

**“DESARROLLO DE UNA INTERFAZ PARA CONSULTAS A NIVEL LOCAL Y
REMOTO DE UNA BASE DE DATOS EN AMBIENTE UNIX, ATRAVES DE
WINDOWS NT CON APLICACIÓN EN LA BIBLIOTECA DE LA CUTB”**

**MARCO ANTONIO CHICO RUIZ
INGRID JOHANNA GELIS MACHADO**

**CORPORACION UNIVERSITARIA TECNOLOGICA DE BOLIVAR
FACULTAD DE INGENIERIA DE SISTEMAS
CARTAGENA DE INDIAS, D.T.C**

2000

**“DESARROLLO DE UNA INTERFAZ PARA CONSULTAS A NIVEL LOCAL Y
REMOTO DE UNA BASE DE DATOS EN AMBIENTE UNIX, A TRAVES DE
WINDOWS NT CON APLICACIÓN EN LA BIBLIOTECA DE LA CUTB”**

**MARCO ANTONIO CHICO RUIZ
INGRID JOHANNA GELIS MACHADO**

**Proyecto de grado presentado como requisito para optar al título de
Ingeniero de Sistemas**

**Director
ALVARO CONTRERAS MONTES
Ingeniero Civil**

**CORPORACION UNIVERSITARIA TECNOLOGICA DE BOLIVAR
FACULTAD DE INGENIERIA DE SISTEMAS
CARTAGENA DE INDIAS, D.T.C**

2000

ARTICULO 105

La **Corporación Universitaria Tecnológica de Bolívar**, se reserva el derecho de propiedad intelectual de todos los trabajos de grado aprobado y no pueden ser explotados comercialmente sin autorización.

Nota de aceptación

Presidente del jurado

Jurado

Jurado

Cartagena, 15 de Diciembre de 1999

A Dios y a mis padres José de los R. y Meredith por su inmensurable paciencia y apoyo espiritual los cuales fueron puntos importantes para la realización de este proyecto.

A mis hermanos y demás familiares por su constante apoyo y comprensión.

A mis amigos, en especial a Ruth y Pedro por su ayuda incondicional.

Marco Antonio Chico Ruiz

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

Alvaro Contreras Montes, Ingeniero Civil y Director del proyecto de grado.

Pedro Hoyos Valle, Ingeniero de Sistemas, por su gran apoyo.

Gonzalo Garzón, Ingeniero de sistemas y asesor del proyecto, por su paciencia con nosotros.

William Narvaez, jefe de sistemas CURN, por su ayuda.

GLOSARIO

ADODB: Objetos de datos ActiveX, utilizados por las ASP para acceder los datos contenidos en el servidor.

API: Interfaz de programación de aplicaciones, las cuales facilitan a los programadores la conexión de una amplia gama de formatos de bases de datos.

ARCHIVOS PLANOS: Son archivos generados por manejadores de bases de datos en formato ASCII, los cuales están delimitados por unos caracteres especiales.

ASP: Servidor de páginas activas, las cuales pueden leer y escribir archivos de información (un servidor Web solo puede leerlos) y producir diferentes resultados cada vez que se corren.

ARCHIVO .ASP: archivo que contiene la consulta a ejecutar (SQL) previo a una ejecución.

ARCHIVO .HTM: Documentos con vínculos a otros documentos. Para presentar el otro documento en un vínculo.

BASE DE DATOS: Es un conjunto de archivos. Un archivo de base de datos que tiene una extensión de acuerdo al DBMS que se utilice y puede contener una o varias tablas, vistas conexiones con orígenes de datos remotos y procedimientos almacenados.

CABIB: Centro de atención bibliográfica para consultas a nivel local y Web en la C.U.T.B.

DAO: Objeto para acceso a bases de datos, son objetos no visibles para acceder a bases de datos ; es decir, no implementan ninguna interfaz propia de cara al usuario. Entre ellos podemos destacar Database, Table, Dynaset, Snapshot y QueryDef.

DATACONTROLS: La finalidad básica de un DataControl es acelerar las fases de desarrollo de mantenimientos de archivos. Poseen de por sí una interfaz con una apariencia física concreta, sobre la cual el usuario puede pulsar con el ratón y desencadenar acciones directamente sin que dichas acciones hayan sido implementadas por el programador.

DBMS: Sistema de gestión de base de datos, consiste en una colección de datos interrelacionados y una colección de programas para acceder a esos datos.

DRIVER: Recibe llamadas de la aplicación hacia funciones de la API de ODBC y traduce dichas llamadas al lenguaje nativo del servidor concreto.

DYNASETS: Conjunto de punteros de registros que apuntan a los datos específicos que existían en el momento en que se creó. Los cambios que se realizan en la información del Dynaset se reflejan en las tablas de la base de las que deriva la información así como en el mismo Dynaset. Estos cambios incluyen adición, edición, y eliminación de registros.

FUENTE DE DATOS: Es la asociación de un nombre de fuente con una información asociada en el servidor de la base de datos de destino. Esta información contiene puntos clave, como el sistema operativo en el que se encuentra el servidor, el driver utilizado, tipo de sistema gestor y la plataforma de red a la que necesitamos acceder para conectarnos a él.

INFORMIX: Es un DBMS que trabaja con Unix.

INTERFACE: Procedimiento que convierte unos archivos planos generados por el manejador de base de datos Informix a una base de datos Access en Windows NT.

INTERFAZ: Ver interface.

internet: Conjunto de interconexiones formado por todas las redes del mundo que utilizan TCP/IP.

SERVIDOR: Es el sistema gestor de la base de datos relacional al que nos queremos conectar.

SNAPSHOTS : Son objetos de acceso a bases de datos pertenecientes al subconjunto de los Recordsets, los cuales están especialmente indicados para aquellas operaciones en que solo se realicen lecturas. Su empleo en tareas repetitivas reduce mucho el tiempo de acceso a los datos.

ODBC: Es una especificación que permite a nuestro programa comunicarse con variedad de Bases de datos, ya sea una PC local o con una computadora central servidora. Su misión es proporcionar acceso a bases de datos relacionales utilizando como vehículo el lenguaje SQL.

PAGINAS DINAMICAS: Páginas Web que son personalizadas ó creadas para cada usuario en tiempo real.

SERVIDOR WEB: Los servidores Web envían y reciben e-mail, compilan datos y pueden cifrar mensajes, dependiendo del nivel de seguridad que soporte el servidor Web.

CONTENIDO

	Pags
INTRODUCCIÓN	1
1. HERRAMIENTAS USADAS PARA EL DESARROLLO DE CABIB.	4
1.1 Familia de protocolos TCP/IP.	4
1.1.1 Descripción del Modelo de estratificación por capas TCP/IP de Internet	6
1.2 ARQUITECTURA CLIENTE-SERVIDOR	8
1.2.1 Cliente / servidor	8
1.2.2 Componentes esenciales de la infraestructura Cliente-Servidor	12
1.2.2.1 Plataforma Operativa	12
1.2.2.2 Entorno de Desarrollo de Aplicaciones	12
1.2.2.3 Gestión de Sistemas	13
1.2.3 Características funcionales	14
1.2.4 Características físicas	16
1.2.5 Características lógicas	18
1.3 RELACION CON LOS SISTEMAS ABIERTOS	19
1.4 ASP	21
1.4.1 Ventajas de las Asp	24
1.4.2 Construcción de Páginas Activas	26
1.5 INTERNET	29

1.5.1 Escenarios de Internet	31
1.6 QUE ES UNA INTRANET	33
1.6.1 Escenarios de intranet	34
1.7 INTERNET EXPLORER	36
1.8 WINDOWS NT	37
1.9 INTERNET INFORMATION SERVER	38
1.9.1 Qué puede hacer con Internet Information Server	39
1.9.2 Cómo funciona Internet Information Server	39
1.9.3 Petición de dirección URL del explorador de Web	40
1.9.4 Respuesta del servidor Web	41
1.9.5 Páginas Estáticas	41
1.9.6 Páginas Dinámicas	42
2. PORQUE UNA INTERFAZ GRAFICA	44
2.1 Análisis y diseño de la interfaz gráfica	44
2.1.1 Análisis del sistema de la interfaz gráfica	45
2.1.2 Diagrama de flujo de la interfaz Gráfica	46
2.1.2.1 Sistema Administrativo C.U.T.B	47
2.1.2.2 Los Datos de Entrada	47
2.1.2.3 Conversión	49
2.1.2.4 Los datos de salida	50
2.1.3 Diccionario de datos	51
2.1.4 Datos de entrada	57

2.1.4.1	Base de datos previa	57
2.1.4.2	Archivo plano 1	57
2.1.4.3	Archivo plano 2	57
2.1.4.4	Archivo plano 3	57
2.1.4.5	Archivo plano 4	58
2.1.4.6	Archivo plano 5	58
2.1.4.7	Archivo plano 6	58
2.1.5	Conversión	58
2.1.5.1	Borrado de datos antiguos	59
2.1.5.2	Abrir archivos planos	59
2.1.5.3	Extracción de los datos	59
2.1.5.4	Almacenamiento de los datos	59
2.1.5.5	Generar nuevas tablas	59
2.1.6	Datos de salida	60
3.	ESTRUCTURA LÓGICA DE LA INTERFAZ	60
3.1	cuerpo parcial del procedimiento de Conversión (interfaz)	60
4.	AUTOMATIZACIÓN DE LA INTERFAZ	65
4.1	INTEGRACIÓN DE LA INTERFACE CON WIN AT	69
5.	ANÁLISIS Y DISEÑO DEL SISTEMA DE INFORMACIÓN BIBLIOGRÁFICA PARA CONSULTA DE LA C.U.T.B CABIB (CENTRO DE ATENCIÓN BIBLIOGRÁFICO PARA CONSULTA A NIVEL DEL WEB Y LOCAL)	71
6.	ESTRUCTURA LÓGICA DEL SISTEMA DE INFORMACIÓN BIBLIOGRÁFICA PARA CONSULTAS DE LA C.U.T.B	75

CABIB (CENTRO DE ATENCIÓN BIBLIOGRÁFICA PARA CONSULTAS A NIVEL DE WEB)

6.1 ESTRUCTURA DEL SITIO WEB DEL SISTEMA DE INFORMACIÓN BIBLIOGRÁFICA PARA CONSULTA EN EL WEB (CABIB)	76
6.1.1 Esquema del sitio Web de la biblioteca virtual	79
6.1.2 Vista previa del sitio Web de la biblioteca	80
6.2 ESTRUCTURA DE UNA PÁGINA WEB EN ASP	82
6.2.1 Estructura lógica de la página de consulta del sitio Web CABIB	83
6.2.2 Análisis del cuerpo de las páginas dinámicas del sitio Web CABIBWEB	84
6.2.2.1 Análisis de la página Resp_Aut.asp	84
6.2.2.2 Análisis de la página Resp_Gen2.asp	91
6.3 ESTRUCTURA FÍSICA DEL SITIO WEB DE LA BIBLIOTECA DE LA C.U.T.B CABIBWEB	95
7. CONCLUSIÓN	98
8. RECOMENDACIONES	101
BIBLIOGRAFÍA	102

LISTADO DE CUADROS

	Pags
Cuadro 1. Maestro	51
Cuadro 2. Autores	52
Cuadro3. Materia	53
Cuadro4. Título	53
Cuadro 5. Inventario	54
Cuadro 6. Autortitulo	54
Cuadro 7. Autornombre	55
Cuadro 8. Notas	55
Cuadro 9. Temas	56
Cuadro 10. Tematítulo	56

LISTA DE FIGURAS

	Pags
Figura 1: Esquema general de CABIB	
Figura 2: Arquitectura de red basada en el modelo OSI	5
Figura 3: Niveles de la arquitectura cliente / servidor	14
Figura 4: Funciones del os servidores	17
Figura 5: ASP	21
Figura 6: Creación de las ASP	26
Figura 7: Internet	29
Figura 8: Escenarios de Internet	31
Figura 9: Internet Information Server	32
Figura 10: servicio RAS	33
Figura 11: Escenario de INTRANET	35
Figura 12: IIS	38
Figura 13: Paginas Estáticas	42
Figura 14: Paginas Dinámicas	43
Figura 15: Esquema del Proyecto	46
Figura 16: Entrada de datos	48
Figura 17: Conversión	49
Figura 18: Salida de datos	50
Figura 19: Pantalla inicial de WinAT	70
Figura 20: Comando ADD de WinAT	71
Figura 21: Objetos	73

Figura 22: Relaciones	73
Figura 23: Diagrama de asociaciones	74
Figura 24: Diagrama específico	75
Figura 25: Esquema del Sitio Web	76
Figura 26: Pantalla del menú de CABIBWEB	80
Figura 27: Consulta bibliográfica por autor	81
Figura 28: Resultado de la consulta bibliográfica por autor	82
Figura 29: Estructura sitio web	95
Figura 30: Subdirectorío de páginas estáticas	96
Figura 31: Subdirectorío de páginas dinámicas	97

RESUMEN

La finalidad de este proyecto es realizar una interfaz gráfica que convierta las bases de datos de Informix en ambiente UNIX a una base de datos SQL en ambiente Windows NT con una cobertura al nivel de Web y aplicado al sistema bibliográfico de la Corporación Universitaria Tecnológica de Bolívar CUTB. Para cumplir con lo establecido, primero se analizó, diseñó y se implementó el procedimiento que realiza la conversión de los archivos planos generados por Informix a una base de datos en Access. Segundo, teniendo las bases de datos en Access, se analizó, diseñó e implementó el sistema de información sistematizado para consultas de la biblioteca de la Corporación Universitaria Tecnológica de Bolívar, con una serie de mejoras y con un ambiente agradable, manejable y flexible. Tercero, se investigó, consulto y aplicó una herramienta para integrar la base datos generada para la interfaz con el World Wide Web a través de paginas dinámicas implementadas en ASP.

El procedimiento de conversión (Interfaz) es automatizado a través de una herramienta de Windows Nt Server, el cual hace que la aplicación se ejecute periódicamente para actualizar las bases de datos en un horario establecido por el usuario (administrador del sistema).

Para la implementación del sistema de información de consulta bibliográfica de la Corporación Universitaria Tecnológica de Bolívar, se tuvo en cuenta todos las inquietudes y recomendaciones (requerimientos) de los empleados de la biblioteca, especialmente del jefe de biblioteca Roberto Lora, el cual nos expuso todo lo que quería con respecto al nuevo software.

Este software trae unas mejoras que hace que la información sea más fluida. Entre los requerimientos expuestos se encuentran: una mejor visibilidad de los datos, flexibilidad en el manejo del software, mostrar información valiosa que no arroja el sistema actual, entre otras. Además, el software de conversión genera otras tablas que son vitales en el modelo relacional para que el sistema de información de consulta bibliográfica funcione adecuadamente.

A partir de los requerimientos abordados se procedió a analizar y diseñar el modelo relacional, para posteriormente implementar la aplicación bajo la herramienta de programación Visual Basic 6.0.

De la conversión (interfaz) y la aplicación del software de consulta bibliográfica se puede decir que cumple con los requerimientos expuestos.

Ya teniendo la base datos convertida, se procede a integrarla con el WWW. Se creó el sitio Web de la biblioteca de la Corporación Universitaria Tecnológica de

Bolívar llamado Bvirtual (Biblioteca Virtual) con paginas dinámicas y no dinámicas (estáticas). Entre las paginas dinámicas se tienen las consultas que se particularizan en paginas de petición y respuesta con tecnología avanzada y entre las paginas estáticas se encuentran la pagina general, la pagina de bienvenida y el menú principal.

Las paginas dinámicas se implementaron basándose en la técnica de ASP (Active Server Pages) o Servidor de Paginas Activas, la cual viene con Internet Information Server versión 3.0 o superior, el cual es un servicio prestado por Windows Nt Server 4.0.

Las ASPs son un tipo especial de paginas Web que incorporan scripts que se ejecutan en el servidor, además, pueden combinar los lenguajes de Jscripts y Vscripts en una misma página dando paso a una nueva generación de aplicaciones basadas en Web.

Las paginas activas de Servidor son compatibles con cualquier browser y además sus respuestas son expuestas en código HTML puro.

Para nuestro proyecto utilizamos ASP para acceder bases de datos y este se realiza a través de ADO (Activex Data Object), que es un sistema que resultará

familiar a quienes hayan programado en visual Basic o en Access. El ADO trata los datos mediante objetos Recordset a través de ODBC.

Para que la ASP funciones tienen que estar en una carpeta en el sitio Web que permita ejecutar secuencias de comandos y tener la extensión “.Asp”. Estos archivos pueden ser generados por varias herramientas como por ejemplo Microsoft Access 97 o Access 2000, Microsoft FrontPage 98 o 2000, las cuales tiene facilidades para generar paginas ASP y luego pueden ser editadas y personalizadas, también pueden publicar datos de cualquier base de datos ODBC.

Con todo lo expuesto, queda sustentado que se cumple con el objetivo de implementar una interfaz que convierta las bases de datos Informix en ambiente UNIX a través de una herramienta implementada en plataforma Windows NT con una cobertura en el ámbito de Web.

Este es el esquema del proyecto:

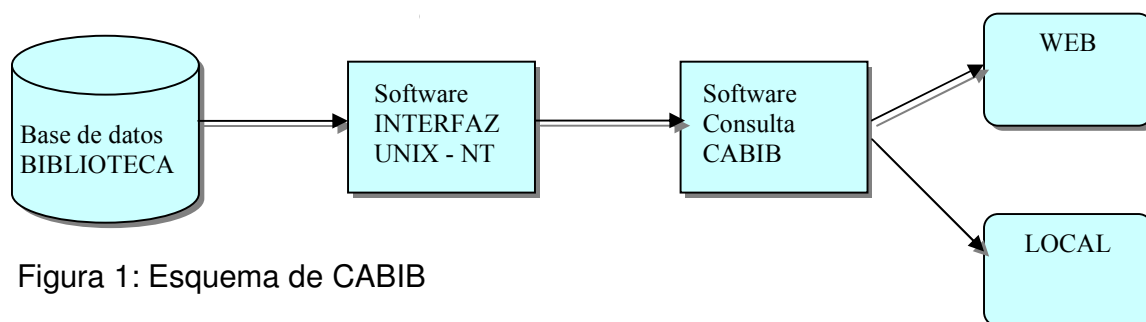


Figura 1: Esquema de CABIB

INTRODUCCION

La razón de ser del hombre en la vida es buscar los mecanismos ó la mejor forma de realizar y ejecutar tareas tendientes a una mejor y fácil manera de resolver cada uno de los problemas que se le presenten en sus labores diarias, para lo cual se ingenia los mecanismos para lograr todo esto;

Hoy día el más grande medio de comunicación existente es Internet(Red de redes), por lo cual la mayoría de las aplicaciones que se hacen para compartir recursos deben soportar tecnología internet, para que resulten lo más eficaz y aplicable posible a lo que se está viviendo en el momento en cuanto a tecnología se refiere.

Por tal motivo la Corporación Universitaria Tecnológica de Bolívar, como un gran ente educativo y formador de hombres que siempre están pensando en transformar, evolucionar y generar cambios importantes donde quiera que vayan, con el pasar del tiempo ha ido mejorando e innovando en todos sus sistemas de información en pro de estar al nivel de las mejores universidades del orbe, y que mejor forma de hacerlo que haciendo una reforma sustancial en uno de los mas importantes lugares con los que cualquier institución educativa debe contar como lo es la Biblioteca, el cual es un sitio que hasta el momento

solo podía ser visitado por el personal residente dentro de la Biblioteca en cuanto a consultas Bibliográficas se refiere, pero como hay que estar a la par con la tecnología y de hecho, la universidad forma personas para que logren estos cambios, de ahora en adelante, la biblioteca de la C.U.T.B, no será mas un sitio visitado por un número exiguo de personas localmente hablando, sino que por todas y cada una de las personas que lo quieran hacer a través del Web, hecho que definitivamente coloca a la C.U.T.B en un lugar más importante que el obtenido hasta el momento, en cuanto avances tecnológicos se refiere.

Para lograr estos cambios en el actual sistema de información bibliográfica se han tenido muchos aspectos relacionados como lo son las bases de datos en el Web y las bibliotecas virtuales.

El World Wide Web, o mas simple el Web, es un proyecto ambicioso cuya meta es ofrecer un consistente y simple interfaz a los recursos vastos de Internet. Los Navegadores y servidores del Web se comunican usando el Protocolo de Transferencia de Hipertexto (HTTP).

En este proyecto se integrarán estos dos aspectos tan importantes en nuestra vida cotidiana, teniendo así la posibilidad de crear documentos HTML dinámicos y que, den la posibilidad a los usuarios de manipular de mejor manera la información.

