

A stylized handwritten signature in black ink, featuring a large, sweeping initial 'V' and a smaller 'C' that loops back. The signature is positioned above the date and name.

MAYO 1997

VICTOR CONTRERAS

DDICNE  
\$150.00

**UNIVERSIDAD DE MONTERREY**  
**DIVISIÓN DE INGENIERÍA Y CIENCIAS NATURALES Y EXACTAS**



0902653

**SISTEMA DE INFORMACIÓN EJECUTIVA DE VENTAS Y  
OPERACIONES DE ABA | TURISMO**

PROGRAMA DE EVALUACIÓN FINAL QUE PRESENTA:  
**ADRIANA ELBA OLMEDA D'ALEJANDRO**

040.0016  
0515  
1997

EN OPCIÓN AL TÍTULO DE:  
**INGENIERO EN COMPUTACIÓN ADMINISTRATIVA Y DE  
PRODUCCIÓN**

SAN PEDRO GARZA GARCÍA, N. L.

MAYO DE 1997

**BIBLIOTECA**  
UNIVERSIDAD DE MONTERREY

## DEDICATORIAS

A DIOS por darme la oportunidad de vivir.

A mis padres Ing. Jose Antonio Olmeda y Lic. Adriana D' Alejandro de Olmeda  
por darme la vida, su cariño y su apoyo.

A mis hermanas Elisa, Mónica y Ana Gabriela.

A mi abuelita.

A mi asesor Ing. Víctor Contreras por ser más que un asesor un amigo por su  
apoyo, paciencia y enseñanzas durante la realización de este proyecto.

A mis amigos por su enorme apoyo y cariño especialmente a mis "Comadres" y  
a los "Chafos".

## AGRADECIMIENTOS

A Mónica Carranza, Paola González, Catia Martínez y Joni Garza por los cafés de los miércoles.

A Héctor y Madeleine, Luis Antonio y Susi, Cris y Roberto, y Bernardo por que siempre estuvieron conmigo.

A mis compañeros pefistas Joni, Dario, Luis, Sandra, Héctor y Daniel por hacer de las desveladas más amenas y menos tensionantes.

Al Lic. Rolando Torres y al Lic. Mario Ortiz por todas sus atenciones y la oportunidad de realizar mi proyecto en ABA|SIS.

A todo el personal de ABA|SIS y UNIMICRO.

---

## PRÓLOGO

Cuenta una leyenda japonesa que en un inicio las sociedades eran regidas por músculos, después por joyas y ahora por espejos. En un inicio el más fuerte era el que poseía el poder, las sociedades evolucionaron y dieron paso a una comunidad regida por el más rico. A partir de ella, la evolución continúa llegando hasta nuestros días a la sociedad de la Información, sociedad que tiene conciencia de sí mismo y de los hechos que la rodean.

Antes ganaba el más fuerte o el más rico, ahora gana el que tenga más y mejor información.

En esta sociedad, los mercados son aguerridos, competitivos, cambiantes y globales. Los ejecutivos de hoy, tienen que encontrar los medios que les ayuden a definir y apoyar sus ventajas competitivas o "core competences", medios que les ayuden a incrementar barreras a nuevos competidores, a incrementar su poder ante la rivalidad de su competencia, a disminuir el poder de los proveedores y a aumentar la necesidad de los clientes hacia sus productos. Estos medios, requieren información o son la información misma.

---

A nivel operativo en cualquier organización, la definición de la información necesaria para realizar el trabajo nunca ha sido problema, pero, a nivel estratégico, los ejecutivos no solo requieren información de su empresa, sino que además necesitan conocer a su mercado, clientes, competencia, proveedores, etcétera.

La gente de informática como tal, debe de habilitar e impulsar el trabajo de los ejecutivos con la utilización de información apoyada con tecnología.

Es en proyectos como éste, donde la aplicación de tecnologías como el Data Warehouse ayudan a que los tomadores de decisiones posean la información que requieren, con el formato adecuado y de manera oportuna.

El proyecto desarrollado por Adriana Olmeda, consiste en el planteamiento y construcción de un Data Warehouse en un Intranet, el cual mediante un navegador va a ser consultado.

Mi amiga Olmeda realizó una extensa investigación acerca del Data Warehouse, de Intranet y de aplicaciones Cliente/Servidor. El capítulo de marco teórico estoy seguro que será de suma ayuda a todos los que deseen emprender dichas construcciones.

Cabe resaltar, que sin contar con herramientas para el desarrollo de este tipo de tecnologías, el proyecto empleó varios de los elementos contenidos en los metadatos, pieza clave de los Data Warehouses.

El Sistema de Información Ejecutivo fue desarrollado en HTML, utilizando como apoyo Microsoft front-page, utiliza controles ActiveX y para hacer accesos a la base de datos utiliza la aplicación Cold Fusion.

Las metodologías utilizadas para el desarrollo del proyecto poseen una orientación a objetos, con alineaciones de actividades para el diseño y construcción del Data Warehouse.

Víctor Iván Contreras Lozano

# ÍNDICE GENERAL

<b>Introducción</b>	1
<b>I. Información general de la empresa</b>	4
1.1. La Empresa	4
1.1.1 Breve historia de la empresa	4
1.1.2 Principales servicios que ofrece	5
1.1.3 Organigrama de la empresa	5
1.1.4 Infraestructura de Informática	5
<b>II. Situación actual</b>	6
2.1 Antecedentes y Descripción del problema	7
2.1.1 Antecedentes	8
2.1.2. Problemas actuales encontrados	9
<b>III. Descripción del sistema propuesto</b>	13
3.1 Solución propuesta	13
3.2 Arquitectura del sistema	13
3.3. Relación con otros sistemas	
3.4 Beneficios de la solución	14
<b>IV. Objetivos del proyecto</b>	15
4.1 Objetivo general del proyecto	15
4.2 Objetivos específicos del proyecto	15
4.3 Actividades particulares para el cumplimiento de los objetivos específicos del proyecto	16
<b>V. Fundamentos del proyecto</b>	18
5.1 Disciplinas/Methodologías utilizadas	18
5.1.1 SDD	19
5.1.2 Análisis y diseño orientado a objetos	22
5.1.2.1 Introducción básica	22
5.1.2.2 Análisis orientado a objetos	22
5.1.2.3 Diseño orientado a objetos	25
5.2 Data Warehouse	26
5.2.1 Breve historia	26
5.2.2 Definición	27
5.2.3 Arquitectura de referencia	28

5.2.4	Enfoque de selección de una arquitectura	33
5.2.4.1	Construcción de un Data Warehouse en arquitectura Cliente/Servidor	34
5.2.5	Los requerimientos de un Data Warehouse	34
5.2.6	La importancia de los metadatos	35
5.2.7	Ciclo del desarrollo del Data Warehouse	39
5.2.7.1	Análisis del Data Warehouse	39
5.2.7.2	Diseño del Data Warehouse	39
5.2.8	La complejidad de construir y usar un Data Warehouse	43
5.2.9	La minería de datos	44
5.2.10	Criterios de evaluación de un Data Warehouse	45
5.2.11	Data Mart vs. Data Warehouse	48
5.2.12	Data Warehouse y su ventaja competitiva en los 90's	50
5.2.13	El futuro	51
5.3	Intranet	52
5.3.1	Internet – Breve Historia	52
5.3.2	Intranet – Breve Historia	53
5.3.3	¿ Que es "Intranet" ?	53
5.3.4	Ventajas de un "Intranet"	55
5.3.5	Parámetros de medición de efectividad y los beneficios de un "Intranet"	57
5.3.6	"Intranet" vs. "Internet"	58
5.3.6	Las tendencias	58
<b>VI.</b>	<b>Resultados del análisis</b>	<b>59</b>
6.1	Análisis orientado a objetos	59
6.1.1	Identificación de objetos	59
6.2	Análisis de la estructura de objetos	60
6.3	Análisis del comportamiento del objeto	61
<b>VII.</b>	<b>Resultados del diseño</b>	<b>62</b>
7.1	Diseño orientado a objetos	62
7.1.1	Diseño de la estructura y del comportamiento de un objeto	62
7.1.2	Diseño de la base de datos (Data Warehouse)	70
<b>VIII.</b>	<b>Resultados del SDD</b>	<b>71</b>
8.1	Documento visión	71
8.1.1	Antecedentes	71
8.1.2	Enunciado de visión	71
8.1.3	Perfil del usuario	72
8.1.4	Necesidades actuales	72
8.1.5	Participantes	73
8.1.6	Arquitectura Conceptual	74
8.1.7	Infraestructura del proyecto	75
8.1.8	Identificación de Riesgos	75
8.2	Documento especificaciones funcionales	75

8.2.1 Antecedentes	75
8.2.2 Objetivo	76
8.2.3 Estructura del proyecto	76
8.2.4 Agenda	78
8.2.5 Descripción funcional	78
8.2.6 Factores críticos	79
8.2.7 Diseño de la interfaz	79
8.2.8 Pantallas y reportes	80
8.3 Diseño Conceptual	81
8.3.1 Antecedentes	81
<b>IX. Alcances y limitaciones del sistema</b>	<b>89</b>
9.1 Alcances	89
9.2 Limitaciones	90
<b>X. El futuro del sistema</b>	<b>91</b>
<b>XI. Conclusiones</b>	<b>93</b>
<b>XII. Bibliografía</b>	<b>96</b>
<b>Apéndices:</b>	
Apéndice A. Diseño y estructura de la base de datos	
Apéndice B. Pantallas y reportes	
Apéndice C. Diseño lógico	
Apéndice D. Diseño físico	
Apéndice E. Código	

## ÍNDICE DE FIGURAS

1	Organigrama de ABA TURISMO	5
2	Arquitectura de Software propuesta	12
3	Relación con otros sistemas	13
4	Modelo de proceso del SDD	21
5	Panorama de la arquitectura de referencia del Data Warehouse	29
6	Ejemplo de un modelo dimensional	41
7	Ejemplo del esquema "Snowflake"	43
8	Resultados del criterio de escalabilidad	46
9	Modelo del Data Warehouse	51
10	Diagrama de clase de agente de ventas	64
11	Diagrama de clase de clientes	65
12	Diagrama de clase de aerolínea	66
13	Diagrama de clase de proveedor de servicios	67
14	Diagrama de clase de hotel	68
15	Diagrama de clase de concentración de ventas	69
16	Arquitectura de software propuesta	74

## ÍNDICE DE TABLAS

1	Tabla de roles y responsabilidades	20
2	Tabla de roles y enfoques	20
3	Tabla de roles y habilidades únicas del proyecto	20
4	Ejemplo de un elemento de metadato	37
5	Comparación entre las bases de datos operacional y para apoyo a la toma de decisiones	40
6	Tabla de similitudes y diferencias entre OLTP y un Data Warehouse	50
7	Los problemas actuales y los beneficios al negocio que pueden encontrarse debido a un "Intranet"	55
8	Lista de Participantes	73
9	Lista de Roles	77
10	Agenda	78
11	Tabla de documentos entregables	81

---

## ABSTRACT

La necesidad urgente por información para la toma de decisiones, requirió comenzar un proyecto de Data Warehouse en ABA|TURISMO, instalarlo en un Intranet y desarrollar un Sistema de Información Ejecutivo.

El Data Warehouse requirió de investigar absorbentemente esta tecnología, la cual se encuentra documentada en el capítulo V. La base de datos concentradora de la información fue construida en SQL Server 6.5.

La Intranet requirió de adquirir conocimientos de Internet, de instalaciones, por ejemplo, Internet Information Server, Windows NT, etcétera, en el servidor de Web y en los clientes el "browser" Microsoft Explorer 3.0.

El Sistema de Información Ejecutivo requirió aprender HTML, controles ActiveX y ColdFusion para realizar las llamadas a la base de datos.

La metodología utilizada fue orientada a objetos, aunada con actividades del diseño y construcción del Data Warehouse.

## INTRODUCCIÓN

El turismo es una industria demasiado competida, incluso, desde sus inicios, las Agencias de Viajes siempre han tenido competidores mundiales.

Con las crisis económicas mundiales, los costos de los servicios vienen a ser un factor competitivo de las agencias y aunado a ellos, el servicio dictará las decisiones de los Clientes.

La diferenciación entre agencias viene a redundar en que los ejecutivos posean la información que requieren para establecer costos competitivos, crear paquetes de viajes, elegir mejores proveedores, buscar mercados para turismo receptivo, establecer contratos más baratos con agencias de autos, pronosticar ventas para ciertos eventos, etcétera.

Para ABA|TURISMO, la tecnología siempre ha sido una necesidad; para la reservación de vuelos, separación de habitaciones en hoteles, registro de renta de automóviles, etcétera. La mayor parte de las aplicaciones computacionales que poseen van tendientes a facilitar la labor de los agentes de viajes.

En este proyecto, la tecnología se utilizará para facilitarle a los ejecutivos el recurso más valioso que poseen las organizaciones, la información. Dicha información la requieren para la toma de decisiones y el establecimiento de estrategias para la venta.

Los sistemas tradicionales que apoyan a los ejecutivos, extraen los datos de las bases de datos operacionales, los consolidan y los presentan en una interfaz diseñada con tal propósito. Las aplicaciones que extraen estos datos, los consolidan y los presentan, poseen en su mayoría tiempos de respuesta altos, los cuales no cubren los requerimientos que poseen los ejecutivos.

Es por ello, que surgen tecnologías como el Data Warehouse, donde los datos son concentrados y relacionados a las bases de datos operacionales mediante metadatos. Estos últimos, son una de las partes más importantes en la construcción y diseño de este tipo de tecnologías. En el capítulo V, podrán encontrar bibliografía al respecto. Este trabajo es el establecimiento de la plataforma tecnológica y filosófica de esta tecnología, la cual servirá de base para siguientes concentraciones de datos de la empresa.

La construcción requirió de investigar a profundidad esta tecnología, y utilizar una metodología de apoyo. El sistema fue desarrollado en base a la metodología orientada a objetos. El resultado final del Data Warehouse es la concepción tecnológica del mismo, el diseño de base de datos físico que lo soporta, el cual incluye ciertos elementos contenidos en los metadatos y las aplicaciones que permitan la consolidación de la información.

Uno de los estándares para el desarrollo de sistemas en ABACO Grupo Financiero, es el trabajar mediante la Disciplina de Desarrollo de Soluciones de Microsoft, la cual sirvió como marco de trabajo para las metodologías antes mencionadas. Los resultados de la utilización de ésta, pueden verse en el capítulo VIII.

La aplicación desarrollada para obtener los datos de la base de datos del Sistema de Ventas al Menudeo, condensarlos y almacenarlos en el Data Warehouse fue desarrollada en arquitectura cliente/servidor; el front-end fue desarrollado en Visual Basic 4.0 y el back-end utiliza la bases de datos del Data Warehouse.

Los datos almacenados en el Data Warehouse serán consultados mediante un Sistema de Información Ejecutivo, el cual fue desarrollado en HTML 3.0 apoyado funcionalmente con VBScripts del Microsoft Explorer 3.0 y de controles ActiveX para las tabulaciones y gráficas de información, el acceso a la base de datos se realiza mediante Cold Fusion 1.5, siguiendo los estándares de diseño de ABA|SIS.

Para el desarrollo de las páginas del Sistema de Información Ejecutivo se utilizó el front-page 97 de Microsoft.

Debido a que las tecnologías de Internet poseen una interfaz estándar, el sistema será fácil de utilizar y preparará a los usuarios para el ambiente futuro.

El Sistema de Información Ejecutivo trabaja en un Intranet instalado con tal propósito, para dicha instalación, se requirió en el servidor, Intranet Information Server y Windows NT.

## I. INFORMACIÓN GENERAL DE LA EMPRESA

### 1.1 La Empresa.

#### 1.1.1 Breve historia de la empresa.

La empresa ABA|TURISMO nace por el año de 1990 como una empresa filial de ABACO Casa de Bolsa, desempeñando un papel promocional de ABACO, ofreciendo los servicios de agencia de viajes a los clientes de la Casa de Bolsa. La empresa de ABA|TURISMO fue creada para dar una imagen a la empresa, con el fin de ofrecer productos de calidad a los clientes, esa era la idea original, pero debido a que ABA|TURISMO era una empresa nueva y no tenía la suficiente experiencia no se logró alcanzar dicho fin.

Posteriormente, ABA|TURISMO sufre una crisis de identidad cuando se realiza la transformación de ABACO Casa de Bolsa a lo que es ahora, ABACO Grupo Financiero. Esto produce un rezago en el desarrollo de ABA|TURISMO debido a que la nueva prioridad del Grupo Financiero era consolidarse como tal, creando ABA|SEGUROS, ABA|RENDA, ABA|FACTOR y ABA|DIVISAS.

En 1992 nace la sucursal de ABA|TURISMO en la Ciudad de México, marcando con esto un nuevo repunte en la empresa, este período es conocido como "El Renacimiento" de ABA|TURISMO.

A partir de 1992 la empresa empieza a darse a conocer a nivel Agencia de Viajes atendiendo al público en general, ofreciendo productos de muy buena calidad y creándose una imagen propia, dando con esto el comienzo de un desarrollo progresista hasta este momento. Actualmente laboran 15 personas en la sucursal de Monterrey y 9 personas en la Ciudad de México.

La compañía ABA|TURISMO, a diferencia de otras compañías no necesita tener muchas sucursales para tener éxito, dado que la mayor parte de las operaciones que se realizan son por teléfono.

ABA|TURISMO se encuentra entre las 10 mejores empresas de su ramo en todo el País, esto lo ha logrado en un lapso de seis años lo que refleja que es una empresa madura y seria. [BASTIERE, 96]

### 1.1.2 Principales servicios que ofrece.

Entre los servicios que ofrece la empresa se encuentran: La reservación de vuelos, hoteles, transportación, servicios de paquetes de eventos especiales, así como paquetes turísticos que se encuentran disponibles a lo largo del año. La empresa también maneja el concepto "Turismo Receptivo" el cual significa el invitar a extranjeros a visitar nuestro país. De este modo ayuda al desarrollo turístico de México.

### 1.1.3 Organigrama de la empresa.

La compañía ABA|TURISMO cuenta con dos sucursales, una que se encuentra en la Ciudad de Monterrey y la otra que se localiza en la Ciudad de México. El organigrama de la empresa es el que se presenta a continuación:

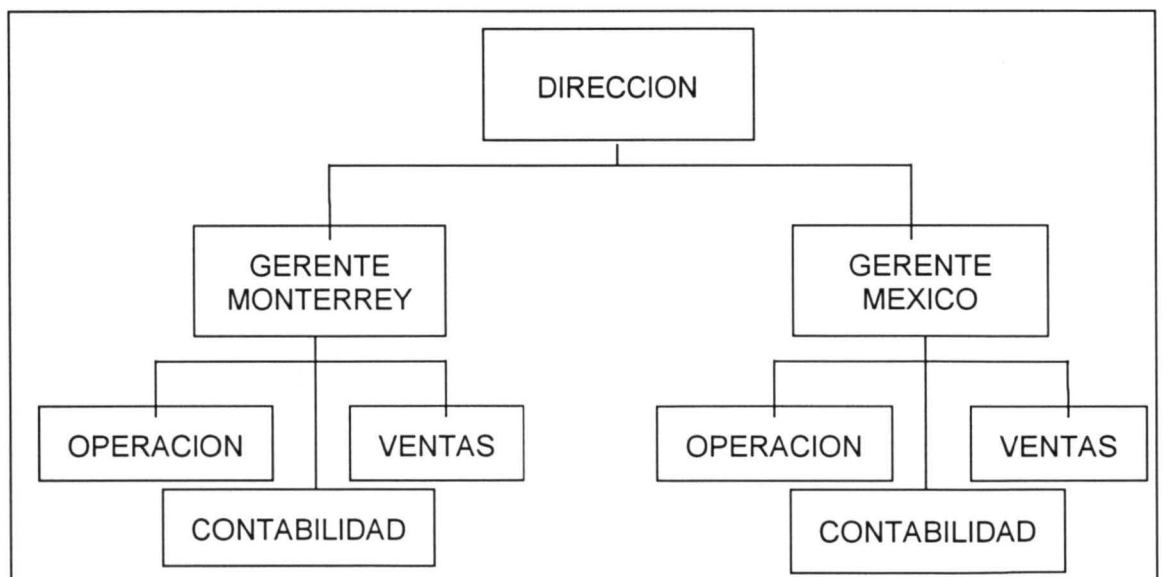


Figura 1. Organigrama de ABA|TURISMO.

El área de Operaciones tiene a su cargo a los Operadores de Viaje quienes tienen como objetivo la adquisición de grandes cantidades de plazas en vuelos, así como habitaciones en hoteles para posteriormente formar paquetes de viajes (eventos especiales o turísticos).

El área de Ventas tiene como finalidad el difundir y vender paquetes de viaje para sus clientes.

#### **1.1.4 Infraestructura de informática.**

ABA|TURISMO cuenta con un servidor Pentium/166MHZ con 32 MB en RAM, 2 GB en disco duro, unidad de DAT y de CD ROM. Cuenta también con 14 computadoras personales "PC" 486/66MHZ con 12 MB en RAM, 10 MB en disco duro y monitor SVGA. En cuanto a software se refiere tienen en sus PC el sistema operativo Windows 95, Excel y Word. En el servidor cuentan con el sistema operativo Windows NT 3.51 y SQL Server 6.5, entre otros.

## II. SITUACIÓN ACTUAL

### 2.1 Antecedentes y Descripción del Problema.

#### 2.1.1 Antecedentes.

ABA|TURISMO es una agencia de viajes la cual ofrece una diversidad de productos entre los que se encuentran: las reservaciones de vuelos, habitaciones de hotel, paquetes para eventos especiales y paquetes turísticos.

Actualmente ABA|TURISMO cuenta con un Sistema de Ventas por Menudeo, el cual apoya al departamento de Ventas. El objetivo del sistema es realizar las reservaciones requeridas por los clientes de ABA|TURISMO y la venta de paquetes de eventos especiales y turísticos. Este sistema cuenta con catálogos de clientes, hoteles, aerolíneas, y proveedores de servicios varios como renta de automóviles, boletos para conciertos, eventos deportivos y obras teatrales. Además, genera la prefacturación de las ventas realizadas por clientes que paguen directamente a ABA|TURISMO. De lo contrario, genera el comprobante (voucher) de reservación para que el cliente pague directamente al proveedor de servicios.

La prefactura y los comprobantes de reservación se mandan al departamento de Contabilidad, en donde se realiza la facturación de los clientes. Posteriormente se realiza un conteo manual de sus operaciones diarias, con el fin de generar reportes de dicha información que luego se archivarán. Esta información contenida en los reportes es utilizada posteriormente por la alta gerencia para observar el comportamiento que tiene la empresa con el fin de definir una estrategia de ventas.

Dicha información se clasifica y se define de la siguiente manera:

- Ventas por cliente: son las ventas reflejadas de cada cliente en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas por forma de pago: son las ventas reflejadas según la forma de pago (efectivo, a crédito, etc.) realizada por los clientes en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas por proveedor de servicios (renta de automóviles, boletos para conciertos, eventos deportivos y obras teatrales): son las ventas reflejadas de cada proveedor de servicios en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas por aerolíneas: son las ventas reflejadas de cada aerolínea en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas por hoteles: son las ventas reflejadas de cada hotel en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas por clase de vuelo: son las ventas reflejadas de cada clase de vuelo en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas por destino: son las ventas reflejadas de cada destino en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas totales por agencia de viajes: son las ventas reflejadas de cada agencia de viajes en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).
- Ventas por vendedor o agente de ventas: son las ventas reflejadas de cada vendedor o agente de ventas en un intervalo de tiempo (anual, trimestral, mensual y de forma diaria).

### **2.1.2 Problemas actuales encontrados.**

A continuación se presentan los problemas que se encontraron:

1. La consulta de las operaciones de ABA|TURISMO suele ser lenta. Ya que los datos provienen de dos fuentes de información (archiveros y la base de datos del Sistema de Ventas por Menudeo, los cuales se encuentran en ABA|TURISMO). Aunado a esto existe un gran volumen de información de clientes, ventas de sus servicios, proveedores, hoteles y aerolíneas.
2. Los cálculos de sus operaciones pueden llegar a ser incorrectos. Suele pasar que en el conteo manual de ventas por cliente en un intervalo de tiempo, ventas por forma de pago, las ventas por proveedor, ventas por líneas aéreas, ventas por hoteles, ventas por clase de vuelo, ventas por

destino, ventas por paquetes, ventas por agencia de viajes y ventas por vendedor se encuentre un margen de error humano al contar.

3. La representación de algunos de sus reportes de operaciones es difícil de entender. Algunas veces la información de ventas se presenta con una estructura o formato inadecuado obstaculizando la visualización y comprensión de la información. Ejemplos de formatos adecuados serían, la representación de la información a través de gráficas o tablas.
4. La información que la alta gerencia necesita para definir una estrategia de ventas en algunas ocasiones no se recibe a tiempo. Se pierde tiempo debido a que tiene que pasar la información por el departamento de contabilidad para posteriormente llegar a la alta dirección.

Con los problemas mencionados es necesario apoyar a la alta gerencia en la definición de su estrategia de ventas, mediante la obtención de información que se requiera de una manera oportuna y confiable de tal manera que no tenga que esperar mucho tiempo para recibir información financiera, tendencias de productos y noticias internas de la empresa. El concepto de información oportuna significa brindar la información de una manera rápida y al alcance del usuario. A la vez, el significado de información confiable es que dicha información sea veraz.

### III. DESCRIPCIÓN DEL SISTEMA PROPUESTO

#### 3.1 Solución Propuesta.

Proporcionar una herramienta computacional que le permita a la alta dirección acceder información oportuna y confiable de su operación diaria aplicando tecnología de "Internet". De esta manera se creará un "Intranet" de información ejecutiva. Un "Intranet" es el uso de tecnología de "Internet" para implementar aplicaciones para el uso interno de la empresa. Se decidió utilizar la tecnología de "Internet" debido a que cuenta con ese tipo de tecnología la empresa y porque el usuario así lo solicitó, ya que es fácil, rápido y entendible.

Con el fin de apoyar el "Intranet" se creará otra herramienta computacional, que tendrá como objetivo recolectar la información necesaria (clientes, proveedores, ventas, hoteles y aerolíneas) de una base de datos ya existente para después desplegar dicha información en el "Intranet". Esta herramienta se desarrollará bajo una arquitectura cliente/servidor.

Para resolver los problemas de las consultas lentas de las operaciones y los cálculos incorrectos de sus operaciones, se desarrollará un Sistema de

Condensación de Información bajo una arquitectura cliente/servidor. Este sistema realizará una serie de procesos internos de cálculos, selección y almacenamiento de datos, con el fin de recabar solamente la información necesaria para el usuario. Dicha información de clientes, proveedores de servicios, hoteles y aerolíneas se encuentra guardada en la base de datos del Sistema de Ventas por Menudeo, en donde la captura la realizan los agentes de ventas. Estos datos se almacenarán en un Data Warehouse (Base de Datos) que será diseñado y creado en este proyecto, el cual residirá en el servidor. Para después utilizar estos datos en el "Intranet". De esta forma se podrá consultar la información de una manera rápida y brindando integridad y seguridad en la información.

Para resolver el problema de la representación de sus reportes en un formato adecuado y la necesidad que la alta gerencia reciba la información requerida a tiempo, se creará utilizando la tecnología que nos ofrece "Internet", un sistema de información ejecutivo, el cual tendrá como objetivo proporcionar un acceso fácil y rápido a la información que se encuentra en el Data Warehouse, el cual apoyará a la alta gerencia a planear, monitorear y analizar el medio ambiente interno de la empresa. Entre las características se encuentran: la utilización de dispositivos de entrada alternos como el mouse, poseerá la capacidad de navegación para acceder a información financiera, tendencias de productos y noticias internas de manera rápida y fácil. Permitirá desplegar la información de ventas por cliente en un intervalo de tiempo, ventas por forma de pago, las ventas por proveedor, ventas por líneas aéreas, ventas por hoteles, ventas por clase de vuelo, ventas por destino, ventas por paquetes, ventas por agencia de viajes y ventas por vendedor en diferentes formatos (escrito o gráfico).

La solución planeada proveerá al usuario un sistema con tecnología de "Internet", el cual permitirá compartir información de manera rápida e interactuar fácilmente con él, ya que el usuario utilizará un "Browser" (Microsoft Explorer 3.0 o superior) para visualizar la información deseada de una manera amigable.

La solución propuesta será desarrollada con la metodología de análisis y diseño orientado a objetos con un enfoque cliente/servidor apoyada en la disciplina de desarrollo de soluciones. [MARTIN, 94] [SDD, 94]

### 3.2 Arquitectura del Sistema (Gráficamente).

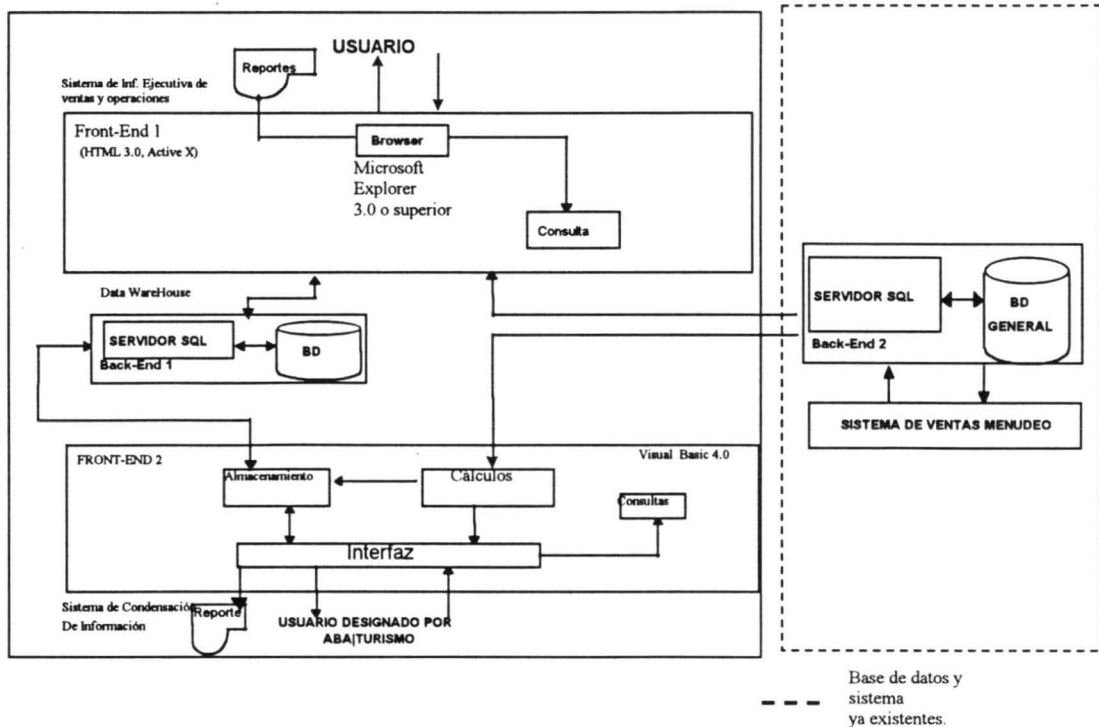


Figura 2. Arquitectura de software propuesta.

A continuación se presenta una breve explicación de la gráfica anterior.

- El "browser" o navegador desplegará de manera escrita o gráfica la información tal como: las ventas por cliente en un intervalo de tiempo, ventas por forma de pago, las ventas por proveedor, ventas por líneas aéreas, ventas por hoteles, ventas por clase de vuelo, ventas por destino, ventas por paquetes, ventas por agencia de viajes y ventas por vendedor. Para que el usuario defina posteriormente una estrategia de ventas.
- El Data Warehouse (base de datos) contendrá información procesada por el sistema de condensación de información.
- El sistema de condensación de información utilizará la base de datos del Sistema de Ventas al Menudeo (esta base de datos ya está creada y contiene información de clientes, proveedores, hoteles y aerolíneas) para posteriormente vaciar la información necesaria al Data Warehouse. Este sistema se realizará bajo una arquitectura cliente/servidor.
- El sistema de condensación de información lo usará un usuario designado por la empresa con el fin de utilizar el sistema cada vez que se requiera actualizar la información.

- La “consulta” del Sistema de Información Ejecutiva de Ventas y Operaciones se harán del Data Warehouse (el cual será diseñado y creado) y en algunas ocasiones de la base de datos general del Sistema de Ventas por Menudeo.
- La información requerida por el Sistema de Condensación de Información se obtendrá de la base de datos del Sistema de Ventas por Menudeo.

Es importante aclarar que el Sistema de Información Ejecutiva de Ventas y Operaciones y el Sistema de Condensación de Información serán desarrollados. También se diseñará y construirá el Data Warehouse (Base de Datos).

### 3.3 Relación con otros sistemas.

El proyecto básicamente se relaciona con los sistemas de ventas a menudeo de la compañía ABA|TURISMO, ya que la información contenida en la base de datos es el requerimiento para la creación del Data Warehouse. Dicha información, será utilizada posteriormente en el sistema de información ejecutiva que será desarrollado.

#### Representación gráfica de la Relación con otros sistemas.

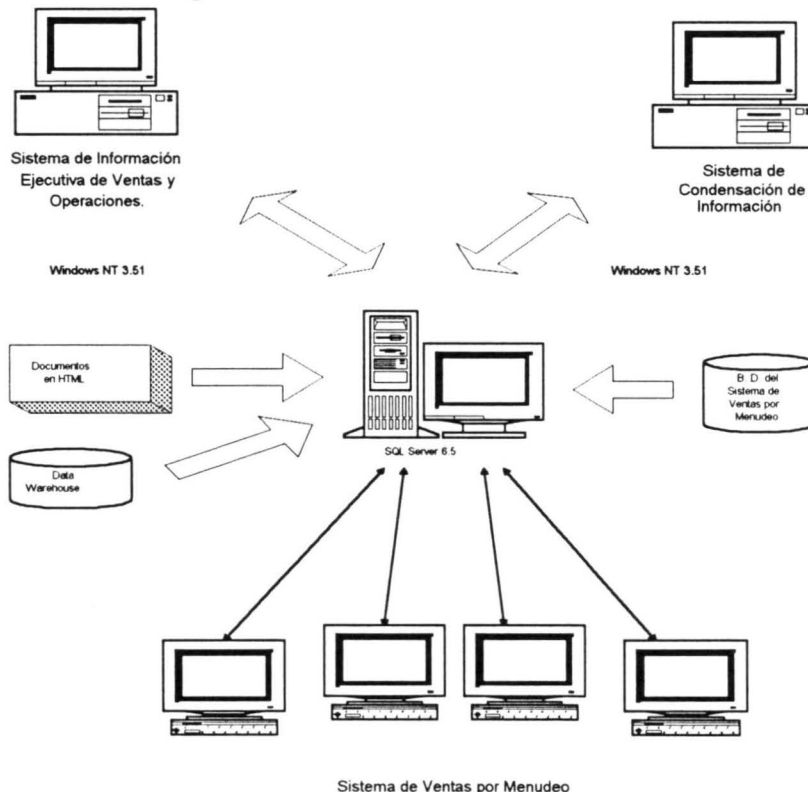


Figura 3. Relación con otros sistemas.

