

**UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERÍA  
ESCUELA DE COMPUTACION**



**“HERRAMIENTA PARA EL ANÁLISIS DE VARIABLES APLICADAS AL  
MARKETING, UTILIZANDO SERVIDORES OLAP Y MINERIA DE DATOS”**

**TESIS DE GRADUACIÓN PARA OPTAR AL GRADO DE INGENIERO EN CIENCIAS  
DE LA COMPUTACIÓN**

**PRESENTADA POR:  
FRANCISCO GERARDO AMAYA SOSA  
BLANCA IRIS CAÑAS ABARCA  
FRANCISCO RICARDO GUTIERREZ CACERES**

**SEPTIEMBRE 2004**

**SOYAPANGO**

**EL SALVADOR**

**CENTRO AMERICA**

# **UNIVERSIDAD DON BOSCO**

## **AUTORIDADES**

**ING. FEDERICO HUGUET RIVERA**  
**RECTOR**

**PADRE VICTOR BERMÚDEZ**  
**VICERRECTOR ACADEMICO**

**LIC. MARIO RAFAEL OLMOS**  
**SECRETARIO GENERAL**

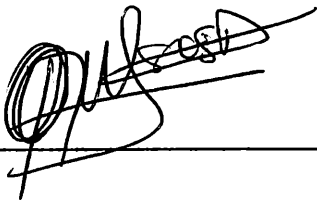
**ING. ERNESTO GODOFREDO GIRON**  
**DECANO DE LA FACULTAD DE INGENIERIA**

**LIC. MAURICIO COTO**  
**DIRECTOR DE ESCUELA DE COMPUTACION**

**UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERÍA**



**JURADO EVALUADOR DEL TRABAJO DE GRADUACION**



**ING. GILBERTO LARA  
ASESOR**



**LIC. OSCAR MENDEZ  
JURADO**



**ING. XIOMARA MARTINEZ  
JURADO**



**ING. ARNOLDO RIVAS  
JURADO**

## **Agradecimientos**

### **A Dios**

Por darme la oportunidad de vivir, por darme fuerzas para seguir adelante a pesar de todos los momentos adversos que tuve a lo largo de la carrera y por ayudarme a comprender que sin el esfuerzo y empeño nada es gratificante.

### **A mi familia**

Este logro se lo dedico con todo cariño a una persona que ya no esta mas con nosotros en este mundo pero que para mí sigue siendo alguien muy especial y que siempre llevo en mi corazón: Mi abuela Ana Maria Torres de Gutiérrez (Q.D.D.G), yo sé que desde el cielo me acompaña y celebra este triunfo. Finalmente lo logramos Mama Ana.

A mi papa: Ricardo Enrique Gutiérrez, por haberme tenido paciencia y ayudado en todo lo posible para lograr este triunfo. Gracias por apoyarme y haberme dado la oportunidad de demostrar a todos que se puede triunfar a pesar de tropezar en el camino.

A mis hermanos: Oscar Ernesto Gutiérrez y Leonel Enrique Gutiérrez, por el cariño y el apoyo hacia mi persona. También a mi familia en Juayúa que me apoyaron siempre. Muchas Gracias.

### **A mis Amigos**

Les agradezco mucho por su inmensa colaboración, va por todas las personas que contribuyeron en el proyecto; desde lo mas complejo hasta lo más elemental. Gracias a: Oliver toro (Ecuatoriano) por los ejemplos y bibliografía (Grande Maestro!!!), Oscar Antonio Rivas y Rodolfo Antonio Pineda por las desveladas en la creación de la base de datos y corrección de errores (sino son ustedes, ¿quién va a ser?), Raúl Rodríguez (Raulito) por colaborar con reportes impresos y papelería, Rene Tobar (Flash) por ayudarnos en programación, Carlos López (Carliños) por ayudarnos en programación, Juan Bonilla por colaborar en programación. Dedicado con todo el cariño para Led Zeppelin por el apoyo infinito recibido en todo momento y por estar pendiente de mis avances en este proceso, Patricia Meléndez por ese apoyo total y la confianza depositada en mi persona, también para mis amigos que siempre estuvieron apoyándome como Helmer Cáceres y Mario Alvarado. Gracias totales.

## **A mis compañeros**

Quiero agradecer a mi compañera Blanca Iris Cañas por afrontar y sobreponerse a los problemas que tuvimos durante este periodo, sabrás por las situaciones que yo pasaba y que me motivaban a salir adelante, de verdad, gracias por poner ganas en este trabajo aunque algunas veces estuvieras atareada, desganada o enferma. Realmente te agradezco por no ser como otros que ni menciono que solo sirvieron para agenciarse créditos que no les corresponden ni que les corresponderán jamás, personas como estas no caben en ninguna parte y solamente se aferran al triunfo a instancias y sacrificios de otros, por tal razón quedan fuera de cualquier agradecimiento. Gracias por ayudar y preocuparte por sacar adelante a nuestro grupo, por la unión mostrada a lo largo de este reto y por considerarme tu amigo.

Gracias a todos los compañeros del centro de computo de la Universidad que colaboraron y nos aguantaron en nuestro proyecto, especiales agradecimientos a Carlos Calles, Carlillos, Sandrita y Maria Elena.

Muchas gracias por todo, esto es para ustedes.

**Francisco Ricardo Gutiérrez Cáceres**

## **Agradecimientos**

### **A Dios**

Primero por regalarme la vida y ayudarme en todas mis necesidades que sin él nada somos en este mundo, a la **Virgen Maria** por su amor el cual se muestra en todos los días de mi vida.

### **A mis Padres**

Infinitas gracias a mis padres Rafael Cañas y Maria Teresa de Cañas por que sin ellos no hubiera logrado nada de lo que soy en estos días, por su paciencia, comprensión y especialmente por su amor hacia mi y estar siempre conmigo tanto en mis triunfos como en mis derrotas.

### **A mis Hermanos**

Maria Guadalupe, Marisol y Rafael por todo su cariño y amor hacia mí y por aguatarme tanto.

### **A mi Familia**

Por todo su apoyo, especialmente a mi Tía Antonia y mis primos por que siempre han sido para mi una ayuda y para mis padres.

### **A mis Amigos**

Porque han estado siempre conmigo durante todo este tiempo y por brindarme toda la ayuda que necesitaba.

### **A mis compañeros de trabajo**

Porque ustedes son para mí como mi segunda familia ya que compartimos tanto tiempo juntos, gracias porque me dieron animo para salir adelante cuando mas lo necesitaba.

### **A mi Asesor**

Por ayudarnos y porque siempre nos defendió a la hora que lo necesitábamos.

### **A mis compañeros de tesis**

Ricardo y Gerardo porque confiaron en mi para esta larga tarea la cual ha costado mucho sacrificios pero al final esta la recompensa y aunque hubieron dificultades siempre se sabia como salir adelante, de todo corazón les deseo lo mejor a los dos.

Estas personas son muy importantes para mí así que este triunfo es de ellos también

**Blanca Iris Cañas Abarca**

## INDICE

<b>INTRODUCCIÓN</b>	
A.	ANTECEDENTES..... 2
B.	JUSTIFICACIÓN DEL TEMA..... 4
C.	PROYECCIÓN SOCIAL..... 7
D.	DEFINICIÓN DEL TEMA..... 8
E.	OBJETIVOS..... 9
F.	ALCANCES Y LIMITACIONES..... 10
<b>CAPITULO I. MARCO TEORICO..... 13</b>	
1.1	DataWarehouse..... 14
1.1.1	Concepto de DataWarehouse..... 16
1.1.2	Características de un DataWarehouse..... 17
1.1.3	Estructura del DataWarehouse..... 18
1.1.4	Partes de un DataWarehouse..... 20
1.1.4.1	DataMart..... 20
1.1.4.2	Base de Datos relacionales..... 20
1.1.4.3	Fuente de Datos..... 20
1.1.4.4	Área de preparación de datos..... 20
1.1.4.5	Servicios de Presentación..... 21
1.1.4.6	Reportes Predefinidos..... 21
1.1.4.7	OLAP..... 21
1.1.4.8	Minería de Datos..... 21
1.2	Minería de Datos..... 22
1.2.1	Concepto de Minería de Datos..... 24
1.2.2	Descubrimiento de conocimiento de base de datos (KDD)..... 25
1.2.3	Historia de la Minería de Datos..... 27
1.2.4	Campos de Aplicación de la Minería de Datos..... 29
1.2.4.1	Marketing..... 29
1.2.4.2	Predicción..... 30
1.2.4.3	Reducción de Riesgos..... 30
1.2.4.4	Detección de Fraudes..... 30

1.2.4.5	Control de Calidad.....	30
1.2.4.6	Procesos Industriales.....	31
1.2.5	Operaciones y técnicas de la Minería de Datos.....	33
1.3	Online Analytical Processing (OLAP).....	35
1.3.1	La complejidad de los datos.....	37
1.3.2	Términos y conceptos de OLAP.....	39
1.3.2.1	Modelos de datos OLAP.....	39
1.3.2.2	Modelos de Agregación y Almacenamiento.....	40
1.3.3	Arquitectura OLAP.....	42
1.4	Expresiones Multidimensionales (MDX).....	45
1.4.1	Diferencias entre SQL y MDX.....	46
<b>CAPITULO II. TECNICAS PROPUESTAS A UTILIZAR Y SITUACIÓN</b>		
	<b>ACTUAL.....</b>	<b>47</b>
2.1	Técnicas a utilizar.....	48
2.2	Elementos claves de la Arquitectura OLAP relacionados con el Software a utilizar.....	52
2.2.1	PivotTable.....	52
2.2.1.1	Seguridad de los Servicios PivotTable.....	54
2.2.1.2	Desarrollo de Aplicaciones para el Cliente.....	54
2.2.1.3	Entorno de Desarrollo utilizando Visual Basic.....	55
2.2.2	Objetos de Apoyo de Decisión.....	57
2.2.2.1	Arquitectura de los Objetos de Apoyo de Decisión.....	59
2.2.2.2	Objetos DSO.....	62
2.3	Equipo Actual (situación actual de la empresa) y estadísticas de datos.....	65
2.3	Análisis actual de los datos.....	67
2.4	Porcentaje de efectividad de los datos.....	69
<b>CAPITULO III. DESARROLLO Y DISEÑO.....</b>		<b>71</b>
3.1	Introducción al Desarrollo y Diseño.....	72
3.1.1	Modelos de Datos.....	73

3.1.2	Características del MER.....	73
3.1.3	Características del Modelo Multidimensional.....	74
3.2	Área de dominio de la herramienta.....	75
3.3	Requerimientos del Sistema.....	76
3.4	Estructura de Base de Datos.....	80
3.4.1	Introducción a los Esquemas Estrella y Esquema Snowflake....	80
3.4.1.1	Esquema de Estrella.....	80
3.4.1.2	Esquema Snowflake (Copo de Nieve).....	80
3.4.2	Tabla Fact o de Hechos.....	81
3.4.3	Tablas Lookup o Dimensionales.....	82
3.5	Cubos.....	85
3.5.1	Cubos Producción.....	87
3.5.2	Cubos DataMart Ventas.....	107
3.6	Minería de Datos.....	113
3.6.1	Modelos de la Minería de Datos.....	113
3.6.2	Microsoft Decisión Trees (Árboles de desición).....	114
3.6.3	Generar y Utilizar Modelos de minería de datos.....	115
3.6.4	Crear modelos de minería de datos.....	115
3.6.5	Crear modelos de minería de datos OLAP.....	116
3.6.6	Columnas de minería de datos.....	122
3.6.7	Requisitos Previos.....	124
3.6.8	Modificar modelos de minería de datos OLAP.....	125
3.6.9	Editor de modelos de minería de datos OLAP.....	125
3.7	MDX (Expresiones Multidimensionales).....	128
3.7.1	Consulta para reportar el número de ventas por tipo de Cliente en la Datamart Ventas.....	128
3.7.2	Consulta para agregar dos dimensiones Datamart Ventas.....	129
3.7.3	Consulta que utiliza la sentencias WHERE.....	131
3.8	La herramienta OLAP Viewer.....	134
3.9	Diseño de la herramienta OLAP Viewer.....	136
3.9.1	Inicio de Sesión.....	136

3.9.2	Menú Principal.....	137
3.9.3	Diseño para la opción crear cubos.....	139
3.9.4	Formulario Consulta MDX.....	144
3.9.5	Formulario Consulta SQL.....	147
3.9.6	Diseño del administrador de cubos.....	149
3.9.7	Formularios de mantenimiento de Datamart Ventas y Datamart Producción.....	153
3.9.7.1	Formulario Mantenimiento Clientes.....	153
3.9.7.2	Formulario Mantenimiento Contacto.....	158
3.9.7.3	Formulario Mantenimiento Localización.....	160
3.9.7.4	Formulario Mantenimiento Tiempo.....	163
3.9.7.5	Formulario Mantenimiento Datos Personales...	165
3.9.7.6	Formulario Mantenimiento Zona.....	167
3.9.7.7	Formulario Mantenimiento Vendedor.....	169
3.9.7.8	Formulario Mantenimiento Producto.....	171
3.9.7.9	Formulario Mantenimiento Pagos.....	173
3.9.7.10	Formulario Data Warehouse Ventas.....	175
3.9.7.11	Formulario Mantenimiento Hechos_Ventas.....	177
CONCLUSIONES.....		183
BIBLIOGRAFÍA.....		184
ANEXOS		
	Anexo A.....	186
	Anexo B.....	211

## INDICE DE IMÁGENES

### ❖ Figuras

Figura 1.1	Proceso KDD.....	26
Figura 1.2	Conexión entre los modelos de minería de datos y sus Operaciones.....	33
Figura 1.3	Operaciones y Técnicas de minería de datos.....	34
Figura 1.4	Cubo de datos.....	39
Figura 1.5	Arquitectura OLAP.....	42
Figura 2.1	Jerarquía de Objeto DSO.....	60
Figura 3.1	Esquema Datamart Ventas.....	83
Figura 3.2	Esquema Datamart Producción.....	84
Figura 3.3	Tabla Hechos Ventas.....	84
Figura 3.4	Analysis Manager.....	88
Figura 3.5	Editor de Cubos.....	89
Figura 3.6	Elección de campos.....	90
Figura 3.7	Selección de dimensiones.....	91
Figura 3.8	Crear dimensiones.....	92
Figura 3.9	Tablas seleccionadas.....	93
Figura 3.10	Uniones de tablas seleccionadas.....	94
Figura 3.11	Selección de niveles.....	95
Figura 3.12	Finalización del proceso de creación de dimensión.....	96
Figura 3.13	Final del proceso de creación del cubo.....	97
Figura 3.14	Analysis Manager –Cubo creado-.....	98
Figura 3.15	Analysis Manager.....	99
Figura 3.16	Tipo de almacenamiento.....	100
Figura 3.17	Opciones de agregaciones.....	101
Figura 3.18	Opciones de agregaciones.....	102
Figura 3.19	Procesamiento del cubo.....	103
Figura 3.20	Resultado final del cubo creado.....	104
Figura 3.21	Dimensiones y medidas del cubo Datawarehouse.....	105

Figura 3.22 Resultado final del cubo DATAWAREHOUSE PRODUCCIÓN...	106
Figura 3.23 Estructura del cubo.....	107
Figura 3.24 Datos cubo Ventas1.....	108
Figura 3.25 Cubo Ventas2.....	109
Figura 3.26 Resultados Cubo Ventas2.....	110
Figura 3.27 Datawarehouse_Ventas.....	111
Figura 3.28 Resultados Datawarehouse_Ventas.....	112
Figura 3.29 Análisis de la minería de datos en el cubo VENTAS.....	117
Figura 3.30 Analysis Manager –nuevo modelo de minería de datos-.....	118
Figura 3.31 Selección de técnica de minería de datos.....	119
Figura 3.32 La dimensión y el nivel del escenario.....	120
Figura 3.33 Entidad a predecir.....	121
Figura 3.34 Datos de entrenamiento.....	122
Figura 3.35 Creación de una nueva dimensión.....	123
Figura 3.36 Finalización del modelo de minería de datos.....	124
Figura 3.37 Editor de modelos de minería de datos OLAP.....	127
Figura 3.38 Consulta MDX ventas por tipo cliente en Datamart Ventas.....	129
Figura 3.39 Consulta para agregar dos dimensiones Datamart Ventas.....	130
Figura 3.40 Consulta de dos dimensiones Datamart ventas.....	131
Figura 3.41 Consulta MDX que utiliza la sentencia WHERE.....	132
Figura 3.42 Inicio de Sesión.....	137
Figura 3.43 Menú Principal.....	138
Figura 3.44 Menú Principal y sus 5 niveles.....	138
Figura 3.45 Opción: Crear Cubo.....	139
Figura 3.46 Primer Paso: Agregar Cubo.....	142
Figura 3.47 Segundo Paso: Identificación de Tabla Hechos.....	142
Figura 3.48 Tercer Paso: Agregar Medidas.....	143
Figura 3.49 Cuarto Paso: Agregar Dimensiones.....	143
Figura 3.50 Quinto Paso: Relacionar Tablas.....	144
Figura 3.51 Formulario Consulta MDX.....	145
Figura 3.52 Formulario Consulta SQL.....	148

## INTRODUCCIÓN

El presente trabajo se desarrolla con la idea de proponer y crear un instrumento informático basado en una completa investigación sobre los principios y conceptos básicos que se encierran dentro de la tecnología OLAP y la minería de datos, la cual se aplica en distintos campos donde se desenvuelve el ser humano con el fin de colaborar y facilitar las actividades que desempeña. Estas tecnologías se desprenden de una tecnología mucha más amplia como lo es la del Data Warehouse.

Para brindar una mejor comprensión de lo explicado anteriormente se profundiza en uno de los campos más conocidos y donde más éxito a tenido la minería de datos a nivel mundial. Se habla del marketing, en la cual la información que tengamos almacenada es importante, cuanto más precisa, mayores posibilidades se tendrá de aumentar los ingresos y rentabilizar al máximo las acciones.

El objetivo fundamental se puede resumir en determinar quién comprará qué, cuándo y dónde. También se trabajara con datos almacenados de forma multidimensional mediante la tecnología OLAP, para tener un análisis rápido de la información. Las herramientas OLAP permiten navegar a través de datos almacenados en el Data Warehouse y analizarlos desde una perspectiva multidimensional, considerando unas variables con relación a otras y no de forma independiente, permitiendo enfocar el análisis desde distintos puntos de vistas.

Se utilizara el Análisis Service de Microsoft SQL Server 2000, el cual introduce la minería de datos, una nueva característica que integra funciones de análisis de datos y de predicción para Analysis Services. Otra característica es que trabaja de la mano con las herramientas OLAP. Además, la biblioteca de Objetos de Apoyo de Decisión (Decision Support Objects -DSO) de Análisis Service de Microsoft SQL Server 2000, proporcionan un juego robusto de Modelo de Objeto de Componente (COM), que son objetos e interfaces que se pueden usar para crear aplicaciones para administrar por medio de programas los objetos Análisis Service.

El algoritmo de minería de datos a utilizar será el de árbol de decisión, el cual se basa en la idea de clasificación. El algoritmo genera un árbol que podrá predecir el valor de una columna según las columnas restantes del conjunto de entrenamiento. En consecuencia, cada nodo del árbol representa un caso particular para una columna.

Después de hacer una investigación de las técnicas a utilizar para crear la herramienta que analice variables, el presente documento tiene como finalidad ofrecer una solución a los problemas del área de mercadeo, apoyándose de una empresa para resolverlos solo para efectos de estudio, con el objetivo de ayudar a una mejor toma de decisiones.

## GENERALIDADES

---

En este apartado se hace una descripción del marco de referencia del trabajo de tesis **“Herramienta para el análisis de variables aplicadas al marketing utilizando Minería de datos y Servidores OLAP”**, el cual contiene los apartados de: antecedentes, justificación del tema, proyección social, definición del tema, objetivos, alcances y limitaciones.

## A. ANTECEDENTES

Los datos almacenados en una base de datos crecen rápidamente día a día y es casi imposible procesarlos para el descubrimiento de patrones en las variables utilizando los métodos clásicos del administrador de base de datos, para superar este problema han surgido técnicas que hacen fácil el procesamiento de datos de una forma avanzada, permitiendo realizar un análisis profundo de los mismos de forma automática. Una tecnología que aplica las técnicas para procesar los datos, extraer información útil contenida en un almacén de datos y de gran importancia para la toma de decisiones es la Minería de Datos.

Para someter los datos a la tecnología de data mining<sup>1</sup>, es indispensable tener acceso a los mismos, por lo que no es de extrañar que esta tecnología se haya desarrollado a la par del almacenamiento de datos (data warehousing). A medida que aumentaba, a finales del siglo pasado, la potencia de los ordenadores y la capacidad de las bases de datos, las empresas empezaron a darse cuenta de que los datos no eran simples receptáculos pasivos sólo útiles a la hora de realizar tareas de facturación o de registro de pedidos. Estas comprendieron que también podían utilizar los datos para la obtención de información que les mostrara el camino que debían seguir. Este planteamiento dio lugar al desarrollo de una nueva generación de sistemas informáticos que no se limitaban a la simple gestión de las empresas (como hacían las primeras aplicaciones informáticas), sino que abordaban también tareas de información y análisis. A estos nuevos sistemas se les denominó "*sistemas de ayuda a la toma de decisiones*"<sup>2</sup>

---

<sup>1</sup> Data Mining, es como se escribe en ingles Minería de Datos.

<sup>2</sup> Executive Information System por sus siglas en ingles EIS, también son conocidos como "Sistema de información ejecutivo".

A medida que fue madurando la tecnología del data warehousing, los sistemas EIS y de apoyo a la toma de decisiones dieron paso a los conceptos más generales de la BI (Business Intelligence) y del data mining.

La BI conlleva la organización de los datos en distintas dimensiones potenciales de análisis de modo que se puedan mostrar y establecer referencias cruzadas entre cualesquiera vistas de datos. (los resultados de ventas, por ejemplo) extraídos de cualesquiera otras dimensiones potenciales (como la región o la línea de producto). La posibilidad de pasar de una dimensión a otra permite descender al detalle o distanciarse para ofrecer una visión más general. La posibilidad de mostrar variaciones en los datos de distintas dimensiones (con frecuencia, de muchas dimensiones a la vez) permite generar informes multidimensionales en tiempo real. Este modo general de entender la manipulación de datos recibió el nombre de OLAP (OnLine Analytical Processing o Procesamiento analítico online), es decir, procesamiento de datos con fines analíticos en lugar de operativos, de esta manera OLAP se constituye como tecnología básica para el análisis de información multidimensional y resulta de gran ayuda para la verificación e interpretación de los datos, OLAP por otra parte utiliza una tecnología paralela para el estudio de los datos, esta recibe el nombre de MDX (Multidimensional Expressions - Expresiones Multidimensionales- ).

## **B. JUSTIFICACIÓN DEL TEMA**

### **Situación actual de la empresa**

BIOLACT, es una empresa conformada por pequeños ganaderos que creen en el desarrollo de la ganadería orgánica y en la asociatividad empresarial, así como la generación de valor agregado elaborando productos de calidad para un mercado saludable, a precio justo para clientes de toda Centro América.

El grupo de ganaderos surge como una forma de organización de pequeños productores de leche que ven la necesidad de conformar un ente que le genere valor agregado a la producción de materia prima, posibilitándoles la compra de la leche, el incentivar la asociatividad y la transformación de ganadería convencional a orgánica y a la vez fomentar el procesamiento de la leche para obtener productos gourmet como son los quesos Europeos (Fresco, Tipo Burgos, Curado, Semicurado, Camembert, etc) reduciendo las importaciones de Europa y proporcionándole al consumidor Centroamericano productos de calidad, higiénicos, nutritivos y a un precio accesible para mayor cantidad de clientes.

Las proyecciones del mercado son realizadas manualmente por lo que no se cuenta con ninguna herramienta automatizada que sea de ayuda para el análisis de grandes volúmenes de información y que permita analizar el comportamiento de variables relacionadas con el marketing en un corto periodo.

## **Problemática Actual**

Actualmente la empresa no cuenta con una base de datos potencialmente estructurada que le permita hacer un análisis más complejo y de utilidad para la toma de decisiones. La empresa no cuenta con software que permita el análisis de variables multidimensionales de los datos actuales e históricos.

## **Como trabaja la empresa**

Estadísticamente un 80% se consolida como información relevante para las empresas a todos aquellos datos que provienen de bases de datos relacionales y procesos como consultas y reportes. Para BIOLACT el 100% de la información proviene de reportes de procesos internos y proyecciones de mercado utilizando modelos estadísticos elaborados manualmente.

Para observar el comportamiento de las variables que están relacionadas en el Marketing es necesario crear una aplicación que permita la detección de todas las pautas y dependencias que existan en los datos mediante campos previamente seleccionados por el administrador, esto se puede lograr utilizando:

- OLAP Server, ya que se constituye como una tecnología rápida y flexible para el análisis de los datos, este análisis requerirá de la habilidad para sumarizar datos en múltiples caminos y ver oportunidades sobre el tiempo.
- Data Mining (Minería de Datos), utilizando algoritmos de árboles de decisión para predicción del comportamiento de variables que la empresa no puede visualizar con métodos convencionales.
- Tecnología MDX, para realizar consultas a cubos (datos multidimensionales).

Estas tecnologías permitirán analizar el 20% de información que no se puede determinar a simple vista y que se constituye como información vital para la gerencia.

## **Datos analizados por la empresa**

Los datos que son utilizados para la proyección de mercado son todos aquellos que al igual que cualquier empresa provienen de procesos como ventas, compras, clientes, productos, etc. y que requieren de ejercicios contables comunes.

## **Volumen de datos analizados**

Muchos datos son analizados en la empresa, la naturaleza de esta permite el estudio de miles de datos debido a la variedad de productos, personal y clientes que posee, y que día a día se incrementan.

Por tal motivo, es imprescindible una herramienta con la capacidad de involucrar todas las variables de marketing, permitiendo de esta manera la agilización del flujo de datos.

Tomando como referencia la problemática actual de la empresa, como trabaja y el volumen de datos analizados se pretende, desarrollar una herramienta para el análisis de variables aplicadas al mercadeo, la cual es un campo muy extenso y de mayor cuidado para las empresas, utilizando modelos, técnicas de minería de datos y servidores OLAP. Dicha herramienta tendrá la capacidad de ser aplicada a diferentes tipos de instituciones, la empresa en estudio es solamente una referencia para el desarrollo de la herramienta, la cual, ayudará a mejorar la eficacia de las actividades de las instituciones, descubrir nuevas fuentes de beneficios y predecir ciertas variables como las ventas, el aprovechamiento de nuevas oportunidades, prevenir problemas de clientes.

## **C. PROYECCIÓN SOCIAL**

El objetivo final de cualquier proyecto de descubrimiento de información utilizando servidores OLAP de datos puede resumirse en:

- Ahorrar dinero mejorando la eficacia de las actividades.
- Ganar dinero descubriendo nuevas fuentes de beneficios.

Esto se logra a partir de un conjunto de datos y técnicas, los resultados suponen una mejora en la información disponible.

El mercadeo es uno de los campos donde ha tenido más éxito la minería de datos. Cuanto más precisa sea la información almacenada que tengamos, mayores serán las posibilidades de aumentar los ingresos y rentabilizar al máximo las acciones de cualquier empresa o institución.

Basándonos en Servidores OLAP, pretendemos desarrollar un modulo que ayude a cualquier empresa a estudiar el comportamiento de los datos multidimensionales, el área de mercadeo para facilitar el procesamiento analítico de apoyo a la toma de decisiones estratégicas.

Por otra parte, la realización de este trabajo pretende convertirse en material de apoyo para futuros profesionales, brindando un soporte académico tanto a la Universidad Don Bosco en el desarrollo de nuevas tecnologías orientadas a la minería de datos y servidores OLAP, así como a otras Universidades. Además sirviendo de apoyo como un material disponible para todo estudiante universitario interesado en explorar las técnicas de minería de datos en el área del mercadeo y Servidores OLAP.

## **D. DEFINICIÓN DEL TEMA**

Herramienta para el análisis de variables aplicadas al marketing utilizando Minería de datos y Servidores OLAP.

Se pretende crear una herramienta que permita observar el comportamiento de variables orientadas al marketing utilizando Servidores OLAP de Análisis Service, con el objetivo de que las empresas permitan diseñar estrategias para buscar mayores beneficios, reducción de costes, fabricación de productos de mayor calidad y el aumento del grado de satisfacción de sus clientes.

## **E. OBJETIVOS**

### **Objetivo General**

Desarrollar una herramienta que analiza el comportamiento de las variables utilizadas en marketing con el propósito de presentar resultados claves y comprensibles utilizando tecnología OLAP como técnica de minería de datos.

### **Objetivos Específicos**

- Diseñar una herramienta que aproveche la tecnología de cubos OLAP para el análisis y tareas de información que abordan las empresas.
- Mejorar los tiempos de análisis de información para la toma de decisiones.
- Generar pronósticos e informes de los datos que recopilan, almacenan y manipulan las empresas.
- Aplicar los beneficios de la tecnología OLAP para organizar y resumir cantidades de datos

## **F. ALCANCES Y LIMITACIONES**

### **Alcances**

- Aplicar Servidores OLAP para el procesamiento de datos en línea bajo el concepto de Datawarehousing.
- Se pretende que el sistema permita generar reportes corporativos que se constituyan como soporte avanzado en la toma de decisiones.
- Mediante la tecnología de Servidores OLAP el sistema podrá analizar el comportamiento de la información de forma multidimensional.
- El sistema de información proporcionara a la empresa un medio para acceder, ver y analizar los datos con alto desempeño y flexibilidad.
- El sistema ayudará a usuarios finales comprender efectivamente la información del almacén de datos, permitiendo así a la empresa reconocer mejor el valor de sus datos.
- La herramienta permitirá el acceso a todos los datos y la información necesaria y relevante, donde sea que ésta resida y no limitada por el volumen.

## **Limitaciones**

- Los datos que se explotarán mediante la tecnología de Servidores OLAP se constituirán como un conjunto de tablas provenientes de los Datamarts.
- Los datos no se pueden actualizar automáticamente, solo se pueden procesar, eso se hará periódicamente, este inconveniente se puede resolver utilizando SQL Enterprise.
- Los datos tomados por el modelo de la minería de datos deben estar almacenados en las tablas de modelado de la base de datos.
- Los cubos OLAP están orientados a Datamarts creados para las necesidades de los departamentos de la empresa.
- Los cubos OLAP proporcionaran la información al usuario a una velocidad constante, la mayoría de las peticiones se deben responder al usuario en cinco segundos o menos.
- Los modelos predictivos de Minería de datos se constituirán como objetos de apoyo para el análisis de la información independientes de la herramienta propuesta para el análisis de servidores OLAP.
- El modo de almacenamiento de los cubos OLAP creados será MOLAP.
- Es indispensable el uso del Análisis Service, ya que se trabaja con objetos DSO. Los cuales, son objetos intrínsecos de SQL Server.

- A la hora de crear consultas de expresiones multidimensionales (MDX), no se permite el manejo de dimensiones compartidas, es decir, solo maneja cubos con dimensiones que no están siendo compartidas con otros cubos.
- No se crean cubos de minería de datos, solamente se analizan modelos previamente creados en SQL Server.

## CAPITULO I. MARCO TEÓRICO.

---

En este capítulo se hace un estudio de los factores que influyen en la herramienta diseñada y son tomados como base, donde se define que es el Data Warehouse, sus características, la estructura que tiene y las partes que lo componen. También se dan conceptos de Minería de Datos, los campos donde se utiliza, se da una breve historia de cómo ha evolucionado y de donde proviene, las técnicas y operaciones que se emplean en esta área, y se habla del “Proceso de descubrimiento de la información”.

Se describe que son los Servidores OLAP, el concepto, los modelos de datos OLAP, de agregación y de almacenamiento. Se habla de las Expresiones Multidimensionales y las diferencias que tiene con respecto a SQL.

## 1.1 Data Warehouse

Las organizaciones recolectan datos en el transcurso normal de sus negocios para atender sus necesidades de información desde que inició la era de la computadora. Se ha venido recolectando información sobre las cuales se toman decisiones importantes, atendiendo la necesidad del usuario de dos maneras, las cuales han evolucionado a través del tiempo, estas son: proporcionando acceso directo a la información contenida dentro de las aplicaciones operacionales o extrayendo los datos desde sus bases de datos operacionales para combinarlos de varias formas no estructuradas.

Las instituciones en su área administrativa saben que para elevar su eficiencia tienen que hacer uso de la información existente en la empresa, lo cual depende de su arquitectura actual tanto en software como en hardware. Es por eso que el Data Warehouse<sup>3</sup> es el centro de atención hoy en día, ya que provee un ambiente para que las organizaciones hagan un mejor uso de la información que está siendo administrada por diversas aplicaciones operacionales.

En un Data Warehouse la información de la institución esta integrada para usarla como soporte en la toma de decisiones gerenciales. Reuniendo datos apropiados desde diversas fuentes de aplicación en un ambiente integral centralizado, simplifica el problema de acceso a la información y en consecuencia, acelera el proceso de análisis, consultas y el menor tiempo de uso de la información.

La toma de decisiones utilizando la arquitectura del Data Warehouse puede hacer mas eficaz un negocio, explotando los datos de una manera mas fácil y práctica, que no se logra cuando se usan sólo los datos que provienen de las aplicaciones operacionales (que ayudan en la operación de la empresa en sus operaciones cotidianas), en los que la información se obtiene realizando procesos independientes y muchas veces complejos.

---

<sup>3</sup> El concepto de Data Warehouse o Almacén de Datos fue introducido a inicios de la década de los 90's, sin embargo, las empresas han empleado la transformación y recopilación de datos por décadas.

Un Data Warehouse se crea al extraer la información desde una o más bases de datos de aplicaciones operacionales, los cuales son transformados para eliminar inconsistencias y resumir si es necesario. El proceso de transformar, crear el detalle de tiempo variante, resumir y combinar los extractos de datos, ayudan a crear el ambiente para el acceso a la información Institucional, el cual ayuda a efectuar la toma de decisiones con más responsabilidad.

El Data Warehouse, permite a cualquier organización hacer un uso más óptimo de los datos, como un ingrediente clave para un proceso de toma de decisiones más efectivo. Las organizaciones tienen que aprovechar sus recursos de datos para crear la información de la operación del negocio, pero deben considerarse las estrategias tecnológicas necesarias para la implementación de una arquitectura completa de Data Warehouse.

### 1.1.1 Concepto de Data Warehouse

Se pueden encontrar muchos términos o definiciones para el Data Warehouse, los más comunes son:

Almacén de datos donde se concentra la información de interés de toda la empresa.

Es un sistema que permite el almacenamiento de la información en un único entorno, proveniente de distintos sistemas de la empresa y que refleja los indicadores claves asociados a los negocios de la misma.

Sistema de información orientado a la toma de decisiones empresariales, que almacenando de manera integra la información relevante del negocio, permite la realización de consultas complejas con tiempos cortos de respuesta.

Un Data Warehouse o Depósito de Datos es una colección de datos orientado a temas, integrado, no volátil, de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales.

Se puede concluir que el Data Warehouse es un almacén estructurado de la información clave de nuestro negocio, que integra datos proveniente de distintas áreas de la empresa, requerida en cualquier momento, permitiendo analizar el funcionamiento de nuestra compañía y tomar decisiones sobre su gestión en tiempos de respuestas cortos.

### 1.1.2 Características de un Data Warehouse

Entre las principales características se tiene:

*Orientado al tema*, la información se clasifica en base a los aspectos que son de interés para la empresa, donde los datos tomados están en contraste con los clásicos procesos orientados a las aplicaciones.

*Integración*, el aspecto más importante del ambiente Data Warehouse, significa que la información encontrada al interior está siempre integrada, no importando de donde proviene o si esta almacenada de forma diferente.

*De tiempo variante*, la información en el Data Warehouse es solicitada en cualquier momento, donde se debe incluir los datos históricos para usarse en la identificación y evaluación de tendencias. El tiempo variante se puede mostrar de 3 maneras:

- La información representa los datos sobre un horizonte largo de tiempo.
- La estructura clave. Cada estructura clave en el Data Warehouse contiene, implícita o explícitamente, un elemento de tiempo como día, semana, mes, etc.
- La información del Data Warehouse, una vez registrada correctamente, no puede ser actualizada

*No Volátil*, esto quiere decir que la información es estable, los datos dentro del almacén solo es útil cuando son estables.

### 1.1.3 Estructura del Data Warehouse

Los componentes del dataWarehouse son:

**Detalle de datos actuales.**- Refleja las ocurrencias más recientes, las cuales son de gran interés, es voluminoso, ya que se almacena al más bajo nivel de granularidad, casi siempre se almacena en disco, el cual es de fácil acceso, aunque su administración sea costosa y compleja.

**Detalle de datos antiguos.**- Se almacena sobre alguna forma de almacenamiento masivo. No frecuentemente se accesa y se almacena a un nivel de detalle, consistente con los datos detallados actuales. Mientras no sea prioritario el almacenamiento es un medio de almacenaje alternativo, es poco usual utilizar el disco como medio de almacenamiento.

**Datos ligeramente resumidos.**- Proviene desde un bajo nivel de detalle encontrado al nivel de detalle actual, casi siempre se almacena en disco

**Datos completamente resumidos.**- Estos datos son compactos y fácilmente accesibles, son parte del Data Warehouse sin considerar donde se alojan los datos físicamente

**Metadata.**- Su contenido no es tomado directamente desde el ambiente operacional. La metadata juega un rol especial y muy importante en el Data Warehouse y es usada como:

Un directorio para ayudar al analista a ubicar los contenidos del Data Warehouse, una guía para el mapping de datos de cómo se transforma, del ambiente operacional al de Data Warehouse, una guía de los algoritmos usados para la esquematización entre el detalle de datos actual, con los datos ligeramente resumidos y éstos, con los datos completamente resumidos.

## 1.1.4 Partes de un DataWarehouse

**1.1.4.1 DataMarts**, en algunas implementaciones es un Data Warehouse en miniatura; en otras es solo un segmento del Data Warehouse. Son utilizados frecuentemente para proveer información a segmentos funcionales de la organización, también pueden ser utilizados para dividir los datos de un Data Warehouse en segmentos que reflejen negocios en comportamientos geográficos en los cuales cada región es relativamente autónoma.