Actualmente existen proyectos que integran las páginas HTML y las Bases de Datos. Estos proyectos se utilizan mas comúnmente en formas de solicitud de servicios en las cuales se llenan con datos específicos del cliente y se actualiza una base de datos.

En este tipo de proyectos es necesario que las páginas del Web no solo desplieguen información estática sino que también sean capaces de hacer "Consultas" (Querys) a la base de datos. Esto se logra haciendo que el Servidor Web, en vez de que despliegue un documento, corra un programa con los datos del usuario, y sea este programa quien actualice la base de datos.

1. HERRAMIENTAS USADAS PARA EL DESARROLLO DE CABIB

1.1 FAMILIA DE PROTOCOLOS TCP/IP

Protocolos: Establecen una descripción formal de los formatos que deberán presentar los mensajes para poder ser intercambiados por equipos de cómputo; además definen las reglas que ellos deben seguir para lograrlo.

Los protocolos están presentes en todas las etapas necesarias para establecer una comunicación entre equipos de cómputo, desde aquellas de más bajo nivel (e.g. la transmisión de flujos de bits a un medio físico) hasta aquellas de más alto nivel (e.g. el compartir o transferir información desde una computadora a otra en la red).

Tomando al modelo OSI (Open Systems Interconnection) como referencia podemos afirmar que para cada capa o nivel que él define existen uno o más protocolos interactuando. Los protocolos son entre pares (peer-to-peer), es decir, un protocolo de algún nivel dialoga con el protocolo del mismo nivel en la computadora remota.

Las Siete Capas del Modelo OSI y sus Funciones Principales

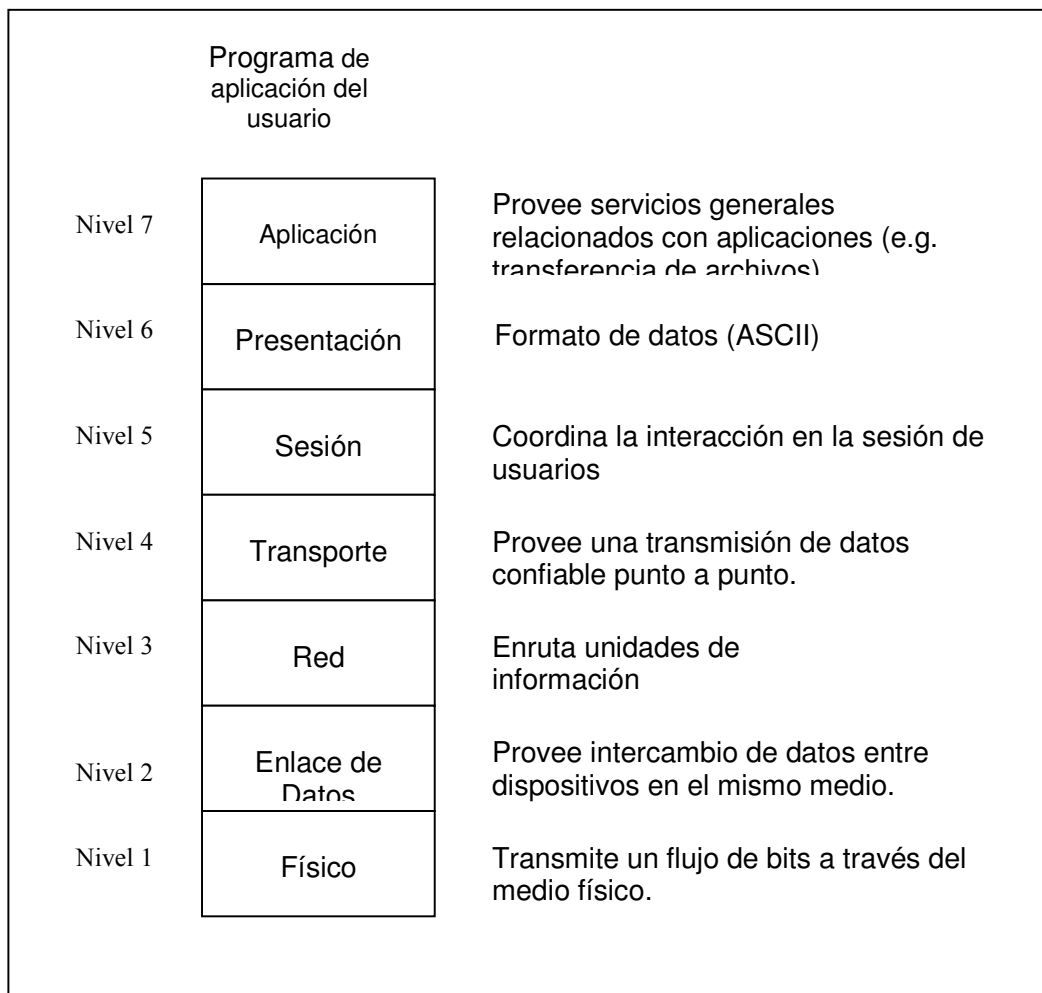


Figura 2: *Arquitectura de red basada en el modelo OSI*

1.1.1 Descripción del Modelo de estratificación por capas TCP/IP de Internet

Capa de Aplicación. Invoca programas que acceden servicios en la red. Interactúan con uno o más protocolos de transporte para enviar o recibir datos, en forma de mensajes o bien en forma de flujos de bytes.

Capa de Transporte. Provee comunicación extremo a extremo desde un programa de aplicación a otro. Regula el flujo de información. Puede proveer un transporte confiable asegurándose que los datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente de tal manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota, esto lo hace añadiendo identificadores de cada una de las aplicaciones. alteraciones durante su transmisión.

Capa Internet. Controla la comunicación entre un equipo y otro, decide qué rutas deben seguir los paquetes de información para alcanzar su destino. Conformar los paquetes IP que serán enviados por la capa inferior. Desencapsula los paquetes recibidos pasando a la capa superior la información dirigida a una aplicación.

Capa de Interface de Red. Emite al medio físico los flujos de bit y recibe los que de él provienen. Consiste en los manejadores de los dispositivos que se conectan al medio de transmisión.

A partir de estos modelos para la comunicación se da un gran paso fundamental para la interconexión de sistemas abiertos y de hecho base fundamental para nuestra investigación.

Por otra parte podemos afirmar que las capas del modelo TCP/IP, especialmente la capa de Interface de red, son base fundamental de apoyo para nuestro proyecto, en la medida en que en esta capa se efectúan los procesos de aceptar los datagramas IP y transmitirlos hacia una red específica. Esta interfaz de red consiste en un dispositivo controlador (por ejemplo , cuando la red es de área local a la que las máquinas están conectadas directamente) ó un completo subsistema que utiliza un protocolo de enlace de datos propio (por ejemplo, cuando la red consiste de conmutadores de paquetes que se comunican con anfitriones utilizando HDLC).

1.2 Arquitectura Cliente - Servidor

1.2.1 Cliente/Servidor. El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos estos recursos de máquina, de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspara a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor, al proceso que responde a las solicitudes.

Es el modelo de interacción más común entre aplicaciones en una red. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo.

Los principales componentes del esquema cliente/servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones.

En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los Clientes interactúan con el usuario, usualmente en forma gráfica.

Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz del usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Los Servidores proporcionan un servicio al cliente y devuelven los resultados.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

Gestión de periféricos compartidos.

Control de accesos concurrentes a bases de datos compartidas.

Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas Cliente/Servidor actuales, se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Entre las principales características de la arquitectura cliente / servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Como ejemplos de servidores pueden citarse servidores de ventanas como X-Windows, servidores para el manejo de bases de datos (como los servidores de SQL), servidores de diseño y manufactura asistidos por computadora, etc.

1.2.2 Componentes esenciales de la infraestructura Cliente/Servidor

Una infraestructura Cliente/Servidor consta de tres componentes esenciales, todos ellos de igual importancia y estrechamente ligados:

1.2.2.1 Plataforma Operativa. La plataforma deberá soportar todos los modelos de distribución Cliente/Servidor, todos los servicios de comunicación, y deberá utilizar, preferentemente, componentes estándar de la industria para los servicios de distribución. Los desarrollos propios deben coexistir con las aplicaciones estándar y su integración deberá ser imperceptible para el usuario. Igualmente, podrán acomodarse programas escritos utilizando diferentes tecnologías y herramientas.

1.2.2.2 Entorno de Desarrollo de Aplicaciones. Debe elegirse después de la plataforma operativa.

Aunque es conveniente evitar la proliferación de herramientas de desarrollo, se garantizará que el enlace entre éstas y el middleware no sea excesivamente rígido. Será posible utilizar diferentes herramientas para desarrollar partes de una aplicación. Un entorno de aplicación incremental, debe posibilitar la coexistencia de procesos cliente y servidor desarrollados con distintos lenguajes de programación y/o herramientas, así como utilizar distintas

tecnologías (por ejemplo, lenguaje procedural, lenguaje orientado a objetos, multimedia), y que han sido puestas en explotación en distintos momentos del tiempo.

1.2.2.3 Gestión de Sistemas. Estas funciones aumentan considerablemente el costo de una solución, pero no se pueden evitar. Siempre deben adaptarse a las necesidades de la organización, y al decidir la plataforma operativa y el entorno de desarrollo.

1.2.3 Características funcionales. Esta arquitectura se puede clasificar en cinco niveles, según las funciones que asumen el cliente y el servidor, tal y como se puede ver en el siguiente diagrama:

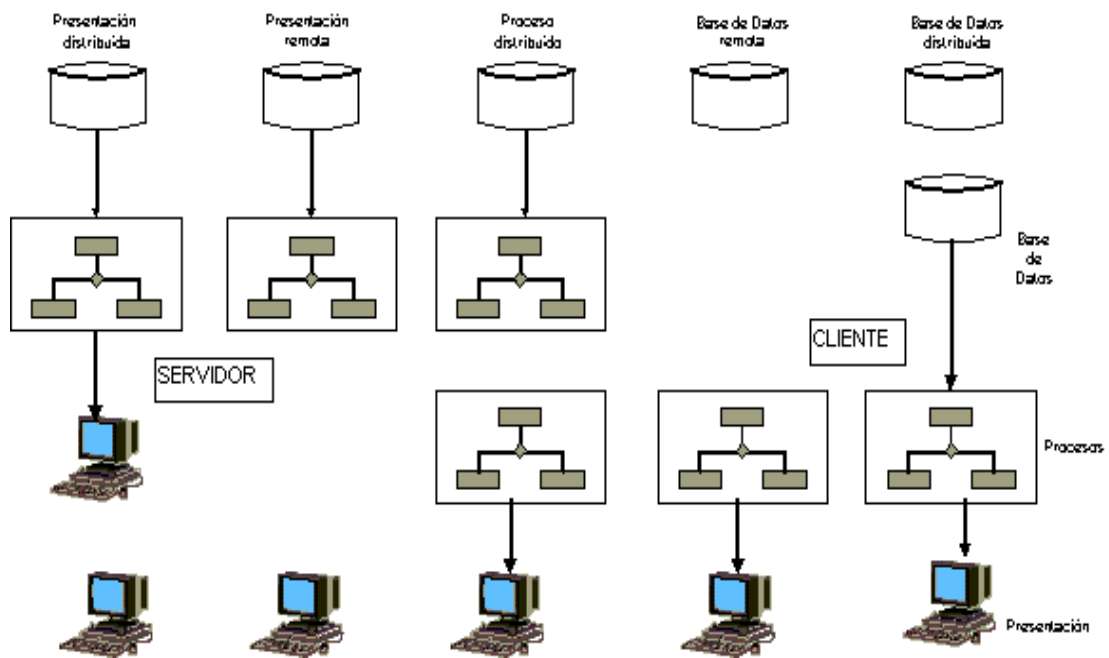


Figura 3: Niveles de la arquitectura cliente servidor.

En el *primer nivel* el cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor, dedicados a esta tarea. Dicha distribución se realiza mediante el uso de productos para el "maquillaje" de las pantallas del mainframe. Esta técnica no exige el cambio en las aplicaciones orientadas a terminales, pero dificulta su mantenimiento.

Además, el servidor ejecuta todos los procesos y almacena la totalidad de los datos. En este caso se dice que hay una presentación distribuida o embellecimiento.

En el *segundo nivel*, la aplicación está soportada directamente por el servidor, excepto la presentación que es totalmente remota y reside en el cliente. Los terminales del cliente soportan la captura de datos, incluyendo una validación parcial de los mismos y una presentación de las consultas. En este caso se dice que hay una presentación remota.

En el *tercer nivel*, la lógica de los procesos se divide entre los distintos componentes del cliente y del servidor. El diseñador de la aplicación debe definir los servicios y las interfaces del sistema de información, de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos, que es responsabilidad exclusiva del servidor. En este tipo de situaciones se dice que hay un proceso distribuido o cooperativo.

En el *cuarto nivel* el cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada. En esta situación se dice que hay una gestión de datos remota.

En el *quinto* y último nivel, el reparto de tareas es como en el anterior y además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos, están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones. En este nivel se da lo que se conoce como bases de datos distribuidas.

1.2.4 Características físicas. El diagrama del punto anterior da una idea de la estructura física de conexión entre las distintas partes que componen una arquitectura cliente / servidor. La idea principal consiste en aprovechar la potencia de los ordenadores personales para realizar, sobre todo, los servicios de presentación y, según el nivel, algunos procesos o incluso algún acceso a datos locales. De esta forma se descarga al servidor de ciertas tareas para que pueda realizar otras más rápidamente.

También existe una plataforma de servidores que sustituye al ordenador central tradicional y que da servicio a los clientes autorizados. Incluso a veces el antiguo ordenador central se integra en dicha plataforma como un servidor más. Estos servidores suelen estar especializados por funciones (seguridad, cálculo, bases de datos, comunicaciones, etc.), aunque, dependiendo de las dimensiones de la instalación se pueden reunir en un servidor una o varias de estas funciones.

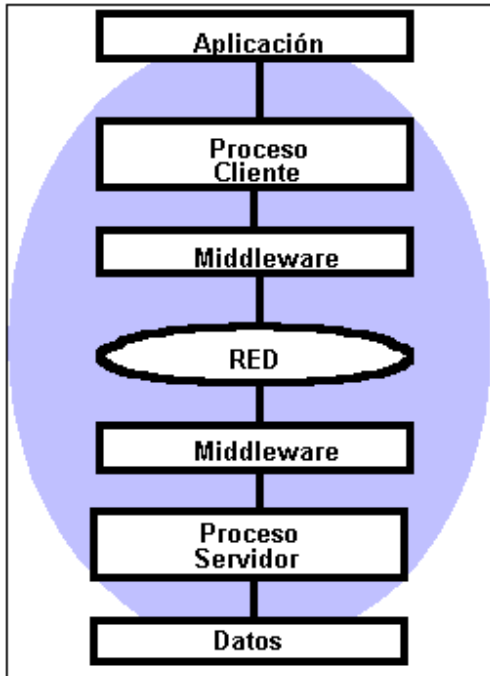


Figura 4: Funciones de los servidores

Las unidades de almacenamiento masivo en esta arquitectura, se caracterizan por incorporar elementos de protección que evitan la pérdida de datos y permiten multitud de accesos simultáneos (alta velocidad, niveles RAID, etc.).

Para la comunicación de todos estos elementos se emplea un sistema de red que se encarga de transmitir la información entre clientes y servidores. Físicamente consiste en un cableado (coaxial, par trenzado, fibra óptica, etc.) o en conexiones mediante señales de radio o infrarrojas, dependiendo de que la red sea local (LAN o RAL), metropolitana (MAN) o de área extensa (WAN).

Para la comunicación de los procesos con la red se emplea un tipo de equipo lógico denominado middleware que controla las conversaciones. Su función es independizar ambos procesos (cliente y servidor). La interfaz que presenta es la estándar de los servicios de red, hace que los procesos "piensen" en todo momento que se están comunicando con una red.

1.2.5 Características lógicas. Una de las principales aportaciones de esta arquitectura a los sistemas de información, es la interfaz gráfica de usuario. Gracias a ella se dispone de un manejo más fácil e intuitivo de las aplicaciones mediante el uso de un dispositivo tipo ratón. En esta arquitectura los datos se presentan, editan y validan en la parte de la aplicación cliente.

En cuanto a los datos, cabe señalar que en la arquitectura cliente / servidor se evitan las duplicidades (copias y comparaciones de datos), teniendo siempre una imagen única y correcta de los mismos, disponible en línea para su uso inmediato.

Todo esto tiene como fin que el usuario de un sistema de información soportado por una arquitectura cliente / servidor, trabaje desde su estación de trabajo con distintos datos y aplicaciones, sin importarle dónde están o dónde se ejecuta cada uno de ellos.

1.3 RELACIÓN CON LOS SISTEMAS ABIERTOS.

Las arquitecturas cliente / servidor se asocian a menudo con los sistemas abiertos, aunque muchas veces no hay una relación directa entre ellos. De hecho, muchos sistemas cliente / servidor se pueden aplicar en entornos propietarios.

En estos entornos, el equipo físico y el lógico están diseñados para trabajar conjuntamente, por lo que, en ocasiones, se pueden realizar aplicaciones cliente / servidor de forma más sencilla y fiable que en los entornos que contienen plataformas heterogéneas.

El problema surge de que los entornos propietarios ligan al usuario con un suministrador en concreto, que puede ofrecer servicios caros y limitados. La independencia del suministrador que ofrecen los entornos de sistemas abiertos, crea una competencia que origina mayor calidad a un menor precio.

Pero, por otra parte, debido a la filosofía modular de los sistemas cliente / servidor, éstos se utilizan muchas veces en entornos de diferentes suministradores, adecuando cada máquina del sistema a las necesidades de las tareas que realizan. Esta tendencia está fomentando el crecimiento de las

interfaces gráficas de usuario, de las bases de datos y del software de interconexión.

Debido a esto, se puede afirmar que los entornos cliente / servidor facilitan el movimiento hacia los sistemas abiertos. Utilizando este tipo de entornos, las organizaciones cambian sus viejos equipos por nuevas máquinas que pueden conectar a la red de clientes y servidores.

Los suministradores, por su parte, basan uno de los puntos clave de sus herramientas cliente / servidor en la interoperabilidad.

1.4 LAS ASP (PÁGINAS DE SERVIDOR ACTIVO)

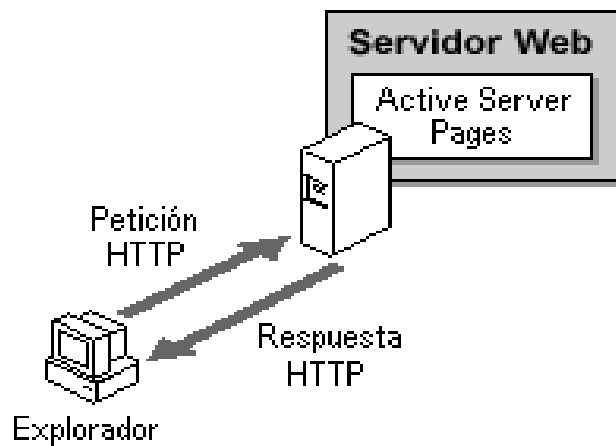


Figura 5: ASP

Active Server Pages (ASP) es el entorno de ejecución "lado - servidor" de Internet Information Server (IIS) 3.0, que posibilita la ejecución de "Scripts ActiveX" y componentes "ActiveX Server" en el servidor. Con la combinación de scripts y componentes, las organizaciones pueden crear fácilmente contenido dinámico y potentes aplicaciones basadas en Web (aplicaciones para Internet).

Con las ASP, IIS 3.0 además de ser un potente "servidor de información Web", se convierte en un "servidor de aplicaciones Web".

Las páginas tienen un Contenido Dinámico, es decir, son las páginas Web que son personalizadas o creadas para cada usuario "en tiempo real", basándose

en las acciones y/o peticiones del usuario. Por ejemplo, a los nuevos visitantes de su servidor Web les pueden ser mostrados diferentes mensajes de bienvenida, que les lleven a páginas o contenido on-line desde las que puedan realizar consultas a bases de datos, de tal forma que los clientes siempre verán la información más actual disponible.

También las Asp son usadas para crear aplicaciones Web "front-end" (lado cliente) basadas en soluciones de negocio existentes, o para crear aplicaciones Web completamente nuevas. Dado que ASP proporciona un entorno de desarrollo abierto, con soporte para VBScript y JScript, las organizaciones pueden mantener y rentabilizar las inversiones ya realizadas en estos "lenguajes de scripting".