### **3.4 Los Beneficios de la Solución.**

- Brindarle al usuario la información necesaria, de manera oportuna y confiable con el fin de que contemple una estrategia de ventas para el beneficio de la organización.
- Disminuir el tiempo requerido de respuesta para obtener y condensar solamente la información que necesita la alta gerencia.
- Automatizar el conteo manual que realiza el departamento de contabilidad para obtener las ventas que ha tenido la empresa.
- Presentar la información con un formato adecuado para el ejecutivo para facilitarle la comprensión de dicha información.
- La interacción con tecnología de "Internet".

## IV. OBJETIVOS DEL PROYECTO

### 4.1 Objetivo General del Proyecto.

Crear dos herramientas computacionales basadas con la información que brinda el sistema de ventas por menudeo. La primera herramienta "Intranet" es para proporcionar a la alta gerencia de ABA|TURISMO información oportuna y confiable de su operación diaria que ayude a la toma de decisiones, apoyada con tecnología de Internet. La segunda herramienta es para apoyar el Intranet que tendrá como objetivo recolectar la información necesaria (clientes, proveedores, ventas, hoteles y aerolíneas) de una base de datos ya existente para el Intranet, respaldada por la arquitectura cliente/servidor.

### 4.2 Objetivos Específicos del Proyecto.

- Desarrollar un sistema de información ejecutiva ("Intranet") que apoye al usuario (alta gerencia) de la empresa ABA|TURISMO en la definición de sus estrategias de ventas en base a los requerimientos

de información financiera, tendencias de productos y noticias internas, mediante el uso de tecnología de Internet.

- Proveer al usuario de la empresa ABA|TURISMO mediante un navegador o "browser" de "Internet" para que realice sus consultas, las cuales obtendrán la información del Data Warehouse (base de datos que será diseñada y creada).
- Generar reportes y consultas (gráficas o escritas) de ventas por clientes, ventas por forma de pago, las ventas por proveedor de servicios, ventas por líneas aéreas, ventas por hoteles, ventas por clase de vuelo, ventas por destino, ventas por agencia de viajes y ventas por vendedor.
- Mantener la integridad y la seguridad del Data Warehouse de la información.
- Reducir los tiempos de búsqueda de información con el propósito de proporcionar oportunamente a la alta gerencia.
- Proveer al usuario de un sistema con interfaz amigable con el fin de que interactúe fácilmente con él, brindando diferentes formatos para la presentación de la información (escrita o gráfica).
- Permitir el uso compartido de la información.

#### **4.3 Actividades particulares para el cumplimiento de los objetivos específicos del proyecto.**

- Realizar entrevistas con los usuarios de ABA|TURISMO.
- Investigar y revisar bibliografía relacionada con el sistema. [DEVELOPMENT, 92] [INMON, 94] [GILL, 94]
- Aprender los lenguajes de programación y aplicaciones necesarias para desarrollar el proyecto. [ALLAIRE, 96] [MEYERS, 95] [MICROSOFT, 94] [MICROSOFT, 95] [MICROSOFT, 95<sup>a</sup>] [MICROSOFT, 95b] [MICROSOFT, 95c] [MICROSOFT, 97]
- Realizar el análisis orientado a objetos para el desarrollo del proyecto. [MARTIN, 94]
- Realizar el diseño con una metodología orientada a objetos del sistema que será implementado en ABA|TURISMO. [SDD, 94] [MARTIN, 94]
- Diseñar y construir la base de datos para el sistema de condensación de información. [DATE, 93] [MEYERS, 95] [MICROSOFT, 94]
- Diseñar y construir el Front-End del *Sistema de Condensación de Información* con interfaz amigable bajo una filosofía cliente/servidor para alimentar el Data Warehouse. [DUMAS, 88] [MICROSOFT, 95]

- Diseñar y construir el Back-End del *Sistema de Condensación de Información* bajo una filosofía cliente/servidor para alimentar el Data Warehouse. [MEYERS, 95] [MICROSOFT, 94]
- Diseñar y construir el Front-End del *Sistema de Información Ejecutiva ("Intranet")* con una interfaz amigable. [DUMAS, 88] [MICROSOFT, 97]
- Diseñar y construir el Back-End del *Sistema de Información Ejecutiva ("Intranet")*. [ALLAIRE, 96]
- Realizar pruebas y ajustes necesarios. [PRESSMAN, 93]
- Implementar el sistema y capacitar al usuario.
- Documentar cada una de las etapas involucradas en el desarrollo del proyecto.
- Preparar y realizar presentaciones del proyecto.
- Asistir a revisiones con el Asesor y el Usuario.

## V. FUNDAMENTOS DEL PROYECTO

### 5.1 Disciplinas/Metodologías utilizadas.

Este capítulo tiene como objetivo hacer un referencia de la disciplina de desarrollo de soluciones y de las metodologías propuestas para este proyecto.

- ⇒ La disciplina empleada en este proyecto fue SDD (Disciplina para el Desarrollo de Soluciones de Microsoft). [SDD, 94]
- ⇒ La metodología en cual nos enfocamos en este proyecto es el Análisis y Diseño Orientado a Objetos de Martin/Odell. [MARTIN, 94]

#### 5.1.1 SDD.

La disciplina para el desarrollo de soluciones consiste básicamente de lo siguiente:

- Un modelo de aplicación basado en servicios que propone a los desarrolladores el visualizar una aplicación como una red de servicios en

donde las presentaciones y funcionalidades pueden ser empaquetadas para reutilizar sobre límites funcionales.

- Un modelo de proceso de desarrollo iterativo del ciclo de vida que se enfoca, navega entre los riesgos y tiene cuatro puntos de revisión.
- Un modelo de equipo de desarrollo escalable que consiste en seis roles importantes. [SDD, 94]

#### 1. Modelo de Equipo.

La disciplina de desarrollo de soluciones invoca a equipos pequeños, integrados y desarrolladores eficientes con alta experiencia, líderes con conocimientos técnicos elevados y miembros con automotivación.

#### 2. Modelo de Proceso.

La disciplina de desarrollo de soluciones provee un modelo de proceso para planear, estructurar y organizar tareas.

#### 3. Modelo de Aplicación.

La disciplina de desarrollo de soluciones recomienda un arquitectura basada en aproximaciones a soluciones para el desarrollo. Estas aproximaciones permiten a los desarrolladores a:

- Crear plataformas de sistemas modulares.
- Manejo de características heterogéneas de arquitectura Cliente/Servidor.
- Alcance de la meta de la reusabilidad.

SDD es una disciplina y no una metodología debido a lo siguiente:

- Una metodología implica un conjunto de tareas desarrolladas en una secuencia. Mientras la disciplina es una guía para realizar este conjunto de tareas.
- Una metodología son métodos específicos.
- Una metodología son procesos revisados.

#### **Modelo de Equipo.**

En el modelo de equipo se identifican los roles de los miembros del grupo, así como, sus responsabilidades (ver Tabla1).

<b>Rol</b>	<b>Responsabilidades</b>
Administrador del Producto	Investigación, promoción y liberación del producto.
Administrador del Programa	Administración específica, coordina el seguimiento del desarrollo.
Desarrollador	Diseño, Construcción y pruebas.
Capacitación	Diseño de documentación, definición de conceptos y pruebas integrales.
Pruebas de Calidad Integrales	Estrategia, pruebas de calidad y soporte.
Planeación de Logística	Preparación y soporte.

Tabla 1. Tabla de Roles y Responsabilidades.

En el modelo de equipo es de suma importancia la identificación y el cumplimiento de los enfoques que proporciona cada rol de los miembros del grupo (ver Tabla 2).

<b>Rol</b>	<b>Enfoque</b>
Administrador del Producto	Satisfacción del cliente.
Administrador del Programa.	Liberación y entrega a tiempo.
Desarrollador	Un sistema que cumpla con las expectativas.
Capacitando	Un sistema fácil de usar.
Pruebas de Calidad Integrales	Un sistema de calidad (cero errores).
Planeación de Logística	Difusión y migración del sistema.

Tabla 2. Tabla de Roles y Enfoques.

En este modelo se identifican los roles de los miembros del grupo, así como, sus habilidades únicas para el desarrollo del proyecto. (ver Tabla 3).

<b>Rol</b>	<b>Habilidades Unicas del Proyecto.</b>
Administrador del Producto	Comunicación.
Administrador del Programa.	Administración del Proyecto.
Desarrollador	Solución a Problemas.
Capacitación	Documentación Técnica.
Pruebas de Calidad Integrales	Habilidad de dar seguimiento a causas y efectos.
Planeación de Logística	Experto en Ambiente Operativo..

Tabla 3. Tabla de Roles y Habilidades Unicas del Proyecto.

### Modelo de Proceso.

Este modelo de proceso combina características de los modelos de cascada y espiral. También alienta a pensar cómo trabajar en términos de procesos en lugar de tareas. Los puntos de revisión enmarcan el seguimiento de cada proceso a realizar. El modelo de proceso de SDD se deriva del ciclo de vida de un producto (ver Tabla 4).



Figura 4. Modelo de Proceso del SDD.

### Características del Modelo de Proceso.

- La aproximación basada en puntos de revisión. El proceso de desarrollo de la aplicación se le da seguimiento por puntos de revisión externos e internos que sirven como guía para el proceso de desarrollo del proyecto.

- Responsabilidad clara y medible. El modelo conecta la responsabilidad de cada punto de revisión con los roles del equipo en el proyecto.
- Calendarización de manejo de riesgos. Es una técnica en donde los componentes de riesgos altos del proyecto se completan lo más pronto posible.
- Liberación de versiones. El concepto de liberación de versiones es importante para los sistemas, ya que recae en las expectativas, la planeación y administración del proyecto.

### **Definición de los Puntos de Revisión.**

- Punto de revisión sobre la visión. Se proporciona la oportunidad para los clientes y el equipo de ponerse de acuerdo en la visión del proyecto.
- Punto de revisión de especificaciones funcionales. Es en donde el cliente y los miembros del equipo están de acuerdo en el establecimiento de expectativas y prioridades. Pone la oportunidad de bajar los riesgos y permite revalidar estimados de recursos.
- Punto de revisión del código completo. Es en donde los usuarios y el grupo contemplan los detalles finales antes de su liberación.
- Punto de revisión de la liberación del producto. Es cuando la liberación del producto se realiza y pasa del grupo a los grupos de soporte y operaciones.

### **Modelo de Aplicación.**

El modelo de aplicación incluye la notación y las reglas para expresar lo lógico. El modelo de aplicación que una organización adapta fuertemente influye en el carácter de la arquitectura de la aplicación.

#### **5.1.2 Análisis y Diseño Orientado a Objetos.**

Esta metodología modifica el punto de vista del analista de sistemas en cuanto al enfoque de los procesos y su descomposición se refiere llegando a pensar en objetos y el comportamiento de cada uno. El Análisis y Diseño Orientado a Objetos (ADOO) brinda una solución sencilla para sistemas complejos. [MARTIN, 94]

##### **5.1.2.1 Introducción Básica.**

Desde que estamos chicos las personas nos formamos conceptos. Cada concepto es una idea específica o la comprensión de la realidad de nuestro mundo. Estos conceptos que adquirimos permiten al ser humano razonar y sentir las cosas que existen en el mundo. A estos significados sobre los cuales

se aplican nuestros conceptos les llamamos **objetos**. Por ejemplo un objeto sería una factura, una organización, una solicitud. En el ADOO lo que nos importa es el comportamiento del objeto, su estructura y herencia.

Un **objeto** es cualquier cosa real o abstracta, acerca de la cual almacenamos datos y métodos que controlan dichos datos. {MARTIN, 94}

Los conceptos que poseemos utilizan los tipos determinados de objetos. Un **tipo de objeto** es una categoría de un objeto. Por lo tanto un objeto es una instancia de un tipo de objeto.

Un objeto puede estar compuesto por otros objetos. Estos últimos, a su vez pueden estar conformados de otros.

Los **métodos** especifican la forma en que se controlan los datos de un objeto. Los métodos en un tipo de objeto solo hacen referencia a las estructuras de datos de ese tipo de objeto. Un objeto es entonces una cosa que contiene propiedades representadas por tipos de datos y métodos los cuales rigen su comportamiento.

El empaque conjunto de datos y métodos se llaman **encapsulado**. El objeto esconde sus datos de los demás y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultamiento de información.

### 5.1.2.2 Análisis Orientado a Objetos.

#### 5.1.2.2.1 Análisis de la Estructura de Objetos.

El análisis de la estructura de objetos (AEO) define las categorías de los objetos que percibimos y las formas en que los asociamos.

En el AEO se definen tres tipos básicos de información :

- ¿ Qué son los tipos de objetos y como se asocian ? La identificación de los objetos y sus asociaciones se representan mediante esquemas de objetos. Esta información guía al diseñador en la definición de clases y estructuras de datos.
- ¿ Cómo se organizan los tipos de objetos en supertipos y subtipos ? Las jerarquías de generalización se pueden organizar en diagramas e indicar al diseñador las direcciones de herencia.
- ¿Cuál es la composición de los objetos complejos ? Se pueden elaborar diagramas de jerarquías compuestas. La composición guía al diseñador en la definición de mecanismos que controlen adecuadamente a los objetos dentro de otros objetos.

Las asociaciones de objetos definen cómo asociar un objeto con otro. Por ejemplo, la asociación entre un objeto persona y otro objeto llamado organización.

Las jerarquías de generalización es la organización de un gran volumen de información que va desde lo general hasta lo más específico. La generalización es el resultado de distinguir un tipo de objeto de lo más general, o incluso, que es más que otro. Todo lo que se aplique a un tipo de objeto también se aplica a sus subtipos. Cada instancia de un tipo de objetos es también una instancia de sus subtipos.

Diagramas de relación entre los objetos. Los tipos de objetos están relacionados con otros tipos de objetos. Los diagramas de relación entre objetos muestra la relación entre los tipos de entes.

#### **5.1.2.2.2 Análisis del Comportamiento de Objetos.**

En el análisis del comportamiento de objetos (ACO) se realizan esquemas de eventos que muestran eventos, la secuencia en que ocurren y cómo los eventos cambian el estado de un objeto.

En el análisis del comportamiento de objetos se identifica la siguiente información:

- ¿ En qué estados puede estar un objeto ? Un objeto puede tener varios conjuntos de estados.
- ¿ Qué transacciones de estado se pueden dar ? Estas se determinan en el diagrama de transición de estado.
- ¿ Qué eventos ocurren ?
- ¿ Qué operaciones se llevan a cabo ? Se traza un esquema de eventos, para mostrar la secuencia de operaciones y eventos.
- ¿ Qué interacciones ocurren entre objetos ? Se traza un diagrama para mostrar los mensajes transferidos entre las clases.
- ¿ Cuáles son las reglas de activación que utilizan los objetos para reaccionar ante el evento ?
- ¿ Cómo se representan las operaciones en los métodos ? Se utilizan los diagramas, enunciados declarativos u otros medios para determinar los métodos con detalles suficientes como para generar código.

Estados de un Objeto. Un objeto puede existir en varios estados. Por ejemplo, un objeto Reservación Aérea puede ser una instancia de alguno de los siguientes tipos de objetos: reservación solicitada, en lista de espera,

confirmada, cancelada, satisfecha y activada. El estado de un objeto es la colección de asociaciones que tiene un objeto.

**Evento.** En el análisis orientado a objetos el mundo se describe en términos de los objetos y sus estados, así como de los eventos que modifican estos estados. Un evento es el cambio de estado de un objeto. Por ejemplo, el objeto de reservación en lista en espera ahora puede pasar al objeto de reservación confirmada.

**Tipos de Eventos.** Los tipos de eventos indican los cambios sencillos en el estado de un objeto. Por ejemplo, cuando se deposita dinero en una cuenta bancaria o se actualiza el salario de un empleado. Básicamente, los tipos de eventos describen las siguientes formas de cambios de estado:

- Un objeto se crea.
- Un objeto se termina.
- Un objeto se clasifica como instancia de un tipo de objeto.
- Un objeto se deja de clasificar como instancia de un tipo de objeto.
- Un objeto cambia de clasificación.
- El atributo de un objeto cambia.

**El ciclo vital de un objeto.** La mayoría de los objetos tienen un ciclo de vida en la que una sucesión de eventos pueden ocurrirle y cada uno de estos modifica su estado.

**Interacciones entre tipos de objetos.** Los diagramas de transición de estado son útiles para expresar el ciclo vital de un objeto particular. Sin embargo, la mayoría de los procesos requieren la interacción de varios objetos. Cuando los diagramas se expresan así, es fácil ver la forma de implementarlos como estructuras de un programa orientado a objetos (OO): los tipos de objetos cambian de estado y las operaciones se convierten como operaciones OO.

**Operaciones.** En el análisis OO, el término operación se refiere a una unidad de procesamiento que puede ser solicitada. El procedimiento se implanta mediante un método. El método es la especificación de cómo llevar a cabo la operación. Es el guión de la operación. A nivel programa, el método es el código que implanta la operación. Las operaciones se invocan. Una operación invocada es una instancia de una operación.

### **5.1.2.3 Diseño Orientado a Objetos.**

#### **5.1.2.3.1 Diseño de la Estructura y del Comportamiento de Objetos.**

El análisis orientado a objetos tiene dos aspectos: el análisis de la estructura de objetos (AEO) y el análisis del comportamiento de objetos (ACO).

En el diseño orientado a objetos, se debe describir también en dos partes: diseño de la estructura de objetos y diseño del comportamiento de objetos (DEO y DCO).

En el diseño de la estructura y comportamiento de objetos se identifican los componentes siguientes:

- ¿ Qué clases se implementarán ? Los tipos de objetos del AEO serán la guía en esta decisión.
- ¿ Qué estructura de datos utilizará cada clase ? Se puede hacer un diagrama para representar la estructura de datos.
- ¿ Qué operaciones ofrecerá cada clase y cuáles serán sus métodos en determinado momento ?
- ¿Cómo se implantarán la herencia de clases y cómo afectará ésta las especificaciones de los datos y operaciones?
- ¿ Cuáles son las variantes ? Se identifican las probables variantes de las clases. ("Igual que, excepto..." se aplica a la mayoría de los componentes reutilizables.)

En el análisis de estructura de objetos, identificamos los tipos de objetos; en el diseño de estructura de objetos nos centramos en la implantación de estos tipos de objetos. La clase se define como la implantación de un tipo de objeto. Especifica la estructura de datos y los métodos operativos permitidos que se aplican a cada uno de sus objetos.

La clase especifica la estructura de datos de cada uno de sus objetos y las operaciones que se utilizan para tener acceso a los objetos. Las especificaciones de cómo se llevan a cabo las funciones de cada clase se llaman métodos. Los objetos se pueden utilizar exclusivamente con métodos específicos.

¿Cuál es la diferencia entre operación y método ? Las operaciones son procesos que se pueden solicitar como unidades. Los métodos son especificaciones del procesamiento de una operación dentro de una clase. Es decir, la operación es el tipo de servicio y el método es su código de programación.

## **5.2 Data Warehouse.**

### **5.2.1 Breve Historia.**

En el transcurso de los años, las empresas han acumulado cantidades muy grandes de datos y son almacenadas en bases de datos. Este tipo de depósitos contiene información histórica de la empresa relacionada con elementos tales como el desempeño corporativo, la competencia, los clientes,

los productos, la experiencia y los presupuestos. Desde los años setenta se esperaba que esta información pudiese ser analizada para obtener ventajas estratégicas. Se encontraban obstáculos para que este conjunto de información se consideraran como una unidad. Es aquí donde surge un concepto llamado Data Warehouse el cual brindará la integración de información y aprovechamiento de ésta para su análisis empresarial.

Los sistemas de información se actualizan constantemente. Desde las primeras raíces de 1950 para formar bases de datos las cuales ayudaran a automatizar el día con día del negocio. Hasta los sistemas manejadores de bases de datos los cuales se definían en un modelo relacional en los años 70's ahora brinda la oportunidad al tomador de decisiones de tener ese modelo relacional para adaptarlo según sus necesidades.

### **5.2.2 Definición**

Data Warehouse:

Es una colección de datos definida con el propósito de apoyar el proceso de la toma de decisiones de la alta gerencia. Los elementos de entrada al Data Warehouse provienen de un ambiente operacional no integrado en casi todos los casos. El Data Warehouse siempre es el almacenamiento de datos que se encuentra físicamente separado del ambiente operacional. [INMON, 94]

Características de un Data Warehouse:

- Orientado a una materia.
- Integrado.
- De tiempo variante.
- No volátil.

Orientado a una materia.

Organiza y orienta los datos desde la perspectiva del último usuario. Muchos sistemas operativos organizan sus datos desde la perspectiva de la aplicación, de modo que el acceso de la aplicación a los datos tenga la mayor eficiencia posible. Con frecuencia, la información que está organizada para que una aplicación del negocio la recupere y actualice con facilidad, no está organizada necesariamente de modo que un analista con herramientas gráficas inteligentes de consulta pueda formular las preguntas empresariales correctas.

Integración.

La integración de los datos del Data Warehouse se puede observar de diferentes maneras: se encuentran en nombramientos convencionales consistentes, medidas de las variables de manera consistente, consistencias de la estructuras de código, consistencia en los atributos físicos de los datos,

etcétera. La esencia de la integración del Data Warehouse es permitir integrar los datos que se encuentran en varias aplicaciones. Por ejemplo, supongamos que una aplicación A define el atributo sexo como F y M. Una aplicación B define el atributo sexo como 1 para femenino y 0 para masculino. Una tercera aplicación C muestra el atributo sexo como FEMENINO y MASCULINO. La integración del Data Warehouse sería definir el atributo sexo como F y M, como si fuese un estándar.

#### Variación en el Tiempo.

Los datos del Data Warehouse son consistentes en un momento del tiempo. Esta característica de los datos en el Warehouse es muy diferente de los datos encontrados en un ambiente operacional. En el ambiente operacional, los datos son consistentes desde el momento de acceso. En otras palabras, cuando tu accedas una unidad de datos, uno espera que refleje valores consistentes desde el momento en que se accesó. Debido a que los datos en el Data Warehouse son consistentes solamente un momento del tiempo, se define al mismo como que es de tiempo variante. La varianza del tiempo de los datos de un Data Warehouse se muestra de diferentes maneras. Una de las maneras más simples de representar los datos en un periodo largo de tiempo (de 5 a 10 años) mientras que en un ambiente operacional los datos son representados en un periodo corto de tiempo (de 60 a 90 días).

#### No volátil.

La cuarta característica del Data Warehouse es que no es volátil. Esto significa que las transacciones del Data Warehouse tienden a buscar datos y no a actualizarlos.

### 5.2.3 Arquitectura de Referencia.

La arquitectura de referencia básicamente define una solución de manera unificada para realizar un Data Warehouse. Es importante definir la arquitectura de referencia ya que ésta divide la solución en componentes bien definidos. Cada uno de los componentes puede construirse o comprarse completo, adquirirse en partes e integrarse en una solución.

La arquitectura de referencia se describe primero desde un punto de vista abstracto y simplificado a alto nivel, de la siguiente manera: [GILL, 94]

- Un conjunto de datos extraídos de bases de datos operacionales.
- Un software que prepara los datos para que los accesen los usuarios.
- Un conjunto de aplicaciones y herramientas que ejecutan un grupo de consultas y análisis complejos.

Después descompondremos sistemáticamente la arquitectura de referencia en mayores detalles. Esto ayuda a comprender un diagrama conceptual simple, al mismo tiempo que expone los detalles necesarios para una implementación ordenada y correcta.

La figura siguiente muestra los componentes de nivel superior de la arquitectura de referencia. La arquitectura de referencia del Data Warehouse divide los componentes del Data Warehouse en bloques como la fuente de datos, el mercado de datos, el Data Warehouse. Acceso y uso, y capas tales como administración de datos, administración de metadatos, Transporte, e infraestructura. Los bloques se relacionan con la funcionalidad específica del Data Warehouse. Las capas representan el ambiente necesario para implementar los bloques.

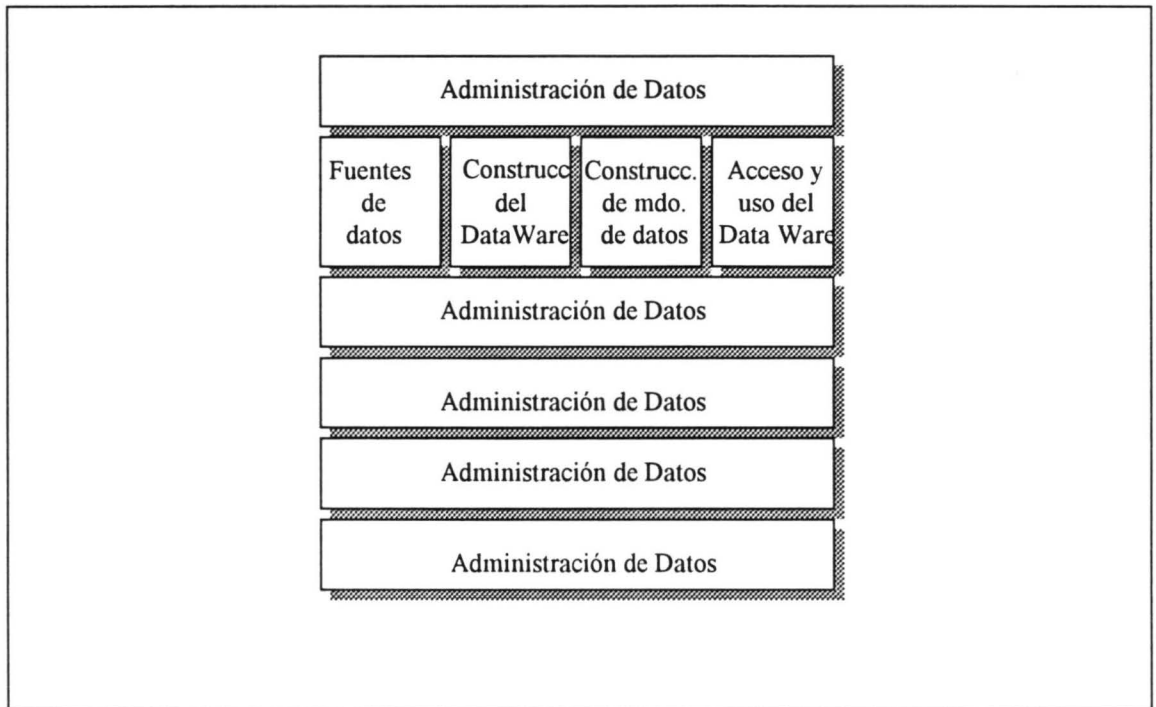


Figura 5. Panorama de la Arquitectura de Referencia del Data Warehouse

#### Bloque de Fuente de Datos.

El bloque de datos se divide en las siguientes categorías:

- Datos de producción. Los datos de producción se refieren a las bases operacionales de datos que contienen la información recopilada de las aplicaciones operacionales. Estas bases de datos operacionales vienen como tecnologías diversas, tales como relacionales, no relacionales o basadas en archivos.

- Datos de herencia. Los datos de herencia están fuera de línea, en archivos perdidos, porque ya no son necesarios para apoyar aplicaciones operacionales. Sin embargo, estos datos tienen un gran valor histórico para analizar tendencias y deben introducirse en un Data Warehouse imprimiendo correctamente la fecha. Esta información es muy útil para la minería de datos, la cual será explicada más adelante.
- Sistemas internos de oficina. Existen fuentes de datos que no están almacenados en una base de datos operacional ni los usa una aplicación operacional del negocio. En general, estos datos son los siguientes:
  1. No estructurados, tal como las formas no electrónicas.
  2. Estructurados, tal como en los reportes, las hojas de cálculos y los documentos de procesadores de palabras.
  3. Semiestructurados, tal como en los reportes anuales o archivos.
- Sistemas Externos. Estas fuentes de información no las controla, posee ni opera la empresa.
- Metadatos para fuentes. Los metadatos son la información de definición acerca de los datos de las fuentes. Los metadatos incluyen el nombre capturado y extraído de las fuentes de información, la definición del contenido de los datos (campo), la fecha que fueron creados y la fuente u origen de los datos.

#### Bloque de construcción del Data Warehouse.

Este bloque se divide en los siguientes componentes:

1. El componente de refinamiento es responsable de estandarizar, limpiar y sacudir, filtrar y confrontar, e imprimir la fecha de origen de la información extraída de las fuentes de datos seleccionadas. En el componente de refinamiento se ubican los metadatos para los nombres y definiciones de datos estándar. Se crean y capturan metadatos adicionales para imprimir la fecha en que los datos extraídos, para las fuentes de datos extraídos y para los campos faltantes que se hayan agregado.
2. El componente de reingeniería es responsable de preparar los datos para que sean congruentes con las necesidades del análisis del usuario empresarial.

#### Bloque de construcción del Mercado de datos.

El bloque de construcción del mercado de datos es usado para crear un mercado de datos a partir del contenido del Data Warehouse. Los componentes para construir un mercado de datos son los siguientes:

- Modelar
- Condensar
- Agregar
- Conciliar y validar

- Desarrollar consultas basadas en la arquitectura
- Crear glosario
- Examinar y navegar por los metadatos

#### Bloque de Acceso y uso del Data Warehouse.

Este bloque tiene dos componentes principales que son:

1. El componente de acceso y recuperación proporciona el acceso directo al Data Warehouse, evitando el mercado de datos y al mercado de datos. Es responsable de transformar los datos recuperados en vistas multidimensionales o de almacenarlos en una base de datos multidimensional para un análisis posterior.
2. El componente de análisis y reporte es responsable de la familia de herramientas y aplicaciones necesarias para aprovechar el Data Warehouse y el mercado de datos. Las herramientas se clasifican en: de reporte, de análisis y soporte de decisiones, de modelado empresarial, y procesamiento analítico, y de minería de datos. Las herramientas para navegar por los metadatos para entender y reportar el contenido del Data Warehouse/mercado de datos son también responsabilidad de este componente.

#### Capa de administración de datos.

Para el Data Warehouse los elementos importantes para esta capa son los siguientes:

- Extracción de los datos apropiados a fin de seleccionar las fuentes de información para el refinamiento posterior, la reingeniería y la incorporación en el Data Warehouse.
- Seguir y llenar las solicitudes de datos nuevos por parte de fuentes de datos nuevos y actuales.
- Capturar los datos correctos en las fuentes de datos operacionales y después actualizar el Data Warehouse.

En esta capa de administración de datos se incluyen las políticas estándar de administración de datos, procedimientos, los programas y operaciones de seguridad, las autorizaciones de acceso, archivar, y restaurar y desechar los datos.

#### Capa de administración de metadatos.

La capa de administración de metadatos se encarga de los siguientes puntos:

- Las definiciones estándar de los datos incluyendo las definiciones técnicas y empresariales de los datos depositados en el Data Warehouse.

- Los metadatos capturados y creados en los bloques de refinamiento y reingeniería.
- Los metadatos acerca de granularidad, segmentación, áreas, tema, adición y condensación.
- Los metadatos que describen las consultas y reportes predefinidos y diseñados.
- Los metadatos que describen índices y perfiles que mejoran el rendimiento en el acceso y la recuperación de datos.
- Los metadatos que describen las reglas para preparar y programar el ciclo de actualización y duplicación.

#### Capa de Transporte.

En esta capa se utiliza tecnología de actualización y duplicación, red de transferencia y entrega de datos, y componentes de middleware. Dicha capa de transporte soluciona los problemas de comunicación entre las plataformas de hardware/software que se encuentran separadas debido a la segmentación de plataformas de los diversos bloques de la arquitectura de referencia.

#### Capa de Infraestructura.

A continuación se explican los diferentes componentes que maneja esta capa:

- El componente de administración de sistemas ofrece la capacidad de que el desarrollador de sistemas y el usuario empresarial invoquen, manejen y terminen las herramientas y las aplicaciones.
- El componente de administración de flujo de trabajo sustenta el proceso de integración y administración para coordinar la ejecución ordenada y especificada de las herramientas, aplicaciones y actividades para extraer, actualizar, duplicar, agregar y resumir el Data Warehouse y el mercado de datos. Además de otras tareas de mantenimiento y administración de sistemas.
- El componente de sistemas de almacenamiento aporta servicios de administración de la base de datos y archivos para la fuente de datos, el Data Warehouse y los catálogos de la base de datos del mercado de datos, al igual que el depósito multidimensional y local para el bloque de acceso y uso.
- El componente de sistemas de procesamiento mantiene los ambientes operativos en los bloques principales de fuentes de datos, el Data Warehouse y el mercado de datos, las herramientas de acceso y uso, el middleware, y otros componentes de la infraestructura que ya se hayan analizado.

#### 5.2.4 Enfoque de selección de una arquitectura.

Para implementar un Data Warehouse es muy importante definir bajo que arquitectura hay que enfocarse:

- El almacenamiento operacional en comparación con el uso de copias de datos operacionales. La visión tradicional que cualquiera se imagina de un Data Warehouse es que hace copias de información operacional e histórica y las almacena para uso privado. Pudiera ser deseable eliminar de manera arquitectónica la necesidad de copiar y utilizar los datos operacionales almacenados en aplicaciones solo de lectura que no los altere. En una arquitectura así el metamodelo del Data Warehouse es un esquema virtual que apunta a los metadatos de base de datos operacionales. Las consultas al Data Warehouse simplemente accederán los datos de las bases de datos operacionales, directamente bajo la guía del metamodelo del Data Warehouse.
- Un sólo Data Warehouse. Esta arquitectura reconoce que, por lo general, todas las aplicaciones del Data Warehouse requieren diversas operaciones que se aplican sobre las fuentes de datos, como depuración, integración, resumen y adición. Por lo tanto, tiene sentido aplicar estas operaciones una vez y traer los datos de las fuentes a un Data Warehouse centralizado. En una arquitectura como ésta, un sólo Data Warehouse lógico alimenta a todos los usuarios finales con información para apoyo en las decisiones.
- Un sólo mercado de datos. Una arquitectura reconoce que cada departamento funcional en una organización tiene sus propias necesidades específicas y que un sólo Data Warehouse corporativo no puede satisfacer todas estas necesidades.
- Data Warehouse y mercado de datos. Esta arquitectura es un reconocimiento de que las necesidades de Data Warehouse específicas de un departamento se deben abordar junto con la necesidad de un Data Warehouse corporativo. Este último actúa como un recopilador y distribuidor de información a partir de fuentes de datos por toda la organización.
- Separación de plataformas e infraestructura. La flexibilidad arquitectónica con la que cuenta tiene la finalidad de compartir plataformas entre las fuentes de datos y el Data Warehouse o para emplear plataformas separadas para las fuentes de datos, el Data Warehouse, los mercados de datos y las estaciones de trabajo de los usuarios. El hecho de compartir plataformas reduce la complejidad de extracción y transformación de datos.
- Arquitectura Cliente/Servidor de dos capas. En la arquitectura cliente/servidor de dos capas, una capa contiene a los plataformas. El servidor pudiera ser una estación de trabajo, o bien, un anfitrión de macrocomputadora. En esta opción, las herramientas de acceso del usuario final se ejecutan en el cliente, en las fuentes de datos; el Data Warehouse y el mercado de datos residen en servidores, ya sea en uno o en distintos servidores. Los clientes son

aplicaciones gráficas que dependen de la capacidad de procesamiento de la estación de trabajo del cliente.

