



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

**“MODELO DEL SISTEMA DE CONTRATACION DE
MEDIOS PARA ARQUITECTURA CLIENTE/SERVIDOR:
VISUAL BASIC Y SQL SERVER, EN UNA
EMPRESA DE PUBLICIDAD”**

TRABAJO DE SEMINARIO :
REDES DE COMPUTADORAS
QUE PARA OBTENER EL TITULO DE:
LICENCIADA EN INFORMATICA
P R E S E N T A:
ZULEMA ORTIZ ORTIZ

ASESOR: ING. JESUS MOISES HERNANDEZ DUARTE

CUAUTITLÁN (ZCALLI, EDO. DE MEX.

1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AVANZADA DE
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

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



DEPARTAMENTO DE
EXAMENES PROFESIONALES

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
PRESENTE.

AT'N: ING. RAFAEL RODRIGUEZ CEBALLOS
Jefe del Departamento de Exámenes
Profesionales de la FES-C.

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario:

Redes de Computadoras. Modelo del sistema de contratación de medios
para arquitectura cliente/servidor: Visual Basic y SQL Server, en
una empresa de publicidad.

que presenta la pasante: Zulema Ortiz Ortiz
con número de cuenta: 9007074-3 para obtener el Título de:
Licenciada en Informática

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO.

ATENTAMENTE.

"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán local, Edo. de México, a 15 de Octubre de 19 97

MODULO:

111

11

111

PROFESOR:

Ing. J. Moisés Hdez. Duarte

Ing. Miguel Alvarez Pasaye

M. I. Gloria Ponce Venegas

FIRMA:

DEP/V08068

**GRACIAS A DIOS POR LA
MARAVILLOSA OPORTUNIDAD
QUE ME DIO DE VIVIR.**

**A MIS PADRES
POR TODO LO QUE ME
HAN DADO, LOS VALORES
QUE ME INCULCARON Y SOBRE TODO
POR SU AMOR.**

**A MIS HERMANOS
JAIME, LULU, ULISES Y HAYDE
POR TODA SU AYUDA Y CARÍÑO.**

**A MIS CUÑADOS GERARDO Y VIKY,
A MIS SOBRINITOS DARIO Y JACKY
Y POR SUPUESTO LOS QUE FALTAN
LOS QUIERO MUCHO A TODOS.**

**A PACO, GRACIAS FLAQUITO
POR TODO EL APOYO Y AMOR QUE
EN TI HE ENCONTRADO Y QUE
ESPERO CONSERVAR NADA MAS...
EL RESTO DE MI VIDA!. TAMCB.**

**A J. MOISES HERNADEZ DUARTE
POR TODO SU APOYO
MIL GRACIAS.**

**A GLORIA PONCE Y MIGUEL
ALVAREZ MUCHAS GRACIAS
POR TODO.**

**A LA LIC. ROSY VALADEZ
POR TODO SU APOYO.**

**A TODOS MIS PROFESORES
POR LOS CONOCIMIENTOS
TRANSMITIDOS.**

**A LA UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO POR
BRINDARME UN LUGAR EN SUS
AULAS Y CON ELLO TENER LA
OPORTUNIDAD DE LOGRAR MI
DESARROLLO PROFESIONAL.**

INDICE

<u>INTRODUCCIÓN</u>	1
<u>CAPITULO I</u>	
<u>ARQUITECTURA CLIENTE/SERVIDOR</u>	1
1.1 CONCEPTO DE CLIENTE/SERVIDOR	1
1.2 MODELOS	3
1.3 TENDENCIAS.	6
<u>CAPITULO II</u>	
<u>IIERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR</u>	9
2.1 MANEJADORES DE BASES DE DATOS.	9
2.2 LENGUAJES DE PROGRAMACIÓN.	15
<u>CAPITULO III</u>	
<u>EL DESARROLLO CON VISUAL BASIC Y SQL SERVER</u>	20
3.1 REQUERIMIENTOS.	20
3.2 MÉTODOS DE ACCESO.	24
<u>CAPITULO IV</u>	
<u>CASO PRACTICO</u>	34
4.1 FUNCIONES Y CARACTERÍSTICAS DE UNA EMPRESA DE PUBLICIDAD.	34
4.2 REQUERIMIENTOS DEL SISTEMA.	38
4.3 DISEÑO DEL SISTEMA.	40

INTRODUCCIÓN

El presente trabajo pretende proporcionar un modelo de contratación de medios que cumpla con las necesidades del usuario en un cien por ciento, trabajando bajo la arquitectura cliente/ servidor.

En el capítulo uno se describe lo que es el modelo cliente/servidor, dado que muchas veces se cree que se está trabajando con una arquitectura cliente/servidor y en realidad no es así, por lo tanto se describirán algunos modelos así como la tendencia de dicha arquitectura, con el fin de dejar bien claro el concepto de la arquitectura bajo la cual se está desarrollando.

Existen diversas herramientas para el desarrollo de aplicaciones bajo cliente/servidor ; en el capítulo dos definiremos las diferentes herramientas que podemos utilizar para crear la interface del usuario y para administrar la base de datos que utilice el modelo.

Como se mencionó en el párrafo anterior, hay diversos manejadores de bases de datos así como lenguajes de programación; para efecto de este modelo se trabajará con Visual Basic como lenguaje de programación para crear la interface; y SQL Server como manejador de la base de datos.

Para llevar a cabo un buen desarrollo se requiere cubrir ciertas expectativas, mismas que son explicadas en el tercer capítulo con el fin de plantear bien los

componentes para el desarrollo de la aplicación, lo cual involucra tanto los requerimientos físicos como los métodos de acceso que Visual Basic puede utilizar para la conexión con el servidor.

Por último, en el caso práctico se presenta la descripción de lo que se realiza dentro de la empresa de publicidad así como los requerimientos tanto en software como en Hardware para la implementación del Sistema de Contratación de Medios. Se menciona también de una manera general como debe funcionar el sistema de acuerdo a las funciones que tienen los departamentos de la empresa.

CAPITULO I

ARQUITECTURA CLIENTE/SERVIDOR

1.1 CONCEPTO DE CLIENTE/SERVIDOR

Cliente/Servidor es un término que describe una tecnología muy general, en la que una tarea de computación dada es distribuida de alguna forma entre más de un punto de ejecución. Una tarea de computación es una función u operación particular que requiere ser procesada. Una tarea distribuida es aquella que es ejecutada en más de un lugar, repartida por varios procesos o máquinas.

Cliente/Servidor es a menudo utilizado para describir un escenario en el que se involucra un servidor de datos centralizado, el cual procesa varias peticiones de clientes, que son típicamente Pc's. La distribución está basada en el hecho de que el servidor analizará y regresará los datos apropiados para el cliente, y éste a su vez se responsabilizará de la presentación de los datos obtenidos al usuario.

Microsoft utiliza el término Cliente/Servidor para describir la relación entre varios componentes del sistema operativo en Windows NT y el Kernel del sistema. En este sentido, está diciendo que el servidor es el Kernel y éste procesa las peticiones de los clientes, por ejemplo, el Graphic Device Interface (GDI).

En general, la arquitectura Cliente/Servidor se define como un sistema que se compone de dos entidades, una llamada Cliente o bien máquina solicitante y la otra parte se conoce como Servidor o máquina proveedora. El primero se encarga de administrar la información, es decir, lleva a cabo la petición y una vez que recibe

dicha información, la utiliza en el momento en que la necesita. El Servidor tiene como función el almacenamiento y distribución tanto de los datos como de las aplicaciones, en otras palabras, está encargado de administrar las peticiones que el Cliente lleve a cabo. También puede participar en la administración de los dispositivos con que se cuenten para satisfacer las necesidades y demandas de los clientes.

El Servidor contiene la información que los Clientes necesitan; por lo que da respuesta a las peticiones de los usuarios, de tal forma que éstos obtengan la información que requieren, de la manera más exacta y rápida posible. Para obtener dicha información, el servidor realiza los cálculos pertinentes y así obtener los datos que le han sido requeridos.