Algunas veces son diseñados como Data Warehouse completos e individuales y contribuyen al conjunto de la organización como miembros de un almacén de datos distribuido. En otros diseños reciben datos de un Data Warehouse maestro a través de actualizaciones periódicas, en cuyo caso el Datamart esta frecuentemente limitado a los servicios de prestación de los clientes.

**1.1.4.2 Bases de datos relacionales**, son los motores fundamentales que proveen el poder al Data Warehouse, son utilizadas para presentar, limpiar y transformar datos de entrada en el área de preparación de datos, contener y manejar cantidades masivas de datos en la base de datos Data Warehouse y dar soporte a los DataMarts.

**1.1.4.3 Fuente de Datos**, Data Warehouse esta orientado a proveer información a las personas que toman decisiones, para hacerlo, obtienen sus datos de diferentes fuentes y los transforman en grupos de datos consistentes que efectivamente reflejan las operaciones de negocios que realizan la organización y su historia.

**1.1.4.4 Área de preparación de datos**, es una base de datos relacional en la cual los datos son extraídos desde sus fuentes, transformados en formatos comunes, revisados para su consistencia e integridad referencial, y los prepara para ser cargados en el Data Warehouse.

**1.1.4.5 Servicios de Presentación**, un almacén de datos que contenga cientos de millones de registros y que no posea herramientas de análisis, no será de mucho valor para los encargados de tomar decisiones. Estas herramientas de análisis pueden variar desde simples reportes hasta sofisticados algoritmos de minería de datos.

**1.1.4.6 Reportes Predefinidos**, pueden servir a los administradores para obtener periódica o en el momento de la demanda, muestras del estado de los negocios de un determinado momento en el tiempo.

**1.1.4.7 OLAP**, es una herramienta diseñada para facilitar un análisis exploratorio de las cantidades masivas de datos en el Data Warehouse.

**1.1.4.8 Minería de Datos**, realiza el análisis exploratorio e identifica bloques de información interesante, además puede crear árboles de decisión que pueden ser utilizados para predecir datos futuros basados en atributos de elementos de datos existentes.

Hoy en día los Servicios de Análisis, incorporan sofisticados algoritmos de minería de datos que pueden ser utilizados para analizar la información en la base de datos relacionales o en cubos OLAP y esos resultados fortalecer el análisis exploratorio.

## 1.2 Minería de datos

En la actualidad el almacenamiento de la información en forma digital es más barato y sencillo, generando grandes cantidades de datos de los cuales se tiene que sacar provecho para tener una mejor toma de decisiones.

Por otro lado, la tecnología de las base de datos ha evolucionado de manera muy positiva gracias al desarrollo en las plataformas de hardware y software, revolucionando la manera de almacenar, organizar y manejar los datos. Es por ésta razón que las organizaciones han venido recolectando información relevante para sus operaciones durante mucho tiempo, almacenándola en grandes depósitos de datos llamados Data Warehouse.

Para utilizar la información como apoyo para la toma de decisiones, es necesario:

- Un sistema de almacenamiento apropiado, el cual puede ser, Data Warehouse o Herramientas OLAP (On Line Analytical Processing).
- Técnicas para extraer resultados e información útil para la toma de decisiones, las cuales pueden ser de Minería de Datos.

La Minería de Datos no es una tecnología que suplanta a otras, mas bien se complementa y se apoya de otras disciplinas, es por eso que se puede decir que es un campo interdisciplinario que atrae desde el análisis estadístico, los sistemas de base de datos, el aprendizaje de máquina (Machine Learning), patrones de reconocimientos, redes neuronales, los sistemas difusos y otras técnicas.

Utilizando diferentes técnicas la Minería de datos puede resolver problemas típicos, ya sea en deportes, educación, mercadeo, en predicción, en control de calidad, para reducir riesgos, en la industria y en otras áreas.

La Minería de Datos es el elemento principal de un proceso mucho mas elaborado llamado descubrimiento de conocimiento en base de datos (Knowledge Discovery in Database o KDD), aunque finalmente haya adquirido el significado de todo el proceso en lugar de la fase de extracción de conocimiento.

### 1.2.1 Concepto de Minería de Datos

En el presente apartado se enuncian dos definiciones de Minería de Datos.

“Proceso que a través del descubrimiento y cuantificación de relaciones predictivas en los datos, permite transformar la información disponible en conocimiento útil del negocio<sup>4</sup>”

“Extracción no trivial de información implícita, previamente desconocida y potencialmente útil a partir de los datos almacenados”.

Se encuentran otras definiciones de Minería de Datos, pero todas tienen el mismo significado, de las anteriores se puede decir entonces, que:

La Minería de Datos aplicando técnicas, transforma la información almacenada por una empresa en datos útiles al negocio, ayudando a un análisis más profundo de la información.

Como se dijo anteriormente, comúnmente se considera a la Minería de Datos como el elemento principal de un proceso mucho más elaborado llamado Knowledge Discovery in Database (Descubrimiento de Conocimiento en Base de Datos) o KDD.

---

<sup>4</sup> W. Frawly, G. Platesky-Shapiro, C. Matheus, “Knowledge Discovery in Databases: An Overview”, AI Magazine, Otoño 1992 (pag. 213-228)

## 1.2.2 KDD

El descubrimiento de conocimiento en base de datos o KDD es un proceso que consiste de una secuencia iterativa de pasos, los cuales se describen a continuación:

- Selección de datos (Data Selection), donde los datos que se analizarán son escogidos de la base de datos.
- Limpieza de datos (Data Cleaning), se encarga del ruido, datos perdidos, erróneos o irrelevantes.
- Integración de datos (Enrichment), donde múltiples datos heterogéneos pueden ser integrados dentro de uno solo.
- Transformación de datos (Coding), los datos son transformados o consolidados dentro de formas apropiadas para los diferentes algoritmos de Minería.
- Minería de Datos (Data Mining), es un proceso esencial, donde métodos inteligentes son aplicados en orden para extraer datos ocultos y conocimiento valioso.
- Representación del Conocimiento (Reporting), son aplicadas técnicas para representar el conocimiento extraído de la Minería de Datos.

A continuación presentamos la figura que corresponde al proceso del descubrimiento del conocimiento en base de datos o KDD por sus siglas en ingles.

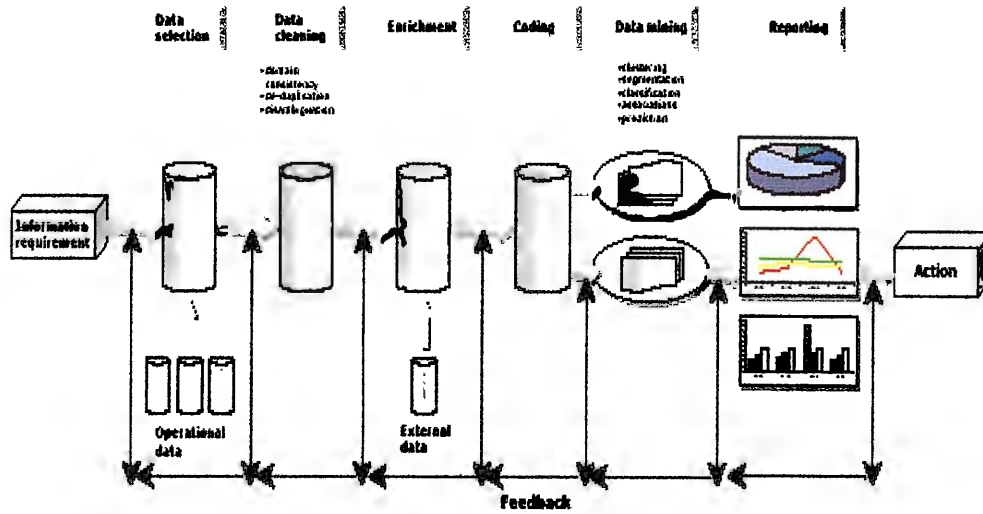


Figura 1.1 Proceso del Descubrimiento del conocimiento en base de datos (KDD)

### 1.2.3 Historia de la Minería de Datos

Pocos años atrás poca gente había oído del término Minería de Datos. Aunque la explotación minera de los datos es la evolución de un campo con una historia larga, el término en sí mismo fue introducido en los años 90's.

Las raíces de la Minería de Datos se remontan a lo largo de tres grandes líneas.

La primera y más larga es la estadística clásica. Sin estadística, no habría Minería de Datos, pues es la fundación de la mayoría de las tecnologías en las cuales se basa. La estadística clásica abraza conceptos tales como análisis de la regresión, distribución estándar, desviación estándar, variación estándar, análisis discriminante e intervalos de la confianza, que se utilizan para estudiar datos y sus relaciones. El análisis estadístico clásico desempeña un papel significativo en las técnicas de Minería de Datos.

La segunda línea es la Inteligencia Artificial. Esta disciplina se construye sobre la heurística en oposición a la estadística y se define como la ciencia que trata la comprensión de la inteligencia y del diseño de máquinas inteligentes, como resultado de implementar en un objeto inanimado las facultades humanas que configuran la inteligencia, es decir, el estudio y la simulación de las actividades intelectuales del hombre como manipulación, razonamiento, percepción, aprendizaje y creación.

Como se requiere energía extensa del tratamiento por el ordenador, no era práctico hasta inicio de los años 80, cuando las computadoras comenzaron a ofrecer poder útil a precios razonables. Ciertos conceptos de la Inteligencia Artificial fueron adoptados por productos comerciales, tales como la optimización de los módulos de consultas para Sistemas de Administración para Base de Datos Relacionales (RDBMS).

La tercera línea de la Minería de Datos es el Aprendizaje de Máquina (Machine Learning), que se describe más exactamente como la unión de la estadística y de la

AI<sup>5</sup>. Mientras que la Inteligencia Artificial no era un éxito comercial, sus técnicas fueron cooptadas en gran parte por el Aprendizaje de Máquina. Se aprovechó de la relación precio/poder de máquina ofrecidos por las computadoras de los 80's y de los 90's, encontró más usos porque los precios de entrada eran más bajos que el de la AI. El Aprendizaje de Máquina se podría considerar una evolución del AI, porque mezcla la heurística de la Inteligencia Artificial con los análisis estadísticos avanzados. El Aprendizaje de Máquina realiza programas que toman diversas decisiones basadas en las calidades de los datos estudiados, usando la estadística para los conceptos fundamentales, más la heurística y algoritmos del AI para alcanzar sus metas.

La Minería de datos, de muchas maneras, es fundamentalmente la adaptación de las técnicas del Aprendizaje de Máquina a los usos de negocios.

La Minería de Datos puede ser mejor descrita como la unión de progresos históricos y recientes en la estadística, AI, y el Aprendizaje de Máquina. Estas técnicas se utilizan juntas para estudiar datos y encontrar patrones y tendencias ocultas. La Minería de Datos está encontrando aceptación en la ciencia y las áreas comerciales que necesitan analizar cantidades grandes de datos para descubrir las tendencias que no podrían encontrar de otra manera.

---

<sup>5</sup> AI. (Artificial Intelligence) siglas en ingles de la inteligencia artificial.

## 1.2.4 Campos de aplicación de la minería de datos.

Los campos en los que se pueden utilizar las técnicas de Minería de Datos son muy variados, prácticamente se puede utilizar en cualquier campo donde se disponga de un conjunto de datos aportando importantes beneficios. En general, se dispone de un modelo que permita simular el comportamiento o predecir la evolución de un sistema, un proceso, las ventas de un producto, etc., teniendo con ello una ventaja competitiva, permitiendo adelantarse y aprovechar oportunidades, así como prevenir problemas. Algunas de las aplicaciones más comunes son:

### 1.2.4.1 Marketing

En esta área es donde los éxitos de la Minería de Datos son más conocidos. Se divide en:

*Targeting*, se puede aumentar el porcentaje de respuesta a una campaña de marketing si se dirige a los objetivos adecuados. La Minería de Datos permite detectar entre los potenciales clientes los que presentan una mayor posibilidad de responder a la campaña y dirigirla a ellos específicamente, reduciendo costos.

*Nuevas oportunidades de mercado*, La Minería de Datos permite detectar nuevas oportunidades de mercado, comparando hábitos de consumo de diferentes clientes o determinando la ubicación más conveniente para un determinado negocio.

#### 1.2.4.2 Predicción

A partir de los datos históricos almacenados y utilizando técnicas de Minería de Datos pueden elaborarse modelos que permitan estimar con precisión la evolución de una variable en el futuro. Esto puede resultar útil en diversos campos:

- Detección de oportunidades.
- Prevención de problemas.
- Gestión óptima de personal.
- Optimización de stocks.

#### 1.2.4.3 Reducción de riesgos

La minería de datos permite construir sistemas de evaluación automática de riesgos, basados en la experiencia previa, resultando de gran utilidad cuando la cantidad de casos a evaluar es excesiva para su procesamiento manual.

#### 1.2.4.4 Detección de Fraudes

Aplicando técnicas de minería de datos, pueden obtenerse modelos que permitan descubrir posibles fraudes, basándose en la detección de comportamientos anómalos, en comparación con los datos registrados anteriormente. Vemos aplicaciones concretas en operadores de telefonía o empresas de gestión de tarjetas de crédito.

#### 1.2.4.5 Control de calidad

Existen numerosos ejemplos en los que se han aplicado técnicas de minería de datos para desarrollar sistemas automáticos de control de calidad y facilitan:

*Detección más precisa de productos defectuosos.* La minería de datos permite desarrollar sistemas automáticos de control de calidad que discriminan los productos defectuosos con un alto grado de precisión y fiabilidad, según un criterio objetivo.

*Localización precoz de defectos.* Se utilizan técnicas de minería de datos para descubrir posibles relaciones que permitan detectar fallos utilizando las variables disponibles durante el proceso.

*Identificación de causas de fallos.* La minería de datos ayuda a identificar los fallos más frecuentes así como las causas que los originan, permitiendo adoptar medidas para evitarlos en el futuro.

*Análisis no destructivo.* Utilizando minería de datos es posible estimar con bastante exactitud parámetros en función de otras características que se pueden medir sin destruir el producto.

#### **1.2.4.6 Procesos industriales**

Otra aplicación básica de la minería de datos en el entorno industrial, además del control de calidad, es el control de procesos. Estas técnicas permiten explotar la información disponible sobre un sistema o proceso y utilizar los modelos desarrollados para:

- Automatizar y optimizar el control del proceso.
- Optimizar el rendimiento.
- Implementar programas de mantenimiento predictivo.

### 1.2.5 Operaciones y técnicas de la Minería de Datos

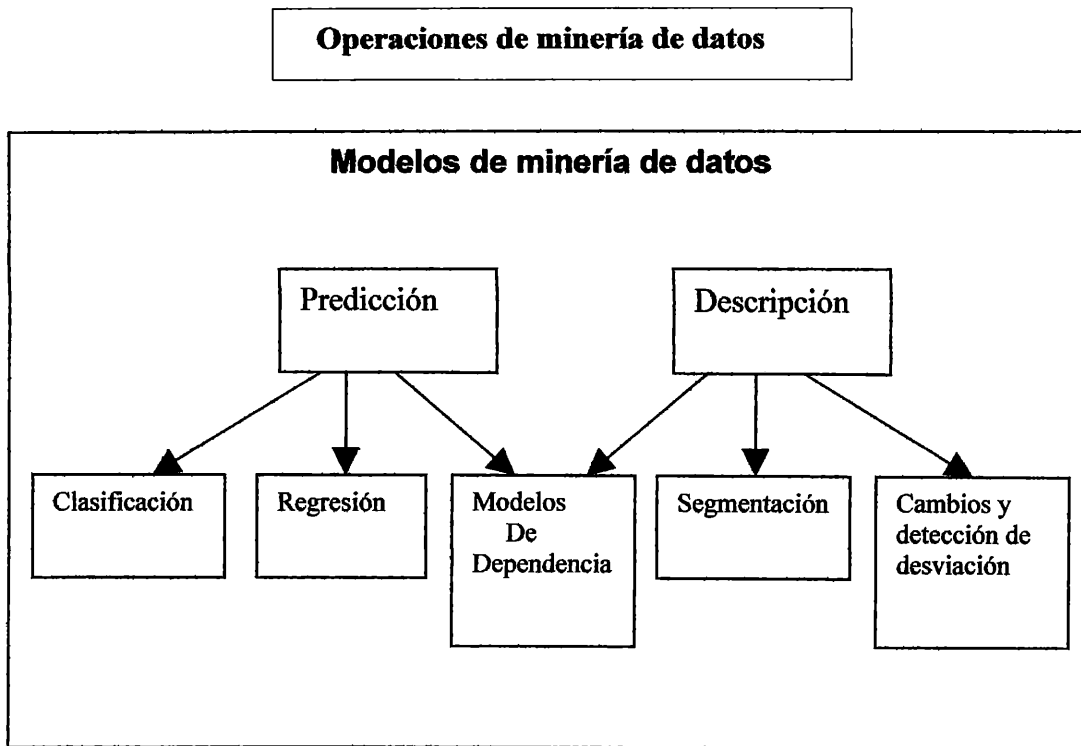
En general, la minería de datos puede ser clasificada en dos grandes categorías:

*Descripción*, encontrando patrones humanos-interpretables, asociaciones o correlaciones que describan los datos.

*Predicción*, Construyendo uno o más conjunto de datos (árboles de decisión, redes neuronales, vectores de soporte) ejecutando inferencias sobre la disponibilidad de ellos e intentando predecir el comportamiento de un nuevo grupo de datos.

La distinción entre los modelos de predicción y descripción no esta bien definida. Los modelos predictivos pueden ser también descriptivos y los modelos descriptivos pueden ser usados para la predicción. Estos modelos son asociados con las cinco operaciones básicas, las cuales son:

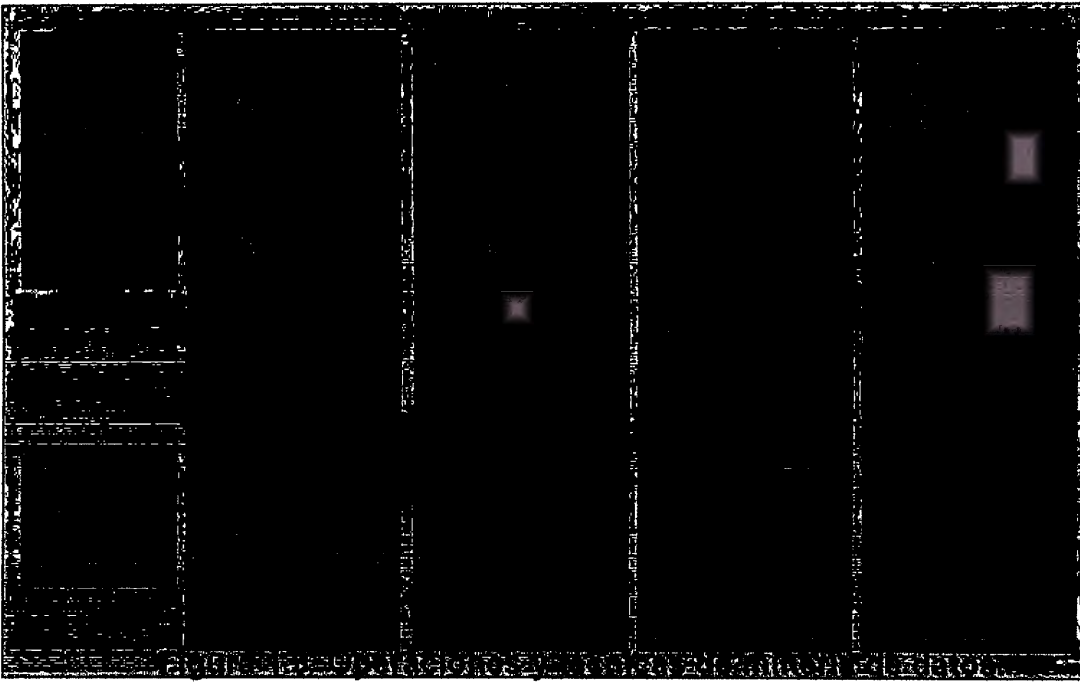
- Clasificación
- Regresión
- Modelos de dependencia
- Segmentación
- Cambios y detección de desviación.



**Figura 1.2 Conexión entre los modelos de minería de datos y sus operaciones.**

Mientras solo hay un par de modelos básicos, hay una amplia variedad de técnicas de minería de datos las cuales hacen estas operaciones posibles. Los sistemas de minería de datos no incluyen normalmente cada una de estas técnicas, pero frecuentemente se combinan dos o más técnicas diferentes entre las cuales el desarrollador puede escoger.

La siguiente figura presenta algunas de las técnicas más comunes asignadas a las operaciones básicas de la minería de datos, resaltando los problemas de regresión y clasificación.



Los dos tipos de problemas de predicción en ingeniería son Clasificación y Regresión. Ejemplos, observaciones de experiencias del pasado con atributos de conocimiento son examinados y generalizados para casos futuros. Clasificación va de la mano con el clustering, los cuales son para identificar clusters clavados en un espacio de datos multidimensionales, donde un cluster es una colección de objetos similares (grupos de datos que son similares unos a otros).

La regresión usualmente definida como un proceso de inducción de modelos de datos de el sistema (utilizando algunos algoritmos del Aprendizaje de Máquina) que serían capaces de predecir respuestas del sistema que será observado. Las respuestas del sistema son usualmente un valor real, mientras para la clasificación son niveles de clases.

### 1.3 Online Analytical processing (OLAP)

La BI (Business Intelligence)<sup>6</sup> conlleva la organización de los datos en distintas dimensiones potenciales de análisis de modo que se puedan mostrar y establecer referencias cruzadas entre cualesquiera vistas de datos (los resultados de ventas, por ejemplo) extraídos de cualesquiera otras dimensiones potenciales (como la región o la línea de producto). La posibilidad de pasar de una dimensión a otra permite descender al detalle o distanciarse para ofrecer una visión más general. La posibilidad de mostrar variaciones en los datos de distintas dimensiones (con frecuencia, de muchas dimensiones a la vez) permite generar informes multidimensionales en tiempo real. Este modo general de entender la manipulación de datos recibió el nombre de OLAP (OnLine Analytical Processing o Procesamiento analítico en línea), es decir, procesamiento de datos con fines analíticos en lugar de operativos. El término “on line” hace referencia al hecho de que los datos analíticos siempre se encuentran disponibles. OLAP garantiza que los datos contenidos en los almacenes de datos se encuentren siempre disponibles en un formato que permita su uso en tareas analíticas de apoyo a la toma de decisiones.

Además, OLAP se caracteriza esencialmente por el preprocesamiento, indexado y almacenamiento de datos en distintas representaciones dimensionales que le permiten generar con rapidez las distintas vistas que requiere la BI.

Pero las herramientas de OLAP que se utilizan en la BI no siempre detectan todas las pautas y dependencias que existen en los datos. Los cubos OLAP están indicados para la exploración de cantidades limitadas de datos y conllevan importantes variaciones en función de dimensiones empresariales críticas y conocidas. Pero cuando cambian las dimensiones como resultado de cambios en la empresa o de que se exploren situaciones nuevas, la tecnología del data mining puede resultar un complemento de

---

<sup>6</sup> "La tecnología de BI no es nueva, ha estado presente de varias formas por lo menos en los últimos 20 años, comenzando por generadores de reportes y sistemas de información ejecutiva en los 80's..." Goodwin, Candice. Net Library. <http://www.netlibraryebooktoolkit.com/>

OLAP enormemente potente y flexible. Las soluciones de data mining son ideales para la criba de cientos de dimensiones de análisis y de combinaciones asociadas que compiten entre sí y que son potencialmente útiles. Todos los algoritmos de data mining incluyen mecanismos capaces de detectar una gran cantidad de pautas posibles en los datos y de presentar los resultados de forma resumida. Los mecanismos de generación de informes de las soluciones de data mining y de OLAP para la BI están hechos el uno para el otro, y si se utilizan conjuntamente, actúan de forma sinérgica.

Los servicios OLAP incluyen un servidor de nivel medio que permite a los usuarios realizar análisis sofisticados en grandes volúmenes de datos con un desempeño excepcional. Una segunda característica de los servicios OLAP es un caché de cliente y una máquina de cálculo llamado el Servicio de PivotTable, el cual ayuda a mejorar el desempeño y reducir el tráfico de red. El servicio de PivotTable le permite a los usuarios conducir análisis mientras se encuentran desconectados de la red corporativa.

El OLAP es un componente clave en el proceso de almacenamiento de datos (*Data Warehouse*) y proporcionan la funcionalidad esencial para una gran variedad de aplicaciones que van desde reportes corporativos hasta soporte avanzado de decisiones.

### 1.3.1 La complejidad de los datos

Algunos retos pueden ser enfrentados con herramientas avanzadas de peticiones (*queries*), las cuales ocultan al usuario final la complejidad de la base de datos. Como sea, para una gran clase de aplicaciones donde el usuario final está viendo información multidimensional, se pensó crear la solución óptima, la cual es, la tecnología OLAP.

Todas las organizaciones tienen datos multidimensionales y la complejidad no es necesariamente una función del tamaño de la compañía. Aún a las más pequeñas compañías les gustaría poder rastrear sus ventas por producto, vendedor, geografía, cliente y tiempo. Las organizaciones han buscado durante mucho tiempo herramientas para acceder, navegar y analizar información multidimensional de una manera fácil y natural.

El OLAP no es un concepto nuevo, sino que se le ha dado este nombre a esta tecnología recientemente. En 1993, el Dr. E. F. Codd, investigador de bases de datos e inventor del modelo de bases de datos relacionales, acuñó el término en su artículo, "Providing OLAP to User Analysis: An IT Mandate", en donde dispuso las 12 reglas que definen las características de las aplicaciones OLAP. Nigel Pendse y Richard Creeth del Reporte del OLAP refinaron más tarde su definición con lo que se llama la prueba FASMI, lo que significa sencillamente que las aplicaciones OLAP deberían proporcionar análisis rápidos de información multidimensional compartida. Para partir este enunciado en sus partes:

- **Rápido.** Proporciona la información al usuario a una velocidad constante. La mayoría de las peticiones se deben de responder al usuario en cinco segundos o menos.
- **Análisis.** Realiza análisis estadísticos y numéricos básicos de los datos, predefinidos por el desarrollador de la aplicación o definido adhoc por el usuario.

- **Compartida.** Implementa los requerimientos de seguridad necesarios para compartir datos potencialmente confidenciales a través de una gran población de usuarios.
- **Multidimensional.** Llena la característica esencial del OLAP.
- **Información.** Acceso a todos los datos y la información necesaria y relevante para la aplicación, donde sea que ésta resida y no está limitada por el volumen.

OLAP proporciona a las organizaciones los medios para acceder, ver y analizar los datos con alto desempeño y flexibilidad. Primero, OLAP presenta los datos a los usuarios a través de un modelo de datos intuitivo y natural. Con este estilo de navegación, los usuarios finales pueden ver y entender más efectivamente la información de sus almacenes de datos, permitiendo así a las organizaciones reconocer mejor el valor de sus datos.

En segundo lugar, OLAP acelera la entrega de información a los usuarios finales que ven estas estructuras multidimensionales al preparar algunos valores computados en los datos por adelantado, en vez de al momento de ejecutarse. La combinación de navegación fácil y rápido desempeño le permite a los usuarios ver y analizar información más rápida y eficientemente, de lo que es posible con tecnologías de bases de datos relacionales solamente. El resultado final: se pasa más tiempo analizando los datos y menos tiempo analizando las bases de datos.

## 1.3.2 Términos y Conceptos de OLAP

### 1.3.2.1 Modelo de datos OLAP

En un modelo de datos OLAP, la información es vista como cubos, los cuales consisten de categorías descriptivas (dimensiones) y valores cuantitativos (medidas). El modelo de datos multidimensional simplifica a los usuarios el formular peticiones complejas, arreglar datos en un reporte, cambiar de datos de resumen a datos de detalle y filtrar o rebanar los datos en subconjuntos significativos. Por ejemplo, las dimensiones típicas de un cubo que contenga información de ventas, incluiría tiempo, geografía, producto, canal, organización y escenario (planeado o real). Las medidas típicas incluirían ventas en dólares (u otra moneda), unidades vendidas, número de personas, ingresos y gastos.



**Figura 1.4 Cubo de Datos.**

Dentro de cada dimensión de un modelo de datos OLAP, los datos se pueden organizar en una jerarquía que represente niveles de detalle de los datos. Una instancia particular del modelo de datos OLAP tendría valores para cada nivel en la jerarquía. Un usuario que vea datos OLAP se moverá entre estos niveles para ver información con mayor o menor detalle.

### 1.3.2.2 Modelos de agregación y almacenamiento

Los cubos, las dimensiones y las jerarquías son la esencia de la navegación multidimensional de OLAP. Al describir y representar la información en esta forma, los usuarios pueden navegar intuitivamente en un conjunto complejo de datos. Sin embargo, el solo describir el modelo de datos en una forma más intuitiva, hace muy poco para ayudar a entregar la información al usuario más rápidamente.

Un principio clave de OLAP es que los usuarios deberían de ver tiempos de respuesta consistentes para cada vista, o rebanada, de datos que requieran. Dado que la información se colecta en el nivel de detalle solamente, el resumen de la información es usualmente calculado por adelantado. Estos valores precalculados, o agregaciones, son la base de las ganancias de desempeño del OLAP.

En los primeros días de la tecnología OLAP, la mayoría de las compañías asumía que la única solución para una aplicación OLAP era un modelo de almacenamiento no relacional.

Después, otras compañías descubrieron que a través del uso de estructuras de base de datos (esquemas de estrella y de copo de nieve), índices y el almacenamiento de agregados, se podrían utilizar sistemas de administración de bases de datos relacionales (RDBMS) para el OLAP. Así, llamaron a esta tecnología OLAP relacional (ROLAP). Las primeras compañías adoptaron entonces el término OLAP multidimensional (MOLAP).

Las implementaciones MOLAP normalmente se desempeñan mejor que la tecnología ROLAP, pero tienen problemas de escalabilidad. Por otro lado, las implementaciones ROLAP son más escalables y son frecuentemente atractivas a los clientes debido a que aprovechan las inversiones en tecnologías de bases de datos relacionales preexistentes.

Un desarrollo reciente ha sido la solución OLAP híbrida (HOLAP), la cual combina las arquitecturas ROLAP y MOLAP para brindar una solución con las mejores características de ambas: desempeño superior y gran escalabilidad. Un tipo de HOLAP mantiene los registros de detalle (los volúmenes más grandes) en la base de datos relacional, mientras que mantiene las agregaciones en un almacén MOLAP separado.

### 1.3.3 Arquitectura OLAP

Los servicios OLAP consisten de componentes de software del servidor y del cliente (nivel medio, *middle-tier*).

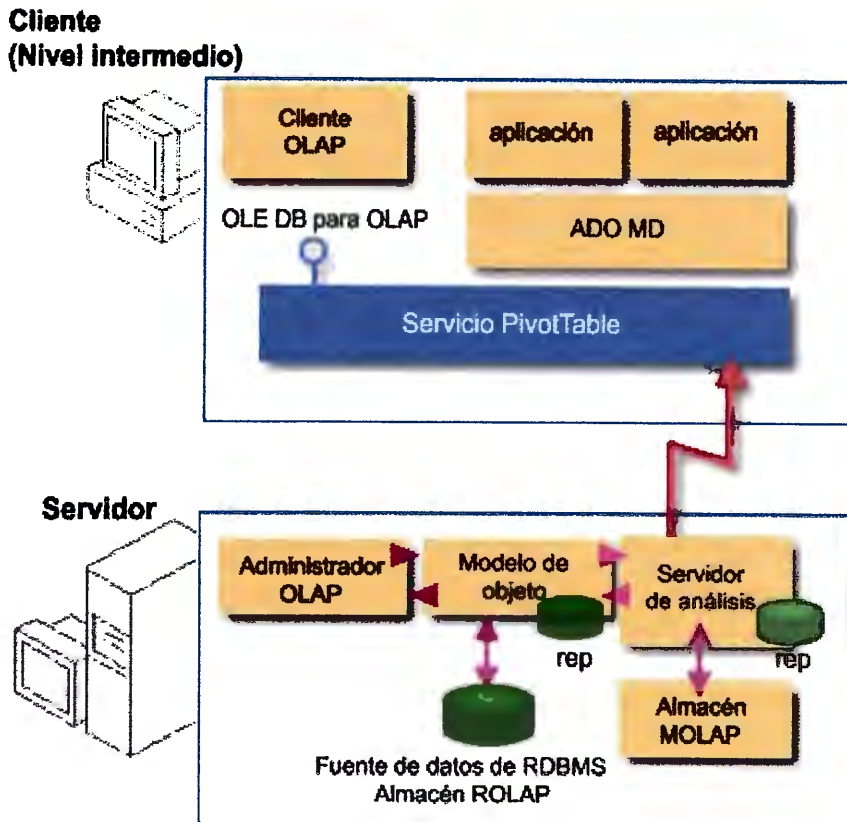


Figura 1.5 Arquitectura OLAP.

El servidor, proporciona la funcionalidad computacional principal. El acceso programático a funciones administrativas en el servidor OLAP es a través de un modelo de objeto llamado Objetos de Soporte de Decisión (*Decision Support Objects, DSO*)<sup>7</sup>.

<sup>7</sup> DSO se pueden usar en aplicaciones para administrar el servidor, crear y mantener OLAP y objetos de minería de datos.

El Administrador OLAP (OLAP Manager), la interfase administrativa interconstruida para los servicios OLAP, está también desarrollada sobre DSO y proporciona una experiencia rica para el usuario sin necesidad de programar. El Administrador OLAP, el cual se puede ejecutar en una computadora separada del servidor OLAP, permite al administrador de la base de datos diseñar modelos de datos OLAP, acceder a la información en almacenes RDBMS, diseñar agregaciones y llenar los almacenes de datos OLAP, entre otras funciones. Las definiciones de metadatos OLAP son almacenadas en un depósito privado, pero pueden ser exportadas al Microsoft Repository con el Modelo de Información Abierta de OLAP (*Open Information Model, OIM*) utilizando una sencilla herramienta.

Los servicios OLAP pueden acceder fuentes de datos en cualquier proveedor de datos compatible con OLE DB, cualquier base de datos que proporcione una interfase de Conectividad de Base de Datos Abierta (*Open Database Connectivity, ODBC*) es también accesible a través de una función OLE DB que encapsula a los controladores ODBC y los exponen como si fueran interfases OLE DB nativas. A través de las capacidades multiplataforma de OLE DB, los datos se pueden acceder desde una amplia variedad de sistemas como si fueran locales al servidor de servicios de OLAP.

En el lado del cliente, Los servicios OLAP incluyen un componente llamado el Servicio PivotTable. El Servicio PivotTable es la función que conecta las aplicaciones cliente de OLAP con el servidor OLAP. Todos los accesos a los datos administrados por los servicios OLAP, por programas personalizados o herramientas cliente, son a través de la interfase OLE DB para OLAP provista por el Servicio PivotTable.

Los componentes cliente y servidor de los servicios OLAP son extensibles en funcionalidad. Los sitios del cliente, desarrolladores de software independientes y consultores, pueden todos extender la funcionalidad, la administración de datos y el cálculo de la aplicación utilizando características bien documentadas en DSO.

## 1.4 MDX

MDX, acrónimo del inglés Multidimensional Expressions (expresiones multidimensionales), es una sintaxis que admite la definición y manipulación de objetos y datos multidimensionales. MDX<sup>8</sup> es similar en muchos aspectos a la sintaxis de SQL (Structured Query Language, lenguaje de consulta estructurado), pero no es una extensión del lenguaje SQL; de hecho, SQL puede ofrecer parte de la funcionalidad que ofrece MDX, aunque no con tanta eficacia ni de manera tan intuitiva.

Al igual que con una consulta SQL, cada consulta MDX requiere una solicitud de datos (la cláusula SELECT), un punto de inicio (la cláusula FROM) y un filtro (la cláusula WHERE). Éstas y otras palabras clave proporcionan las herramientas utilizadas para extraer partes específicas de datos de un cubo para su análisis.

MDX proporciona también un sólido conjunto de funciones para la manipulación de datos recuperados, así como la capacidad de ampliar MDX con funciones definidas por el usuario.

MDX, al igual que SQL, proporciona sintaxis del lenguaje de definición de datos (DDL) para administrar estructuras de datos. Hay comandos MDX para crear y eliminar cubos, dimensiones, medidas y sus objetos subordinados.

---

<sup>8</sup> MDX proporciona más de 100 funciones para agregar, modificar o consultar cubos de datos.

### 1.4.1 Diferencias entre SQL y MDX

La principal diferencia esta en la habilidad que tiene MDX para consultar múltiples dimensiones. Aunque es posible utilizar consultas SQL exclusivamente para cubos en el Análisis Service de SQL, MDX da comandos que están específicamente diseñados para recuperar datos en estructuras multidimensionales con cualquier número de dimensiones.

El proceso para crear una consulta en SQL es también diferente que al crear una en MDX. Al crearla en SQL se define y visualiza una estructura en dos dimensiones, escribiendo una consulta para poblar de datos una o más tablas. Al contrario, en MDX se define y visualiza una estructura multidimensional, realizando la consulta para poblar un cubo de datos. Resultando un conjunto de datos multidimensionales con cualquier número de dimensiones.

Aunque SQL y MDX tengan similar sintaxis. En MDX es notablemente robusta y puede ser compleja, sin embargo, fue diseñado para consultar datos multidimensionales de una forma fácil y efectiva.

## **CAPITULO II. TECNICAS PROPUESTAS A UTILIZAR Y SITUACIÓN ACTUAL.**

---

Este capítulo da a conocer las técnicas para el desarrollo de la herramienta, explicamos el porque el uso de la plataforma seleccionada, se explican los elementos claves de la arquitectura OLAP relacionados con el software utilizado. Se presenta la situación actual de la empresa tomada como referencia para el desarrollo de la herramienta, como es el equipo con que cuenta, se hace un análisis de los datos que manejan y se mide el porcentaje de efectividad de los mismos.

## II. TÉCNICAS PROPUESTAS A UTILIZAR Y SITUACIÓN ACTUAL.

### 2.1 Técnicas a utilizar.

Debido al poco desarrollo que se le ha dado a la minería de datos en nuestro país y a la poca información especializada encontrada en libros, toda la información que se encontró referente a minería de datos fue en Internet. Con respecto a los servicios OLAP, se encontró mas bibliografía de forma convencional y en Internet, este tema se ha desarrollado más comúnmente que la minería de datos, establecemos la técnica y modelo de minería de datos a utilizar, porque utilizarla, y la ventaja de trabajar con servicios OLAP.