- Arquitectura Cliente/Servidor de tres capas. En esta opción arquitectónica hay tres capas: una capa cliente basada en una estación de trabajo, una capa intermedia con base en un servidor y una tercera capa establecida en una macrocomputadora. La capa basada en macrocomputadora es responsable de manejar las fuentes de datos y las transformaciones opcionales de las fuentes de datos.

#### **5.2.4.1 Construcción de un Data Warehouse en Arquitectura Cliente/Servidor.**

La arquitectura Cliente/Servidor es una clave para el éxito de la implementación de un Data Warehouse. Sin esta tecnología los beneficios del Data Warehouse no serían reconocidos. Muchas organizaciones están utilizando el Data Warehousing como otra entrada al ambiente de Cliente/Servidor. Para mucha gente es necesario aplicar este tipo de tecnología realizando una aplicación piloto.

Al principio en las aplicaciones que se construían bajo un esquema Cliente/Servidor una de las primeras cosas que se tenía que aprender, era observar la falta de partición en las primeras generaciones de las herramientas. Este tipo de herramientas sufrían un síndrome llamado "Cliente Obeso" el cual significaba que el procesamiento de los datos se realizaba en el cliente y solamente el servidor extraía datos antes de ser procesados. Cuál fue la solución para esto, el realizar todos estos procedimientos en el servidor.

Mientras la nueva tecnología brinde a los desarrolladores algo diferente, éstos necesitan siempre utilizar la capacidad de una guía para observar las capacidades de sus aplicaciones.

Cuál es la ventaja que nos ofrece el Data Warehouse construido bajo los esquemas de una arquitectura Cliente/Servidor, como solución, simplemente el dar apoyo a los administradores a monitorear lo que se tiene.

#### **5.2.5 Los requerimientos de un Data Warehouse.**

El autor nos expone los siguientes puntos para formular los requerimientos de un Data Warehouse. [GILL, 94]

- Un Data Warehouse es simplemente un sistema de aplicación empresarial con su propia base de datos. Esta base de datos es generada a partir de otras bases de datos operacionales, no de información inicial que se introduce. El Data Warehouse ofrece características y funciones para

implementar procesos empresariales y enlazarlos con otros procesos fuera del ámbito del Data Warehouse. En forma muy similar a otros sistemas de aplicaciones empresariales, se requiere que el Data Warehouse proporcione al usuario final estos conjuntos preescritos de características y funciones del modo más eficiente posible.

- Un Data Warehouse es de capacidad latente. Almacena información resumida que se organiza de acuerdo con temas empresariales, tales como clientes y productos, para analizar la información con más facilidad. La carga de mostrar, organizar y reportar la información que guarda el Data Warehouse corresponde a las herramientas que deben incorporarse en el mismo. En ese sentido, un Data Warehouse posee una capacidad latente que solo se vuelve útil cuando las herramientas de análisis y reporte se aplican con inteligencia a los datos que se conservan en el Data Warehouse. Se requiere que el Data Warehouse sostenga un extenso rango de herramientas de acceso, operadas por una extensa gama de usuarios finales. El Data Warehouse también debe guardar y administrar un ámbito de información grande para servir a una extensa clientela.
- El Data Warehouse es una base de datos históricos, la cual es una acumulación de muchos años de información transaccional en línea, organizada para hacer más eficiente el almacenamiento y facilitar la recuperación. Es necesario que el Data Warehouse organice grandes cantidades de información de manera compacta y eficiente. También se requiere que proporcione técnicas para resumir, a fin de que los usuarios finales comprendan las lecciones de los antecedentes con más facilidad.
- En ocasiones, el Data Warehouse es una tienda de datos operacionales. Entrega información operacional a un amplio rango de usuarios copiando información de los sistemas operacionales de base de datos. En este caso, se requiere que el Data Warehouse distribuya información operacional de manera eficiente a un gran rango de usuarios. También se necesita que el Data Warehouse haga los cambios tecnológicos necesarios para mover la información de su base de datos operacional a la tecnología de almacenamiento que se emplea en el Data Warehouse.

### **5.2.6 La Importancia de los metadatos.**

Al comenzar a construir un proyecto que involucra Data Warehouse es necesario empezar con los metadatos. El concepto de metadato se define como datos acerca de los datos. En una base de datos, los metadatos son la representación de los diversos objetos que definen una base de datos. En una base de datos relacional, esta representación consistiría en las definiciones de las tablas, columnas, bases de datos, visualización y otros objetos. Entonces se utilizarán los metadatos para hacer referencia a todo lo que defina un objeto del Data Warehouse (ya sea una tabla, una columna, un reporte, una consulta, una

regla empresarial o una transformación dentro de un Data Warehouse). [GILL, 94]

### ¿Qué son los metadatos?

Los metadatos impregnan todos los aspectos del Data Warehouse. Los metadatos constan de los siguientes tipos de elementos: [GILL; 94]

- Ubicación y descripción de servidores, bases de datos, tablas, nombres y resúmenes del Data Warehouse.
- Reglas para la profundización automática al detalle o al resumen y a través de jerarquías de dimensión empresarial, tales como productos, mercados y cuadros contables.
- Nombres elegidos o alias definidos por el usuario final para los encabezados y hechos de datos con nombres más técnicos.
- Reglas para cálculos personalizados definidos por el usuario final.
- Seguridad a nivel personal, de grupo de trabajo y de empresa, para visualizar, cambiar y distribuir resúmenes adaptados, cálculos y otros análisis del usuario final.
- Descripciones de fuentes originales y transformaciones.
- Definiciones lógicas de tablas y atributos del Data Warehouse.
- Definiciones físicas de tablas y columnas, así como de sus características.
- Ubicación integrada de las tablas del Data Warehouse.
- Antecedentes de extracción.
- Información de alias.
- Algoritmos de resumen.
- Ubicación de área tema.
- Antecedentes de relaciones.
- Propiedades/Herencia.
- Patrones de acceso.
- Tablas de referencia y datos codificados.
- Indicador de calidad de datos.
- Seguridad.
- Unidades de medida.

En forma adicional, los metadatos pudieran también contener componentes de ubicación para auxiliar en las siguientes tareas:

- Identificación de fuentes operacionales.
- Ubicación sencilla de atributo a atributo.
- Conversiones de atributos.
- Conversiones de características físicas.
- Conversiones de codificación y tabla de referencia.
- Cambios de nombre.
- Cambios de llave.

- Valores predeterminados que se utilizan.
- Razón predeterminada.
- Lógica para elegir entre varias fuentes operacionales.
- Formulación algorítmica empleada.

A continuación veremos un ejemplo de un elemento de metadato.

Nombre de la entidad	Cliente
Nombre alternos	Cliente
Definición	Un cliente puede llegar a ser una persona o una empresa.
Fecha de creación	1 de Febrero de 1997
Fecha de última actualización	5 de Abril de 1997
Llaves	Clienteld, ClienteNombre
Ciclo de actualización	Extraído cada mes
Ciclo de archivado	Archivado después de seis meses
Gerente de Datos	Adriana Olmeda
Propietario de Datos	Rolando Torres
Patrones de acceso	Fecha de último acceso 5 de mayo de 1997

Tabla 4. Ejemplo de un elemento de metadato.

### ¿Por qué son importantes los metadatos?

Los metadatos son como el mapa de carreteras hacia los datos. En forma muy parecida a la que una ficha de catálogo de una biblioteca apunta tanto al contenido como a la ubicación de los libros de una biblioteca.

A continuación se definen puntos importantes de los metadatos.

- Debe de servir como un mapa de las ubicaciones en donde se encuentra la información almacenada en el Data Warehouse. Y así de esta manera se conocerá que se encuentra guardado en el Data Warehouse.
- En su forma óptima, el catálogo del Data Warehouse debe tener un componente de definición para cada elemento – la definición que requiere la tecnología de base de datos (tal como el nombre de la tabla, el propietario de la tabla, el tipo de tabla, los nombres de las columnas, los tipos de datos y los valores predeterminados).

- Debe proporcionar un anteproyecto de la forma en la que un tipo de información se deriva de otro. El resumen es un buen ejemplo de dicha derivación.
- Debe proporcionar un anteproyecto de los mecanismos que extraigan datos de las aplicaciones operacionales y los depositen en el Data Warehouse.
- Debe almacenar las reglas empresariales incorporadas en el Data Warehouse. Las reglas empresariales son los lineamientos formales que establecen las organizaciones para resolver las inconsistencias de datos.
- Debe almacenar las reglas de control de acceso y seguridad para ofrecer una administración confiable. Los controles de acceso y seguridad dependen de permisos que se conectan a los metadatos.
- Los metadatos deben registrar los cambios a través del tiempo. Con frecuencia la organización de los datos que reúne a una corporación a través de los años también ha cambiado.
- Los metadatos deben registrar la fecha de su creación para captar su cambio histórico.
- Se requiere almacenar la estructura y el contenido del Data Warehouse.
- Se debe definir con claridad e identificar formalmente el sistema de registro para el Data Warehouse.
- Como una parte regular del Data Warehouse, debe estar disponible la lógica de transformación e integración que transfiere los datos del ambiente operacional al ambiente del Data Warehouse.
- Como parte de los metadatos, debe almacenarse la historia de las actualizaciones, para permitir a los usuarios determinar la periodicidad y precisión.
- Se requiere almacenar un esquema de medidas, de modo que el usuario final pueda determinar si una consulta será larga o corta antes de proponerla.

### **La importancia de los metadatos durante el desarrollo del Data Warehouse.**

Un aspecto importante de los metadatos es la necesidad de ubicar con mapas desde las fuentes hasta el Data Warehouse a través de un proceso de extracción, refinamiento y reingeniería. Se deben mantener estos mapas para cumplir los siguientes objetivos:

1. Verificación de la calidad de datos. Los mapas contienen información sobre los diversos cambios que han sufrido los datos antes de integrarse al Data Warehouse.
2. Sincronización y actualización. Conforme se genera nueva información operacional, se debe actualizar en el Data Warehouse, los nuevos datos

deben experimentar las mismas transformaciones que los datos cargados antes en el Data Warehouse.

3. Integración. Los mapas establecen relaciones entre los datos que reflejan las reglas empresariales de interés para los usuarios finales.

### **5.2.7 Ciclo del desarrollo del Data Warehouse.**

#### **5.2.7.1 Análisis del Data Warehouse.**

La fase del ciclo de desarrollo del data Warehouse significa convertir los requerimientos acoplados en la fase anterior de requerimientos, en un conjunto de especificaciones que puedan apoyar al diseño. En lo abstracto, hay tres especificaciones principales de entrada para el Data Warehouse. Estas son:

- Requerimientos de enfoque empresarial que delínean las fronteras de la información que debe comprender el Data Warehouse. El enfoque empresarial determinará también la audiencia y sus requerimientos de información.
- Especificaciones de requerimientos de fuentes de datos que delínean las fronteras de información disponible en las fuentes de datos actuales.
- Especificaciones de requerimientos de usuario final y acceso, las cuales definen como se utilizará la información del Data Warehouse. Junto con éstas se encuentra la especificación de los tipos de herramientas y técnicas de exhibición que utilizan.

Por lo tanto el proceso de análisis consiste en derivar modelos físicos y lógicos de datos para el Data Warehouse y los mercados de datos, y definir los procesos necesarios para conectar las fuentes de datos, el Data Warehouse, los mercados de datos y técnicas de exhibición que requieren.

#### **5.2.7.2 Diseño del Data Warehouse.**

En la fase del diseño, los modelos lógicos desarrollados en la fase del análisis se convierten en modelos físicos. También se identifican y detallan los procesos que requiere, de manera interna, cada bloque de la arquitectura de referencia del Data Warehouse.

Data Warehousing es un proceso de integración de los datos de una organización corporativa en un depósito simple e donde los usuarios pueden realizar un análisis de los datos. Y así podemos ver al Data Warehousing como una estrategia para el negocio. {INFORMIX, 96}

### Modelo Clásico de Entidad-Relación y la Toma de Decisiones.

Las herramientas para apoyar a la toma de decisiones promueve el análisis de los datos de un forma bastante compleja. Los modelos de datos diseñados para apoyar Data Warehouses necesitan estrategias de optimización que puedan manejar este tipo de software.

Una de las muchas razones donde el modelo Entidad-Relación fracasa es en el contexto de la toma de decisiones porque tiene un rendimiento demasiado pobre.

Modelos de datos normalizados son diseñados para proveer el acceso a los datos de una manera eficiente para un gran número de transacciones involucrando muy pocos registros. En sistemas para el apoyo a la toma de decisiones son pocas las transacciones, sin embargo, involucran un gran volumen de registros.

A continuación se presenta una comparación entre las bases de datos operacionales y de apoyo a la toma de decisiones.

<b>Funciones</b>	<b>Operacional</b>	<b>De Apoyo a la Toma de Decisiones</b>
<b>Contenido de los datos</b>	Valores Actuales	Datos históricos , datos sumariados, datos calculados.
<b>Organización de los datos</b>	Aplicación por aplicación	Áreas a través de la empresa
<b>Naturaleza de los datos</b>	Dinámico	Estático hasta que se actualice
<b>Estructura y formato de los datos</b>	Compleja, apta para computación operacional	Sencilla, apta para análisis del negocio
<b>Probabilidad de acceso</b>	Alta	Moderada a baja
<b>Actualización de los Datos</b>	Actualización campo por campo	Accesada y manipulada y no directamente actualizada
<b>Uso</b>	Procesamiento estructurado de alta repetición	Procesamiento no estructurado de análisis
<b>Tiempo de Respuesta</b>	Segundos	de segundos a minutos

Tabla 5. Comparación entre las base de datos operacional y para apoyo a la toma de decisiones.

## Modelo Dimensional.

El modelo dimensional es una técnica de desarrollo para datos estructurados alrededor de conceptos naturales del negocio y provee una base para un análisis de datos sofisticado.

Los modelos tradicionales de Entidad-Relación constan de entidades y relaciones. La estrategia se enfoca en desmenuzar la información en muchas tablas, donde cada una describe exactamente una entidad.

En cambio en el modelo dimensional la estructura de datos esta organizada en medidas y dimensiones. Las medidas son los datos numéricos trazados. Y éstas están almacenadas en tablas centrales llamadas "fact tables". Las dimensiones son los parámetros naturales del negocio que define cada transacción. Las dimensiones están almacenadas en tablas unidas a las "fact tables".

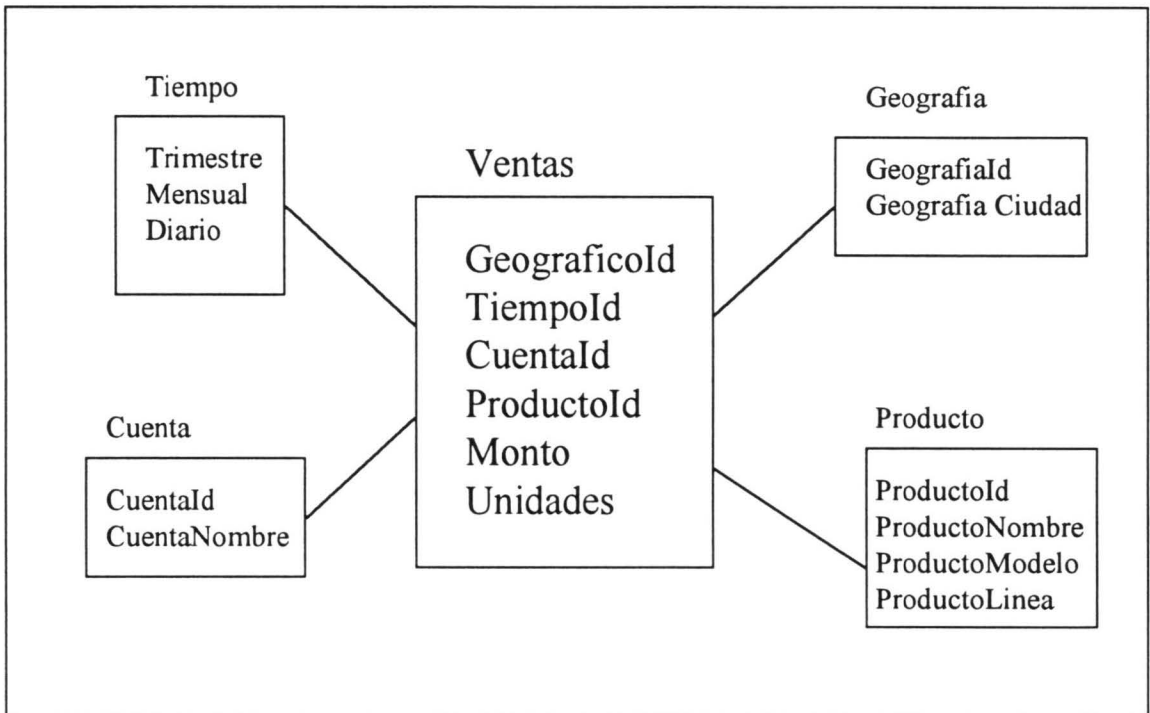


Figura 6. Ejemplo de un modelo Dimensional.

Este esquema muestra que tiene cuatro dimensiones y una medida.

Las ventajas del modelo dimensional son:

- Permite estructuras de datos complejas y multidimensionales para ser definidas con modelos de datos simples.
- Reduce el número de uniones físicas que el “query” tiene que hacer en un proceso.
- Simplificando la visualización del modelo de datos reduce el tiempo de “queries” muy grandes.
- Permite que el Data Warehouse se expanda requiriendo relativamente poco mantenimiento.

### **Esquema Estrella.**

La arquitectura física de un modelo relacional se describe por los esquemas de estrella. Como vimos en el ejemplo anterior, ver figura 6., la tabla central se encuentra rodeada por tablas formando una estrella.

### **Desnormalización de Dimensiones.**

Una característica de la definición del esquema de estrella es que las tablas de dimensiones se encuentran desnormalizadas. La desnormalización es una aproximación del diseño de una base de datos en donde los datos se almacenan respectivamente en tablas individuales para la simplicidad del diseño y mejoramiento del rendimiento. Y las atributos dimensionales se deben almacenar en tablas de múltiples tiempos.

### **Agregaciones.**

Las agregaciones son el proceso en el cual los datos de un nivel bajo son sumariados y son colocados en tablas intermedias que almacenan información que se va a agregar.

Una arquitectura de un Data Warehouse típico comienza con un almacenamiento masivo de transacciones en un nivel atómico o bajo. Las medidas son almacenadas en un “fact table” principal en la forma más detallada, a fin de que en las siguientes etapas del análisis de datos y reporte se puedan utilizar.

Pero la extracción de datos del nivel atómico no tiene el rendimiento adecuado, ya que las “fact tables” tienden a ser muy grandes. Es por eso que es más útil para los datos accesados la presumarización. Esto permite resultados intermedios o agregaciones que pueden ser utilizadas reduciendo significativamente los recursos requeridos para enviar los resultados finales de una consulta. Consideraciones para determinar que agregaciones se van a crear:

1. La densidad de los datos. ¿ Dónde se encuentran los datos concentrados, y en qué dimensión los elementos incrementan el número de sus renglones ?
2. Patrones utilizados. ¿ Cuáles agregaciones deben mejorar el performance para las consultas que vaya a realizar el usuario final. ?

### El Esquema “Snowflake”.

Cuando se realizan normalizaciones a las dimensiones se dice que se convierte en el esquema “Snowflake”. El esquema de snowflake puede mejorar el rendimiento almacenando en cada tabla de dimensión una llave. La principal desventaja de esto es que ocupa más espacio en disco.

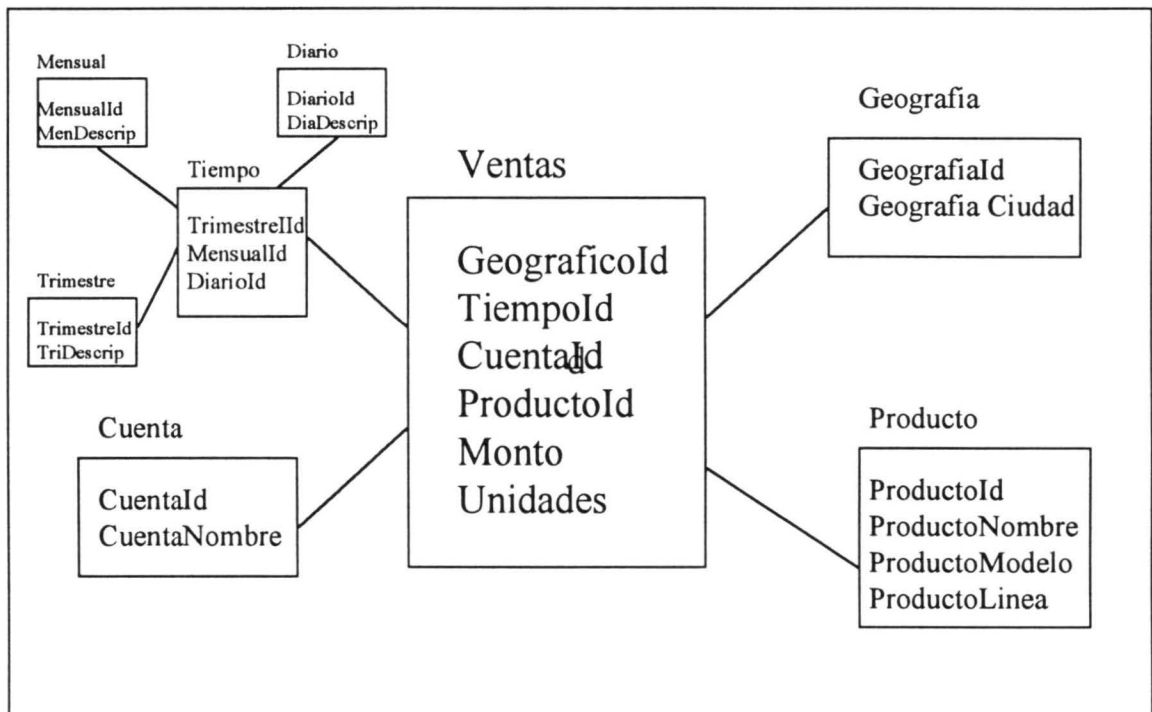


Figura 7. Ejemplo del esquema Snowflake.

### 5.2.8 La complejidad de construir y usar un Data Warehouse.

Es difícil construir un Data Warehouse debido a que es muy amplio el rango de técnicas que se requieren para formular, desarrollar, implementar, desplegar y explotar uno de ellos, por ejemplo, técnicas empresariales relacionadas con la tecnología y de administración del programa. Las técnicas empresariales requieren principalmente el comprender los datos que tendrá el Data Warehouse. Los cuales a su vez se van a relacionar con la determinación

de los requerimientos de la empresa y se traducirán en consultas que puedan ser satisfechas por el Data Warehouse. Así que los datos contenidos en el Data Warehouse se deben de relacionar con los procesos del negocio de la empresa. El valor de esta información es muy reconocido para las personas que saben cómo utilizarla a fin de formular estrategias que mejoren su negocio y proporcionen una ventaja competitiva.

En la mayoría de los casos, el Data Warehouse se basa en resúmenes de información que provienen de algún sistema de producción. Esto implica comprender la forma en que manejan y almacenan la información estos sistemas. Así como entender como se realizará la extracción de dichos datos de estos sistemas al Data Warehouse. Con esto podemos resumir que un Data Warehouse almacena información histórica importante de la empresa de una manera organizada. El implementar un Data Warehouse es una tarea compleja por la necesidad de interactuar con muchas tecnologías, distribuidores y usuarios finales para entregar resultados que sean eficaces en costo y tiempo.

Los procesos que envuelve la complejidad de construir un Data Warehouse que involucra el proporcionar datos desde un punto hasta otro (Data Warehouse) son los siguientes: [GILL, 94]

- Extracción de Datos. Los datos se deben obtener de las bases de datos operacionales. Generalmente, involucra un conjunto de bases de datos heterogéneas de donde los datos serán extraídos.
- Consolidación. Varios conjuntos de datos unidos a un conjunto de datos maestro. Esto involucrará estandarización de los campos o tipos de datos.
- Depurar. Los datos deben ser limpiados y removidos con el fin de evitar inconsistencias en los datos. Esto se dice que es el aspecto más retador para construir un Data Warehouse.
- Resumido. Los elementos de los datos resultantes se necesitan estar bien resumidos de tal manera que reduzcan el tiempo de búsqueda de la información necesaria.
- Actualización. Se necesita definir un tiempo de actualización de datos más esto no significa que se vayan a estar actualizando constantemente. Es importante permitir una actualización periódica o cuando se necesite.

### **5.2.9 La Minería de Datos.**

El término de minería de datos se refiere a la extracción no trivial de información implícita, la cual debe ser conocida previamente. Esto deriva a un

número de aproximaciones técnicas tales como sumarización de datos, reglas de clasificación, de aprendizaje, localización de redes dependientes, cambios analizados y anomalías detectadas.

Otra definición sería, la búsqueda de relaciones y patrones globales que existen en bases de datos grandes que se encuentran ocultas en la cantidad enorme de datos, como ejemplo, la relación que existe entre los datos de un paciente y los datos del diagnóstico de un médico.

Básicamente la minería de datos se dedica al análisis y el uso de técnicas de software para encontrar patrones y regularidades en el conjunto de datos. El análisis de la minería de datos tiende a trabajar con los datos y las mejores técnicas orientadas a manejar grandes volúmenes de información, haciendo posible el uso de la colección de datos como sea posible con el fin de tomar decisiones al respecto.

#### **5.2.10 Criterios de Evaluación de un Data Warehouse.**

Las investigaciones del potencial en cuanto a las opciones de Data Warehouses genera una serie de preguntas que a lo mejor en un principio se pueden contestar, pero analizando detenidamente podemos darnos cuenta que dichas preguntas son demasiado complejas. Por lo tanto es requerido un análisis a fondo de la tecnología y del medio ambiente.

Entre las principales preguntas se encuentran:

- ¿ Qué tan bien los sistemas manejadores de bases de datos (DBMS) ejecutan consultas simples o complejas ?
- ¿ Cuáles pueden ser predeterminados con incrementos en el volumen de datos ?
- ¿ Qué tan bien maneja combinaciones de consultas con el aumento en el volumen de usuarios ?
- ¿ Se puede empezar con hardware que es pequeño y no tan costoso y seguir creciendo ?
- ¿ Es posible tener mi plataforma abierta o encerrarme en una plataforma limitando mis opciones ?
- ¿ Qué tan grande puede llegar a crecer una base de datos y qué tan fácil es su manejo ?
- ¿ Qué tan fácil es monitorear las actividades y qué facilidades estarán disponibles para el manejo proactivo de las consultas ?

Para encontrar una guía la cual pueda resolver estas preguntas, y monitorear una extensa agenda de investigación que esta enfocada en la escalabilidad de los sistemas manejadores de bases de datos para carga de

trabajo con procesamiento de transacciones de una línea, se propone poner toda la atención en los sistemas que apoyan las decisiones (DSS) y los requerimientos del Data Warehouse. En la siguiente figura se muestran los criterios de escalabilidad en los sistemas DSS complejos y Data Warehouse.

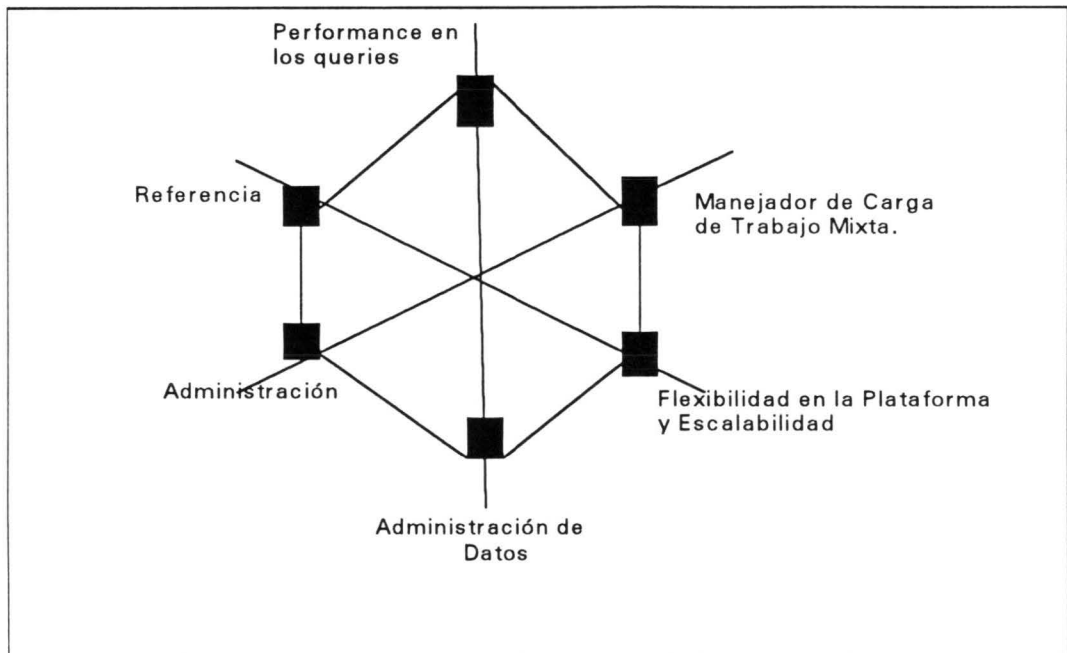


Figura 8. Resultados del Criterio de Escalabilidad.

Este criterio según el autor [GARTNER, 96] se seleccionó con el fin de reflejar las capacidades reales de los productos y de esa manera utilizarlos como base para las evaluaciones de los vendedores de productos DBMS para futuras investigaciones.

Los criterios de evaluación son hechos para proveer a las organizaciones con requerimientos esenciales de manera seleccionada para la implementación de sistemas DBMS para Data Warehouse. Estos criterios son los siguientes:

- "Performance" de las Consultas.
- Manejador de Carga de Trabajo Mixta.
- Flexibilidad de Plataforma y Escalabilidad.
- Administración de Datos.
- Administración.

- Referencias.

### **“Performance” de las Consultas.**

Aquí es donde nace la pregunta de que tan rápido va a ser el resultado de las consultas ? Es donde muchas organizaciones las cuales van a implementar Data Warehouse se realizan esta pregunta. Desafortunadamente, la mayoría de estas consultas son de gran complejidad, y sin embargo, presentan un performance impredecible. La gran mayoría de las consultas involucran la unión de múltiples tablas y requerirán de niveles de agregación o análisis comparativos de datos de periodos diferentes de tiempo.

Las consultas paralelas están creciendo en popularidad, donde los beneficios de estos queries son la obtención de información mediante el incremento a la velocidad y escalabilidad del performance de un query.

### **Administración de Carga de Trabajo Mixta.**

Mucha gente opina que los usuarios de un Data Warehouse se encuentran fuera de la realidad y las necesidades que esperan los usuarios son muy altas. A largo plazo los usuarios deben ser realistas sobre la concurrencia de consultas demasiadas complejas, que por naturaleza consumirán demasiados recursos y regresarán a ser como OLTP en forma “batch”.

El impacto de la utilización de consultas en ambiente multiusuario, incluirá requerimientos tanto simples como complejos, y dará como resultado una carga de trabajo estresante e impredecible. Este se encuentra contrastando en la carga de trabajo de un ambiente transaccional donde el mismo conjunto de transacciones se repiten durante el día, habilitando al sistema manejador de base de datos a que ejecuten “stored procedures” o estatutos de SQL estático.

A medida que la base de datos crece en diferentes dimensiones, el número de áreas y el volumen de datos agregados en un intervalo de tiempo, los mecanismos de carga se harán substancialmente más complejos. En algunas ocasiones causa que los procesos de carga se encuentren utilizados por los usuarios en horas pico. Por esta razón y el hecho de la implementación de un Data Warehouse se convierten los recursos rápidamente en requerimientos disponibles cada 24 horas, los sistemas manejadores de base de datos deben de tener un modelo concurrente que pueda soportar carga de datos hacia la base de datos y al mismo tiempo encuentren peticiones de los usuarios para ejecutar una consulta.

## **Flexibilidad en la Plataforma y Escalabilidad.**

Uno de los problemas principales en la construcción de la infraestructura de un Data Warehouse es el seleccionar un sistema manejador de base de datos que pueda brindar la flexibilidad necesaria de una plataforma de hardware y de esa manera permitir el crecimiento adecuado del potencial del mismo. Las empresas de hoy en día están buscando la posibilidad de iniciar una inversión la cual tenga un menor costo pero un nivel de eficiencia mayor y después crecer la capacidad de incrementar su Data Warehouse sin enfrentarse a barreras que no permitan el crecimiento.

No obstante, el rango de opciones a crecido considerablemente y las empresas buscan opciones en las cuales puedan realizar una combinación entre algún producto manejador de bases de datos y la arquitectura de hardware requerida.

## **Administración de Datos.**

En algunas empresas se han propuesto tener grandes cantidades de información en sus bases de datos. A menos que seleccionen un sistema manejador de base de datos relacional que mida las tareas de la administración de datos, los sistemas no funcionarán y no alcanzarán las metas originales. No solamente el observar la carga de datos y combinarla con datos ya existentes, sino también ver las restricciones de tiempo que existen. Por lo tanto, es necesario tener una organización física para encontrar cargas de trabajo y proveer diferentes niveles de disponibilidad.

Para lograr esto los sistemas manejadores de bases de datos relacionales de los Data Warehouse necesitan ver aspectos de administración de cantidades de datos, los cuales deben de ser introducidos después de observar las demandas de los usuarios reales, para así mantener una gran disponibilidad.

La administración del sistema manejador de base de datos relacional del Data Warehouse será difícil de controlar, es por eso que es necesario mantener el sistema de una manera equilibrada para esperar cargas de trabajo impredecibles; los datos se estarán constantemente agregando a la base de datos y el enfoque de las consultas que necesitan los usuarios estará en un constante cambio.

### **5.2.11 Data Mart vs. Data Warehouse.**

La tecnología de Data Mart está cambiando de tal manera que provee el crecimiento por debajo de las raíces ofreciendo un acceso fácil y rápido para

algún objeto de una área de información para un grupo pequeño de usuarios. Las Organizaciones se están dando cuenta que el implementar la tecnología de Data Warehousing es realmente complicada en su construcción y uso. Para implementar un Data Warehouse se necesitan una serie de requerimientos significativos como el software, hardware, tiempo, gente y costo.

### **Diferencias entre Data Warehouse y Data Mart.**

- Un Data Mart se enfoca solamente a una área o a un grupo de usuarios.
- Una empresa puede tener un Data Warehouse de toda la organización o si no un conjunto de Data Marts.
- Los Data Marts no contienen información almacenada de datos operacionales, a diferencia de los Data Warehouses.
- Los Data Marts contienen menos información que los Data Warehouses, así los Data Marts son más fáciles de navegar y entender que los Data Warehouses. Un Data Warehouse contienen mucha información necesaria para los usuarios.

### **Problemas para acceder un Data Mart.**

- El performance en el Data Mart no es muy bueno a medida que va creciendo el volumen de información.
- Los usuarios en ocasiones necesitan información de varios Data Marts.
- Las empresas no pueden administrar fácilmente los múltiples Data Marts existentes.
- Las organizaciones se encuentran con dificultades para construir un Data Mart.