Por otro lado también permiten el desarrollo de una nueva generación de aplicaciones basadas en Web, incluyendo la extensión de servicios al Cliente y servicios de Ventas a servidores Web, proporcionando acceso a bases de datos y aplicaciones corporativas a cualquier Browser (Explorador o Navegador de Internet) en Internet o en una Intranet. Estas páginas pueden trabajar con cualquier browser. La salida de un fichero. ASP es HTML "simple y plano", por lo que el contenido es adaptado a las capacidades del cliente.

También mantiene el "estado" de la conexión, es decir, que le permite definir variables de aplicación y de sesión que pueden ser extendidas a múltiples páginas y ser manipuladas en distintos puntos/usos del servidor Web. Esta funcionalidad puede ser aplicada a puntos tan sencillos como recordar un nombre de usuario, y es necesaria en aplicaciones de compra en Internet para seguir el listado de productos seleccionados.

Las páginas de servidor activo son un entorno seguro, puesto que es un componente de Internet Information Server, y por tanto usa la seguridad integrada de Windows NT. Los ficheros ASP pueden ser fácilmente restringidos a un grupo de usuarios usando autenticación segura de Windows NT, autenticación Web básica o certificación "lado cliente". Además, para seguridad adicional, todas las comunicaciones entre cliente y servidor pueden ser aseguradas con SSL (Secure Socket Layer).

Una aplicación de las páginas de servidor activo puede integrarse con cualquier base de datos compatible-ODBC, incluyendo bases de datos Microsoft SQL Server, Oracle, Sybase, Informix, DB2, Access, FoxPro, dBase, Clíper, etc. Cualquier aplicación OLE 2, tal como Lotus Notes, Microsoft Excel o Microsoft Word, también pueden ser manejadas para acceder o procesar información.

Tales páginas son una forma más sencilla de que su servidor acceda a información no directamente accesible por el cliente (tal y como una base de datos SQL), actuando como una pasarela entre el cliente y el servidor para producir la información que el cliente puede ver y usar.

Debido a que se ejecuta en el mismo proceso que el servidor Web, gestionan por tanto las peticiones de clientes de forma más rápida y eficiente. Es mucho más sencillo desarrollar contenido dinámico y aplicaciones Web con Active Server Pages.

1.4.1 Ventajas de las asp. Las ASP son la apuesta tecnológica de Microsoft en el campo de las aplicaciones de servidor en Internet/Intranet/Extranet. Por lo que, Microsoft trata de que la programación de servidores de Internet sea una prolongación natural de sus otros sistemas de programación, con especial énfasis en Visual Basic, que es el más popular.

Las ventajas de las ASP podríamos resumirlas en lo siguiente:

1. Son de aplicaciones que van a correr en servidores Windows NT exclusivamente, por lo que pueden ser probadas localmente en un PC con sistemas operativos de Microsoft.

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

3. También puede utilizarse un acceso a datos más tradicional basado en ODBC.

4. Ofrece persistencia de datos entre distintas ejecuciones de una misma página.

Esto es, nos evitamos la utilización de cookies o campos type = hidden para mantener variables entre distintas ejecuciones de un CGI. Con las ASP disponemos de propiedades del objeto Session a las que les podemos asignar valores permanentes.

1.4.2 Construcción de Páginas Activas

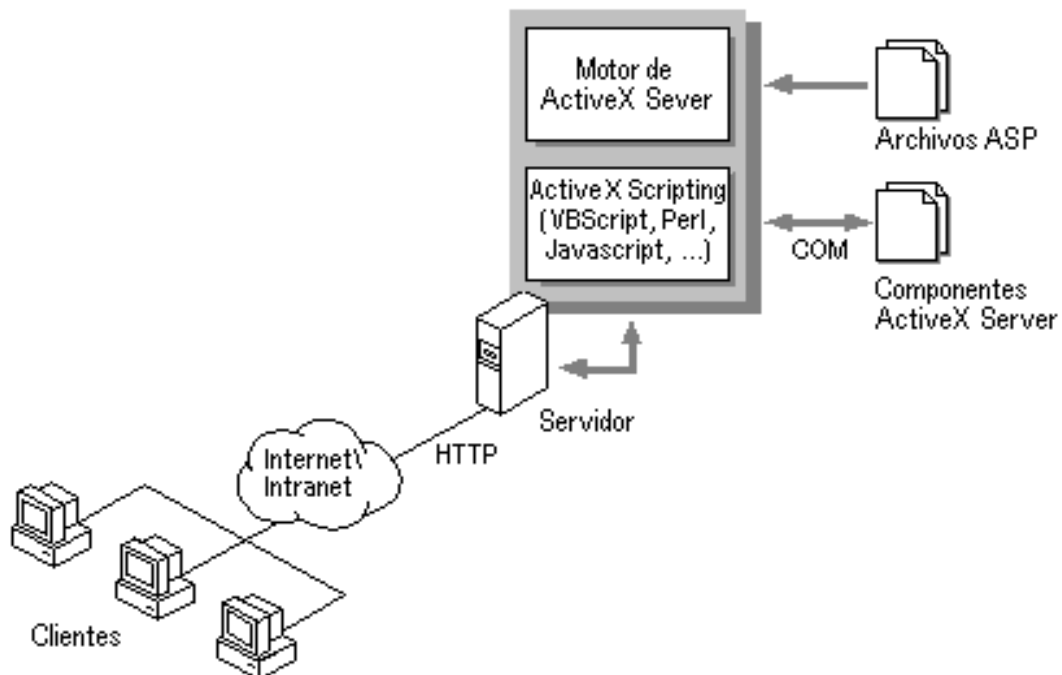


Figura 6: Creación de las ASP

La creación de nuestro sitio de Registro está basado en el uso de Páginas Activas. Estas páginas combinan programación que define el comportamiento, secuencia e interacción con el usuario.

La definición del comportamiento, interfaz con Windows NT e interfaz con bases de datos, se realiza en el servidor de Windows NT donde se encuentra ejecutando nuestro sitio de registro. Las páginas activas hacen uso de lenguajes de scripting para lograr esta interacción. Los lenguajes de Scripting utilizados por default en el IIS son VBScript y JavaScript. Se ha elegido

VBScript en la creación del código que se ejecuta en el servidor de IIS por que ofrece mejor funcionalidad, y tiene una interfaz más limpia y clara con componentes COM de los que hacemos uso dentro de nuestro sistema de Registro.

Una vez interpretado y ejecutado el código de VBScript de la Página Activa en el servidor de IIS, se envía una respuesta al explorador de la computadora cliente que estableció la comunicación con el servidor. La respuesta regresa en formato HTML, que contiene los resultados producidos por la página activa. Por ejemplo, si se solicita la página Default.asp, una vez interpretado el código y obtenido resultados por parte del IIS, se enviarán éstos en una hoja llamada Default.htm, que es la hoja que será vista por el usuario en su explorador.

Dentro de la hoja HTML producida, se realizará la interacción con el usuario. Ésta, representa la solicitud de información que se le pide al usuario. Él proporcionará la información y ejecutará un comando de envío nuevamente hacia el servidor de IIS, el cuál procesará su información dentro de otra hoja ASP y enviará una respuesta con resultados en forma de una hoja HTML.

En el proceso de solicitud de información será necesario validar información que el usuario proporcione. Aquí se involucra la programación de reglas que permitan validar los datos del usuario. De esta manera, se programan rutinas de

validación utilizando el lenguaje que Internet Explorer, soporte. La elección de lenguajes como (VBScript o JavaScript) depende del tipo de usuario que obtendrá acceso a nuestro sitio. Como una regla, se eligió la utilización de JavaScript, con este lenguaje, crearemos las rutinas de validación de información.

Este código puede ser ejecutado en la computadora del usuario, para lograr un uso más eficiente de los recursos. Con lo anterior, estamos evitando mandar la información al servidor sin antes validarla.

Si no se valida la información en la computadora del cliente, antes de enviarla, el servidor tendrá que hacer dicha validación y en caso de error manipularlo adecuadamente enviando una solicitud de verificación hacia el usuario, este proceso se puede volver lento.

Con la validación en la computadora cliente, se hace un uso más efectivo del tiempo del usuario a quien le ofrecemos el servicio.

1.5 INTERNET

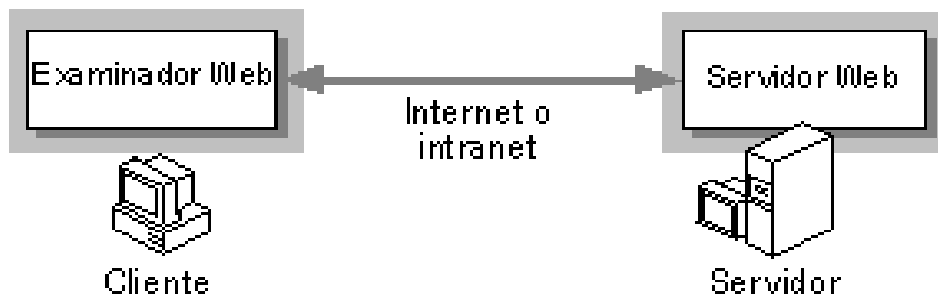


Figura 7: Internet

Un fenómeno que no se puede dejar de considerar es el crecimiento permanente de Internet.

Actualmente se utiliza para un conjunto de propósitos (correo electrónico, transferencia de archivos, WWW). La disponibilidad de los WWW, ha modificado mucho las cosas y los cambios mayores aún están por producirse.

Internet es una red mundial de equipos que se comunican usando un lenguaje común. Es similar al sistema telefónico internacional: nadie posee ni controla todo el sistema, pero está conectado de tal manera que hace que funcione como una red muy grande.

World Wide Web (WWW o simplemente Web) le ofrece una interfaz gráfica y sencilla para recorrer y consultar los documentos de Internet. Dichos documentos, así como los vínculos entre ellos, componen una red o “Web” de información.

Los archivos o páginas de Web están interconectados. Para conectarse a otras páginas puede hacer clic en el texto o en los gráficos especiales, que se llaman hipervínculos.

Las páginas pueden contener texto, imágenes, películas, sonidos, casi cualquier cosa. Estas páginas se pueden encontrar en equipos situados en cualquier parte del mundo. Cuando se conecta con Web, dispone del mismo acceso a la información en todo el mundo.

Los hipervínculos son palabras o gráficos que tienen direcciones de Web incrustadas en ellos.

Haciendo clic en un hipervínculo se salta a una página concreta de un determinado sitio Web. Los hipervínculos se identifican fácilmente: el texto de los hipervínculos suele tener un color distinto que el resto del texto de una página de Web y los gráficos de los hipervínculos tienen normalmente un borde de color.

Cada página de Web, incluida la página principal de un sitio Web, tiene una dirección única llamada dirección URL; por ejemplo, <http://www.microsoft.com/home.htm>. La dirección URL especifica el nombre del equipo en el que se almacena la página y su ruta de acceso exacta.

1.5.1 Escenarios de Internet. Internet Information Server puede funcionar en Internet como simple servidor Web dedicado (ver figura 8).

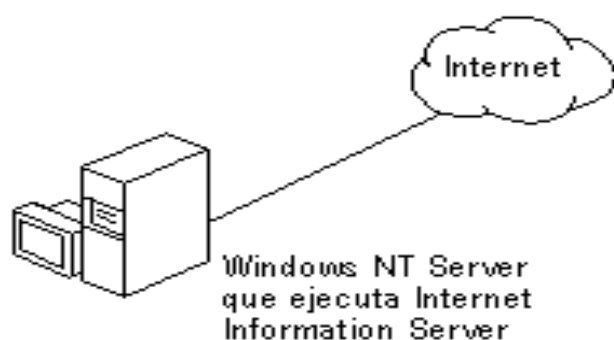


Figura 8: Escenarios de Internet

En organizaciones mayores puede proporcionar acceso desde su red interna a Internet Information Server, permitiendo que los empleados exploren el servidor o utilicen herramientas de creación, como Microsoft FrontPage para crear contenido para su servidor (ver figura 9).

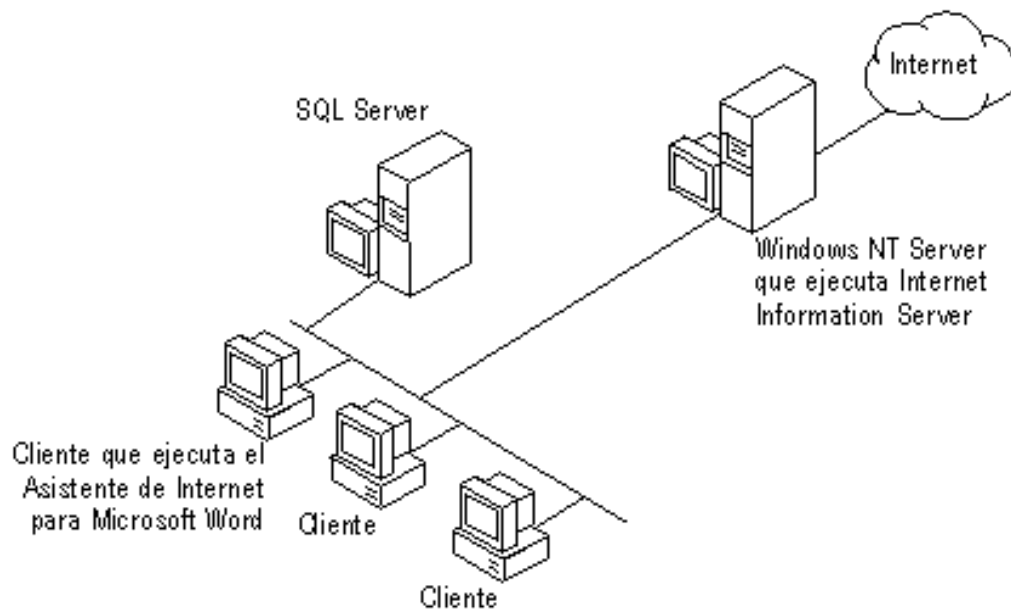


Figura 9: Internet Information Server

La integración de Internet Information Server con todos los servicios de Windows NT puede crear también servidores con varias funciones. Por ejemplo, una organización con oficinas en distintas partes del mundo puede usar Internet Information Server para proporcionar comunicaciones entre las distintas oficinas, con la flexibilidad añadida del acceso a Internet. Incluso puede agregar RAS a Internet Information Server para ofrecer acceso telefónico a su intranet o a Internet (ver figura 10).

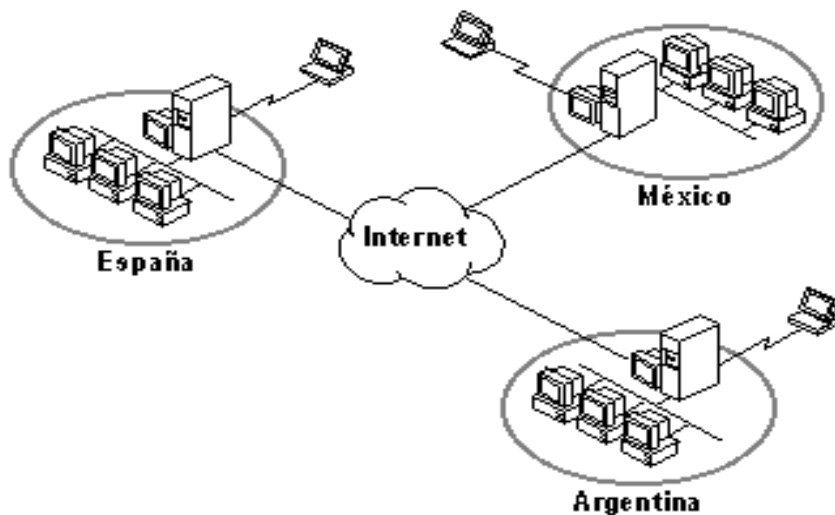


Figura 10: Servicio RAS

Nota: Muchos escenarios de conexión con Internet necesitan encaminadores o dispositivos de seguridad de terceros que filtren los paquetes de la red entre su equipo e Internet. Los encaminadores y los dispositivos de seguridad no se muestran en las ilustraciones anteriores.

1.6 QUÉ ES UNA INTRANET.

En esta guía, intranet hace referencia a cualquier red TCP/IP que no esté conectada a Internet pero que utilice estándares y herramientas de comunicación de Internet para suministrar información a los usuarios de la red privada. Por ejemplo, una organización puede instalar servidores Web a los que

los empleados sólo puedan tener acceso para publicar boletines informativos de la organización, cifras de ventas y otros documentos de la organización. Los empleados tienen acceso a la información mediante exploradores de Web.

Los servidores Web pueden configurarse para suministrar una intranet con las mismas características y servicios que se encuentran en Internet, como páginas de hipertexto (que pueden contener texto, hipervínculos, imágenes y sonidos), que respondan a las peticiones de información de los clientes del Web y tengan acceso a bases de datos. En esta guía, estos servicios de publicación se describen como “servicios de Internet”, independientemente de que se ejecuten en una intranet o en Internet.

1.6.1 Escenarios de Intranet. Internet Information Server se integra bien en la mayoría de los entornos existentes. Como Internet Information Server integra las capacidades de redes y seguridad de Windows NT, casi siempre podrá agregar el software a equipos existentes y usar las cuentas de usuario ya creadas. No es necesario usar un equipo dedicado para ejecutar Internet Information Server.

Por ejemplo, en un pequeño grupo de trabajo puede agregar Internet Information Server a un servidor de archivos y de impresión existente. El servidor Web del grupo de trabajo puede albergar páginas personales de estilo

Web, aplicaciones de grupo de trabajo personalizadas, servir como interfaz para la base de datos del Lenguaje de consulta estructurado (SQL) del grupo de trabajo o utilizar Servicio de acceso remoto (RAS) para otorgar acceso telefónico a los recursos del grupo de trabajo desde sitios lejanos (ver figura 11).

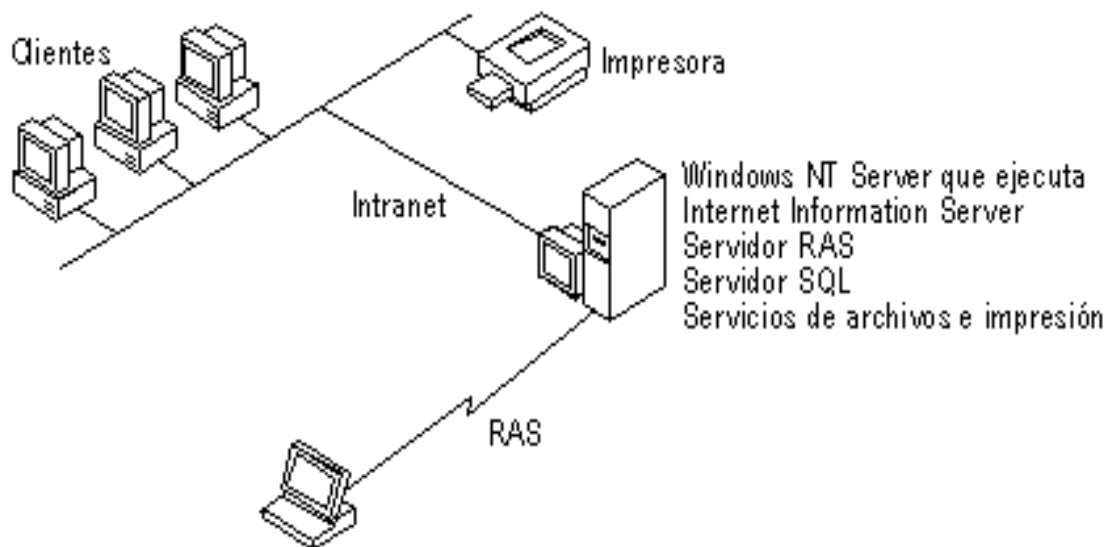


Figura 11. Escenario de Intranet

1.7 QUÉ ES INTERNET EXPLORER

Microsoft Internet Explorer es un explorador de Web. De la misma forma que Microsoft Word es una herramienta para crear y dar formato a documentos, o que Microsoft Excel es una herramienta para crear hojas de cálculo y realizar cálculos, Internet Explorer es una herramienta para recorrer y tener acceso, o “explorar”, a la información del Web.

La barra de herramientas de Internet Explorer proporciona una gran variedad de funciones y comandos detallados para administrar el explorador. La barra de direcciones, situada debajo de la barra de herramientas, muestra la dirección de la página Web actual. Para ir a una página Web nueva, escriba la dirección URL de la página directamente en el espacio en blanco de esta barra y, a continuación, presione ENTRAR en el teclado. También puede ir a una página nueva haciendo clic en un hipervínculo que salte a la nueva página.

El sistema operativo Microsoft Windows NT incluye Internet Explorer para Windows NT.

