limpias.

```
%>
</TABLE>
</BODY>
</HTML>
```


El resultado final del código anterior se puede ver en la siguiente figura 17.

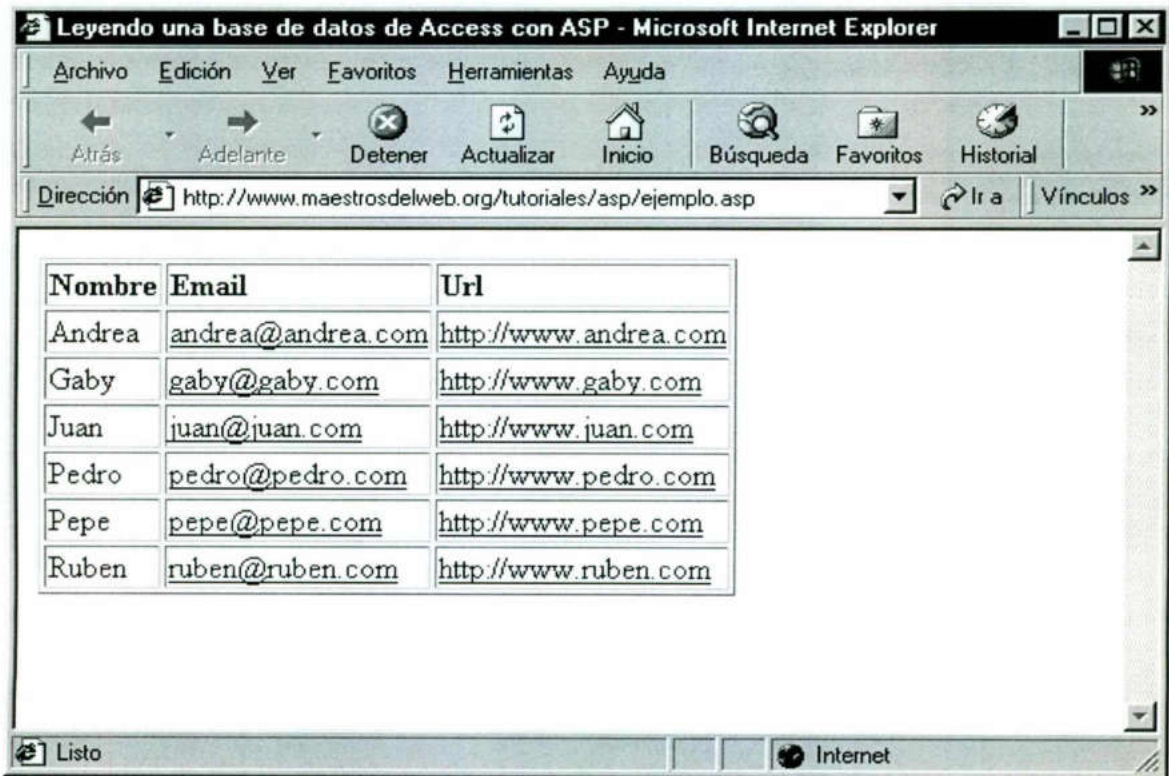


Figura 17. Vista final

Como se puede apreciar, el comunicarse con una base de datos no es tan complicado y nos permite crear páginas con contenidos que pueden estar actualizándose fácilmente por medio de cambios directos a la base de datos y no a nuestros archivos HTML.

Ejemplo no.2 " Selección de Fecha y Hora "

En ocasiones, las personas seleccionan una fecha y hora para poder comenzar algún tipo de procesamiento: reservas de hoteles, vehículos, vuelos, entradas a espectáculos; en informes de actividades, reservas de citas en laboratorios, de asistencia al médico, etc.

En este ejemplo se proporcionan cinco listas desplegables para poder seleccionar: día, año, hora y minuto. Cada una de las listas desplegables se sitúa en el valor correspondiente a la fecha del sistema que posee el equipo servidor cuando ejecuta la página ASP.

La página Web que se le presenta al usuario contiene listas desplegables con: 31 días, 12 meses, 30 años, 24 horas y 60 minutos. Para evitar que la página Web de tipo .htm tenga demasiado tamaño en el servidor, se ha optado por convertirla en .asp e incluir los bucles necesarios para generar todas las marcas <OPTION> de las listas desplegables.