### **La importancia del Performance.**

El performance del Data Mart involucra el balance de dos componentes: el tiempo de respuesta para el usuario final y el performance de cargar los datos. Un Data Mart esta diseñado para brindar una respuesta rápida al usuario en el acceso a tablas resumidas y valores agregados que se encuentran en el Data Mart. Desgraciadamente, la creación de tablas resumidas y valores agregados incrementa de manera significativa el tiempo para los procedimientos de carga de datos.

### 5.2.12 El Data Warehouse y su ventaja competitiva en los 90's.

Tomadores de decisiones de una empresa necesitan respuestas a preguntas que directamente afectan la habilidad de ser competitivo en los mercados cambiantes de nuestro tiempo. Estos tomadores de decisiones necesitan respuestas claras y significativas para cualquier pregunta por más compleja que sea sobre los datos de una manera rápida.

Los datos muchas veces se almacenan en una variedad de formatos tal es el caso de las bases de datos jerárquicas, archivos planos. Estos tipos de formatos se realizaron para ofrecer y automatizar transacciones de día a día surgiendo los sistemas de procesamiento de transacciones en línea (OLTP).

Las desventajas de un OLTP se manifiestan de muy distintas formas, por ejemplo, el analizar los datos se lleva mucho tiempo, se encuentra interferencia en el de tiempo de respuesta de la transacción e interrupción constante en las operaciones diarias. Además; para la gente que toma decisiones en este tipo de sistemas los datos aún no se encuentran tan claros que se dificulta la visualización de los hechos.

Al encontrarse tantas desventajas con los OLTP se ideó el concepto Data Warehouse. El Data Warehouse se especializa en poder brindar y analizar información de una manera rápida y sencilla.

A continuación se presenta una tabla comparativa entre las similitudes y diferencias entre OLTP y un Data Warehouse.

	OLTP	Data Warehouse
<b>Propósito</b>	Se corre operaciones día a día	Brinda información con el fin de analizarla.
<b>Estructura</b>	RDBMS	RDBMS
<b>Modelo de Datos</b>	Normalizado	Multidimensional
<b>Acceso</b>	SQL	SQL más las extensiones de análisis de datos.
<b>Tipo de Datos</b>	Datos los cuales operan el negocio.	Datos que analizan el negocio
<b>Condición de los Datos</b>	Cambiante e incompleta	Descriptiva e histórica

Tabla 6. Tabla de la Similitudes y Diferencias entre OLTP y un Data Warehouse.

Como ya se había mencionado anteriormente Data Warehousing es el proceso de extracción y transformación de datos operacionales a datos que

informan y se cargan a un almacén central de datos o Warehouse. Una vez cargados, será sencillo accederlos por un tomador de decisiones utilizando una serie de opciones como consultas y herramientas de análisis.

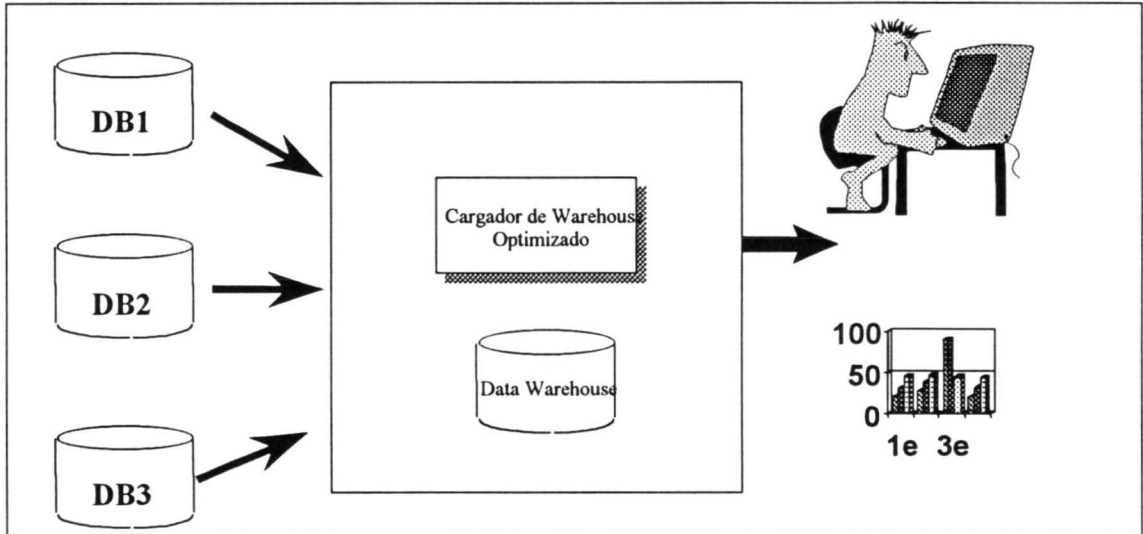


Figura 9. Modelo del Data Warehouse.

Es en los años 90's donde las empresas para tomar decisiones no buscan la información que se actualiza día con día, sino la información donde puedan visualizar los hechos para apoyo a la toma de decisiones de una empresa.

### 5.2.13 El Futuro.

Data Warehousing para el año 2000 será bastante diferente de los modelos que actualmente se han presentado por vendedores de hoy. Tanto OLTP como Data Warehouse coexistirán usando la misma base de datos. Los datos serán proporcionados a los usuarios de una manera fácil. Y cuando esto ocurra permitirá realizar un análisis momento a momento de sus transacciones. Las bases de datos se verán corriendo en terabytes. Algunos tipos de datos se encontrarán de una manera más rápida. Se observará que la siguiente generación de Data Warehouse se encontrará de todo tipo de información y no solamente del negocio como actualmente se ve.

### **5.3 Intranet.**

#### **5.3.1 Internet - Breve Historia.**

El término de "Internet" fue utilizado por primera vez en 1982 para referirse a la enorme colección de redes interconectadas que utilizan protocolos TCP/IP.

No obstante, a ganado reconocimiento en los últimos tres años. El "Internet" ha existido en varias presentaciones desde los años 60's.

Originalmente, el departamento de defensa de los Estados Unidos de Norteamérica introdujo la "Red de Administración de Proyectos de Investigación Avanzada" (ARPAnet), con la intención de experimentar en una área de red extensa que sobreviviría de una guerra nuclear.

Al paso de los años 70's y 80's, comienza a crecer una pequeña cantidad de redes pequeñas y conectadas al "Internet via getaways" con el objetivo del intercambio de correo electrónico. A mediados de los años 80's se vio un crecimiento significativo en el número de "hosts" para "Internet", y desde los últimos años de la década de los 80's, se ha presenciado con un crecimiento potencial en "Internet".

El crecimiento de "Internet provee a los usuarios con el significado de compartir y distribuir información. El "World Wide Web" (WWW) ha desarrollado como parte de "Internet", el permitir a la gente el crear fácilmente los "Web Sites" donde otros usuarios pueden llegar a acceder cualquier tipo de información que brinden estos. Las oportunidades que el WWW ofrece son en muchas maneras disparar la imaginación de la gente, así como, el abrir a corporaciones los ojos del potencial de las oportunidades comerciales.

#### **El "Internet" y su tecnología.**

Los componentes de la tecnología de "Internet " son:

- Protocolo de Comunicaciones. La habilidad de conectarse y comunicarse entre redes y dispositivos individuales.
- Transferencia de Archivos. La habilidad de transferir archivos entre un punto y otro punto diferente.
- Correo Electrónico. La habilidad de proveer comunicación directa de un punto a otro punto diferente entre individuos o grupos.
- "Web browsing" (Visulizador de red). La habilidad de proveer acceso a información de una a varias bases en demanda.
- Emulación de Terminal. La habilidad de acceder aplicaciones de infraestructura existentes.
- Interfases con el usuario.

### 5.3.2 Intranet- Breve Historia.

Como hace un año, se presentoóun área nueva la cual con tecnología de "Internet" puede ser aplicada para integrar a ambientes de tecnología de información ejecutiva. Y es ahora donde el término "Intranet" se ha convertido en un concepto muy utilizado y el cual se muestra en cualquier anuncio de hardware y software, en donde al principio la palabra "Intranet" se pensó que se había escrito mal la palabra "Internet".

Pero detrás de toda esta mercadotecnia existe un nuevo conjunto de aspectos para un administrador de red, particularmente la creciente preocupación de que la infraestructura de la red pueda soportar aplicaciones de "Intranet".

### 5.3.3 ¿ Qué es "Intranet" ?

Mientras que todo el mundo habla de los WWW y nos encontramos de una manera constante bombardeados con "Webs" corporativos en cualquier tipo de medio, lo que realmente está sucediendo es que las compañías están trabajando en la utilización de la tecnología de WWW sobre sus redes internas. De hecho, en esta evolución de tecnología ha surgido una nueva palabra llamada "Intranet".

Un "Intranet" es básicamente una red de computadoras las cuales pueden compartir datos entre, utilizando protocolos estandarizados de WWW de manera interna. Mediante la utilización de "Webs" internos para el almacenamiento de su información, así como, el proveer a cada computadora en la red con un "browser" o viasulizador. Esto permite la distribución de una manera fácil y no costosa de información ejecutiva interna. En donde lo anterior es un ingrediente esencial para el mundo competitivo actual. [RWAY, 97]

Mientras que las aplicaciones de "Intranet" no se encuentran limitadas, por lo general caen a una de estas tres categorías:

1. Aplicaciones Públicas, que son en esencia de una a muchas comunicaciones. Por ejemplo: grupos, departamentos, u organizaciones enteras pueden tener páginas donde envían información.
2. Aplicaciones Transaccionales, las cuales tienen dos vías de interacción, así como el bajar información y verificar los beneficios de la información.
3. Aplicaciones de Comunidad, que son las interacciones de muchas a muchas. Incluyen grupos de noticias que facilitan el intercambio de información a los miembros del grupo, haciendo posible la disposición de información a otros fuera de grupo.

En términos más simples un "Intranet" es crear aplicaciones con tecnología de "Internet" para uso interno exclusivamente.

El impacto que un "Intranet" se observa en la operación diaria de una organización tanto en su desarrollo como en su misma cultura. Para comprender de manera completa el concepto de "Intranet" necesitamos observar varias áreas:

- Las demandas actuales del negocio.
- La tecnología de "Internet".
- El "Internet" vs. "Intranet".
- La revolución del "Intranet".

Antes de definir cada área, se examinará un ejemplo del impacto del "Intranet". Imaginemos un escenario donde una compañía tiene 20 "sites" y 1000 personas la cual normalmente necesitan tener acceso frecuentemente a las noticias de la compañía, cambios en la políticas de la organización, procedimientos de recursos humanos aún más simple la necesidad de tener directorio telefónico de los empleados, las especificaciones de los productos e información de precios.

Normalmente, se utiliza demasiado material impreso el cual contribuye a mucho consumo de tiempo e implica un gasto en recursos. Ahora bien, se tiene también el problema de garantizar a la gente la llegada a tiempo de la información necesaria. Aunado a este problema, el cambio constante de especificaciones de la empresa.

La siguiente página nos muestra los problemas actuales y los beneficios al negocio que pueden encontrarse debido a un "Intranet" . (verTabla 7)

Tipo de Problema	Beneficios en el Negocio
Dificultad de almacenar o acceder a archivos físicos.	<ul style="list-style-type: none"> <li>• Bajos costos operacionales.</li> <li>• Fácil distribución.</li> <li>• Mayor accesibilidad en la demanda.</li> </ul>
Problemas de información en plataforma cruzada.	<ul style="list-style-type: none"> <li>• Estándares simples.</li> <li>• Bajas conversiones en costos.</li> </ul>
Problemas de comunicación funcional.	<ul style="list-style-type: none"> <li>• Comunicaciones de correo electrónico estándar.</li> <li>• Facilita la distribución de la información.</li> <li>• Mayor apoyo en la toma de decisiones.</li> </ul>
Falta de estándares para la comunicación.	<ul style="list-style-type: none"> <li>• Demanda de acceso de información.</li> <li>• Consulta en tiempo real.</li> <li>• Capacitación.</li> </ul>
Se toma demasiado tiempo en la distribución de la información.	<ul style="list-style-type: none"> <li>• Bajo costo en la distribución.</li> <li>• Más satisfacción en los usuarios.</li> </ul>

Tabla 7. Los problemas actuales y los beneficios al negocio que pueden encontrarse debido a un "Intranet".

La solución a este problema requiere tecnología la cual:

- Pueda entregar la información que se demande a tiempo.
- Pueda garantizar que la información es la más nueva y correcta disponible.
- Asegure que la información puede mantenerse en una sola fuente.
- Permita a la información sea mantenida por la gente la cual es normalmente mantenida y preparada la información original.

#### 5.3.4 Ventajas de un "Intranet".

El término "Intranet" y las maravillas que se pueden hacer con una red que tenga toda la información que usted requiere para tomar decisiones, manejar operaciones de la empresa, cerrar ventas, comprar materia prima, ponerse al tanto de la industria y capturar la atención de sus clientes potenciales de una manera dirigida, son algunas de las ventajas de contar con este tipo de herramienta.

Dentro de las actuales formas de realizar negocios globales, el factor competitivo ha cobrado gran importancia. Actualmente la gente de las empresas requieren soluciones que contribuyan a que la operación y los procesos de la organización se vuelvan efectivos para que los grupos de trabajos se mantengan en unión constante, en colaboración para gestar con ello un nuevo

sistema de colaboración que además se mantenga en constante actualización en cuanto a innovación tecnológica con el fin de integrar estas herramientas a sus procesos productivos siempre que sea necesario.

De tal manera que resulta muy importante que los empleados cuenten con el poder y la habilidad de contribuir entre ellos, basándose en la inteligencia y experiencia que resulte de utilizar adecuadamente los recursos, para que puedan así distribuir estos conocimientos en toda la organización.

Las ventajas que un "Intranet puede aportar dentro de esta nueva dinámica empresarial van en función de una nueva concepción de comunicación Intraempresarial, ya que brinda la posibilidad de distribuir información de manera inmediata, intuitivamente fácil y sumamente efectiva.

Además, un "Intranet" puede ser utilizado como apoyo para una amplia gama de soluciones, entre las que se destaca la capacidad de desarrollar aplicaciones para realizar negocios interactivos entre dos o más partes sin importar en donde se encuentren, mediante la combinación de los recursos de la empresa y los de "Internet". El "Intranet" tiene la característica de trabajar mediante el uso estructurado de la tecnología de "Internet".

#### **5.3.5 Parámetros de medición de efectividad y los beneficios de un "Intranet".**

Existen dos parámetros para medir la efectividad y los beneficios de un "Intranet":

1. Posibilidad de uso. Un "Intranet" permite al usuario corporativo la creación, organización y publicación de documentos e información de interés para la empresa en el lenguaje HTML (Hyper Text Markup Language) que permite el manejo de aplicaciones multimedia e interactividad en línea entre los usuarios de manera muy simple. Este tipo de documentos se pueden organizar en un índice secuencial de publicación que puede ser clasificada por departamento o por oficina desde el escritorio del usuario.

El resultado de esto es un ambiente de acceso sencillo a la información y desde cualquier punto de la compañía, sin dejar de tomar en cuenta los niveles de seguridad que se requieren para el manejo de información confidencial o delicada, la cual no podrá ser accesada por todos los usuarios.

Este control de acceso así como la seguridad permiten que el correo electrónico y los grupos de discusión sean completamente privados, y que todas las partes sean autenticadas dentro de la red. Además, los contenidos del mismo correo y los grupos de discusión pueden ser distribuidos a través de sistemas o incluso llevados fuera de la línea para uso "desconectado"

La implementación de un "Intranet" tiene la gran ventaja de convertir el impresionante mundo de información contenida en "Internet" y en los archivos de la empresa, en una impresionante herramienta para que los usuarios extraigan solo lo que necesitan y encuentren lo que están buscando sin necesidad de más tramites que teclear un tema o nombre para que la red se encargue de desplegar un listado de todos los sitios de información que existan relacionados con el tema en todos los servidores de la empresa e incluso de "Internet".

2. Servicios de la red. Un "Intranet" contempla un servicio de directorio, el cual captura y maneja información acerca de:

- Empleados y Clientes.
- Control de acceso.
- Configuración del servidor.
- Sitios de aplicación específica.

El servicio completo de "Intranet" ofrece un amigable manejo de interfaz basado en el lenguaje HTML que permite que todos los sitios y servidores sean manejados de manera segura desde cualquier punto de la red.

### **Crecimiento de los "Intranets".**

Las "Intranets" están creciendo porque ofrecen lo siguiente:

- Costo/Eficiencia.
- Facilidad de actualización.
- Utilidad para liberar información.
- Utilidad para manejar información disponible en demanda.
- Seguridad.
- Facilidad relativa de configuración, uso y coordinación.
- Aptas para correr en toda la compañía, así como por departamento o por área funcional.
- Capacidad de crear aplicaciones en multimedia.
- Reducción de los costos de impresión y tiempo de distribución.
- Capacidad de correr a través de cualquier plataforma y para integrar a todos los clientes.
- Un puente hacia "Internet".
- Aptas para integrarse con bases de datos internas de corporativos.
- Utilidad para desarrollar contenido por medio de cualquier usuario, en lugar de que lo realicen técnicos profesionales.
- Accesibilidad al usuario al estar basada en la tecnología de "Internet".

### 5.3.6 “Intranet” vs. “Internet”.

Al hablar de “Internet” nos estamos refiriendo a eso que llaman la red de redes, que es un enlace multinacional y anárquico en el que es posible incluir a millones de redes de computadoras sin importar a qué distancia se encuentren. “Intranet”, al igual que “Internet”, es un concepto global de muchas propiedades y no un objeto tangible.

Necesita y utiliza una gran variedad de productos de hardware y software, aún cuando no existan paquetes realmente imprescindibles. Es además un especie de “Web” de uso privado para una empresa que puede utilizar muchas de las herramientas creadas para “Internet” sin ninguna alteración, y que están mantenidas sobre redes internas para que la información manejada sea vista tan solo por los empleados de dicha empresa.

### 5.3.7 Las Tendencias.

El desarrollo e incremento de soluciones como los “Intranets” son según los expertos, solo el comienzo para satisfacer la necesidad de integrar protocolos y plataformas, para que los usuarios puedan intercambiar información bajo cualquier formato y puedan realizar transacciones de manera transparente y segura a través de aplicaciones convencionales tipo “CGI” (Common Gateway Interface).

De acuerdo a la visión de muchos expertos, se están comenzando a utilizar tecnologías tipo “Web” para difundir información electrónica con aditamentos y extensiones para poder intercambiar datos de manera más eficiente. Sin embargo este tipo de soluciones de alguna forma se contraponen con la visión que tienen algunas empresas respecto a manejar una red como si fuera una computadora personal “PC”.

Pero lo cierto, es que soluciones como los “Intranets” van a favorecer lo que últimamente se ha discutido sobre las máquinas que no necesiten ni discos duros ni “drives” para “diskettes”.

Todo este panorama permite ver que a los “intranets” tienden a erigirse como toda una transición hacia un nuevo y revolucionario vehículo de transmisión de información, en el que los antiguos procesos comunicativos intraempresariales se sustituyen por sus equivalentes en lenguajes operativos de informático.

## VI. RESULTADOS DEL ANÁLISIS

### 6.1 Análisis Orientado a Objetos.

En este punto se presentan los resultados del análisis según la metodología propuesta de Análisis Orientado a Objetos de Martin/Odell.[MARTIN, 94]

#### 6.1.1 Identificación de Objetos.

1. Clientes: Es toda persona que realiza la compra de un servicio de hotel, reservación de vuelos y proveedor de servicios.
2. Hoteles : Este objeto representa el servicio de proporcionar una habitación en cualquier ciudad.
3. Proveedores de servicios: Representa las compañías de renta de transporte, empresas proveedoras para algún evento en especial.
4. Aerolíneas: Este objeto proporciona a los clientes transportación aérea de un destino.
5. Agente de Ventas: Es toda persona que realiza la venta de un servicio de hotel, reservación de vuelos y proveedor de servicios.
6. Concentración de Ventas: Las ventas realizadas de los servicios de reservación de vuelos, hotel y proveedor de servicios.

## 6.2 Análisis de la Estructura de Objetos.

El análisis de la estructura de objetos define las categorías de los objetos que percibimos y las formas en que los asociamos.

Agente de Ventas:

- Número de agente de ventas
- Nombre de agente de ventas
- Dirección de agente de ventas
- Teléfono de agente de ventas

Clientes:

- Número del Cliente
- Nombre del Cliente
- Dirección del Cliente
- Teléfono del Cliente
- Sexo del Cliente
- Nacionalidad del Cliente
- Datos del pasaporte del Cliente
- Edad

Aerolínea:

- Número de Aerolínea
- Descripción de Aerolínea
- Dirección de Aerolínea
- Teléfono de Aerolínea
- Clase de vuelo
- Número de vuelo
- Destino

Hotel:

- Número de Hotel
- Descripción del Hotel
- Categoría del Hotel
- Ubicación del Hotel
- Teléfono del Hotel
- Categoría del Hotel

Proveedores de servicios:

- Número de Proveedor
- Nombre del Proveedor
- Dirección del Proveedor
- Teléfono del Proveedor

**Concentración de Ventas:**

- Fecha
- Número del Cliente
- Número de Agente de Ventas
- Número de Forma de Pago
- Número de Hotel
- Número de Proveedor de Servicios
- Número de Aerolínea
- Monto

**6.3 Análisis del Comportamiento del Objeto.**

En el análisis del comportamiento de objetos (ACO) se realizan la identificación de los métodos que cada objeto va realizar. [MARTIN, 94]

**Agente de Ventas:**

- Consultar agentes de ventas
- Consultar la información de agente de ventas

**Clientes:**

- Consultar clientes
- Consultar la información de clientes

**Hotel:**

- Consultar hoteles
- Consultar la información de hoteles

**Proveedores de servicios:**

- Consultar proveedores de servicios
- Consultar la información de proveedores de servicios

**Aerolíneas:**

- Consultar las aerolíneas
- Consultar información de las aerolíneas

**Concentración de Ventas:**

- Crear tabulaciones con las ventas
- Condensar las ventas a detalle
- Graficar las ventas

## VII. RESULTADOS DEL DISEÑO

### 7.1 Diseño.

El diseño que se utilizó fue siguiendo la metodología de Martín/Odell [MARTIN, 94] y en el diseño de la base de datos (Data Warehouse) se utilizaron los esquemas Star y Snowflake [INFORMIX, 96].

#### 7.1.1 Diseño de la Estructura y del Comportamiento de un Objeto.

En el diseño de la estructura y comportamiento de objetos se identifica lo siguiente: ¿ Qué clases se implementarán ?, ¿ Qué estructura de datos utilizará cada clase ?, ¿ Qué operaciones ofrecerá cada clase y cuáles serán sus métodos en determinado momento ?

##### Identificación de Clases

Agente de Ventas: se consultará (operación)

- Número de agente de ventas
- Nombre de agente de ventas
- Dirección de agente de ventas

- Teléfono de agente de ventas

Clientes: se consultará (operación)

- Número del Cliente
- Nombre del Cliente
- Dirección del Cliente
- Teléfono del Cliente
- Sexo del Cliente
- Nacionalidad del Cliente
- Datos del pasaporte del Cliente
- Edad

Aerolínea: se consultará (operación)

- Número de Aerolínea
- Descripción de Aerolínea
- Dirección de Aerolínea
- Teléfono de Aerolínea
- Clase de vuelo
- Número de vuelo
- Destino

Hotel: se consultará (operación)

- Número de Hotel
- Descripción del Hotel
- Categoría del Hotel
- Ubicación del Hotel
- Teléfono del Hotel

Proveedores de servicios: se consultará (operación)

- Número de Proveedor
- Nombre del Proveedor
- Dirección del Proveedor
- Teléfono del Proveedor

Concentración de Ventas: se creará, condensará y graficará (operaciones)

- Fecha
- Número del Cliente
- Número de Agente de Ventas
- Número de Forma de Pago
- Número de Hotel
- Número de Proveedor de Servicios
- Número de Aerolínea
- Monto

A continuación se presentan los diagramas de clases del diseño los cuales representan cada objeto unido por sus operaciones definidas.

### Agente de Ventas

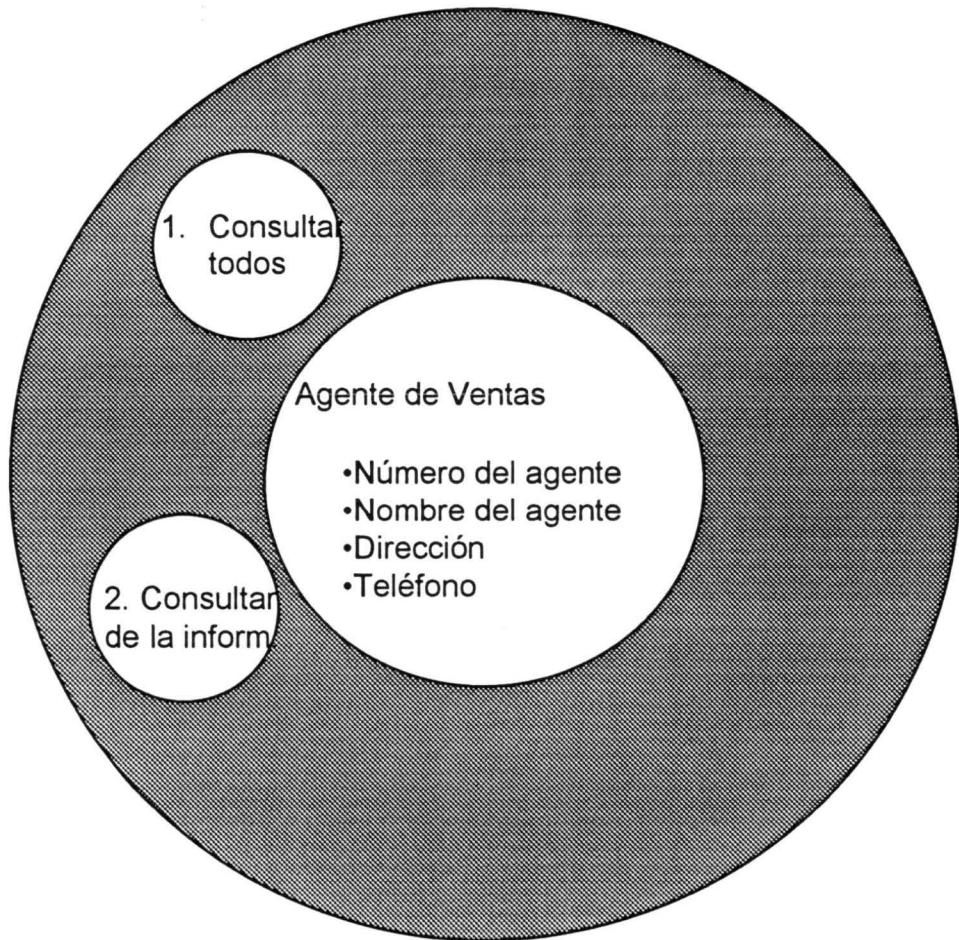


Figura 10. Diagrama de Clase de Agente de Ventas.

Cliente

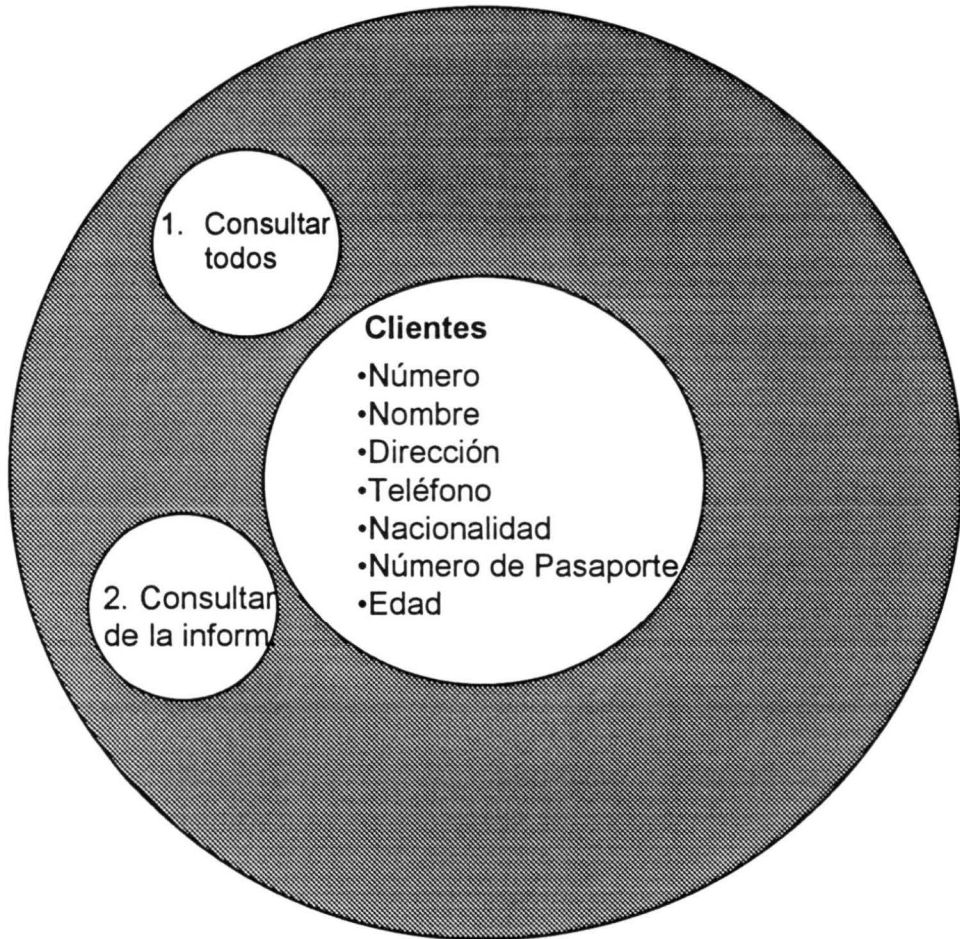


Figura 11. Diagrama de Clase de Clientes.

## Aerolínea

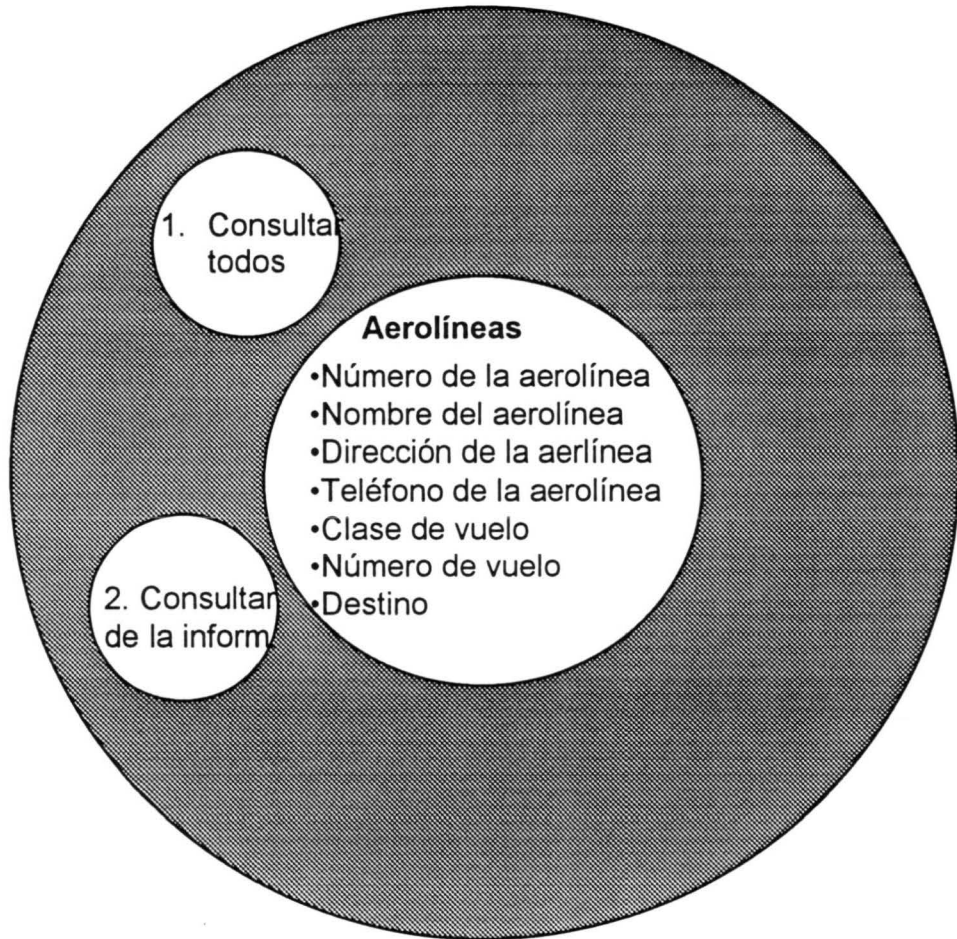


Figura 12. Diagrama de Clase de Aerolínea.

## Proveedores de Servicios

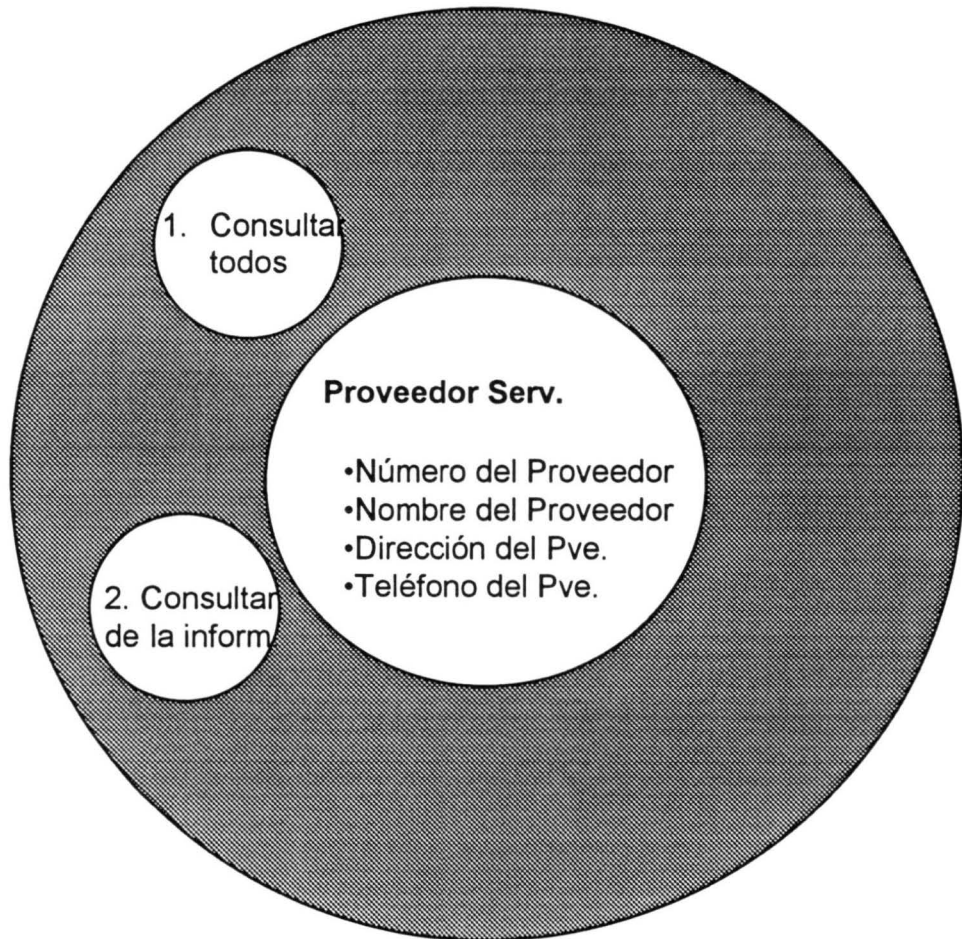


Figura 13. Diagrama de Clase de Proveedor de Servicios.