1.8 WINDOWS NT

Windows NT Server 4.0, con la potencia combinada de Internet Information Server, Index Server, y el software incluido de Microsoft FrontPage, proporciona la plataforma más completa para nuestra solución intranet. Como Windows NT Server 4.0 extiende las soluciones tradicionales de archivos / impresoras y de aplicaciones servidoras, disponemos de una plataforma escalable e integrada que nos permite implantar las soluciones intranet más avanzadas con las últimas novedades en facilidad y flexibilidad. La seguridad, base de los Servicios de Directorio de Windows NT Server 4.0, es una importante característica de la solución, ofreciéndonos un control total sobre quién tiene acceso a la información corporativa importante. Combinado con el servidor proxy de Microsoft, constituye una solución segura y eficiente que potenciará nuestra intranet.

Los siguientes productos y tecnologías se incluyen en Windows NT Server 4.0 y proporcionan la plataforma más completa para nuestras necesidades intranet más exigentes.

1.9 INTERNET INFORMATION SERVER

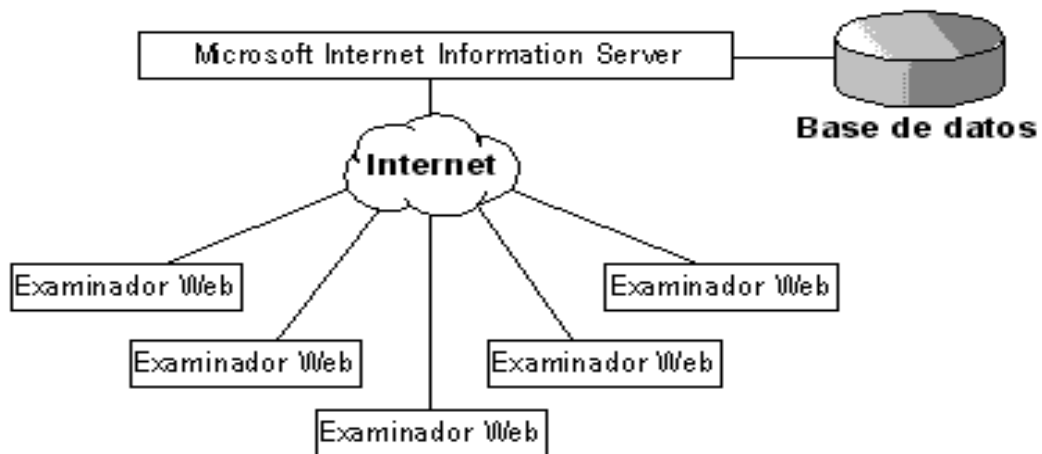


Figura 12:IIS

Microsoft Internet Information Server es un servidor Web que le permite publicar información en una intranet de la organización o en Internet. Internet Information Server transmite la información mediante el Protocolo de transferencia de hipertexto (HTTP). Internet Information Server puede configurarse también para proporcionar servicios de Protocolo de transferencia de archivos (FTP) y gopher. El servicio FTP permite que los usuarios transfieran archivos a y desde su sitio Web.

1.9.1 Qué puede hacer con Internet Information Server. Las posibilidades creativas de lo que puede ofrecer en un equipo con Microsoft Internet Information Server son infinitas. Algunos de los usos más comunes son:

Publicar en Internet una página principal de su organización que presente boletines informativos, información de ventas u ofertas de empleo.

- Publicar un catálogo y aceptar pedidos de sus clientes.
- Publicar programas interactivos.
- Ofrecer a su equipo de ventas remoto un acceso sencillo a la base de datos de ventas.
- Usar una base de datos de seguimiento de pedidos.

1.9.2 Cómo funciona Internet Information Server. Web es fundamentalmente un sistema de peticiones y respuestas. Los exploradores de Web solicitan información enviando una dirección URL a un servidor Web. El servidor Web responde devolviendo una página de Hypertext Markup Language (HTML).

La página HTML puede ser una página estática a la que ya se haya dado formato y esté almacenada en el sitio Web, una página que el servidor crea dinámicamente como respuesta a la información proporcionada por el usuario o

una página que presente una lista de los archivos y carpetas disponibles en el sitio Web.

1.9.3 Petición de dirección URL del explorador de Web. Todas las páginas de una intranet o de Internet tienen una dirección URL única que la identifica.

Los exploradores de Web solicitan una página mediante el envío de una dirección URL a un servidor Web. El servidor utiliza la información de la dirección URL para encontrar y mostrar la página.

La sintaxis de las direcciones URL es una secuencia específica de protocolo, nombre de dominio y ruta de acceso a la información solicitada. El protocolo es el método de comunicación utilizado para tener acceso a la información; por ejemplo, el Protocolo de transporte de hipertexto (HTTP).

Una dirección URL también puede contener información que el servidor Web debe procesar antes de devolver una página. Los datos se agregan al final de la ruta de acceso. El servidor Web pasa los datos a un programa o a un archivo de comandos para su procesamiento y devuelve los resultados en forma de una página Web. En la siguiente tabla se enumeran varios ejemplos de tipos de peticiones:

1.9.4 Respuesta del servidor Web. Un servidor Web responde a la petición de un explorador de Web devolviendo una página HTML. La página devuelta puede ser de tres tipos: una página HTML estática, una página HTML dinámica o una página de lista de directorio.

1.9.5 Páginas estáticas. Las páginas estáticas son páginas HTML fijas ya preparadas antes de la petición. El servidor Web devuelve las páginas HTML al usuario, pero no realiza ninguna acción especial. El usuario pide una página estática escribiendo una dirección URL - (<http://www.organizacion.com/home.htm>, en la siguiente figura) o haciendo clic en un vínculo que apunte a una dirección URL. La petición de dirección URL se envía al servidor y éste responde devolviendo la página HTML estática (ver figura 13).



Figura 13: Páginas estáticas

1.9.6 Páginas dinámicas. Las páginas dinámicas se crean como respuesta a las peticiones de los usuarios. Un explorador de Web recopila información presentando una página con cuadros de texto, menús y casillas de verificación que el usuario completa o activa. Cuando el usuario hace clic en un botón de un formulario, los datos procedentes del formulario se envían al servidor Web. El servidor pasa los datos a un archivo de comandos o una aplicación para su procesamiento, o bien consulta o envía datos a una base de datos. Entonces, el servidor devuelve los resultados al usuario en forma de una página HTML.

Si los usuarios quieren enviar consultas sin especificar un archivo en concreto, puede crear un documento predeterminado para un sitio Web o para un determinado directorio, o puede configurar su servidor para explorar los directorios. Si no se crea ningún documento predeterminado para un directorio y está configurada la exploración de los directorios, se devuelve al usuario una

lista de directorios (una versión hipertexto de una lista de un Explorador de Windows o del Administrador de archivos) en forma de página HTML. Después, el usuario puede ir al archivo apropiado haciendo clic en él dentro de la lista de directorios(ver figura 14).

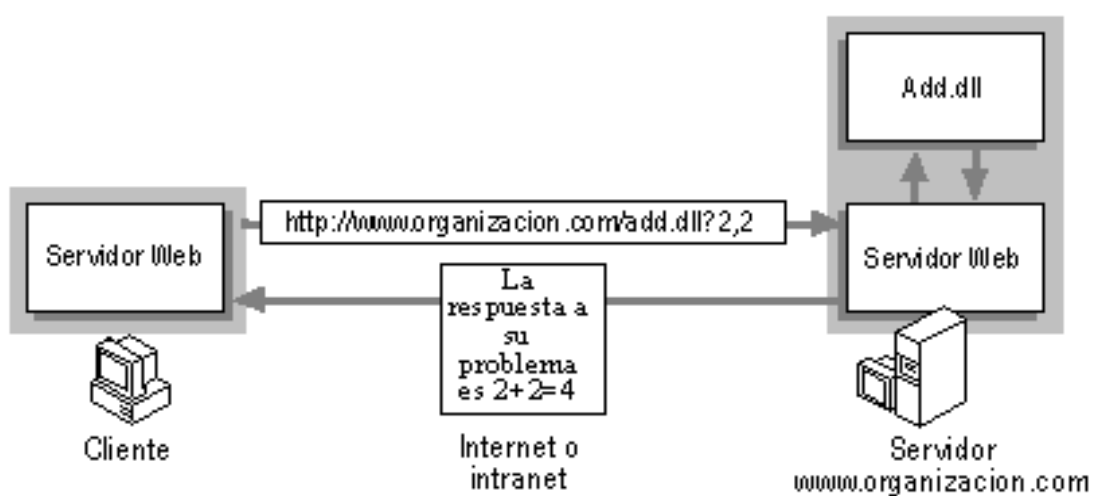


Figura 14: Páginas dinámicas

2. POR QUÉ UNA INTERFAZ GRÁFICA?

La CUTB desde sus inicios en el mundo de las redes, ha necesitado de generar información de sus bases de datos en Informix a sus docentes y administrativos en forma de archivos planos, los cuales son procesados manualmente para su posterior uso en un sistema fácil de manipular como lo es Windows 95, 98 o 2.000.

El propósito de crear una interfaz que facilite este método es con el fin de agilizar este proceso tener la información en sistema deseado.

Esta interface tiene la facultad de coger unos archivos planos arrojados o generador por el DBMS Informix y convertirlos a través de una serie de procesos en una base de datos en Access, el cual después puede ser utilizado con cualquier otra herramienta a través de la automatización.

2.1 ANÁLISIS Y DISEÑO DE LA INTERFACE GRÁFICA

Debido a que la mayoría del personal administrativo y docente no se siente a gusto con el método para convertir archivos planos generados por Informix a

una base de datos en Access de Microsoft, se vio la oportunidad de crear un procedimiento que fuera capaz de hacer esta conversión, lo cual requiere que se le suministre unos archivos planos y esto los convierte en una tabla de Access.

Para su realización se debe tener en cuenta ciertos parámetros como son:

1. Depositar los archivos planos en una carpeta previamente fijada.
2. Generar una base de dato en blanco previamente, con su estructura original de Informix.

2.1.1 Análisis del sistema de la interfaz gráfica. Una vez que los archivos planos estén en la carpeta predeterminada, se procede a coger uno a uno y se le realiza la conversión.

Por lo general el archivo plano tiene la información separada por unos delimitadores que en nuestro caso son unos pai (|) , (generalmente conocidos en lenguaje C como el símbolo “OR”) y cada línea es un registro , entonces se reconoce la línea hasta encontrar un pai y se extrae ese dato. Luego de encontrar todos los datos se procede a guardarlos en la base de datos creada, a continuación se lee otra línea y se repite el procedimiento.

Ahora como es imposible conocer los nombres de los campos dentro de un archivo plano, se debe hacer este procedimiento para cada archivo plano presente, especificando cada uno en la base de datos en cuestión.

Para nuestro proyecto se vio la necesidad de crear otras tablas a partir de las ya creadas, para poder hacer un buen trabajo de relacionalidad entre ellos.

2.1.2 Diagrama de flujo de la Interfaz Gráfica

A groso modo la interfaz gráfica se compone de tres cuerpos fundamentales que son: Datos de entrada, procesamiento de los datos o Conversión y datos de salida. Ver Figura 15

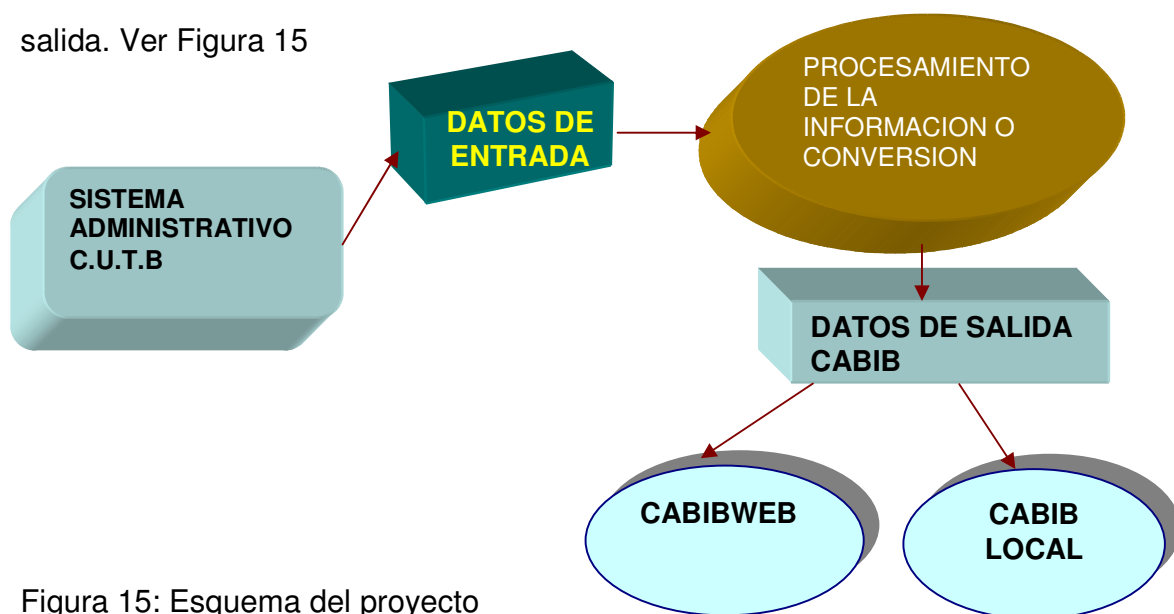


Figura 15: Esquema del proyecto

2.1.2.1 Sistema Administrativo C.U.T.B. Es aquel que se encarga de proporcionar los datos de entrada para ser procesados.

2.1.2.2 Los datos de entrada. Son los que se encargan de alimentar el sistema, estos son los archivos planos que proporciona Informix para su procesamiento y la base de datos creada previamente.

Para nuestro caso son 6 archivos planos y 1 base de datos creada con la estructura de la base de datos en Informix, pero con todos los campos de tipo carácter.

A continuación se muestra este inciso detallado en la Figura 16

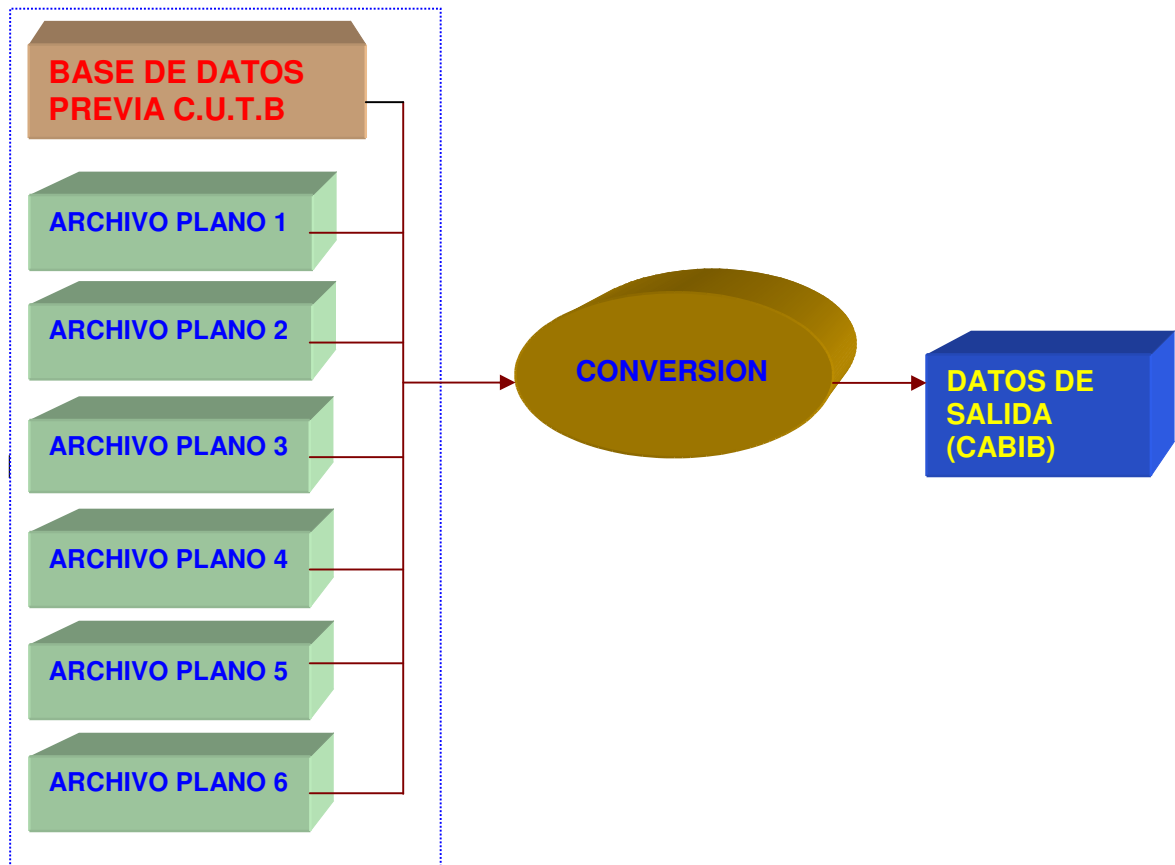


Figura 16: Entrada de datos

2.1.2.3 Conversión. Este proceso se encarga de coger los archivos planos, procesarlos, extraer los datos y guardarlos en varias variables para su posterior uso. Ver Figura 17

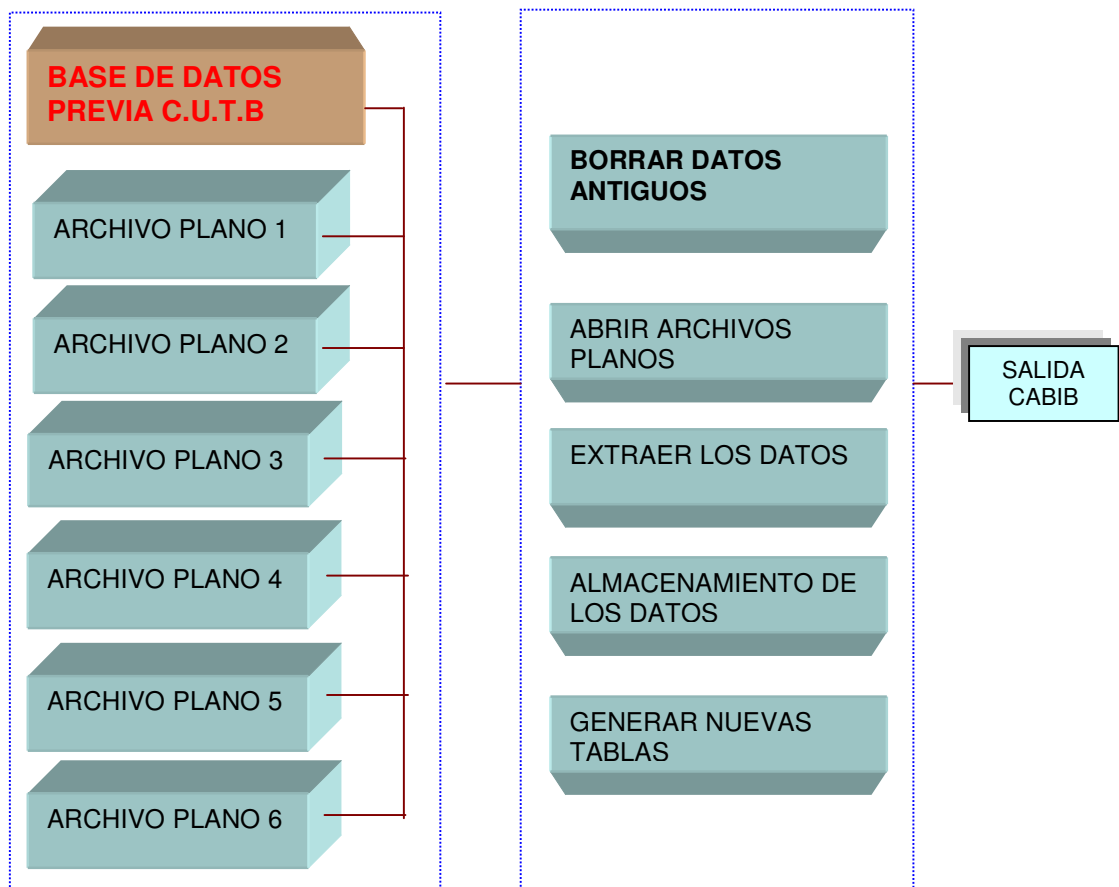


Figura 17: Conversión

2.1.2.4 Los datos de Salida: Son los datos que proporciona el sistema para su manipulación, estos datos son los que actualizan la base de datos.