El cliente obtiene los datos sin procesar y lleva a cabo las operaciones necesarias para presentarlo al usuario en un formato que sea claro para éste, a través de una aplicación que contiene la interface con la que dicho usuario realiza sus peticiones.

Anteriormente se mencionó que el Servidor es quien almacena las aplicaciones, sin embargo en algunas organizaciones, tal es el caso de la empresa de publicidad que se presentará en el caso práctico; la base de datos se encuentra en el servidor y la aplicación está instalada en la máquina del cliente, esto quiere decir que los datos están almacenados en tablas remotas en un servidor de base de datos y los clientes envían consultas y actualizaciones al servidor para recuperar los datos que necesiten.

En la arquitectura Cliente/Servidor, se pueden utilizar diversos tipos de equipos así como diferentes sistemas operativos y todo ello dentro de un único ambiente de procesamiento. En este modelo, las operaciones que se llevan cabo para dar respuesta a las peticiones del cliente son jerarquizadas y éstas junto con la red determinan cuales son las necesidades del usuario, de tal manera que solo le envían la información que es realmente útil para él. Cuando se implanta este tipo de arquitectura, se impulsa la utilización de sistemas abiertos puesto que tanto el Cliente como el Servidor tienen la capacidad de correr bajo plataformas diferentes, tanto en hardware como en software, esto permite que las organizaciones adquieran productos de distintos proveedores, sin que por ello el desempeño de las aplicaciones y dispositivos con que cuente la red se vean minimizados de alguna forma.

Actualmente no se puede medir de una manera cuantitativa los beneficios que la arquitectura Cliente/Servidor trae a las empresas, estos más bien se reflejan de una forma cualitativa, es decir, se puede contar con una mejor toma de decisiones o bien con una satisfacción alta por parte de los clientes así como una mejora en la calidad de los productos y servicios, sin embargo no se puede garantizar que haya costos bajos a nivel personal.

1.2 MODELOS

Modelo Tradicional Cliente/Servidor.

Este modelo está basado en la distribución física de los recursos. Cuando una aplicación es ejecutada, la localidad en la que se encuentran almacenados los

recursos debe de ser transparente y por tanto al mover la localización de uno de los componentes, la forma en que la aplicación está desarrollada no debe de sufrir modificaciones.

En este modelo el Cliente es el encargado de la presentación de funciones al usuario, en tanto que el servidor es el límite organizacional entre la información de la organización y los procesos del negocio.

Modelo del Sistema Centrado en los Datos.

En este modelo se implementaron grandes aplicaciones monolíticas, a través de la aproximación procedural. "Las consideraciones de manejo se encuentran en la lógica del programa principal, el cual dicta como el usuario debe operar y suministrar su información. Aún más, cada programa dicta la forma en que el usuario navega entre las pantallas."¹

El siguiente esquema describe el modelo:

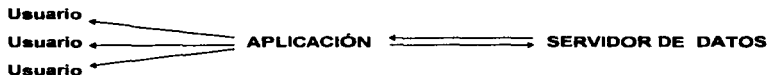


Figura 1. Modelo del Sistema Centrado en los datos

¹Seminario Desarrollo Cliente/Servidor con Visual Basic y SQL Server, Editorial Ddemesis, México 1994, p.4

En este tipo de aplicaciones, cada una de ellas contiene una sola interface con el usuario que está diseñada para acceder la base de datos o el servidor. Esto era funcional cuando los sistemas de información contaban con requerimientos limitados del corporativo, y en el que el objetivo principal era el manejo de los datos.

Modelo del Sistema por Servicios.

En este modelo los usuarios comenzaron a manipular los sistemas, es decir, paso la etapa en la que eran más bien los sistemas quienes de alguna manera manipulaban al usuario, es decir, dejaron de controlarlo. Para la implementación de este modelo se necesitaba contar con una infraestructura bien definida que permitiera al usuario obtener el mayor provecho de los recursos con que contase.

Al utilizar este modelo los usuarios accesan a los recursos del sistema de una manera mucho más flexible.

Modelo de Servicios Centrados en el Usuario

De acuerdo a este modelo un desarrollo en Cliente/Servidor debe tomar en cuenta los siguientes puntos:

1. La interface del usuario y la navegación en el programa.
2. La funcionalidad del negocio.
3. El acceso a los datos.

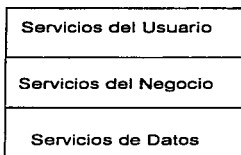


Figura 2. Modelo de Servicios Centrados en el Usuario.

1.3 TENDENCIAS.

Principios.

La arquitectura Cliente/Servidor empezó a crecer desde que Oracle tuvo éxito a finales de la década de los 80's. A partir de ese momento diversas cosas comenzaron a cambiar. En primer lugar, las tecnologías fundamentales y la definición de lo que es un servidor. Por ejemplo, regresando a los primeros servidores donde las minicomputadoras y los clientes eran PC's con interfaces en modo texto y CPU's relativamente lentos.

A través del tiempo los servidores han llegado a ser más accesibles para los usuarios, departamentos pequeños o grupos de trabajo de negocios. Esto hizo que se creara la necesidad de que se simplificara la instalación, mantenimiento y en general la ejecución de una aplicación Cliente/Servidor. La corporación Centura Software Corporation fue la pionera en crear un grupo de trabajo de bases de datos: SQLBase.

Otras compañías han reconocido este mercado: Microsoft produjo Microsoft SQL Server; ORACLE introdujo una tecnología más accesible de instalar y definir llamada ORACLE Workgroup 2000; y Sybase a través de su adquisición de Power Soft sacó SQL Anywhere 5.0 y su grupo de trabajo de base de datos en 1995.

Estos grupos de trabajo de bases de datos se han hecho exitosos y populares, con ello han contribuido al éxito de las aplicaciones Cliente/Servidor y de los sistemas operativos de red tales como Windows NT Advanced Server, UNIX y Net Ware de Novell.

Perspectivas.

Actualmente ya se está hablando de la nueva forma de aprovechar el concepto Cliente/Servidor, el cual es a través de Intranet que se define como una red para una empresa que contiene la información de la misma.

La implementación de Intranet utiliza la tecnología de Internet, es decir, realiza un acceso de datos a través de búsquedas en un servidor. Entre la tecnología que utiliza se encuentra:

- Protocolos de comunicación. A través del cual se conectan y comunican redes y computadoras.
- Transferencia de archivos. Transferir archivos entre locaciones punto a punto.
- Correo. Comunicación directa punto a punto entre individuos o grupos.
- Web Browsing. Acceso de información de una a muchas bases de datos, de acuerdo a la demanda.

- Emulación de Terminal. Es el acceso a la infraestructura de las aplicaciones con que se cuenta.
- Interfaces de Usuario. A través de dichas interfaces, se presenta al usuario la forma a través de la cual se obtendrán los datos. Dicha interface debe presentarse de manera transparente y amigable.

El uso del concepto de Intranet se perfila realmente útil, puesto que además de cubrir la comunicación corporativa, puede poner a disposición de los empleados la información de uso cotidiano, tales como, los manuales de procedimientos, las listas de precios, la información de proveedores, etc.

Esta tendencia de aprovechar el concepto Cliente/Servidor a través de Intranet, se da en el momento en que se considera que el problema de tener que desarrollar un cliente y su código, así como un servidor y sus propios códigos, se resuelve con Internet e Intranet, donde el cliente es universal y es un browser (buscador) con un lenguaje también universal, lo mismo que el servidor.

Para la implementación de un Intranet, las organizaciones deben realizar básicamente, un análisis del impacto de la solución dentro del plan de negocios, así como el costo beneficio que dicha solución le representará a la compañía.