Hotel

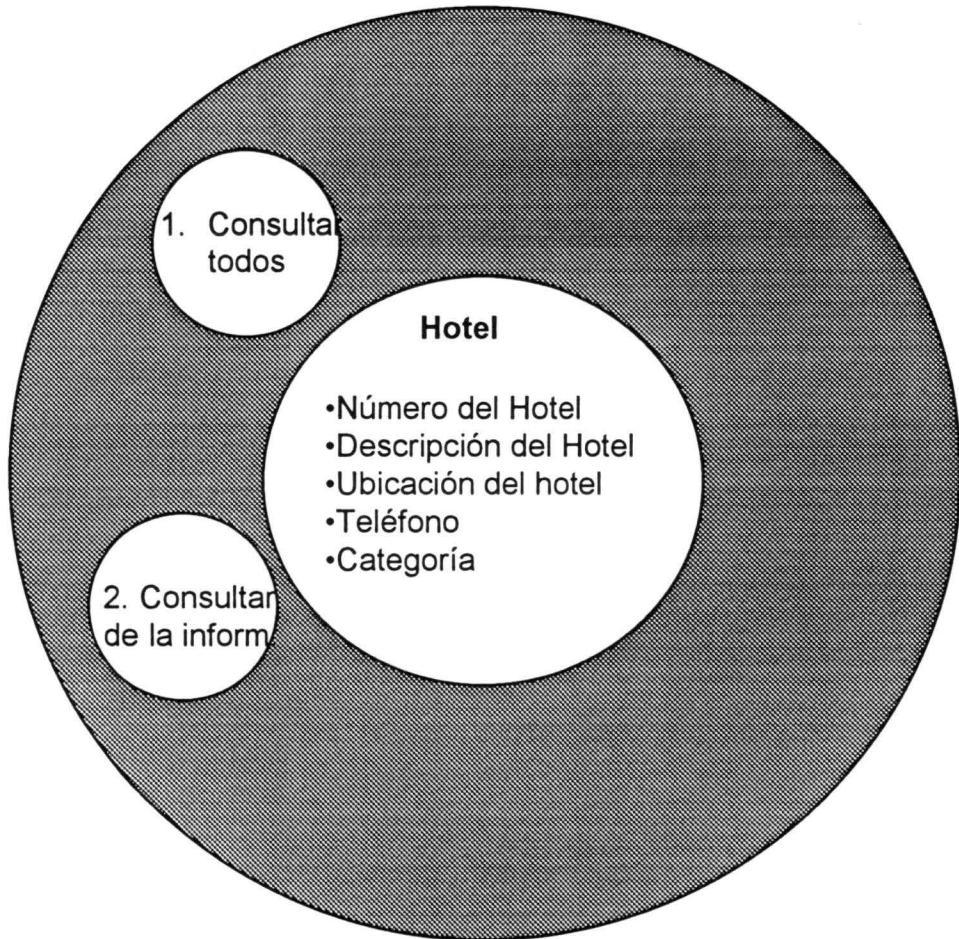


Figura 14. Diagrama de Clase de Hotel.

## Concentración de Ventas

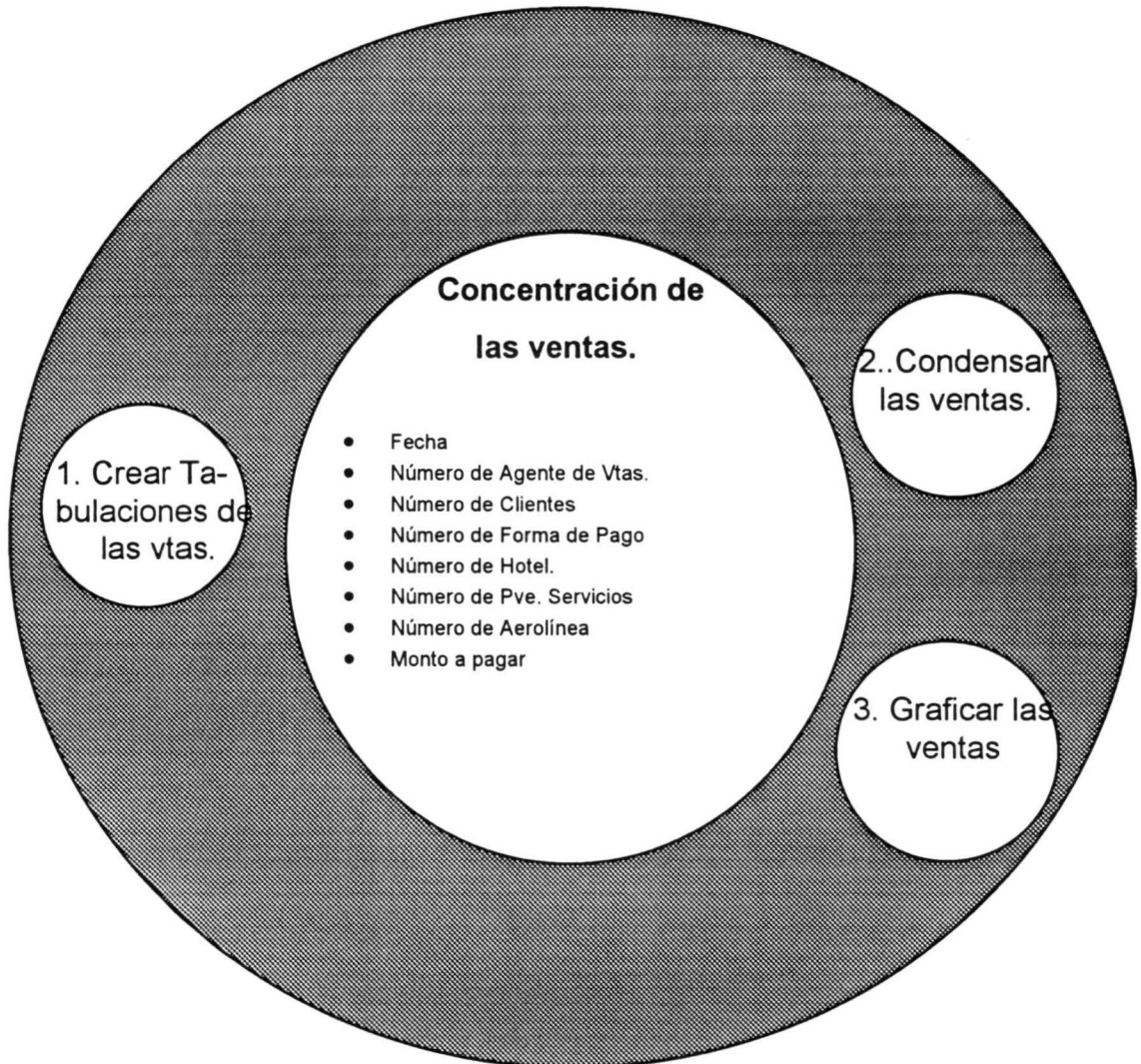


Figura 15. Diagrama de Clase de Concentración de Ventas.

### **7.1.2 Diseño de la Base de Datos (Data Warehouse).**

Para diseño y la estructura de la base de datos el cual se siguió el esquema de estrella y "SnowFlake" parcial, el cual se encuentra explicado en el Apendice A.

## VIII. RESULTADOS DEL SDD

### **8.1 Documento Visión.**

#### **8.1.1 Antecedentes.**

ABA|TURISMO es una agencia de viajes la cual ofrece una diversidad de productos entre los que se encuentran: las reservaciones en vuelos, habitaciones de hotel, servicios varios, paquetes para eventos especiales y paquetes turísticos.

Con el fin de apoyar a la alta gerencia de ABA|TURISMO en la toma de decisiones es necesario obtener la información que se requiere de una manera oportuna y confiable de tal manera que no tenga que esperar mucho tiempo la alta dirección para recibir información financiera, tendencias de productos y noticias internas de la empresa.

#### **8.1.2 Enunciado de visión.**

Crear dos herramientas computacionales basadas en la información que brinda el sistema de ventas por menudeo (Sistema de apoyo a agentes de

ventas). La primera herramienta es el Sistema de Información Ejecutiva de Ventas la cual proporcionará a la alta gerencia de ABA|TURISMO información financiera, tendencias de productos y noticias internas de manera oportuna y confiable de su operación diaria que ayude a la toma de decisiones. La segunda herramienta es un recolector de información para apoyar al Sistema de Información Ejecutiva de Ventas que tendrá como objetivo agrupar, ordenar y condensar la información necesaria (clientes, proveedores, ventas, hoteles y aerolíneas) de una base de datos ya existente para posteriormente vaciar dicha información en el Data Warehouse. El Sistema de Información Ejecutiva de Ventas será entregado en un tiempo estimado de cuatro meses.

### **8.1.3 Perfil del Usuario.**

El sistema será utilizado principalmente por la alta gerencia de la empresa ABA|TURISMO la cual está conformada por cuatro personas (un administrador y tres supervisores).

- **Administrador:** no pertenece al área de sistemas, sin embargo tiene un conocimiento de computación mínimo, ya que utilizan aplicaciones como Word, Excel, punto de venta, entre otras. El Administrador será el que tome las decisiones según lo requerido por el sistema.
- **Supervisor:** no pertenece al área de sistemas, sin embargo tiene un conocimiento de computación, ya que utiliza aplicaciones como Word, Excel, punto de venta, etc.

### **8.1.4 Necesidades actuales.**

A continuación se presentan los siguientes problemas o necesidades:

1. La consulta de las operaciones de ABA|TURISMO suele ser lenta debido al gran volumen de información. Esta consulta proviene de dos fuentes: de la información ejecutiva histórica y de la base de datos del Sistema de Ventas por Menudeo, los cuales se encuentran en ABA|TURISMO.
2. La representación de algunos de sus reportes de operaciones no tienen un formato adecuado. Algunas veces se representa la información de ventas con una estructura o formato inadecuado obstaculizando la visualización y comprensión de la información. Ejemplos de formatos adecuados serían, la representación de la información a través de gráficas o tablas.
3. La información que la alta gerencia necesita para definir una estrategia de ventas en algunas ocasiones no se recibe a tiempo. Se pierde tiempo debido a que la información es extraída del sistema de ventas por menudeo y luego

pasa por varias personas para su ordenamiento, agrupación y condensación antes de llegar ésta a la alta dirección.

Para resolver el problema de las consultas de las operaciones, se brindará un Sistema de Condensación de Información bajo una arquitectura cliente/servidor. Este sistema realizará una serie de procesos internos de cálculos, selección, ordenamiento, agrupación, condensación y almacenamiento de datos, con el fin de recabar solamente la información necesaria para el usuario. Dicha información de clientes, proveedores de servicios, hoteles, ventas y aerolíneas se encuentra guardada en la base de datos del Sistema de Ventas por Menudeo, en donde la captura la realizan los agentes de ventas. Estos datos se almacenarán en un Data Warehouse que será diseñado y creado, el cual residirá en el servidor. Para después utilizar estos datos en el Sistema de Información Ejecutiva de Ventas. De esta forma se podrá consultar la información de una manera rápida, brindando integridad y seguridad en la información.

Para resolver los problemas de los reportes en un formato adecuado y que la información se le proporcione a tiempo a la alta gerencia, se creará mediante la tecnología que nos ofrece "Internet" un sistema de información ejecutivo, el cual tendrá como objetivo proporcionar un acceso fácil y rápido a la información que se encuentra en el Data Warehouse, el cual apoyará a la alta gerencia a planear, monitorear y analizar el medio ambiente de la empresa. Entre las características del SIE se encontrarán: la utilización de dispositivos de entrada como el mouse, poseerá la capacidad de navegación con acceso a información financiera, tendencia de productos y noticias internas de manera rápida y fácil. Permitirá desplegar la información de ventas en diferentes formatos (tabulares o gráficos).

#### 8.1.5 Participantes.

<b>Area de Participación</b>	<b>Nombre</b>
Administrador del Producto	Christophe Bastiere
Administrador del Programa	Rolando Torres
Desarrollador	Adriana Olmeda
Control de Calidad	Rolando Torres y Christophe Bastiere
Educación al Usuario	Adriana Olmeda
Planeación y Logística	Rolando Torres

Tabla 8. Lista de Participantes.

### 8.1.6 Arquitectura Conceptual.

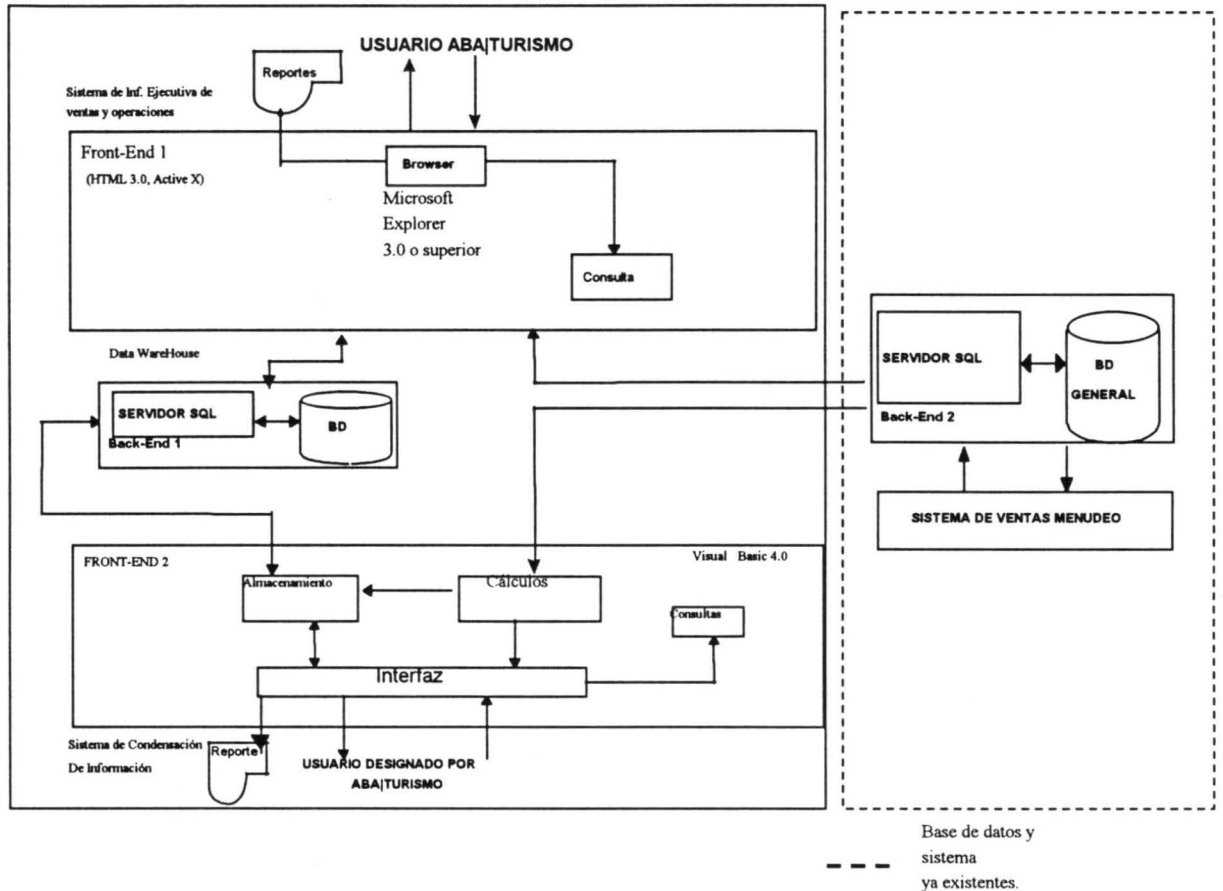


Figura 16. Arquitectura de software propuesta.

A continuación se presenta una breve explicación de la gráfica anterior.

- El "browser" o navegador desplegará de manera escrita o gráfica la información tal como: las ventas por cliente, ventas por forma de pago, las ventas por proveedor, ventas por líneas aéreas, ventas por hoteles, ventas por clase de vuelo, ventas por destino, ventas por agencia de viajes y ventas por vendedor. Para que el ejecutivo defina posteriormente una estrategia de ventas.
- El Data Warehouse (base de datos) contendrá información procesada por el sistema de condensación de información.
- El sistema de condensación de información utilizará la base de datos del Sistema de Ventas al Menudeo (esta base de datos ya está creada y contiene información de clientes, proveedores, hoteles y aerolíneas) para

posteriormente vaciar la información necesaria al Data Warehouse. Este sistema se realizará bajo una arquitectura cliente/servidor.

- El sistema de condensación de información lo usará un usuario designado por la empresa con el fin de utilizar el sistema cada vez que se requiera actualizar la información.
- La “consulta” del Sistema de Información Ejecutiva de Ventas y Operaciones se harán del Data Warehouse (el cual será diseñado y creado) y en algunas ocasiones de la base de datos general del Sistema de Ventas por Menudeo.

La información requerida por el Sistema de Condensación de Información se obtendrá de la base de datos del Sistema de Ventas por Menudeo.

### **8.1.7 Infraestructura del proyecto.**

ABA|SIS cuenta con un servidor Pentium/166MHZ con 32 MB en RAM , 2 GB en disco duro, unidad de DAT y de CD ROM. Cuenta también con 2 computadoras personales “PC” 486/66MHZ con 24 MB en RAM, 850 MB en disco duro y monitor SVGA. En cuanto a software se refiere tienen en sus PC el sistema operativo Windows 95, Visual Basic 4.0, Excel, Word y Explorer 3.0. En el servidor cuenta con el sistema operativo Windows NT 3.51 y SQL Server 6.5, entre otros.

### **8.1.8 Identificación de Riesgos.**

- Una de las áreas de riesgos del proyecto es que se logre desarrollar a tiempo el sistema de condensación de información para el Data Warehouse, de lo contrario se presentará un retraso para el inicio del sistema de información ejecutiva en cuanto a programación y pruebas se refiere.
- El usuario no dedique el tiempo necesario al proyecto.
- No contar con el equipo adecuado para el desarrollo del proyecto.

## **8.2 Documento de Especificaciones Funcionales.**

### **8.2.1 Antecedentes.**

ABA|TURISMO es una agencia de viajes la cual ofrece una diversidad de productos entre los que se encuentran: las reservaciones de vuelos, habitaciones de hotel, servicios varios, paquetes para eventos especiales y paquetes turísticos.

Con el fin de apoyar a la alta gerencia de ABA|TURISMO en la toma de decisiones es necesario obtener la información que se requiere de una manera

oportuna y confiable de tal manera que no tenga que esperar mucho tiempo la alta dirección para recibir información financiera, tendencias de productos y noticias internas de la empresa.

### **8.2.2 Objetivo.**

Crear dos herramientas computacionales basadas en la información que brinda el sistema de ventas por menudeo (Sistema de apoyo a agentes de ventas). La primera herramienta es el Sistema de Información Ejecutiva de Ventas el cual proporcionará a la alta gerencia de ABA|TURISMO información financiera, tendencias de productos y noticias internas de manera oportuna y confiable de su operación diaria que ayude a la toma de decisiones. La segunda herramienta es un recolector de información para apoyar al Sistema Información Ejecutivo de Ventas que tendrá como objetivo agrupar, ordenar y condensar la información necesaria (clientes, proveedores, ventas, hoteles y aerolíneas) de una base de datos ya existente para posteriormente almacenar dicha información en el Data Warehouse. El Sistema de Información Ejecutiva de Ventas será entregado en un tiempo estimado de cuatro meses.

### **8.2.3 Estructura del proyecto.**

#### **Desarrollo.**

1. Se desarrollará un Sistema de Condensación de Información bajo una arquitectura cliente/servidor. Este sistema realizará una serie de procesos internos de cálculos, selección y almacenamiento de datos, con el fin de recabar solamente la información (clientes, proveedores de servicios, hoteles y aerolíneas) necesaria para el usuario.
2. Dicha información de clientes, proveedores de servicios, hoteles y aerolíneas se encuentra guardada en la base de datos del Sistema de Ventas por Menudeo, en donde la captura la realizan los agentes de ventas. Estos datos se almacenarán en un Data Warehouse que será diseñado y creado, el cual residirá en el servidor. Para después utilizar estos datos en el Intranet. De esta forma se podrá consultar la información de una manera rápida y brindando integridad y seguridad en la información.
3. Se creará a través de la tecnología que nos ofrece "Internet" un sistema de información ejecutivo, el cual tendrá como objetivo proporcionar un acceso fácil y rápido a la información que se encuentra en el Data Warehouse, el cual apoyará a la alta gerencia a planear, monitorear y analizar el medio ambiente interno de la empresa. Entre sus características se encuentran: la utilización de dispositivos de entrada alternos como el mouse, poseerá la

capacidad de navegación para acceder información financiera, tendencias de productos y noticias internas de manera rápida y fácil. Permitirá desplegar la información de ventas por cliente en un intervalo de tiempo, ventas por forma de pago, las ventas por proveedor, ventas por líneas aéreas, ventas por hoteles, ventas por clase de vuelo, ventas por destino, ventas por paquetes, ventas por agencia de viajes y ventas por vendedor en diferentes formatos (escrito o gráfico).

4. Se realizarán pruebas integrales del sistema.

#### **Roles.**

<b>Rol</b>	<b>Responsabilidad</b>	<b>Misión</b>
Administrador del Producto	Christophe Bastiere	Realizar la promoción del Sistema de Info. Ejecutiva de Vtas. Que se esta desarrollando.
Administrador del Proyecto	Rolando Torres	Dar seguimiento del proceso de construcción del Sistema de Info. Ejecutiva de Vtas. y el sistema de Condensación de Info.
Administrador del Desarrollo	Adriana Olmeda	La construcción del Sistema de Inf. Ejecutiva de Vtas. y el Sistema de Condensación de Info.
Calidad y Logística	Rolando Torres	Realizar pruebas integrales de ambos sistemas. Ayudar con pruebas contundentes para que los sistemas se encuentren con cero defectos.
Capacitación al Usuario	Adriana Olmeda	Realizar la Documentación necesaria del producto que se está construyendo.
Liberación del Producto	Rolando Torres	Preparar la liberación de los sistemas y mantenerlos después de liberar los sistemas.

Tabla 9. Lista de Roles.

### 8.2.4 Agenda del proyecto.

#### **Agenda.**

Las siguientes son las metas mayores.

<b>Nombre de la Tarea</b>	<b>Fecha de Terminación</b>
Documento de visión 1.0	7/02/97
Firma del documento de visión	8/02/97
Primer Beta	02/06/97
Especificaciones funcionales 1.0	10/03/97
Firma de especificaciones funcionales	11/03/97
Instalación de Hardware	28/05/97
Instalación al usuario	29/05/97

Tabla 10. Agenda.

### 8.2.5 Descripción Funcional.

#### **Ambiente.**

ABA|TURISMO cuenta con un servidor Pentium/166MHZ con 32 MB en RAM, 2 GB en disco duro, unidad de DAT y de CD ROM. Cuenta también con 2 computadoras personales "PC" 486/66MHZ con 24 MB en RAM, 850 MB en disco duro y monitor SVGA. En cuanto a software se refiere tienen en sus PC el sistema operativo Windows 95, Excel, y Word. En el servidor cuenta con el sistema operativo Windows NT 3.51 y SQL Server 6.5, entre otros.

#### **Servicios de usuario.**

- La interfaz debe seguir los estándares establecidos por ABACO.
- El sistema debe evitar ser complejo en su interfaz, es decir que tenga una interfaz amigable.
- Ambos sistemas deben encontrarse conectados con el servidor.

**Servicios de negocio.**

- Crear nuevos procedimientos para ambos sistemas que en un futuro se vuelvan a utilizar.
- Reutilizar procedimientos ya creados para el sistema existente.

**Servicio de datos.**

- Crear la base de datos con los estándares parecidos a la base de datos existente, para que el usuario la utilice con el sistema.
- Proporcionar acceso a la base de datos nueva y a la existente de acuerdo al usuario del sistema.
- Se debe seguir un estándar para la definición de formas, tablas, procedimientos y variables.

**Lista de Funciones.**

- Proveer un navegador o "browser" de "Internet".
- Que el sistema genere reportes y consultas (gráficas o escritas).
- Apoyo a la alta gerencia con información necesaria a la mano.
- Reducir los tiempos de búsqueda de información necesaria.
- Un sistema con interfaz amigable.

**8.2.6 Factores críticos.**

- Un Factor Crítico del proyecto es que el desarrollo del sistema de condensación de información para el Data Warehouse se realice en el tiempo estimado, de lo contrario se presentará un retraso para el inicio del sistema de información ejecutiva en cuanto a programación y pruebas se refiere.
- Otro factor es el no contar con el equipo adecuado para el desarrollo del proyecto.

**8.2.7 Diseño de la interfaz.*****Estándares de interfaz.*****Sistema de Condensación de Información.**

- La forma principal del Sistema de Condensación de Información contendrá los botones de minimizar, maximizar y cerrar la forma. Contendrá un menú al

cual se le asignarán “short cuts” para desplegar otras formas. Otro de los estándares que se le brindará al usuario será una barra de íconos con el fin de realizar los procesos que se necesitan. Se proporcionará una barra de estatus para observar el procesamiento de información del sistema.

- Todas las formas que constituirán el Sistema de Condensación de Información tendrán como “background” un panel de Visual Basic. Se recomienda que se utilicen íconos que ayuden al usuario a identificar de una manera sencilla el proceso que esta realizando. Para mantener el tamaño de cada forma se empleará un margen de tamaño de 360 por 1185 twips. El botón de *Cancelar* se utiliza para deshacer y cerrar la forma. *Aceptar* indica que se esta de acuerdo en realizar el procesamiento de información.

#### Sistema de Información Ejecutiva.

- La forma principal del Sistema de Información Ejecutiva contendrá los botones de minimizar, maximizar y cerrar la forma. Contendrá un menú al cual se le asignarán frases con un color distinto al documento con el propósito de utilizarlos como ligas para desplegar otros documentos y así “navegar” cuando sea necesario. Otro de los estándares que se le brindará al usuario será una barra de íconos y botones del lado izquierdo de la forma con el fin de realizar los procesos que se necesiten. Se proporcionará una barra de estatus para observar el procesamiento de información del sistema.
- Se utilizará el logo de la empresa para el cual se le esta desarrollando el Sistema de Información Ejecutiva. Se emplearán imágenes con el fin de proporcionar una mejor presentación al sistema.
- Como estándar se tratará de utilizar el mouse más que el teclado.

#### **8.2.8 Pantallas y Reportes.**

En el Apendice B se encuentran las pantallas y los reportes tanto del Sistema de Información Ejecutiva de Ventas y Operaciones, así como el Sistema de Condensación.

#### **Objetivos de diseño del producto.**

- Proveer al usuario de la empresa ABA|TURISMO de un navegador o “browser” de “Internet” con acceso a información financiera, tendencias de productos y noticias Internas de una manera rápida y sencilla.

- Generar reportes y consultas (gráficas o escritas) de ventas por cliente, ventas por forma de pago, las ventas por proveedor, ventas por líneas aéreas, ventas por hoteles, ventas por clase de vuelo, ventas por destino y ventas por vendedor a través de un visualizador o "browser", permitiendo acceso a la información desde cualquier punto de la empresa.
- Reducir tiempos de búsqueda de información necesaria para que la alta gerencia defina su estrategia de ventas.
- Proveer al usuario un sistema con interfaz amigable con el fin de que interactue fácilmente con él, brindando diferentes formatos para la presentación de la información (escrita o gráfica).

#### Entregables.

Documento Visión	<i>Rolando Torres Y Christophe Bastiere</i>
Documento de Especificación Funcional	<i>Rolando Torres</i>
Manual de Usuario	<i>Christophe Bastiere</i>
Documento PEF	<i>Rolando Torres</i>

Tabla 11. Tabla de Documentos Entregables.

### 8.3 Diseño Conceptual.

#### 8.3.1 Escenarios del Sistema de información Ejecutiva de Ventas de ABA|TURISMO.

##### Escenario1: El usuario tenga acceso al Sistema de Información Ejecutiva de ventas de ABA|TURISMO.

##### Precondiciones:

1. Que el usuario necesite utilizar el sistema.

##### Acciones:

1. Capturar la dirección en el Explorer.
  - Si es valida dicha dirección.
    - a. Se pide una clave de acceso.
      1. Si es valida esa clave de acceso.
        - a. Se ejecuta la aplicación.
      2. Si no es valida esa clave de acceso.
        - a. Se regresa al escenario 1.

- Si no es válida la dirección.
  - a. Se regresa al escenario 1.

**Postcondiciones:**

1. Desplegar la primera página sin ningún problema para el usuario.

**Escenario 2: Consulta de las ventas de un cliente en particular.****Precondiciones:**

1. La información del cliente exista en la base de datos.

**Acciones:**

1. Desplegar la lista de clientes.
2. Seleccionar el cliente.
3. Capturar el intervalo de tiempo de la información que se desea.
4. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 3: Consulta de las ventas de todos los clientes.****Precondiciones:**

1. Que se encuentre la información (clientes) almacenada en la base de datos.

**Acciones:**

1. Desplegar la lista de clientes.
2. Capturar el intervalo de tiempo de la información que se desea.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 4: Consulta de las ventas de una aerolínea en particular.****Precondiciones:**

1. La información de la aerolínea exista en la base de datos.

**Acciones:**

1. Desplegar la lista de aerolíneas.
2. Seleccionar la aerolínea.
3. Capturar el intervalo de tiempo de la información que se desea.
4. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 5: Consulta de las ventas de todas las aerolíneas.****Precondiciones:**

1. Que se encuentre la información (aerolíneas) almacenada en la base de datos.

**Acciones:**

1. Desplegar la lista de las aerolíneas.
2. Capturar el intervalo de tiempo de la información que se desea.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 6: Consulta de las ventas de una aerolínea según un destino.****Precondiciones:**

1. La información de la aerolínea y de los destinos exista en la base de datos.

**Acciones:**

1. Desplegar la lista de aerolíneas.
2. Seleccionar la aerolínea.
3. Desplegar la lista de destinos.
4. Seleccionar el destino.
5. Capturar el intervalo de tiempo de la información que se desea.
6. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 7: Consulta de las ventas de un hotel en particular.****Precondiciones:**

1. La información del hotel exista en la base de datos.

**Acciones:**

1. Desplegar la lista de hoteles.
2. Seleccionar el hotel.
3. Capturar el intervalo de tiempo de la información que se desea.
4. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 8: Consulta de las ventas de todos los hoteles.****Precondiciones:**

1. Que se encuentre la información (hoteles) almacenada en la base de datos.

**Acciones:**

1. Desplegar la lista de los hoteles.
2. Capturar el intervalo de tiempo de la información que se desea.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 9: Consulta de las ventas de un hotel según un destino.****Precondiciones:**

1. La información del hotel y de los destinos exista en la base de datos.

**Acciones:**

1. Desplegar la lista de hoteles.
2. Seleccionar el hotel.
3. Desplegar la lista de destinos.
4. Seleccionar el destino.
5. Capturar el intervalo de tiempo de la información que se desea.
6. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 10: Consulta de las ventas de un proveedor de servicios en particular.****Precondiciones:**

1. La información del proveedor de servicios exista en la base de datos.

**Acciones:**

1. Desplegar la lista de proveedores de servicios.
2. Seleccionar el proveedor de servicios.
3. Capturar el intervalo de tiempo de la información que se desea.
4. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 11: Consulta de las ventas de todos los proveedores de servicios.****Precondiciones:**

1. Que se encuentre la información (proveedores de servicios) almacenada en la base de datos.

**Acciones:**

1. Desplegar la lista de los proveedores de servicios.
2. Capturar el intervalo de tiempo de la información que se desea.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 12: Consulta de las ventas de acuerdo a la forma de pago de los clientes.****Precondiciones:**

1. Que se encuentre la información (forma de pagos) almacenada en la base de datos.

**Acciones:**

1. Desplegar la lista de los tipos de forma de pago.
2. Capturar el intervalo de tiempo de la información que se desea.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 13: Consulta de las ventas según su clase de vuelo.****Precondiciones:**

1. La información de la aerolínea y clases de vuelo existan en la base de datos.

**Acciones:**

1. Desplegar la lista de aerolíneas.
2. Seleccionar la aerolínea.
3. Desplegar la lista de clases de vuelo.
4. Seleccionar la clase de vuelo.
5. Capturar el intervalo de tiempo de la información que se desea.
6. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 14: Consulta de las ventas realizadas por los agentes de ventas de la empresa.****Precondiciones:**

1. Que se encuentre la información (agentes de ventas) y sus ventas almacenada en la base de datos.

**Acciones:**

1. Desplegar la lista de los agentes de ventas.
2. Capturar el intervalo de tiempo de la información que se desea.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 15: Consulta de las ventas anuales de ABA|TURISMO.****Precondiciones:**

1. Que se encuentre la información (ventas) almacenada en la base de datos.

**Acciones:**

1. Desplegar la lista de los años.
2. Seleccionar el año deseado.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 16: Consulta de las ventas de sus mejores clientes.****Precondiciones:**

1. Que se encuentre la información (clientes) almacenada en la base de datos.

**Acciones:**

1. Capturar el intervalo de tiempo de la información que se desea.
2. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 17: Consulta de las ventas de sus mejores destinos.****Precondiciones:**

1. Que se encuentre la información (destinos) almacenada en la base de datos.

**Acciones:**

1. Capturar el intervalo de tiempo de la información que se desea.
2. Obtener la información de la base de datos y desplegar dicha información en pantalla con un formato tabular.

**Postcondiciones:**

1. El usuario observará sus ventas con el fin de tomar una decisión o contemplar una estrategia de ventas.

**Escenario 18: Consulta de las noticias internas de la empresa.****Precondiciones:**

1. La información de los boletines exista en la base de datos.

**Acciones:**

1. Desplegar la lista de boletines.
2. Seleccionar un boletín.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla.

**Postcondiciones:**

1. El usuario estará actualizado con lo que sucede en la empresa.

**Escenario 19: Consulta de la información de los clientes.****Precondiciones:**

1. La información de los clientes exista en la base de datos.

**Acciones:**

1. Desplegar la lista de clientes.
2. Seleccionar un cliente.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla.

**Postcondiciones:**

1. El usuario obtendrá la información del cliente deseado.

**Escenario 20: Consulta de la información de las aerolíneas.****Precondiciones:**

1. La información de las aerolíneas exista en la base de datos.

**Acciones:**

1. Desplegar la lista de aerolíneas.
2. Seleccionar una aerolínea.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla.