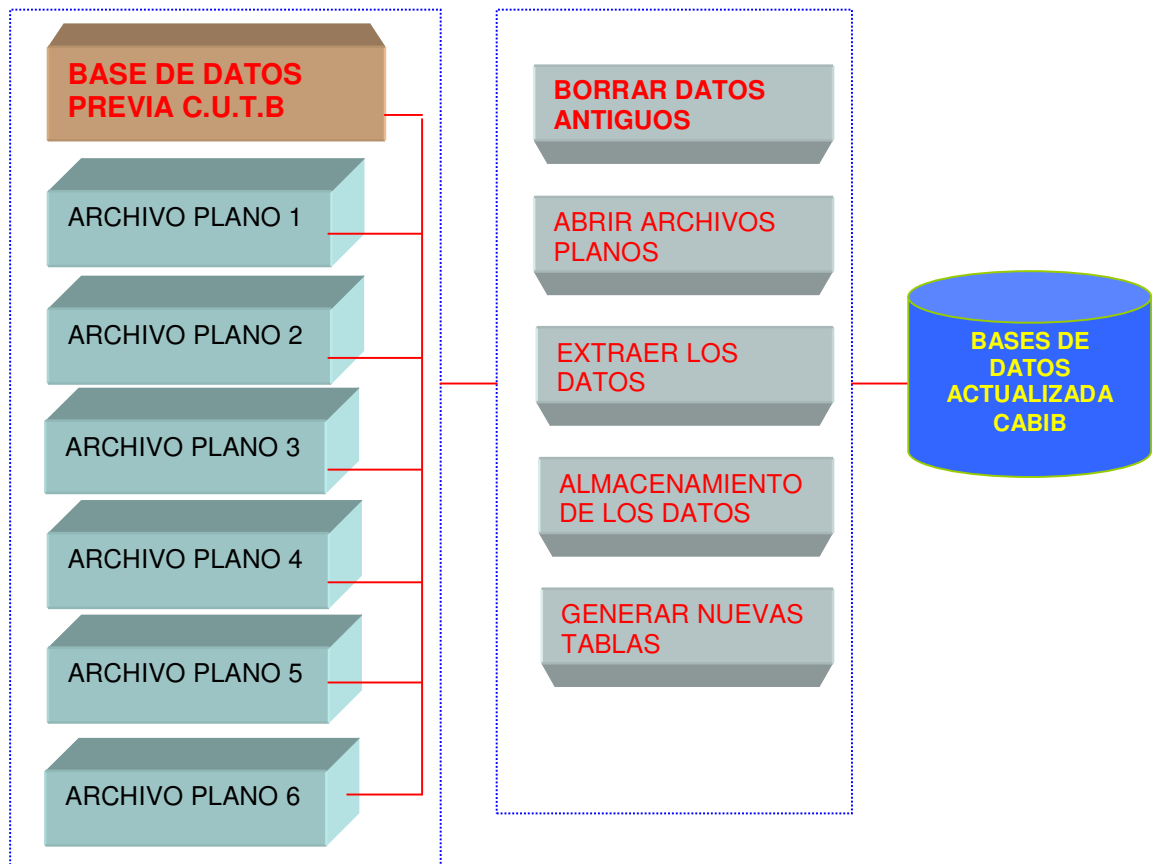


Figura 18: Salida de datos

Así esta fundamentalmente la interfaz gráfica, la cual se realizó bajo el lenguaje de programación Visual Basic 6.0.

2.1.3 Diccionario de Datos. Esto explica el contenido de cada una de las tablas que contiene la base de datos actualizada CABIB.

1. Cuadro Maestro

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Código	Texto	Código del libro	5		No
Ind_sig	Texto		2		No
Dewex	Texto	Signatura topografica	30		No
Cantidad	Texto	Cantidad de libros	3		No
Prestados	Texto	Libro prestado	3		No
Ind_fecha	Texto		1		No
Fecha1	Texto		4		No
Fecha2	Texto		4		No
Cad_Orig	Texto		3		No
Ilustra	Texto	Ilustración del libro	4		No
F_Rep	Texto		1		No
F_cant	Texto	Forma de contenido	4		No
P_ofic	Texto		1		No
Idioma	texto	Idioma del libro	3		No
País	Texto	País de procedencia	3		No
Frecuent	Texto		1		No
ISDS	Texto		2		No
Tipo_PS	Texto	Tipo publicación seriada	1		No
Soporte	Texto	Soporte físico	1		No
Indice	Texto	Indice acumulativo	1		No
Alfa_O	Texto	Alfabeto original	1		No
Ind_CC	Texto		1		No
Cata_col	Texto	Catálogo colección nacional	5		No
Tipo_col	Texto	Tipo colección	1		No
Tipo_doc	Texto	Tipo documento	1		No
Bib_nac	Texto	Bibliografía nacional	3		No
Ind_autor	Texto	Indice de autor	3		No

Continuación del Cuadro Maestro

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Edición	Texto	Edición del libro	50		No
Publica	Texto	Publicación	50		No
Fondos	Texto		50		No
Descri	Texto	Descripción	50		No
Designa	Texto		50		No
Serie	Texto		50		No
ISBN	Texto		20		No

El cuadro Maestro, es el cuadro más importante ya que este contiene información de todos los demás cuadros y está involucrada en todas las relaciones y consultas hechas.

2. Cuadro Autores

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Código	Texto	Código del libro	4	No	Si con duplicado
Tipo	Texto	Tipo de autor	3	No	No
Ind	Texto		3	No	No
Campo 1	Texto	Nombre completo del autor	75	No	No
Campo 5	Texto	Identifica la clase de autor	30	No	No
Activo	Texto	Existencia del libro	1	No	No

Este cuadro contiene información de los autores de los libros que son consultados a través de CABIB.

3. Cuadro Materia

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Código	Texto	Código del libro	4		Si con duplicado
Tipo	Texto	Etiqueta de la materia	3		No
Ind	Texto		3		No
Tema	Texto	Materia a la que pertenece el libro	110		No
Activo	Texto	Si el libro esta activo o no	1		No

El cuadro materia tiene información de todas las materias, registradas en la base de datos para ser consultadas.

4. Cuadro Título

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Indexado
Código	Texto	Código del libro	4	No
Tipo	Texto	Etiqueta característica del libro	3	No
Ind	Texto		3	No
Título	Texto	Título del libro	110	Si con duplicados
Activo	Texto	Si el libro esta activo o no	1	No

Este cuadro contiene información de los títulos de los libros que el usuario consultará.

5. Cuadro Inventario

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Identi	Texto		5		Si con duplicado
Vol	Texto	Numero de volumen del libro	3		No
Procede	Texto	Procedencia del libro	1		No
Vr_compra	Moneda	Valor de la compra		2	No
Prove	Texto	Proveedor	30		No
Fedra_c	Fecha hora	Fecha de la compra	Fecha corta		No
Estado	Texto	Estado del libro	1		No
C_costos	Texto	Centro de costos	5		No
Area	Texto	Ubicación donde se encuentra el libro	20		No
Tipo_col	Texto		1		No
Vigencia	Texto	Vigencia	1		No
Código	Texto	Código del libro	4		No

Este Cuadro es importante ya que a partir de esta, se visualiza información correspondiente al libro en consulta.

6. Cuadro Autortítulo

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Codautor	N	Código del autor	Enterolargo	Automático	Si con duplicado
Codtitulo	Texto	Código del título	4	No	no

Cuadro intermedia que relaciona al autor con los libros escritos por él.

7. Cuadro Autonombre

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Codautor	Autonumerico	Código del autor	Enterolargo	No	No
Nombreautor	Texto	Nombre completo del autor	75	No	Si con duplicado

Asociar al autor con un código único para su mejor desempeño dentro del software CABIB

8. Cuadro Notas

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Lugares decimales	Indexado
Código	Texto	Código del libro	4		Si con duplicado
Tipo	Texto	Etiqueta característica del libro	3		No
Nota	Texto	Notas alusivas al libro	80		No
Activo	Texto	Si el libro esta activo o no	1		No

Con este cuadro se obtiene información acerca de las notas que cada libro consultado tenga.

9. Cuadro Temas

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Indexado
Codtema	Autonumerico	Código de la materia	Entero largo	No
Tema	Texto	Nombre de la materia	110	Si

El cuadro temas tiene información relacionada con los temas que contienen las materias que serán consultadas.

10. Cuadro Tematitulo

Nombre del campo	Tipo de datos	Descripción	Tamaño del campo	Indexado
Codtema	Numérico	Código de la materia	Entero largo	No
Codtitul	texto	Código del titulo	4	No

Este Cuadro contiene información que relaciona un tema específico con diversos títulos.

2.1.4 Datos de entrada. Son los datos proporcionados por el sistema administrativo de la C.U.T.B.

2.1.4.1 Base de datos previa. A continuación se describe la estructura de datos de la base de datos preestablecida para la conversión.

2.1.4.2 Archivo plano 1. Este archivo se llama bibctb01, este es el archivo que contiene los datos de la tabla generada por Informix del Maestro.

2.1.4.3 Archivo plano 2. El nombre de este archivo es bibctb02, este es el archivo que contiene los datos de la tabla generada por Informix de Inventario.

2.1.4.4 Archivo plano 3. Este archivo es llamado bibctb03, este es el archivo que contiene los datos de la tabla generada por Informix de Autores.

2.1.4.5 Archivo plano 4. Este archivo plano se llama bibctb04, este es el archivo que contiene los datos de la tabla generada por Informix de Título.

2.1.4.6 Archivo plano 5. Este archivo se llama bibctb05, este es el archivo que contiene los datos de la tabla generada por Informix de Materia.

2.1.4.7 Archivo plano 6. El archivo plano 6 se llama bibctb06, el cual contiene los datos de la tabla generada por Informix de Notas.

2.1.5 Conversión. Este proceso es el que se encarga de realizar el procedimiento completo de la interfaz, es decir, es el que se encarga de pasar los archivos planos a una base de datos Access. El cual consta de los siguientes subprocesos: borrado de datos antiguos, abrir archivos planos, extracción de datos, almacenamiento de los datos extraídos y generación de las nuevas tablas.

2.1.5.1 Borrado de datos antiguos. La función de este subproceso es el de eliminarlos datos contenidos en cada una de las tablas que están en la base de datos previa, para posteriormente ser actualizada.

2.1.5.2 Abrir archivos planos. La función de este subproceso es la de abrir los archivos planos para su respectiva manipulación.

2.1.5.3 Extracción de los datos. Este subproceso se encarga de extraer cada uno de los datos, contenidos en un archivo plano previamente abierto.

2.1.5.4 Almacenamiento de los datos. Este subproceso se encarga de recoger cada uno de los datos extraídos por el subproceso anterior y los

almacena en una variable, para posteriormente ser almacenados o guardados en su nuevo formato de base de datos.

2.1.5.5 Generar nuevas tablas. Este subproceso se encarga de generar tablas auxiliares, que aportan mas eficacia al software.

2.1.6 Datos de salida. Esta es la etapa final en la cual queda actualizada la base de datos en Access, para su posterior manipulación por parte del usuario final.

3. ESTRUCTURA LÓGICA DE LA INTERFAZ

Después de haber estudiado a fondo la metodología de la interfaz, se visualiza ahora la estructura lógica del software, lo cual comprende los algoritmos fundamentales, empleados para su funcionamiento, cuyo código se describe a continuación.

Este procedimiento se describe para uno de los archivos planos generados por Informix y es idéntico para cualquier archivo plano que se quiera convertir.

3.1 CUERPO PARCIAL DEL PROCEDIMIENTO DE CONVERSIÓN (INTERFAZ)

LIN es el Buffer de entrada de datos

Archiplano es el nombre de la ubicación y del archivo plano en ASCII

```
Dim ArchiPlano, LIN As String
```

```
Dim I, Ind, R As Long
```

Variables contadores y de localización del programa.

```
Dim Copia(34) As String
```


Contiene en memoria los campos convertidos para luego ser grabados en la base de datos.

```
DoEvents
```

Permite ejecutar otro evento en el mismo momento que se ejecuta el actual procedimiento

```
Label1.Visible = True
```

```
Label2.Visible = True
```

Refresca el recordset de una base de datos.

```
Data1.Refresh
```

El siguiente procedimiento permite establecer si un recordset tiene elementos y los eliminamos para poder actualizarlos con los nuevos que nos vienen en el archivo plano.

Determina si no existen registros en el Recordset

```
If Data1.Recordset.AbsolutePosition <> -1 Then
```

Mover al último

```
Data1.Recordset.MoveLast
```

Determina la cantidad de registros en el Recordset

```
Tito% = Data1.Recordset.RecordCount
```

```
Data1.Recordset.MoveFirst
```

```
For II% = 1 To Tito%
```

```
    Data1.Recordset.Delete
```

```
    Data1.Recordset.MoveNext
```

```
Next II%
```

```
Data1.Refresh
```

```
End If
```

```
Label1.Caption = "Notas Registro #"
```

```
ArchiPlano = "c:\BDBibVir\bibctb06"
```

Determinara el # del siguiente archivo a abrirse.

```
Fnum = FreeFile
```

Abre el archivo en modo lectura.

```
Open ArchiPlano For Input Access Read As #Fnum
```

```
Me.MousePointer = 11
```

```
I = 1
```

El siguiente procedimiento determina la lectura del archivo plano hasta que se encuentre su final (EOF) = True.

```
Do While Not EOF(Fnum)
    Label2.Caption = I
```

Se lee un registro

```
Line Input #Fnum, LIN
```

Creamos un índice igual a cero y lo incrementaremos hasta que sea mayor que el numero de campos del recorset definido en nuestra estructura.

```
Ind = 0
Do
```

Determina la posición del carácter “|”

```
R = InStr(1, LIN, "|")
```

Almacena en el arreglo previamente definido los caracteres de este campo.

```
Copia(Ind) = Mid$(LIN, 1, R - 1)
```

Recorta nuestro buffer de entrada haciendolo R caracteres más pequeños.

```
LIN = Mid$(LIN, R + 1)
Ind = Ind + 1
```

```
Loop While Ind <= data1.recorsets.fields.count
```

Determina el bucle con la longitud igual al numero de campos del recordset.

```
Data1.Recordset.AddNew
```

Adiciona un registro

```
Data1.Recordset!Codigo = Copia(0)
```

almacena cada uno de los campos del archivo ASCII en el recordset.

```
Data1.Recordset!Tipo = Copia(1)
```

```
Data1.Recordset!Nota = Copia(2)
```

```
Data1.Recordset!Activo = Copia(3)
```

```
Data1.Recordset.Update
```

```
I = I + 1
```

```
DoEvents
```

```
Loop
```

```
Close #Fnum
```

```
Data1.Refresh
```

```
Label3.Visible = True
```

```
Label4.Visible = True
```

```
Data2.Refresh 'Refresca el recorset
```

4. AUTOMATIZACIÓN DE LA INTERFAZ

Para que la interfaz mantenga el sistema actualizado periódicamente, se debe automatizar, o sea, hacer que se ejecute la aplicación cada determinado periodo de tiempo.

Para lograr esto se optó por recurrir a un servicio de Windows Nt, llamado Win AT, el cual tiene la facultad de ejecutar cualquier tarea cada cierto tiempo.

El servicio WinAT tiene las siguiente cualidades:

El comando At programa la ejecución de comandos y programas en un equipo en una fecha y hora especificadas. Para utilizar este comando, debe estar en funcionamiento el servicio Programación.

```
At [\equipo [[id] [/delete [/yes]]
```

```
At [\equipo] hora [/interactive] [/every:fecha[,...] | /next:fecha[,...]] "comando"
```

Parámetros

ninguno

Si se utiliza sin parámetros, At muestra una lista de los comandos programados.

\\equipo

Especifica un equipo remoto. Si se omite este parámetro, los comandos se programan para el equipo local.

id

Es un número de identificación asignado a un comando programado.

/delete

Cancela un comando programado. Si se omite el modificador id, se cancelan todos los comandos programados en el equipo.

/yes

Fuerza una respuesta afirmativa a todas las consultas del sistema cuando se eliminan los sucesos programados.

hora

Especifica la hora a la que debe ejecutarse el comando. La hora se expresa como horas: minutos en la notación de 24 horas (00:00 [medianoche] hasta 23:59).

/interactive

Permite que el trabajo interactúe con el escritorio del usuario que haya iniciado la sesión en el momento de ejecutarse el trabajo.

/Every: fecha[,...]

Ejecuta el comando todos los días especificados de la semana o del mes (por ejemplo, todos los jueves o el tercer día de cada mes). El parámetro fecha debe especificarse como uno o más días de la semana (L, M, Mi, J, V, S, D) o uno o más días del mes (indicando números comprendidos entre 1 y 31). Separe las entradas múltiples de fecha con comas. Si se omite fecha, se asume el día actual del mes.

/next: fecha[,...]

Ejecuta el comando especificado el siguiente día al que se indique (por ejemplo, el siguiente jueves). El parámetro fecha debe especificarse como uno o más días de la semana (L, M, Mi, J, V, S, D) o uno o más días del mes (indicando números comprendidos entre 1 y 31). Separe las entradas múltiples de fecha con comas. Si se omite fecha, se asume el día actual del mes.

"comando"

Es el comando o programa (archivo .EXE o .COM) de Windows NT o un programa por lotes (archivo .BAT o .CMD) que debe ejecutarse. Si el comando

requiere como argumento una ruta de acceso, utilice la ruta absoluta, es decir, el nombre de ruta de acceso completo, que comienza con la letra de unidad. Si el comando está en un equipo remoto, especifique el servidor y el nombre de recurso compartido, en lugar de la letra de unidad remota. Puede utilizar comillas al principio y al final del comando, sea con el comando At en la línea de comandos o en un archivo por lotes. Si el comando incluye modificadores que se utilicen con el propio comando y con At, debe poner comando entre comillas. Si el comando no es un archivo ejecutable (.EXE), es necesario incluir delante de él `Cmd /c`; por ejemplo `Cmd /c Dir > c:\test.out`.

El comando At requiere ser miembro del grupo local Administradores. Además, el comando At no carga automáticamente Cmd, el intérprete de comandos, antes de ejecutar los comandos. A menos que esté ejecutando un archivo ejecutable (.EXE), debe cargar explícitamente CMD.EXE al principio del comando; por ejemplo, `Cmd /c Dir > c:\test.out`.

4.1 INTEGRACIÓN DE LA INTERFAZ CON WIN AT

Como esta interface se ejecutará a altas horas de la noche, y el primer día de cada mes, los pasos para programar estas herramientas son:

- ✓ Oprimir el botón de inicio de Windows Nt.
- ✓ Hacer click en el comando ejecutar del menú Inicio.
- ✓ Teclar WinAt y aparecerá la siguiente pantalla:

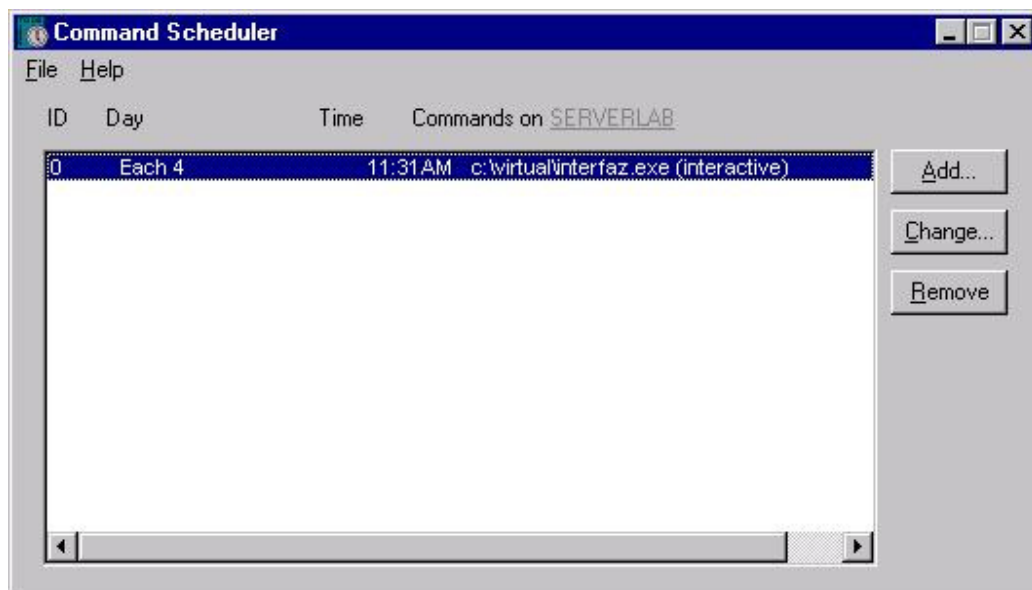


Figura 19: Pantalla inicial de winAT

Oprimir el botón Add y nos mostrará la siguiente pantalla:

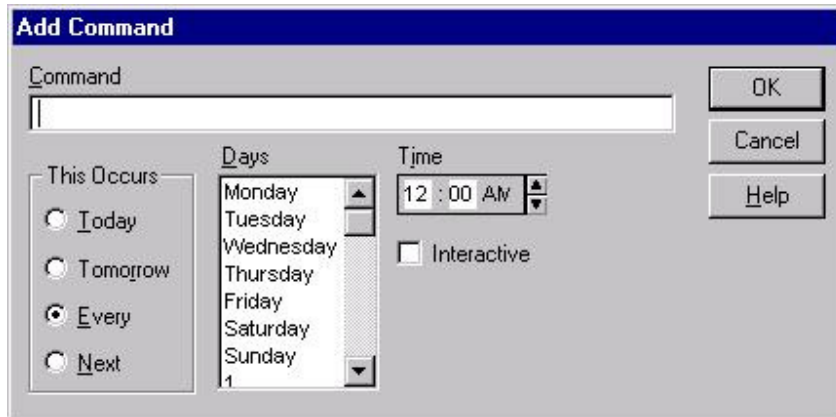


Figura 20: comando Add de WinAT

- ✓ En la línea de comando introducir la ruta donde se encuentra la interfaz:
D:\Archivos de programas\proyect1\BIBVirtual.EXE.
- ✓ Seleccionar la opción EVERY
- ✓ En los días seleccionar 1
- ✓ En horario seleccionar 1:00 A.M
- ✓ Por último oprimir el botón OK .