CAPITULO II HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR

En el ámbito del Software existen diversas herramientas para el desarrollo de las aplicaciones bajo Cliente/Servidor. A continuación se presenta el concepto de algunos manejadores de bases de datos y también lenguajes de programación.

2.1 MANEJADORES DE BASES DE DATOS.

Sybase Sistema XI.

Originalmente fue desarrollado para correr bajo plataforma UNIX. Sin embargo los productos actuales corren en diferentes plataformas de hardware, desde Intel Pentium Pro hasta Alpha Dec, y en diversos sistemas operativos, desde OS/2 Warp de IBM hasta SCO UNIX de UNIX.

En 1989, Sybase anunció su tecnología: Las interfaces de programación de aplicaciones (API's), Open Client y Open Server. Estas dos API's provieron a Sybase con las bases para que desarrollara una serie de herramientas que permiten la conectividad entre cualquier clase de dispositivo de cliente, como una PC, Macintosh ó UNIX y cualquier clase de datos.

La interface Open Client, reside en las computadoras cliente en una red y está usado como una interface para la transmisión de protocolos de una red Sybase, conocida colectivamente como una Netlibrary. Open Client, es una interfaz genérica que todos los programadores pudieran escribir sin necesidad de saber cualquier

cosa acerca de las bases del transporte de red. Netlibrary está escrito directamente para cada tipo de protocolos a bajo nivel, como un Microsoft TCP/IP y el SPX de Novell.

Componentes de Sistema XI.

Sistema XI de Sybase es más que solo una manera de guardar datos en una base de datos relacional de SQL, Sistema XI es una familia de productos que trabajan en conjunto para proveer un sistema completa de administración de datos.

** Data Server.*

Es el componente principal que controla toda la administración de datos y la administración de usuarios. El data Server es el almacén central para todas las bases de datos. El Data Server es multihilado y puede manejar múltiples y simultáneas peticiones de diferentes clientes en una red.

**Backup Server.*

El backup server es una máquina/herramienta que ejecuta backups y restores del data server. En un ambiente normal, el backup server siempre está corriendo mientras el data server está corriendo.

El backup Server es un dispositivo independiente que soporta las más importantes formas de medios para respaldar y recuperar, incluyendo todas las formas de cinta; directo al disco duro, respaldos remotos a otros servidores y dispositivos ópticos-magnéticos.

**Audit Server*

Trabaja como un auditor de transacciones colocado entre el Data Server y los clientes que ejecutan operaciones. Con él se pueden monitorear a todos los usuarios, así como los reportes o herramientas de petición de adhoc.

El Audit Server tiene su propia base de datos para grabar todos los eventos del servidor de base de datos y los datos contienen niveles variantes de detalle de auditoría disponibles para peticiones y respaldos, como en cinta. Audit Server está considerado como una herramienta esencial para las compañías e instituciones que tienen especial cuidado en el área de la seguridad y que requieren de un análisis detallado de todas las operaciones que se lleven a cabo en la base de datos.

***Secure SQL Server.**

Es un producto diseñado para mantener la seguridad en las organizaciones. Implementa todos los criterios requeridos por el certificado B-1 del gobierno de los Estados Unidos en cuanto al entorno de la seguridad. Sigue los lineamientos establecidos por lo que se conoce como el Libro Naranja, DOD 52.00.28-STD, el Departamento de Defensa. Implementa el esquema de control de acceso para reforzar la seguridad disponible en el SQL Server.

***Replication Server.**

Es un producto que ejecuta replicaciones de datos de un servidor de bases de datos a otro a través de la red. La replicación permite distribuir datos entre servidores para los beneficios de redundancia de datos y también para el balanceo de las búsquedas en los servidores a través de una red grande.

Trabaja a través de la instalación de un proceso a nivel de servidor para leer el log de transacciones de la base de datos. Cuando una acción es confirmada y

escrita al log, el Replication Server viaja a través de la red y aplica la misma transacción en los otros servidores de bases de datos que están suscritos a los datos.

***MPP de Sybase.**

Significa Proceso Masivo en Paralelo, es soportado por Sybase y está diseñado para manejar bases masivas en exceso de un terabyte en tamaño y soporta arquitecturas de hardware con cientos de CPU's que trabajan en paralelo para ejecutar peticiones a una velocidad óptima.

***IQ de Sybase**

Es un componente del Sybase XI que está diseñado para Data Ware Housing. Usa un propio algoritmo de almacenamiento de datos binarios que es veloz en datos arriba de quinientas veces de funcionamiento comparado con Sybase SQL Server.

SQL Server

Microsoft SQL Server es un manejador de Base de Datos Relacional, está basado en la arquitectura Cliente/Servidor, que soporta aplicaciones de procesamiento de transacciones y además brinda soporte en la toma de decisiones.

SQL Server ofrece una gran seguridad y garantía en el procesamiento de transacciones. Es considerado una excelente herramienta para el desarrollo de aplicaciones dorsales en estaciones de trabajo, minicomputadoras y mainframes, gracias a sus capacidades avanzadas entre las que se encuentran los triggers, que son fragmentos de código que se disparan cuando se quiere hacer alguna transacción en la base de datos, tal como insertar, actualizar o borrar registros,

también se contemplan los procedimientos almacenados las reglas y omisiones así como la tecnología de llamado a procedimiento remoto.

En 1993, surge una versión de SQL Server diseñada especialmente para el sistema operativo Windows NT, el cual ofrece un excelente rendimiento así como seguridad y escalabilidad, estas características le hicieron muy adecuado para las grandes organizaciones.

A través de la explotación de las capacidades de Windows NT, como son el procesamiento simétrico, multitareas, multi-usuarios y la seguridad avanzada, SQL Server provee una plataforma sumamente avanzada para las aplicaciones de acceso a datos de misión crítica. Dado que Windows NT puede operar en diferentes tipos de hardware como son una PC con procesador Intel hasta servidores con múltiples procesadores RISC, las soluciones implementadas en SQL Server pueden ser adaptadas para resolver diversos problemas que se presenten en la organización.

Las características principales de SQL Server son:

- Realiza una administración remota de los recursos con que cuenta el sistema.
- Tiene una arquitectura de 32 bits
- Manejo de transacciones con capacidad de roll-forward y roll-back lo cual permite la recuperación de transacciones al reiniciar el sistema.
- Una arquitectura de un solo proceso pero multitarea que asegura un desempeño eficiente del sistema para muchos usuarios con un bajo consumo de recursos.
- El reforzamiento centralizado de la integridad de los datos que minimiza el mantenimiento, programación y reduce los costos brindando la seguridad de que los datos son correctos.

- Permite interface de programación remota como ODBC (Open Database Connectivity) , por ejemplo.

SQL Server permite reducir aplicaciones en minicomputadoras y mainframes y con ello reducir los costos. Y dado que SQL Server es compatible con Sybase, se pueden escalar soluciones de hardware desde PC hasta mainframes. Puede decirse que SQL Server es la opción más flexible y poderosa en lo que se refiere a manejadores de bases de datos.

ORACLE

ORACLE es un manejador de base de datos y no una base de datos. ORACLE crea bases de datos relacionales, en las cuales los datos son organizados dentro de tablas de renglones y columnas. Cada tabla contiene una serie de elementos que componen una clase. Las columnas contienen los atributos de cada elemento. La intersección de un renglón con una columna contiene la información en cuanto al aspecto de un elemento en particular.

ORACLE tiene las siguientes características:

- Es poderoso
- Almacena la información propiamente categorizada e interrelacionada con otra información.
- Presenta los datos en orden razonable y con ellos pueden realizar diversos cálculos o análisis.
- Una de las características más importantes es la portabilidad. Las versiones de ORACLE están disponibles para muchas microcomputadoras, minicomputadoras y

superminis, así como algunos mainframes. Como resultado de esto, se puede construir un sistema de base de datos en una computadora pequeña y transportarlo a computadoras más poderosas. Por lo tanto, ORACLE puede correr el mismo sistema de base de datos lo mismo en una en una red con computadoras de diferentes tipos que en una computadora personal y el cambio de una a otra no necesita alteraciones al sistema para que éste funcione.