Una vez adquirido los conocimientos para realizar la herramienta que se plantea desarrollar, se tiene que elegir el lenguaje de programación más adecuado, el administrador de base de datos, así como el modelo y técnica de minería de datos a utilizar. También ampliamos los componentes principales de la arquitectura OLAP utilizadas por el administrador de base de datos y el lenguaje de programación.

#### **Plataforma Windows.**

- Administrador de bases de datos: SQL Server 2000.
- Lenguaje de programación: Visual Basic 6.0

#### **¿Por que SQL Server 2000?**

Ya que SQL Server 2000 esta dotado de una importante infraestructura de minería de datos y sus algoritmos básicos que permiten la realización de tareas de análisis de clusters y de árboles de decisión, aumentando las prestaciones de los Servicios OLAP.

Además, la biblioteca de Objetos de Apoyo de Decisión (Decision Support Objects - DSO) de Análisis Service de Microsoft SQL Server 2000, proporcionan un juego robusto de Modelo de Objeto de Componente<sup>9</sup> (COM), que son objetos e interfaces que se pueden usar para crear aplicaciones para administrar por medio de programas los objetos Análisis Service. También cuenta con el servicio de PivotTable, que es la interfaz primaria para aplicaciones que actúan recíprocamente con Análisis Service Microsoft SQL Server 2000, utilizado para construir aplicaciones del cliente que actúan recíprocamente con los datos multidimensionales.

### ¿Por que Visual Basic 6.0?

Ya que es un excelente programa que interactúa de gran manera con SQL Server 2000, permitiendo tener al o a los usuario(s) un excelente entorno gráfico y reduciendo problemas de conectividad y compatibilidad.

También los Objetos de Apoyo de Decisión (DSO) pueden ser usados con cualquier lenguaje que apoye la Automatización de COM, como: Microsoft Visual Basic.

Las aplicaciones para el cliente de el servicio PivotTable, pueden desarrollarse ejecutando una variedad de técnicas y entornos. Se utiliza Microsoft ActiveX Data Objects (Multidimensional) (ADO MD) para implementar las aplicaciones del cliente utilizando cualquier Modelo de Componentes de Objetos (COM) de Visual Basic.

El servicio de PivotTable, el cual juega un papel importante en la arquitectura OLAP y el Administrador OLAP (OLAP Manager), la interfase administrativa interconstruida para los servicios OLAP, desarrollada sobre DSO, son soportadas tanto como por el administrador de base de datos Microsoft SQL Server 2000, así como por el lenguaje de programación Microsoft Visual Basic 6.0, y son de las razones principales por los cuales fueron seleccionados para el desarrollo de la herramienta.

---

<sup>9</sup> Es la traducción que nosotros hemos dado a la palabra en ingles **Component Object Model**.

El modelo de minería de datos propuesto a utilizar será el de Predicción, construyendo uno o más conjunto de datos y prediciendo el comportamiento de un nuevo conjunto de datos.

La técnica propuesta a utilizar será la de los árboles de decisión, ya que es una de las más comúnmente utilizadas en los problemas de predicción en ingeniería e identifica los requerimientos de datos críticos que rodean el proceso de decisión. Otra razón de la elección de las plataformas es que, se puede trabajar con esta técnica, por medio del administrador de base de datos y en el lenguaje de programación seleccionados.

Los árboles de decisión pueden ser definidos como un diagrama que representa en forma secuenciales acciones y condiciones, permitiendo mostrar las relaciones entre cada condición y el grupo de acciones permisibles asociadas con ellas. Alguna de las características de esta técnica es:

- Representa funciones Booleanas.
- Cada nodo en el árbol corresponde a una prueba sobre el valor de una de las propiedades.
- Las ramas de los árboles corresponden a los posibles valores de las propiedades.
- Cada nodo hoja del árbol representa una clasificación del registro.

Algunas de las herramientas utilizadas en los árboles de decisión son:

- See5
- CART
- MineSet
- Enterprise Miner.

## **Servidores OLAP**

OLAP garantiza que los datos almacenados en el Data Warehouse se encuentren siempre disponibles en un formato que permita su uso en tareas analíticas de apoyo a la toma de decisiones. Se caracteriza por el procesamiento, indexado y almacenamiento de datos en distintas representaciones dimensionales generándolas con rapidez.

### **¿Por qué utilizar OLAP?**

Las herramientas OLAP no siempre detectan todas las pautas y dependencias que existen en los datos. Los cubos OLAP están indicados para la exploración de cantidades limitadas de datos y conllevan importantes variaciones en función de dimensiones empresariales críticas y conocidas.

La tecnología de la minería de datos puede resultar un complemento de OLAP de enorme potencial y flexible, cuando cambian las dimensiones como resultado de cambios en la empresa o que se exploren situaciones nuevas.

Todos los algoritmos de la minería de datos incluyen mecanismos capaces de detectar una gran cantidad de pautas posibles en los datos y representar los resultados de forma resumida. Si se utilizan conjuntamente los mecanismos de generación de informe de las soluciones de la minería de datos y de OLAP actúan de forma sinérgica. Es por eso, que ya hay en el mercado un programa de desarrollo que mezcla la funcionalidad de la minería de datos con las herramientas OLAP.

## **2.2 Elementos claves de la arquitectura OLAP relacionados con el software a utilizar.**

### **2.2.1 PivotTable**

El Servicio de PivotTable es la interfaz primaria para aplicaciones que actúan recíprocamente con Análisis Service Microsoft SQL Server 2000, realiza tareas como la conexión y consulta de un cubo o un modelo de minería de datos, y recupera la información del esquema. Se usa para construir aplicaciones del cliente que actúan interactivamente con los datos multidimensionales. PivotTable Service también mantiene métodos para el análisis de datos en línea y fuera de línea de minería de datos multidimensionales y datos correlativos. El Servicio de PivotTable es incluido como la parte del Análisis Service, y puede redistribuirse por las aplicaciones de cliente a terceros.

Como proveedor autosuficiente, el Servicio de PivotTable proporciona la habilidad a las aplicaciones del cliente de crear los archivos de cubos locales y los modelos de minería de datos (con la sentencia CREATE MINING MODEL) relacionales y multidimensionales. Los modelos de minería de datos pueden ser creados:

en el servidor, usando los Objetos de Apoyo de Decisión (DSO), o localmente usando la sentencia CREATE MINING MODEL.

Los modelos pueden ser preparados y usados para producir predicciones basadas en tendencias identificadas en la selección de información.

El Servicio de PivotTable es un proveedor OLE DB para datos multidimensionales y modelos de minería de datos. Esto significa que mantiene la funcionalidad de aplicaciones OLE DB que necesitan el acceso a los datos multidimensionales y servicios de minería de datos. Apoyándose de un subconjunto de sentencias SQL y de las Expresiones Multidimensionales (MDX), permite a las aplicaciones recuperar datos

tabulares y multidimensionales. Los datos pueden desplegarse, incluidos los de un cubo local, analizados o actualizados usando sofisticados algoritmos de data mining.

La velocidad de la ejecución de estas operaciones complejas permite realizar análisis sofisticados en la propia computadora del cliente. Esto, a su vez, permite a las aplicaciones remotas del cliente funcionar independientemente de una red de gran velocidad, conexión de la intranet, o la presencia física a una zona geográfica determinada. Por ejemplo, un representante de las ventas puede prever los márgenes de ganancia al sitio de un cliente potencial basado en las propuestas de él, aun cuando su computadora no se conecta a la red de su propia compañía. También puede repoblar el cubo local con los nuevos datos siempre que se necesite, usando Internet para conectarse a su servidor del Análisis corporativo y actualizando los datos.

PivotTable Service también apoya los DDL de la cadena de conexión de la aplicación del cliente para que los clientes desconectados puedan crear y modificar los cubos locales fuera de línea y define los objetos multidimensionales temporales para el uso en el análisis. Esta habilidad permite a las aplicaciones del cliente ser extensibles ante las prácticas de negocio cambiantes y por los límites comerciales. Pueden derivarse los datos para los cubos locales de una fuente de datos multidimensional o relacional como Microsoft SQL Server, Microsoft Access, u Oracle.

### **2.2.1.1 Seguridad de los Servicios PivotTable.**

*Seguridad a los cubos*, Permisos de lectura y lectura/escritura para las bases de datos y cubos.

*Seguridad de miembros*, visibles solo a los usuarios que han sido dados los permisos para leerlos.

*Seguridad de celdas*, sentencias que involucran a una o más celdas con seguridad, devuelven un código o valor de error, esto depende de la propiedad Secured Cell Value, de la cadena de conexión.

El servicio de PivotTable tiene tres formas de conectarse a las fuentes de datos:

Conexión a un servidor de Internet, a una interfaz de proveedor de servicios seguros y utilizando roles de usuarios diferentes.

### **2.2.1.2 Desarrollando aplicaciones para el cliente.**

El servicio de PivotTable es el método primario de comunicación entre las aplicaciones cliente y modelos de minería de datos, como Análisis Service Microsoft SQL Server 2000. Usado por aplicaciones de Microsoft (como el Microsoft Excel PivotTable y PivotChart) y aplicaciones de terceros, como vendedores.

Las aplicaciones cliente pueden recuperar, desplegar y manipular datos desde fuentes multidimensionales, como cubos OLAP o modelos de minería de datos, los cuales pueden ser creados en la computadora local y llenarlos con datos provenientes de un proveedor OLE DB, como SQL Server o un cubo OLAP. Después de que son creados, las aplicaciones pueden examinar y analizar los datos que contienen.

Las aplicaciones cliente implementadas con lenguaje de automatización COM, como Visual Basic, pueden acceder a los Servicios PivotTable usando Microsoft ActiveX Data Objects (Multidimensional) (ADO MD), o usando las interfaces COM siempre por OLE DB para OLAP.

Microsoft Query es utilizado para crear un reporte de Excel usando PivotTable, teniendo un cubo de OLAP como fuente de datos. Las consultas proporcionan la capacidad de establecer las conexiones a los cubos. El componente Microsoft Office es opcional, por lo tanto, se necesitara determinar su existencia dentro de la PC.

También se necesitará el proveedor Microsoft OLAP, incluido en una típica instalación de Excel 2000, el cual consiste de el Data Source Driver y el software del cliente necesario para acceder a cubos creados por Análisis Service Microsoft SQL Server 2000.

### **2.2.1.3 Entorno de desarrollo utilizando Visual Basic**

Microsoft ActiveX Data Objects (Multidimensional) (ADO MD) provee fácil acceso a datos multidimensionales, se puede examinar un esquema multidimensional, consultar un cubo y recuperar datos.

La librería Microsoft ActiveX Data Objects (Multidimensional) (ADO MD) tiene un cierto número de objetos que se pueden utilizar con los Servicios de PivotTable. El objeto *Cellset* permite editar consultas MDX<sup>10</sup> contra cubos en el Análisis Service o archivos de cubo local. El objeto *CubeDef* habilita para recuperar esquemas de información multidimensionales.

---

<sup>10</sup> Sintaxis usada para definir objetos y manipular datos multidimensionales.

Para usar la librería ADO MD independientemente de la librería ADO, se usa la propiedad ActiveConnection de los objetos Catalog o Cellset .

Cellset, es el resultado de una consulta de datos multidimensionales.

Cuando utilizamos la librería ADO, esta brinda un objeto Connection, que provee métodos como: Open, OpenSchema, y Execute.

ADO y ADO MD están relacionados, pero son modelos de objetos separados. ADO provee objetos para conectar a la fuente de datos, ejecutar comandos, recuperar datos tabulares y esquemas de metadatos en un formato tabular. ADO MD da objetos para recuperar datos multidimensionales y vistas de esquemas de datos multidimensionales.

### **2.2.2 Objetos de Apoyo de Decisión - Decision Support Objects - (DSO).**

Análisis Service Microsoft SQL Server 2000, proporciona una gama amplia de proceso analítico en línea (OLAP) y la funcionalidad de minera de datos. El diseño es muy flexible y extensible, también se pueden agregar servicios de tres dimensiones y paquetes, como los proveedores de los algoritmos de minería de datos, incluso extender más allá las capacidades de Servicios de Análisis. Sin embargo, con una amplia gama de funcionalidad viene una amplia gama de complejidad, y los Servicios de Análisis (Analysis Services) proporcionan un complejo y robusto juego OLAP y características de minería de datos.

Para acceder a la amplia gama, de rasgos de gran alcance en una moda simple y sincera, la biblioteca de los Objetos de Apoyo de Decisión (DSO) proporcionan un objeto de modelo jerárquico, para usar con cualquier ambiente de desarrollo que puede apoyar a los objetos e interfaz del Modelo de Componentes de objetos (COM), como Microsoft Visual C++, Microsoft Visual Basic, y Microsoft Visual Basic Scripting Edition.

Uno de los rasgos del modelo de objetos DSO, es el uso de la interfaz de MDStore. Esta interfaz compartida permite desarrollar entornos que usan el enlace tardío, como Microsoft Visual Basic Scripting Edition, para apoyar fácilmente el complejo modelo jerárquico. La interfaz de MDStore se usa en DSO por objetos que proporcionan la funcionalidad para base de datos, cubos, particiones, y agregaciones en los Servicios de Análisis (Analysis Services).

#### **¿Por qué Usar DSO?**

DSO es el modelo de objeto usado para crear y manipular objetos OLAP en el Análisis Service por medio de programas, así como el Análisis Manager se utiliza para crear y manipular estos objetos a través de una interfaz del usuario. Usando DSO se puede tener más control sobre estas operaciones. Se podría empaquetar en un conjunto del

Análisis Service las tareas específicas en una unidad ejecutable para ser usadas en una fecha más tarde. Por ejemplo, se podría crear código para realizar una partición de OLAP de un juego de transacciones de la tienda de comestibles como una tarea automatizada al final de cada semana. DSO permite esconder las complejidades de crear las bases de datos, asignando las fuentes de datos, creando los cubos, y así sucesivamente. En este caso, DSO podría usarse para ejecutar simplemente un subconjunto de los funcionamientos disponible a través del Análisis Manager.

Dentro del DSO y Análisis Service, se encuentra Microsoft ActiveX Data Objects Multidimensional (ADOMD). ADOMD es un conjunto de extensiones a la interfaz del programa de la aplicación (API), la cual se usa para las consultas de las fuentes de los datos multidimensionales. ADOMD usa las Expresiones Multidimensionales (MDX) para preguntar las fuentes de los datos multidimensionales a través de ADOMD.

### **Usando DSO.**

Los Objetos de Apoyo de Decisión (DSO) se pueden usar en aplicaciones para administrar el servidor, crear y mantener OLAP y objetos de minería de datos. Los objetos DSO también pueden crearse e incorporarse a las funciones del Análisis Service de Microsoft SQL Server 2000 en las aplicaciones del cliente.

Una sucesión común de funcionamientos para una aplicación que usa DSO es:

- Conectar a un Análisis Service.
- Crear un objeto de la base de datos para contener dimensiones y cubos.
- Agregar una fuente de datos que contenga los datos.

- Crear dimensiones y sus niveles.
- Crear un cubo y especificar dimensiones y medidas.
- Procesar un cubo para cargar su estructura y datos.

### **2.2.2.1 Arquitectura a los Objetos de Apoyo de Decisión**

Los Objetos de Apoyo de decisión (DSO) es una biblioteca del Modelo de Componente de Objeto (COM), las clases e interfaces proporcionan el acceso al Servidor de Análisis (Analysis Services). Estas clases e interfaces, cuando se usan juntos, forman un modelo de objeto que corresponde a la estructura interior de los objetos manejado por Análisis Service Microsoft SQL Server 2000 y pueden ser manejados por medio de programación.

Conceptualmente, DSO usa grupos de objetos jerárquicamente colocados para definir elementos básicos de datos de almacenamiento del Análisis Service, implementados por el servidor. Estos elementos básicos son base de datos, las fuentes de los datos, las dimensiones, los cubos, los modelos de minería de datos, y roles. DSO mantiene estos elementos básicos en una estructura jerárquica donde cada uno contiene otro elemento en un árbol, con el objeto del servidor a la raíz del árbol. Otros objetos apoyan esta estructura básica. Por ejemplo, las bases de datos, cubos, particiones, y agregaciones apoyan las dimensiones. El diagrama siguiente muestra la jerarquía ejemplar a una apreciación global del objeto de DSO.

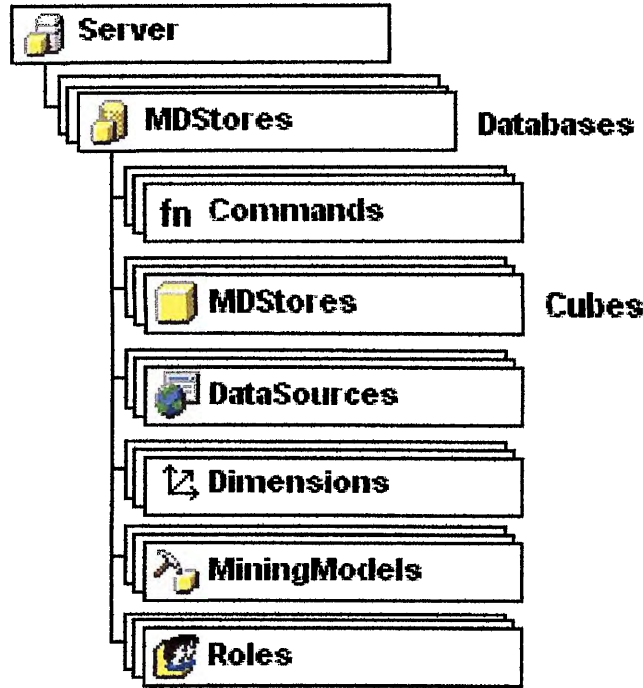


Figura 2.1 Jerarquía del objeto DSO

El objeto Servidor DSO contiene una colección que define las bases de datos accedida por el servidor. Cada banco de datos puede contener grupos de objetos que definen cubos, cubos unidos, o los cubos virtuales. Un *cubo*, contiene una o más particiones que contienen una o más agregaciones. Los *cubos unidos*, sirven para proporcionar el acceso del servidor local a un cubo en otro servidor; el servidor remoto publica el cubo, y el servidor local se suscribe a él creando un cubo unido. Un *cubo virtual*, es un caso especial de un cubo, combinando porciones de los cubos que contiene, similar a una vista en una base de datos relacional que combina porciones de tablas. Una base de datos también puede contener uno o más relaciones o modelos de minería de datos OLAP, representados en DSO por el objeto Mining Model. Estos modelos pueden contener uno o más datos de las columnas de la minería. Las bases de datos también pueden contener los roles, usados para manejar la seguridad sobre ellas y sus cubos asociados y datos de los modelos de minería.

## **Interfaz MDStore**

En el diagrama se nota que dos de las colecciones más importantes de objetos en DSO, bases de datos y cubos, se encuentran en una colección nombrada MDStores.

Bases de datos, cubos, particiones, y agregaciones exponen una interfaz común, llamada MDStore que proporcionan los métodos y propiedades que se usan para manipularlos. Estos deben provenir del objeto padre, y no pueden crearse independientemente; la única manera de crearlos, es a través de la colección MDStores del objeto padre. Por ejemplo, la única manera de obtener una referencia del objeto a una base de datos en DSO, o para crear un nuevo banco de datos que usa DSO, es a través la colección MDStores del objeto Servidor DSO.

Esta colección especial mantiene las referencias a los objetos que constituyen los elementos de esta jerarquía bajo el servidor. La colección de MDStores proporciona las aplicaciones especiales de agregación, búsqueda, y eliminación de los métodos, y un método de AddNew conveniente que mantiene las relaciones de padre-hijo entre los varios objetos de DSO. La propiedad de ContainedClassType determina qué tipo de objetos puede contener la colección de MDStores; el valor de esta propiedad corresponde directamente a la propiedad de ClassType de MDStore contenida por la colección.

La interfaz de MDStore también usa la propiedad de SubClassType, usada para diferenciar un cubo, un cubo unido, y un cubo virtual.

En el anexo A.1, se muestra una vista extendida de la estructura de DSO, como sus colecciones, incluso el valor de la propiedad ClassType de los objetos contenido en cada colección MDStore..

La interfaz MDStore se lleva a cabo por objetos<sup>11</sup> en el DSO que contienen datos multidimensionales. Aunque todos estos objetos llevan a cabo sus propias interfaces interiores, la de MDStore es la primaria a ser usada por estos objetos. Para diferenciar entre los objetos de la interfaz MDStore, usa la propiedad ClassType. La tabla siguiente lista los objetos que llevan a cabo la interfaz MDStore y los valores asociados de la propiedad ClassType.

Objeto	Class type
Database	clsDatabase
Cube	clsCube
Partition	clsPartition
Aggregation	clsAggregation

### 2.2.2.2. Objetos DSO

Hay dos clasificaciones principales de objetos en DSO: los que se pueden acceder y manejar usando su interfaz predefinida directamente, y los objetos que implementan otro DSO además de su interfaz predefinida.

Los objetos que se pueden acceder y manejar directamente tienen sus propias colecciones, métodos, y propiedades, y estos llevan a cabo su interfaz predefinida totalmente; no hay ningún cambio en la conducta de la interfaz basada en su uso o asociación padre-hijo. Por ejemplo, el objeto DSO DataSource se comporta de la misma manera si el padre es el objeto DSO Servidor o el objeto DSO Cube.

Los objetos que se llevan a cabo con más de un uso de la interfaz, tienen un subconjunto de colecciones, métodos, y propiedades asociadas con la interfaz por la variación implementada de un objeto DSO; por ejemplo, la interfaz *Command* se lleva a cabo diferentemente para los comandos de la base de datos, cubos, y roles. Cada colección, método y descripción de propiedad contiene los nombres de los objetos en

<sup>11</sup> En la tabla A.2 del anexo A se presenta los objetos DSO.

que aparece. Recíprocamente, cada descripción del objeto contiene los nombres de las colecciones, métodos, y propiedades que se llevan a cabo.

### **2.2.2.3 Interfaces DSO.**

Hay varias interfaces en los Objetos de Apoyo de Decisión (DSO). Objetos que tienen similar funcionalidad implementan una interfaz común.

Por ejemplo, bases de datos, cubos, particiones, y agregaciones llevan a cabo la interfaz de MDStore. Un objeto de MDStore es un recipiente de datos multidimensionales. Las bases de datos contienen cubos de información relacionada, estos contienen particiones que guardan los datos, y las agregaciones son los resúmenes precalculados de datos asociados con las particiones. Los objetos de MDStore tienen estructuras similares, contienen colecciones de dimensiones que categorizan los datos, las fuentes de datos que especifican qué administrador de sistemas de base de datos relacionales (RDBMS) contiene la tabla hecho y tablas de las dimensiones, los roles que definen los permisos de seguridad, y así sucesivamente.

Dado una referencia a una interfaz de MDStore o cualquier otra interfaz DSO, se puede determinar qué tipo de objeto está examinando mediante la propiedad `ClassType`. Los objetos que llevan a cabo la interfaz pueden tener los tipos de la clase siguiente: `clsDatabase`, `clsCube`, `clsPartition`, y `clsAggregation`.

No todos los objetos que llevan a cabo un instrumento de la interfaz lo implementan de la misma manera. Algunos no llevan a cabo todas las propiedades de la interfaz, métodos, y colecciones. Por ejemplo, el objeto base de datos (el `clsDatabase`) implementa el método `BeginTrans` de la interfaz común del MDStore, pero los objetos del cubo (`clsCube`) no lo hacen. Algunos objetos restringen el acceso a ciertas propiedades para que ellas solo sean de lectura en lugar de lectura/escritura. Si se intenta acceder a una propiedad o invocar un método que no se lleva a cabo, DSO levanta un error.

Algunos objetos de DSO no llevan a cabo una interfaz común. Se puede acceder a estos objetos usando su interfaz predefinida: el `clsServer`, `clsDataSource`, `clsMiningModel`, `clsColumn`, `clsCubeAnalyzer`, `clsPartitionAnalyzer`, y `clsMemberProperty`.

### 2.3 Equipo actual (situación actual de la empresa) y estadística de datos.

BIOLACT cuenta con equipo tecnológico de oficina moderno, sofisticado, nuevo y disponible para el departamento de administración, todos los artículos se detallan de la siguiente forma:

Cantidad	Descripción
7	Computadoras Personales
2	Impresoras Láser
2	Scanner

Todas las computadoras poseen componentes internos muy actualizados, para mencionar algunos tenemos:

Cantidad	Descripción (Hardware)
1	Microprocesador Pentium IV
1	Tarjeta de Red
1	Tarjeta de video
1	Tarjeta de Sonido
2	Dimms de 128 M
1	Disco Duro de 80 G

BIOLACT además cuenta con un centro de cómputo común para todas las microempresas del área, las características del equipo se especifican a continuación:

Cantidad	Descripción
40	Computadoras Personales
4	Impresoras Láser
1	Scanner

## 2.4 Análisis actual de los datos

En BIOLACT los datos son analizados de la siguiente manera:

Los registros de clientes, productos, etc. son llevados por los administradores de la empresa quienes se encargan de analizar la información, estos datos son almacenados en hojas de cálculo.

La desventaja de almacenar la información en hojas de cálculo sin poseer un administrador de bases de datos es que la información que se tiene es completamente estática y no es contemplada para análisis más profundos de información.

Un ejemplo sería el de los registros de los clientes:

BIOLACT						
Formulario de Servicio al Cliente				N° 1254		
Cliente	Dirección	Teléfono	Unidades(Lbs)	Descripción	Fecha Pedido	Fecha Entrega
Hotel Siesta	XXXXXXXXXX	999-9999	23	Queso Camembert	XXXXXXXXXX	XXXXXXXXXX
Hotel Tesoro Beach	XXXXXXXXXX	999-9999	21	Queso Semicurado	XXXXXXXXXX	XXXXXXXXXX
Hotel Paradise	XXXXXXXXXX	999-9999	35	Queso Curado	XXXXXXXXXX	XXXXXXXXXX
Hotel Radisson	XXXXXXXXXX	999-9999	42	Queso Sedar	XXXXXXXXXX	XXXXXXXXXX
Hotel Presidente	XXXXXXXXXX	999-9999	40	Queso Burgos	XXXXXXXXXX	XXXXXXXXXX
Hotel Intercontinental	XXXXXXXXXX	999-9999	38	Queso Fresco	XXXXXXXXXX	XXXXXXXXXX
Total			199			
Precio			\$1,194.00			

Como se puede ver no existe un mecanismo que permita una búsqueda de patrones en la información existente.

De igual manera se manejan los formularios de ventas, compras, etc. por lo que se facilitaría el análisis de información mediante la implantación de un sistema que permita tecnologías como Servidores OLAP y minería de datos, esto aumentaría considerablemente los tiempos de respuesta de los datos en consultas y se disminuirían variables como el tiempo, costos, etc.

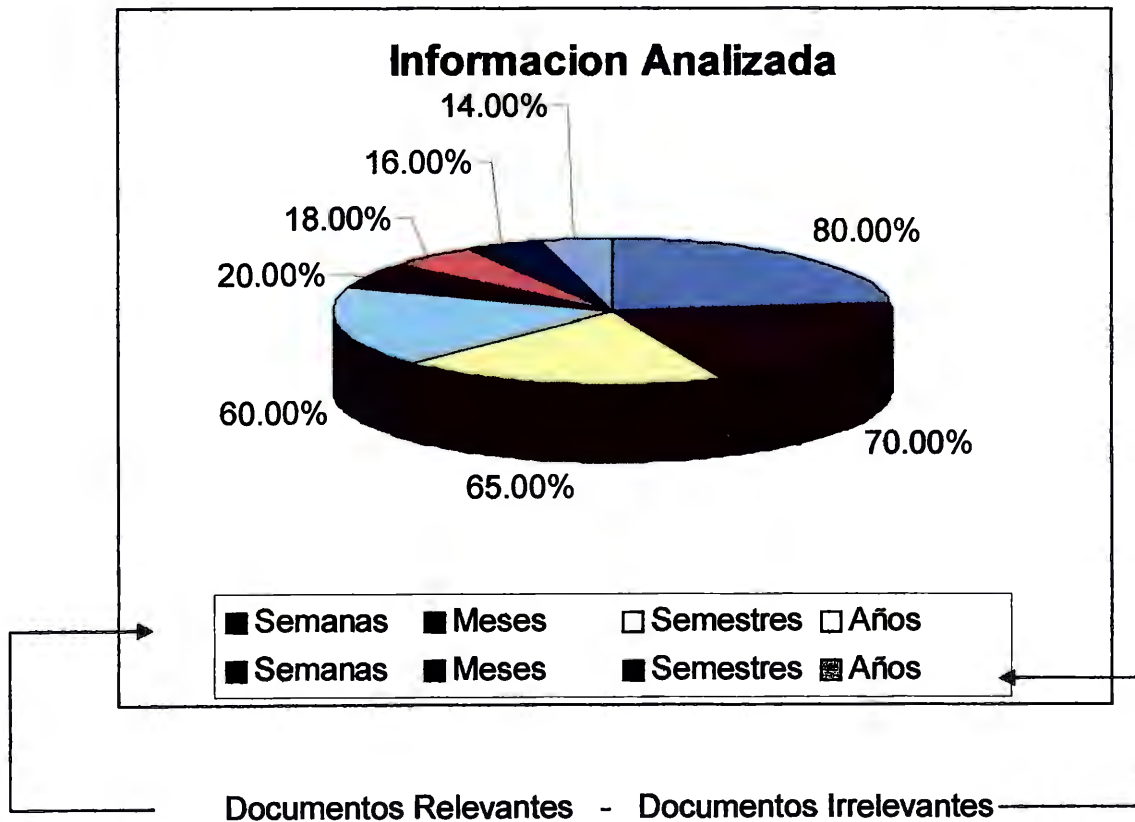
A diferencia de poseer datos estáticos se contaría con una herramienta seria capaz de administrar datos multidimensionalmente.

Las proyecciones de mercado se realizan manualmente debido a que se utilizan formulas estadísticas y matemáticas, por lo que es imposible (sin trabajar con herramientas que apliquen tecnologías como OLAP Server) el conocer algunos focos de información que pueden ser de importancia para la toma de decisión y que estadísticamente representan un amplio camino sin recorrer y explotar.

**2.4.1 Porcentaje de efectividad en los datos**

Una representación del análisis y estudio de los datos en términos de porcentaje se pueden observar de la siguiente manera.

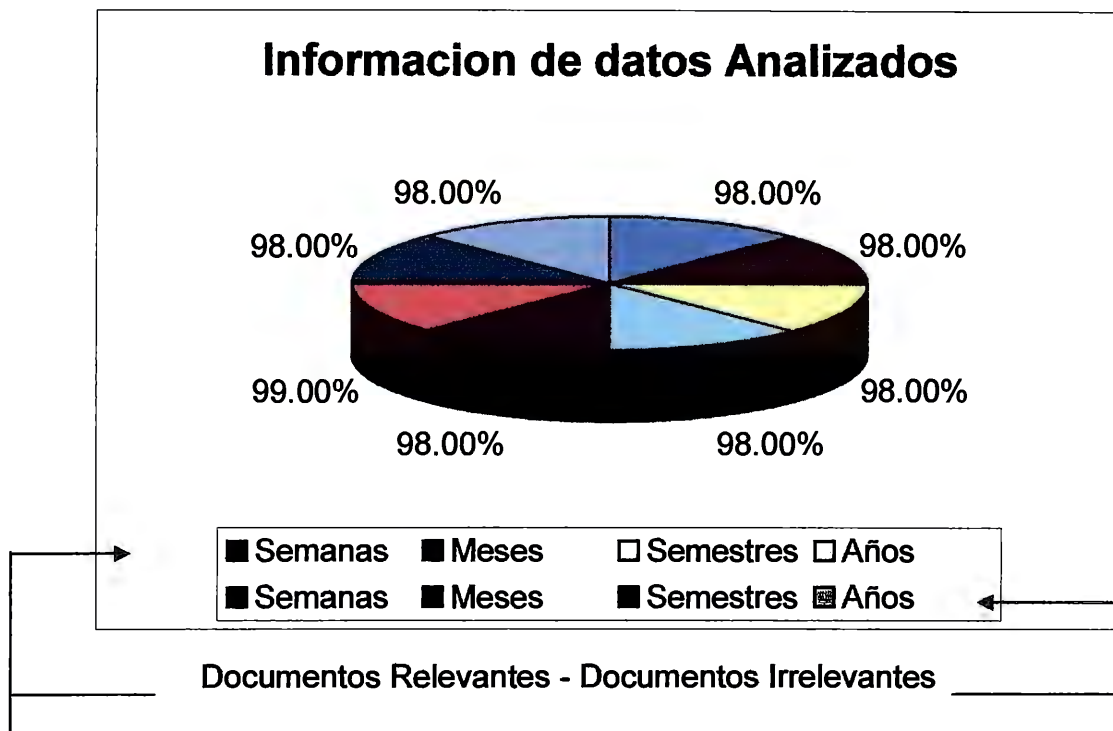
INFORMACION ANALIZADA PARA LA TOMA DE DECISION								
Tipo de Información	Documentación Relevante				Documentación Irrelevante			
	Semanas	Meses	Semestres	Años	Semanas	Meses	Semestres	Años
Clientes	80.00%	70.00%	65.00%	60.00%	20.00%	18.00%	16.00%	14.00%
Ventas	80.00%	70.00%	65.00%	60.00%	18.00%	16.00%	14.00%	12.00%
Compras	85.00%	65.00%	63.00%	60.00%	20.00%	18.00%	16.00%	14.00%
Mantenimiento y equipo	82.00%	70.00%	65.00%	60.00%	15.00%	13.00%	11.00%	9.00%



En el grafico podemos observar que el comportamiento de los documento involucrados en cada uno de los proceso de la empresa presentan un decremento en el porcentaje a medida avanza el tiempo, esto significa que la información no analizada es grande por parte de BIOLACT por lo que se desarrollara una herramienta que permita estudiar todos los atributos que contemple el marketing y disminuya en gran parte esta desventaja porcentual que ofrecen los procesos manuales con información estática.

Partiendo de estos conocimientos se pretende lograr lo siguiente:

INFORMACION ANALIZADA PARA LA TOMA DE DECISIÓN								
Categoría de Información	Documentación Relevante				Documentación Irrelevante			
	Semanas	Meses	Semestres	Años	Semanas	Meses	Semestres	Años
Clientes	98.00%	98.00%	98.00%	98.00%	98.00%	99.00%	98.00%	98.00%
Plantas	99.00%	99.00%	98.00%	98.00%	100.00%	99.00%	98.00%	98.00%
Empresas	99.00%	99.00%	98.00%	98.00%	99.00%	99.00%	98.00%	98.00%
Maquinaria y equipo	98.00%	98.00%	98.00%	98.00%	98.00%	98.00%	98.00%	98.00%



## **CAPITULO III. DESARROLLO Y DISEÑO.**

---

En este capítulo se presentan los requerimientos mínimos que se deben tener a la hora de ejecutar la herramienta “OLAP Viewer”, la plataforma en la que fue desarrollada, los esquemas de los datamarts utilizados, se explica como crear cubos de datos, modelos de minería de datos y consultas multidimensionales, a través del asistente del Análisis Manager de SQL Server 2000. También se explica el diseño de los formularios que se utilizan en la aplicación.

### **III. DESARROLLO Y DISEÑO.**

#### **3.1 Introducción al desarrollo y diseño**

El sistema a implementar es “Herramienta para el análisis de variables aplicadas al marketing utilizando Minería de datos y Servidores OLAP”, es un sistema de apoyo a la toma de decisiones con de la empresa BIOLACT, la cual fue tomada como referencia para el desarrollo del mismo.

Podemos definir un Sistema de Apoyo de decisiones como:

Sistemas de información computarizados que trabajan con la interacción que se da entre personas y computadoras. Para poder ligar la información, es indispensable compartir una base de datos en común como fuente.

Una de las características de estos sistemas, es que se involucra una base de datos para un objetivo específico de toma de decisiones.

Es una herramienta aplicada al Marketing, que es uno de los campos de mayor éxito de la minería de datos, se usa servidores OLAP, los cuales nos ayudan a trabajar con cubos, para mantenimiento de datos multidimensionales, apoyándonos en la tecnología Data Warehouse.

La tecnología Datawarehousing<sup>12</sup> debido a su orientación analítica, impone un procesamiento y pensamiento distinto, la cual se sustenta por un modelado de Bases de Datos propio, conocido como modelo multidimensional, el cual busca ofrecer al usuario su visión respecto de la operación del negocio.

---

<sup>12</sup> En algunas bibliografías esta es otra forma de llamar a la tecnología Data Warehouse

Modelo multidimensional, es una técnica para modelar bases de datos simples y entendibles al usuario final. La idea fundamental es que el usuario visualice fácilmente la relación que existe entre las distintas componentes del modelo.

### **3.1.1 Modelos de Datos**

Un factor clave presente durante todo el diseño de un DW<sup>13</sup>, fue expresado por Codd en 1983: “Ustedes pueden pensar que el significado de los datos es simple...pero no es así”. Para construir un DW se debe primero tener claro que existe una diferencia entre la estructura de la información y la semántica de la información, y que esta última es mucho más difícil de abarcar y que también es precisamente con ella con la que se trabaja en la construcción de un DW.

Aquí se encuentra la principal diferencia entre los sistemas operacionales y el DW: cada uno de ellos es sostenido por un modelo de datos diferente. Los sistemas operacionales se sustentan en el Modelo Entidad Relación (MER) y DDW trabaja con el Modelo multidimensional.

### **3.1.2 Características del MER**

Maneja la redundancia fuera de los datos. Por lo tanto realizar un cambio en la base, significa tocarla en un solo lugar.

Divide los datos en entidades, las que son representadas como tablas en una base de datos. Los MER crecen fácilmente, haciéndose más y más complejos.

Se puede apreciar la existencia de muchos caminos para ir de una tabla a otra. Sería natural pensar que al tener diversos caminos para llegar desde una tabla a otra,

---

<sup>13</sup> Hemos utilizado la abreviatura DW, para Data Warehouse

cualquiera entregaría el mismo resultado, pero lamentablemente esto no siempre sucede así.

El diagrama se visualiza simétrico, donde todas las tablas se parecen, sin distinguir a priori la importancia de unas respecto a otras. No es fácil de entender tanto para usuarios como para los diseñadores.

### 3.1.3 Características del Modelo Multidimensional

En general, la estructura básica de un DW para el Modelo Multidimensional está definida por dos elementos: esquemas y tablas.