5. ANÁLISIS Y DISEÑO DEL SISTEMA DE INFORMACIÓN BIBLIOGRÁFICA PARA CONSULTA DE LA C.U.T.B CABIB (CENTRO DE ATENCIÓN BIBLIOGRÁFICO PARA CONSULTA A NIVEL DEL WEB Y LOCAL)

Para esto fue necesario analizar y comprender el entorno circundante, identificar y definir los objetos necesarios que intervienen en el sistema.

El sistema de tratamiento de información bibliográfica es un gestor de referencia de información de tal manera que se pueden clasificar en uno o varios índices, visualizar para su consulta y reclasificar al sistema proceso la petición del usuario devolviendo un mensaje e indicando el éxito o el fracaso de la petición.

Definición de entradas del sistema de Información Bibliográfica de la C.U.T.B, CABIBWEB.

Los objetos a tratar son los siguientes: Libro, Indice, Sistema y Usuario.

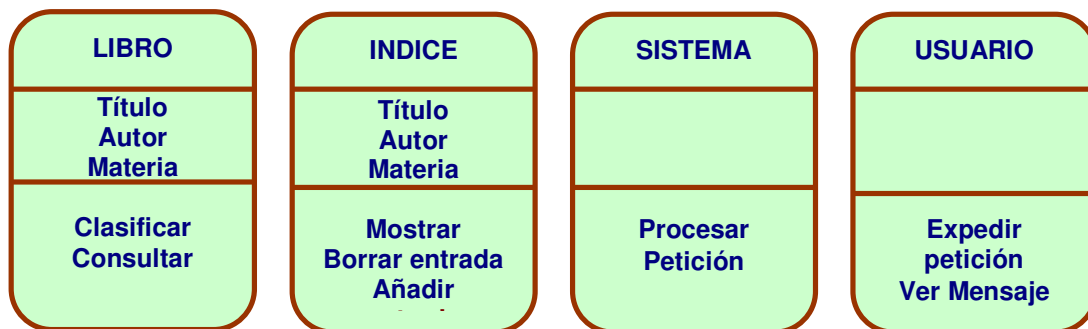


Figura20: Objetos

Estos objetos se relacionan o asocian de la siguiente manera:

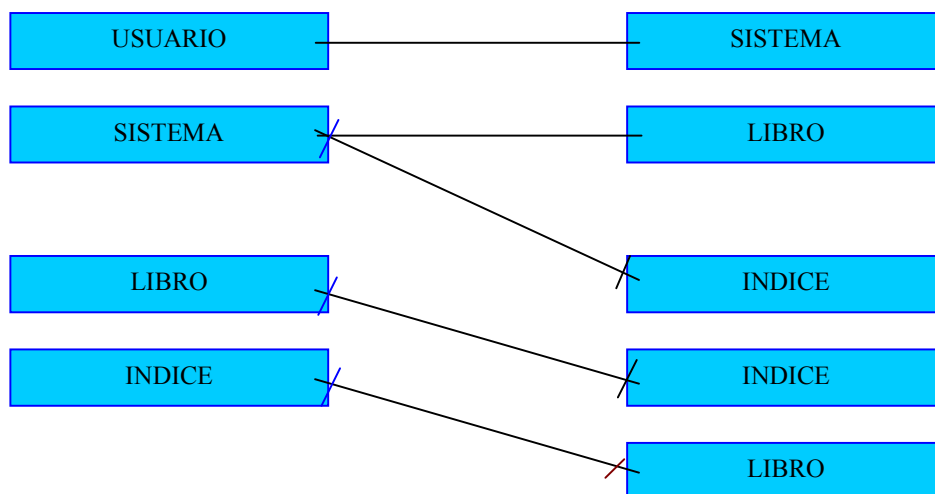


Figura 21: Relaciones

El diagrama de las asociaciones entre ellos quedará así:

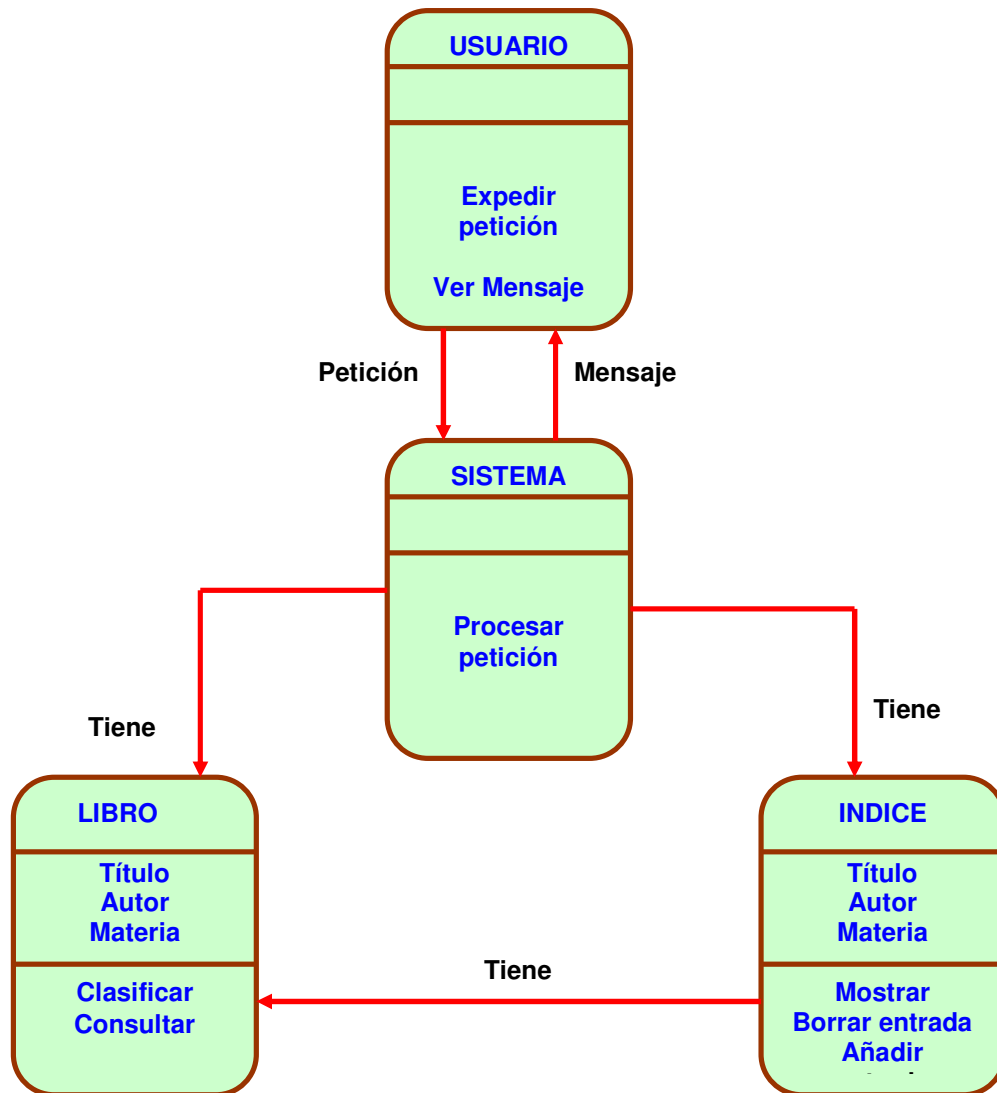
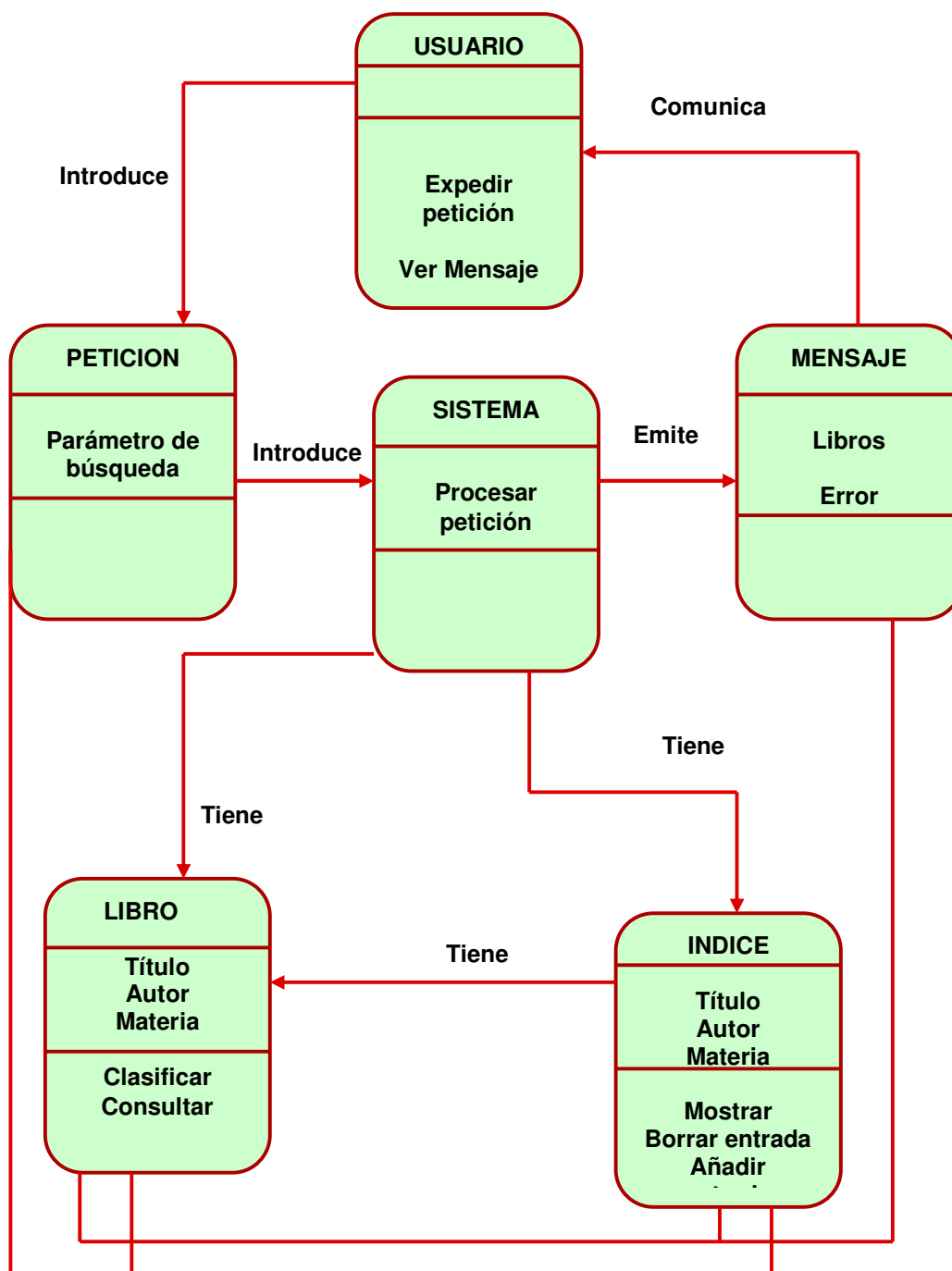


Figura 22: Diagrama específico

Cuando se consulta un libro, se debe indicar el título, el autor ó la materia ; así como los índices a los que se va a asociar.

Por último el diseño específico sería el siguiente:



6. ESTRUCTURA LÓGICA DEL SISTEMA DE INFORMACIÓN BIBLIOGRÁFICA PARA CONSULTAS DE LA C.U.T.B CABIB (CENTRO DE ATENCIÓN BIBLIOGRÁFICA PARA CONSULTAS A NIVEL DE WEB)



Figura 24 Pantalla inicial del Sitio Web CABIB

6.1 ESTRUCTURA DEL SITIO WEB DEL SISTEMA DE INFORMACIÓN BIBLIOGRÁFICA PARA CONSULTA EN EL WEB (CABIB)

El sitio Web de la biblioteca de la CUTB para CABIB está conformado por página estáticas y dinámicas; entre las páginas estáticas encontramos:

- 🖥️ Default.HTM: Es la página que le aparece al usuario tan pronto entra al sitio de Web.
- 🖥️ General.HTM: Es la página donde aparece contenido el menú principal, la cabecera (logo) y el cuerpo de la consulta, la portada.
- 🖥️ Menú principal HTM : Contiene las opciones a escoger para la realización de la consulta . Aquí se encuentra el autor, el titulo, la materia ó el tema, ayudas e información de los empleados de biblioteca de la CUTB.
- 🖥️ Entrada.HTM: pantalla temporal de bienvenida
- 🖥️ Ayuda.HTM: Guía al usuario a través de definiciones y ejemplos para que comprenda el entorno del sitio Web.
- 🖥️ Acercade.HTM: Información general acerca de los empleados, realizadores del sitio Web, profesores y otras tópicos de interés común

Las páginas dinámicas están elaborados en ASP y son los que se encargan de rescatar la petición del usuario y procesarlo para darle una respuesta al usuario.

Estas páginas son:

- Autor.Asp: Este programa se encarga de recoger el parámetro de búsqueda del usuario por autor.
- Titulo.Asp: Este programa se encarga de recoger el parámetro de búsqueda del usuario por título.
- Materia.ASP: este programa se encarga de recoger el parámetro de búsqueda del usuario por materia.

Las siguientes páginas son el motor de la consulta.

Elas son los que capturan la petición generada por el usuario y realiza el proceso de consulta a través de SQL y herramientas propias de ASP para el cuerpo de las bases de datos a través de ODBC y objetos ADO.

- Resp_Aut : Inserta el parámetro de consulta en cadena SQL y genera la primera tabla con respecto al autor solicitado.
- Resp_MAT: Inserta el parámetro de consulta en cadena SQL y genera la primera tabla con respecto a la materia solicitada.

- Resp_TIT: Inserta el parámetro de consulta en cadena SQL y genera la primera tabla con respecto al título solicitado.
- Resp_GEN :Es llamada si la consulta es mayor a 5 registros en la petición de título.
- Resp_GEN1: Es llamada si la consulta es mayor a 5 registros en la petición de materia.
- Resp_Gen2: Es llamada si la consulta es mayor a 5 registros en la petición de autor.
- Ayudita. Asp: Página que guía al usuario acerca de como debe usar la página de consulta.

6.1.1 Esquema del sitio Web de la biblioteca virtual

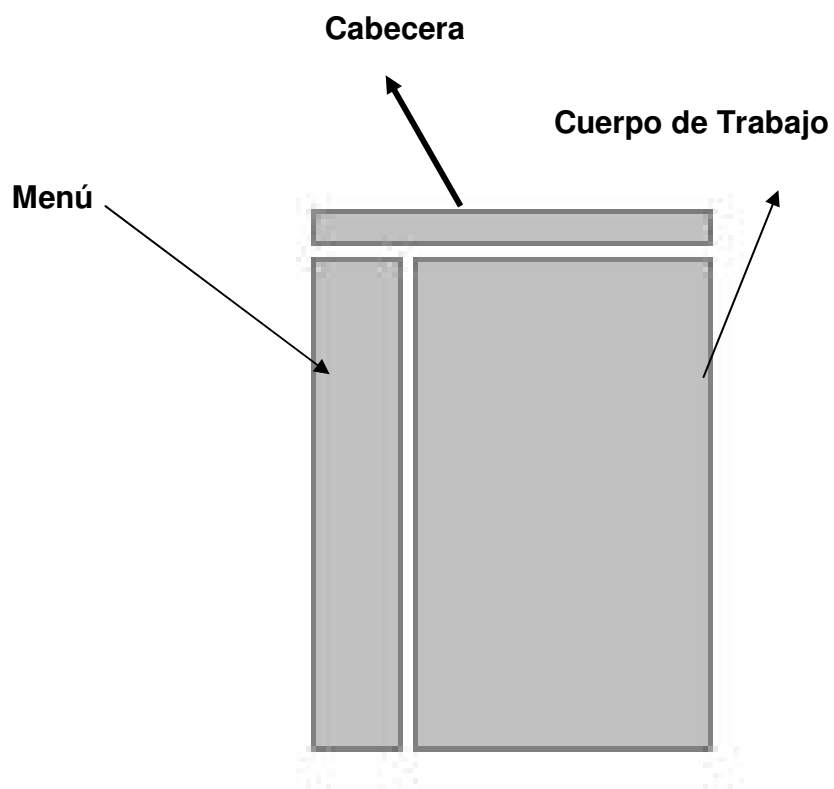


Figura 25: Esquema sitio WEb

6.1.2 Vista previa del sitio Web de la biblioteca

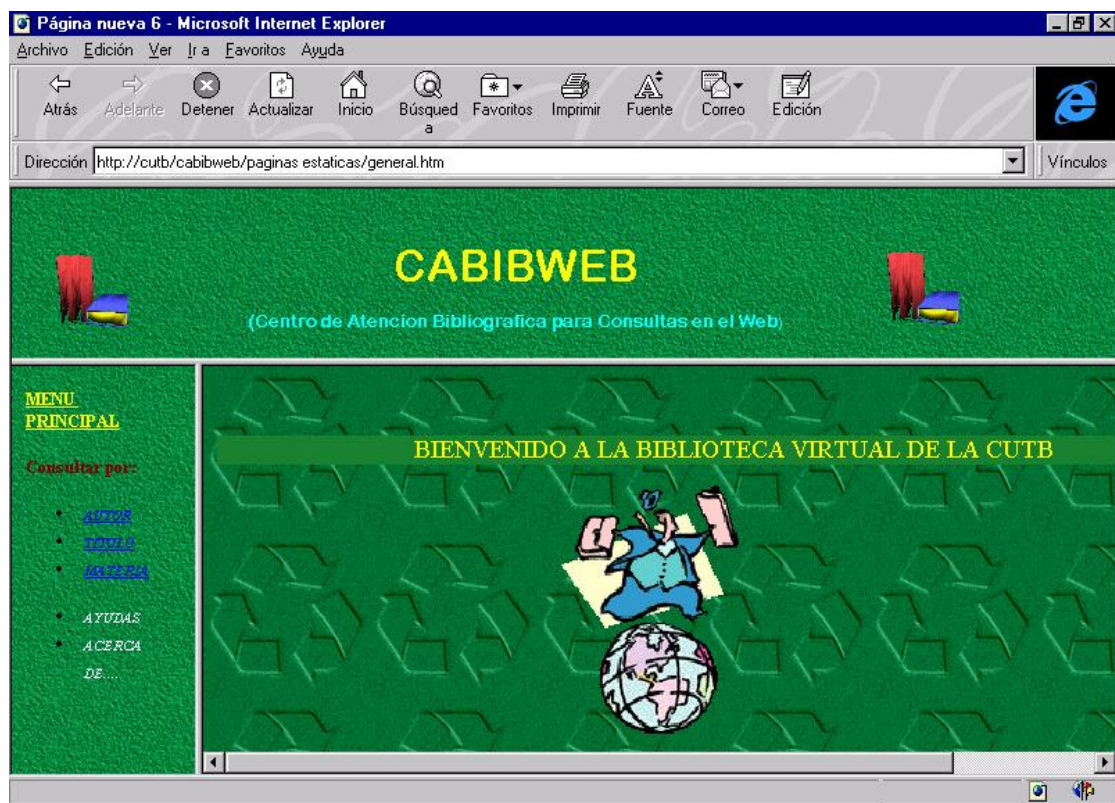


Figura 26: Aquí tenemos la pantalla de menú de CABIBWEB.