La página Web que contiene el formulario se encuentra en el fichero

Ejemplo6-2 htm.asp

```
1. <HTML>
2.
3. <BODY>
4. <FONT SIZE="5"> SELECCIÓN DE FECHA Y HORA </FONT>
5. <FORM ACTION="Ejemplo6-2.asp" METHOD="POST">
6.
7. <TABLE>
8. <TR>
9. <TD> Día
10.   <SELECT NAME="DiaSeleccionado">
11.     <% For Día=1 to 31
12.       if Day(Date)=dia Then
13.         Seleccionado="SELECTED"
14.       Else
15.         Seleccionado=" "
16.       End if %>
17.     <OPTION <%=Seleccionado%> > <%=Dia%> </OPTION>
18.     <% Next %>
19.   </SELECT> </TD>
20.
21. </TD> Mes
22.   >SELECT NAME="MesSeleccionado">
23.     <% For Mes=1to 12
24.       if Month (Date)=Mes Then
```



```

25.   Seleccionado="SELECTED"
26.   Else
27.   Seleccionado=" "
28.   End If %>
29.   <OPTION <%=Seleccionado%> VALUE=<%=Mes%> > <%=MonthName (Mes) %>
    </OPTION>
30.   <% Next %>
31.   </SELECT> </TD>
32.
33.   <TD> Año
34.   <SELECT NAME="AnioSeleccionado">
35.   <% For Anio=2000 to 2030
36.   If Year(Date)=Anio Then
37.   Seleccionado="SELECTED"
38.   Else
39.   Seleccionado=" "
40.   End If %>
41.   <OPTION <%=Seleccionado%> > <%=Anio%> </OPTION>
42.   <% Next %>
43.   </SELECT> </TD>
44.
45.   <TD> Hora
46.   <SELECT NAME="HoraSeleccionada">
47.   <% For Hora=0 To 23
48.   If Hour(Time)= Hora Then
49.   Seleccionado="SELECTED"
50.   Else
51.   Seleccionado=" "
52.   End If %>
53.   <OPTION <%=Seleccionado%> > <%=Hora%> </OPTION>
54.   <% Next %>
55.   </SELECT> </TD>
56.
57.   <TD> Minuto
58.   <SELECT NAME="MinutoSeleccionado">
59.   <% For Minuto=0 To 59
60.   If Minute(Time)= Minuto Then
61.   Seleccionado="SELECTED"
62.   Else
63.   Seleccionado=" "
64.   End If %>
65.   <OPTION <%=Seleccionado%> > <%=Minuto%> </OPTION>
66.   <% Next %>
67.   </SELECT> </TD>
68.
69.   </TD> <INPUT TYPE="submit" VALUE="Enviar"> </TD> </TR>
70.
71. <TABLE>
72.
73. </FORM>
74. </BODY>
75. </HTML>

```

(Vea figura 18)