2.2 LENGUAJES DE PROGRAMACIÓN.

Power Builder

Power Builder es descrito como una aplicación cliente/servidor basada en gráficos en PC's. Este lenguaje provee un fácil diseño de pantallas, además de procedimientos lógicos que encajan en las secuencias SQL. Es semiorientado a objetos, permitiendo a los programadores heredar ventanas.

La manera en que Power Builder trabaja es la siguiente:

Los programadores diseñan pantallas utilizando una paleta de objetos gráficos denominados controles, tales como botones, listas, cuadros de selección, botones de radio y ventanas de datos.

La ventana de datos es una entidad especial en Power Builder, son usadas para la manipulación, modificación y presentación de los datos. Actualmente hay dos tipos de ventanas de datos, una es denominada Control y la otra es un Puente entre el control y el almacenamiento de los datos. Los objetos pueden traer datos vía SQL estático, SQL con argumentos o a través de stored procedures.

SQLWindows de Gupta

SQLWindows es un ambiente de desarrollo para aplicaciones gráficas. El éxito de Power Builder forzó a Gupta a retomar una idea que tenía hace tiempo. En los primeros días de PowerBuilder, Gupta contribuía para que la programación no fuera para novatos. Si los programadores querían adoptar esta clase de género, ellos tenían que aprender las bases, y esto incluía SQL. Gupta se burló de los intentos de PowerSoft por minimizar la exposición de los programadores al lenguaje de Query. Por otro lado SQLWindows no pareció proveer mucha ventaja con la combinación de C y SQL.

Todo ha cambiado con la versión 4.0 de Gupta. Tiene un requerimiento simple de acceso de datos. Gupta clamó que las aplicaciones sofisticadas maestras y detalladas pueden ser construidas con poco conocimiento de programación de base de datos.

SQLWindows consistentemente ganó puntos por su robusta facilidad de multidesarrolladores. Contando con un diccionario de almacenamiento central que usa SQLBase, SQL Server de Gupta a bajo nivel. También recibe un alto precio por su completa implementación de fundamentos de programación orientada a objetos, tales como polimorfismo; encapsulación y múltiples herencias. Mientras los programadores experimentados tienen este conocimiento, los novatos no necesitan preocuparse de tales cosas. Los sistemas pilotos pueden ser construidos en pseudoestilo orientado a objetos, permitiendo a los programadores embestir en las aguas de OOP a su propia velocidad.

Generalmente, SQLWindows requiere que los programadores escriban más código que en Power Builder, pero brinda una mayor facilidad en la administración de proyectos.

Visual Basic

Visual Basic es una herramienta de desarrollo de aplicaciones en Windows bajo la arquitectura Cliente/Servidor.

En algunos casos no es pertinente comparar a Visual Basic con Power Builder y SQLWindows ya que ambos productos están dedicados exclusivamente a la tarea de construir aplicaciones Cliente/Servidor. Por otro lado, Visual Basic se puede utilizar para construir prácticamente cualquier aplicación para Windows que se requiera.

La programación de Visual Basic es parecida a Power Builder y SQLWindows. Se seleccionan algunos objetos gráficos de una barra y se colocan en una forma o pantalla. Dichos objetos cuentan con métodos, eventos y propiedades a través de las cuales se puede modificar el tamaño, el color, tipo de letra, etc. Al dar doble click en alguno de estos objetos aparece un área de construcción de fragmentos de código, que nos permite colocar las sentencias que se deben cumplir de acuerdo al evento que se requiera.

Visual Basic ha causado gran sensación en los negocios de computadoras. Compuesto por pedazos de código ejecutable llamado VBX's. Dichas piezas de código actualmente llegaron a ser parte del ambiente de desarrollo de Visual. La

funcionalidad de estas herramientas salvan a los programadores de escribir cientos de líneas de código.

Microsoft, pretende que los programadores utilicen OCX's (Ole Custom Controls), para ello está intentando persuadir a la gente de dejar los VBX's.

Los OCX's son los nuevos controles a 32 bits destinados para la actualización de las versiones de Visual Basic. Estos controles pueden ser encajados en cualquier aplicación disponible para actuar como un contenedor OLE y son más versátiles que los VBX's.

JAVA

Java es el lenguaje actual en que se desarrollan aplicaciones para Internet/Intranet. Es un lenguaje universal y corre bajo cualquier plataforma, esto representa una gran ventaja, ya que no se tienen que capacitar gente en diversos lenguajes.

Mientras el Internet público es lo más glamoroso, los Intranet's pueden ser la más importante oportunidad de negocios. Una Intranet es una forma de Cliente/Servidor (como lo es Internet y el World Wide Web) y puede ser mezclado con otras arquitecturas Cliente/Servidor. La Intranet tienen notables ventajas, como lo es el correr en clientes con diversas plataformas, capacidades de multimedia y la capacidad de soltar a cualquier elemento de una red distribuida de datos y como desventajas podemos mencionar el que no cuenta con una conexión persistente y tampoco tiene la noción de transacciones.

Java anima páginas en el Web y las hace interactivas. Con hipertexto, las bases para la organización de la información en el Web, se puede seleccionar la información que se requiere. Los programadores pueden crear algo de interactividad de programas gateway que usan archivos de hipertexto como interfaces. Cuando se utiliza una página WEB como un programa de gateway, se puede acceder a bases de datos y recibir una respuesta a las peticiones hechas.

Java añade a estas posibilidades de comunicación, la opción de distribuir contenidos ejecutables. Esto da a los proveedores de información de WEB, la oportunidad de crear una página de hipertexto que ajusta usuarios en interacción continua, en tiempo real y complejo. Este contenido ejecutable es literalmente bajado a las computadoras del usuario, una vez que esto se ha realizado, dicho contenido ejecutable puede correr una animación, ejecutar un cálculo o guiar al usuario a través de más información en sitios remotos en red.

Una metáfora para el hipertexto es que éste ofrece una página de información visualmente estática, la cual puede incluir texto, gráficos, sonido y vídeo. La página de hipertexto puede tener "profundidad" donde contiene hiperligas conectando a otros documentos o recursos.

Java transforma esta metáfora de página estática en una dinámica. La información en una página Java en el WEB, no tiene que ser visualmente estática o limitada a predefinidas formas de interactuar con los usuarios. Dichos usuarios encuentran en los programas de Java una gran variedad de modos interactivos, limitada solamente a la imaginación y habilidad del programador. De esta manera, Java transforma una página de hipertexto en un escenario completo con posibilidades de que los actores y jugadores aparezcan y las cosas ocurran.

CAPITULO III

EL DESARROLLO CON VISUAL BASIC Y SQL SERVER

Para el desarrollo de una aplicación utilizando Visual Basic y SQL Server se deben cumplir ciertos requerimientos físicos, además de que se debe tomar una decisión en cuanto al método de acceso que se utilizará.

3.1 REQUERIMIENTOS.

La Red.

Para sacar el mayor provecho del tiempo se debe contar con una conexión en red, ya que si no se contase con ella se consumiría gran parte del tiempo que se dedica al proyecto en copiar el código a cada una de las computadoras.

Al estar conectados en red, los programadores tienen la posibilidad de acceder archivos comunes y además compartir el código del sistema. Para ello el administrador de la red, debe otorgar los permisos necesarios a los participantes en el desarrollo.

Versiones

Una versión es aquella que establece el estado en el que se encuentra una aplicación.

El primer estado que toma una aplicación se conoce como prototipo, y se caracteriza porque contiene únicamente el código necesario para la presentación de las ventanas del usuario. El objetivo de esta versión es mostrar al usuario un panorama general del funcionamiento de la aplicación. Con la versión prototipo se puede llevar a cabo un estimado del tiempo total de codificación.