**Tablas DW:** como cualquier base de datos relacional, un Data Warehouse se compone de tablas. Hay dos tipos básicos de tablas en el Modelo Multidimensional:

**Tablas Fact :** contienen los valores de las medidas de negocios, por ejemplo: ventas promedio en dólares, n número de unidades vendidas, etc.

**Tablas Lookup:** contienen el detalle de los valores que se encuentran asociados a la tabla Fact (tabla hechos).

**Esquemas DW:** la colección de tablas en el DW se conoce como Esquema. Los esquemas caen dentro de dos categorías básicas: esquemas estrellas y esquemas snowflake.

### **3.3 Requerimientos del Sistema.**

La herramienta desarrollada de la presente investigación, posee una interfaz gráfica de

### **3.2 Área de dominio de la Herramienta.**

Como ya hemos visto con anterioridad los campos donde se puede aplicar la minería de datos son muchos, para efectos de investigación y construcción de la herramienta, lo aplicaremos al área del marketing, la cual es una de mayor éxito y aceptación, donde se analizarán las variables que se aplican a dicho campo, por esta razón se aplicarán Servidores OLAP y minería de datos, ya que son técnicas efectivas para el manejo y análisis de datos multidimensionales.

Para el análisis de las variables, el cual ayudará a la toma de decisiones utilizaremos la arquitectura del Data Warehouse, que puede hacer mas eficaz un negocio, consolidando y organizando la información.

Un Data Warehouse usualmente contiene datos históricos, los cuales limpia para su eficiencia y consistencia, y los organiza de tal forma que puedan ser consultados, fácil y eficientemente.

El Objetivo final de cualquier proyecto de minería de datos se puede resumir en: ahorrar dinero mejorando la eficacia de las actividades o ganar dinero descubriendo nuevas fuentes de beneficios. En el campo elegido (Mercadeo) para aplicar la herramienta podemos llegar al objetivo, a partir de un conjunto de datos e implementando una o más técnicas. Cuanto más precisa sea la información que tengamos sobre los clientes, mayores posibilidades se tendrá de aumentar los ingresos y rentabilizar al máximo las acciones.

En el Marketing el objetivo fundamental puede resumirse en determinar quién comprará que, cuando y donde. Aplicando la minería de datos podemos detectar nuevas oportunidades de mercado, podemos conocer cómo evolucionara una variable en el futuro, a partir de datos históricos.

### **3.4 Estructuras de Base de Datos**

#### **3.4.1 Introducción a los Esquema Estrella y Esquema Snowflake**

##### **3.4.1.1 Esquema de Estrella**

En general, el modelo multidimensional también se conoce con el nombre de esquema estrella, pues su estructura base es similar: una tabla central y un conjunto de tablas que la atienden radialmente.

El esquema estrella deriva su nombre del hecho que su diagrama forma una estrella, con puntos radiales desde el centro. El centro de la estrella consiste de una o más tablas fact, y las puntas de la estrella son las tablas lookup.

Este modelo entonces, resulta ser asimétrico, pues hay una tabla dominante en el centro con varias conexiones a las otras tablas. Las tablas Lookup tienen sólo la conexión a la tabla fact y ninguna más.

##### **3.4.1.2 Esquema Snowflake (Copo de Nieve)**

La diferencia del esquema snowflake comparado con el esquema estrella, está en la estructura de las tablas lookup: las tablas lookup en el esquema snowflake están normalizadas. Cada tabla lookup contiene sólo el nivel que es clave primaria en la tabla y la foreign key de su parentesco del nivel más cercano del diagrama.

### 3.4.2 Tabla Fact o de Hechos

Es la tabla central en un esquema dimensional. Es en ella donde se almacenan las mediciones numéricas del negocio. Estas medidas se hacen sobre el grano o unidad básica de la tabla.

El grano o la granularidad de la tabla queda determinada por el nivel de detalle que se almacenará en la tabla. Por ejemplo, para el caso de producto, mercado y tiempo antes visto, el grano puede ser la cantidad de madera vendida ‘mensualmente’. El grano revierte las unidades atómicas en el esquema dimensional.

Cada medida es tomada de la intersección de las dimensiones que la definen. Idealmente está compuesta por valores numéricos, continuamente evaluados y aditivos. La razón de estas características es que así se facilita que los miles de registros que involucran una consulta sean comprimidos en unas pocas líneas en un conjunto de respuesta.

La clave de la tabla fact recibe el nombre de clave compuesta o concatenada debido a que se forma de la composición (o concatenación) de las llaves primarias de las tablas dimensionales a las que está unida.

Así entonces, se distinguen dos tipos de columnas en una tabla fact: columnas fact y columnas key.

Donde la columna fact es la que almacena alguna medida de negocio y una columna key forma parte de la clave compuesta de la tabla.

### 3.4.3 Tablas Lookup o Dimensionales

Estas tablas son las que se conectan a la tabla fact, son las que la alimentan. Una tabla lookup almacena un conjunto de valores que están relacionados a una dimensión particular. Tablas no contienen hechos, en su lugar los valores en las tablas lookup son los elementos que determinan la estructura de las dimensiones. Así entonces, en ellas existe el detalle de los valores de la dimensión respectiva.

Una tabla lookup está compuesta de una primary key que identifica unívocamente una fila en la tabla junto con un conjunto de atributos, y dependiendo del diseño del modelo multidimensional puede existir una foreign key que determina su relación con otra tabla lookup.

Para decidir si un campo de datos es un atributo o un hecho se analiza la variación de la medida a través del tiempo. Si varía continuamente implicaría tomarlo como un hecho, caso contrario será un atributo.

Los atributos dimensionales son un rol determinante en un DDW. Ellos son la fuente de todas las necesidades que debieran cubrirse. Esto significa que la base de datos será tan buena como lo sean los atributos dimensionales, mientras más descriptivos, manejables y de buena calidad, mejor será el DDW.

Los diagramas utilizados en nuestro estudio son una combinación de esquemas estrella y copo de nieve (como lo muestra la Figura 3.1). Los Datamarts de Producción y Ventas fueron creados de tal forma que se adecuarán a las necesidades de la empresa en los departamentos de producción y ventas\_marketing.

La Figura 3.1 muestra un esquema combinado (estrella - copo de nieve) para el Datamart Ventas.

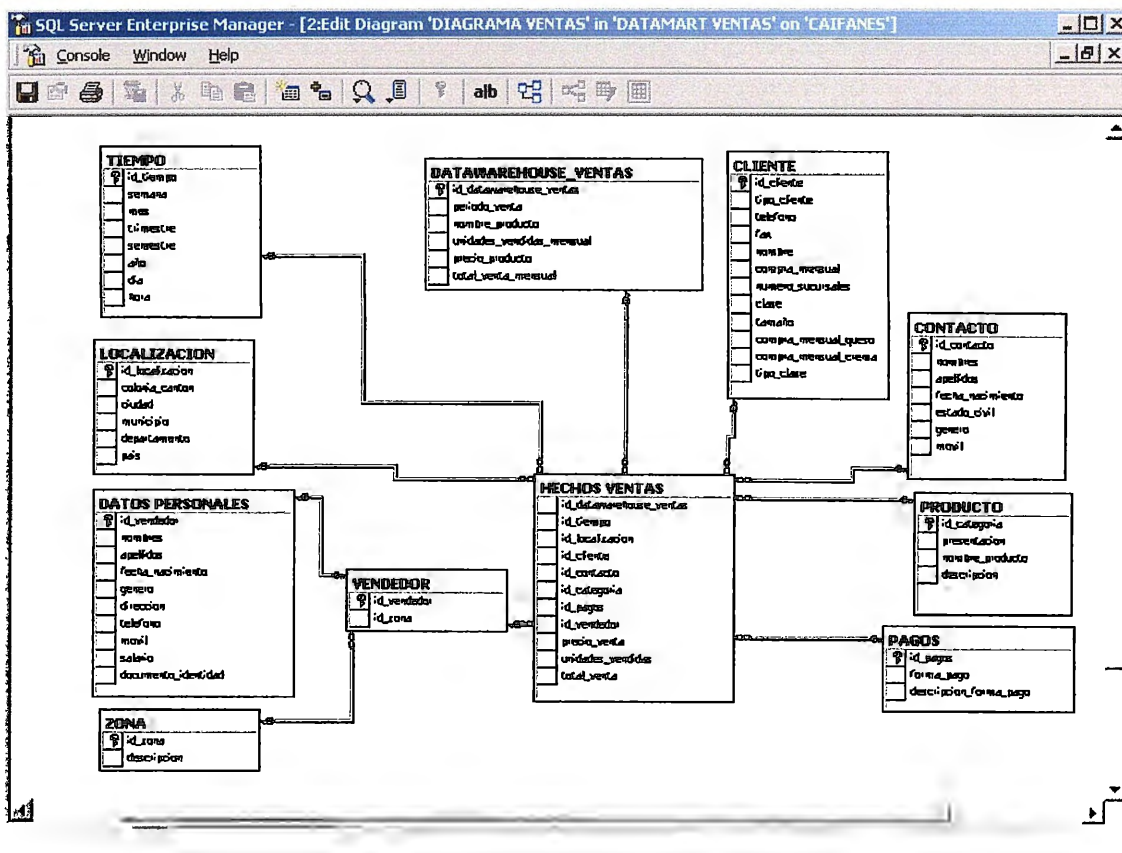


Figura 3.1 Esquema Datamart Ventas.

La mayor parte de su estructura muestra una tendencia al esquema estrella (debido al detalle de la información), mientras que el resto presenta esquema copo de nieve. Al igual que el Datmart Ventas un segundo Datamart es creado para el análisis del departamento de producción denominado Datamart Producción y que tan bien reúne las características del de Ventas (Figura 3.2).

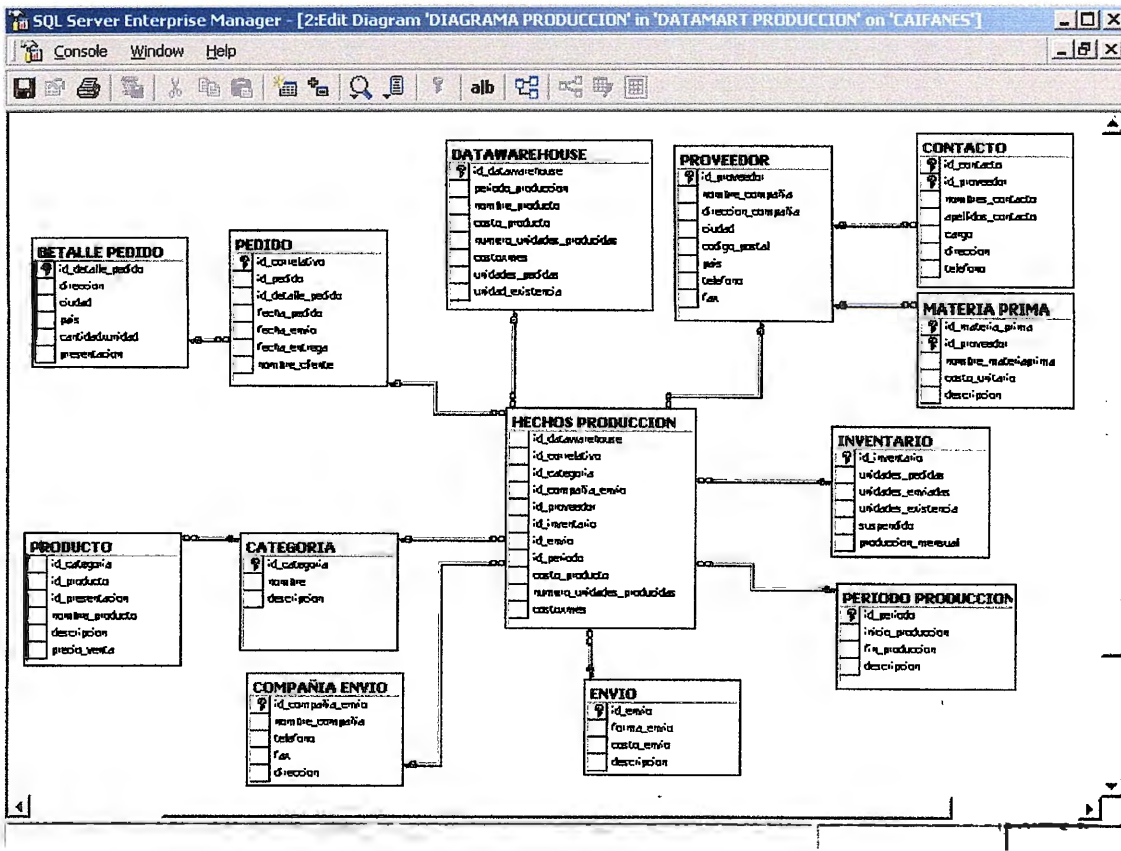


Figura 3.2 Esquema Datamart Producción

La tabla de hechos la constituyen en su mayoría campos numéricos que se utilizan para relacionar datos entre tablas y para conocer variables de importancia (Figura 3.3).

HECHOS VENTAS	
id_datawarehouse_ventas	
id_tiempo	
id_localizacion	
id_cliente	
id_contacto	
id_categoria	
id_pagos	
id_vendedor	
precio_venta	
unidades_vendidas	
total_venta	

Figura 3.3 Tabla Hechos Ventas

### 3.5 Cubos

Los cubos utilizados en el proceso analítico en línea (OLAP) proporcionan una vista multidimensional de los datos del almacén que los usuarios finales pueden utilizar y explorar fácilmente cuando buscan respuestas a cuestiones comerciales.

Análisis Service organiza los datos de un almacén de datos en cubos multidimensionales con una información de resumen calculada previamente para proporcionar respuestas rápidas a las consultas analíticas complejas.

Los cubos son los principales objetos del proceso analítico en línea (OLAP, *Online Analytic Processing*), una tecnología que proporciona rápido acceso a los datos de un almacén de datos. Un cubo es un conjunto de datos que normalmente se construye a partir de un subconjunto de un almacén de datos y se organiza y resume en una estructura multidimensional definida por un conjunto de dimensiones y medidas.

Un cubo proporciona un mecanismo fácil de usar para consultar datos con tiempos de respuesta rápidos y uniformes. Los usuarios finales usan aplicaciones cliente para conectarse con Análisis Server y consultar los cubos del servidor. En la mayoría de las aplicaciones cliente, los usuarios finales ejecutan una consulta en un cubo mediante la manipulación de los controles de la interfaz de usuario, que determinan el contenido de la consulta. Esto evita la necesidad de escribir consultas basadas en lenguaje. Los datos de resúmenes precalculados denominados agregaciones proporcionan el mecanismo para obtener tiempos de respuesta rápidos y uniformes a las consultas.

Las agregaciones se crean para un cubo antes de que los usuarios finales puedan obtener acceso al mismo. Los resultados de una consulta se recuperan de las agregaciones, de los datos de origen del cubo del almacén de datos, de una copia de estos datos en el Análisis Service, de la caché del cliente o de una combinación de estos orígenes. Un Análisis Server puede admitir muchos cubos distintos, por ejemplo, un cubo para ventas, uno para producción, otro para clientes, etc.

Cada cubo tiene un esquema, que es el conjunto de tablas combinadas del almacén de datos del que el cubo obtiene sus datos de origen. La tabla central del esquema es la tabla de hechos, el origen de las medidas del cubo. Las demás tablas son las tablas de dimensiones, los orígenes de las dimensiones del cubo. Para obtener más información acerca de los esquemas. Un cubo se define por las medidas y dimensiones que contiene.

Cada dimensión de un cubo puede contener una jerarquía de niveles para especificar la división en categorías disponible para los usuarios finales. Los niveles de dimensión son una herramienta eficaz de modelado de datos porque permiten que los usuarios finales hagan preguntas en un nivel alto y posteriormente, expandan una jerarquía de dimensión para revelar más detalles.

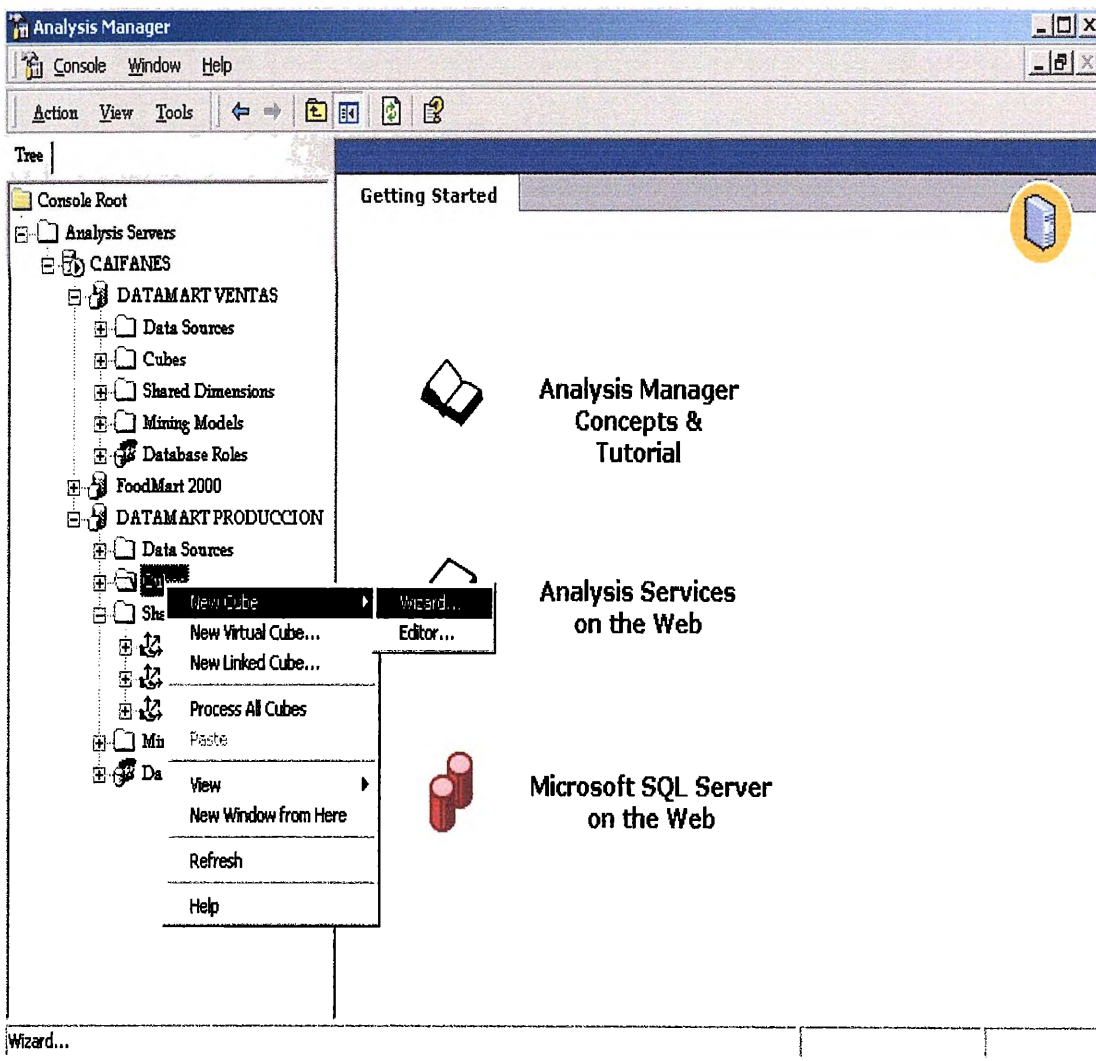
De tal forma, para el desarrollo de nuestra investigación se realizaron varios cubos que permitirán un mejor análisis de lo descrito anteriormente.

### **3.5.1 Cubos Producción**

Un primer cubo es aplicado a la base de datos DATAMART PRODUCCIÓN con el objetivo de conocer los costos totales de los productos enviados en un determinado momento. Por ejemplo el total de los costos de producción del producto Queso Fresco en el mes de Febrero del 2003.

#### **Desarrollo por medio del Análisis Manager**

Cuando un cubo es procesado, se calculan los agregados diseñados para él y se carga con los agregados y datos calculados. Procesarlo implica leer las tablas de dimensiones para llenar los niveles con miembros de los datos actuales, leer la tabla de hechos, calcular los agregados específicos y almacenar los resultados en el cubo. Una vez procesado se puede consultar.



**Figura 3.4 Analysis Manager**

Una vez creada la conexión hacia la base de datos se procede a la creación del cubo haciendo clic derecho en la carpeta cube, new cube, wizard tal como se muestra en la Figura 3.4.

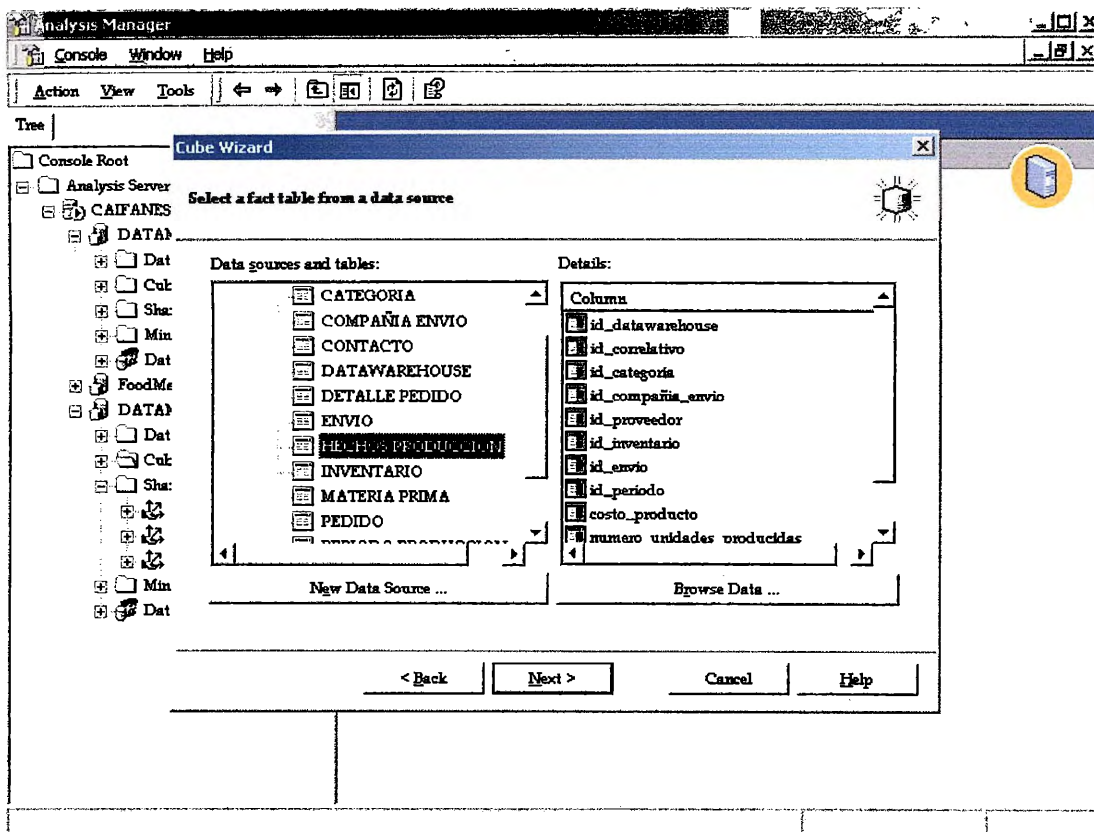


Figura 3.5 Editor de cubos

La figura 3.5 muestra el editor de cubos, en esta parte se elige la tabla de hechos, donde se encuentran los datos que se convertirán en medidas para el análisis del cubo.

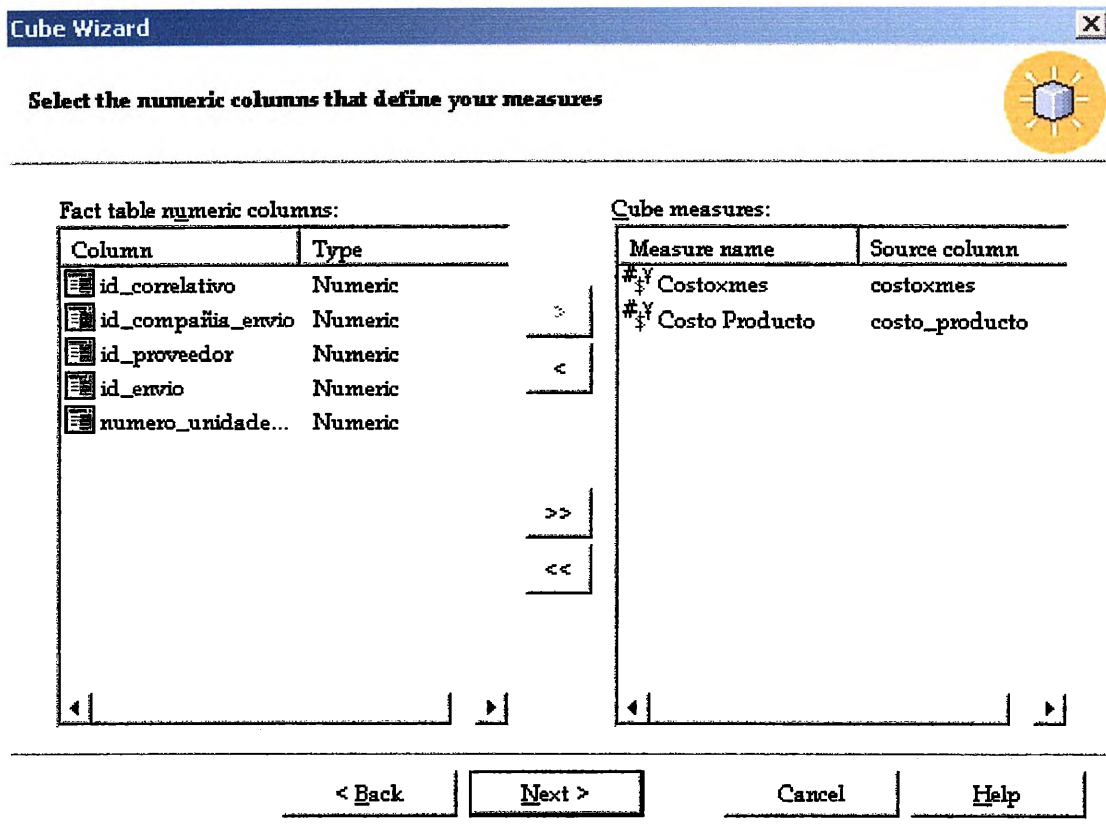
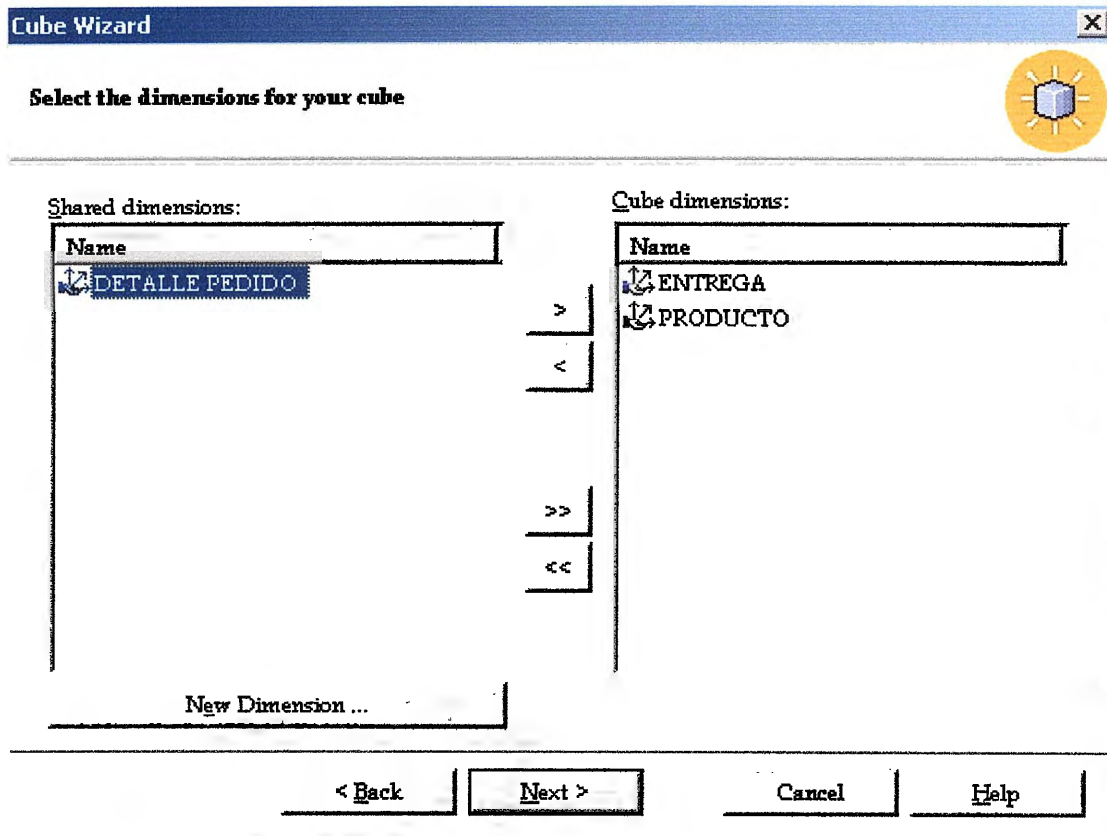


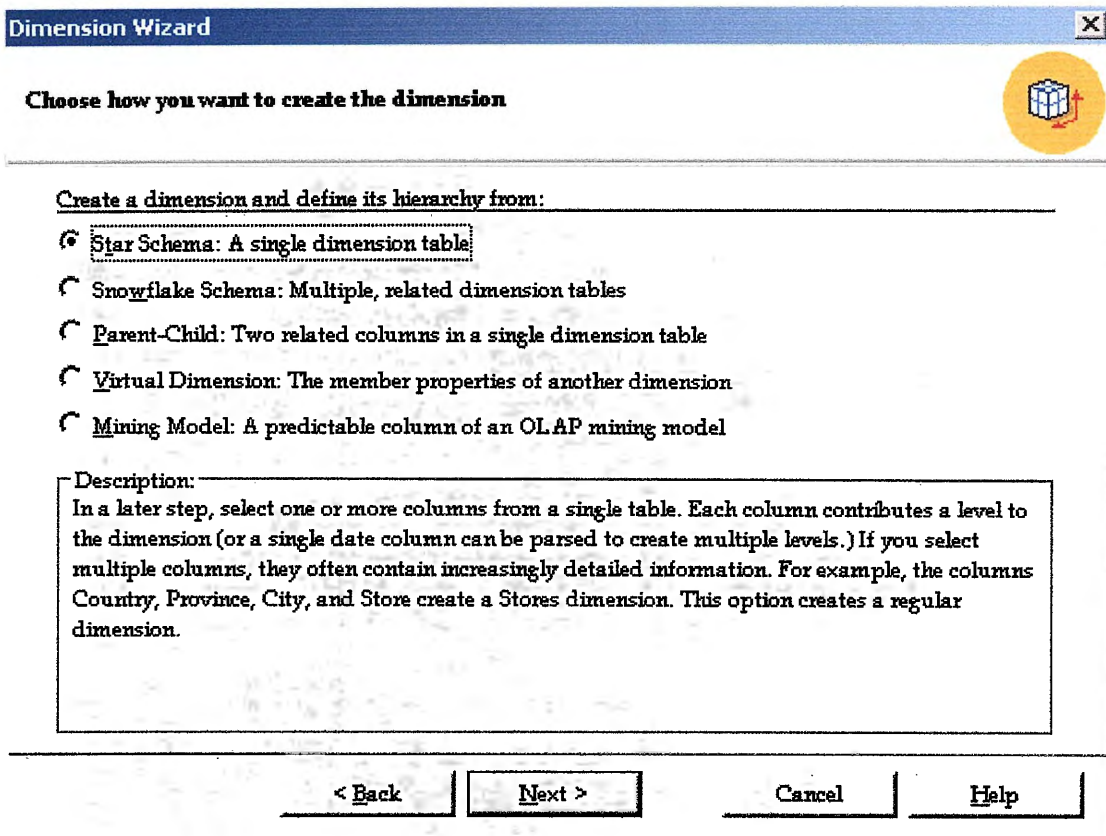
Figura 3.6 Elección de campos

La figura 3.6 muestra el proceso de elección de campos numéricos de importancia de la tabla Hechos Producción, mismos que se convierten en variables denominadas medidas. En un cubo, una medida es un conjunto de valores de una columna de la tabla de hechos, los cuales suelen ser numéricos. Además, las medidas son los valores centrales del cubo que se analiza. Es decir, son datos numéricos de gran interés para los usuarios finales.



**Figura 3.7 Selección de dimensiones**

Una vez elegidas las medidas se procede a definir las dimensiones (Figura 3.7), las cuales son una jerarquía organizada de categorías, conocidas como niveles, que describe los datos de las tablas de hechos de los almacenes de datos. Normalmente, las dimensiones describen un conjunto similar de miembros en que el usuario desea basar un análisis y son componentes fundamentales de los cubos.



**Figura 3.8 Crear dimensiones**

Cuando las dimensiones son creadas, es necesario tener prediseñado el modelo para poder definir su jerarquía. La Figura 3.8 muestra los diferentes métodos que existen para hacer de uno o varios campos una dimensión. Para el ejemplo escogeremos la opción Copo de Nieve, esto debido a que el diseño de la base de datos Datamart Producción se constituye como un esquema de ese tipo, por otra parte no queda excepto de constituirse también como esquema estrella, esto dependiendo de las variables a analizar.

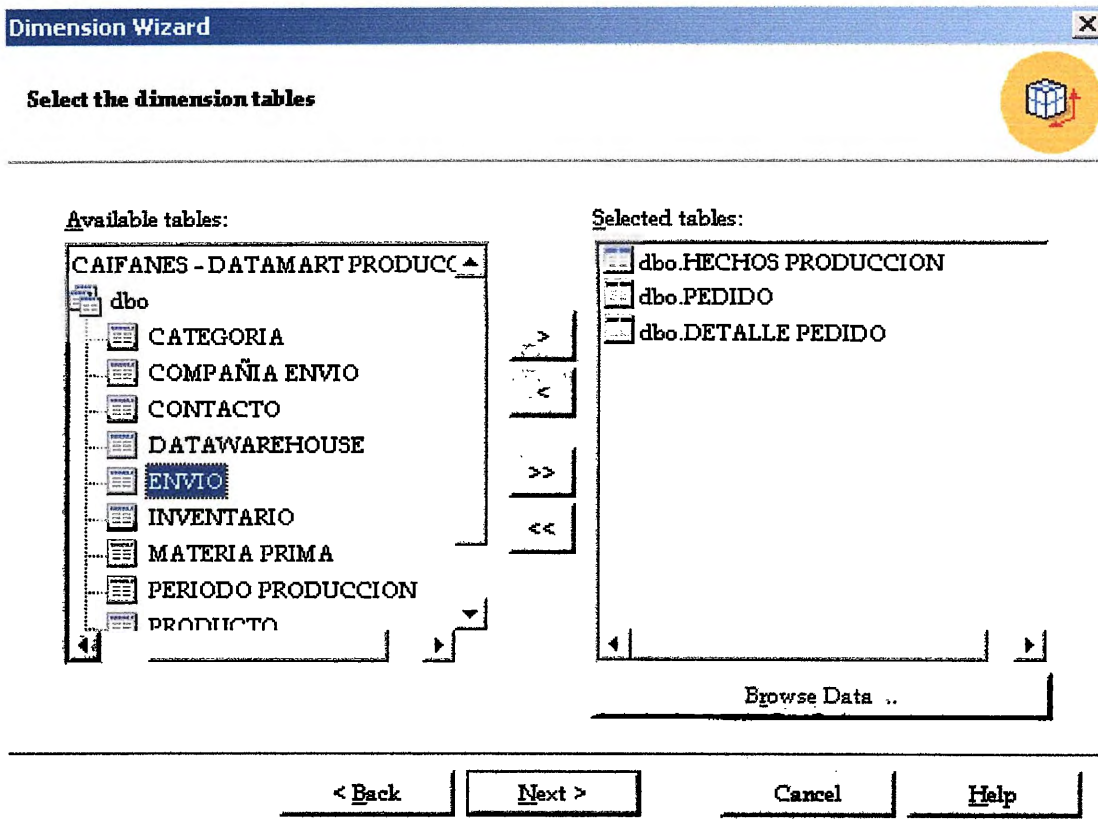
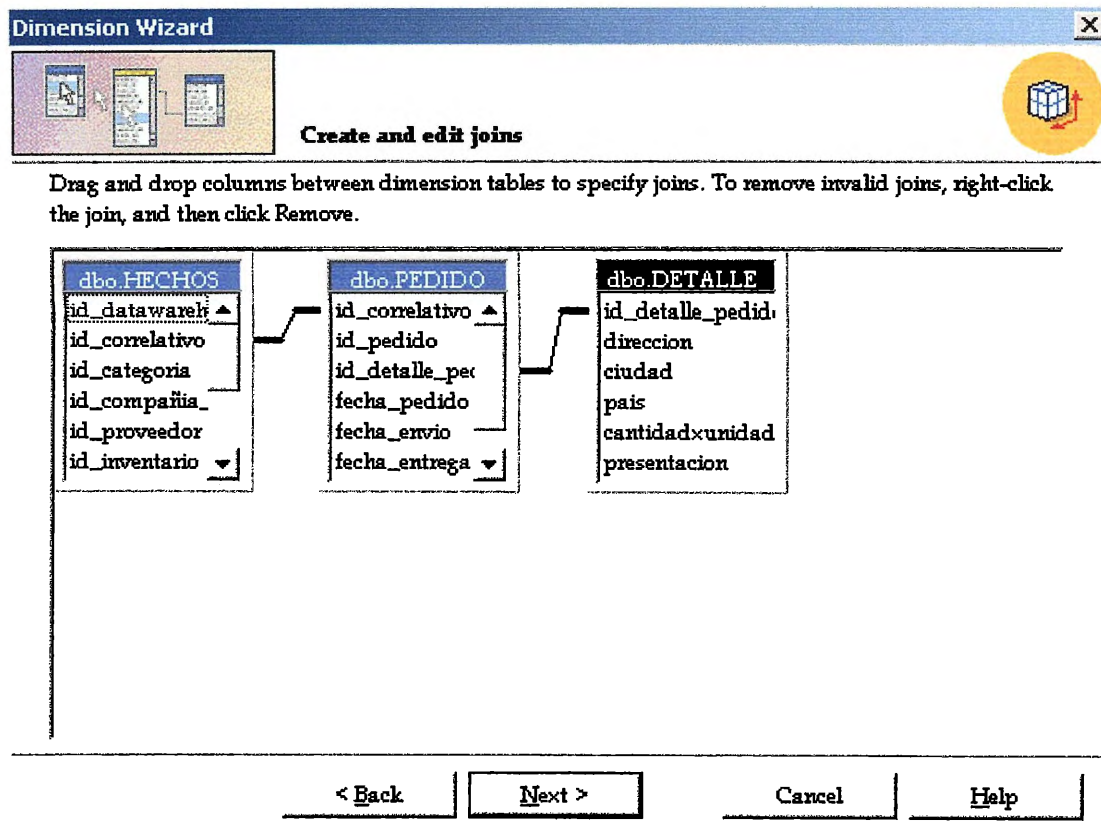


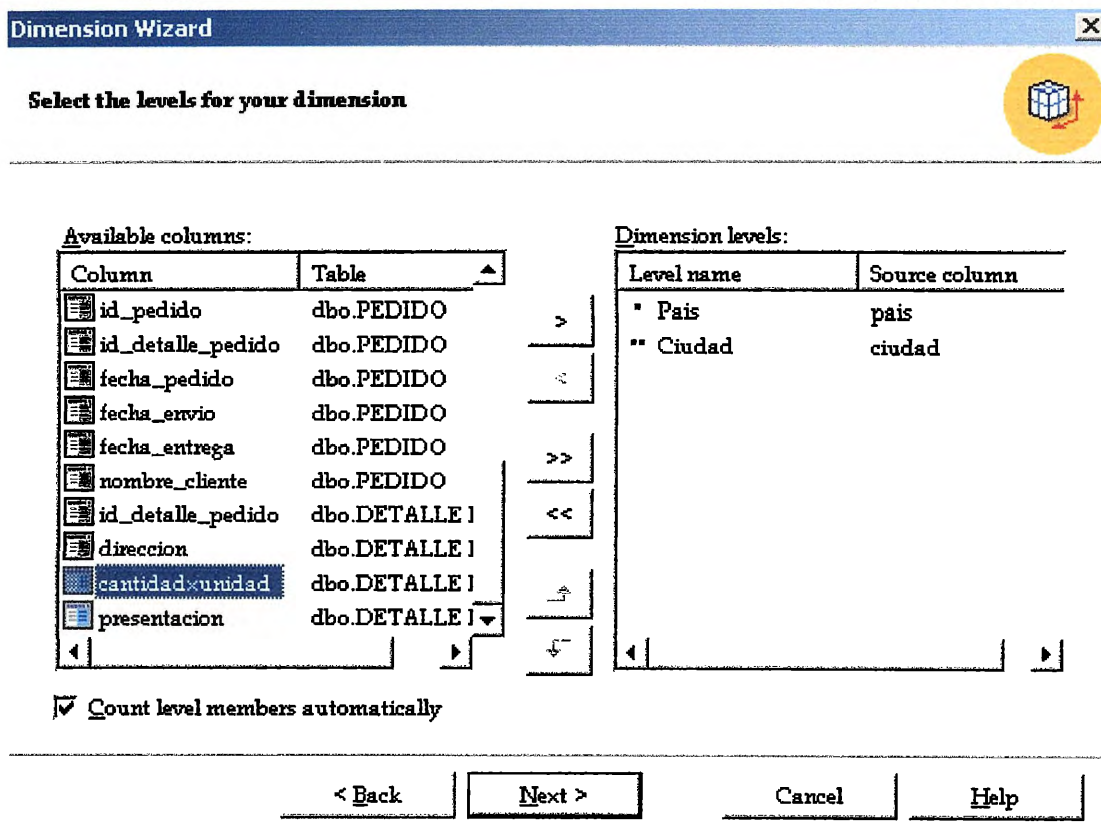
Figura 3.9 Tablas seleccionadas

Se han elegido las tablas Hechos Ventas, Pedido y Detalle de Pedido para la primera dimensión. Más adelante se crearán otras 2 dimensiones con lo que se fortalece mas el cubo y su objeto de análisis (Ver Figura 3.9).