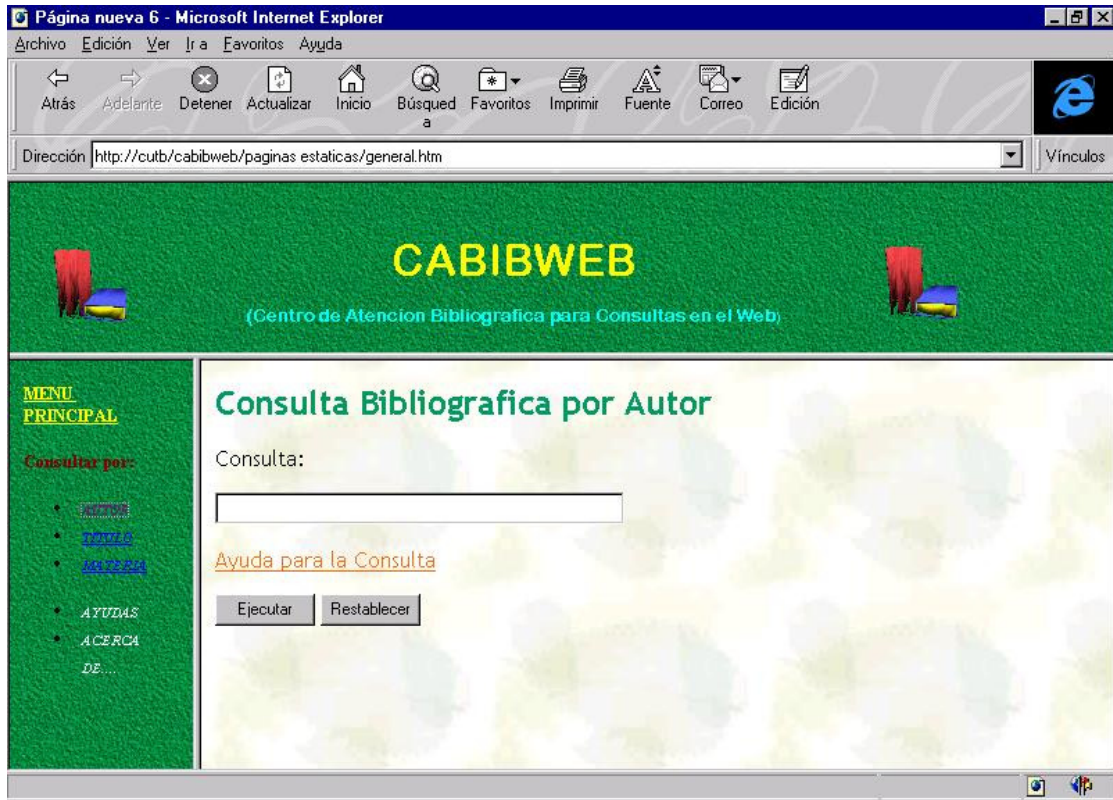


Figura 27 Aquí tenemos un ejemplo de consulta bibliográfica por autor.

Página nueva 6 - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

Atrás Adelante Detener Actualizar Inicio Búsqueda Favoritos Imprimir Fuente Correo Edición

Dirección http://cutb/cabibweb/paginas_estaticas/general.htm Vínculos

CABIBWEB

(Centro de Atención Bibliográfica para Consultas en el Web)

MENU PRINCIPAL

Consultar por:

- [AUTOR](#)
- [TITULO](#)
- [MATERIA](#)
- [AYUDAS](#)
- [ACERCA DE...](#)

Consulta Bibliografica Por Autor:

Resultado de la Consulta

Página 1

Titulo	Signatura Topografica	Materia	Apellido1	Apellido2	Nombre1	Nombre2
ADMINISTRACION POR POLITICA HOSHIN-KANRI	658.4012/M425	ADMINSTRACION	MATAMALA S.		RICARDO	

Figura 28 Resultado de la consulta bibliográfica por autor.

6.2 ESTRUCTURA DE UNA PÁGINA WEB EN ASP

El código de una página asp se compone de la siguiente forma:

```
< HTML >  
< BODY >  
< % secuencia de comandos en VBScript% >  
< / BODY >  
< / HTML >
```

Todas las páginas ASP siempre tienen secuencias de comandos dentro de los símbolos < %...% >, las etiquetas que no se encuentran con estos símbolos son propias del lenguaje HTML.

Ejemplo:

```
< HTML >  
< BODY >  
< % for i =1 to 10% >  
< B> Esto es una muestra<% i %> </B >  
  
< % NEXT i%>  
< / BODY >  
< / HTML >
```

6.2.1 Estructura lógica de la página de consulta del sitio Web CABIB. Aquí se describe en forma parcial el código de las páginas activas del sitio Web CABIB, debido a que ese mismo código es similar entre todos.

Estas páginas primero abren la base de datos a gestionar y los registros a procesar.

Luego toman el parámetro arrojado por la página de consulta en una variable llamada `sql`, luego introduce este parámetro en la cadena de consulta llamada `fp_sQry`, a través de un procedimiento complejo elaborado en Vscripts; una vez la cadena `fp_sQry` está completa se procede a invocar el objeto de ADODB que ejecuta la cadena `sql` y arroja los resultados de la consulta en forma tabulada; luego si la consulta es mayor de 1 página (una página es una serie de registros predefinidos) se genera una nueva petición a otra página sin el procedimiento de adición de parámetro.

Por último cierran el objeto y se acaba la transacción.

6.2.2 Análisis del cuerpo de las páginas dinámicas del sitio Web CABIBWEB. Para este análisis se tomaron como marco de referencia a las páginas Resp_Aut.asp y Resp_Gen2.asp.

6.2.2.1 Análisis de la página Resp_Aut.asp. A continuación se esbozará parte del código fuente de la página Resp_Aut.asp y seguir su lógica a fin de comprender su funcionamiento:

Esta es la cabecera del programa donde se declaran todas las variables a manejar.

```
<%  
'---- Valores de CursorTypeEnum ----  
Const adOpenForwardOnly = 0  
Const adOpenKeyset = 1  
Const adOpenDynamic = 2  
Const adOpenStatic = 3  
  
'---- Valores de LockTypeEnum ----  
Const adLockReadOnly = 1  
Const adLockPessimistic = 2  
Const adLockOptimistic = 3  
Const adLockBatchOptimistic = 4
```

Aquí empieza la captura del parámetro enviado por el usuario el cual se guarda temporalmente en la variable sql

```
sql = Request("sql")  
If sql = "" Then  
    Response.Redirect("autor.asp")  
End If
```

Esta es la invocación a los objetos de ADODB y se abre la base de datos conectada a ODBC

```
Set Conn = Server.CreateObject("ADODB.Connection")
Set RS = Server.CreateObject("ADODB.RecordSet")
```

Nombre de la conexión ODBC con la base de datos original
Conn.Open "biblioteca"

Conversión del parámetro de búsqueda Sustituye parámetros de formulario en la cadena de consulta.

La cadena SQL o cadena de consulta en la cual se insertará el parámetro introducido por usuario

```
fp_sQry="SELECT DISTINCT Titulo_ubicacion.Titulo,
Titulo_ubicacion.Dewey AS [Signatura Topografica],
Titulo_ubicacion.Materia, Titulo_ubicacion.Apellido1,
Titulo_ubicacion.Apellido2, Titulo_ubicacion.Nombrel,
Titulo_ubicacion.Nombre2, Titulo_ubicacion.Ubicacion,
Titulo_ubicacion.ISBN, notas.Nota FROM notas RIGHT JOIN
Titulo_ubicacion ON notas.Codigo = Titulo_ubicacion.Codigo
WHERE ((Titulo_ubicacion.Apellido1) LIKE '%sql%') ORDER BY
Titulo_ubicacion.Apellido1;"

fp_sDefault = ""
fp_sNoRecords = "Ningún registro devuelto"
fp_iMaxRecords = 0
fp_iTimeout = 0
fp_iCurrent = 1
fp_fError = False
fp_bBlankField = False
If fp_iTimeout <> 0 Then Server.ScriptTimeout = fp_iTimeout
Do While (Not fp_fError) And (InStr(fp_iCurrent, fp_sQry, "%")
<> 0)
```

Encontró abrir comillas, busque las comillas de cierre

```
fp_iStart = InStr(fp_iCurrent, fp_sQry, "%")
fp_iEnd = InStr(fp_iStart + 2, fp_sQry, "%")
If fp_iEnd = 0 Then
    fp_fError = True
    Response.Write "<B>Error de Región de base de datos:
delimitadores de parámetros incorrectos</B>"
Else
    fp_sField = Mid(fp_sQry, fp_iStart + 2, fp_iEnd -
fp_iStart - 2)
    If Mid(fp_sField,1,1) = "%" Then
        fp_sWildcard = "%"
        fp_sField = Mid(fp_sField, 2)
    Else
        fp_sWildCard = ""
    End If
    fp_sValue = Request.Form(fp_sField)
    ' añadir el comodin de busqueda exhaustiva
    fp_sValue = trim(fp_sValue) + "%"
```

Esto encuentra a continuación el valor de campo de formulario nombrado y lo sustituye comillas sencillas duplicadas para todas las comillas sencillas del valor literal de forma que SQL no se confunda al ver comillas sencillas desemparejadas.

```
If (Mid(fp_sQry, fp_iStart - 1, 1) = "\"") Then
    fp_sValue = Replace(fp_sValue, "\"", "\"\"")
ElseIf (Mid(fp_sQry, fp_iStart - 1, 1) = "'") Then
    fp_sValue = Replace(fp_sValue, "'", "'\"")
ElseIf Not IsNumeric(fp_sValue) Then
    fp_sValue = ""
End If
```

```

        If (len(fp_sValue) = 0) Then fp_bBlankField = True

        fp_sQry = Left(fp_sQry, fp_iStart - 1) + fp_sWildcard
+ fp_sValue + Right(fp_sQry, Len(fp_sQry) - fp_iEnd - 1)

'Corrige la nueva posición actual para que sea después del
valor sustituido

        fp_iCurrent = fp_iStart + Len(fp_sValue) +
Len(fp_sWildcard)
        End If
Loop

```

Reasignar a SQL la cadena definitiva

```
sql = fp_sQry
```

Realizar la consulta solicitada

```
RS.Open sql, Conn, adOpenKeyset,adLockReadOnly
```

El número de registros a mostrar será de 5 registros para mayor comodidad del usuario, para que no se fatigue al ver tantos al mismo tiempo

```
RS.PageSize = 5 ' Número de filas por página
```

Si el parámetro Action el cual trae la dirección a la siguiente pagina es vacío y el numero de registros consultados es mayor a 5 entonces se hace una nueva consulta llamando a la siguiente página llamada Resp_gen1.asp

```
if Request("Action") = "" Then
    FormAction = "Resp_gen1.asp"
else
    Response.Redirect("autor.asp?sql=" & Server.URLEncode(sql))
end if

ScrollAction = Request("ScrollAction")
if ScrollAction <> "" Then
    PageNo = mid(ScrollAction, 7)
    if PageNo < 1 Then
        PageNo = 1
    end if
else
    PageNo = 1
end if
RS.AbsolutePage = PageNo
%>
```

La parte principal de la página Html nos muestra una imagen sencilla donde el usuario observará su respuesta de la consulta pedida.

```
<html>

<head>
<title>Resultado de la consulta por autor</title>
<meta name="Microsoft Theme" content="nature 011">
<meta name="Microsoft Border" content="none">
</head>

<body>

<h1><small>Consulta Bibliografica Por Autor: </small></h1>
```

```

<h1><small><small>Resultado de la Consulta</small></small></h1>

<hr align="center">

<h2><small>Página <%=PageNo%></small></h2>

<form METHOD="GET" ACTION="<%=FormAction%>">
  <input type="hidden" name="sql" value="<%=sql%>"><% Do while
not (RS is nothing) %>
<table BORDER="1">
  <tr>

```

Aquí se construye la tabla donde se muestran los resultados solicitados, esta tabla se genera automáticamente dependiendo de la cantidad de campos y registros estipulados

```

<% For i = 0 to RS.Fields.Count - 1 %>
  <td
    bgcolor="#800000"><font
color="#FFFFFF"><b><%=RS(i).Name %></b></font></td>
<% Next %>
</tr>
<%
  RowCount = rs.PageSize
  Do While Not RS.EOF and rowcount > 0
  %>
  <tr>
<% For i = 0 to RS.Fields.Count - 1 %>
  <td><%
    Response.Write RS(i)

  %>
</td>
<% Next %>
</tr>
<%
  RowCount = RowCount - 1
  RS.MoveNext
  Loop
  %>

```

```

</table>
<p><%
    set RS = RS.NextRecordSet
Loop

```

Terminada la tabla de datos y terminada la consulta se procede a cerrar la conexión ODBC.

```

Conn.Close
set rs = nothing
set Conn = nothing
%> </p>
    <p><% if PageNo > 1 Then %> <input TYPE="SUBMIT"
NAME="ScrollAction" VALUE="<%= "Página " & PageNo-1%>"> <% end if
%> <% if RowCount = 0 Then %> <input
TYPE="SUBMIT" NAME="ScrollAction" VALUE="<%= "Página " &
PageNo+1%>"> <% end if %> <input
TYPE="SUBMIT" NAME="Action" VALUE="Otra consulta"> </p>
</form>

<p>&nbsp;</p>
</body>
</html>

```

6.2.2.2 Análisis de la página Resp_Gen2.asp. Esta página omite el procedimiento de insertar el parámetro de búsqueda dentro de la cadena sql, ya que esta cadena viene configurada con la cadena a consultar, esta cadena queda almacenada en el servidor hasta que culmine la consulta por completo.

Cada vez que la consulta tenga mas de 5 registros en una pagina esta se llamará recursivamente.

El cuerpo de la página contiene la siguiente estructura lógica:

```
<%
'---- Valores de CursorTypeEnum ----
Const adOpenForwardOnly = 0
Const adOpenKeyset = 1
Const adOpenDynamic = 2
Const adOpenStatic = 3

'---- Valores de LockTypeEnum ----
Const adLockReadOnly = 1
Const adLockPessimistic = 2
Const adLockOptimistic = 3
Const adLockBatchOptimistic = 4

sql = Request("sql")
If sql = "" Then
    Response.Redirect("autor.asp")
End If
Set Conn = Server.CreateObject("ADODB.Connection")
Set RS = Server.CreateObject("ADODB.RecordSet")
Conn.Open "biblioteca"
RS.Open sql, Conn, adOpenKeyset,adLockReadOnly

RS.PageSize = 5 ' Número de filas por página
```



```

if Request("Action") = "" Then
    FormAction = "Resp_gen1.asp"
else
    Response.Redirect("autor.asp?sql=" & Server.URLEncode(sql))
end if
ScrollAction = Request("ScrollAction")
if ScrollAction <> "" Then
    PageNo = mid(ScrollAction, 7)
    if PageNo < 1 Then
        PageNo = 1
    end if
else
    PageNo = 1
end if
RS.AbsolutePage = PageNo
%>
<html>

<head>
<title>Respuesta general por Autor</title>
<meta name="Microsoft Theme" content="nature 011">
<meta name="Microsoft Border" content="none">
</head>

<body>

<h1><small>Resultado de la consulta</small></h1>

<hr align="center">

<h2><small>Página <%=PageNo%></small></h2>

<form METHOD="GET" ACTION="<%=FormAction%>">
    <input type="hidden" name="sql" value="<%=sql%>"><% Do while
not (RS is nothing) %>
<table BORDER="1">
    <tr>
<% For i = 0 to RS.Fields.Count - 1 %>
        <td
                                bgcolor="#800000"><font
color="#FFFFFF"><b><%=RS(i).Name %></b></font></td>
<% Next %>
    </tr>
<%
    RowCount = rs.PageSize
    Do While Not RS.EOF and rowcount > 0
    %>
        <tr>

```

```

<% For i = 0 to RS.Fields.Count - 1 %>
    <td><%

        Response.Write RS(i)

    %>
</td>
<% Next %>
</tr>
<%
    RowCount = RowCount - 1
    RS.MoveNext
    Loop
    %>
</table>
<p><%
    set RS = RS.NextRecordSet
Loop
Conn.Close
set rs = nothing
set Conn = nothing
%> </p>
<p><% if PageNo > 1 Then %> <input TYPE="submit"
NAME="ScrollAction" VALUE="<%= "Página " & PageNo-1%>"> <% end if
%> <% if RowCount = 0 Then %> <input
    TYPE="SUBMIT" NAME="ScrollAction" VALUE="<%= "Página " &
PageNo+1%>"> <% end if %> <input
    TYPE="SUBMIT" NAME="Action" VALUE="Otra consulta"> </p>
</form>

<p>&nbsp;</p>
</body>
</html>

```

Las demás páginas tienen la misma estructura salvo las páginas estáticas y las dinámicas que piden el parámetro de consulta.

6.3 ESTRUCTURA FÍSICA DEL SITIO WEB DE LA BIBLIOTECA DE LA C.U.T.B CABIBWEB

Todo sitio Web debe estar dentro de un servidor Web, el sitio Web de la biblioteca se encuentra el directorio /Inetpub/WebRoot/Cabibweb (ver figura 29).

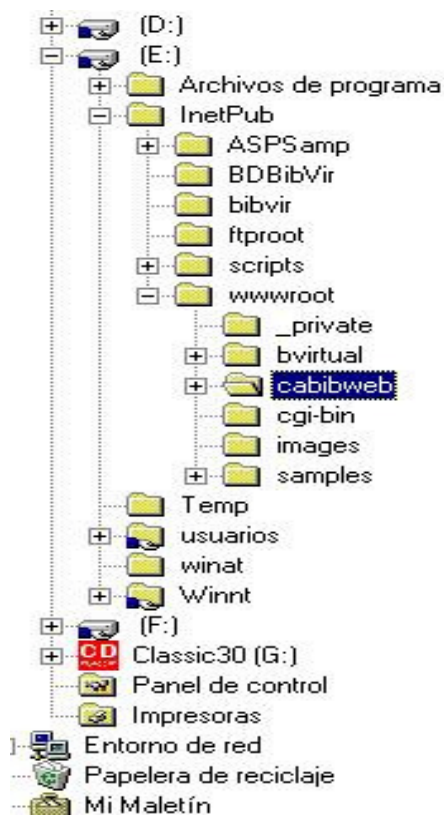


Figura 29: Estructura sitio WEB

En el cual se encuentra la página de bienvenida default.htm.

Las páginas estáticas se encuentran en el subdirectorio páginas estáticas (ver figura 30).

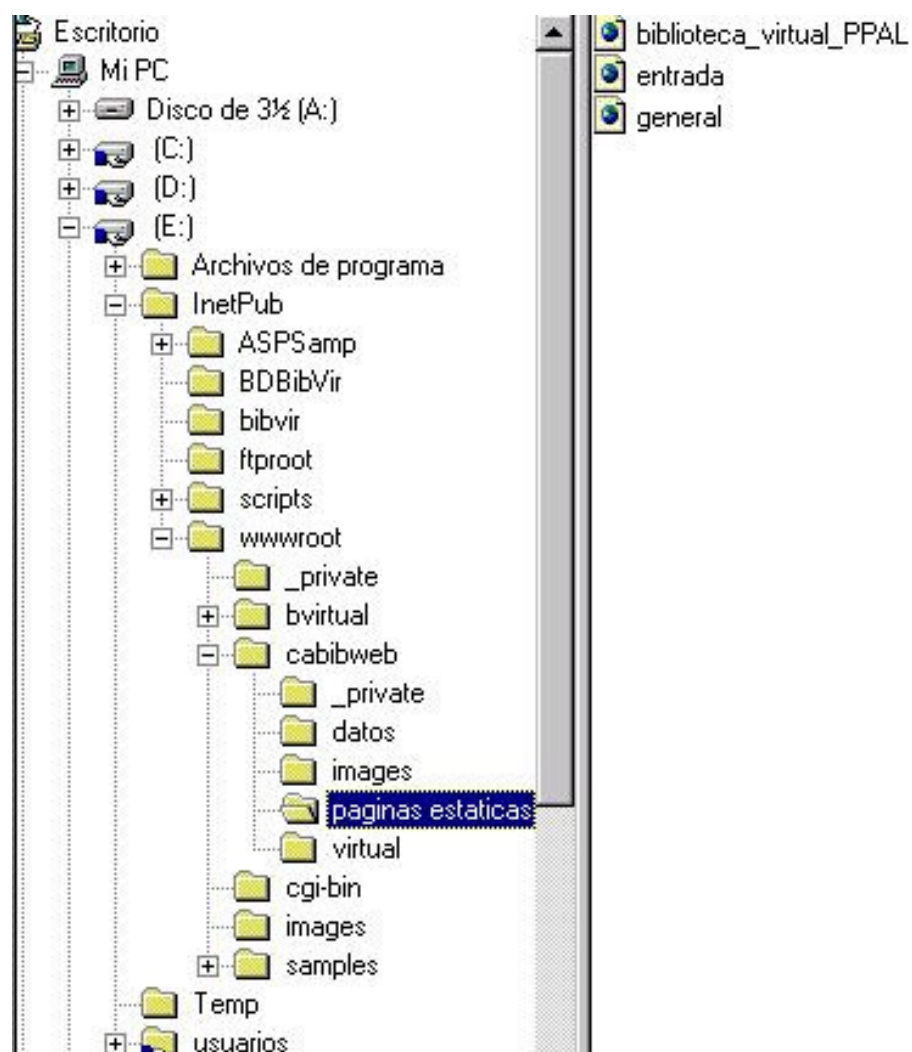


Figura 30: subdirectorio páginas estáticas

Las páginas dinámicas se encuentran el subdirectorio virtual(ver figura 31).