**Postcondiciones:**

1. El usuario obtendrá la información de la aerolínea deseada.

**Escenario 21: Consulta de la información de los proveedores de servicios.****Precondiciones:**

1. La información de los proveedores de servicios exista en la base de datos.

**Acciones:**

1. Desplegar la lista de los proveedores de servicios.
2. Seleccionar un proveedor de servicios.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla.

**Postcondiciones:**

1. El usuario obtendrá la información del proveedor de servicios deseado.

**Escenario 22: Consulta de la información de los hoteles.****Precondiciones:**

1. La información de los hoteles exista en la base de datos.

**Acciones:**

1. Desplegar la lista de hoteles.
2. Seleccionar un hotel.
3. Obtener la información de la base de datos y desplegar dicha información en pantalla.

**Postcondiciones:**

1. El usuario obtendrá la información del hotel deseado.

En el Apéndice C y D se encuentran explicados el diseño lógico y el diseño físico del proyecto.

## IX. ALCANCES Y LIMITACIONES DEL SISTEMA

Este capítulo expone los alcances y limitaciones del sistema los cuales se presentan desglosados por herramienta.

### 9.1 Alcances.

#### Sistema de Condensación de Información:

- ◆ El Data Warehouse contiene solamente información recabada de la base de datos operacional del Sistema de Ventas por Menudeo o Sistema de Apoyo a los Agentes de Ventas.
- ◆ La actualización de la información del sistema operacional al Data Warehouse debe de ser realizada mediante parámetros de tiempo (fecha inicial a fecha final) y clasificación de información (hoteles, proveedores de servicios, aerolíneas).
- ◆ La entrada a el sistema de condensación cuenta con mecanismos de seguridad pidiendo una contraseña y de esta manera prevenir que solamente las personas autorizadas lo utilicen.
- ◆ Una sola pantalla de alta, baja y cambios de usuarios del sistema.
- ◆ Realizar consultas de usuarios por número de identificación, nombre o apellidos.

### Sistema de Información Ejecutiva de Ventas y Operaciones:

- ◆ Es una aplicación que funciona dentro del Intranet, es decir, no funciona ni no requiere que posea comunicación externa (Internet).
- ◆ Las pantallas son dinámicas permitiendo seleccionar las opciones que se requieran para formar la consulta de alguna venta, para posteriormente obtener los datos del Data Warehouse y desplegarlos en forma de HTML.
- ◆ La entrada al sistema ejecutivo se realiza mediante una contraseña única por usuario con el propósito de permitir la entrada solamente a usuarios autorizados.
- ◆ El conjunto de pantallas del sistema en el "browser" poseen los estándares de Interfaz para navegadores de ABACO Grupo Financiero.
- ◆ Las consultas que se pueden realizar son las siguientes: ventas por cliente, ventas por agente de ventas, ventas por forma de pago, ventas por hotel, ventas por proveedor de servicios, ventas por aerolínea, ventas totales, ventas por clase de vuelo, ventas por aerolínea según un destino, ventas por hotel según un destino, ventas de sus 10 mejores clientes y las ventas de sus 10 mejores destinos.
- ◆ El Sistema cuenta con un desplegado de las noticias internas de ABA|TURISMO organizadas por fecha.

## **9.2 Limitaciones.**

### Sistema de Condensación de Información:

- ◆ El concepto de metadatos no puede ser tecnológicamente apreciado, debido a que no se cuentan con herramientas que permitan la realización de esto.
- ◆ Al no contar con herramientas que permitan la definición de los metadatos, se dificulta la condensación de la información, teniéndola que realizar a partir de estatutos de SQL.

### Sistema de Información Ejecutiva de Ventas y Operaciones:

- ◆ Las funciones definidas para la exploración del Data Warehouse no pueden ser apreciadas como tales debido básicamente a dos restricciones: no se cuenta con herramientas desarrolladas para ese fin y la utilización del navegador es por su misma naturaleza demasiado primitivo, lo cual dificulta el desarrollo de pantallas que simulen la exploración utilizando esas funciones.
- ◆ Las consultas se limitan al nombre de los clientes, agentes de ventas, pve. de servicios, aerolínea y hotel, es decir, no se muestran los identificadores de cada uno de ellos, ni datos acerca de los mismos.

## X. EL FUTURO DEL SISTEMA

Las dos herramientas desarrolladas se realizaron con tecnología de punta, la cual brinda un gran futuro a las mismas. A continuación se exponen una serie de propuestas las cuales pueden ser tomadas en consideración para mejorar el presente proyecto en un futuro.

- Desarrollar o comprar herramientas de definición de Data Warehouse con el propósito de facilitar la definición de los metadatos.
- Incrementar el alcance del Data Warehouse concentrando datos de otras aplicaciones hasta contener todos los datos de ABA|TURISMO.
- Agregar aspectos de seguridad para la implementación de la aplicación en "Internet" en caso de que sea requerido.
- En el Sistema de Información Ejecutivo se podrían agregar parámetros para configurar las gráficas y las tabulaciones que se generan, por ejemplo, parámetros del color, tipo, tamaño, estilo, etcétera.

- El Sistema de Información Ejecutivo permita realizar tabulaciones y gráficas comparativas entre distintos criterios, por ejemplo, las ventas de los hoteles en el año de 1995 comparadas con las ventas de los hoteles en el presente año.
- El Sistema de Información Ejecutivo podría permitir grabar las consultas realizadas, con el propósito de obtenerlas después, evitando tener que volver a llenar los parámetros de las mismas.
- El Sistema de Información Ejecutivo podría permitir elaborar consultas a partir de los datos existentes, contemplándolos como variables para realizar los “queries” y así lograr consultas de alta complejidad.
- Las consultas del Sistema de Información Ejecutivo podrían realizarse en una pantalla más elaborada donde contenga todas las categorías de los distintos datos del Data Warehouse y así facilitarle las consultas al Ejecutivo.
- Integración de las herramientas del navegador con el Sistema de Ventas por Menudeo, permitiendo realizar consultas a los datos de este sistema, es decir, acceso a los datos del día a día.

## XI. CONCLUSIONES

Todo proyecto debe de comenzar con la obtención de los requerimientos y necesidades de los usuarios, para posteriormente alinearlos a los objetivos de las organizaciones. Una vez que se realiza la alineación, estaremos en posibilidad de generar una arquitectura del sistema que realmente satisfaga las necesidades antes recabadas.

Esa arquitectura generada, servirá a la gente de sistemas como guía y a partir de ella se generará el conjunto de aplicaciones que podemos desarrollar. Para todas las aplicaciones deberemos definir hacia dónde queremos llegar, qué funciones debe de contener y definiremos un conjunto de estándares que deben de estar presentes en cada uno de estos proyectos, como por ejemplo, las herramientas que se pueden utilizar dependiendo del tipo de proyecto.

Para cada proyecto, será necesario definir la metodología o metodologías que se van a emplear, así como el planteamiento de las herramientas más adecuadas para resolverlo.

En este proyecto en particular, una vez que obtuve los requerimientos, me di cuenta de la importancia del "performance" que deberían tener las consultas ya que mi usuario es un ejecutivo. A partir de ello, empecé a buscar las tecnologías que me ayudaran a satisfacer este requerimiento. Es aquí donde surge Data Warehouse como una necesidad.

El primer paso para la construcción del Data Warehouse, es conocer a detalle esta tecnología, y esto incluye empaparse de la tecnología en sí, de las herramientas existentes, de las metodologías de la construcción, pero principalmente de los metadatos y de las funciones para la consulta del mismo.

Entre más completa sea la definición de los metadatos, más fácilmente podrán los ejecutivos realizar las consultas.

A mi consideración, este tipo de proyectos se realizarían de forma más completa, pero, principalmente con menos dificultades si se cuenta con herramientas desarrolladas con propósitos de generación de Data Warehouses y herramientas de explotación.

El haber utilizado como interfaz del Sistema de Información Ejecutivo un navegador o "browser", me enriqueció bastante, pero, a la vez vi limitada mucha funcionalidad al querer explotar la información contenida en el Data Warehouse.

Un factor importante en el desarrollo de este proyecto, es que actualmente en ABA|TURISMO, se están desarrollando más sistemas y el Data Warehouse viene a ser la consolidación de información necesaria para el ejecutivo.

La utilización de herramientas como Microsoft Front-Page me facilitaron el desarrollo de las páginas de Web, es decir, la generación del lenguaje HTML; así como, el ERWIN, herramienta que me permitió generar y modificar más fácilmente la base de datos.

Yo pienso que si se contasen con herramientas que ayuden a la construcción y explotación del Data Warehouse, se podrían aplicar de una mejor manera los conceptos que aprendí al respecto, como por ejemplo, si tuviese una herramienta donde pudiera definir los metadatos, realmente los Ejecutivos y tomadores de decisiones estarían en la posibilidad de explotar el Data Warehouse y obtener cualquier consulta que ellos desearan. Pero, sin las herramientas, obviamente es necesaria la construcción de un front-end estático, es decir, que requiere ser construido para cada tipo de consulta establecido por los ejecutivos.

El haber utilizado la metodología orientada a objetos para el desarrollo del sistema, requirió para el proyecto alinear actividades de la construcción del Data Warehouse, como por ejemplo, el diseño lógico y físico de la base de datos.

Por último, el desarrollo del PEF, me proporcionó el contacto con el mundo real, el cual considero uno de los mayores logros de nuestra Universidad.

## XII. BIBLIOGRAFÍA

[BASTIERE, 94] Bastiere Christophe, Documento de Historia e Información General de ABA|TURISMO ,México, 1996 .

[BAUM, 96] Baum, David, Data Warehouse. Building Blocks for the Next Milenium, Oracle Magazine, E.U.A., March/April 1996.

[BUTLER, 97] Butler GroBusiness, The Web Management Guide , Butler Group, E.U.A. ,1997.

[DATE, 93] Date C.J., Sistemas de Bases de Datos, Addison-Wesley , Iberoamericana E. U. A., 1993.

[DECISYS, 96] Decisys, Intranet Impact , Decisys, Inc. Staff Member, E.U.A., August 1997.

- [DEVELOPMENT, 92] Development Library, Designing Client-Server Applications for Enterprise Data Base Connectivity, MS-Development Library E.U. A. 1992.
- [DUBE, 96] Dube, Jerry & Pierson, Donna, IBM's Intranet Offerings: Pathways to New Heights, Intranet and Client/Server, IBM Corporation, E.U.A., 1996.
- [DUMAS, 88] Dumas, Joseph S., Designing User Interfaces for Software, Prentice-Hall. E. U. A., 1988.
- [ELLISON, 96] Ellison, Larry, Data Warehouse DBMS Scalability: Introducing a Set of Evaluation Criteria, Gartner Group 1996.
- [FLORES, 97] Flores Gabriel, Intranet: una solución de tráfico a la información corporativa, RED, México 1997
- [GILL, 96] Gill, Harjinder S. y Rao, Prakash C., Data Warehousing, Prentice Hall. México, 1996.
- [GILLOOLY, 97] Gillooly, Caryn, Blazing a Trail in Intranet Usage, Information Week, E.U.A. 1997.
- [HAN, 97] Han Kevin, How could the Intranet redefine corporate life as we know it?, San Francisco, C.A. E.U.A., 1997.
- [INFORMIX, 96] Informix, Designing the Data Warehouse on Relational Databases, Stanford Technology Group, Inc. An Informix Company. E.U.A. 1997.
- [INMON, 94] Inmon, W.H. & Hackathorn, R.D., Using the Data Warehouse, Wiley-QED Publications. E. U. A., 1994.
- [INMON, 95] Inmon, W.H, Tech Topic. What is a Data Warehouse?, Prism Solutions, Inc., E.U.A., 1995
- [JEON, 96] Jonathan, Jeon, The Intranet- a Cooperation Revolution, Intranet Discussion Board, E.U.A 1997.
- [KIMBALL, 96] Kimball, Ralph, Data Warehouse Architecture- Nine Decisions in the Design of a Data Warehouse, DBMS online, E.U.A. 1996.
- [LINTHICUM, 96] Linthicum, David. S, Java gets ready for client/server primetime with the JDBC database access API, Internet Systems, October 1996.

- [LINTHICUM, 96a] Linthicum, David. S, Taking stock of the available development tools and what may work for you, Internet Systems, E.U.A., October 1996.
- [MARTIN, 94] Martin, James & Odell, James J., Análisis Orientado a Objetos , Prentice-Hall México, 1994
- [MCGUFF, 96] McGuff Frank, Data Modeling for Data Warehouses , Evolution of Data Ware, E.U.A 1996.
- [MEYER, 97] Meyer, Ingrid, Seeking Shelter in the 'Intranet' , CommunicationsWeek, E.U.A., April 1997.
- [NEILD, 97] Neild Jerry, Building a Better Web Site , IBM Network Computing Solutions. Web Site Architect., IBM Corporation, E.U.A., 1996.
- [PARKES, 96] Parkes, Clara H., A behind- the scenes look at powerful, databases-driven web sites., Internet Systems, E.U.A., October 1996.
- [PUERTAS, 96] Puertas Antonio, Data Warehouse, se ve caro... lo es , Expansion, México , Noviembre 1996.
- [RAHMEL, 96] Rahmel Dan, Evaluating scripting engines from Netscape and Microsoft, Internet Systems, E.U.A., October 1996.
- [RED, 97] Red Brick Systems, Specialized Requirements for Relational Data Warehouse Servers, A Red Brick Systems White Paper, Los Gatos, C.A. E.U.A 1997.
- [RED, 97a] Red Brick Systems, Star Schemas and STARjoin Technology , A Red Brick System White Paper, Los Garos, C.A. E.U.A. 1997
- [RED, 97b] Red Brick Systems, Decision-Makers, Business Data, and RISQL , A Red Brick System White Paper, Los Garos, C.A. E.U.A. 1997.
- [RODRIGUEZ, 96] Rodríguez Leyva Ricardo, Data Warehousing: Bases de datos para la toma de decisiones. RED Ejecutivos, México, 1996.
- [RODRÍGUEZ, 96a] Ricardo Rodríguez Leyva, Data Warehousing: Bases de datos para la toma de decisiones. , RED, México 1996.
- [RWAY , 97] Rway Communications, So what is an Intranet? , Rway Communications Intranet Article, E.U.A. 1997.

[SDD, 94] Solutions Development Discipline. , Microsoft Solutions Framework., Microsoft Corporation 1994

[SURF, 97] SurfCONTROL, Internet-A Brief History , SurfCONTROL JSB Computer Systems Ltd, E.U.A. 1997

[SURF, 97<sup>a</sup>] SurfCONTRL, The Intranet- A Corporate Revolution , SurfCONTROL JSB Computer Systems Ltd, E.U.A. 1997.

[WILSON, 97] Wilson, Tim, Intranets Reach Critical Mass- Businesses Ride the Wave , CommunicationsWeek, E.U.A., April 1997.

#### Manuales :

[ALLAIRE, 96] Allaire Corporation., Cold Fusion 1.5 , Allaire Corporation E.U.A., 1996

[MEYERS, 95] Meyers III, Walter H., Erwin/ERX 2.1 , Microsoft Corporation E.U.A., 1995

[MICROSOFT, 94] Building Client-Server Solutions with Microsoft Access, Visual Basic and SQL Server. (Part 1, Part2)  
Microsoft Consulting Services E. U. A., 1994.

[MICROSOFT, 95] Microsoft Education and Certification, Implementing a Database Design on Microsoft SQL Server 6.5 ,Microsoft Corporation E. U. A., 1995.

[MICROSOFT, 95<sup>a</sup> ] Microsoft Corporation, Microsoft Visual Basic 4.0 Manual del Usuario., Microsoft Corporation E.U.A., 1995

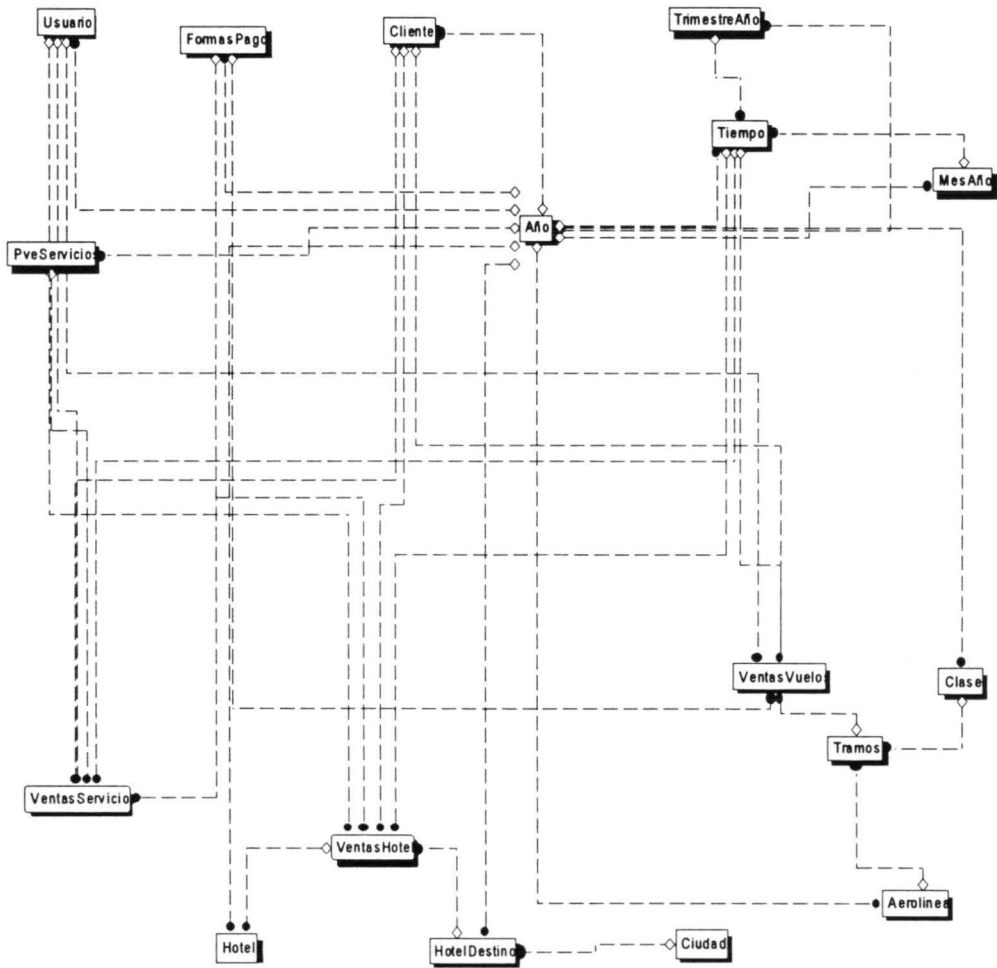
[MICROSOFT, 95b] Microsoft Corporation, Microsoft Windows 95 Manual del Usuario. Microsoft Corporation E.U.A., 1995

[MICROSOFT, 95c] Microsoft Corporation, Microsoft SQL Server 6.5 Manual del Usuario. Microsoft Corporation E.U.A., 1995

[MICROSOFT, 97] Microsoft Corporation, Microsoft Front Page 97 Manual del Usuario. Microsoft Corporation E.U.A., 1995

# APÉNDICE A

## Diseño de la Base de Datos



## Estructura de la Base de Datos.

Aerolínea				
Aerolineald	P	int	not null	Identificador de la aerolínea.
AñoId	F	int	not null	Identificador del año
AerolineaDescripcion		char(30)	not null	Nombre de la aerolínea
AerolineaAbreviación		char(30)	not null	Abreviación del nombre de la aerolínea
AerolineaMonto		money	null	Monto condensado anual de la aerolínea

Cliente				
Clienteld	P	int	not null	Identificador del cliente
AñoId	F	int	not null	Identificador del año
ClienteNombre		char(20)	not null	Nombre de la cliente
ClienteApPaterno		char(15)	not null	Apellido Paterno del cliente
ClienteApMaterno		char(15)	null	Apellido Materno del cliente
ClienteMonto		money	null	Monto condensado anual del cliente

Hotel				
Hoteld	P	int	not null	Identificador del hotel
AñoId	F	int	not null	Identificador del año
HotelNombre		char(45)	not null	Nombre del hotel
HotelMonto		money	null	Monto condensado anual del hotel

PveServicios				
PveServiciosId	P	int	not null	Identificador del PveServicios
AñoId	F	int	not null	Identificador del año
PveServiciosNombre		char(45)	not null	Nombre del proveedor de servicios
PveServiciosMonto		money	null	Monto condensado anual del proveedor de servicios

UsuarioAg				
UsuarioAgId	P	int	not null	Identificador del agente de ventas
AñoId	F	int	not null	Identificador del año
UsuarioAgNombre		char(20)	not null	Nombre del agente de ventas
UsuarioAgApPaterno		char(15)	not null	Apellido Paterno del agente de ventas
UsuarioAgApMaterno		char(15)	null	Apellido Materno del agente de ventas
UsuarioAgMonto		money	null	Monto condensado anual del agente de ventas

FormasPago				
FormasPagold	P	int	not nul	Identificador de la forma de pago.
Añold	F	int	not nul	identificación del año
FormasPagoDescripcion		char(25)	not nul	descripcion de la forma de pago
FormasPagoMonto		money	null	Monto condensado anual de la forma de pago

Ciudad				
CiudadId	P	int	not null	Identificador de la Ciudad
CiudadDescripcion		char(45)	not null	Descripcion o Nombre de la Ciudad

Clase				
Claseld	P	int	not null	Identificador de la clase.
Añold	F	int	not null	Identificador de el año
ClaseDescripcion		char(25)	not null	Descripción de la clase
ClaseMonto		money	null	Monto condensado anual de la clase

Tramos				
TramosId	P	int	not null	Identificador del tramo.
Aerolineald	F	int	not null	Identificador de la aerolínea
Claseld	F	int	not null	Identificador de la clase
CdDestinold		int	not null	Identificador de la ciudad destino
CdSalidald		int	not null	Identificador de la ciudad salida
TramosMonto		money	null	Monto condensado anual de los tramos

Año				
Añold	P	int	not null	Identificador del año
AñoDescripcion		char(4)	not null	Descripción del año
AñoMonto		money	null	Monto condensado anual

TrimestreAño				
TrimestreAñold	P	int	not null	Identificador del trimestre según el año
Añold	F	int	not null	Identificador del año
TrimestreAñoDescripcion		char(5)	not null	Descripción del trimestre segun el año
TrimestreAñoMonto		money	null	Monto condensado trimestral

MesAño				
MesAñold	P	int	not null	Identificador del mes segun un año
Añold	F	int	not null	Identificador del año
MesId		int	not null	Identificador del mes segun un año
MesAñoDescripcion		char(20)	not null	Descripción del mes
MesAñoMonto		money	null	Monto condensado mensual

VentasVuelos				
VentasVuelosId	P	int	not null	Identificador de la ventas de vuelos
TramosId	F	int	null	Identificador del tramo
Usuarioid	F	int	null	Identificador del agente de ventas
FormasPagoid	F	int	null	Identificador de la forma de pago
ClienteId	F	int	null	Identificador del cliente
Tiempoid	F	int	null	Identificador del tiempo
VentasVuelosMonto		money	null	Monto condensado de los vuelos

Ventas Hotel				
Ventas HotelId	P	int	not null	Identificador de las ventas de hotel
HotelId	F	int	null	Identificador del hotel
HotelDestinoid	F	int	null	Identificador del hotel según un destino
Usuarioid	F	int	null	Identificador del agente de ventas
FormasPagoid	F	int	null	Identificador de la forma de pago
ClienteId	F	int	null	Identificador del cliente
Tiempoid	F	int	null	Identificador del tiempo
VentasHotelMonto		money	null	Monto condensado de las ventas de hotel

VentasServicios				
VentasServiciosId	P	int	not null	Identificador de las ventas de servicios
PveServiciosId	F	int	null	Identificador del proveedor de servicios
Usuarioid	F	int	null	Identificador del agente de ventas
FormasPagoid	F	int	null	Identificador de la forma de pago
ClienteId	F	int	null	Identificador del cliente
Tiempoid	F	int	null	Identificador del tiempo
VentasHotelMonto		money	null	Monto condensado de ventas de servicios

HotelDestino				
HotelDestinoid		int	not null	Identificador del hotel según su destino
AñoId		int	not null	Identificador del año
CiudadId		int	not null	Identificador de la ciudad
HotelId		int	not null	Identificador del hotel según su destino
HotelNombre		char(45)	not null	Descripción o Nombre del hotel
HoteldestinoMonto		money	null	Monto condensado anual del hotel según un destino

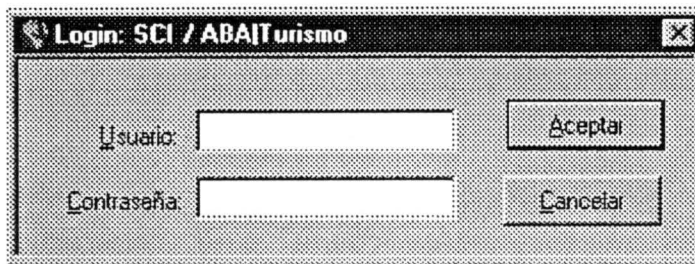
## APÉNDICE B.

### Pantallas y Reportes

A continuación se presentan las pantallas y reportes principales de este proyecto.

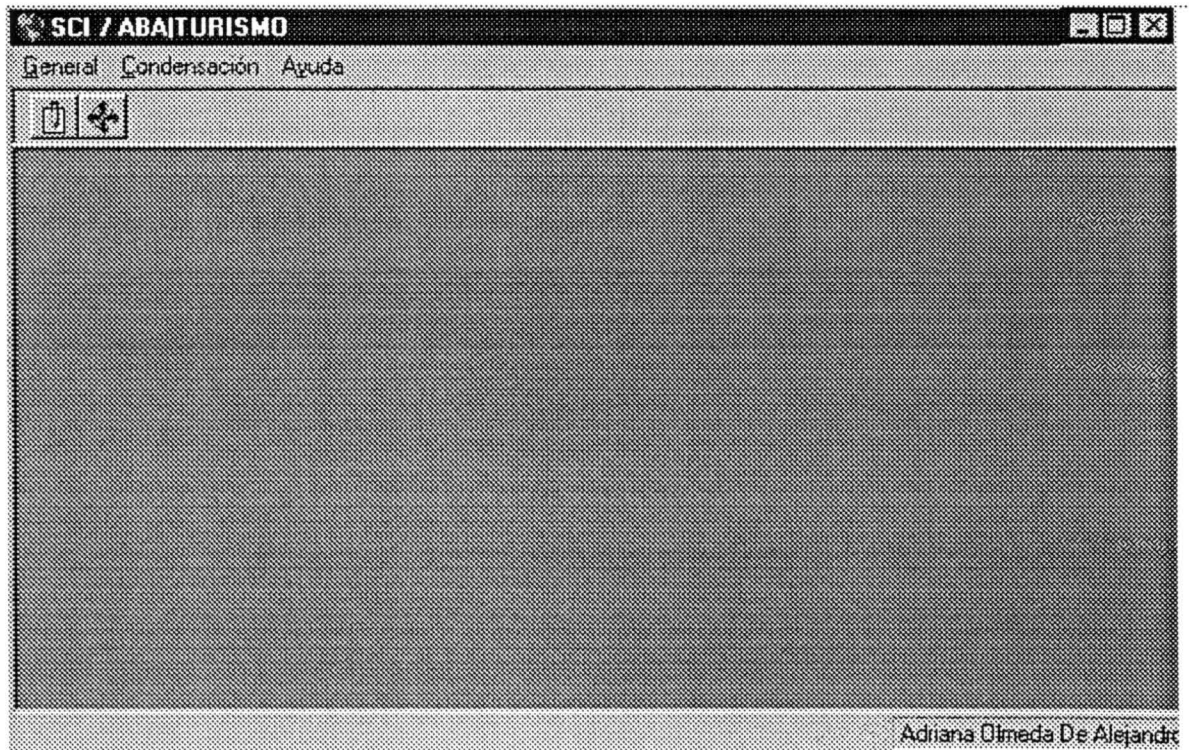
#### Sistema de Condensación de Información

Pantalla de login al sistema.



The screenshot shows a window titled "Login: SCI / ABAITurismo". It contains two input fields: "Usuario:" and "Contraseña:". To the right of the "Usuario:" field is a button labeled "Aceptar". To the right of the "Contraseña:" field is a button labeled "Cancelar".

#### Pantalla principal



## Pantalla de Usuario

The screenshot shows a window titled "SCI / Usuarios". At the top, there is a "Número:" label followed by a text input field and a small square checkbox. Below this is a section titled "Datos del Usuario" which contains several fields: "Nombre:" with a text input field, "Apellido Paterno:" with a text input field, "Apellido Materno:" with a text input field, "Teléfono:" with a text input field containing "( )", "Login:" with a text input field, "Password:" with a text input field, and "Prioridad:" with a dropdown menu. At the bottom of the window, there are four buttons: "Limpiar", "Agregar", "Eliminar", and "Cancelar".

## Pantalla de Condensación

The screenshot shows a window titled "SCI / Condensar". At the top, there is a "Fecha:" label followed by a text input field containing "5/15/97" and a small square checkbox. To the right of the date field is a label "A" followed by another text input field. Below this is a section titled "Opciones" which contains four checkboxes: "Ventas Globales", "Ventas Hotel", "Ventas Servicios", and "Ventas Vuelos". At the bottom of the window, there are two buttons: "Agregar" and "Cancelar".

---

Sistema de Información Ejecutiva de Ventas y Operaciones

## Pantalla Principal

ABACO  
GRUP  
FINANCIERO

ABA|TURISMO

Información Financiera

Noticias Internas

Directorio

**Bienvenidos al Home  
Page de la empresa  
ABA|TURISMO**

ABA|TURISMO es una agencia de viajes la cual ofrece una diversidad de productos entre los que se encuentran: las reservaciones en vuelos, habitaciones de hotel, paquetes para eventos especiales y paquetes turísticos.

Esta página tiene como fin brindar a los usuarios INFORMACION FINANCIERA, NOTICIAS INTERNAS y TENDENCIA DE PRODUCTOS de manera oportuna y confiable. Así como un DIRECTORIO de los servicios.



Pantalla de información Financiera





## APÉNDICE C

### Diseño Lógico

#	Iniciador de la acción	Acción	Objeto de Negocio	Servicio	Entradas	Salidas
1	Usuario	Captura nombre y password	Aplicación	Entrar a la aplicación		
2	Usuario	Seleccionar el Cliente	Cliente	Proveer lista de Clientes	Datos del Cliente	
	Usuario	Captura el intervalo de tiempo.	Cliente	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por un cliente en particular.
	Aplicación	Validar datos	Cliente	Checar que los datos esten correctos.		
3	Usuario	Seleccionar todos los Clientes	Cliente	Proveer lista de Clientes	Datos del Cliente	
	Usuario	Captura el intervalo de tiempo.	Cliente	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Clientes..
	Aplicación	Validar datos	Cliente	Checar que los datos esten correctos.		
4	Usuario	Seleccionar la Aerolínea	Aerolínea	Proveer lista de Aerolíneas	Datos de la Aerolínea	
	Usuario	Captura el intervalo de tiempo.	Aerolínea	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por una Aerolínea en particular.
	Aplicación	Validar datos	Aerolínea	Checar que los datos esten correctos.		

#	Iniciador de la acción	Acción	Objeto de Negocio	Servicio	Entradas	Salidas
5	Usuario	Seleccionar la Aerolínea y el Destino.	Aerolínea	Proveer lista de Aerolínea y Destino	Datos de la Aerolínea y Destino	
	Usuario	Captura el intervalo de tiempo.	Aerolínea	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las vtas. por aerolínea según un destino.
	Aplicación	Validar datos	Aerolínea	Checar que los datos esten correctos.		
6	Usuario	Seleccionar todas las Aerolíneas	Aerolínea	Proveer lista de Aerolíneas	Datos de la Aerolínea	
	Usuario	Captura el intervalo de tiempo.	Aerolínea	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Aerolíneas .
	Aplicación	Validar datos	Aerolínea	Checar que los datos esten correctos.		
7	Usuario	Seleccionar el Hotel	Hotel	Proveer lista de Hoteles	Datos de los Hoteles	
	Usuario	Captura el intervalo de tiempo.	Hotel	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por un Hotel en particular.
	Aplicación	Validar datos	Hotel	Checar que los datos esten correctos.		
8	Usuario	Seleccionar todos los hoteles	Hotel	Proveer lista de Hoteles	Datos del Hotel	
	Usuario	Captura el intervalo de tiempo.	Hotel	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Hoteles .
	Aplicación	Validar datos	Hotel	Checar que los datos esten correctos.		

#	Iniciador de la acción	Acción	Objeto de Negocio	Servicio	Entradas	Salidas
9	Usuario	Seleccionar el Hotel y el Destino.	Hotel	Proveer lista de Hoteles y Destino	Datos del Hotel y Destino	
	Usuario	Captura el intervalo de tiempo.	Hotel	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por hotel según un destino.
	Aplicación	Validar datos	Hotel	Checar que los datos esten correctos.		
10	Usuario	Seleccionar el Pve. Servicio	Pve. Servicio	Proveer lista de Pve. Servicios	Datos de los Pve. Servicios	
	Usuario	Captura el intervalo de tiempo.	Pve. Servicio	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por un Pve. Servicio en particular.
	Aplicación	Validar datos	Pve. Servicio	Checar que los datos esten correctos.		
11	Usuario	Seleccionar todos los Pve. Servicios	Pve. Servicio	Proveer lista de Pve. Servicios	Datos de los Pve. Servicios	
	Usuario	Captura el intervalo de tiempo.	Pve. Servicio	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Pve. Servicios .
	Aplicación	Validar datos	Pve. Servicio	Checar que los datos esten correctos.		

#	Iniciador de la acción	Acción	Objeto de Negocio	Servicio	Entradas	Salidas
12	Usuario	Seleccionar el Forma de Pago	Forma de Pago	Proveer lista de Forma de Pagos	Datos de los Forma de Pagos	
	Usuario	Captura el intervalo de tiempo.	Forma de Pago	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Forma de Pago
	Aplicación	Validar datos	Forma de Pago	Checar que los datos esten correctos.		
13	Usuario	Seleccionar todos los Clase de Vuelo	Aerolínea	Proveer lista de Clases de Vuelo	Datos de la Clase de Vuelo	
	Usuario	Captura el intervalo de tiempo.	Aerolínea	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Clase de Vuelo
	Aplicación	Validar datos	Aerolínea	Checar que los datos esten correctos.		
14	Usuario	Seleccionar el Agente de Ventas	Agente de Ventas	Proveer lista de Agente de Ventas	Datos de los Agente de Ventas	
	Usuario	Captura el intervalo de tiempo.	Agente de Ventas	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Agente de Ventas
	Aplicación	Validar datos	Agente de Ventas	Checar que los datos esten correctos.		
15	Usuario	Captura el intervalo de tiempo.	Todos	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas por Globales.
	Aplicación	Validar datos	Todos	Checar que los datos esten correctos.		