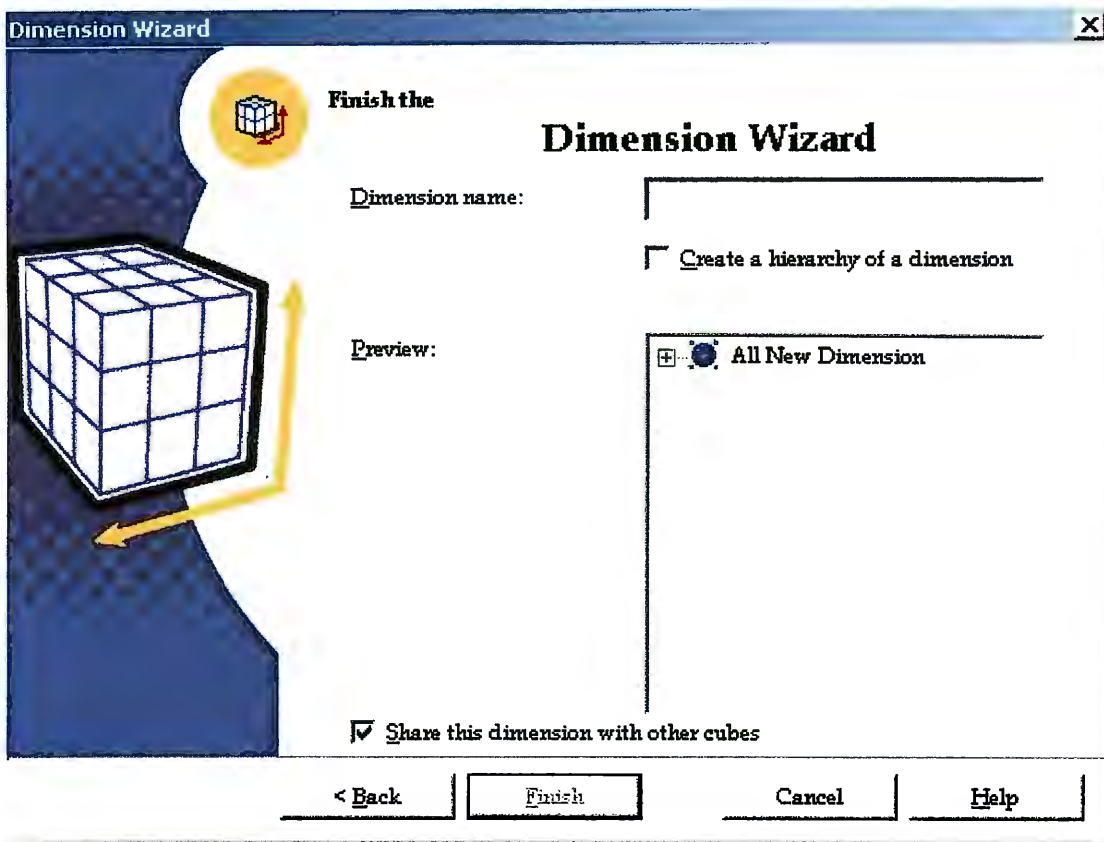
**Figura 3.10 Uniones de tablas seleccionadas**

De acuerdo con las especificaciones se puede perfectamente editar o borrar las uniones que se crean convenientes al análisis (Ver Figura 3.10).



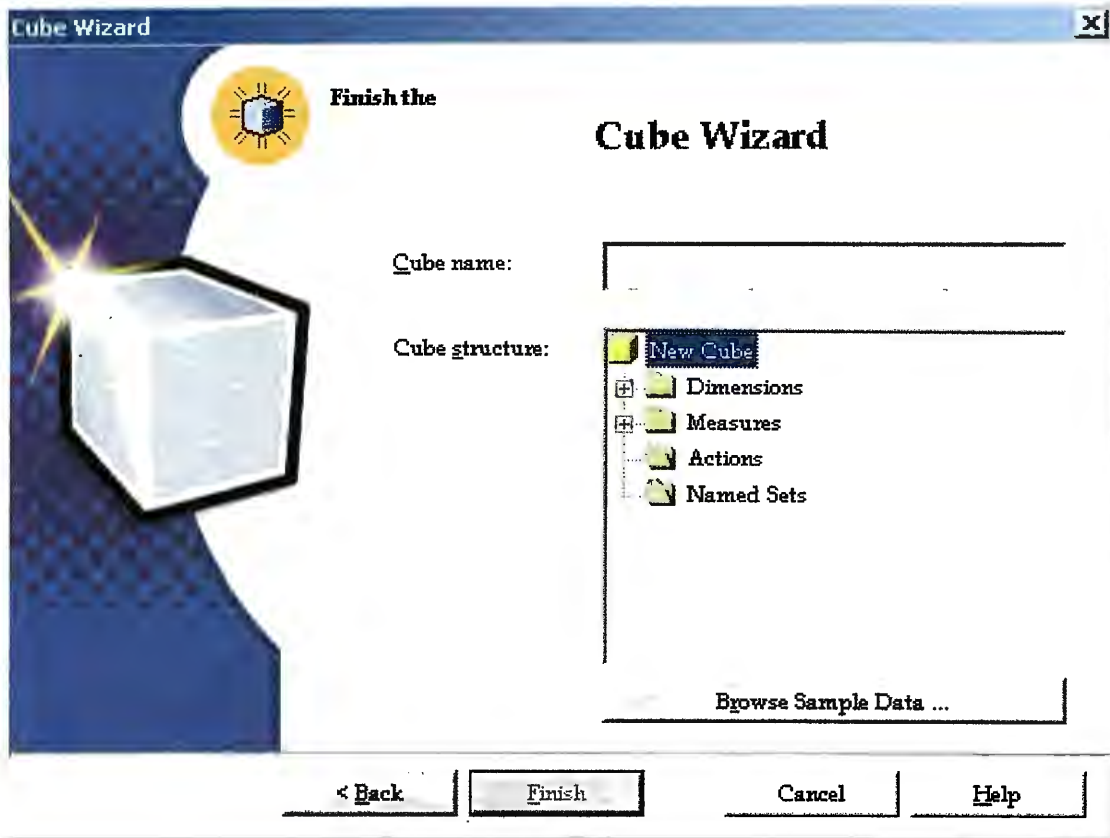
**Figura 3.11 Selección de niveles**

En la Figura 3.11 se muestra el cuadro de dialogo que permite definir niveles en una dimensión, en esta etapa se seleccionan los siguientes campos: País, Ciudad, Dirección, Cantidadxunidad.



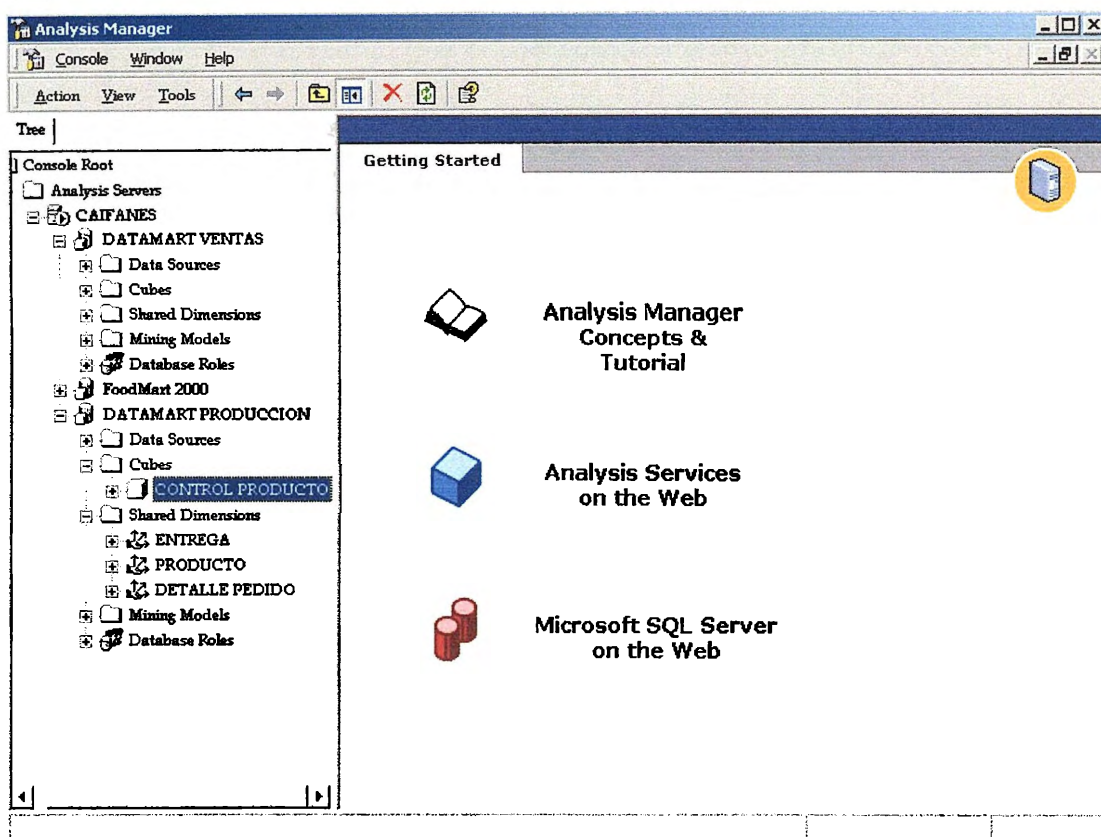
**Figura 3.12 Finalización del proceso de creación de dimensión**

La figura 3.12 muestra la finalización del proceso de creación de dimensión, como último se le da un nombre. De igual manera se han creado las dimensiones ENTREGA y PRODUCTO con lo que se tiene una mejor perspectiva de los datos con respecto a los ejes tiempo y costo.



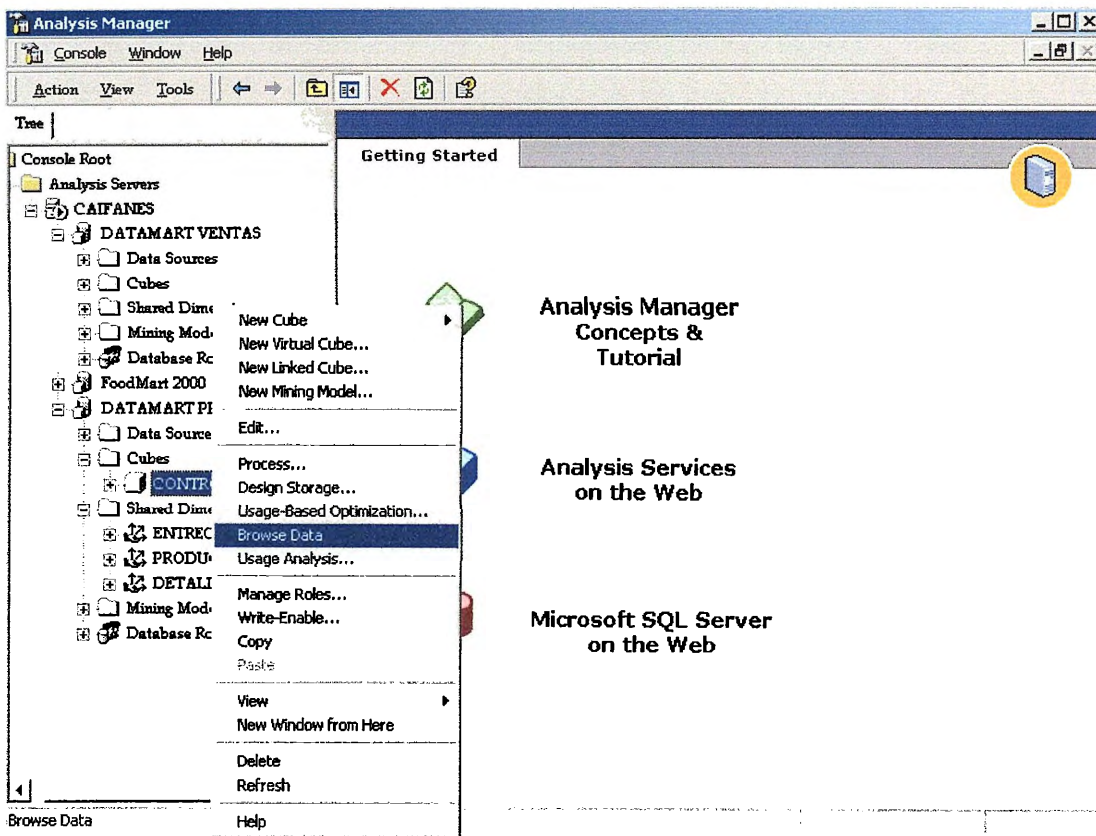
**Figura 3.13 Final del proceso de creación del cubo**

Una vez definidas las dimensiones se finaliza el proceso de creación de cubos asignándole un nombre (Figura 3.13).



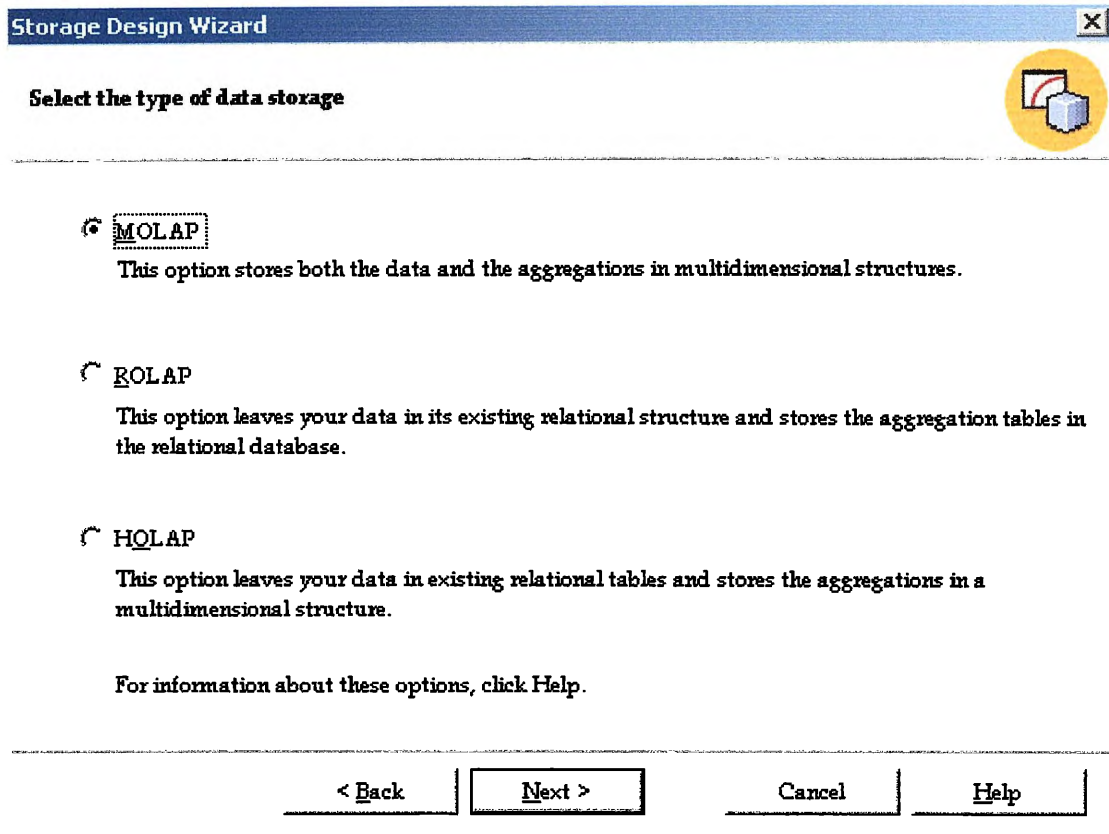
**Figura 3.14 Analysis Manager –Cubo creado-**

La figura 3.14 muestra como en el panel de árbol del Análisis Manager se observa el cubo creado, de igual manera se puede extender la carpeta de dimensiones compartidas, de acuerdo con las especificaciones del cubo teniendo tres dimensiones que serán de gran utilidad para la identificación de patrones en el almacén de datos.



**Figura 3.15 Analysis Manager**

Para observar los datos y analizar la información por medio de consultas multidimensionales se da clic derecho en el cubo creado y se selecciona el ítem Buscar Datos, como lo muestra la Figura 3.15.

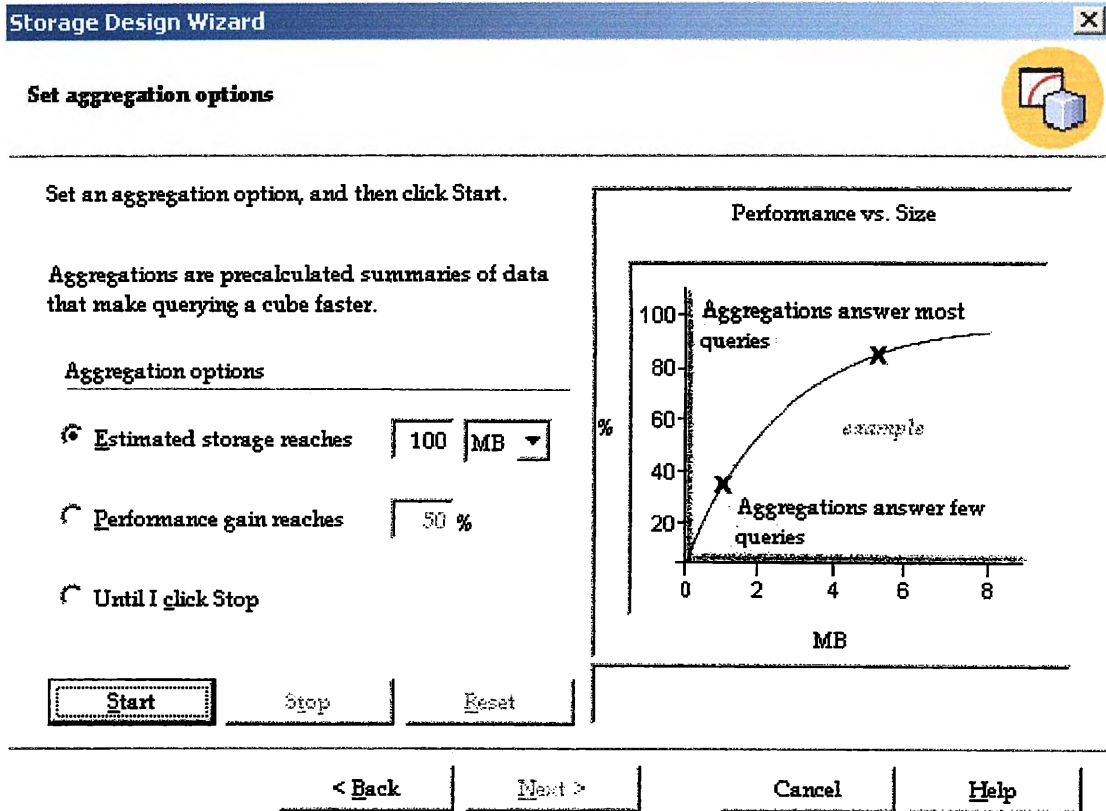


**Figura 3.16 Tipo de almacenamiento**

Para crear un cubo es necesario aplicarle un modo de almacenamiento, una dimensión puede tener uno de los siguientes: OLAP multidimensional (MOLAP) u OLAP relacional (ROLAP). MOLAP es el predeterminado de una dimensión.

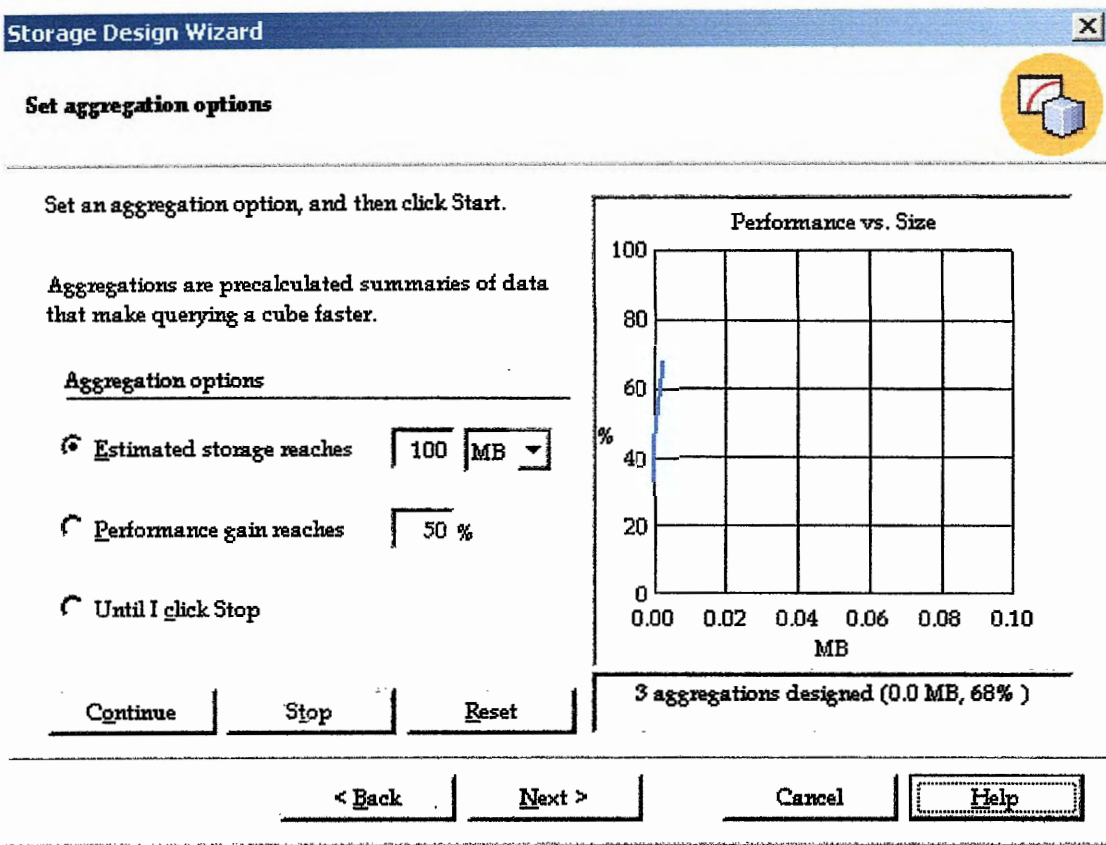
El modo de almacenamiento determina la ubicación y la forma de los datos de una dimensión. Los datos de una dimensión MOLAP se almacenan en una estructura multidimensional en el servidor OLAP., la cual se crea al procesarla.

Las dimensiones MOLAP proporcionan un mejor rendimiento de las consultas que las ROLAP. Sin embargo, las que son gigantes, que pueden tener 10 millones de miembros o más, no admiten el modo de almacenamiento MOLAP. En el cubo creado se elige MOLAP como lo muestra la Figura 3.16.



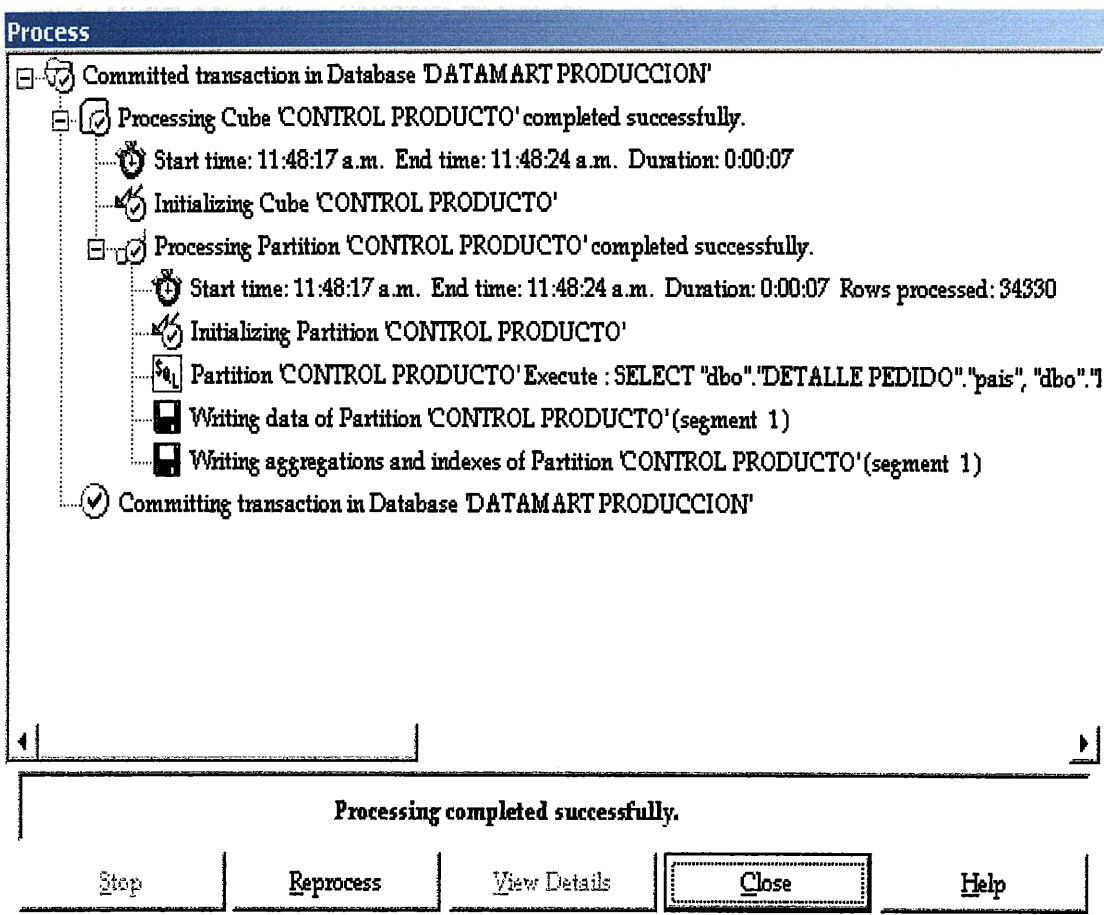
**Figura 3.17 Opciones de agregaciones**

La figura 3.17 muestra las opciones de las agregaciones que se crean en el cubo, incrementando el performance de acuerdo al tamaño de datos en disco.



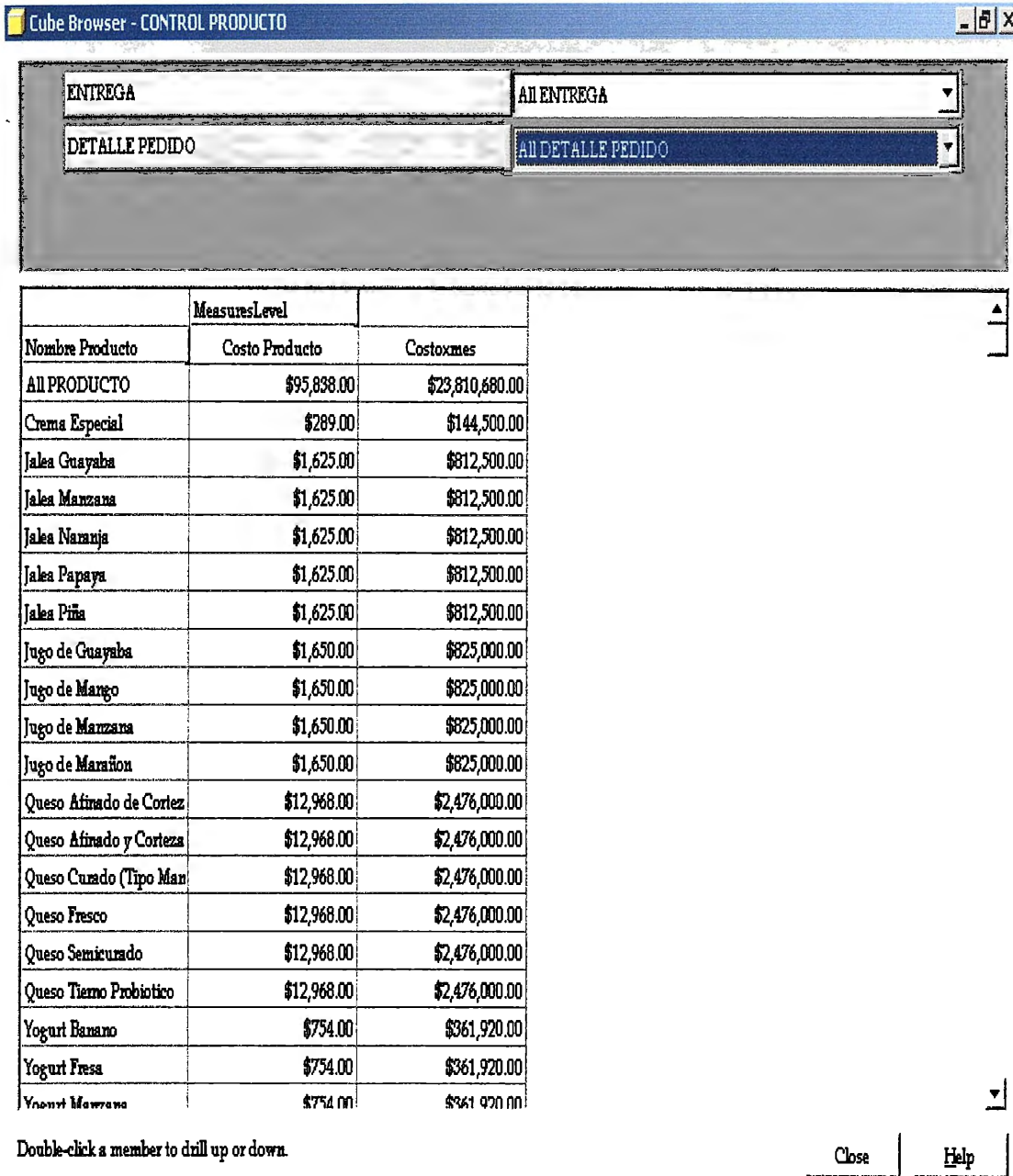
**Figura 3.18 Opciones de agregaciones**

Cuando el proceso de agregación concluye se da clic en siguiente, tal como se muestra en la Figura 3.18.



**Figura 3.19 Procesamiento del cubo**

Finalmente se observa como se procesa el cubo, incluyendo medidas, dimensiones y agregaciones. Clic en cerrar para terminar el procedimiento. Ver figura 3.19.



**Figura 3.20 Resultado final del cubo creado**

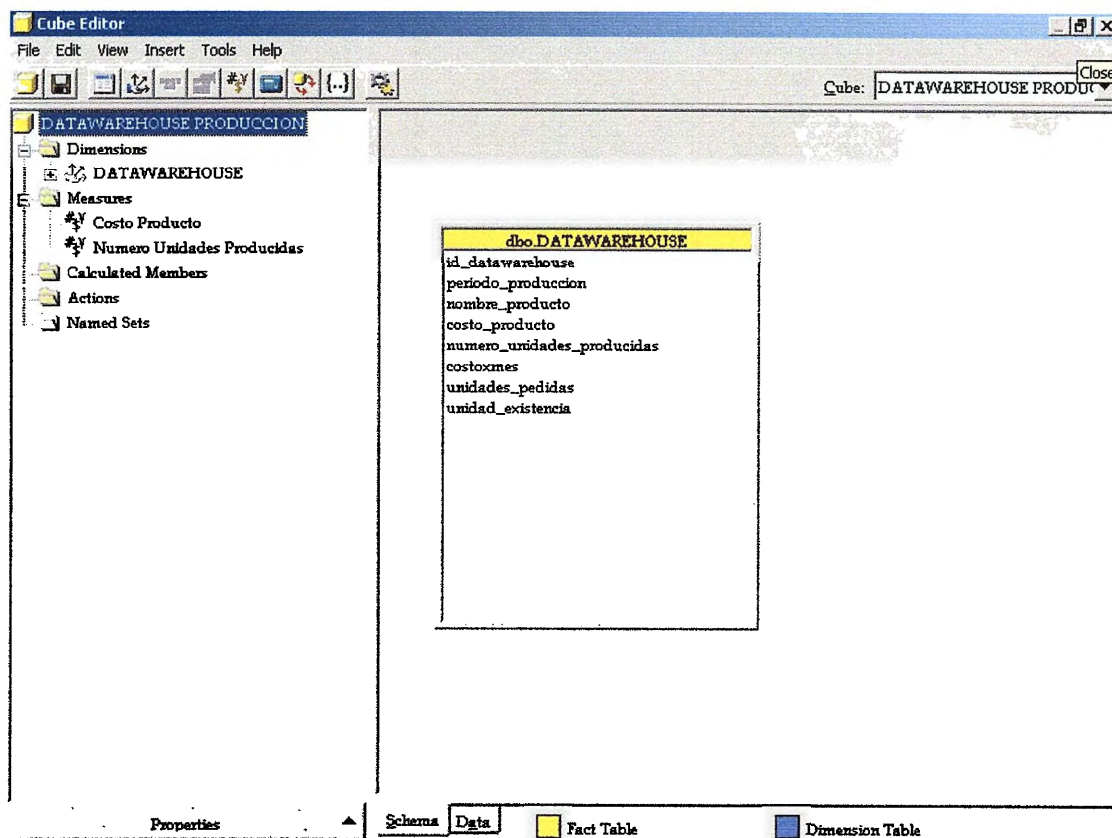
La Figura 3.20 muestra el resultado final del cubo que se realizo, en esta parte es donde se modelan los datos de las dimensiones y tablas de hechos del almacén, proporcionando así sofisticadas consultas y análisis a las aplicaciones cliente.

## Datawarehouse

Un segundo cubo aplicado a la base de datos Datamart Producción es el de Datawarehouse, en donde se analiza los costos en los años 2002 y 2003 para todos los productos.

## Desarrollo

El mismo procedimiento anterior es válido para la realización de este cubo, de manera que las especificaciones que se le asignen son las siguientes.



**Figura 3.21 Dimensiones y medidas del cubo Datawarehouse**

La Figura 3.21 detalla las dimensiones y medidas del cubo Datawarehouse. Obsérvese que se analiza una sola tabla, también se puede crear dimensiones en una sola tabla y niveles propios sin ningún inconveniente. Por consiguiente se escoge un modelo de almacenamiento MOLAP y que aplique a un esquema de estrella.

El resultado final del cubo DATAWAREHOUSE PRODUCCION puede apreciarse en la figura 3.22.