El segundo estado de las aplicaciones se denomina como versión Alfa, en estas versiones ya existe un funcionamiento parcial de la aplicación, esto incluye la codificación de una parte de las especificaciones funcionales del sistema; una vez generada la versión pasa a revisión a las personas que revisaran el funcionamiento, las cuales son totalmente ajenas al desarrollo de la aplicación.

El tercer estado es nombrado versión Beta, en esta versión, se cubre el total de las especificaciones funcionales, sin embargo, se carece de las pruebas necesarias para asegurar totalmente el funcionamiento de la aplicación.

En esta etapa se debe tener muy en cuenta el tipo de sistema que se está desarrollando puesto que de esto depende la magnitud del error, además de ello, se debe considerar el tiempo promedio en el que se genera un error. Tomando en cuenta estos dos puntos se hace la generación de versiones Beta hasta que se considere que el funcionamiento de la aplicación es total.

Una vez que se concluyen las pruebas se genera la versión de Liberación. En dicha versión, el usuario ya puede hacer uso de la aplicación sin ningún problema, Toda aplicación debe contar con ayuda para el usuario y con sus respectivos manuales: el de usuario y el técnico.

Organización.

Para tener un mejor control de los archivos que componen al sistema es necesario contar con un directorio propio para dichos archivos y para los subdirectorios de cada una de las versiones generadas, por ejemplo. Es recomendable crear un subdirectorio para almacenar todo lo referente a la aplicación, en cuanto a desarrollo y otro para guardar todo lo que pertenezca a documentación del sistema.

Estándares de codificación.

Es importante que se establezcan estándares de codificación para que los programadores "hablen el mismo idioma", por así decirlo. El equipo de desarrollo debe de crearse un hábito para utilizar los estándares dado que esto puede reportar un buen aumento en la calidad y productividad en el proceso de codificación.

El uso de estándares tiene ciertas ventajas como son:

- El programador no pierde tiempo en tratar de descifrar el código que alguien haya realizado en alguna forma o control.
- Cuando un programador está familiarizado con los estándares, el cambio de un proyecto a otro no representa un alto costo de aprendizaje
- Permite identificar de manera más rápida y fácil el tipo de una variable, la clase a la que pertenece un control, entre otros.

Además de contar con un estándar en cuanto al código, es importante también contar con estándares para la interface, puesto que esto representa una

gran ventaja para el usuario puesto que le permite familiarizarse rápidamente con las partes que componen el sistema además de ello cuando se añaden nuevos módulos al sistema al usuario le resulta mucho más sencillo familiarizarse con las nuevas ventanas.

A través de los estándares en la interface, la empresa puede obtener ciertos beneficios tales como el establecimiento de reglas a las que los programadores deben atenerse en la creación de las ventanas que compondrán el sistema.

Aun cuando los estándares varían de una organización a otra, es importante que para la creación del sistema se tomen en cuenta ciertas medidas con el fin de que la aplicación no aparente haber sido desarrollada de manera aislada; por ejemplo, se puede crear una ventana estándar lo cual permitirá a los programadores determinar las características generales de la ventana dentro de la aplicación. En dicha creación, se toma en cuenta tanto las formas como los controles.

Uso de correo electrónico.

Cuando se cuenta con correo electrónico, se ahorra un tiempo considerable ya que no hay necesidad de llevar a cabo reuniones de trabajo para asuntos que no sean de gran relevancia.

Hardware.

Es importante que todos los programadores cuenten con la misma plataforma de hardware, cuando menos en cuanto al monitor y el controlador de vídeo, dado

que para las pruebas de la interface se deben realizar en el mismo ambiente con el fin de que los resultados sean lo más uniforme posibles.

3.2 MÉTODOS DE ACCESO.

Para acceder a SQL Server como aplicación cliente, se utilizan las llamadas API's, que son Interfaces de Programación para Aplicaciones y que pueden describirse también como librerías de ligado dinámico o bien DLL's. Dichas librerías se usan para comunicarse con el manejador de Bases de Datos, en los siguientes casos:

- Si se requiere iniciar un canal de comunicación con el servidor.
- Cuando se envían peticiones al servidor.
- Para procesar los datos obtenidos de las solicitudes al servidor.
- Cuando se finaliza la comunicación con el servidor.

Cuando se establece la conexión con SQL Server, la aplicación cliente trabaja con él utilizando el lenguaje Transact-SQL, el cual es un lenguaje que permite enviar instrucciones al manejador de Bases de Datos. Cuando se está ejecutando la aplicación cliente, el servidor se encuentra en un estado de espera, una vez que la aplicación necesita información, el sistema operativo se encarga de activar a SQL Server para que reciba el lote de peticiones de las cuales debe revisar que cumplan con los siguientes requisitos :

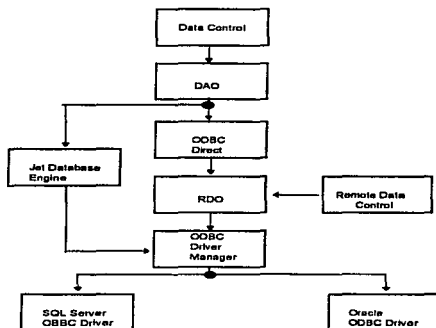
- Que la sintaxis sea correcta.

- Que los objetos a los que hace referencia la sentencia (tablas, campos, etc.) existan en la base de datos.
- Que el usuario cuente con los permisos para el acceso a la información.

Cuando alguno de los requisitos no se cumple SQL Server notifica al programa cliente que se ha fracasado, en caso contrario, ejecuta una a una las sentencias y regresa el bloque de resultados.

Actualmente Visual Basic cuenta con varios métodos de acceso remoto a bases de datos bajo Cliente/Servidor. Algunas de estas técnicas utilizan el Data Control o Data Access Object, o el Remote Data Control y el Remote Data Object, mientras que otras utilizan un acceso de datos a través de Application Programming Interface (API), tal como VBSQL o el ODBC API.

Figura 3. Métodos de acceso.



Jet Database es la base de datos que crea el Jet Engine para almacenar los movimientos que se lleven a cabo, a pesar de que esta base de datos almacena los movimientos, no es indispensable ya que como se describe adelante existe un método de acceso que no invoca al Jet Engine.

Para tener acceso a través de ODBC a SQL o SYBASE es necesario contar con el driver correspondiente al que vayamos a acceder, el driver es una librería que se carga en memoria y que realizará el enlace entre la aplicación y la base de datos.

DAO. (Data Access Object)

Además de tener acceso a manejadores de base de datos, Visual Basic es por sí mismo un sistema controlador de Base de Datos, es decir, cuenta con controles que tienen la capacidad de conectarse e interoperar con el Jet Engine por sí mismos.

Visual Basic tiene acceso transparente a los siguientes tipos de Base de Datos:

- El formato nativo de archivos de Microsoft Access.
- Bases de Datos externas tales como: dBase, Btrieve, Paradox y FoxPro
- Bases de Datos externas ODBC como son: SQL Server y ORACLE.

Para la manipulación de Bases de Datos a través de DAO, se requiere del objeto Recordset, a través de este objeto tenemos acceso a la información almacenada en la base de datos. Generalmente es creado por medio de una sentencia Select con la que se obtendrán los datos de una o más tablas de la Base

de Datos y la información se guardará en el recordset semejando una tabla para la representación temporal de los datos.

RDO. (Remote Data Object)

Este método de acceso, depende de la API ODBC, el driver ODBC escogido y la Base de Datos . Con RDO es posible ejecutar la mayoría de las funciones del API de ODBC directamente.

Usando RDO se pueden ejecutar "queries" con o sin parámetros, crear sencillos o múltiples juegos de resultados y procesar los mismos desde el query por medio de un código independiente de bases de datos . RDO incluye un diversos objetos, propiedades y métodos específicamente diseñados para trabajar con Stored Procedures y sus argumentos, parámetros de entrada y salida y valores de status de retorno.