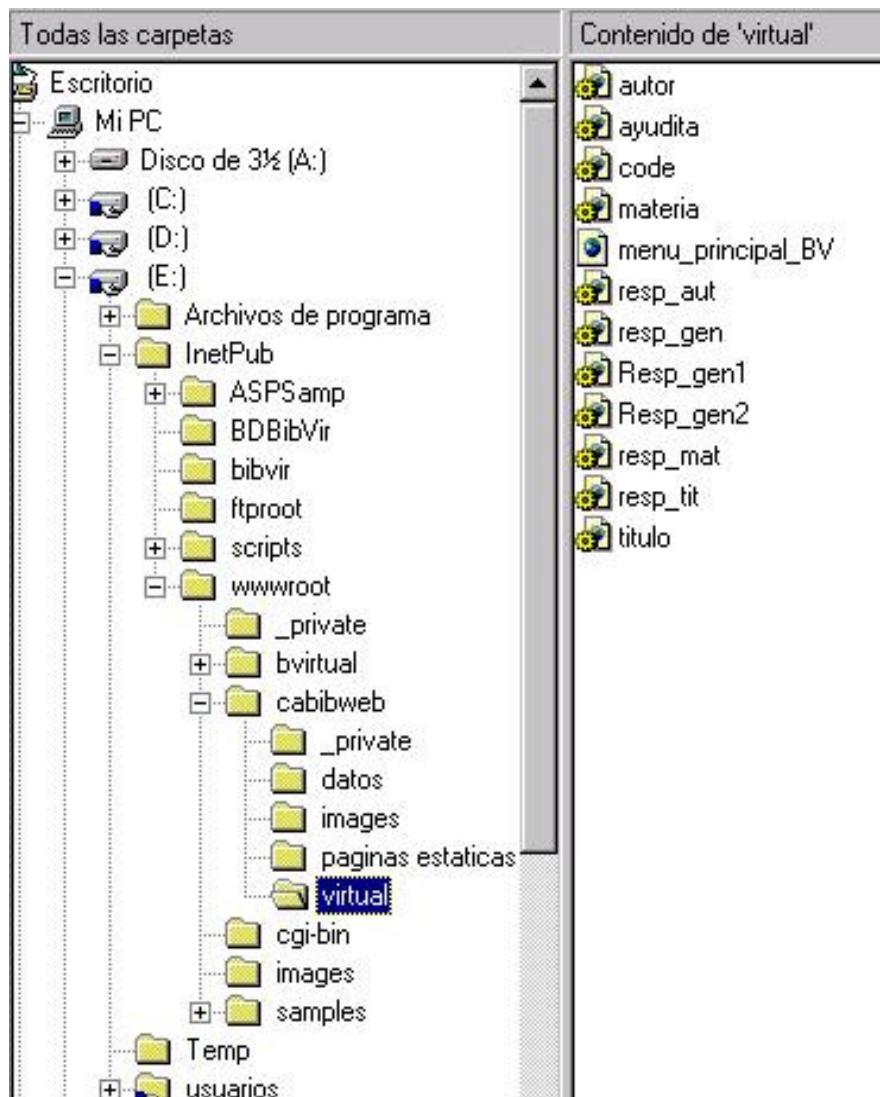


Figura 31: subdirectorio páginas dinámicas

De esta forma queda plasmado y sustentado el sitio Web de la biblioteca de la C.U.T.B CABIBWEB, implementado con la nueva tecnología en materia de Internet como son las páginas ASP.

7. CONCLUSION

El proyecto CABIB, ha contribuido tanto con el área de investigación que involucra la tecnología Internet como también con mejoras sustanciales al actual sistema de consultas Bibliográficas de la Biblioteca de la Corporación Universitaria Tecnológica de Bolívar.

A continuación describimos en forma más detallada las conclusiones del proyecto:

- El proyecto de CABIB, representa un modelo diferente de aplicación para gestionar bases de datos de la C.U.T.B a través del Web, gracias a la investigación que se hizo de las ASP, con lo cual se obtuvo una mejora sustancial, en la medida en que el proceso de consulta bibliográfica en la C.U.T.B puede abarcar a un mayor número de usuarios.
- Con la implementación de la interfaz a través de la plataforma Windows NT de las bases de datos en Informix, se pone de manifiesto una vez mas que la red institucional de la C.U.T.B, es eminentemente escalable, ya que está

en capacidad de soportar sin ningún problema variadas plataformas de sistemas operativos para gestionar tanto servicios Intranet e Internet.

- CABIB ha logrado una gran mejora con relación al servicio que se prestaba anteriormente en la biblioteca de la C.U.T.B en materia de consultas bibliográficas, gracias al ambiente gráfico fácil de manejar, como también el hecho que está implementado en lenguaje visual que es indudablemente mucho más agradable al usuario. También permite a los usuarios un mecanismo rápido y flexible para la consulta de la información en la biblioteca de la C.U.T.B, debido a que los cambios que se hicieron lograron mejorar tanto la búsqueda de la información como el tiempo de respuesta de la misma.
- Con CABIBWEB se logró implementar una aplicación que le permite a la C.U.T.B intercambiar y compartir recursos bibliográficos con otros sistemas que lo requieran en la medida en que se hace a través de la gran red de redes Internet y de una u otra forma ubica a la Corporación Universitaria Tecnológica de Bolívar a la altura de las grandes universidades del país en avances tecnológicos relacionados al Internet.
- Con este proyecto se investigaron todos los aspectos relacionados a las aplicaciones de las páginas de servidor activas ASP, lo cual es un campo

nuevo y de mucha cobertura, flexibilidad, seguridad, y buenos tiempos de respuesta a la hora de gestionar bases de datos; para extraer información bibliográfica contenida en las bases de datos y publicarla a través del Web, por lo que podemos afirmar que estamos aportando un buen material a la biblioteca de la C.U.T.B, que sirva para investigaciones futuras e implementaciones nuevas y mejores para proyectos futuros.

- Las ASP son una de las mejores herramientas de desarrollo de aplicaciones, manteniendo sus conocimientos actuales, sus fuentes de datos, sus componentes y sus aplicaciones puede llevarlas de forma rápida a la Web.

8. RECOMENDACIONES

Después de haber realizado una investigación detallado con respecto al sistema de información bibliográfica para consulta integración en el Web, así como también el desarrollo de una interfaz para la conversión de los archivos planos a una base de datos Access, se llega a las siguientes recomendaciones:

- ⇔ El software de conversión debe ser administrado por una persona no tan experta en el sistema, debido a la gran flexibilidad con que cuenta el software.
- ⇔ El software CABIB es una aplicación que debe seguir siendo explotado e investigado para que pueda realizar préstamos y estadísticos a nivel de Web, y cualquier otro cambio que sea requerido.
- ⇔ Realizar mantenimiento a la base de datos que arroja la interface.
- ⇔ Los nombres de los archivos planos tienen que ser iguales a los expuestos en el esquema del proyecto; es decir, el archivo plano que contiene la información de la tabla Maestro en Informix debe ser bibctb01 y así con los demás archivos planos.

BIBLIOGRAFIA

MARTIN James. Análisis y diseño orientado a objetos. México:Prentice Hall, 1994. 986 Pag.

RUSSEL Charlie. Running Microsoft Windows NT Server. Madrid: Mc Graw-Hill,1995. 795 Pag.

GONZALEZ Alfons. Programación de bases de datos en Visual Basic. México: Alfaomega,1997.567 Pag.

INTERNET, red de redes

COMMER Douglas. Sistemas de Bases de datos. México: Mc Graw – Hill,1994.815 Pag.

TANEMBAUN, Andrew S. Sistemas Operativos Distribuidos. México: Prentice Hall,1996. 825 Pag.

STOLTZ, Kevin. Todo Acerca de Redes de Computación. México: Prentice Hall, 1995. 485 Pag.

SENN, James A. Análisis y Diseño de Sistemas de Información.2ed. México: Mc Graw Hil,1995. 920 Pag.

MCKELVY, Mike. Visual Basic 5. México: Prentice Hall, 1997.845 Pag.

MANUAL DEL USUARIO

INTERFAZ INFORMIX - NT

INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE:

Para instalar el software de conversión de una base datos Informix a una base de datos Access en su equipo de tener en cuenta los siguientes requisitos:

- ⇔ Sistema operativo Windows 95, 98 o NT versión 4.0 o superior.
- ⇔ Un equipo con las siguientes características:
 - ⇔ Computadora Pentium con 166 MHZ o superior.
 - ⇔ Un mínimo de 16 Megabytes de RAM.
 - ⇔ Disco duro con 2.1 Gigabytes mínimo.

Una vez constatado cada uno de estos requisitos se procede a insertar los disquetes en la unidad respectiva.

Nota: Este software debe ser instalado en el servidor de Windows NT en un directorio que tenga permiso total al administrador y permiso de ejecución a los usuario del dominio.

Luego siga los siguientes indicaciones:

1. CONFIGURACIÓN DE LOS CONTROLADORES ODBC PARA EL SITIO WEB

- ✓ Inserte el disco Base de Datos de CABIB en la unidad respectiva.
- ✓ Crear el directorio BDBibVir en la partición C:\ del disco duro del servidor de Windows NT con permiso de solo lectura para usuarios del dominio y control total para el administrador.
- ✓ Copiar la base de datos del disco a la carpeta BDBibVir.
- ✓ Abrir el panel de control de Windows NT y haga doble-click en el icono ODBC de 32 bits. Cuando se abre el administrador ODBC, se ve la pantalla Orígenes de datos, en la cual están todas las fuentes de datos ODBC disponibles.
- ✓ Hacer click en la pestaña DSN de sistema y pulse el botón agregar, luego escoger de la lista el controlador **Microsoft Access Driver (*.MDB)** y oprimir finalizar, aparecerá la pantalla Instalación de ODBC para Microsoft Access.
- ✓ En el cuadro **Nombre del origen de datos** escriba ***biblioteca*** y luego pulse el botón **Seleccionar** para escoger la base de datos a manipular; que en nuestro caso es C:\BDBibVir\Biblivirtual.mdb.
- ✓ Hacer click en aceptar y estará creada la fuente de datos remota.

2. INSTALACIÓN DE LA INTERFAZ INFORMIX - NT

- Inserte el disco número 1 en la unidad A:

- Desde el menú inicio de Windows, escoja el comando Ejecutar.

- Digite en la línea de comando la siguiente ruta: **A:\setup.exe** y pulse la tecla enter.

- Por último siga los pasos que el programa de instalación le indica durante el proceso.

Recomendaciones que deben tenerse en cuenta:

1. Se debe crear un directorio llamado BDBibVir el cual contendrá siempre los archivos planos(En ASCII), que vendran de la base de datos Informix, la cual es matriz de esta información (este directorio es usado para almacenar los archivos planos generados por informix y se debe crear en el disco C).

NOTA: Si usted desea cambiar el nombre al directorio o los archivos allí implicados debera maniobrar en el programa fuente el cual yo ya le hecho entrega.

2. La base de datos nominada BibliVirtual.mdb cuya estructura fue diseñada y creada en ACCESS según argumentos preestablecidos en estructura anexa

entregada con la copia de prueba de archivos en disquete, podrá también cambiarse de nombre según criterios suyos.

3. El procedimiento en cuestión sólo necesita de estar en tiempo de ejecución y presionar el botón que allí se muestra para que el en forma automática ejecute en BATCH (El botón se encuentra en la barra de menú de inicio de Windows NT), la conversión mencionada, la cual hará lo siguiente:
 - i. Tomará cada una de las tablas(Recordset) de la base de datos y le borrará sus registros.
 - ii. Luego le adicionará cada uno de los archivos ASCII previamente actualizados.
 - iii. Le mostrará en el proceso de actualización el número de registros que actualmente se habrá actualizado.

3. PANTALLA DE LA INTEFACE

A continuacion se mostrara la pantalla de la interface cuando esta procesando los datos:



MANUAL DEL USUARIO

CABIB

INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE:

Para instalar el software de consulta bibliográfica CABIB en su equipo de tener en cuenta los siguientes requisitos:

- ⇔ Sistema operativo Windows 95, 98 o NT versión 4.0 o superior.
- ⇔ Un equipo con las siguientes características:
 - ⇔ Computadora Pentium con 166 MHZ o superior.
 - ⇔ Un mínimo de 16 Megabytes de RAM.
 - ⇔ Disco duro con 2.1 Gigabytes mínimo.

Una vez constatado cada uno de estos requisitos se procede a insertar los disquetes en la unidad respectiva.

Nota: Este software debe ser instalado en el servidor de Windows NT en un directorio que tenga permiso total al administrador y permiso de ejecución a los usuario del dominio.

Luego siga los siguientes pasos:

- Inserte el disquete número 1 en la unidad A:
- Desde el menú inicio de Windows, escoja el comando Ejecutar.
- Digite en la línea de comando la siguiente ruta: **A:\setup.exe** y pulse la tecla Enter.
- Por último siga los pasos que el programa de instalación le indica durante el proceso.

PROGRAMA EN EJECUCIÓN

Una vez instalado el software haga doble click en el icono CABIB y saldrá la siguiente pantalla:



En la cual hay que hacer un click en el interior de la pantalla para pasar a la siguiente pantalla.

Esta pantalla esboza los principales aspectos del software como son el nombre de la universidad, el nombre del software, la versión y una advertencia para evitar la copia del programa por parte de personas mal intencionadas.



Este es la pantalla principal, en la cual se encuentra el menú principal y el cuerpo donde aparecerá las ventanas y objetos a utilizar durante la consulta.

El menú principal se compone de cuatro elementos mostrados en forma heurística en el cual los usuarios a través de su percepción sabrán escoger la clase de consulta que desean realizar



Estos botones de izquierda responden a los nombres de Título, Materia o Tema, Autor y Salida.

Al oprimir el botón Título mostrará la pantalla siguiente:

The screenshot shows the CABIB (Centro de Atención Bibliográfica para Consultas) interface. At the top, there is a header with the university logo and a 30th anniversary emblem (1970-2000). The main section is titled 'INFORMACION GENERAL'. Below this, there is a search parameter section with two input fields: 'Título:' containing 'ERGONOMIA APLICADA' and 'Autor(es):' containing 'WARR PETER'. Below the search fields, there is a section titled 'Titulo Buscado' which displays a list of search results. The results are as follows:

Titulo
EL REGIONALISMO ABIERTO EN AMERICA LATINA Y EL CARIBE: LA INTEGRACION ECONOMICA AL SERVICIO DE LA TRANSFORMACION
ELECTRONICA INDUSTRIAL: TECNICAS DE POTENCIA
▶ ERGONOMIA APLICADA
ESTRUCTURA DE DATOS Y DISEÑO DE PROGRAMAS
FM AND REPEATERS
FOXPRO. RELATIONAL DATABASE MANAGEMENT SYSTEM FOR WINDOWS
FUNDAMENTOS DE ENTREVISTAS CONDUCTUAL. REVISION TEORICA

Esta pantalla está compuesta por un campo donde el usuario deposita o digita el nombre del título del libro a consultar, y a medida que el usuario va escribiendo en el cuadro de abajo se van listando todos los títulos relacionados con el buscado y el usuario tiene la libertad de escoger el libro buscado en un tiempo corto.

El usuario también tiene la posibilidad de escoger el libro buscado desplazándose con las flechas ubicadas en la parte superior derecha de la caja de títulos la cual tiene la siguiente apariencia:

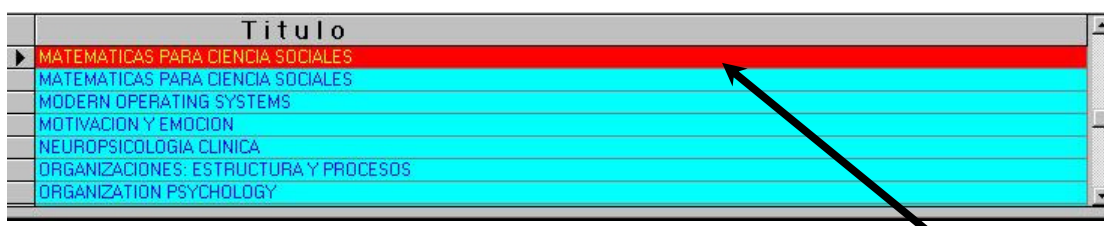


Con ellas el usuario puede ubicarse al principio del archivo, al final, en el libro anterior o en el libro posterior.

A medida que el usuario se desplaza por los libros puede ir observando el nombre de los autores en el campo inmediatamente posterior al campo del título, conformado por una caja múltiple donde se ilustran todos los autores y coautores esta pantalla es la siguiente:



Una vez encontrado el titulo buscado, se hace un doble-click sobre el en la lista de títulos y esta enviará al usuario al otra pantalla donde encontrará una información mucho más rica con respecto a ese libro.

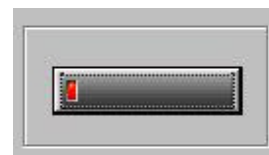


La pantalla de información específica tiene la siguiente apariencia:

Consulta General por Título			
Signatura Topográfica	Ubicación	Número de Ejemplares	ISBN
616.80475//A676	Biblioteca 1	1	968-9263-17-8
Título			
NEUROPSICOLOGIA CLINICA			
Nombre del Autor(es)			
ARDILA ALFREDO			
Materia			
NEUROPSICOLOGIA			
Notas			
 CENTRO DE ATENCION BIBLIOGRAFICA PARA CONSULTAS			

Esta pantalla contiene información de la signatura topográfica del libro, la ubicación del libro, cuantos ejemplares hay en inventario, el numero ISBN, la materia (s) a la que pertenece y una notas importantes acerca de ese libro, como también información de los autores.

Para regresarse al menú principal solo hay que pulsar el botón negro ubicado en la parte inferior derecha de esta pantalla.



Luego de estar de vuelta en el menú principal, si el usuario quiere consultar ya sea por tema o por autor debe realizar los mismos pasos se dieron con el titulo.

La opción de salir esta disponible solo para el administrador del sistema que conocerá una combinación de teclas especiales para salir del programa.

Estos son ejemplos de consultas por autor y materia:

Consulta General por Temas

Signatura Topográfica	Ubicación	Número de Ejemplares	ISBN
658.1511/R628	Biblioteca 3	1	958-601-397-9

Temas

ADMINISTRACION FINANCIERA

Título

GESTION FINANCIERA BAJO INFLACION: INCLUYE LOS DECRETOS 2911 Y 2912 DE 1991

Materia

ADMINISTRACION FINANCIERA

ADMINISTRACION FINANCIERA INFLACION Y CONTABILIDAD

CABIB
CENTRO DE ATENCION BIBLIOGRAFICA
PARA CONSULTAS

Esta pantalla ilustra una consulta específica por tema mostrando el título del libro, si ese tema tiene varios libros el usuario puede seleccionar uno de ellos si le interesa y obtener los datos de ese libro solo pulsando enter.

Ventajas de CABIB

- ✓ Entorno gráfico agradable al usuario.
- ✓ Fácil de manejar.
- ✓ Fácil de entender.
- ✓ Es totalmente heurístico.
- ✓ Ofrece mayor rapidez en la búsqueda de un libro específico.
- ✓ Es mucho más flexible.
- ✓ Expresa información que realmente le interesa al usuario.

CONTENIDO

<i>MANUAL DEL USUARIO</i>	<i>103</i>
<i>Instalación y configuración del software:</i>	<i>103</i>
<i>Programa en ejecución</i>	<i>109</i>
<i>Ventajas de CABIB</i>	<i>117</i>