Cube Browser - DATAWAREHOUSE PRODUCCION		
	MeasuresLevel	
+ Periodo Produccion	Costo Producto	mero Unidades Producid
All DATAWAREHOUSE	\$1,059.28	159,436.00
+ Abril 2002	\$46.94	6,853.00
+ Abril 2003	\$49.60	7,720.00
+ Agosto 2002	\$46.94	6,853.00
+ Agosto 2003	\$49.60	7,720.00
+ Diciembre 2001	\$46.94	6,853.00
+ Diciembre 2002	\$49.60	7,720.00
+ Enero 2002	\$46.94	6,853.00
+ Enero 2003	\$49.60	7,720.00
+ Febrero 2002	\$46.94	6,853.00
+ Febrero 2003	\$49.60	7,720.00
+ Julio 2002	\$46.94	6,853.00
+ Julio 2003	\$49.60	7,720.00
+ Junio 2002	\$46.94	6,853.00
+ Junio 2003	\$49.60	7,720.00
+ Marzo 2002	\$46.94	6,853.00
+ Marzo 2003	\$49.60	7,720.00
+ Mayo 2002	\$46.94	6,853.00
+ Mayo 2003	\$49.60	7,720.00
+ Noviembre 2002	\$46.94	6,853.00
+ Octubre 2002	\$46.94	6,853.00
+ Septiembre 2002	\$46.94	6,853.00
+ Septiembre 2003	\$49.60	7,720.00

Double-click a member to drill up or down.

Close | Help

**Figura 3.22 Resultado final del cubo DATAWAREHOUSE PRODUCCION**

### 3.5.2 Cubos Datamart Ventas

#### Ventas1

El propósito de este cubo es observar el comportamiento de las variables cliente, tiempo y producto del año 2003. La estructura del cubo se observa en la figura 3.23.

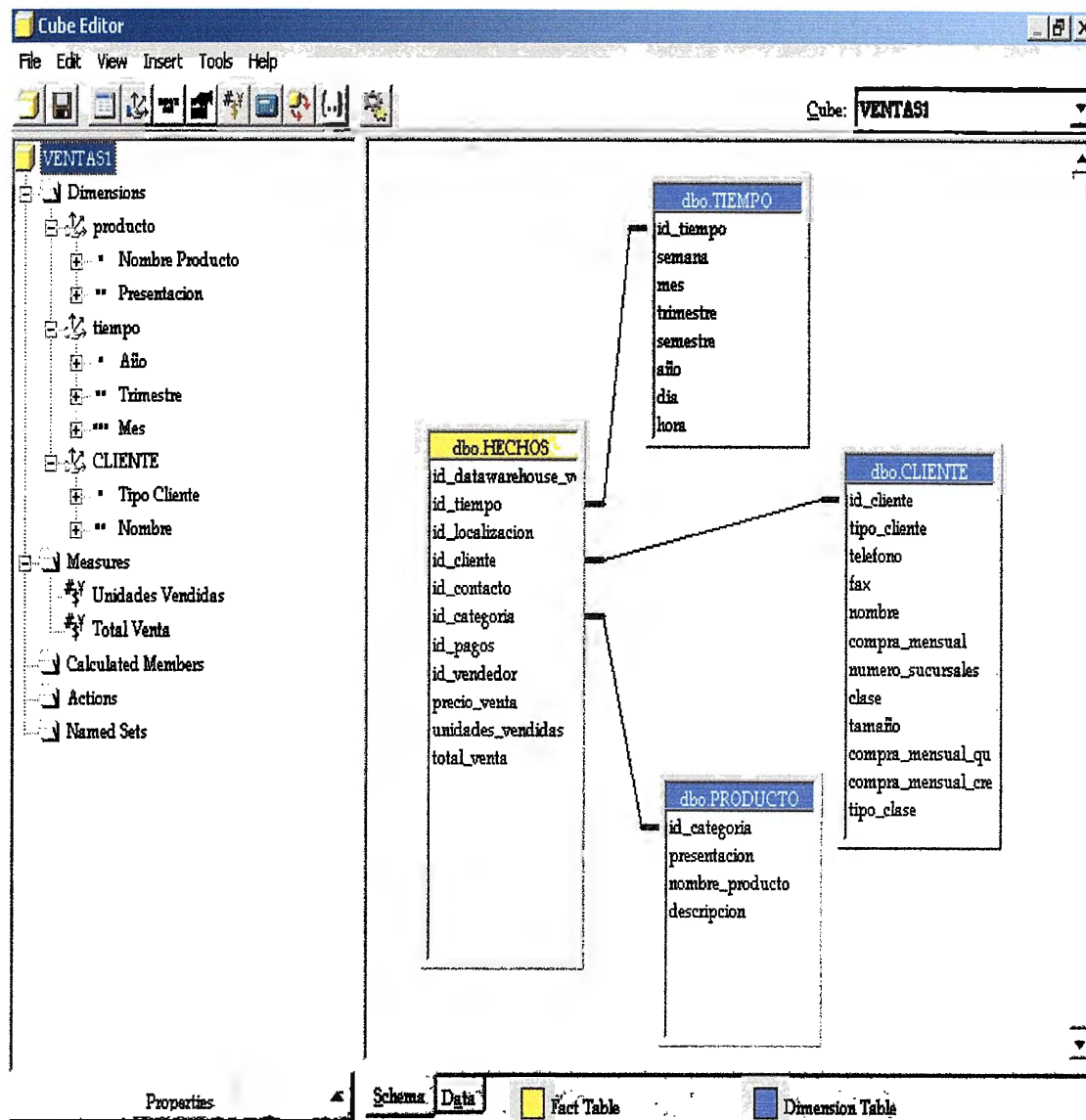
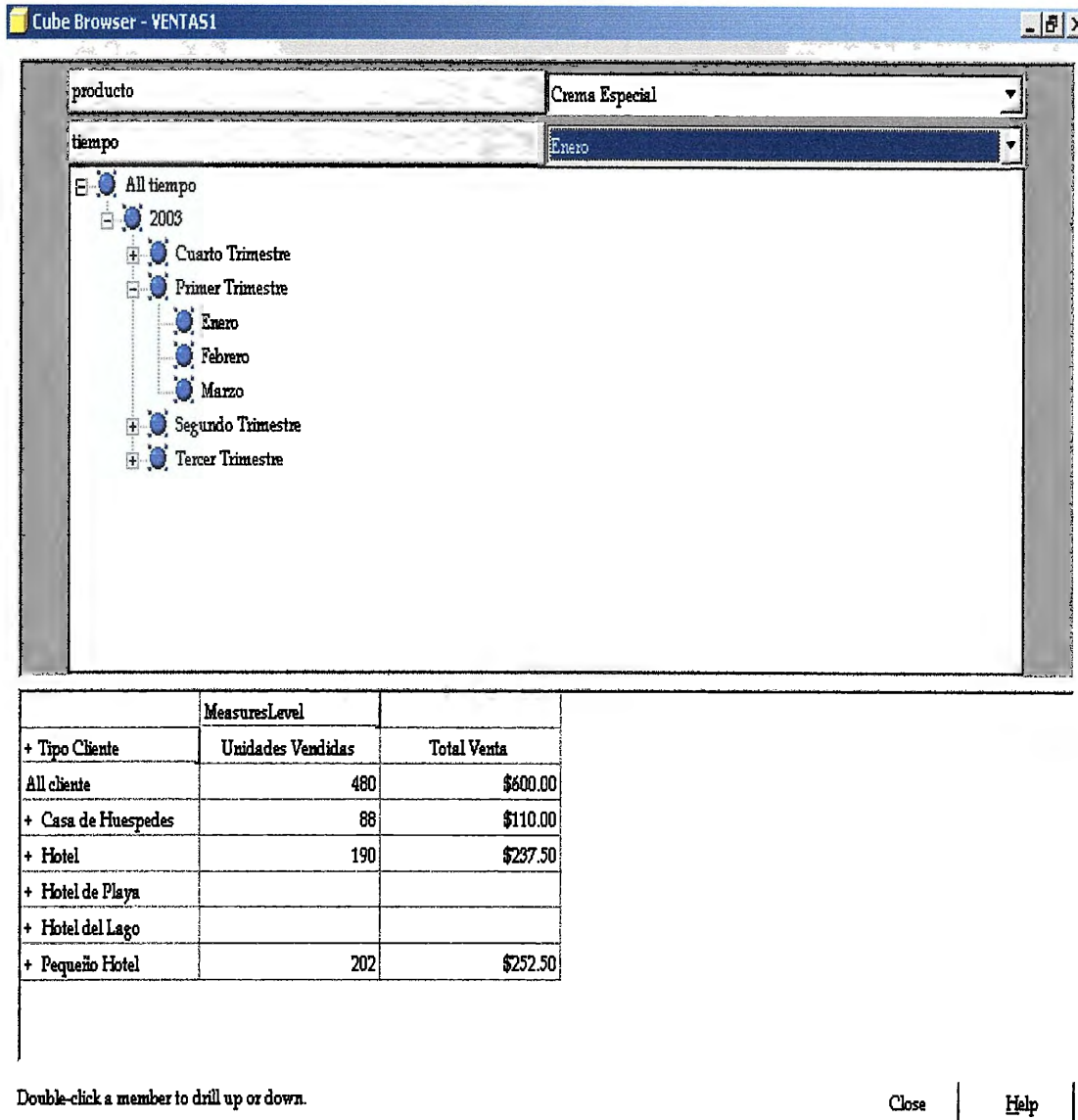


Figura 3.23 Estructura del cubo

El cubo Ventas1 presenta un formato de esquema estrella con un modo de almacenamiento MOLAP, igual que los anteriores y analiza alrededor de 40,000 registros.

Será interesante conocer el comportamiento de las ventas de la Crema Fresca en un periodo determinado, un ejemplo sería analizar las ventas de enero.



**Figura 3.24 Datos cubo Ventas1**

Se observa que para la dimensión Cliente existe un primer nivel que es Tipo cliente, la cual presentará como primera información los totales de venta para Casa de Huéspedes, Hotel, Hotel de Playa, Hotel de Lago y Pequeño Hotel para el Mes de Enero y para Crema Especial.

### Ventas2

Este cubo a diferencia de Ventas1 presentará información basada en una dimensión adicional denominada Localización, así además de conocer el producto, el cliente y el tiempo, se puede saber el lugar donde se realiza la venta. La estructura del cubo se muestra en la Figura 3.25.

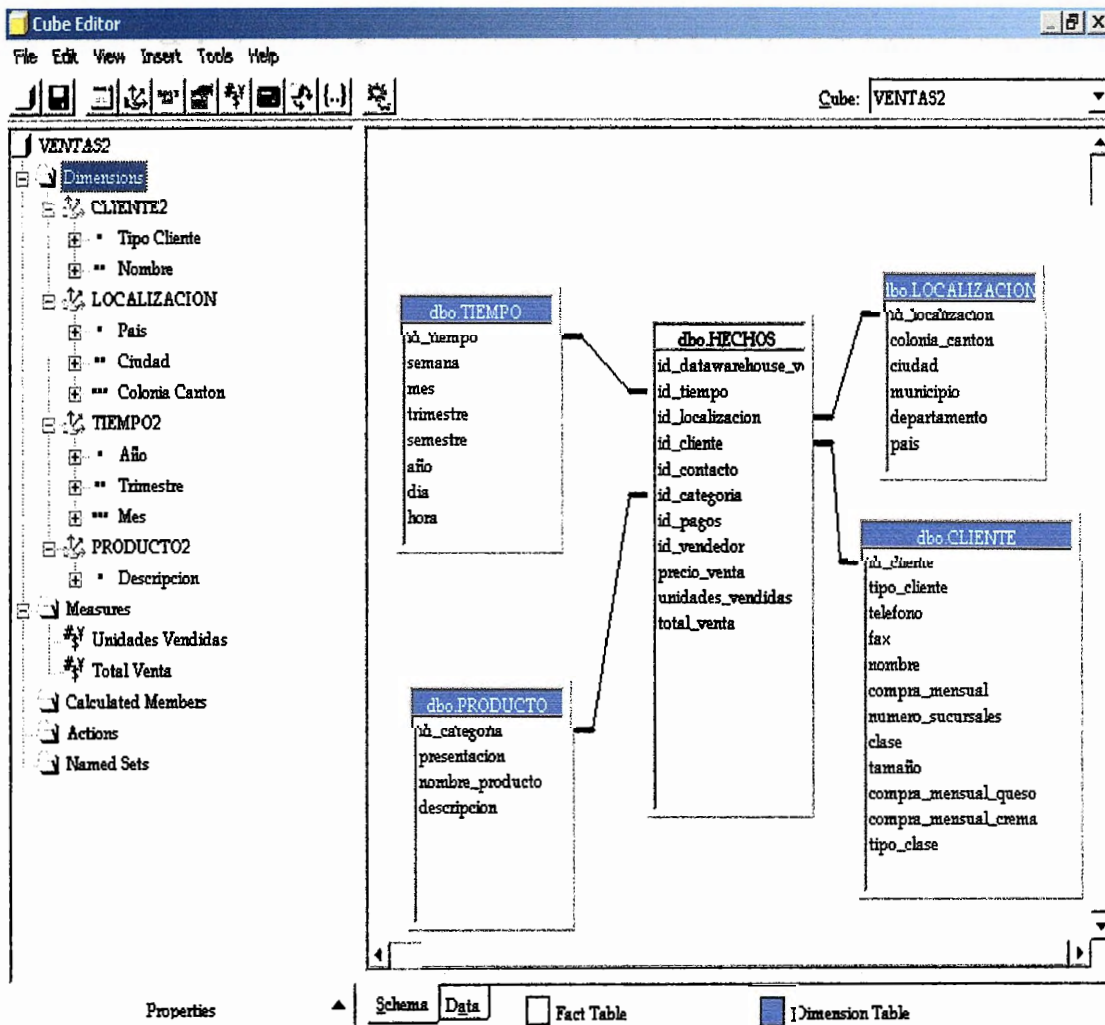
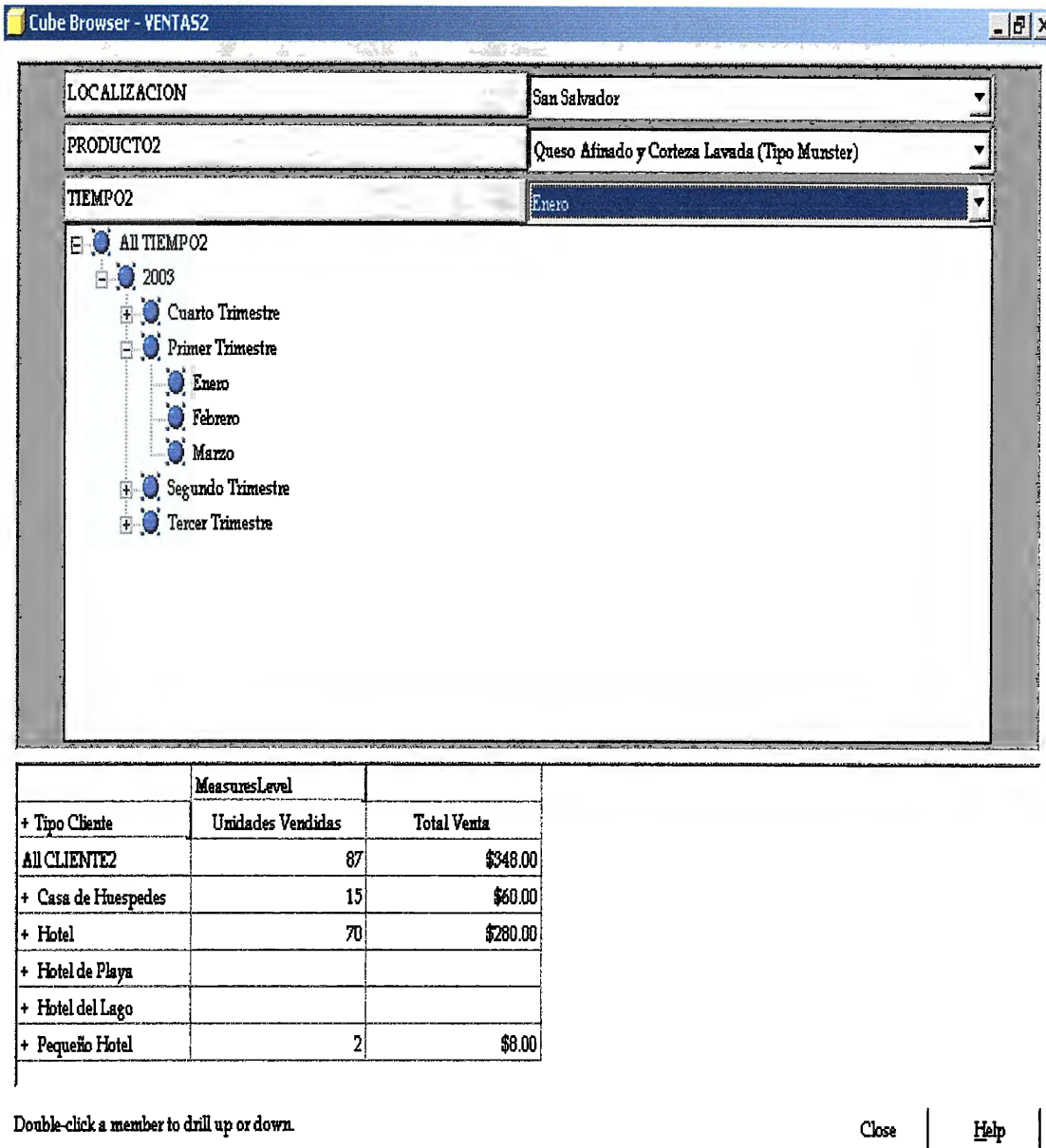


Figura 3.25 Cubo Ventas2

El estudio de variables multidimensionales ofrece una mejor perspectiva de los datos analizados, mientras más dimensiones existan en un cubo, mejor será el desglose de los datos.

La figura 3.26 presenta los resultados del cubo Ventas 2 , resultaría interesante conocer las unidades vendidas de Queso Afinado y Corteza Lavada (Tipo Munster) y el total de ventas para el mes de enero de los clientes de la empresa.



**Figura 3.26 Resultados Cubo Ventas2**

## Datawarehouse Ventas

El cubo Datawarehouse Ventas presenta las comparaciones de los productos por periodos, los años contemplados son 2002 y 2003. La estructura de este cubo se enfoca en dos medidas y una sola dimensión debido a que la información que se analiza se presenta en una sola tabla, Datawarehouse\_Ventas.

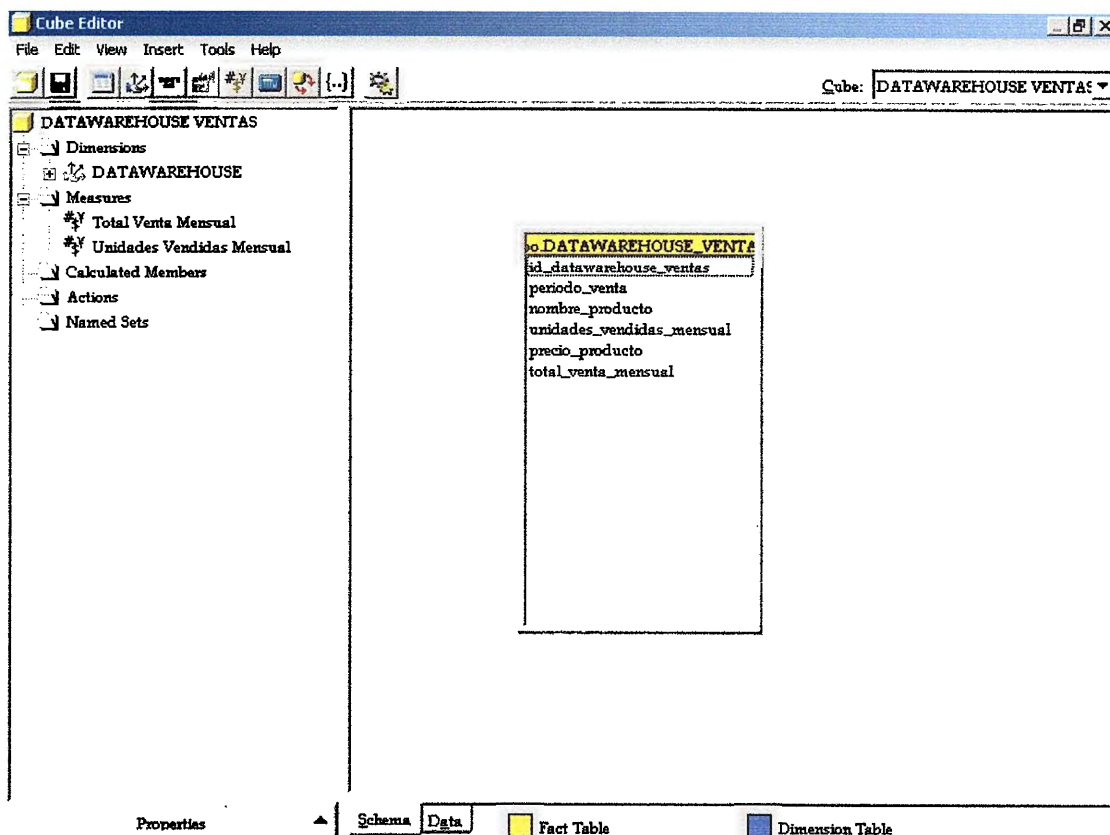


Figura 3.27 Datawarehouse\_Ventas

La Figura 3.27 muestra la estructura básica de este cubo presentando únicamente dos medidas y una sola dimensión, necesarias para el ejemplo.

Cube Browser - DATAWAREHOUSE VENTAS

+ Periodo Venta	MeasuresLevel	
	Total Venta Mensual	Unidades Vendidas Mensual
All DATAWAREHOUSE	\$373,943.20	137,902
+ Abril 2002	\$15,880.50	6,781
+ Abril 2003	\$18,368.72	7,643
+ Agosto 2002	\$15,880.50	6,781
+ Agosto 2003	\$18,390.72	7,667
+ Diciembre 2002	\$15,880.50	6,781
+ Enero 2002	\$15,880.50	6,781
+ Enero 2003	\$18,366.72	7,641
+ Febrero 2002	\$15,880.50	6,781
+ Febrero 2003	\$18,190.72	7,667
+ Julio 2002	\$15,880.50	6,781
+ Julio 2003	\$18,337.72	7,655
+ Junio 2002	\$15,880.50	6,781
+ Junio 2003	\$18,328.22	7,637
+ Marzo 2002	\$15,880.50	6,781
+ Marzo 2003	\$18,337.72	7,655
+ Mayo 2002	\$15,880.50	6,781
+ Mayo 2003	\$18,390.72	7,667
+ Noviembre 2002	\$15,880.50	6,781
+ Octubre 2002	\$15,880.50	6,781
+ Octubre 2003	\$18,337.72	7,655
+ Septiembre 2002	\$15,880.50	6,781
+ Septiembre 2003	\$18,328.22	7,643

Double-click a member to drill up or down.

Close Help

**Figura 3.28 Resultados Datawarehouse\_Ventas**

Finalmente la figura 3.28, muestra las comparaciones de los periodos del 2002 versus los del 2003 teniendo como medidas el total de venta mensual y las unidades vendidas mensuales, este ejemplo es útil para hacer comparaciones de ventas en periodos similares pero en diferentes años.

### **3.6 Minería de Datos**

La tecnología data mining analiza los datos de bases relacionales y cubos OLAP para descubrir información de interés. Las características de minería de datos del Análisis Service de Microsoft SQL Server 2000 se incorporan en una implementación abierta y extensible de la nueva especificación OLE DB para minería de datos.

#### **3.6.1 Modelos de la Minería de Datos**

Análisis Service de Microsoft SQL Server 2000 introduce la minería de datos, una nueva característica que integra funciones de análisis de datos y de predicción. Además de las extensiones de Análisis Manager, también se han realizado de los Objetos de ayuda para la toma de decisiones (DSO).

Análisis Service admite la minería de datos tanto de orígenes relacionales como multidimensionales. Sus algoritmos pueden entrenar modelos de minería con datos procedentes de cualquier origen relacional, que admita accesos a OLE DB, así como de cubos multidimensionales creados con él.

Análisis Service se puede ampliar para su uso con herramientas de otro fabricante, como componentes de visores y de minería de datos, que proporcionan una mayor flexibilidad y mejoras.

- Se ha mejorado la compatibilidad OLE DB mediante la especificación de OLE DB para minería de datos.
- Se ha actualizado Análisis Manager y se le han agregado nuevos asistentes y herramientas que le ayudarán a diseñar, crear, entrenar y examinar modelos de minería de datos. Las dimensiones y los cubos virtuales también se pueden crear a partir de modelos de minería de datos OLAP.

- Se ha ampliado DSO con la inclusión del objeto MiningModel y otros objetos de ayuda auxiliares.

SQL Server 2000 incluye algoritmos de minería de datos entre los cuales se encuentra:

### **3.6.2 Microsoft decisión trees (árbol de decisión)**

El algoritmo de Microsoft Decision Trees se basa en la idea de clasificación, genera un árbol que podrá predecir el valor de una columna según las restantes del conjunto de entrenamiento. En consecuencia, cada nodo del árbol representa un caso particular para una columna. La decisión de dónde colocar este nodo la realiza el algoritmo, y en profundidades diferentes a sus elementos iguales pueden representar casos diferentes de cada columna. Por ejemplo, considerar la siguiente tabla la cual se ha tomado del modelo MINERIA VENTAS CLIENTE en el cual se necesita analizar que tipo de programa de descuento se dará a cada cliente dependiendo de las compras mensuales que estos realicen

En los conceptos clave que hay tras el algoritmo del árbol de decisión: una columna se puede usar en más de una ubicación del árbol, y su importancia en la predicción puede cambiar en consecuencia.

### **3.6.3 Generar y utilizar modelos de minería de datos**

La minería de datos, una atractiva característica introducida como parte de Análisis Service de Microsoft SQL Server 2000, proporciona nuevas herramientas para los análisis de ayuda para la toma de decisiones por medio del descubrimiento de patrones y reglas de datos y de su utilización para los análisis de predicción, mediante algoritmos de minería de datos estándar de la industria.

El mecanismo principal de la minería de datos es el modelo a implementar, un objeto abstracto que almacena la información en una serie de conjuntos de filas de esquema. Se puede obtener acceso fácilmente a los modelos mediante gran diversidad de herramientas. Se puede utilizar el Asistente para modelo de minería de datos para su creación y el Examinador de modelos de minería de datos para presentar el contenido en un formato gráfico.

Los modelos de minería de datos se pueden utilizarse para realizar sofisticados análisis de ayuda para la toma de decisiones en grandes cantidades de datos, ya sean relacionales u OLAP, y con una gran variedad de algoritmos.

### **3.6.4 Crear modelos de minería de datos**

Se utiliza el Asistente para modelo de minería de datos para crear nuevos modelos de minería de datos. El Asistente para modelo de minería de datos guía por los diversos pasos necesarios para establecer el tipo de modelo y generarlo, con el que se trabajará el conjunto de escenarios y elegir la técnica que se utilizará para construir un nuevo modelo de minería de datos.

Se utilizan dos tipos de modelos, basados en el tipo de datos del conjunto de escenarios que deben procesarse. Para procesar tablas de bases de datos relacionales habituales, se diseña un modelo de minería de datos relacional, mientras que el modelo de datos OLAP se diseña para procesar datos OLAP en forma de cubos.

Los modelos comparten los mismos esquemas y estructuras básicas, independientemente si se basan en datos relacionales u OLAP. El elemento más importante en su construcción es el escenario, el cual es un conjunto de columnas de minería de datos que se utiliza para definir la información que usará el modelo para identificar, estudiar y proporcionará los datos de predicción.

El tipo de contenido de las columnas de minería de datos es variable: Las columnas clave se utilizan para identificar un escenario específico, las de entrada se usan para proporcionar información que podrá utilizar el proveedor de minería de datos para analizar un escenario y las de predicción proporcionan resultados significativos, en forma de datos de histograma, procedentes del escenario analizado.

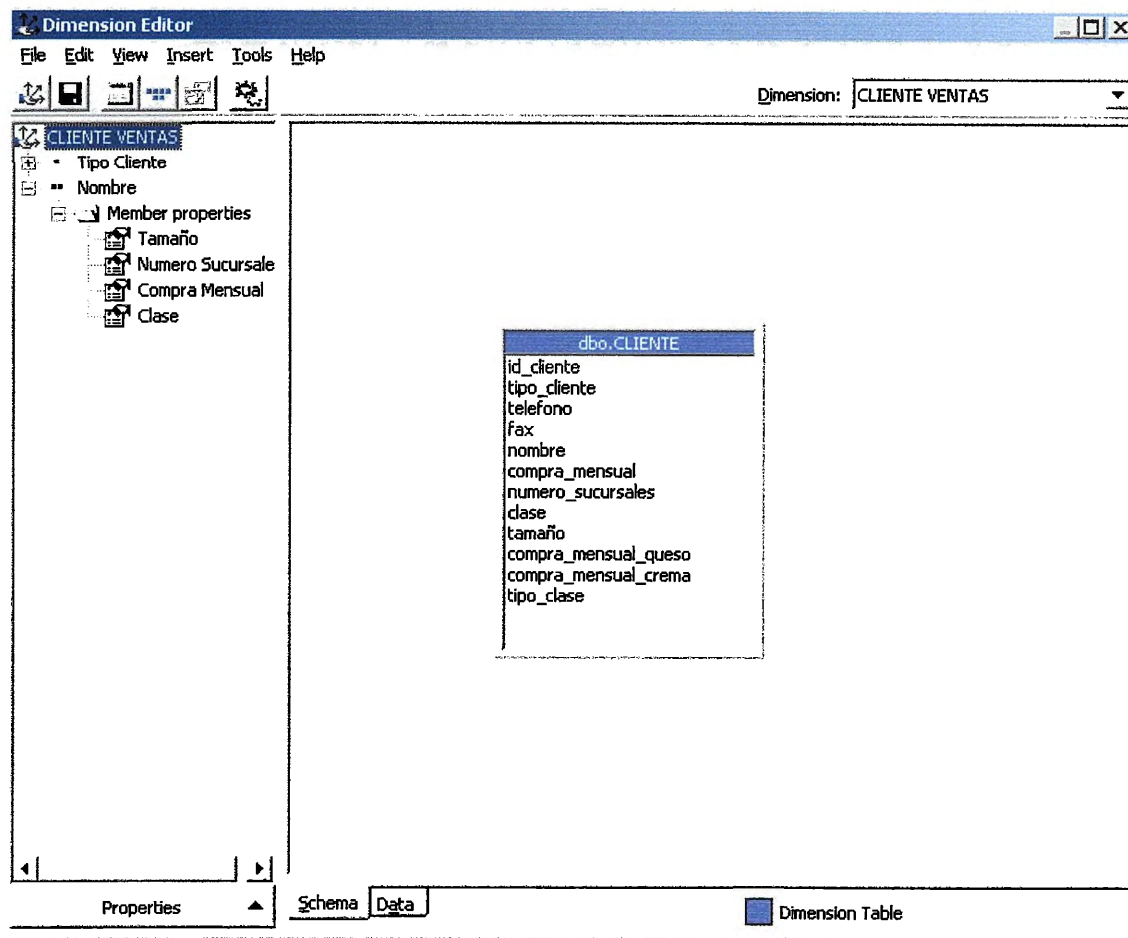
### **3.6.5 Crear modelos de minería de datos OLAP**

El proceso de creación de un modelo de minería de datos OLAP es similar al de un modelo relacional. Para crear un modelo de datos OLAP, se utiliza el asistente para modelo de minería de datos. Los pasos necesarios para la creación de un modelo OLAP son casi idénticos a los que se necesitan para crear un relacional.

El Asistente utiliza el cubo de origen, la técnica de minería de datos, la dimensión y el nivel del escenario, la entidad que se va a predecir y los datos de entrenamiento para crear un modelo de minería de datos OLAP.

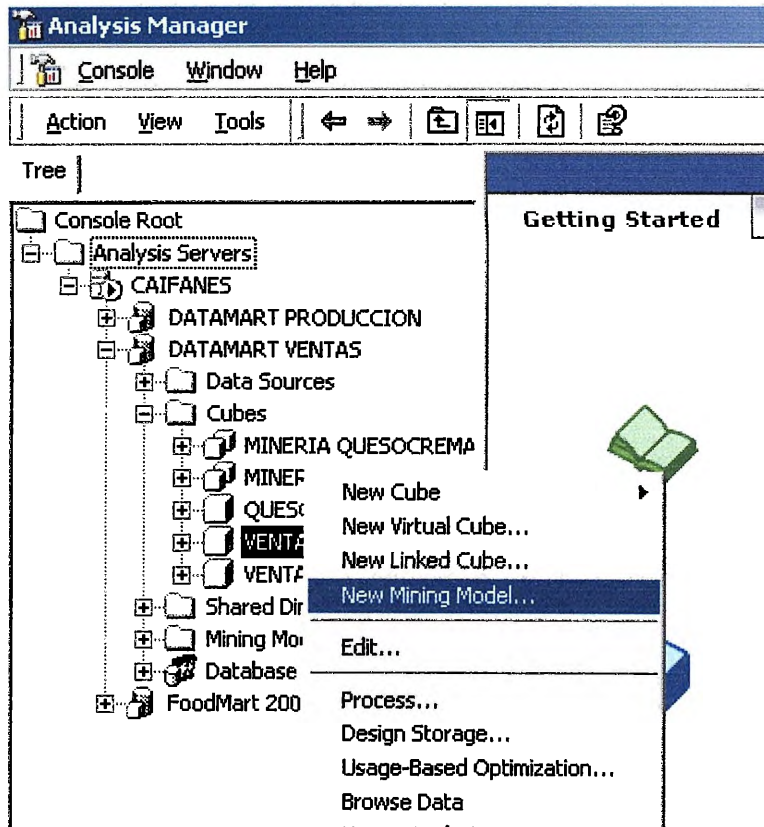
Para poder determinar el análisis de la minería de datos en el cubo VENTAS es necesario especificar los campos o columnas del modelo de minería, se han creado para la aplicación de un programa de descuento (Figura 3.29) las columnas Clase, Compra Mensual, Numero Sucursales y Tamaño.

La dimensión que se utiliza para este ejemplo es CLIENTE VENTAS, la columna que define el modelado es Nombre a partir de esta se crea los diferentes niveles de análisis y se determina la clase u objeto a analizar



**Figura 3.29 Análisis de la minería de datos en el cubo VENTAS**

El cubo de origen (Figura 3.30) proporciona al Asistente para modelo de minería de datos la información necesaria para crear el conjunto de escenarios.

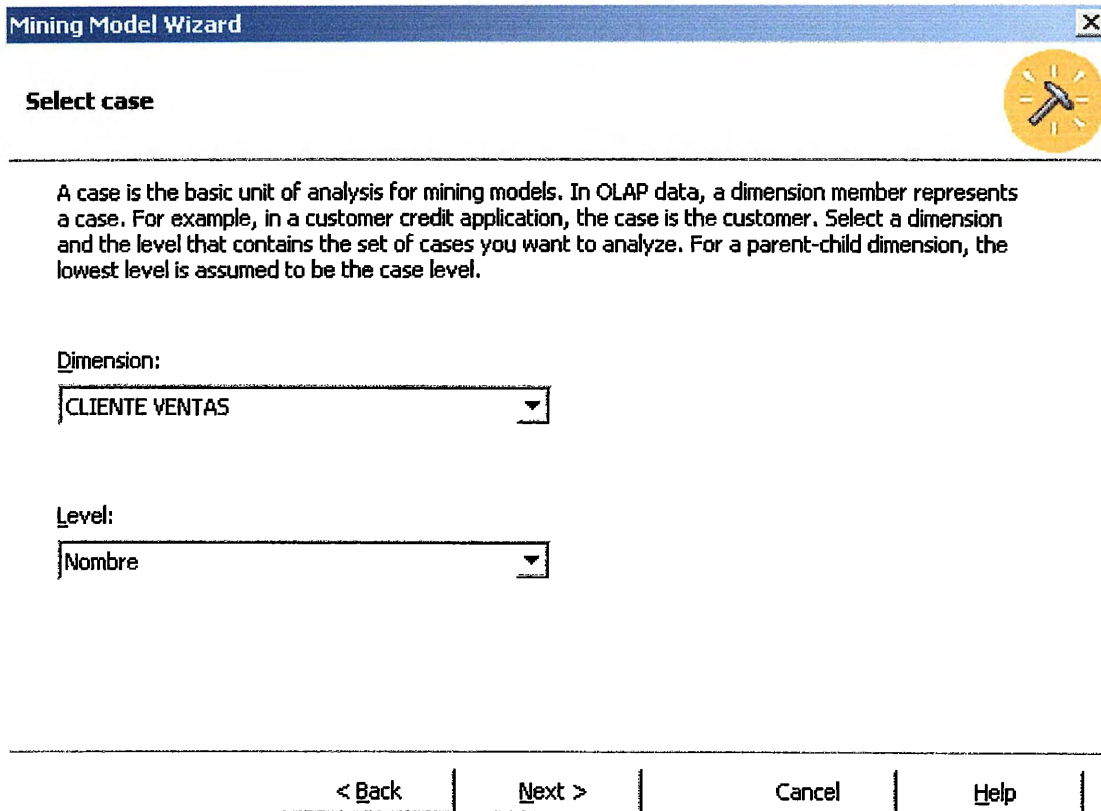


**Figura 3.30 Analysis Manager –nuevo modelo de minería de datos-**

Ya que los cubos pueden contener muchos grupos de información, el modelo utiliza la dimensión y el nivel del cubo indicados por el usuario para establecer las columnas de clave para el conjunto de escenarios.

Al seleccionar la técnica de minería de datos, también se selecciona el proveedor de minería de datos que proporciona los algoritmos (Figura 3.31) y la estructura para el modelo.