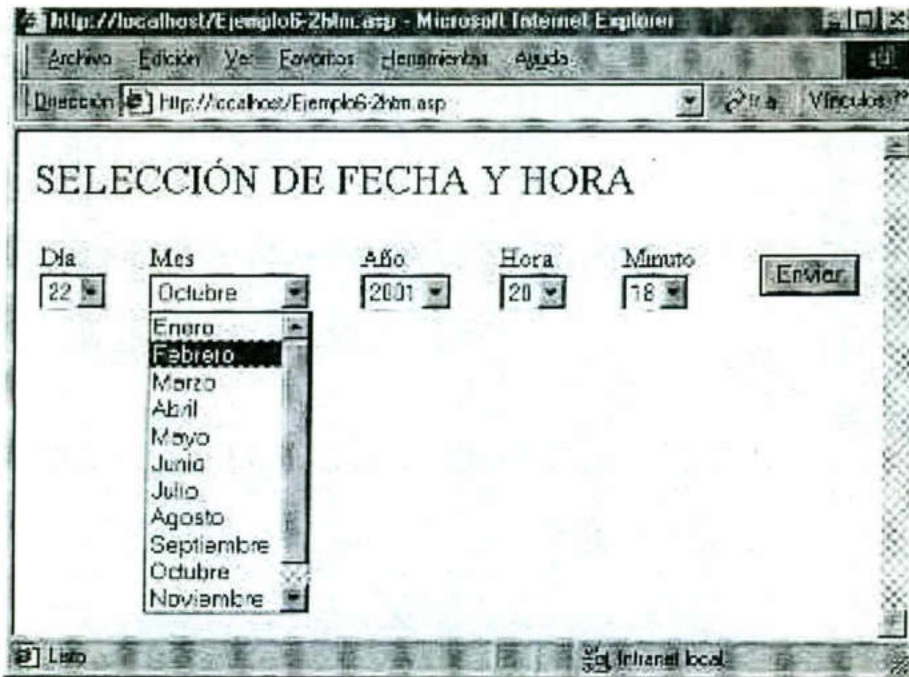


Figura 18. Código HTML

Las cinco listas desplegables se definen de la siguiente manera: día en la línea 10 con nombre DiaSeleccionado; mes en la línea 22 con nombre MesSeleccionado; año en la línea 34 con nombre AñoSeleccionado; hora en la línea 46 con nombre HoraSeleccionada; minuto en la línea 58 con nombre MinutoSeleccionado.

La creación de las opciones (<OPTION>) de cada lista desplegable se puede explicar refiriéndonos a una sola de ellas; por ejemplo, la primera lista corresponde a los días: las líneas 11 y 18 definen el bucle For que va a iterar de 1 a 31 días. En los demás casos la iteración se realizará del mes 1 al 12, de la hora 0 a 23, etc. Las líneas 12 a 16 determinan si el día actual (respecto a la fecha del servidor) coincide con el día marcado en cada iteración del bucle; si es así, se asigna el valor SELECTED a la variable Seleccionado. Esto servirá para presentar al usuario el día actual en la lista desplegable. La línea 17 escribe las marcas <OPTION>....</OPTION> y el valor del día entre ellas. Se escribirán tantas líneas de opción como iteraciones hace el bucle. Sólo una de las marcas <OPTION> de los meses, presenta una particularidad. En este caso, en lugar de presentarle valores numéricos al usuario, se rellena la lista desplegable con los nombres de los meses, pero ello se emplea MonthName(Mes). Para continuar trabajando

internamente con valores numéricos, se incluye el parámetro VALUE en la marca <OPTION>.

Los distintos elementos del formulario se han colocado en campos de una misma tabla con el fin de conseguir que las listas desplegables se visualicen horizontalmente en el navegador.

La página web que recoge y procesa los datos seleccionados por el usuario se codificará en el fichero

Ejemplo6-2.asp

```
1. <HTML>
2. <HEAD>
3. <TITLE> Selección de fecha y hora </title>
4. </HEAD>
5.
6.     <BODY>
7.     Fecha y hora seleccionados: <BR>
8.
9.     <% Dia=CInt(Request.Form("DiaSeleccionado")) %>
10.    <% Mes=CInt(Request.Form("MesSeleccionado")) %>
11.    <% Anio=CInt(Request.Form("AnioSeleccionado")) %>
12.    <% Fecha=DateSerial(Anio,Mes,Dia) %> </TD>
13.    <% If Mes=Month (fecha) Then %>
14.    <TABLE BORDER=1>
15.    <TR>
16.    <TD> <%=CStr(Dia) & " / " & CStr(Mes) & " / " & CStr(Anio) %></TD>
17.    <TD>
18.    <%=Request.Form("HoraSeleccionada")%> :
19.    <%=Request.Form("MinutoSeleccionado")%>
20.    <TD>
21.    <TD> <%=WeekDayName(WeekDay(Fecha)) %> </TD>
22.    </TR>
23.    </TABLE>
24.    <% Else %>
25.    La Fecha introducida es errónea
26.    <% End If %>
27.    </BODY>
28.    </HTML>
```

(Vea figura 19)