#	Iniciador de la acción	Acción	Objeto de Negocio	Servicio	Entradas	Salidas
16	Usuario	Captura el intervalo de tiempo.	Cliente	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas de los mejores Clientes
	Aplicación	Validar datos	Cliente	Checar que los datos esten correctos.		
17	Usuario	Captura el intervalo de tiempo.	Aerolinea	Proveer la consulta segun el tiempo deseado.	Datos del tiempo	Despliega los datos de las ventas de los mejores Destinos
	Aplicación	Validar datos	Aerolinea	Checar que los datos esten correctos.		
18	Usuario	Seleccionar el número de boletín y la fecha	Todas	Proveer la consulta segun el número y el tiempo deseado.	Datos del tiempo	Despliega los datos de las noticias del negocio
	Aplicación	Validar datos	Todas	Checar que los datos esten correctos.		
19 20 21	Usuario	Seleccionar el Cliente, Hotel, Aerolinea, Pve. Servicios	Todas	Proveer la consulta de Cliente, Hotel, Aerolínea, Pve. Servicios	Datos del tiempo	Despliega los datos de Cliente, Hotel, Aerolínea, Pve. Servicios
	Aplicación	Validar datos	Todas	Checar que los datos esten correctos.		

## APÉNDICE D.

### Diseño Físico.

Objeto de Negocio	Servicio de Usuario	Servicio de Negocio	Servicio de Datos	Quién lo hace	Forma
Aplicación	Entrar_Aplicación		TmpLogin	SUB	TmpLogin
Hotel	Selección de Datos		TmpSeleccionHotel	CF	TmpSeleccionHotel
Hotel	Despliegue de Ventas		TmpGridHotel	CF	TmpGridHotel
Aerolínea	Selección de Datos		TmpSeleccion Aerolínea	CF	TmpSeleccion Aerolínea
Aerolínea	Despliegue de Ventas		TmpGridAerolínea	CF	TmpGridAerolínea
Agente de Ventas	Selección de Datos		TmpSeleccion UsuarioAg	CF	TmpSeleccion UsuarioAg
Agente de Ventas	Despliegue de Ventas		TmpGridUsuarioAg	CF	TmpGridUsuarioAg
Pve. Servicio	Selección de Datos		TmpSeleccion Pve.Servicios	CF	TmpSeleccion Pve.Servicios
Pve. Servicio	Despliegue de Ventas		TmpGridServicios	CF	TmpGridServicios
Cliente	Selección de Datos		TmpSeleccion Cliente	CF	TmpSeleccion Cliente
Cliente	Despliegue de Ventas		TmpGridCliente	CF	TmpGridCliente
Cliente	Selección de Datos		TmpSeleccion Noticia	CF	TmpSeleccion Noticia
Cliente	Despliegue de Noticias		TmpNoticias	CF	TmpNoticias
Todos	Selección de Datos		TmpSeleccion Directorio	CF	TmpDirectorio
Todos	Despliegue de Información		TmpDirectorio	CF	TmpDirectorio

## APÉNDICE E.

### Código

#### Sistema de Condensación – Back-End.

```
/****** Object: Table dbo.Aerolinea  Script Date: 5/15/97 1:00:14 AM *****/
```

```
CREATE TABLE dbo.Aerolinea (  
    Aerolineaid int NOT NULL ,  
    Aºold int NOT NULL ,  
    AerolineaDescripcion char (15) NOT NULL ,  
    AerolineaAbreviacion char (5) NOT NULL ,  
    AerolineaMonto money NULL
```

```
)  
GO
```

```
/****** Object: Table dbo.Aºo  Script Date: 5/15/97 1:00:15 AM *****/
```

```
CREATE TABLE dbo.Aºo (  
    Aºold int IDENTITY (1, 1) NOT NULL ,  
    AºoDescripcion char (4) NOT NULL ,  
    AºoMonto money NOT NULL
```

```
)  
GO
```

```
/****** Object: Table dbo.Ciudad  Script Date: 5/15/97 1:00:15 AM *****/
```

```
CREATE TABLE dbo.Ciudad (  
    CiudadId int NOT NULL ,  
    CiudadDescripcion char (25) NOT NULL
```

```
)  
GO
```

```
/****** Object: Table dbo.Clase  Script Date: 5/15/97 1:00:15 AM *****/
```

```
CREATE TABLE dbo.Clase (  
    Claseld int NOT NULL ,  
    ClaseDescripcion char (25) NULL ,  
    ClaseMonto money NOT NULL ,  
    Aºold int NOT NULL
```

```
)  
GO
```

```
/****** Object: Table dbo.Cliente  Script Date: 5/15/97 1:00:15 AM *****/
```

```
CREATE TABLE dbo.Cliente (  
    Clienteld int NOT NULL ,  
    ClienteNombre char (20) NOT NULL ,  
    ClienteApPaterno char (15) NOT NULL ,  
    ClienteApMaterno char (15) NULL ,  
    ClienteMonto money NULL ,  
    Aºold int NOT NULL
```

```
)  
GO
```

```
/****** Object: Table dbo.FormasPago  Script Date: 5/15/97 1:00:16 AM *****/
```

```
CREATE TABLE dbo.FormasPago (  
    FormasPagold int NOT NULL ,  
    FormasPagoDescripcion char (25) NOT NULL ,  
    FormasPagoMonto money NOT NULL ,  
    Aºold int NOT NULL
```

```
)  
GO
```

```
/****** Object: Table dbo.Hotel  Script Date: 5/15/97 1:00:16 AM *****/
```

```
CREATE TABLE dbo.Hotel (  
    HotelId int NOT NULL ,  
    HotelNombre char (45) NOT NULL ,  
    HotelMonto money NOT NULL ,  
    Aºold int NOT NULL
```

```
)
```

GO

/\*\*\*\*\* Object: Table dbo.HotelDestino Script Date: 5/15/97 1:00:16 AM \*\*\*\*\*/

```
CREATE TABLE dbo.HotelDestino (  
    HotelDestinoId int IDENTITY (1, 1) NOT NULL ,  
    HotelId int NOT NULL ,  
    AñoId int NOT NULL ,  
    CiudadId int NOT NULL ,  
    HotelNombre char (45) NOT NULL ,  
    HotelDestinoMonto money NULL
```

)

GO

/\*\*\*\*\* Object: Table dbo.MesAño Script Date: 5/15/97 1:00:17 AM \*\*\*\*\*/

```
CREATE TABLE dbo.MesAño (  
    MesAñoId int IDENTITY (1, 1) NOT NULL ,  
    MesId int NOT NULL ,  
    AñoId int NOT NULL ,  
    MesAñoDescripcion char (20) NOT NULL ,  
    MesAñoMonto money NULL
```

)

GO

/\*\*\*\*\* Object: Table dbo.Metadato Script Date: 5/15/97 1:00:17 AM \*\*\*\*\*/

```
CREATE TABLE dbo.Metadato (  
    MetadatoId int NOT NULL ,  
    MetadatoNombreSP char (25) NOT NULL ,  
    MetadatoDescripcion char (50) NULL ,  
    MetadatoTablasFuentes char (50) NULL ,  
    MetadatoTablasAfectadas char (50) NULL ,  
    MetadatoUltActual datetime NULL ,  
    MetadatoRespNombre char (20) NOT NULL ,  
    MetadatoRespApPaterno char (15) NOT NULL ,  
    MetadatoRespApMaterno char (15) NULL
```

)

GO

/\*\*\*\*\* Object: Table dbo.PveServicios Script Date: 5/15/97 1:00:17 AM \*\*\*\*\*/

```
CREATE TABLE dbo.PveServicios (  
    PveServiciosId int NOT NULL ,  
    PveServiciosNombre char (45) NOT NULL ,  
    PveServiciosMonto money NOT NULL ,  
    AñoId int NOT NULL
```

)

GO

/\*\*\*\*\* Object: Table dbo.Temporal# Script Date: 5/15/97 1:00:18 AM \*\*\*\*\*/

```
CREATE TABLE dbo.Temporal# (  
    FechaYY int NULL ,  
    FechaQTR varchar (30) NULL ,  
    FechaMM int NULL ,  
    Fechadd int NULL
```

)

GO

/\*\*\*\*\* Object: Table dbo.Tiempo Script Date: 5/15/97 1:00:18 AM \*\*\*\*\*/

```
CREATE TABLE dbo.Tiempo (  
    TiempoId int IDENTITY (1, 1) NOT NULL ,  
    AñoId int NULL ,  
    TrimestreAñoId int NULL ,  
    MesAñoId int NULL ,  
    TiempoDia int NULL
```

)

GO

/\*\*\*\*\* Object: Table dbo.Tramos Script Date: 5/15/97 1:00:18 AM \*\*\*\*\*/

```
CREATE TABLE dbo.Tramos (  
    TramosId int IDENTITY (1, 1) NOT NULL ,  
    Aerolineaid int NOT NULL ,
```

```
        Claseld int NOT NULL ,
        A=old int NOT NULL ,
        TramosCdSalidald int NOT NULL ,
        TramosCdDestinold int NOT NULL ,
        TramosMonto money NULL
    )
GO

/***** Object: Table dbo.TrimestreA=0  Script Date: 5/15/97 1:00:19 AM *****/
CREATE TABLE dbo.TrimestreA=0 (
    TrimestreA=old int IDENTITY (1, 1) NOT NULL ,
    A=old int NOT NULL ,
    TrimestreA=0Descripcion char (5) NOT NULL ,
    TrimestreA=0Monto money NOT NULL
)
GO

/***** Object: Table dbo.Usuario  Script Date: 5/15/97 1:00:19 AM *****/
CREATE TABLE dbo.Usuario (
    UsuarioId int IDENTITY (1, 1) NOT NULL ,
    UsuarioLogin char (12) NOT NULL ,
    UsuarioNombre char (20) NOT NULL ,
    UsuarioApPaterno char (15) NOT NULL ,
    UsuarioApMaterno char (15) NULL ,
    UsuarioTelefono char (10) NOT NULL ,
    UsuarioCelular char (10) NULL ,
    UsuarioContacto char (35) NULL ,
    UsuarioPassword char (8) NOT NULL ,
    UsuarioPrioridad tinyint NULL ,
    UsuarioTS timestamp NOT NULL ,
    UsuarioBorrar binary (1) NULL
)
GO

/***** Object: Table dbo.UsuarioAg  Script Date: 5/15/97 1:00:19 AM *****/
CREATE TABLE dbo.UsuarioAg (
    UsuarioAgId int NOT NULL ,
    UsuarioAgNombre char (20) NOT NULL ,
    UsuarioAgApPaterno char (15) NOT NULL ,
    UsuarioAgApMaterno char (25) NULL ,
    UsuarioAgMonto money NOT NULL ,
    A=old int NOT NULL
)
GO

/***** Object: Table dbo.VentasHotel  Script Date: 5/15/97 1:00:19 AM *****/
CREATE TABLE dbo.VentasHotel (
    VentasHotelId int IDENTITY (1, 1) NOT NULL ,
    HotelId int NULL ,
    HotelDestinold int NULL ,
    UsuarioAgId int NULL ,
    FormasPagold int NULL ,
    Clienteld int NULL ,
    Tiempold int NULL ,
    VentasHotelMonto money NOT NULL
)
GO

/***** Object: Table dbo.VentasPveServicios  Script Date: 5/15/97 1:00:20 AM *****/
CREATE TABLE dbo.VentasPveServicios (
    VentasPveServiciosId int IDENTITY (1, 1) NOT NULL ,
    PveServiciosId int NULL ,
    UsuarioAgId int NULL ,
    FormasPagold int NULL ,
    Clienteld int NULL ,
    Tiempold int NULL ,
    VentasPveServiciosMonto money NOT NULL
)
GO
```

/\*\*\*\*\*\* Object: Table dbo.VentasVuelos Script Date: 5/15/97 1:00:20 AM \*\*\*\*\*/

```
CREATE TABLE dbo.VentasVuelos (
    VentasVuelosId int IDENTITY (1, 1) NOT NULL ,
    TramosId int NULL ,
    UsuarioAgId int NULL ,
    FormasPagoid int NULL ,
    ClientId int NULL ,
    Tiempoid int NULL ,
    VentasVuelosMonto money NOT NULL
)
GO
```

/\*\*\*\*\*\* Object: Stored Procedure dbo.spDelTablas Script Date: 5/15/97 1:00:20 AM \*\*\*\*\*/

```
CREATE PROCEDURE spDelTablas
AS
    TRUNCATE TABLE A#o
    TRUNCATE TABLE Ciudad
    TRUNCATE TABLE Clase
    TRUNCATE TABLE Cliente
    TRUNCATE TABLE FormasPago
    TRUNCATE TABLE Hotel
    TRUNCATE TABLE HotelDestino
    TRUNCATE TABLE MesA#o
/*
    TRUNCATE TABLE Metadato */
    TRUNCATE TABLE PveServicios
    TRUNCATE TABLE Tiempo
    TRUNCATE TABLE Tramos
    TRUNCATE TABLE TrimestreA#o
    TRUNCATE TABLE UsuarioAg
/*TRUNCATE TABLE VentasHotel
    TRUNCATE TABLE VentasPveServicios
    TRUNCATE TABLE VentasVuelos
    TRUNCATE TABLE Aerolinea*/
GO
```

/\*\*\*\*\*\* Object: Stored Procedure dbo.spInsAerolineaA#o Script Date: 5/15/97 1:00:20 AM \*\*\*\*\*/

```
CREATE PROCEDURE spInsAerolineaA#o
AS
    SELECT DATEPART(year,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) Fecha,
        Turismo..PapeletaVuelo.PapeletaVueloPrecioPublico Monto,
        Turismo..Aerolinea.AerolineaDescripcion Descripcion,
        Turismo..Aerolinea.AerolineaAbreviacion Abreviacion,
        Turismo..Aerolinea.Aerolineaid Aerolineaid2
    INTO #Temporal
    FROM Turismo..PapeletaVuelo, Turismo..Aerolinea, Turismo..Papeleta
    WHERE Turismo..PapeletaVuelo.Aerolineaid = Turismo..Aerolinea.Aerolineaid and
        Turismo..PapeletaVuelo.Papeletaid = Turismo..Papeleta.Papeletaid and
        Turismo..Papeleta.PapeletaStatus = '1'

    INSERT INTO Aerolinea(Aerolineaid,AerolineaDescripcion,AerolineaAbreviacion,AerolineaMonto,A#oid)
    SELECT Aerolineaid2,Descripcion,Abreviacion,sum(Monto) Monto,A#oid
    FROM #Temporal,A#o
    WHERE A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.Fecha)
    GROUP BY Aerolineaid2,Descripcion,Abreviacion,A#oid
    ORDER BY Monto DESC
GO
```

/\*\*\*\*\*\* Object: Stored Procedure dbo.spInsCiudad Script Date: 5/15/97 1:00:21 AM \*\*\*\*\*/

```
CREATE PROCEDURE spInsCiudad
AS
    INSERT INTO Ciudad
    SELECT CiudadId,
        CiudadDescripcion
    FROM Turismo..Ciudad
GO
```

```

/***** Object: Stored Procedure dbo.spInsClaseA= Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE spInsClaseA=
AS

```

```

SELECT DATEPART(year,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) Fecha,
        Turismo..PapeletaVuelo.PapeletaVueloPrecioPublico Monto,
        Turismo..Clase.ClaseDescripcion Descripcion,
        Turismo..Clase.ClaseId2 ClaseId2
INTO #Temporal
FROM Turismo..PapeletaVuelo, Turismo..Clase, Turismo..Papeleta
WHERE Turismo..PapeletaVuelo.ClaseId = Turismo..Clase.ClaseId and
        Turismo..PapeletaVuelo.PapeletaId = Turismo..Papeleta.PapeletaId and
        Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO Clase(ClaseId,ClaseDescripcion,ClaseMonto,A=old)
SELECT ClaseId2,Descripcion,sum(Monto) Monto,A=old
FROM #Temporal,A=
WHERE A=A=Descripcion = CONVERT(CHAR(4), #Temporal.Fecha)
GROUP BY ClaseId2,Descripcion,A=old
ORDER BY Monto DESC
GO

```

```

/***** Object: Stored Procedure dbo.spInsClienteA= Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE spInsClienteA=
AS

```

```

SELECT DATEPART(year,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) Fecha,
        Turismo..PapeletaVuelo.PapeletaVueloPrecioPublico Monto,
        Turismo..Cliente.ClienteNombre Nombre,
        Turismo..Cliente.ClienteApPaterno ApPaterno,
        Turismo..Cliente.ClienteApMaterno ApMaterno,
        Turismo..Cliente.ClienteId2 ClienteId2
INTO #Temporal
FROM Turismo..PapeletaVuelo, Turismo..PapeletaCliente, Turismo..Cliente, Turismo..Papeleta
WHERE Turismo..PapeletaVuelo.PapeletaId = Turismo..PapeletaCliente.PapeletaId and
        Turismo..PapeletaCliente.ClienteId = Turismo..Cliente.ClienteId and
        Turismo..PapeletaVuelo.PapeletaId = Turismo..Papeleta.PapeletaId and
        Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATEPART(year,Turismo..PapeletaHotel.PapeletaHotelFeEntrada),
        Turismo..PapeletaHotel.PapeletaHotelPrecioPublico Monto,
        Turismo..Cliente.ClienteNombre Nombre,
        Turismo..Cliente.ClienteApPaterno ApPaterno,
        Turismo..Cliente.ClienteApMaterno ApMaterno,
        Turismo..Cliente.ClienteId2 ClienteId2
FROM Turismo..PapeletaHotel, Turismo..PapeletaCliente, Turismo..Cliente, Turismo..Papeleta
WHERE Turismo..PapeletaHotel.PapeletaId = Turismo..PapeletaCliente.PapeletaId and
        Turismo..PapeletaCliente.ClienteId = Turismo..Cliente.ClienteId and
        Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
        Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATEPART(year,Turismo..PapeletaServicio.PapeletaServicioFeInicio) Fecha,
        Turismo..PapeletaServicio.PapeletaServicioPrecioPublico Monto,
        Turismo..Cliente.ClienteNombre Nombre,
        Turismo..Cliente.ClienteApPaterno ApPaterno,
        Turismo..Cliente.ClienteApMaterno ApMaterno,
        Turismo..Cliente.ClienteId2 ClienteId2
FROM Turismo..PapeletaServicio, Turismo..PapeletaCliente, Turismo..Cliente, Turismo..Papeleta
WHERE Turismo..PapeletaServicio.PapeletaId = Turismo..PapeletaCliente.PapeletaId and
        Turismo..PapeletaCliente.ClienteId = Turismo..Cliente.ClienteId and
        Turismo..PapeletaServicio.PapeletaId = Turismo..Papeleta.PapeletaId and

```

```
Turismo..Papeleta.PapeletaStatus = '1'
```

```
INSERT INTO Cliente(ClienteId,ClienteNombre,ClienteApPaterno,ClienteApMaterno,ClienteMonto,A#old)
SELECT ClienteId2,Nombre,ApPaterno,ApMaterno,sum(Monto) Monto,A#old
FROM #Temporal,A#o
WHERE A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.Fecha)
GROUP BY ClienteId2,Nombre,ApPaterno,ApMaterno,A#old
ORDER BY Monto DESC
GO
```

```
/****** Object: Stored Procedure dbo.splnsFormasPagoA#o Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE splnsFormasPagoA#o
AS
```

```
SELECT DATEPART(year,Turismo..Factura.FacturaFecha) Fecha,
Turismo..Factura.FacturaMonto Monto,
Turismo..FormasPago.FormasPagoDescripcion Descripcion,
Turismo..FormasPago.FormasPagold FormasPagold2
INTO #Temporal
FROM Turismo..Factura, Turismo..FormasPago
WHERE Turismo..Factura.Formapagold = Turismo..FormasPago.Formaspagold
```

```
INSERT INTO FormasPago(FormasPagold,FormasPagoDescripcion,FormasPagoMonto,A#old)
SELECT FormasPagold2,Descripcion,sum(Monto) Monto,A#old
FROM #Temporal,A#o
WHERE A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.Fecha)
GROUP BY FormasPagold2,Descripcion,A#old
ORDER BY Monto DESC
GO
```

```
/****** Object: Stored Procedure dbo.splnsHotelA#o Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE splnsHotelA#o
AS
```

```
SELECT DATEPART(year,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) Fecha,
Turismo..PapeletaHotel.PapeletaHotelPrecioPublico Monto,
Turismo..Hotel.HotelNombre HotelNombre,
Turismo..Hotel.HotelId HotelId2
INTO #Temporal
FROM Turismo..PapeletaHotel, Turismo..Hotel, Turismo..Papeleta
WHERE Turismo..PapeletaHotel.HotelId = Turismo..Hotel.HotelId and
Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
Turismo..Papeleta.PapeletaStatus = '1'
```

```
INSERT INTO Hotel(HotelId,HotelNombre,HotelMonto,A#old)
SELECT HotelId2,HotelNombre,sum(Monto) Monto,A#old
FROM #Temporal,A#o
WHERE A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.Fecha)
GROUP BY HotelId2,HotelNombre,A#old
ORDER BY Monto DESC
GO
```

```
/****** Object: Stored Procedure dbo.splnsHotelDestinoA#o Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE splnsHotelDestinoA#o
AS
```

```
SELECT DATEPART(year,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) Fecha,
Turismo..PapeletaHotel.PapeletaHotelPrecioPublico Monto,
Turismo..Hotel.HotelNombre,
Turismo..Ciudad.CiudadId CiudadId2,
Turismo..Hotel.HotelId HotelId2
INTO #Temporal
FROM Turismo..PapeletaHotel, Turismo..Hotel, Turismo..Ciudad, Turismo..Papeleta
WHERE Turismo..PapeletaHotel.HotelId = Turismo..Hotel.HotelId and
Turismo..PapeletaHotel.CiudadId = Turismo..Ciudad.CiudadId and
Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
Turismo..Papeleta.PapeletaStatus = '1'
```

```

INSERT INTO HotelDestino(HotelId,Añoold,CiudadId,HotelNombre,HotelDestinoMonto)
SELECT HotelId2,Año.Añoold,CiudadId2,HotelNombre,sum(Monto) Monto
FROM #Temporal,Año
WHERE Año.AñoDescripcion = CONVERT(CHAR(4), #Temporal.Fecha)
GROUP BY HotelId2,Añoold,CiudadId2,HotelNombre
ORDER BY Monto DESC
GO

```

```

/***** Object: Stored Procedure dbo.spInsPveServicioAño Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE spInsPveServicioAño
AS

```

```

SELECT DATEPART(year,Turismo..PapeletaServicio.PapeletaServicioFInicio) Fecha,
       Turismo..PapeletaServicio.PapeletaServicioPrecioPublico Monto,
       Turismo..PveServicio.PveServicioNombre Descripcion,
       Turismo..PveServicio.PveServicioId PveServicioId2
INTO #Temporal
FROM Turismo..PapeletaServicio, Turismo..PveServicio, Turismo..Papeleta
WHERE Turismo..PapeletaServicio.PveServicioId = Turismo..PveServicio.PveServicioId and
       Turismo..PapeletaServicio.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO PveServicios(PveServiciosId,PveServiciosNombre,PveServiciosMonto,Añoold)
SELECT PveServicioId2,Descripcion,SUM(Monto)Monto,Año.Añoold
FROM #Temporal,Año
WHERE Año.AñoDescripcion = CONVERT(CHAR(4),#Temporal.Fecha)
GROUP BY PveServicioId2,Descripcion,Año.Añoold
ORDER BY Monto DESC
GO

```

```

/***** Object: Stored Procedure dbo.spInsTablas Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE spInsTablas
AS

```

```

EXECUTE spInsVtasAño
EXECUTE spInsVtasMesAño
EXECUTE spInsVtasTriAño
EXECUTE spInsAerolineaAño
EXECUTE spInsCiudad
EXECUTE spInsClaseAño
EXECUTE spInsClienteAño
EXECUTE spInsFormaspagoAño
EXECUTE spInsHotelAño
EXECUTE spInsHotelDestinoAño
EXECUTE spInsPveServicioAño
EXECUTE spInsTramosAño
EXECUTE spInsUsuarioAgAño
EXECUTE spInsTiempo
EXECUTE spInsVentasHotel
EXECUTE spInsVentasServicios
EXECUTE spInsVentasVuelos
GO

```

```

/***** Object: Stored Procedure dbo.spInsTiempo Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE spInsTiempo
AS

```

```

SELECT DATEPART(year,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) FechaYY,
       DATENAME(quarter,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) FechaQTR,
       DATEPART(month,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) FechaMM,
       DATEPART(day,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) Fechadd
INTO #Temporal
FROM Turismo..PapeletaVuelo,Turismo..Papeleta
WHERE Turismo..Papeleta.PapeletaId = Turismo..PapeletaVuelo.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal

```

```

SELECT DATEPART(year,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) FechaYY,
       DATENAME(quarter,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) FechaQTR,
       DATEPART(month,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) FechaMM,
       DATEPART(day,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) Fechadd
FROM   Turismo..PapeletaHotel,Turismo..Papeleta
WHERE  Turismo..Papeleta.PapeletaId = Turismo..PapeletaHotel.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATEPART(year,Turismo..PapeletaServicio.PapeletaServicioFeInicio) FechaYY,
       DATENAME(quarter,Turismo..PapeletaServicio.PapeletaServicioFeInicio) FechaQTR,
       DATEPART(month,Turismo..PapeletaServicio.PapeletaServicioFeInicio) FechaMM,
       DATEPART(day,Turismo..PapeletaServicio.PapeletaServicioFeInicio) Fechadd
FROM   Turismo..PapeletaServicio,Turismo..Papeleta
WHERE  Turismo..Papeleta.PapeletaId = Turismo..PapeletaServicio.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO Tiempo(A#old,TrimestreA#oid,MesA#oid,TiempoDia)
SELECT A#o.A#old, TrimestreA#o.TrimestreA#old, MesA#o.MesA#old, FechaDD
FROM   #Temporal,A#o,TrimestreA#o,MesA#o
WHERE  A#o.A#oDescripcion = CONVERT(CHAR(4),#Temporal.FechaYY) and
       TrimestreA#o.A#old = A#o.A#old and
       MesA#o.A#old = A#o.A#old and
       TrimestreA#o.TrimestreA#oDescripcion = #Temporal.FechaQTR and
       MesA#o.MesId = FechaMM
GROUP BY A#o.A#old,TrimestreA#o.TrimestreA#old,MesA#o.MesA#old, FechaDD
GO

```

/\*\*\*\*\* Object: Stored Procedure dbo.splnsTramosA#o Script Date: 5/15/97 1:00:21 AM \*\*\*\*\*/

```

CREATE PROCEDURE splnsTramosA#o
AS
SELECT DATEPART(year,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) Fecha,
       Turismo..PapeletaVuelo.PapeletaVueloPrecioPublico Monto,
       Turismo..PapeletaVuelo.PapeletaId PapeletaId2,
       Turismo..PapeletaVuelo.PapeletaVueloCdSalida CdSalida,
       Turismo..PapeletaVuelo.PapeletaVueloCdDestino CdDestino,
       Aerolinea.AerolineId AerolineId2,
       Clase.ClaseId ClaseId2

```

```

INTO #Temporal
FROM   Turismo..PapeletaVuelo, Turismo..Aerolinea,Aerolinea,Clase
WHERE  Turismo..PapeletaVuelo.AerolineId = Turismo..Aerolinea.AerolineId and
       Turismo..PapeletaVuelo.AerolineId = Aerolinea.AerolineId and
       Turismo..PapeletaVuelo.ClaseId = Clase.ClaseId

```

```

INSERT INTO Tramos(AerolineId,ClaseId,A#old,TramosCdSalida,TramosCdDestino,TramosMonto)
SELECT AerolineId2,ClaseId2,A#old,CdSalida,CdDestino,sum(Monto) Monto
FROM   #Temporal,A#o
WHERE  A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.Fecha)
GROUP BY AerolineId2,ClaseId2,A#old,CdSalida,CdDestino
ORDER BY Monto DESC
GO

```

/\*\*\*\*\* Object: Stored Procedure dbo.splnsUsuarioAgA#o Script Date: 5/15/97 1:00:21 AM \*\*\*\*\*/

```

CREATE PROCEDURE splnsUsuarioA#o
AS
SELECT DATEPART(year,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) Fecha,
       Turismo..PapeletaVuelo.PapeletaVueloPrecioPublico Monto,
       Turismo..Usuario.UsuarioNombre Nombre,
       Turismo..Usuario.UsuarioApPaterno ApPaterno,
       Turismo..Usuario.UsuarioApMaterno ApMaterno,
       Turismo..Usuario.UsuarioId UsuarioId2
INTO #Temporal
FROM   Turismo..PapeletaVuelo, Turismo..Papeleta, Turismo..Usuario
WHERE  Turismo..PapeletaVuelo.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.UsuarioId = Turismo..Usuario.UsuarioId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATEPART(year,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) Fecha,
       Turismo..PapeletaHotel.PapeletaHotelPrecioPublico Monto,
       Turismo..Usuario.UsuarioNombre Nombre,
       Turismo..Usuario.UsuarioApPaterno ApPaterno,
       Turismo..Usuario.UsuarioApMaterno ApMaterno,
       Turismo..Usuario.UsuarioId Usuariold2
FROM   Turismo..PapeletaHotel, Turismo..Papeleta, Turismo..Usuario
WHERE  Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.UsuarioId = Turismo..Usuario.UsuarioId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATEPART(year,Turismo..PapeletaServicio.PapeletaServicioFeInicio) Fecha,
       Turismo..PapeletaServicio.PapeletaServicioPrecioPublico Monto,
       Turismo..Usuario.UsuarioNombre Nombre,
       Turismo..Usuario.UsuarioApPaterno ApPaterno,
       Turismo..Usuario.UsuarioApMaterno ApMaterno,
       Turismo..Usuario.UsuarioId Usuariold2
FROM   Turismo..PapeletaServicio, Turismo..Papeleta, Turismo..Usuario
WHERE  Turismo..PapeletaServicio.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.UsuarioId = Turismo..Usuario.UsuarioId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO
UsuarioAg(UsuarioAgId,UsuarioAgNombre,UsuarioAgApPaterno,UsuarioAgApMaterno,UsuarioAgMonto,A#old)
SELECT Usuariold2,Nombre, ApPaterno, ApMaterno, sum(Monto) Monto,A#old
FROM   #Temporal,A#o
WHERE  A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.Fecha)
GROUP BY Usuariold2,Nombre,ApPaterno,ApMaterno,A#old
ORDER BY Monto DESC
GO

```

/\*\*\*\*\*\* Object: Stored Procedure dbo.splnsVentasHotel Script Date: 5/15/97 1:00:21 AM \*\*\*\*\*/

```

CREATE PROCEDURE splnsVentasHotel
AS

```

```

TRUNCATE TABLE VentasHotel

```

```

SELECT DATEPART(day,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) FechaDD,
       DATEPART(month,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) FechaMM,
       DATEPART(quarter,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) FechaQTR,
       DATEPART(year,Turismo..PapeletaHotel.PapeletaHotelFeEntrada) FechaYY,
       Turismo..Hotel.HotelId Hotelld2,
       Turismo..Usuario.UsuarioId Usuariold2,
       Turismo..FormasPago.FormasPagold FormasPagold2,
       Turismo..Cliente.Clienteld Clienteld2,
       Turismo..PapeletaHotel.PapeletaHotelPrecioPublico Monto

```

```

INTO #Temporal
FROM   Turismo..PapeletaHotel, Turismo..Hotel, Turismo..Papeleta,Turismo..Usuario,
       Turismo..FormasPago,Turismo..Cliente,Turismo..PapeletaCliente,Turismo..Factura
WHERE  Turismo..Usuario.UsuarioId = Turismo..Papeleta.UsuarioId and
       Turismo..Cliente.Clienteld = Turismo..PapeletaCliente.Clienteld and
       Turismo..PapeletaCliente.PapeletaId =Turismo..Papeleta.PapeletaId and
       Turismo..PapeletaCliente.Clienteld = Turismo..Factura.Clienteld and
       Turismo..Factura.FormaPagold = Turismo..FormasPago.FormasPagold and
       Turismo..PapeletaHotel.HotelId = Turismo..Hotel.Hotelld and
       Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO VentasHotel(Hotelld,HotelDestinold,UsuarioAgId,FormasPagold,Clienteld,Tiempold,VentasHotelMonto)
SELECT
       #Temporal.Hotelld2,HotelDestino.HotelDestinold,#Temporal.Usuariold2,#Temporal.FormasPagold2,#Temporal
       .Clienteld2,Tiempo.Tiempold,#Temporal.Monto

```

```

FROM #Temporal.A#o,TrimestreA#o,MesA#o,Tiempo,HotelDestino
WHERE #Temporal.HotelId2 = HotelDestino.HotelId and
      A#o.A#oDescripcion = CONVERT(CHAR(4),#Temporal.FechaYY) and
      A#o.A#old = Tiempo.A#old and
      TrimestreA#o.TrimestreA#oDescripcion = CONVERT(CHAR(1), #Temporal.FechaQTR) and
      TrimestreA#o.TrimestreA#old = Tiempo.TrimestreA#old and
      MesA#o.MesId = #Temporal.FechaMM and
      MesA#o.MesA#old = Tiempo.MesA#old and
      Tiempo.TiempoDia = #Temporal.FechaDD
GROUP BY HotelId2,HotelDestinoId,Usuariold2,FormasPagold2,Clienteld2,Tiempo.Tiempold,Monto
ORDER BY Monto DESC
GO

```

```

/***** Object: Stored Procedure dbo.splnsVentasServicios  Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE splnsVentasServicios
AS

```

```

TRUNCATE TABLE VentasPveServicios

```

```

SELECT DATEPART(day,Turismo..PapeletaServicio.PapeletaServicioFeInicio) FechaDD,
       DATEPART(month,Turismo..PapeletaServicio.PapeletaServicioFeInicio) FechaMM,
       DATEPART(quarter,Turismo..PapeletaServicio.PapeletaServicioFeInicio) FechaQTR,
       DATEPART(year,Turismo..PapeletaServicio.PapeletaServicioFeInicio) FechaYY,
       Turismo..PveServicio.PveServiciold PveServiciold2,
       Turismo..Usuario.UsuarioId Usuariold2,
       /*Turismo..FormasPago.FormasPagold FormasPagold2,*/
       Turismo..Cliente.Clienteld Clienteld2,
       Turismo..PapeletaServicio.PapeletaServicioPrecioPublico Monto

```

```

INTO #Temporal

```

```

FROM Turismo..PapeletaServicio, Turismo..PveServicio, Turismo..Papeleta,Turismo..Usuario,
     Turismo..FormasPago,Turismo..Cliente,Turismo..PapeletaCliente,Turismo..Factura
WHERE Turismo..Usuario.UsuarioId = Turismo..Papeleta.UsuarioId and
     Turismo..Cliente.Clienteld = Turismo..PapeletaCliente.Clienteld and
     Turismo..PapeletaCliente.PapeletaId = Turismo..Papeleta.PapeletaId and
     /*Turismo..PapeletaCliente.Clienteld = Turismo..Factura.Clienteld and*/
     /*Turismo..Factura.FormaPagold = Turismo..FormasPago.FormasPagold and*/
     Turismo..PapeletaServicio.PveServiciosId = Turismo..PveServicio.PveServiciold and
     Turismo..PapeletaServicio.PapeletaId = Turismo..Papeleta.PapeletaId and
     Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO