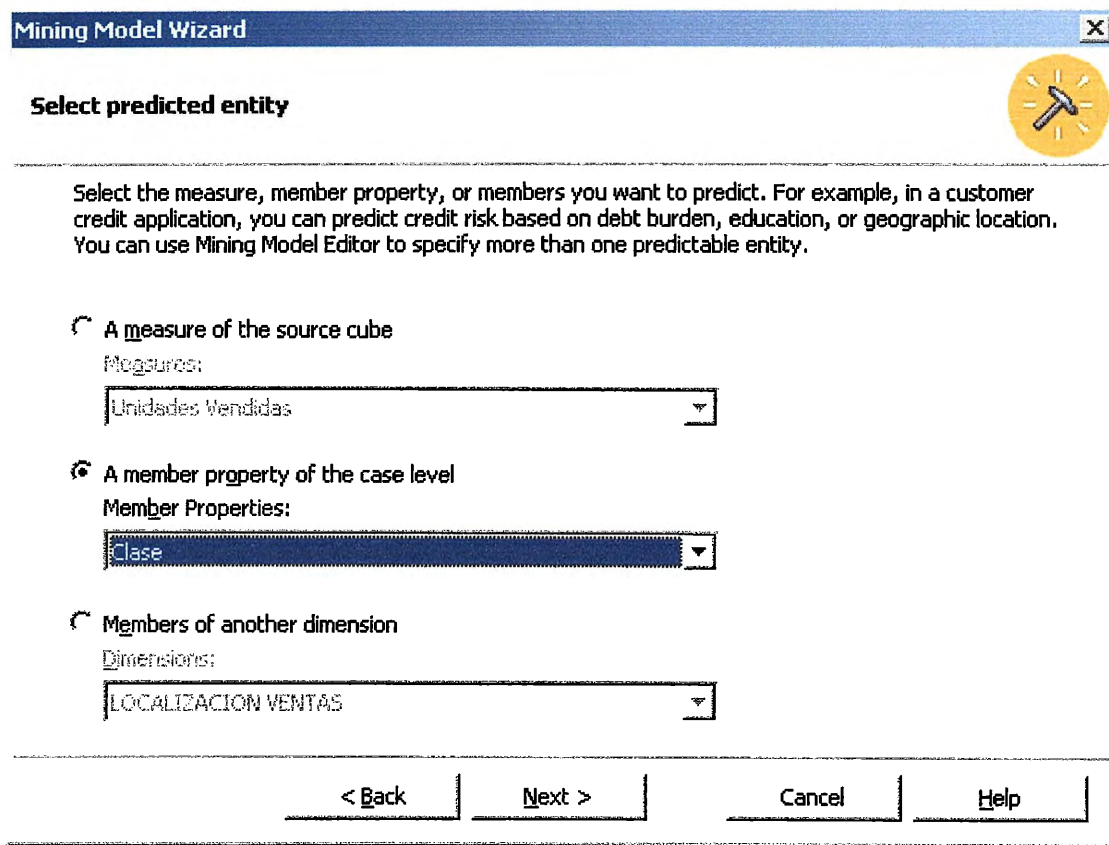


**Figura 3.32 La dimensión y el nivel del escenario**

La entidad que se va a predecir (Figura 3.33) puede ser una de las entidades siguientes:

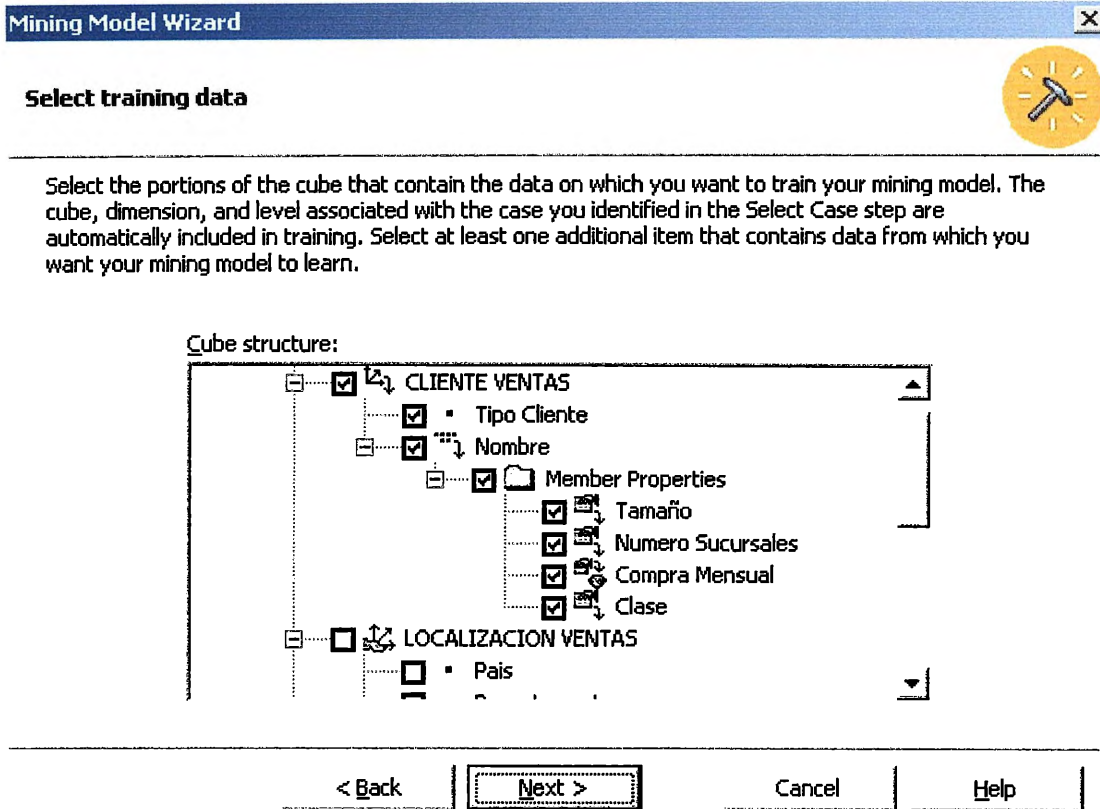
- Una medida del cubo de origen
- Una propiedad de miembro de la dimensión y nivel del escenario, seleccionada previamente en el Asistente para modelo de minería de datos.
- Miembros de otra dimensión del cubo

Esto proporciona flexibilidad al tratar con procesos potencialmente complejos de análisis de predicción de datos OLAP.



**Figura 3.33 Entidad a predecir**

Los datos de entrenamiento (Figura 3.34), en forma de dimensiones, niveles, propiedades de miembros y medidas, se utilizan para procesar el modelo de minería de datos OLAP y definir con más detalle la estructura de columnas del conjunto de escenarios.



**Figura 3.34 Datos de entrenamiento**

### 3.6.6 Columnas de minería de datos

Se puede ver y cambiar el comportamiento y la estructura de las columnas del data mining, mediante el editor de modelos relacionales o el editor de modelos OLAP de minería de datos. En ambos editores, el panel de estructura contiene las columnas utilizadas para definir el modelo; las propiedades de cada columna, como tipo de datos y tipo de contenido, pueden verse en el panel de propiedades.

Las columnas de minería de datos se agregan al modelo en los distintos pasos del Asistente, según el tipo que se elija. Seleccionar la entidad que se va a predecir, en este caso se trata de la entidad Clase, se crea las columnas de predicción y el paso Seleccionar los datos de entrenamiento crea las columnas de entrada de minería de datos.

Debido a la diversidad de posibles proveedores de algoritmos de minería de datos, las definiciones de columnas tienen un diseño flexible y ampliable.

Opcionalmente, el asistente puede crear una nueva dimensión para el cubo de origen o un cubo virtual basado en el cubo de origen (Figura 3.35). Esto permite realizar consultas en los datos del data mining de la misma manera que harían en los datos OLAP.

**Mining Model Wizard**

**Create a dimension and virtual cube (optional)**

You can create a new dimension from the results of the mining model analysis of OLAP data. You can also optionally create a virtual cube that contains the created dimension and the original cube. You can browse this virtual cube to see the results of the data mining analysis.

Create a new dimension based on this mining model

Dimension name:

Create a new virtual cube

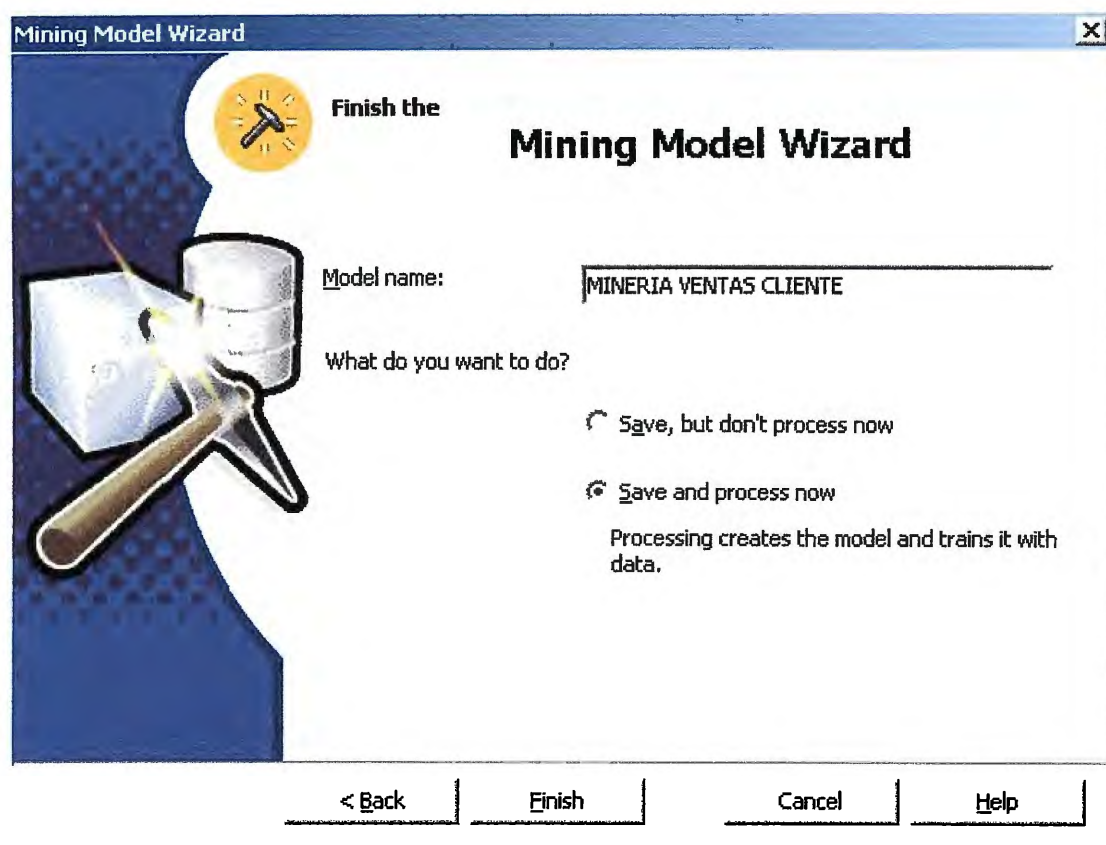
This cube will contain the source cube on which the mining model is based and the new dimension you specify in this wizard step.

Virtual cube name:

< Back    Next >    Cancel    Help

**Figura 3.35 Creación de una nueva dimensión**

Después de este paso se guarda y procesa el modelo de la Minería de Datos (Figura 3.36)



**Figura 3.36 Finalización del modelo de minería de datos**

### 3.6.7 Requisitos previos

Las columnas de minería de datos OLAP se construyen a partir de niveles visibles del cubo en que se basa el modelo. Para crear un modelo de minería de datos, el Asistente requiere un cubo cuyas dimensiones contengan al menos un nivel visible.

Los modelos de minería de datos no se pueden crear a partir de orígenes de datos relacionales, y las dimensiones virtuales creadas con los Servicios OLAP de Microsoft SQL Server 7.0 no se pueden utilizar en los modelos OLAP. Además, un modelo no se puede basar en un cubo virtual que contenga una dimensión de minería de datos.

### **3.6.8 Modificar modelos de minería de datos OLAP**

Se puede utilizar el Editor de modelos de minería de datos OLAP para modificar la estructura ellos. Y, de forma parecida sucede con el Editor de modelos relacionales de minería de datos, este puede procesar un modelo y ver el contenido resultante.

El Editor de modelos de minería de datos OLAP permite cambiar propiedades básicas de ellos, como el algoritmo seleccionado. También permite modificar las propiedades de las dimensiones, niveles y medidas que componen el conjunto de escenarios definido para el modelo.

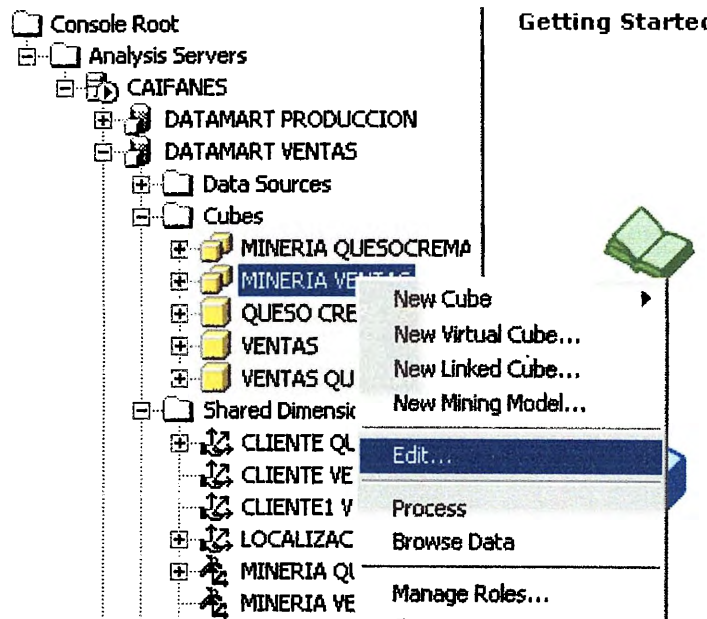
Si el modelo se ha entrenado, el Editor también muestra el contenido del modelo gráficamente, mediante el Examinador de modelos de minería de datos.

### **3.6.9 Editor de Modelos de Minería de Datos OLAP**

Utilizamos esta herramienta para examinar el contenido de un modelo de minería de datos OLAP y modificar su estructura.

El Editor de modelos de minería de datos OLAP aparece cuando realiza la siguiente

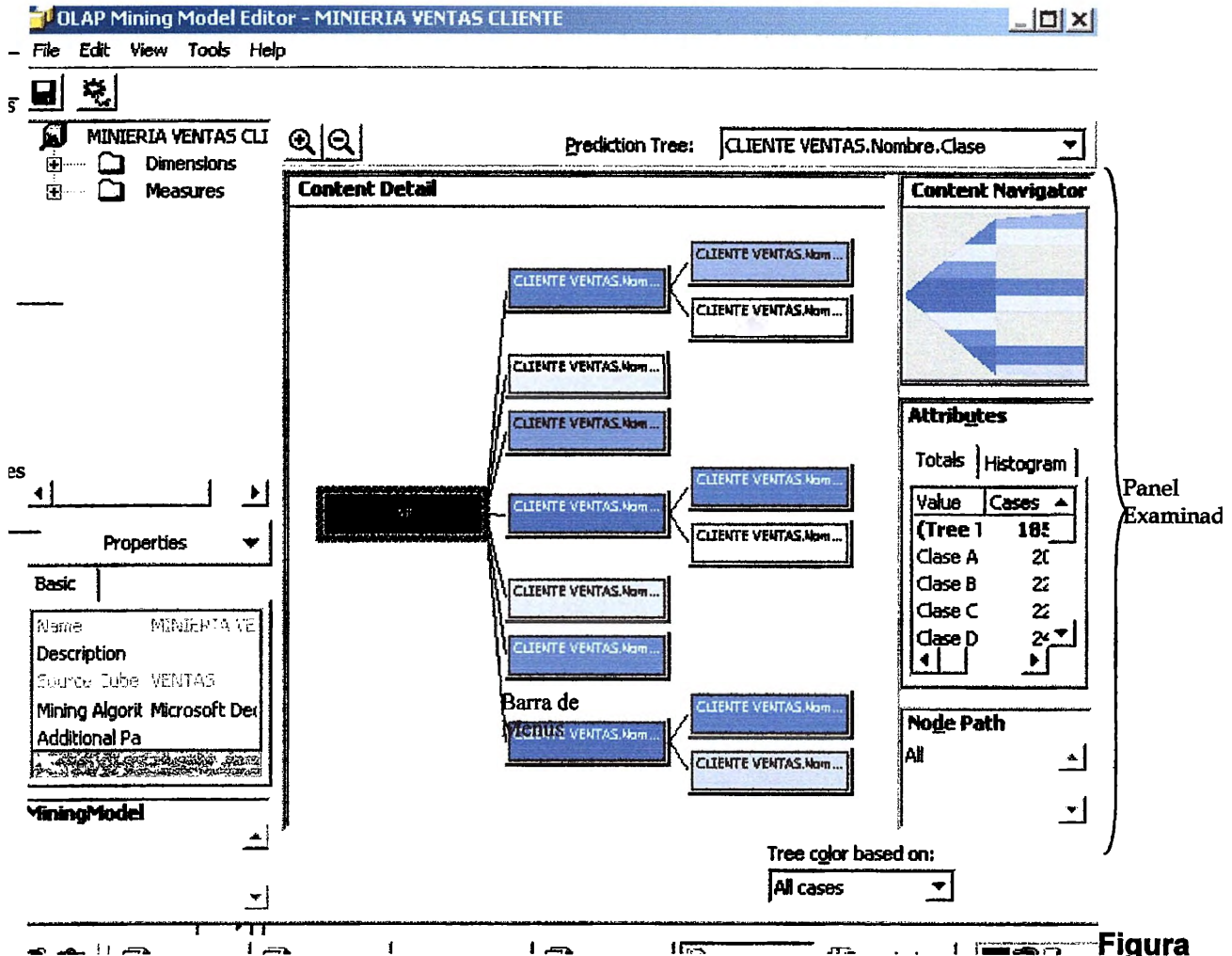
- En el panel de árbol del Análisis Manager, hacer clic con el botón secundario (derecho) del *mouse* (ratón) en uno de los modelos y a continuación hacer clic en Edit



El Editor de modelos de minería de datos OLAP (Figura 3.37) tiene cinco elementos, así como se muestra en la siguiente tabla:

Para	Opción a utilizar
Realizar varias tareas básicas en el Editor de modelos de minería de datos OLAP.	Barra de menús
Realizar funciones habituales en el Editor de modelos de minería de datos OLAP.	Barra de herramientas
Ver la estructura del modelo de minería de datos OLAP, como las columnas de entrada y de predicción. Modificar la estructura del modelo de minería de datos OLAP.	Panel de estructura
Ver y modificar las propiedades asociadas al modelo de minería de datos	Panel de propiedades

<p>OLAP y sus columnas de minería de datos. Ver las descripciones de las propiedades y los objetos a los que se aplican.</p>	
<p>Mostrar el contenido del modelo de minería de datos OLAP procesado.</p>	<p>Panel examinador</p>



3.37 Editor de modelos de minería de datos OLAP

### 3.7 MDX

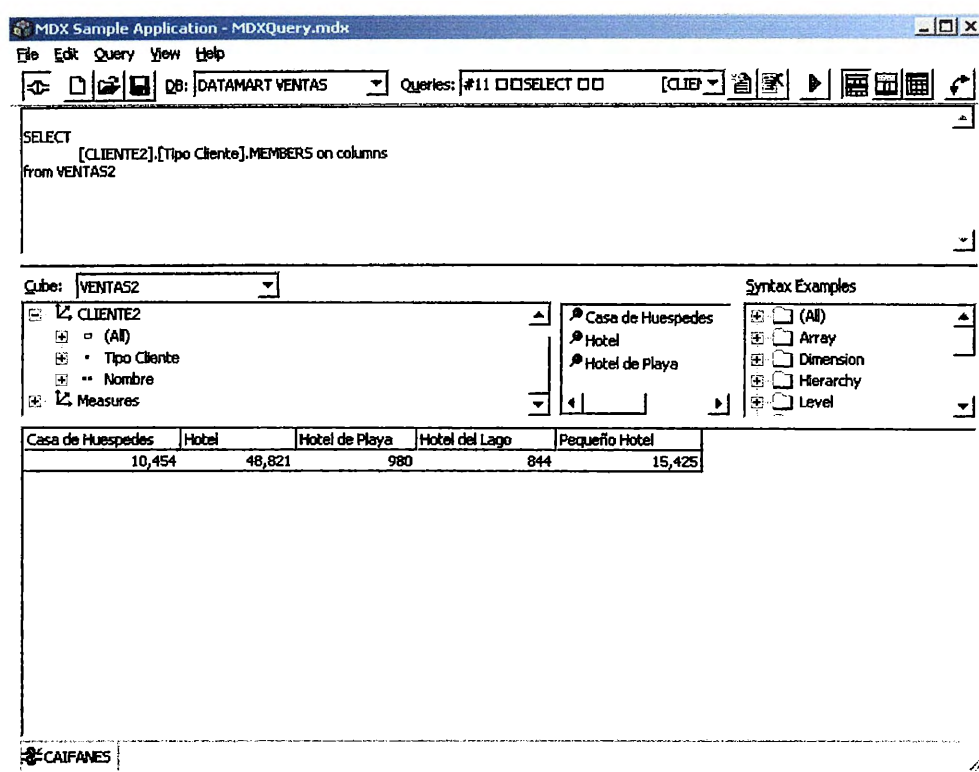
El lenguaje de expresiones multidimensionales (MDX) se utiliza para manipular información de muchas dimensiones en el Análisis Service de Microsoft SQL Server 2000. MDX está definido en las extensiones OLAP de OLE DB. De manera similar a SQL en muchos aspectos, MDX proporciona una rica y poderosa sintaxis para la recuperación y manipulación de datos multidimensionales, como los datos almacenados en cubos en Análisis Server.

MDX, acrónimo del inglés Multidimensional Expressions (expresiones multidimensionales), es una sintaxis que admite la definición y manipulación de objetos y datos multidimensionales. Es similar en muchos aspectos a la sintaxis de SQL (Structured Query Language, lenguaje de consulta estructurado), pero no es una extensión dicho lenguaje; de hecho, puede ofrecer parte de la funcionalidad de MDX, aunque no con tanta eficacia ni de manera tan intuitiva.

#### 3.7.1 Consulta para reportar el número de ventas por tipo de cliente en la Datamart ventas

```
SELECT  
    [CLIENTE2].[Tipo Cliente].MEMBERS on columns  
from VENTAS2
```

La consulta anterior mostrara en pantalla el total de ventas en el año por tipo de cliente, tal como se muestra en la Figura 3.38.



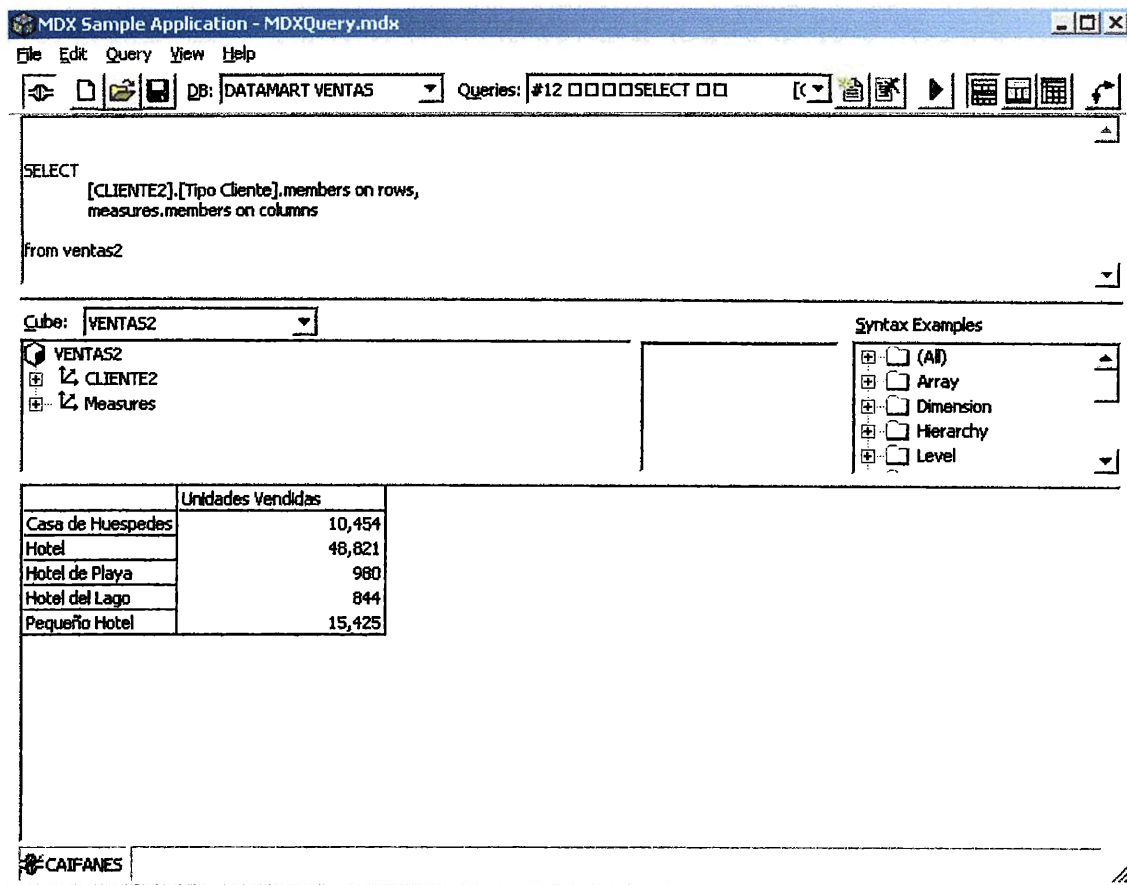
**Figura 3.38 Consulta MDX ventas por tipo cliente en Datamart Ventas**

### 3.7.2 Consulta para agregar dos dimensiones Datamart ventas

```
SELECT
  [CLIENTE].[Tipo Cliente].members on rows,
  measures.members on columns

from ventas1
```

Con esta consulta se ven los datos en dos dimensiones (2 ejes) los cuales nos permitirán encontrar las unidades vendidas en el año para cada categoría de cliente, tal como se muestra en la Figura 3.39.



**Figura 3.39 Consulta para agregar dos dimensiones Datamart Ventas**

Una consulta más profunda muestra información de dos dimensiones en el cubo Ventas. El resultado será el total de las unidades vendidas para cada categoría de cliente con respecto a los productos. Figura 3.40

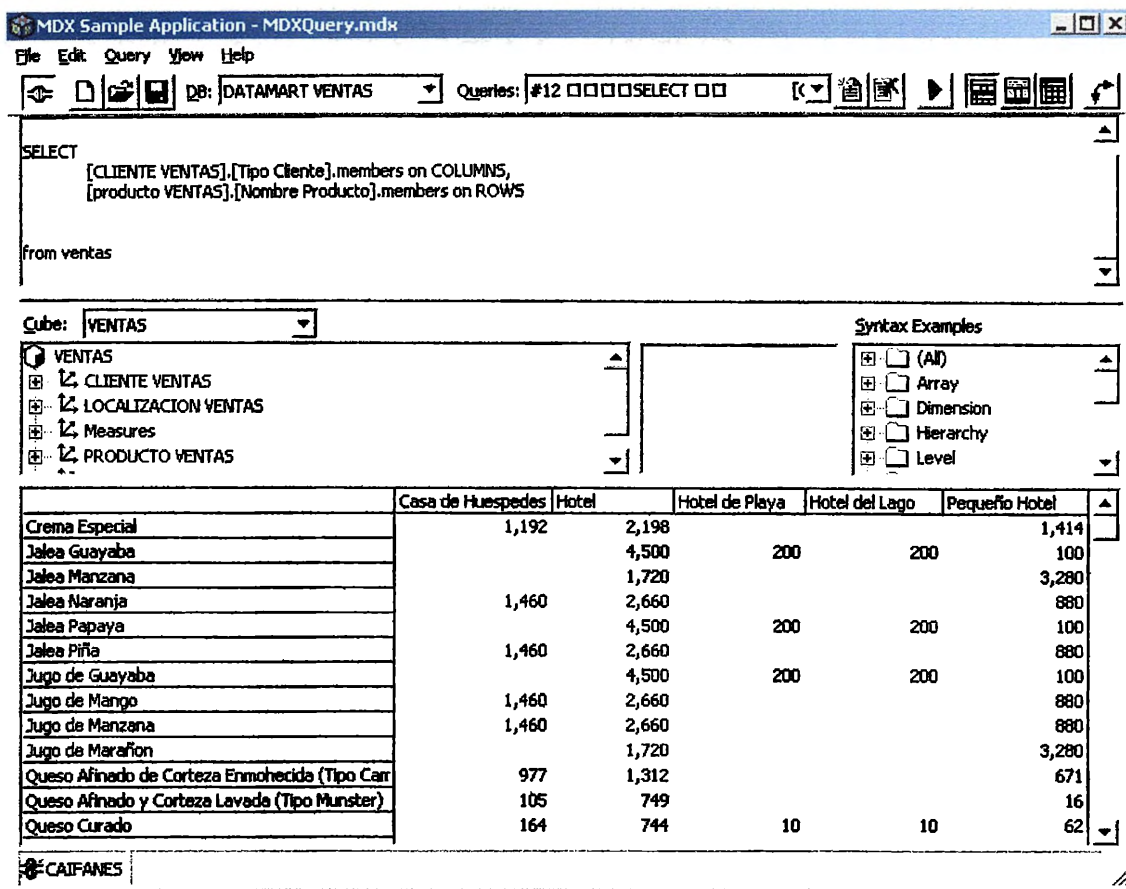
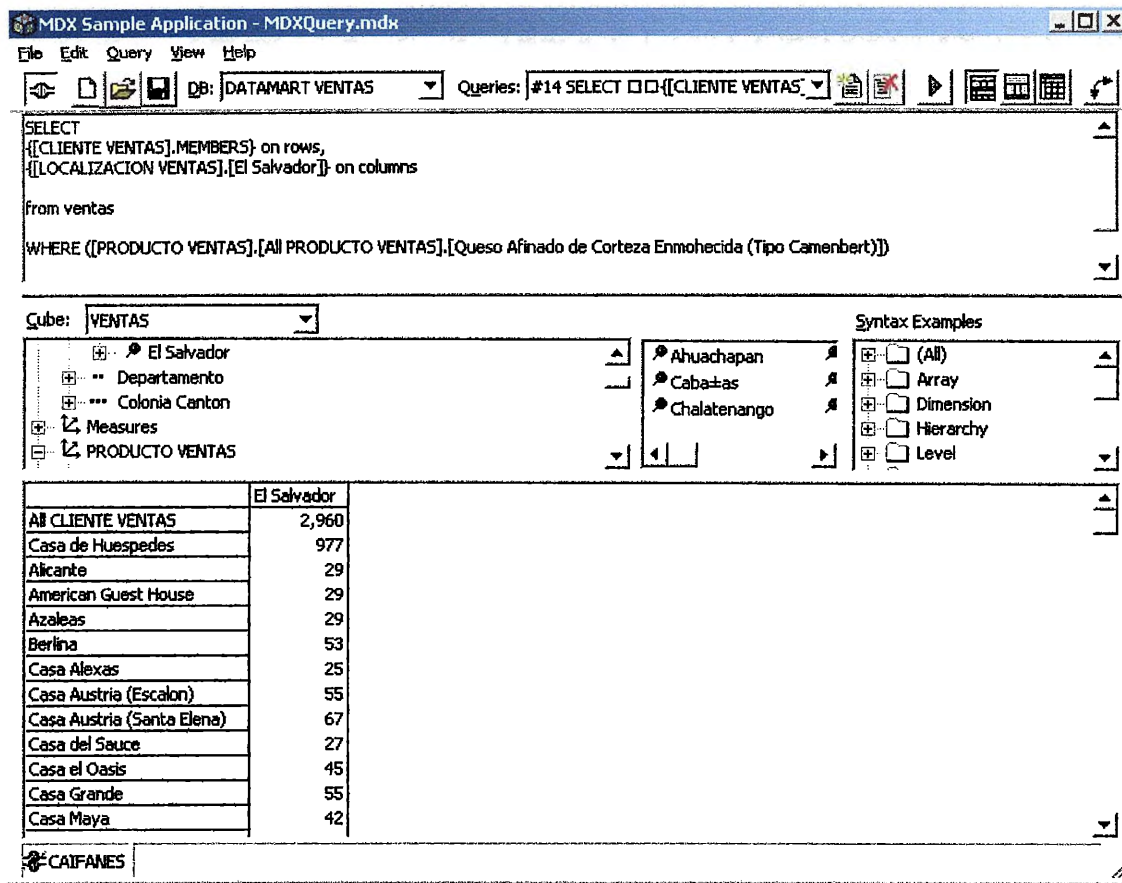


Figura 3.40 Consulta de dos dimensiones Datamart ventas

### 3.7.3 Consulta que utiliza la sentencia WHERE

```
SELECT
  {[CLIENTE VENTAS].MEMBERS} on rows,
  {[LOCALIZACION VENTAS].[El Salvador]} on columns
from ventas
WHERE ([PRODUCTO VENTAS].[All PRODUCTO VENTAS].[Queso Afinado de
Corteza Enmohecida (Tipo Camenbert)])
```



**Figura 3.41 Consulta MDX que utiliza la sentencia WHERE**

La Figura 3.41 muestra los totales de productos vendidos para cada cliente y específicamente para un tipo de producto, esto se logra utilizando la sentencia WHERE en la consulta.

De la misma manera se pueden agregar los campos que se crean convenientes en la sentencia WHERE para profundizar las consultas tal como se realiza en SQL, a diferencia que en MDX solo se analizan datos multidimensionales OLAP.

Se han presentado ejemplos de cubos, minería de datos y sentencias MDX, realizadas desde el Análisis Manager de SQL Server 200, con la intención de presentar en forma gráfica una parte de lo que se ha desarrollado en el presente trabajo y tener una idea de lo que se intento realizar. La herramienta que se hizo en esta investigación esta mejor explicada en el Manual el Usuario, con sus respectivas pantallas y comentarios, y la parte técnica en el Manual del Programador.

### 3.8 La herramienta OLAP Viewer.

*Olap Viewer* es el nombre que hemos asignado a la herramienta desarrollada en esta investigación, es una poderosa herramienta de análisis que permite la interacción con datos multidimensionales en los sistemas operativos con tecnología NT, apoyándose en las técnicas de minería de datos, cubos y servidores OLAP, aprovechando todas las ventajas que brinda Windows SQL Server 2000 y la capacidad de poder conectar con Visual Basic los objetos, componentes y referencias necesarias para finalizarla..

El sistema se conecta de manera directa con el Análisis Service de Windows SQL Server 2000, para recuperar los datos almacenados de forma multidimensional y mostrarlos al usuario en tablas dinámicas o gráficos dinámicos.

*Olap Viewer*, permite el acceso a cubos de datos MOLAP, además de poder analizar modelos de minería de datos previamente definidos en SQL Server 2000. La herramienta también posee mantenimiento de datos para que el usuario pueda retroalimentar las bases de datos y poder actualizar la información para que los cubos cambien en relación con los movimientos de las variables de marketing que posee la empresa BIOLACT (empresa que sirve como modelo).

Otra característica importante es que OLAP Viewer posee:

- Asistente para la creación de cubos MOLAP.
- Asistente para la creación de dimensiones.

*Olap Viewer*, por otra parte tiene la capacidad de generar información proveniente de consultas MDX es decir consultas multidimensionales a partir de un cubo, permitiendo así la creación de reportes para la impresión en papel y ayudando al usuario a analizar de mejor manera los datos que están almacenados.

**OLAP Viewer** permite consultas SQL para información no multidimensional, dejando que el usuario realice consultas de una forma más sencilla.

Todo esto mediante la operabilidad y manejo que permiten los diseños de sistemas bajo el concepto de Data Warehouse, cubos OLAP y Minería de Datos, el cual descubre datos que no se pueden observar a simple vista, haciendo un mejor análisis de la información y ayudando a los usuarios a una mejor toma de decisiones.

Cuenta con ayuda para que el usuario pueda manejar más fácilmente la herramienta, ya que esta le guiara por todo el entorno, siendo comprensible y detallada.

En un ambiente agradable para la vista del usuario y de fácil manejo, la herramienta espera cumplir con los objetivos planteados al inicio de la investigación.

### 3.9 Diseño de la herramienta OLAP Viewer

#### 3.9.1 Inicio de Sesión.

El formulario para el inicio de sesión aparece en el centro de la pantalla de la herramienta “OLAP Viewer”, de color de fondo blanco y con unas dimensiones de:

Ancho = 4800.

Alto = 3165.

En el lado izquierdo hay una etiqueta de una combinación de colores que da un tipo de azul, con un valor hexadecimal de #H008F6000, con un alto de 3255 y un ancho de 755.

La etiqueta para el tema del formulario, en este caso “Inicio de Sesión”, tiene unas dimensiones de 315 de alto y 1755 de ancho, ubicada a 120 twips<sup>14</sup> abajo del borde superior del formulario y a 1800 twips del borde izquierdo, con tipo de letra Book Antiqua de tamaño 12 y estilo negrita.

Las etiquetas para el nombre y contraseña de usuario están ubicadas a 1080 twips del borde izquierdo del formulario y a 960 twips del borde superior del formulario la etiqueta nombre y 1320 la etiqueta contraseña, con un color rojo magenta, un tipo de letra MS Sans Serif, estilo en negrita y un tamaño de 8.

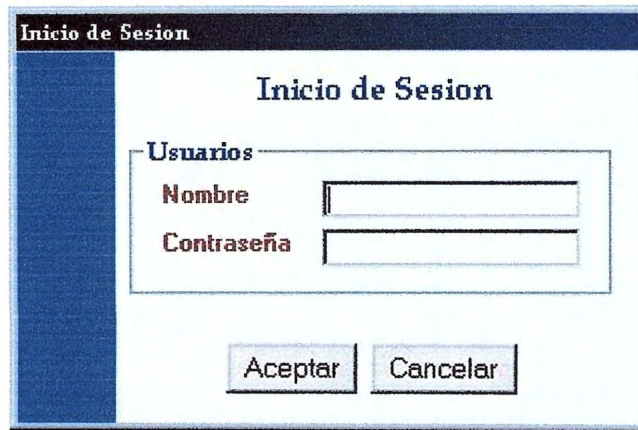
Las cajas de texto están ubicadas a 3120 tw. del borde izquierdo del formulario.

Los botones de Aceptar y Cancelar están ubicados a 2160 tw<sup>15</sup> del borde superior del formulario y tienen dimensiones de 375 y 975 tw de alto y ancho respectivamente. En la siguiente figura observamos el formulario.

---

<sup>14</sup> Un twips es la unidad de medida más pequeña para monitores, viene dada por 2 valores, alto y ancho.

<sup>15</sup> La abreviatura que hemos asignado a la unidad de medida twips es tw.