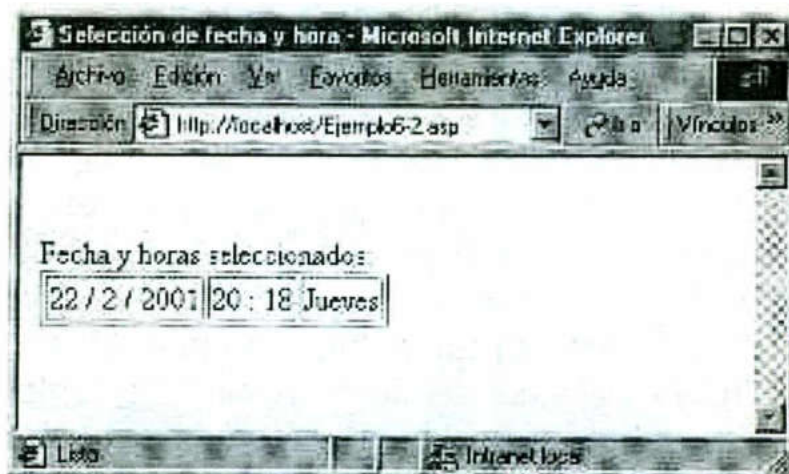


Figura 19. Código ASP

Las líneas 9,10, y 11 del ejemplo Ejemplo6-2.asp recoge el día, mes y año seleccionado por el usuario; con estos valores se obtiene la fecha como subtipo Date (línea 12). Para saber si el usuario ha seleccionado una fecha errónea se emplea el if situado en la línea 13 (junto a la 24 y la 26). En caso de que el día seleccionado se pase de los posibles días del mes escogido (ejemplo 31 de febrero), la función Date Serial situada en la línea 12 suma un mes a la fecha devuelta. De esta manera, se pregunta si el mes escogido coincide con el devuelto por DateSerial; si no es así, se visualiza un mensaje de error (línea 25).

Las líneas 16,18,19 y 21 se encargan de visualizar, en los campos de una tabla, la fecha, hora y día de la semana de los datos seleccionados por el usuario.

9. ¿ El acceso remoto a las bases de datos a través de Internet, beneficiará a alguien?

Efectivamente, si beneficia a alguien.

Ciertamente, la información que se tiene a la mano mediante una conexión a Internet es muy vasta y adecuada en la cuestión de economía.

Citando a frase anterior, se mostrarán algunos puntos en donde sobresale el factor económico.

1.- La Internet sea convertido en parte de nuestra forma de aprender, descubrir y disfrutar el mundo que nos rodea; siempre y cuando tengamos acceso a una computadora.

La mayoría de las personas requieren hoy en día cierta veracidad en la información que recibe del mundo que las rodea; para esto, las personas que requieren más información sobre un asunto que les interese, recurrían a fuentes escritas de información, tales como libros, revistas, periódicos, folletos, volantes y demás; o preferían escuchar la radio, a personas que tenían cierta información o comúnmente ver televisión.

Estas acciones eran hace unos 10 años. Pero con la llegada de unas computadoras personales a los hogares, la Internet tomo una demanda mayor de la esperada por expertos en estadística comercial. La Internet significo "entender " el mundo como siempre se había deseado: sin límites, sin barreras, sin marcas territoriales, sin credos, sin preferencias sexuales, sin razas, en fin , de todo una gama de restricciones que la misma sociedad imponen a todos sus miembros que somos cada uno de nosotros.

El beneficio económico vino cuando gente que ofrecía un servicio, quería vender algo, y simplemente quería llamar la atención; la Internet resulto un imán de grandes posibilidades de crecer económicamente sin restricciones.

Al ver este fiebre por la Internet, las grandes compañías que tenían sus negocios establecidos, quisieron entrar al cyber espacio para promover sus productos o servicios.

Poco después, las compañías que manejaban un rublo diferente a la comercial, tales como el gobierno, instituciones humanitarias e instituciones educativas, vieron la posibilidad de darse a conocer, recibir ayuda y promover su participación en la comunidad.

Como se ilustra arriba, la Internet resulto una opción rentable que beneficia a personas tanto a las dedicadas a el área de computación como a las que tienen otras perfiles académicos.