El modelo RDO tiene las siguientes características:

- Es un modelo de programación que realiza accesos a 32 bits.
- Opera con programación conducida por eventos.
- Soporte para trabajar en modo síncrono, asíncrono, conexiones conducidas por eventos y queries.
- Administrador avanzado para el almacenamiento de los datos obtenidos.

ODBC. (Open Database Connectivity)

El Open Database Connectivity (ODBC), representa un acceso viable para conectarse a bases de datos externas desde Visual Basic. El método de acceso a través de ODBC puede utilizarse tanto con SQL Server como con ORACLE y algunos otros manejadores de bases de datos Cliente/Servidor.

La interface ODBC accesa a los datos usando de Sentencias SQL como un estándar. Esta interface permite una máxima interoperabilidad, esto es, una aplicación puede acceder a distintos manejadores de bases de datos. Esto permite que las aplicaciones sean desarrolladas de tal modo que no se piense en un manejador de bases de datos específico. Para lograr la interface con otros manejadores de bases de datos únicamente se debe conseguir el driver correspondiente para ligar la aplicación.

La interface de ODBC tiene las siguientes características:

- Cuenta con una librería de llamadas que permiten a una aplicación acceder a la información de un servidor, ejecutar las sentencias correspondientes y regresar los datos requeridos.
- Tiene un conjunto estándar de códigos de error.
- Representación estándar para los tipo de datos.
- La interface es flexible, por ejemplo se pueden construir las sentencias SQL explícitamente desde el código fuente o bien ser construidas en tiempo de ejecución.
- Puede utilizarse el mismo código objeto para el acceso a diferentes manejadores de Bases de Datos.

- Los valores de datos pueden ser contenidos dentro del formato más conveniente para su operación dentro de la aplicación y de esta manera ser enviados o capturados como resultado de una petición.

La arquitectura ODBC se compone principalmente de cuatro elementos:

- La Aplicación. Esta realiza el procesamiento, llama a las funciones ODBC para presentar las sentencias SQL y regresar la información obtenida con la petición.
- El Driver Manager. Tiene como función cargar los drivers necesarios en la ejecución de una aplicación.
- El Driver. Procesa las llamadas a funciones ODBC, se encarga también de procesar las peticiones SQL a un data source específico, y regresa los resultados a la aplicación. El Driver puede modificar una petición de tal modo que cumpla con la sintaxis requerida por el manejador de Base de Datos que se este utilizando.
- El Data Source. Esta compuesto por los datos que el usuario quiere acceder, el sistema operativo asociado, el manejador de Bases de Datos y la plataforma de red (en caso de que exista) que se está utilizando para acceder al manejador de Bases de Datos.

Figura 4. Arquitectura del Open Database Connectivity.



La aplicación realiza las siguientes tareas:

- Requiere de la conexión al data source.
- Envía peticiones al data source.
- Establece las áreas en las que se almacenarán los resultados de las peticiones.
- Maneja el procesamiento de errores.
- Reporta los resultados al usuario.
- Finaliza la conexión con el data source.

Las funciones del Driver Manager son:

- Su función principal es la carga de drivers.
- Utiliza el archivo ODBC.INI para mapear el nombre del data source al driver correspondiente.
- Realiza el procesamiento de las llamadas de inicialización ODBC.
- Provee la validación de parámetros y de secuencias para las llamadas ODBC.

El Driver se encarga de:

- Establecer la conexión al data source.
- Presentar las peticiones al data source.
- Traducir los datos a los formatos requeridos por la aplicación.
- Formatea los errores.
- Inicia transacciones en caso de que el data source requiera del inicio explícito de ellas.
- Regresa los datos a la aplicación.

El Data Source tiene como función:

- Establecer la conexión del manejador de Bases de Datos sobre el sistema operativo que se esté utilizando y que es accesado por una red determinada.

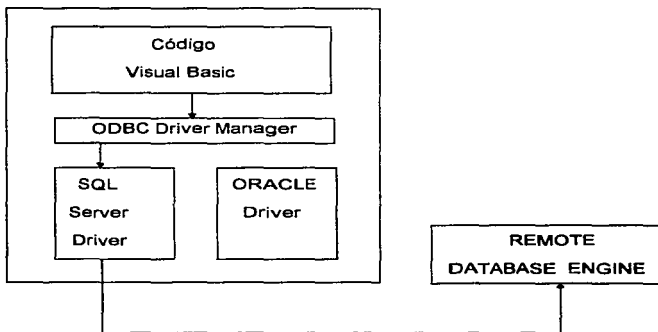


Figura 5. Diagrama de acceso vía ODBC.

ODBC Direct.

El método ODBC Direct es la manera de acceso que se incluyó en la versión más reciente de Visual Basic. Con este método no se carga en memoria el Jet Engine y utiliza los mismos objetos, métodos y propiedades que utiliza DAO para acceder a la interface con RDO.

Cuando el ODBC Direct es utilizado, el Jet Engine no se invoca y DAO es el encargado de rutear todos los requerimientos de datos a través de la interface RDO, el cual utiliza el ODBC Driver Manager para comunicarse al servidor remoto por medio del driver especificado.

El Acceso a través de ODBC Direct se ilustra a continuación:

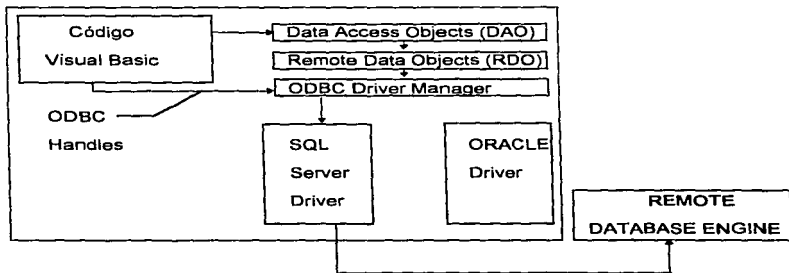


Figura 6. Diagrama de acceso a través de ODBC Direct.

VBSQL.

La librería de Visual Basic para SQL Server es una implementación API, de una DB-Library y es aportada como una interface nativa a Microsoft SQL Server. Si se está creando una nueva aplicación de Cliente/Servidor se necesita flexibilidad y funcionamiento de un modelo API, se debe usar RDO en lugar de VBSQL.

Se utiliza cuando la aplicación necesita acceder solo sistemas de Microsoft SQL Server, además de algunas versiones de Sybase SQL Server.

Las características principales de VBSQL son:

- Operación totalmente asíncrona.
- Acceso únicamente a SQL Server.
- Es una interface nativa para Microsoft SQL Server.
- Especifica las funciones API para direccionar cada acceso a SQL Server.
- Interface Bulk Copy
- Contienen manejador de errores.

CAPITULO IV CASO PRACTICO

4.1 FUNCIONES Y CARACTERÍSTICAS DE UNA EMPRESA DE PUBLICIDAD.

Podemos definir la publicidad como el conjunto de medios a través de los cuales se da a conocer algún tipo de producto a la gente.

Un medio es el que servirá como canal emisor para dar a conocer el producto que se quiere difundir.

Un producto es aquel objeto que satisfecerá alguna necesidad, ya sea primaria o secundaria.

La empresa de publicidad a la que se hace referencia en este caso, es mejor conocida como agencia de medios, se encarga básicamente de la contratación de los medios de comunicación para difundir los productos de diferentes empresas.

Para el funcionamiento de la agencia de medios se necesita antes que nada que establezca lo que se conoce como cuentas, esto es que sea contratada por alguna otra empresa para difundir sus productos en alguno de los medios de comunicación, los cuales pueden ser televisión, radio, prensa, revistas, exteriores (se refieren a los carteles que se colocan en los edificios, por ejemplo) y especiales, estos últimos se refieren a publicidad en taxis u hoteles, por ejemplo.