```

```

VentasPveServicios(PveServiciosId,UsuarioAgld,/*FormasPagold,*/Clienteld,Tiempold,VentasPveServiciosMonto)
SELECT PveServiciold2,Usuariold2,/*FormasPagold2,*/Clienteld2,Tiempo.Tiempold,Monto
FROM #Temporal.A#o,TrimestreA#o,MesA#o,Tiempo
WHERE A#o.A#oDescripcion = CONVERT(CHAR(4),#Temporal.FechaYY) and
      A#o.A#old = Tiempo.A#old and
      TrimestreA#o.TrimestreA#oDescripcion = CONVERT(CHAR(1), #Temporal.FechaQTR) and
      TrimestreA#o.TrimestreA#old = Tiempo.TrimestreA#old and
      MesA#o.MesId = #Temporal.FechaMM and
      MesA#o.MesA#old = Tiempo.MesA#old and
      Tiempo.TiempoDia = #Temporal.FechaDD
GROUP BY PveServiciold2,Usuariold2,/*FormasPagold2,*/Clienteld2,Tiempo.Tiempold,Monto
ORDER BY Monto DESC
GO

```

```

/***** Object: Stored Procedure dbo.splnsVentasVuelos  Script Date: 5/15/97 1:00:21 AM *****/
CREATE PROCEDURE splnsVentasVuelos
AS

```

```

TRUNCATE TABLE VentasVuelos

```

```

SELECT DATEPART(day,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) FechaDD,
       DATEPART(month,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) FechaMM,
       DATEPART(quarter,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) FechaQTR,
       DATEPART(year,Turismo..PapeletaVuelo.PapeletaVueloFeSalida) FechaYY,
       Turismo..Aereolinea.Aereolineald Aereolineald2,
       Turismo..Usuario.UsuarioId Usuariold2,
       Turismo..FormasPago.FormasPagold FormasPagold2,

```

```

Turismo..Cliente.Clienteld Clienteld2,
Turismo..PapeletaVuelo.PapeletaVueloPrecioPublico Monto

INTO #Temporal
FROM Turismo..PapeletaVuelo, Turismo..Aerolinea, Turismo..Papeleta, Turismo..Usuario,
Turismo..FormasPago, Turismo..Cliente, Turismo..PapeletaCliente, Turismo..Factura

WHERE Turismo..Usuario.UsuarioId = Turismo..Papeleta.UsuarioId and
Turismo..Cliente.Clienteld = Turismo..PapeletaCliente.Clienteld and
Turismo..PapeletaCliente.PapeletaId = Turismo..Papeleta.PapeletaId and
Turismo..PapeletaCliente.Clienteld = Turismo..Factura.Clienteld and
Turismo..Factura.FormaPagold = Turismo..FormasPago.FormasPagold and
Turismo..PapeletaVuelo.Aerolineald = Turismo..Aerolinea.Aerolineald and
Turismo..PapeletaVuelo.PapeletaId = Turismo..Papeleta.PapeletaId and
Turismo..Papeleta.PapeletaStatus = '1'

INSERT INTO VentasVuelos(TramosId, UsuarioId, FormasPagold, Clienteld, Tiempold, VentasVuelosMonto)
SELECT TramosId, UsuarioId2, FormasPagold2, Clienteld2, Tiempo, Tiempold, Monto
FROM #Temporal, A#o, TrimestreA#o, MesA#o, Tiempo, Tramos
WHERE #Temporal.Aerolineald2 = Tramos.Aerolineald and
A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.FechaYY) and
A#o.A#old = Tiempo.A#old and
TrimestreA#o.TrimestreA#oDescripcion = CONVERT(CHAR(1), #Temporal.FechaQTR) and
TrimestreA#o.TrimestreA#old = Tiempo.TrimestreA#old and
MesA#o.MesId = #Temporal.FechaMM and
MesA#o.MesA#old = Tiempo.MesA#old and
Tiempo.TiempoDia = #Temporal.FechaDD
GROUP BY TramosId, UsuarioId2, FormasPagold2, Clienteld2, Tiempo, Tiempold, Monto
ORDER BY Monto DESC
GO

/***** Object: Stored Procedure dbo.spInsVtasA#o Script Date: 5/15/97 1:00:22 AM *****/
CREATE PROCEDURE spInsVtasA#o
AS

SELECT DATEPART(year, PapeletaVueloFeSalida) Fecha,
PapeletaVueloPrecioPublico Monto
INTO #Temporal
FROM Turismo..PapeletaVuelo, Turismo..Papeleta
WHERE Turismo..PapeletaVuelo.PapeletaId = Turismo..Papeleta.PapeletaId and
Turismo..Papeleta.PapeletaStatus = '1'

INSERT INTO #Temporal
SELECT DATEPART(year, PapeletaHotelFeEntrada) Fecha,
PapeletaHotelPrecioPublico Monto
FROM Turismo..PapeletaHotel, Turismo..Papeleta
WHERE Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
Turismo..Papeleta.PapeletaStatus = '1'

INSERT INTO #Temporal
SELECT DATEPART(year, PapeletaServicioFelnicio) Fecha,
PapeletaServicioPrecioPublico Monto
FROM Turismo..PapeletaServicio, Turismo..Papeleta
WHERE Turismo..PapeletaServicio.PapeletaId = Turismo..Papeleta.PapeletaId and
Turismo..Papeleta.PapeletaStatus = '1'

INSERT INTO A#o (A#oDescripcion, A#oMonto)
SELECT CONVERT(char(4), Fecha), SUM(Monto) Monto
FROM #Temporal
GROUP BY Fecha
GO

/***** Object: Stored Procedure dbo.spInsVtasMesA#o Script Date: 5/15/97 1:00:22 AM *****/
CREATE PROCEDURE spInsVtasMesA#o
AS

```

```

SELECT DATEPART(month,PapeletaVueloFeSalida) Mes,
       DATEPART(year,PapeletaVueloFeSalida) Año,
       DATENAME(month,PapeletaVueloFeSalida) Descripcion,
       PapeletaVueloPrecioPublico Monto
INTO #Temporal
FROM Turismo..papeletaVuelo, Turismo..Papeleta
WHERE Turismo..PapeletaVuelo.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATEPART(month,PapeletaHotelFeEntrada) Mes,
       DATEPART(year,PapeletaHotelFeEntrada) Año,
       DATENAME(month,PapeletaHotelFeEntrada) Descripcion,
       PapeletaHotelPrecioPublico Monto
FROM Turismo..papeletaHotel, Turismo..Papeleta
WHERE Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATEPART(month,PapeletaServicioFeInicio) Mes,
       DATEPART(year,PapeletaServicioFeInicio) Año,
       DATENAME(month,PapeletaServicioFeInicio) Descripcion,
       PapeletaServicioPrecioPublico Monto
FROM Turismo..papeletaServicio, Turismo..Papeleta
WHERE Turismo..PapeletaServicio.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO MesAño(MesId,AñoId,MesAñoDescripcion,MesAñoMonto)
SELECT Mes,AñoId,Descripcion,sum(Monto) Monto
FROM #Temporal,Año
WHERE Año.AñoDescripcion = CONVERT(CHAR(4), #Temporal.Año)
GROUP BY Mes,AñoId,Descripcion
GO

```

```

/***** Object: Stored Procedure dbo.splnsVtasTriAño  Script Date: 5/15/97 1:00:22 AM *****/
CREATE PROCEDURE splnsVtasTriAño

```

AS

```

SELECT DATENAME(quarter,PapeletaVueloFeSalida) Trimestre,
       DATEPART(year,PapeletaVueloFeSalida) Año,
       PapeletaVueloPrecioPublico Monto
INTO #Temporal
FROM Turismo..papeletaVuelo, Turismo..Papeleta
WHERE Turismo..PapeletaVuelo.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATENAME(quarter,PapeletaHotelFeEntrada) Trimestre,
       DATEPART(year,PapeletaHotelFeEntrada) Año,
       PapeletaHotelPrecioPublico Monto
FROM Turismo..papeletaHotel, Turismo..Papeleta
WHERE Turismo..PapeletaHotel.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO #Temporal
SELECT DATENAME(quarter,PapeletaServicioFeInicio) Trimestre,
       DATEPART(year,PapeletaServicioFeInicio) Año,
       PapeletaServicioPrecioPublico Monto
FROM Turismo..papeletaServicio, Turismo..Papeleta
WHERE Turismo..PapeletaServicio.PapeletaId = Turismo..Papeleta.PapeletaId and
       Turismo..Papeleta.PapeletaStatus = '1'

```

```

INSERT INTO TrimestreAño(AñoId,TrimestreAñoDescripcion,TrimestreAñoMonto)

```

```

SELECT A#old,Trimestre,sum(Monto) Monto
FROM #Temporal,A#o
WHERE A#o.A#oDescripcion = CONVERT(CHAR(4), #Temporal.A#o)
GROUP BY A#old,Trimestre
GO

```

## Sistema de Condensación de Información – Front-End.

'definición de variables

'llamado de store procedures segun lo seleccionado

```

If chkvtasglobales.Value = False And chkvtashotel.Value = False And chkvtasservicios.Value = False And
chkvtasvuelos.Value = False Then

```

```

    MsgBox "Para poder agregar se necesita seleccionar una opcion al menos.", 48, "SCI"

```

```
End If
```

```

If chkvtasglobales.Value = True Then

```

```

    Basicas
    Dimensiones

```

```
End If
```

```

If chkvtashotel.Value = True Then

```

```

    FactTableHotel

```

```
End If
```

```

If chkvtasservicios.Value = True Then

```

```

    FactTableServicios

```

```
End If
```

```

If chkvtasvuelos.Value = True Then

```

```

    FactTableVuelos

```

```
End If
```

```

MsgBox "Se ha cargado el Data Warehouse", 48, "SCI"

```

```
End Sub
```

```

Dim spDelTablas As New CSQLQuery
Dim splnsVtasAño As New CSQLQuery
Dim splnsVtasTriAño As New CSQLQuery
Dim splnsVtasMesAño As New CSQLQuery
Dim splnsCiudad As New CSQLQuery
Dim splnsTiempo As New CSQLQuery

```

```
Dim result As Boolean
```

'Store Procedures del llenado basico

```
Screen.MousePointer = 11
```

```

spDelTablas.ClearQuery
spDelTablas.SP = "spDelTablas"
result = spDelTablas.Execute(giConn)

```

```

splnsCiudad.ClearQuery
splnsCiudad.SP = "splnsCiudad"
result = splnsCiudad.Execute(giConn)

```

```

splnsVtasAño.ClearQuery
splnsVtasAño.SP = "splnsVtasAño"

```

```
result = splnsVtasAño.Execute(giConn)
```

```
splnsVtasTriAño.ClearQuery  
splnsVtasTriAño.SP = "splnsVtasTriAño"  
result = splnsVtasTriAño.Execute(giConn)
```

```
splnsVtasMesAño.ClearQuery  
splnsVtasMesAño.SP = "splnsVtasMesAño"  
result = splnsVtasMesAño.Execute(giConn)
```

```
splnsTiempo.ClearQuery  
splnsTiempo.SP = "splnsTiempo"  
result = splnsTiempo.Execute(giConn)
```

```
Screen.MousePointer = 0
```

```
End Sub
```

```
Dim splnsAerolineaAño As New CSQLQuery  
Dim splnsClaseAño As New CSQLQuery  
Dim splnsClienteAño As New CSQLQuery  
Dim splnsFormasPagoAño As New CSQLQuery  
Dim splnsHotelAño As New CSQLQuery  
Dim splnsHotelDestinoAño As New CSQLQuery  
Dim splnsPveServicioAño As New CSQLQuery  
Dim splnsTramosAño As New CSQLQuery  
Dim splnsUsuarioAgAño As New CSQLQuery
```

```
Dim result As Boolean
```

```
'Store Procedures para llenar dimensiones
```

```
Screen.MousePointer = 11
```

```
splnsAerolineaAño.ClearQuery  
splnsAerolineaAño.SP = "splnsAerolineaAño"  
result = splnsAerolineaAño.Execute(giConn)
```

```
splnsClaseAño.ClearQuery  
splnsClaseAño.SP = "splnsClaseAño"  
result = splnsClaseAño.Execute(giConn)
```

```
splnsClienteAño.ClearQuery  
splnsClienteAño.SP = "splnsClienteAño"  
result = splnsClienteAño.Execute(giConn)
```

```
splnsFormasPagoAño.ClearQuery  
splnsFormasPagoAño.SP = "splnsFormasPagoAño"  
result = splnsFormasPagoAño.Execute(giConn)
```

```
splnsHotelAño.ClearQuery  
splnsHotelAño.SP = "splnsHotelAño"  
result = splnsHotelAño.Execute(giConn)
```

```
splnsHotelDestinoAño.ClearQuery  
splnsHotelDestinoAño.SP = "splnsHotelDestinoAño"  
result = splnsHotelDestinoAño.Execute(giConn)
```

```
splnsPveServicioAño.ClearQuery  
splnsPveServicioAño.SP = "splnsPveServicioAño"  
result = splnsPveServicioAño.Execute(giConn)
```

```
splnsTramosAño.ClearQuery  
splnsTramosAño.SP = "splnsTramosAño"  
result = splnsTramosAño.Execute(giConn)
```

```
splnsUsuarioAgAño.ClearQuery  
splnsUsuarioAgAño.SP = "splnsUsuarioAgAño"
```

```
result = splnsUsuarioAgAño.Execute(giConn)
```

```
Screen.MousePointer = 0
End Sub
```

```
Dim splnsVentasHotel As New CSQuery
Dim result As Boolean
```

```
Screen.MousePointer = 11
```

```
splnsVentasHotel.ClearQuery
splnsVentasHotel.SP = "splnsVentasHotel"
result = splnsVentasHotel.Execute(giConn)
```

```
Screen.MousePointer = 0
End Sub
```

```
Dim splnsVentasServicios As New CSQuery
Dim result As Boolean
```

```
Screen.MousePointer = 11
```

```
splnsVentasServicios.ClearQuery
splnsVentasServicios.SP = "splnsVentasServicios"
result = splnsVentasServicios.Execute(giConn)
```

```
Screen.MousePointer = 0
End Sub
```

```
Dim splnsVentasVuelos As New CSQuery
Dim result As Boolean
```

```
Screen.MousePointer = 11
```

```
splnsVentasVuelos.ClearQuery
splnsVentasVuelos.SP = "splnsVentasVuelos"
result = splnsVentasVuelos.Execute(giConn)
```

```
Screen.MousePointer = 0
End Sub
```

## Sistema de Información Ejecutiva de Ventas y Operaciones – Front-End.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>Home Page ABATURISMO</title>
</head>

<body background="Gray_DottedA025.gif" bgcolor="#FFFFFF">

<table border="0" width="700">
<tr>
<td valign="top" width="25"><p></p>
<p></p>
<p><a href="informac.htm"></a></p>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaTurismo/Noticias.dbm"
method="POST" name="formnt" target="_top">
```

```

        <p><input type="image" name="I1" src="MenuNI.gif"
        align="bottom" border="0" width="170" height="42"></p>
    </form>
    <p><a href="director.htm"></a></p>
    </td>
    <td>&nbsp;</td>
    <td valign="top" width="50%"><p align="center"><font
    size="4" face="Arial"><strong>Bienvenidos al Home Page de
    la empresa ABA|TURISMO</strong></font></p>
    <p><font
    size="2" face="Arial">ABA|TURISMO es una agencia de
    viajes la cual ofrece una diversidad de productos entre
    los que se encuentran: las reservaciones en vuelos,
    habitaciones de hotel, paquetes para eventos especiales y
    paquetes turísticos.</font></p>
    <p><font size="2" face="Arial">Esta página tiene como
    fin brindar a los usuarios </font><a href="informac.htm"
    target="Información Financiera"><font size="2"
    face="Arial">INFORMACION FINANCIERA</font></a><font
    size="2" face="Arial">, NOTICIAS INTERNAS y TENDENCIA DE
    PRODUCTOS de manera oportuna y confiable. Así como un </font><a
    href="director.htm"><font size="2" face="Arial">DIRECTORIO</font></a><font
    size="2" face="Arial"> de los servicios.</font></p>
    </td>
    <td>&nbsp;</td>
    <td valign="top" width="30%"></td>
</tr>
</table>

<p>&nbsp;</p>

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

!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML/EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>INFORMACION FINANCIERA</title>
</head>

<body background="Gray_DottedA025.gif" bgcolor="#FFFFFF">

<table border="0" width="760">
    <tr>
        <td width="5%"></td>
        <td>&nbsp;</td>
    </tr>
</table>

<table border="0">
    <tr>
        <td valign="top"><form
        action="/Scripts/dbml.exe?Template=/IntranetAbaTurismo/SeleccionAerolinea.dbm"
        method="POST" name="formae" target="_top">
            <p><input type="image" name="I1" src="informae.gif"

```

```

        align="bottom" border="0" width="170" height="42"></p>
</form>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaturismo/SeleccionAerolineaDestino.dbm"
method="POST" name="formaed" target="_top">
  <p><input type="image" name="I1" src="informaed.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaturismo/SeleccionUsuario.dbm"
method="POST" name="formag" target="_top">
  <p><input type="image" name="I1" src="informag.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaTurismo/SeleccionCliente.dbm"
method="POST" name="formacte" target="_top">
  <p><input type="image" name="I1" src="informct.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaturismo/SeleccionFormasPago.dbm"
method="POST" name="formfo" target="_top">
  <p><input type="image" name="I1" src="informfo.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
</td>
<td valign="top" width="40%"><p><a
href="ventasMeClientes.htm"></a></p>
<p></p>
<p><a href="ventasGlobales.htm"></a></p>
</td>
<td valign="top"><p><a href="ventasMeDestinos.htm"></a></p>
<p></p>
</td>
<td valign="top" width="60%"><form
action="/Scripts/dbml.exe?Template=/IntranetAbaturismo/SeleccionHotel"
method="POST" name="formho" target="_top">
  <p><input type="image" name="I1" src="informho.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaturismo/SeleccionHotelDestino"
method="POST" name="formhod" target="_top">
  <p><input type="image" name="I1" src="informhod.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaturismo/SeleccionPveServicios"
method="POST" name="formpv" target="_top">
  <p><input type="image" name="I1" src="informpv.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
<form
action="/Scripts/dbml.exe?Template=/IntranetAbaturismo/SeleccionClase.dbm"
method="POST" name="formcl" target="_top">
  <p><input type="image" name="I1" src="informcl.gif"
    align="bottom" border="0" width="170" height="42"></p>
</form>
</td>
</tr>
</table>
</body>
</html>

```

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>DIRECTORIO</title>
</head>

<body background="Gray_DottedA025.gif" bgcolor="#FFFFFF">

<table border="0">
<tr>
<td></td>
<td width="60%"></td>
<td></td>
</tr>
</table>

<p> </p>

<form
action="/Scripts/dbml.exe?Template=/IntranetAbaTurismo/GridDirectorio.dbm"
method="POST" name="Formdir" target="_top">
<hr>
<p><font face="Arial">Seleccione el objeto que necesite.
Teclée el nombre y/o apellido de la persona o teclee el
nombre del objeto que busca y presiona el botón para buscar. <br>
<br>
Opción: <select name="cboDirectorioOp" size="1">
<option selected value="0"> Cliente </option>
<option value="1"> Agente de Ventas </option>
<option value="2"> Aereolínea </option>
<option value="3"> Hotel </option>
<option value="4"> Proveedor de Servicios </option>
</select> </font></p>
<p><font face="Arial">Nombre: <input type="text" size="20"
name="txtDirectorioNombre"> </font><input type="submit"
name="cmdbuscar" value="Buscar" language="VBScript"> </p>
</form>

<hr>

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

<html>

<title>NOTICIAS INTERNAS</title>

<body background="http://troy/AbaTurismo/Gray_DottedA025.gif" bgcolor="#FFFFFF">

<table border="0">
<tr>
<td align="center"><p align="center"><font size="4"
face="Arial"></font></p>
</td>
</tr>
</table>

<form
action="/Scripts/dbml.exe?Template=/IntranetAbaTurismo/DespliegaNoticia.dbm"

```

```

method="POST" name="formnota" target="_top">
<table border="0" width="760" bordercolor="#000000">
  <tr>
    <td valign="top"><hr>
      <p><font face="Arial">Fecha :</font><font size="2"
face="Arial"><br>
</font><font face="Arial"><select
name="cboNoticiaFecha" size="1">
          <option> 1997-04-15 00:00:00.000
          <option> 1997-04-11 00:00:00.000
          <option> 1997-04-23 00:00:00.000
        </select></font></p>
      <p><font face="Arial">Número de Noticia:</font><font
size="2" face="Arial"><br>
</font><select name="cboNoticiaNum" size="1">
          <option> 1
          <option> 2
          <option> 3
        </select> </p>
      <hr>
      <p><input type="submit" name="cmdBuscar"
value="Buscar"></p>
    </td>
  </tr>
</table>
</form>
</body>
</html>

```

```

<P><HR><B>Queries</B><P><PRE>
<B>COMBONOTICIAFECHA</B> (Records=3, Time=110ms)
SQL = SELECT Boletin.BoletinInicio
      FROM Boletin

<B>COMBONOTICIANUM</B> (Records=3, Time=82ms)
SQL = SELECT Boletin.BoletinId
      FROM Boletin
</PRE>

```

## Sistema de Información Ejecutiva de Ventas y Operaciones – Back-End.

```

!-Query para extraer las ventas Anuales de todos los clientes)->
<DBIF #Form.cboaerolinea# is "Todos">
  <DBIF #form.cboperiodicidad# is "Anual">
    <DBQUERY NAME="vtasaerolinea" DATASOURCE="DataW1"
      SQL=" SELECT
Aerolinea.Aerolineaid,Aerolinea.AerolineaDescripcion,Aerolinea.AerolineaMonto,Año.AñoDescripcion
      FROM Aerolinea, Año
      WHERE Aerolinea.AñoId = Año.AñoId and
      CONVERT(int(4), Año.AñoDescripcion) >= #form.cboanioinicio# and
      CONVERT(int(4), Año.AñoDescripcion) <= #form.cboaniofinal#
      ORDER BY Aerolinea.Aerolineaid">
    </DBIF>

<!--Query para extraer las ventas Anuales de un cliente)->
<DBELSE>
  <DBQUERY NAME="vtasaerolinea" DATASOURCE="DataW1"

```

```

SQL=" SELECT
Aerolinea.Aerolineald,Aerolinea.AerolineaDescripcion,Aerolinea.AerolineaMonto,Año.AñoDescripcion
      FROM Aerolinea, Año
      WHERE Aerolinea.Añoold = Año.Añoold and
            Aerolinea.AerolineaDescripcion= '#form.cboaerolinea#' and
            CONVERT(int(4), Año.AñoDescripcion) >= #form.cboanioinicio# and
            CONVERT(int(4), Año.AñoDescripcion) <= #form.cboaniofinal# ">

```

</DBIF>

<!--Query para extraer las ventas Mensuales de todos los pve. servicios-->

<DBIF #Form.cboaerolinea# is "Todos">

<DBIF #form.cboperiodicidad# is "Mensual">

<DBQUERY NAME="vtasaerolinea" DATASOURCE="DataW1"

```

SQL=" SELECT Aerolinea.AerolineaDescripcion,
            SUM(VentasVuelos.VentasVuelosMonto) AerolineaMonto,
            MesAño.MesId

```

```

      FROM Tiempo,Aerolinea,VentasVuelos,MesAño,Tramos
      WHERE Tramos.TramosId = VentasVuelos.TramosId and
            Aerolinea.Aerolineald = Tramos.Aerolineald and
            Tiempo.Tiempold = VentasVuelos.Tiempold and
            Tiempo.MesAñoold =MesAño.MesAñoold and
            mesaño.mesid >= #form.cboMesInicio# and
            mesaño.mesid <= #form.cboMesFinal#
      Group BY Aerolinea.AerolineaDescripcion,MesAño.mesid">

```

</DBIF>

<DBELSE>

<DBIF #form.cboperiodicidad# is "Mensual">

<DBQUERY NAME="vtasaerolinea" DATASOURCE="DataW1"

```

SQL=" SELECT Aerolinea.AerolineaDescripcion,
            SUM(VentasVuelos.VentasVuelosMonto) AerolineaMonto,
            MesAño.MesId

```

```

      FROM Tiempo,Aerolinea,VentasVuelos,MesAño,Tramos
      WHERE Tramos.TramosId = VentasVuelos.TramosId and
            Aerolinea.Aerolineald = Tramos.Aerolineald and
            Tiempo.Tiempold = VentasVuelos.Tiempold and
            Tiempo.MesAñoold =MesAño.MesAñoold and
            mesaño.mesid >= #form.cboMesInicio# and
            mesaño.mesid <= #form.cboMesFinal# and
            Aerolinea.AerolineaDescripcion = '#form.cboaerolinea#'

```

```

      Group BY Aerolinea.aerolineaDescripcion,MesAño.mesid">

```

</DBIF>

</DBIF>

<html>

<title>Ventas por Aerolinea</title>

<body background="http://troy/Abaturismo/Gray\_DottedA025.gif" bgcolor="#FFFFFF">

<Script language="VBScript">

<!--

Sub window\_onLoad()

    call Despliegagrid()

End Sub

Sub Despliegagrid()

```

set form = document.grid
dim col
dim renglones
dim mes

col=0
renglones=0
form.grid.col=0
form.grid.row=0
form.grid.text="Aerolinea"
<DBOUTPUT QUERY="vtasaerolinea" group="AerolineaDescripcion">
    col=0
    renglones = renglones + 1
    form.grid.colwidth(col)=1500
    form.grid.col=col
    form.grid.row=renglones
    form.grid.text="#AerolineaDescripcion#"
    <DBOUTPUT>
        col=col + 1
        form.grid.colwidth(col)=1000
        form.grid.col=col
        form.grid.row=renglones
        form.grid.text="#dollarformat(AerolineaMonto)#"
    </DBOUTPUT>
</DBOUTPUT>

End sub
-->
</Script>

<form name="grid">

    <table border="0" width="760">
        <tr>
            <td width="20%"> </td>
            <td valign="top"></td>
        </tr>
    </table>
    <table border="0" width="760">
        <tr>
            <td><object id="grid" name="grid"
            classid="clsid:A8C3B720-0B5A-101B-B22E-00AA0037B2FC"
            border="0" width="550" height="200"><param
            name="_Version" value="65536"><param name="_ExtentX"
            value="7276"><param name="_ExtentY" value="4657"><param
            name="_StockProps" value="77"><param name="BackColor"
            value="16777215"><param name="Rows" value="15"><param
            name="Cols" value="12"></object></td>
        </tr>
    </table>

</form>
</body>
</html>

<!Query para llenar el combo de aerolineas->
<DBQUERY NAME = "ComboAerolinea" DATASOURCE = "DataW1"
    SQL="SELECT Aerolineaid,AerolineaDescripcion,AerolineaAbreviacion
    FROM Aerolinea
    ORDER BY AerolineaDescripcion">

<html>
<Title> Ventas por Aerolínea </Title>

<body background="http://troy/Abaturismo/Gray_DottedA025.gif" bgcolor="#FFFFFF">

```

```
<script language="VBScript">
<!--
Sub ValidacionDeCaptura
    set form = document.form1

    iCboMesIniSel=form.cboMesInicio.SelectedIndex
    itxDialniSel=form.txtDialInicio.value

    Select Case iCboMesIniSel

    Case 0: if itxDialniSel < 1 or itxDialniSel > 31 then
            msgbox "El mes de Enero tiene 31 dias",48,"Datos"
            exit sub
        end if

    Case 1: if itxDialniSel < 1 or itxDialniSel > 28 then
            msgbox "El mes de Febrero tiene 28 dias",48,"Datos"
            exit sub
        end if

    Case 2: if itxDialniSel < 1 or itxDialniSel > 31 then
            msgbox "El mes de Marzo tiene 31 dias",48,"Datos"
            exit sub
        end if

    Case 3: if itxDialniSel < 1 or itxDialniSel > 30 then
            msgbox "El mes de Abril tiene 30 dias",48,"Datos"
            exit sub
        end if

    Case 4: if itxDialniSel < 1 or itxDialniSel > 31 then
            msgbox "El mes de Mayo tiene 31 dias",48,"Datos"
            exit sub
        end if

    Case 5: if itxDialniSel < 1 or itxDialniSel > 30 then
            msgbox "El mes de Junio tiene 30 dias",48,"Datos"
            exit sub
        end if

    Case 6: if itxDialniSel < 1 or itxDialniSel > 31 then
            msgbox "El mes de Julio tiene 31 dias",48,"Datos"
            exit sub
        end if

    Case 7: if itxDialniSel < 1 or itxDialniSel > 31 then
            msgbox "El mes de Agosto tiene 31 dias",48,"Datos"
            exit sub
        end if

    Case 8: if itxDialniSel < 1 or itxDialniSel > 30 then
            msgbox "El mes de Septiembre tiene 30 dias",48,"Datos"
            exit sub
        end if

    Case 9: if itxDialniSel < 1 or itxDialniSel > 31 then
            msgbox "El mes de Octubre tiene 31 dias",48,"Datos"
            exit sub
        end if

    Case 10: if itxDialniSel < 1 or itxDialniSel > 30 then
            msgbox "El mes de Noviembre tiene 30 dias",48,"Datos"
            exit sub
        end if

    Case 11: if itxDialniSel < 1 or itxDialniSel > 31 then
            msgbox "El mes de Diciembre tiene 31 dias",48,"Datos"
            exit sub
        end if

end if
end sub
end script
```

```
End Select

iCboMesFinSel=form.cboMesFinal.SelectedIndex
itxtDiaFinSel=form.txtDiaFinal.value

Select Case iCboMesFinSel

Case 0: if itxtDiaFinSel < 1 or itxtDiaFinSel > 31 then
        msgbox "El mes de Enero tiene 31 dias",48,"Datos"
        exit sub
    end if

Case 1: if itxtDiaFinSel < 1 or itxtDiaFinSel > 28 then
        msgbox "El mes de Febrero tiene 28 dias",48,"Datos"
        exit sub
    end if

Case 2: if itxtDiaFinSel < 1 or itxtDiaFinSel > 31 then
        msgbox "El mes de Marzo tiene 31 dias",48,"Datos"
        exit sub
    end if

Case 3: if itxtDiaFinSel < 1 or itxtDiaFinSel > 30 then
        msgbox "El mes de Abril tiene 30 dias",48,"Datos"
        exit sub
    end if

Case 4: if itxtDiaFinSel < 1 or itxtDiaFinSel > 31 then
        msgbox "El mes de Mayo tiene 31 dias",48,"Datos"
        exit sub
    end if

Case 5: if itxtDiaFinSel < 1 or itxtDiaFinSel > 30 then
        msgbox "El mes de Junio tiene 30 dias",48,"Datos"
        exit sub
    end if

Case 6: if itxtDiaFinSel < 1 or itxtDiaFinSel > 31 then
        msgbox "El mes de Julio tiene 31 dias",48,"Datos"
        exit sub
    end if

Case 7: if itxtDiaFinSel < 1 or itxtDiaFinSel > 31 then
        msgbox "El mes de Agosto tiene 31 dias",48,"Datos"
        exit sub
    end if

Case 8: if itxtDiaFinSel < 1 or itxtDiaFinSel > 30 then
        msgbox "El mes de Septiembre tiene 30 dias",48,"Datos"
        exit sub
    end if

Case 9: if itxtDiaFinSel < 1 or itxtDiaFinSel > 31 then
        msgbox "El mes de Octubre tiene 31 dias",48,"Datos"
        exit sub
    end if

Case 10: if itxtDiaFinSel < 1 or itxtDiaFinSel > 30 then
        msgbox "El mes de Noviembre tiene 30 dias",48,"Datos"
        exit sub
    end if

Case 11: if itxtDiaFinSel < 1 or itxtDiaFinSel > 31 then
        msgbox "El mes de Diciembre tiene 31 dias",48,"Datos"
        exit sub
    end if

End Select
```

End Sub

-->  
</script>

0902653

<form  
action="/Scripts/dbml.exe?Template=/IntranetAbaTurismo/GridAerolinea.dbm"  
method="POST" name="form1" target="\_top">

```
<table border="0" width="760">
  <tr>
    <td width="20%"> </td>
    <td valign="top"></td>
  </tr>
</table>
<table border="0" width="760">
  <tr>
    <td valign="top"><font size="2" face="Arial">Aerolínea:<br>
    Nombre:<br clear="left">
    <select name="cboAerolinea" size="1">
      <option selected value="Todos"> [Todos] </option>
      <DBOUTPUT QUERY="ComboAerolinea">
      <option> #AerolineaDescripcion#
      </DBOUTPUT>
    </select></font></p>
    <p><font size="2" face="Arial">Periodicidad:<br>
    <select name="cboPeriodicidad" size="1">
      <option selected> Anual </option>
      <option> Trimestral </option>
      <option> Mensual </option>
      <option> Diario </option>
    </select></font></p>
    <p align="right">&nbsp;</p>
    <p align="right">&nbsp;</p>
    <td valign="top"><marquee bgcolor="#FFFFFF">Seleccione la Periodicidad , la Fecha de Inicio y la Fecha
Final.</marquee><font
size="2" face="Arial">Fecha Inicio:<br>
<input type="text" size="2" name="txtDiaInicio"
value="01"> </font><select name="cboMesInicio"
size="1">
  <option selected value="1"> Enero </option>
  <option value="2"> Febrero </option>
  <option value="3"> Marzo </option>
  <option value="4"> Abril </option>
  <option value="5"> Mayo </option>
  <option value="6"> Junio </option>
  <option value="7"> Julio </option>
  <option value="8"> Agosto </option>
  <option value="9"> Septiembre </option>
  <option value="10"> Octubre </option>
  <option value="11"> Noviembre </option>
  <option value="12"> Diciembre </option>
</select> <select name="cboAnioInicio" size="1">
  <option value="1996"> 1996 </option>
  <option selected value="1997"> 1997 </option>
</select> <p><font size="2" face="Arial">Fecha Final:<br>
<input type="text" size="2" name="txtDiaFinal"
value="01"> </font><select name="cboMesFinal"
size="1">
  <option selected value="1"> Enero </option>
  <option value="2"> Febrero </option>
  <option value="3"> Marzo </option>
```

```
<option value="4"> Abril </option>
<option value="5"> Mayo </option>
<option value="6"> Junio </option>
<option value="7"> Julio </option>
<option value="8"> Agosto </option>
<option value="9"> Septiembre </option>
<option value="10"> Octubre </option>
<option value="11"> Noviembre </option>
<option value="12"> Diciembre </option>
</select> <select name="cboAnioFinal" size="1">
  <option value="1996"> 1996 </option>
  <option selected value="1997"> 1997 </option>
</select></p>
<p>&nbsp;</p>
<hr size="3">
<p align="left"><font size="2" face="Arial">Resultados:<br>
<input type="radio" checked name="optTabular"
value="V1"> Tabular <br>
<input type="radio" name="optGrafica" value="V2">
Gráfica</font></p>
<p align="left"><input type="submit"
name="cmdCalcular" value="Calcular"
language="VBScript"
onclick="call ValidacionDeCaptura()"></p>
</td>
</tr>
</table>
</form>
</body>
</html>
```