La mayoría de las compañías al manejar información para sus funciones y procesos corporativos, no pueden mostrar todo lo que hacen estas instituciones en la Internet. Así como hay información que requiere ser mostrada a todo el mundo, existe información que requiere ser clasificada como "confidencial".

La información confidencial para una empresa, se ha convertido en una "activo fijo" que requiere mantenimiento, uso y lo más importante actualización continua.

Esta información confidencial requiere ser clasificada, ordenada y acomodada de forma óptima dentro de cada una de las empresas. Para tal objetivo, se hace uso de sistemas de almacenamiento llamadas "Bases de Datos". Las bases de datos juegan un papel importante en la vida activa de una empresa lucrativa o no.

Las Bases de datos constituyen el punto focal para la toma de decisiones que los miembros de una empresa debe realizar. Sin ella, la toma de decisiones sería basado en las llamadas "corazonadas " y se estaría echando mano de la "Intuición" que bien se sabe que la dichos mecanismos resultando obsoletos y poco prácticos a la hora de determinar el rumbo de una empresa, ya que carecen se fundamentación y teorías probadas.

Las Bases de datos al contener información importante de una institución, necesitan seguridad para que este en las manos adecuadas. Para esto, se utilizan sistemas de seguridad que hacen esta función. Los sistemas de seguridad ayudan a las miembros de las empresas a tener el control total de la información para que este en las manos adecuadas.

Los sistemas de información tienen la función de permitir el acceso solo a personas que están autorizadas para manejar, usar o conocer dicha información.

De tal manera, que cuando una persona a través de una conexión a Internet necesite información sobre alguna institución en particular; dicha persona solo podrá ver, usar y manejar la información que los sistemas de seguridad le permitan.

Pero, cuando esta persona sea parte de la corporación y este en otra parte del mundo, por lo regular necesitará tener acceso a la información clasificada para cerrar algún negocio o promover sus productos. Físicamente, no podrá viajar hasta donde este la empresa, deberá acceder desde una computadora mediante una "interfaz" que le permita tener acceso a alguna base de datos de la compañía.

El beneficio económico es enorme, pues el vendedor o el promotor solo tendrá que conectarse a Internet para obtener información. Desafortunadamente, esta práctica dejó de tener uso cuando la información se volvió sinónimo de "dinero", ya que no solo interesaba a personas que pertenecían a la compañía, sino que también a terceros.

Así que el acceder a la información de una compañía, se convirtió para algunas personas en una fuente de prestigio, poder y dinero, que sin la ayuda del Internet sería imposible conseguirlos.

Por último, cabe resaltar que el beneficio económico surgió en varios panoramas donde tanto los dueños de la información como los que no pertenecían a la compañía, obtuvieran beneficios.

Bibliografía

Procesamiento de Bases de Datos.

Fundamentos, Diseño e Instrumentación

David M. Kroenke.

Editorial: Prentice Hall Hispanoamericana, S.A.

© 1996, Estado de México.

Capítulo 16. "Cliente Servidor y aplicaciones relacionales".

Pag. 483 - 540.

ASP Active Server Pages 3.0

Iniciación y Referencia

Jesus Bobadilla, Alejandro Alcocer, Luis Rodríguez-Manzaneque.

Editorial: Mc Graw Hill Interamericana de España S.A.U

© 2001. Madrid, España.

Capítulo 6 "Programas", pag.131-148

Capítulo 9, "Bases de datos" pag 249 - 278

Páginas Web de Referencia Bibliográfica

El texto completo de la sentencia puede ser consultado en

<http://www.onnet.es/aranzadi.pdf>

Para poder leer los archivos PDF se precisa el programa ACROBAT READER de Adobe, que puede obtenerse gratuitamente en

<http://www.adobe.com>

TODO DE ASP

<http://www.maestrosdelweb.org/editorial/tecnologias/asp.asp>

<http://www.maestrosdelweb.org/tutoriales/asp/database.asp>

Introducción a CGI

<http://www.maestrosdelweb.org/editorial/tecnologias/cgi.asp>

Autores de este trabajo: A.S.S. Javier Alberto Besso & A.S.S. Cristian Fabian Borghello

<http://www.dbinternet.com.ar/index.htm>