En la agencia de medios a los anuncios comerciales se les conoce como spots para el caso de que se transmita por televisión o radio y se conocen como inserciones cuando se tratan de anuncios en revistas, periódicos, medios exteriores y medios especiales.

La empresa de publicidad tiene las siguientes áreas:

MEDIOS.

Existe un área de este tipo para cada cuenta, aunque si es una cuenta menor puede ser incorporada al equipo de trabajo de alguna cuenta mayor, se encarga principalmente de elaborar los planes o campañas que por determinado periodo promoverán un artículo o producto para los diferentes medios de comunicación. Les compete analizar cual es el mejor "lugar" para promocionar el producto tomando en cuenta a que tipo de público va dirigido, por ejemplo, si se elige como medio de comunicación la televisión se basan en estadísticas que les indican que tipo de público, es decir, jóvenes, amas de casa, niños, señores, etc., son aquellos que ven la televisión a determinado horario, en base a ello planean los horarios para el anuncio de los productos.

De igual modo sucede con las revistas, de acuerdo al tipo de producto se analiza en que revista se considera que tendrá más éxito la publicidad de dicho producto tomando en cuenta qué personas son quienes compran la revista de que se trate.

Todo este análisis lleva a la construcción de un plan global que es el que involucra una campaña determinada para un producto dado, este plan global tiene un periodo de duración el cual puede ser desde seis meses hasta un año, además incluye la suma que el cliente de la agencia está dispuesto a invertir en la publicidad para su producto. Dentro de dicho plan se generarán subplanes, los cuales son las contrataciones para la transmisión de los spots o la publicación de las inserciones.

Una vez que se ha elaborado el subplan que contiene el presupuesto de la campaña de publicidad del producto se transmite al cliente para que de el visto bueno y una vez que ha sido autorizado pasa a la siguiente área.

COMPRAS

El área de compras es la que recibe los presupuestos autorizados por los clientes y se encarga de realizar la contratación de los medios.

Esta área está dividida por cada uno de los medios, es decir, se encuentra el área de compras para televisión, para radio, para revistas, etc.

Su principal función es la contratación del medio para la transmisión de la publicidad del producto y además vigilar que se cumplan los requisitos bajo los que fue contratado el servicio. Esto es, cuando se realiza una contratación para televisión en la que se transmitirá un spot de veinte segundos por ejemplo, hay una persona encargada de monitorear que la transmisión sea a la hora correcta y que además el tiempo de duración del spot se cumpla totalmente ya que solo en ese caso puede ser cubierto el monto de la transmisión.

Para llevar a cabo su función el departamento de compras requiere de las llamadas ordenes, las cuales son el formato a través del cual se solicita la transmisión de un spot o bien la publicación de una inserción.

Una vez que se confirma que los spots o inserciones fueron transmitidos o publicados correctamente se les da un status de confirmado y con este pueden pasar a la siguiente área.

FINANZAS

Es el área encargada del manejo monetario, recibe las ordenes confirmadas por el departamento de compras y extiende las facturas correspondientes para el pago a los medios contratados.

En este departamento se lleva a cabo todo el proceso contable de la empresa a través del registro de los movimientos que la empresa realiza, esto incluye el crédito que otorgan y el área de cobranza en el que se recibe el pago por parte de los clientes correspondiente al monto de la comisión que la agencia cobra por la contratación que hace.

4.2 REQUERIMIENTOS DEL SISTEMA.

Para la realización del sistema se requiere antes que nada de la definición del software y hardware que se utilizará para la implantación del sistema.

En este caso el lenguaje de programación a utilizar es Visual Basic a través de éste se creará la interface con el usuario. Como manejador de bases de datos se utilizará SQL Server.

El tipo de acceso a la base de datos será a través de ODBC, pues como ya se describió anteriormente cuenta con diversa ventajas para establecer la conexión entre la aplicación y el servidor de datos.

La red con la que trabaja la empresa utiliza el sistema operativo Novell y cuenta con cableado UTP bajo una arquitectura de bus.

Las estaciones de trabajo son máquinas con procesador 486 y 16 Mb en RAM, aún cuando la mayoría de los archivos se guardan en los directorios del servidor, cuentan con un disco duro de 500 Mb.

De acuerdo a las funciones que la empresa realiza, se desea contar con un sistema que cubra la parte de la contratación de medios los cuales serán utilizados principalmente por el departamento de medios y el departamento de compras para que realicen con mayor rapidez y precisión las funciones propias de cada uno y que ya fueron descritas anteriormente.

El siguiente esquema muestra el proceso que la agencia sigue en cuanto a la contratación de medios el cual debe ser cubierto por el Sistema de Contratación de Medios para la optimización del mismo.

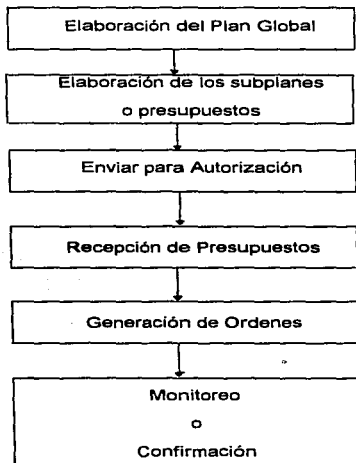


Figura 7. Diagrama del proceso de Contratación.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

En base al diagrama anterior podemos identificar que de acuerdo a la descripción de funciones por departamento presentadas previamente, los requerimientos para el departamento de medios serán cubiertos hasta la recepción de los presupuestos y a partir de ahí se deben cumplir los utilizados por el área de compras.

En general podemos en listar los requerimientos de la siguiente forma:

- Un sistema que permita al usuario dar de alta planes globales.
- Que permita crear a partir del plan global, el subplan o presupuesto para las inserciones o spots requeridos.
- La opción de marcar un presupuesto que ha sido enviado a autorización y de igual modo marcarle cuando sea recibido.
- Que permita la generación de las ordenes para los diferentes medios.
- Una opción que permita colocar un status de confirmado cuando las inserciones o spots hayan sido transmitidos correctamente.

En base a los principales requerimientos del sistema, se presenta el siguiente diseño.

4.3 DISEÑO DEL SISTEMA.

En primer lugar se debe considerar el diseño de la base de datos que será utilizada para guardar la información correspondiente a los presupuestos, ordenes, inserciones y spots que serán creados a través del sistema.

Las siguientes tablas son las que se proponen para componer la base de datos correspondiente.

Presupuestos

Plan_Id	integer
Presupuesto_Id	integer
Promotor_Id	integer
Cliente_Id	integer
Medio_Id	integer
Presupuesto_Creado	date
Presupuesto_Total	long
Presupuesto_Status	smallint
Presupuesto_Folio	long

Ordenes

Plan_Id	integer
Orden_Id	integer
Cliente_Id	integer
Medio_Id	integer
Presupuesto_Id	integer
Orden_Creado	date
Orden_Total	integer
Orden_Status	smallint
Orden_Folio	integer

Cientes

Cliente_Id	integer
Cliente_Nom	varchar(255)
Cliente_Dom	varchar(255)
Cliente_Tel	varchar(9)
Cliente_RFC	varchar(14)

Promotores

Promotor_Id	integer
Promotor_Nom	varchar(255)
Promotor_Dom	varchar(255)
Promotor_Tel	varchar(9)
Promotor_RFC	varchar(14)

Spots

Spot_Id	integer
Presupuesto_Id	integer
Orden_Id	integer
Cliente_Id	integer
Medio_Id	integer
Promotor_Id	integer
Spot_Duración	integer
Spot_Horario	time
Spot_Creado	date
Spot_Status	smallint
Spot_Tarifa	long

Inserciones

Inserción_Id	integer
Presupuesto_Id	integer
Orden_Id	integer
Cliente_Id	integer
Medio_Id	integer
Promotor_Id	integer
Inserción_Méridas	integer
Inserción_Edición	date
Inserción_Creado	date
Inserción_Status	smallint
Inserción_Tarifa	long

Medios

Medio_Id	Integer
Medio_Nom	Varchar(10)

Planes

Plan_Id	Integer
Cliente_Id	Integer
Plan_Producto	Varchar(255)
Plan_Campaña	Varchar(255)
Plan_Monto	Long
Plan_FechaIni	Date
Plan_FechaFin	Date

Las tablas propuestas estarán ligadas por el número de presupuesto, el número de orden, el número de cliente, el número de promotor y el número de medio principalmente, aunque para los presupuestos es necesario que se haga una liga con la tabla de planes.

Para la interface se debe contar con una pantalla principal en la que exista un menú en el que figuren las siguientes opciones:

- Planes.
- Subplanes.
- Presupuestos.
 - Enviado.
 - Recibido.
- Ordenes
- Monitoreo.

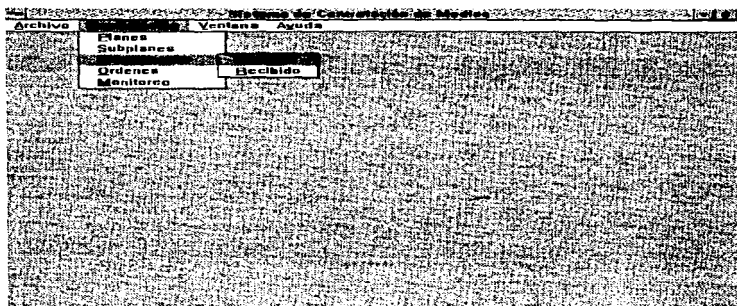


Figura 8. Pantalla Principal del SCM

Al elegir la opción de Planes el usuario accederá a una pantalla en la que capturará el nombre del cliente, así como el nombre del producto y el nombre de la campaña que dicho cliente desee promocionar, además de ello deberá capturarse tanto el periodo de duración de la campaña como el monto que el cliente está dispuesto a invertir para la publicidad.

The image shows a data entry screen titled "Planes". On the left side, there are four labels: "Cliente", "Producto", "Campaña", and "Periodo". To the right of each label is a rectangular input field. The "Producto" field is wider than the others. At the top right of the window, there are standard window control buttons (minimize, maximize, close).

Figura 9. Pantalla para la captura de los datos de planes.

En la segunda opción, se elegirá un cliente, producto y campaña que ha sido previamente asignado a un plan, dentro de la pantalla de trabajo del subplan, se debe elegir el medio para el que se va a presupuestar y dar de alta los spots o inserciones. Para el caso de los spots, se debe introducir la fecha en la que será transmitido y el horario, para las inserciones se captura la fecha en la que será publicado y además el costo en ambos casos.

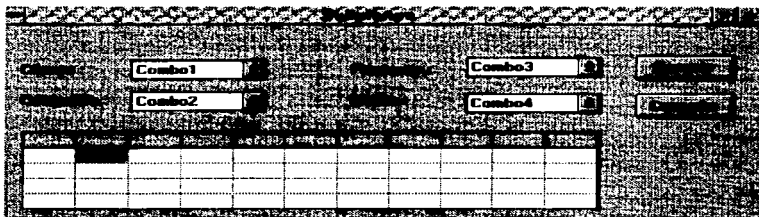


Figura 10. Pantalla para la información acerca de subplanes.

En la opción de presupuestos, se presentará un submenú para la elección de enviar o de recibir el presupuesto, al entrar a alguna de estas pantallas se colocará una marca que indicará si el presupuesto ha sido enviado para la autorización del cliente. El elegir la opción de recibir, coloca la marca que indica que el presupuesto a sido autorizado y por lo tanto puede comenzarse con la generación de las ordenes.

Dentro de la opción de ordenes, se presentará una pantalla en la que el usuario invocará a través del folio del presupuesto las inserciones o spots que le componen. Una vez que las inserciones o spots son desplegados, el usuario podrá mandar a generar la orden, la cual contendrá los datos del cliente así como los del promotor a quien se girará la orden. Debe contener también el folio del presupuesto y el folio correspondiente a la orden. En el detalle de los spots o inserciones se imprimirá la fecha de la transmisión o publicación, y si es el caso de un spot se incluirá el horario y la duración del mismo.

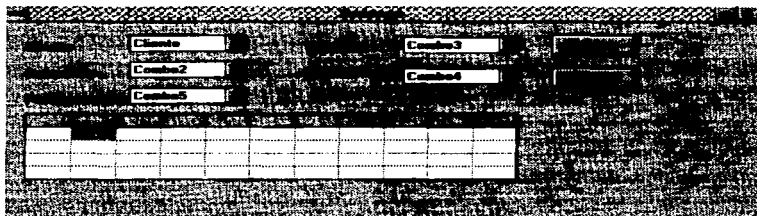


Figura 10. Pantalla para la generación de ordenes.

Quando se trabaja dentro de la opción de monitoreo se trabaja con una pantalla en la que se deben desplegar todas las inserciones o spots que se giraron en la orden. Dicho despliegue se llevará a cabo cuando el usuario teclee el folio de la orden, a través de dicho folio se obtendrán las inserciones correspondientes de las cuales se marcaran como confirmadas las que hayan cumplido con los requisitos establecidos en la requisición.

El manejo del Sistema de Contratación de Medios, permitirá que cada una de las áreas a quienes va dirigido, realicen las actividades que les competen de la forma más rápida y clara posible, para que en el momento en que la información sea requerida, dichos departamentos estén en condiciones de darla, dado que la información oportuna y clara es un factor de gran importancia para la toma de decisiones.

Vale la pena aclarar que el Sistema de Contratación de Medios es un proyecto que puede crecer en gran porcentaje, pensando por ejemplo , que en un futuro se implemente vía Intranet, con lo cual seguiría conservando el concepto bajo el que ahora se está diseñando que es la arquitectura Cliente/Servidor obteniendo además las ventajas que ofrece la implementación de la Intranet.

CONCLUSIONES

En conclusión para el desarrollo de una aplicación bajo la arquitectura Cliente/Servidor se debe seguir todo un proceso. Como primer punto se debe entender lo que es el modelo Cliente/Servidor y en este momento analizar su tendencia, ya que como se mencionó en el primer capítulo la implementación de las Intranet's es un fenómeno que se está dando con mucha fuerza y también el impacto de Java.

Por lo tanto, al iniciar con la implementación se debe definir que lenguaje de programación se va a utilizar así como el manejador de Bases de Datos y analizar que métodos de acceso hay entre ellos y en base a ello trabajar con el que mejor nos convenga tomando en cuenta los requerimientos del sistema.

Por ultimo se puede decir, que la informática es una parte muy importante dentro de las empresas sea cual sea el giro de ésta, ya que a través de la automatización de los procesos, la información se obtiene de manera oportuna, un hecho que es fundamental para una buena toma de decisiones.

BIBLIOGRAFÍA

1. Microsoft Visual Basic 3.0 Programming System for Windows. Editorial Microsoft Corporation, México 1993, 674pp.
2. Microsoft SQL Server 6.0. Editorial Microsoft Corporation, México 1995, 728 pp.
3. Seminario Desarrollo Cliente/Servidor con Visual Basic y SQL Server. Editorial Ddemésis, México, 1994, 111 pp.
4. Microsoft SQL Server 6.0 Unleashed. Editorial Sams Publishing, USA 1996, 761 pp.
5. Using Sybase System XI. Editorial QUE, USA 1996, 761 pp.
6. Visual Basic 4. Editorial QUE, USA 1996, 637 pp.
7. JAVA Unleashed. Editorial Sams Net, USA 1996, 971 pp.
8. Using ORACLE. Ted Crooks. Editorial QUE, USA 1991, 640 pp.
9. Listen to Those Grinding Gears, Revista Network Computing, 15 de julio de 1996, Num 11, pp. 143
10. Thin Clients, Thin Networks...Maybe, Revista MidRange Systems, 25 de octubre de 1996, Núm. 16, pp. 20