**Fig. 3.42 Inicio de Sesión**

### **3.9.2 Menú principal.**

Tiene un color de fondo con valor hexadecimal de H008F6F0A (Figura 3.43).

Tiene un icono de un cubo de 3d y el texto "OLAP Viewer" en la barra de título.

Dimensiones del formulario:

Alto =9000

Ancho =12000

Abarcará toda la pantalla al momento de ejecutarse, esto es, aparecerá maximizado.

Consta de una barra de Menú con 4 opciones Principales las cuales son:

Archivo, Edición, Utilidades, Ayuda.

El menú es en forma de cascada ya que cada opción principal tiene otras opciones secundarias y éstas a su vez tiene otra opciones llegando hasta 5 niveles (Fig. 3.44).

En la barra de estado del formulario presenta la hora el sistema al lado izquierdo de la pantalla, con el formato tradicional esto es hora:minuto:segundo am/pm, el logo de la Universidad Don Bosco en el centro de la barra con dimensiones de 16\*16 y la leyenda "Administrador de datos multidimensionales" al lado derecho.



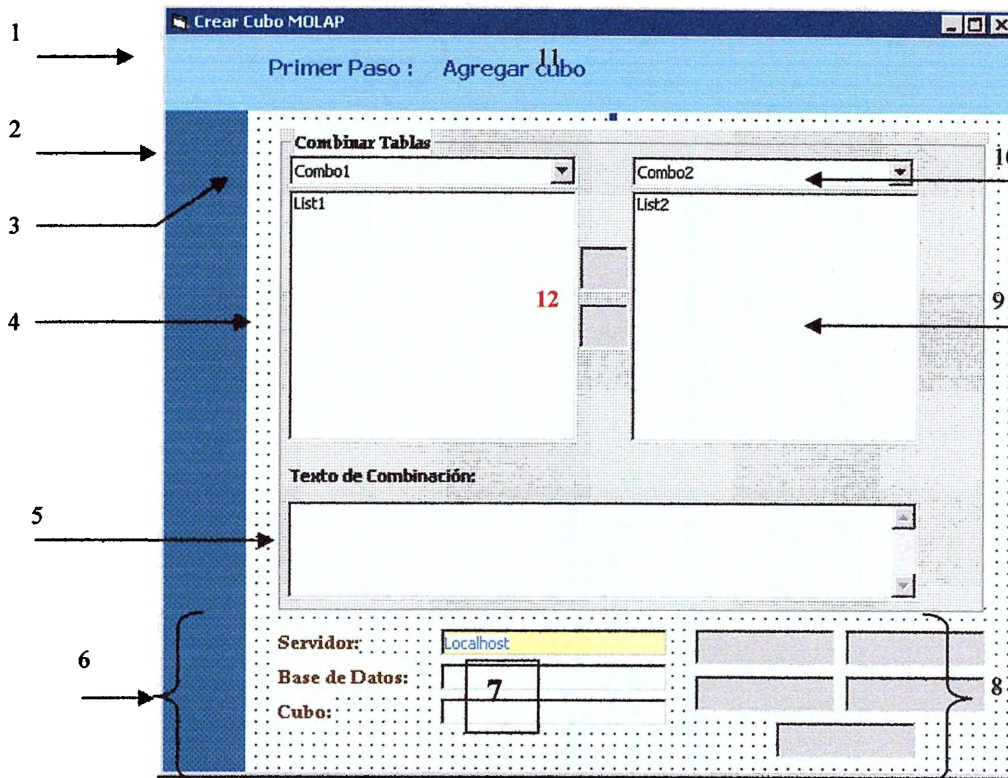
**Fig. 3.43 Menú Principal**



**Fig. 3.44 Menú Principal y sus 5 niveles**

### 3.9.3 Diseño para la opción crear cubo.

El diseño que se ha utilizado para crear un cubo es similar al diseño del Servidor de Análisis de Microsoft SQL Server 2000, a continuación explicamos el formulario (Figura 3.45) que sirve como asistente para la creación de un cubo:



**Fig. 3.45 Opción: Crear Cubo**

Las características generales del formulario Crear Cubo son:

En la barra de título tiene el texto “Crear Cubo MOLAP”.

Color blanco de fondo con relleno negro simulando cuadro.

Las dimensiones son: alto = 8010, ancho = 8940.

Objetos dentro del formulario:

- 1- PictureBox: con un color aqua de fondo, con 765 tw de alto y 9285 de ancho contiene una etiqueta (11), la cual le indica al usuario en el paso que se encuentra a la hora de crear el cubo, la letra es de color azul, de tipo Tahoma, de tamaño 11 y estilo negrita, se encuentra a 72 tw del borde izquierdo interno del PictureBox.

2- Etiqueta: de color azul con un ancho de 855 y un alto de 7815 tw.

3- ComboBox1:

Alto: 315

Ancho: 2985

Distancia desde el borde superior interno del frame combinado 270 y desde el borde izquierdo 90.

Letra tipo Tahoma, tipo regular y de tamaño 8.

4- List1: color de fondo blanco.

Alto: 2595

Ancho: 2985

Distancia desde el borde superior interno del frame combinado 630 y desde el borde izquierdo 90.

Letra tipo Tahoma, tipo regular y de tamaño 8.

5- TextBox (“Texto de Combinación”): fondo de color blanco.

Alto: 1005

Ancho: 6540

Distancia desde el borde superior interno del frame combinado 3840 y desde el borde izquierdo 90.

Letra tipo Tahoma, tipo regular y de tamaño 8.

6- Etiquetas (Servidor, Base de Datos, Cubo): letra tipo Platino linotype, tipo negrita y de tamaño 10, y color rojo marrón. Ubicadas a 1200 del borde interno izquierdo del formulario, con un alto de 270 cada una.

7- Cajas de texto (para el servidor, base de datos y cubo):

Alto: 285

Ancho: 2355

Distancia desde el borde izquierdo interno del formulario 2880.

Letra tipo Tahoma, tipo regular y de tamaño 8, alineado a la izquierda. El color de fondo para la caja de texto del servidor es amarillo, para las otras dos es blanco.

8- Botones (Tabla Hechos, Medidas, Dimensiones, Procesar, Salir)

Alto: 375

Ancho: 1455

Ubicadas al lado inferior derecho de la pantalla , con una distancia desde el borde superior interno del formulario de 6120 y desde el borde izquierdo 5520 (tomando como referencia el PictureBox de Tabla Hechos).

Letra tipo Tahoma, tipo regular y de tamaño 8.

9- List2: color de fondo blanco.

Alto: 2595

Ancho: 2985

Distancia desde el borde superior interno del frame combinado de 630 y desde el borde izquierdo de 3645.

Letra tipo Tahoma, tipo regular y de tamaño 8.

10- ComboBox2:

Alto: 315

Ancho: 2940

Distancia desde el borde superior interno del frame combinado de 270 y desde el borde izquierdo 3645.

Letra tipo Tahoma, tipo regular y de tamaño 8.

12- Botones de Agregar y Borrar:

Alto: 465

Ancho: 510

Distancia desde el borde superior interno del frame combinado de 1800 y desde el borde izquierdo de 3105.

Los comobox 1 y 2 las listas 1 y 2 los botones de agregar y borrar y la caja de texto se encuentran dentro de un frame, el cual tienen las siguientes dimensiones:

Alto: 4980

Ancho: 7320

Con una ubicación, desde los bordes internos del formulario de:

A la izquierda: 1200

Al superior: 960.

El diseño que se a utilizado para la creación de un nuevo cubo (MOLAP), es el mismo para los 5 pasos, variando la cantidad de controles en los formularios dependiendo del paso, la explicación del formulario anterior abarca todos los

controles utilizados en la creación de un cubo. Presentamos las 5 pantallas (Figura 3.46 a Figura 3.50) con el objetivo de hacer notar que el diseño es el mismo, no así el número de controles en ellos.

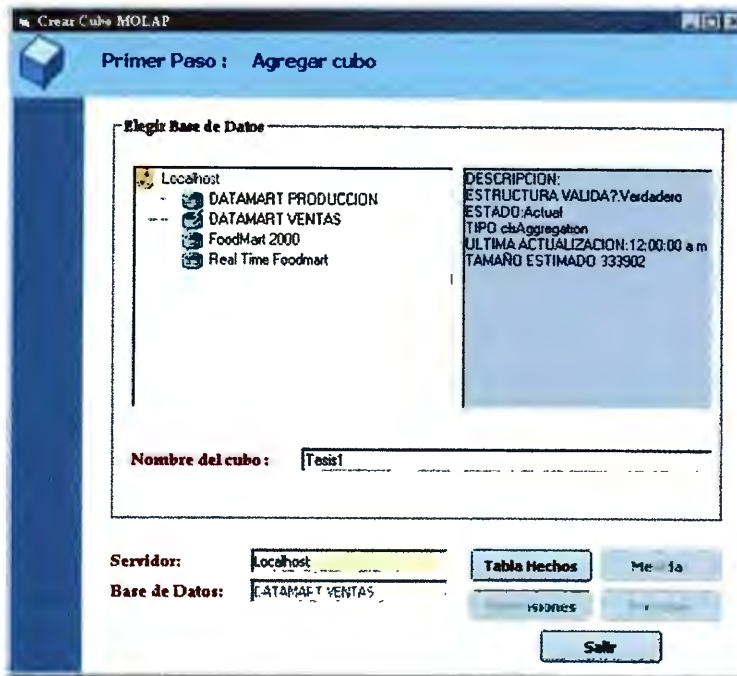


Fig. 3.46 Primer Paso: Agregar Cubo

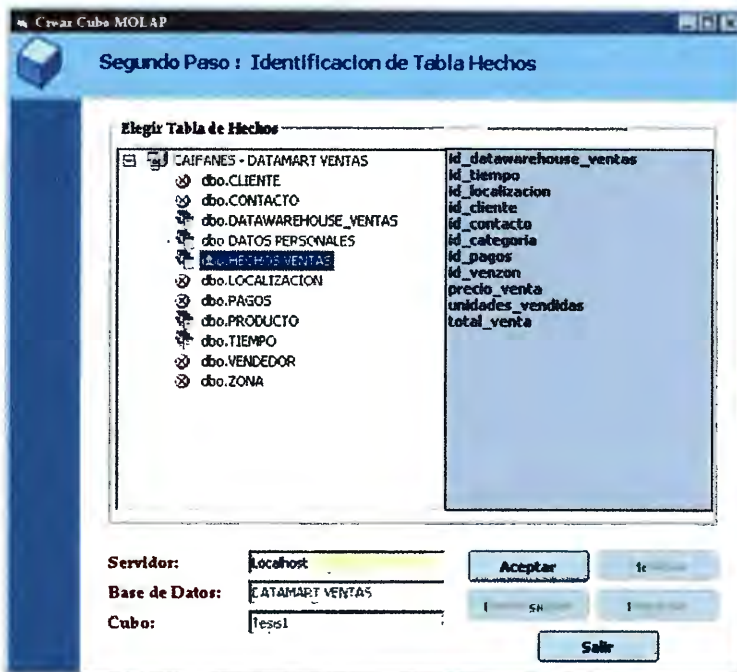
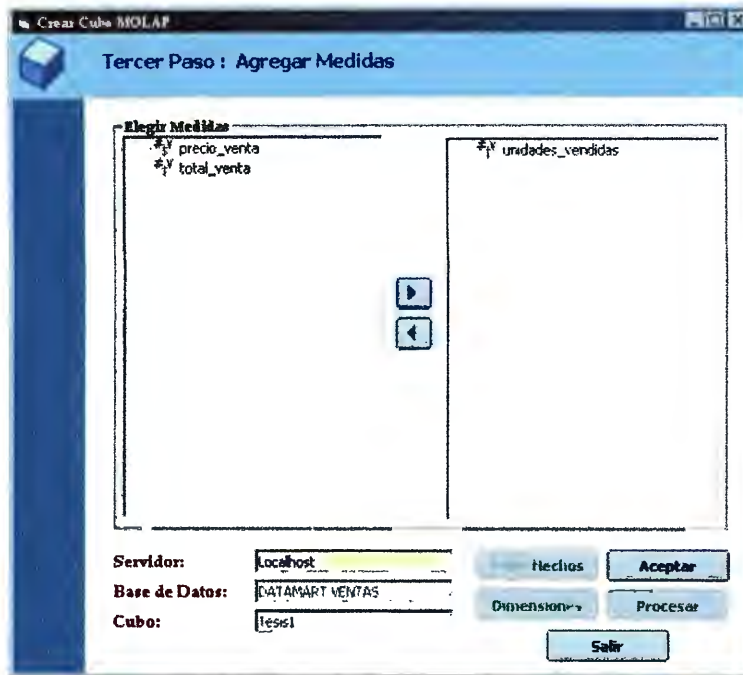
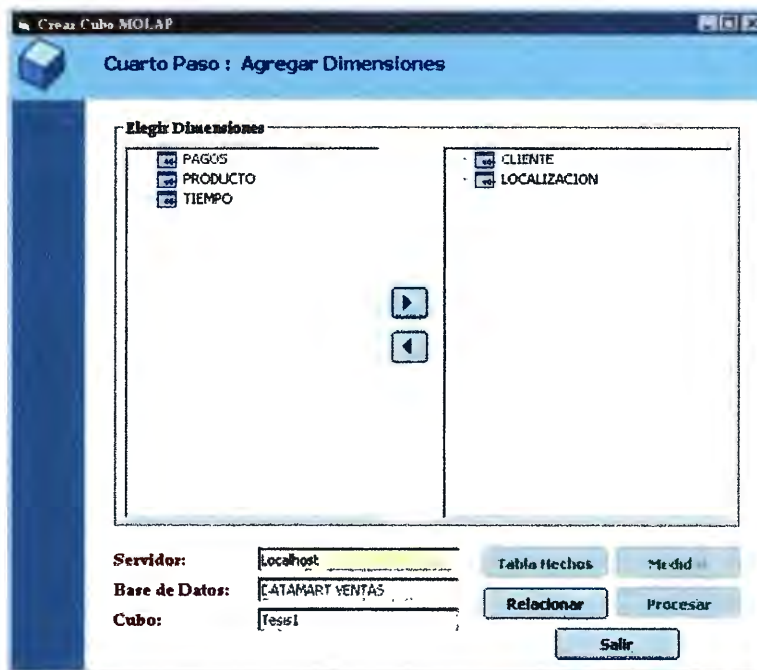


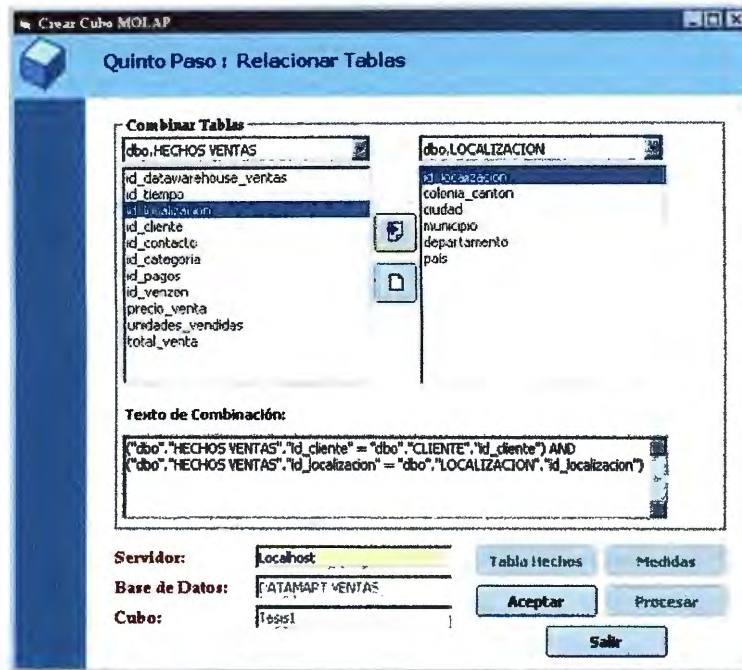
Fig. 3.47 Segundo Paso: Identificación de Tabla Hechos



**Fig. 3.48 Tercer Paso: Agregar Medidas**



**Fig. 3.49 Cuarto Paso: Agregar Dimensiones**

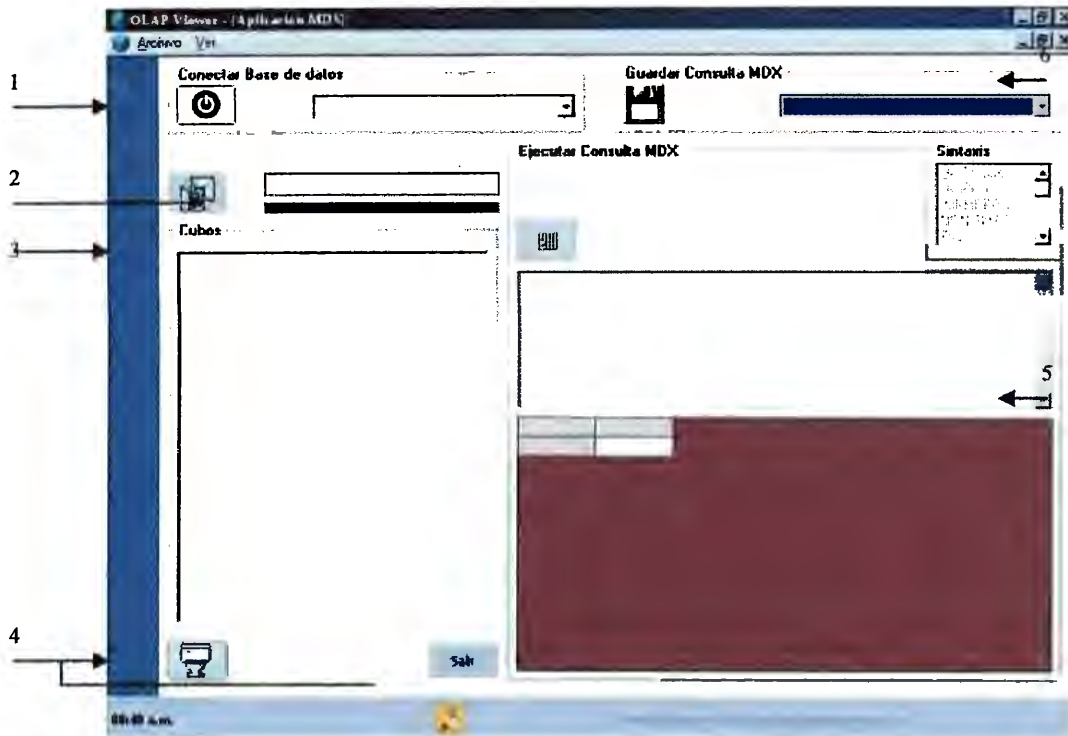


**Fig. 3.50 Quinto Paso: Relacionar Tablas**

### 3.9.4 Formulario Consulta MDX.

El formulario (Fig. 3.51) aparece a 840 tw del borde izquierdo de la pantalla principal de la aplicación, es de fondo blanco, tiene dimensiones de:

4575 de alto y 3855 de ancho. Consta de un menú en la parte superior con las opciones de Archivo y Ver, los cuales tienen otro submenú abrir consulta MDX, guardar consulta MDX y salir para la opción archivo, y reporte MDX para la opción ver. Los objetos dentro del formulario los detallamos a continuación, tomando como referencia la siguiente figura:




**Fig. 3.51 Formulario Consulta MDX**

1- Conector Base de Datos (Frame1): fondo de color amarillo ubicado a 720 tw del borde izquierdo interno del formulario y a 120 del borde superior interno. Con dimensiones de 855x5175<sup>16</sup> tws.

En su interior posee un botón con dimensiones de 495x375 y una lista despegable de dimensiones 315x3255.

2- Botón, caja de texto y etiqueta:

El botón , esta ubicado a 720 tw del borde izquierdo interno del formulario y a 1440 del borde superior interno. Con dimensiones de 495x735 tws. Estilo Windows XP.

La caja de texto tiene dimensiones de 285x2895, tiene un fondo de color amarillo y esta ubicado a 1920 tw del borde interior izquierdo del formulario, a la misma altura del botón.

Para la estética del formulario se a colocado una etiqueta de color azul con dimensiones 135x2895, ubicada a 1800 tw del borde superior interno del formulario.

<sup>16</sup> Cuando se utilice este formato de aquí en adelante para las dimensiones se leerá: alto x ancho

3- Cubos (Frame2), posee un fondo de color amarillo, con borde simple, dimensiones de 5055x4095, ubicado a 2040 tw del borde superior interno del formulario, la distancia hacia el borde interno izquierdo es la misma que el del frame1, contiene un objeto TreeView, el cual tiene dimensiones de 4575x3855, con un estilo de jerarquía de objetos cada uno con etiqueta e icono.

4- Botones Limpiar y Salir:




Limpiar Datos, con dimensiones de 495x375, ubicado a 7200 tw del borde superior y 720 del borde izquierdo internos del formulario, estilo Windows XP.



Botón Salir, sus dimensiones son 435x855, ubicado a 3960 tw del borde interno izquierdo del formulario.

5- Ejecutar consulta MDX (Frame4), tiene un color gris de fondo, su ubicación es 4920 tw del borde interno izquierdo del formulario y 1080 del borde superior, el título tiene fuente MS Sans Serif de tamaño 8 y estilo negrita, las dimensiones son 6615x6855. En su interior tiene un botón, una caja de texto y un grid flexible (MSFlexGrid), los cuales están ubicados a 5040 tw del borde interior izquierdo del formulario, las dimensiones de cada objeto son:

-  botón que ejecuta la consulta: 495x735, es tipo Windows XP.
- Para la caja de texto: 1695x6615. La fuente del tipo MS Sans Serif de tamaño 8 y estilo regular.
- Para el objeto MSFlexGrid: 3135x6615. Fuente del tipo MS Sans Serif de tamaño 8 y estilo negrita.

Dentro del Frame “Ejecutar consulta MDX”, encontramos el otro frame que lleva por título “Sintaxis”, el cual tiene fondo de color amarillo, esta ubicado en el extremo superior derecho del frame4 a 5160 del borde izquierdo interno de éste, con dimensiones de 1455x1695.

Dentro del frame “Sintaxis” tenemos un objeto de tipo lista (List1), con dimensiones 1035x1455 y fuente del tipo MS Sans Serif de tamaño 8 y estilo regular.

6- “Guardar Consulta MDX” (Frame3), fondo de color amarillo ubicado a 6240 tw del borde izquierdo interno del formulario y a 120 del borde superior interno. Con dimensiones de 855x5535 tws.

En su interior posee un botón con dimensiones de 495x495 y una lista desplegable de dimensiones 315x3375.

### 3.9.5 Formulario Consulta SQL.

El formulario es de fondo blanco, tiene dimensiones de: 7260 de alto y 9300 de ancho. A la izquierda se dibuja en tiempo de ejecución una etiqueta de color azul de 6855x855. La etiqueta para darle el título al formulario tiene dimensiones de 345x1980, ubicada a 1080 del borde interno izquierdo del formulario, con una fuente tipo Book Antiqua, de tamaño 14 y estilo negrita. Los objetos dentro del formulario los detallamos a continuación, la ubicación de estos al borde izquierdo interno es la misma para todos y esta a 1080 tw, vemos la siguiente figura (Fig. 3.52) para detallarlo:

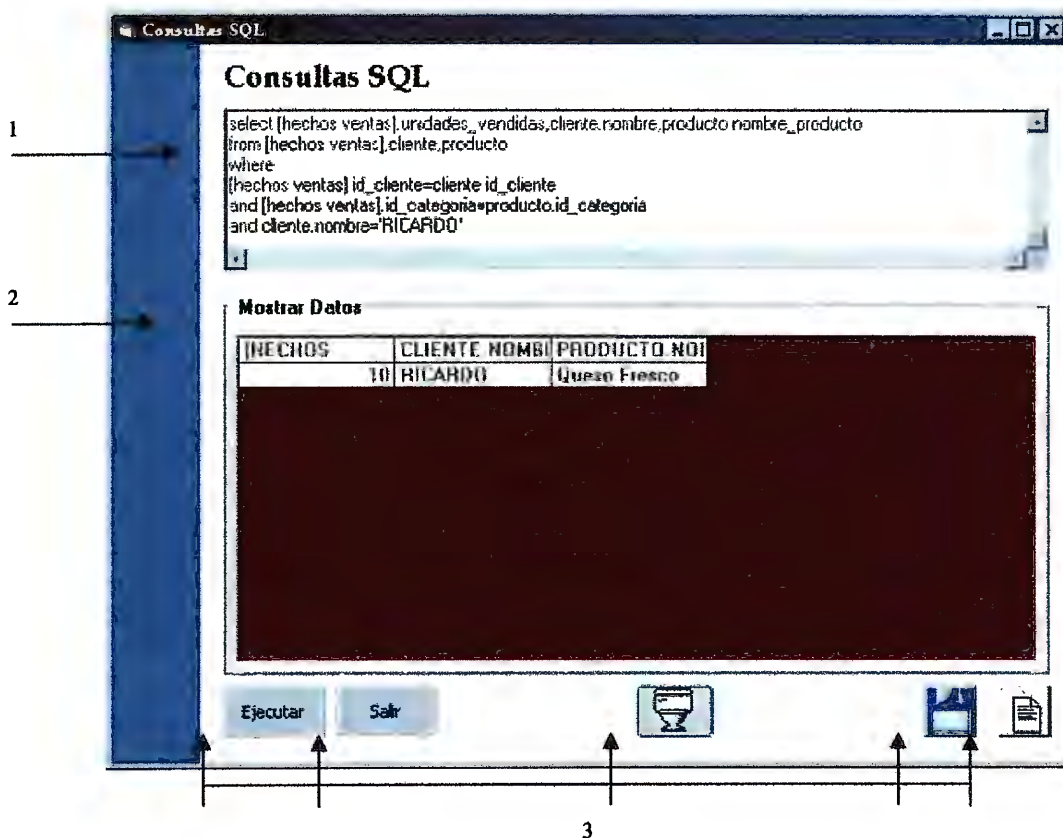


Fig.3.52 Formulario Consulta SQL


**1- Caja de texto**

Dimensiones: 1575x7935, con letra del tipo MS Sans Serif de un tamaño de 8 y del tipo regular.

**2- Frame (Mostrar Datos)**

Dimensiones: 3615x7935, ubicado a 2400 tw. De borde superior del formulario. Contiene en su interior un cuadrícula del tipo MSFlexGrid, con dimensiones de 3135x7695, los datos dentro de ella aparecen con el tipo MS Sans Serif de un tamaño de 8 y del tipo negrita.

3- Botones, todos se encuentran a 6120tw del borde superior interno del formulario, debajo de la cuadrícula donde son mostrados los datos de la consulta.

- Ejecutar y Salir, tienen dimensiones de 495x975, están ubicados a 1080 y 2160 tw del borde interior izquierdo del formulario.
- Botón Limpiar  , tiene dimensiones 495x735 y esta a 5040 del borde izquierdo interno del formulario, tiene una apariencia de un botón de Windows XP.
- Botones de Guardar y Abrir una consulta, son los 2 botones que están en la esquina inferior derecha del formulario, tienen dimensiones de 495x495, y están ubicados a 7800 y 8520 del borde izquierdo interior del formulario.

### 3.9.6 Diseño del Administrador de Cubos.

El formulario es de fondo blanco, con título “Administrador de Cubos”, tiene dimensiones de:

8595 de alto y 12000 de ancho. A la izquierda se dibuja en tiempo de ejecución una etiqueta de color azul de 7815x615, la cual está a 240 tw del borde superior interno del formulario.

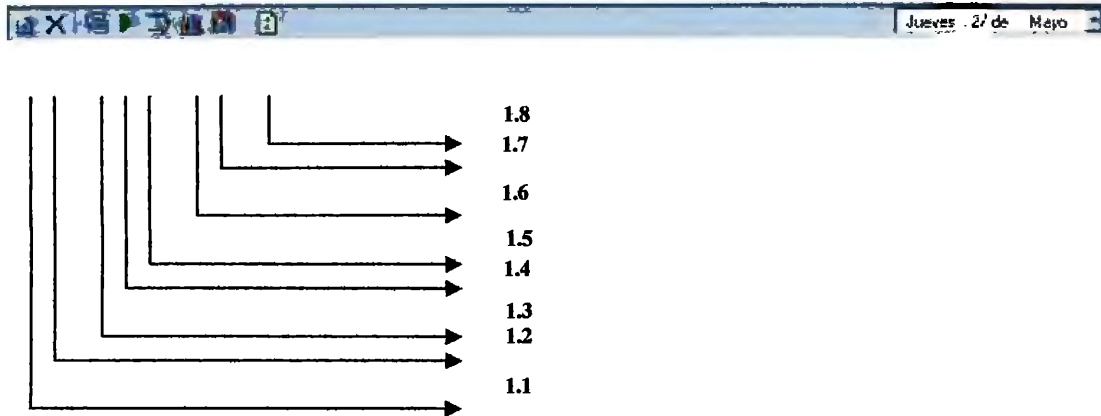
Tiene una barra de herramientas (1) en la parte superior la cual tiene dimensiones de 330x11880, esta barra contiene los botones para:

(1.1) un nuevo cubo, (1.2) eliminar un cubo, (1.3) agregar dimensión, (1.4) procesar un cubo, (1.5) el del explorador dinámico de datos, (1.6) gráfico dinámico de datos, (1.7) explorador de minería de datos y (1.8) el de refrescar, el diseño de la barra de herramientas se presenta en la figura 3.53.

En la parte derecha de la barra de herramientas encontramos un calendario el cual indica la fecha actual, posicionado a 9600 tw del borde izquierdo interno del formulario, el formato que presenta la fecha es el de día/mes/año, como por ejemplo: Jueves, 27 de Junio, presenta la fecha del sistema, la fuente de la letra es MS Sans Serif, estilo regular y de tamaño 8, este calendario es despegable donde se puede modificar la fecha que se desplegará. Las dimensiones del objeto son:

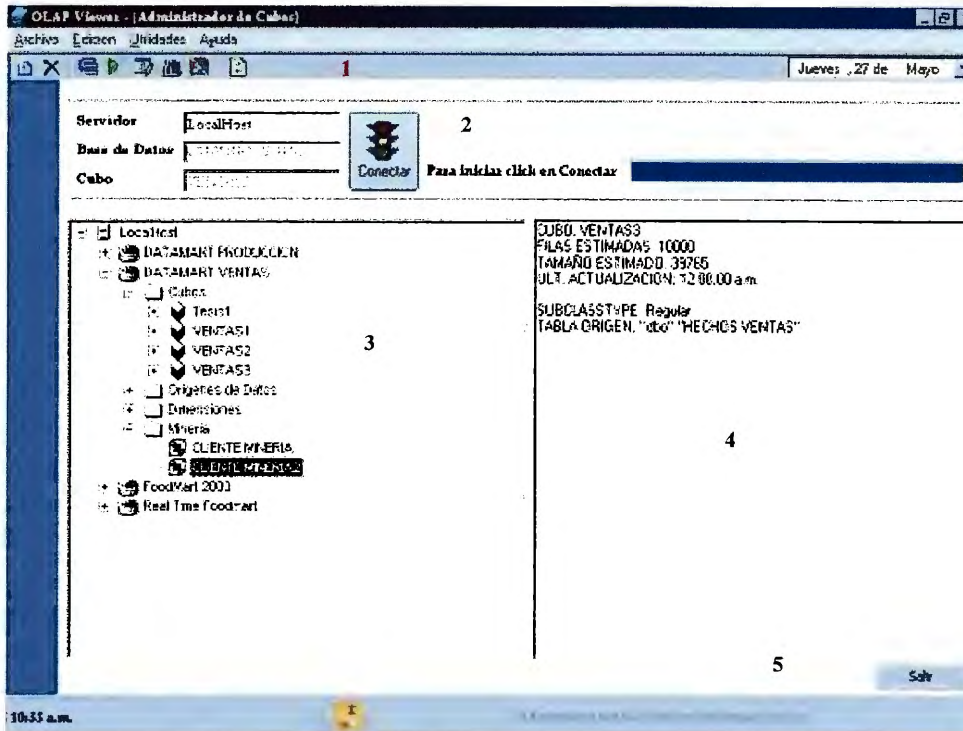
Alto: 300

Ancho 2295



**Fig. 3.53** Diseño de la barra de herramientas

Los objetos dentro del formulario los detallamos en la figura 3.54, la ubicación de estos al borde izquierdo interno es la misma para los objetos 2 y 3 y esta a 720 tw, vemos la siguiente figura para detallarlo:



**Fig. 3.54** Administrador de Cubos

2- Frame que contiene 3 cajas de texto, 5 etiquetas y 1 botón, sus dimensiones son de 1335x11175 y está a 480 tw del borde superior del formulario, los detalles de los objetos dentro de él son:

- Cajas de texto, las dimensiones de éstas son de 285x1575, ubicadas a 1440 tw del borde izquierdo interno, con una distancia de 360 tw entre ellas.
- Etiquetas, la distancia de las etiquetas con los textos: Servidor, Base de Datos, Cubos, es de 120 tw con respecto al borde izquierdo interno del frame, tienen una altura de 210 tw, el ancho de ellas varía con respecto al ancho de los textos, y las separarán 360 tw una de otra, el texto es de tipo Bookman Old Style, de estilo negrita y un tamaño de 8, color negro.

La etiqueta con el texto “Para iniciar click en Conectar” tiene dimensiones de 240x1340, ubicado a 4440 del borde izquierdo y a 840 tw del borde superior interno del frame, la fuente es Book Antiqua, de estilo negrita y un tamaño de 8, color negro.

La otra etiqueta es para la estética del formulario tiene dimensiones de 255x4095, ubicado a 6960 del borde izquierdo y a 840 tw del borde superior interno del frame, y de color azul.

- Botón con dimensiones de 975x855, ubicado a 3240 del borde izquierdo y a 240 tw del borde superior interno del frame, con un estilo de Windows XP.

3- Objeto TreeView, tiene una apariencia de 3D como todos los objetos, tiene una letra de tipo arial, de tamaño 8 y de estilo regular, tiene dimensiones de 5415x5685, ubicada a 2040 tw del borde superior interno del formulario, el árbol tiene el estilo de etiqueta con mapa de bits a la par, como se observa en la figura.

4- Etiqueta, las letras que aparecen en ella son de fuente MS Sans Serif, de estilo regular y tamaño 8, con dimensiones de 5415x5415, ubicada a 6480 del borde izquierdo y a 2040 tw del borde superior interno del formulario.

- 5- Botón Salir, con dimensiones de 330x1215, ubicado en la parte inferior derecha de la pantalla a 10560 del borde izquierdo y a 7560 tw del borde superior interno del formulario.

### 3.9.7 Formularios de mantenimiento de Datamart Ventas y Datamart Producción.

#### 3.9.7.1 Formulario Mantenimiento Clientes.

The screenshot shows a window titled 'CLIENTE' with a sub-header 'Mantenimiento Clientes'. The form is set against a dotted grid background. It includes the following fields and controls:

- id\_cliente: text input
- tipo\_cliente: text input
- telefono: text input
- fax: text input
- nombre: text input
- compra\_mensual: text input
- numero\_sucursales: text input
- clase: text input
- tamaño: text input
- compra\_mensual\_queso: text input
- compra\_mensual\_cremas: text input
- tipo\_clase: text input
- Combo1, Combo2, Combo3, Combo4, Combo5, Combo6, Combo7, Combo8, Combo9: dropdown menus
- Buttons: 'Buscar', 'Agregar', 'Modificar', 'Eliminar', 'Cancelar', 'Salir'

**Fig. 3.55 Formulario Mantenimiento Clientes**

#### Dimensiones

Alto: 7125

Ancho: 8910

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario aparece el texto CLIENTE.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 6735 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento de Clientes"), tiene un ancho de 3210, un alto de 345, esta a 3240 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y listas despegables para datos de clientes:*

Todas las etiquetas del formulario para los datos del cliente tienen:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Todas las cajas de texto del formulario para los datos del cliente tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Las listas despegables (ComboBox) tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Con excepción de la lista despegable del id\_cliente, la cual es estilo de la fuente de texto es negrita, las demás características de la fuente son iguales a las listas despegables anteriores.

Las dimensiones de la etiquetas, de las cajas de texto y las listas despegables son diferentes, así como, las ubicación de cada una en el formulario y se detallan a continuación.

Para el id\_cliente:

Etiqueta:

Dimensiones: alto = 240, ancho = 840, se encuentra a una distancia de 840 tw del borde superior y a 1680 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

**Caja de texto:**

Dimensiones: alto = 285, ancho = 975, se encuentra a una distancia de 840 tw del borde superior y a 3840 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

**Lista despegable:**

Dimensiones: alto = 315, ancho = 1335, se encuentra a una distancia de 810 tw del borde superior y a 6480 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Para tipo\_cliente:

**Etiqueta:**

Dimensiones: alto = 240, ancho = 990, se encuentra a una distancia de 1155 tw del borde superior y a 1680 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

**Caja de texto:**

Dimensiones: alto = 285, ancho = 2295, se encuentra a una distancia de 1155 tw del borde superior y a 3840 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

**Lista despegable:**

Dimensiones: alto = 315, ancho = 1815, se encuentra a una distancia de 1140 tw del borde superior y a 6480 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Para teléfono:

**Etiqueta:**

Dimensiones: alto = 240, ancho = 690, se encuentra a una distancia de 1485 tw del borde superior y a 1680 del borde izquierdo interno del formulario, una apariencia de

tercera dimensión.

Caja de texto:

Dimensiones: alto = 285, ancho = 1575, se encuentra a una distancia de 1485 tw del borde superior y a 3840 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Para fax:

Etiqueta:

Dimensiones: alto = 240, ancho = 285, se encuentra a una distancia de 1800 tw del borde superior y a 1680 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Caja de texto:

Dimensiones: alto = 285, ancho = 1575, se encuentra a una distancia de 1800 tw del borde superior y a 3840 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Para nombre:

Etiqueta:

Dimensiones:

Alto = 240, ancho = 645, se encuentra a una distancia de 2115 tw del borde superior y a 1680 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Caja de texto:

Dimensiones: alto = 285, ancho = 2295, se encuentra a una distancia de 2115 tw del borde superior y a 3840 del borde izquierdo interno del formulario, con una longitud máxima para 75 caracteres, una apariencia de tercera dimensión.

Para compra\_mensual:

Etiqueta:

Dimensiones: alto = 240, ancho = 1395, se encuentra a una distancia de 2445 tw del borde superior y a 1680 del borde izquierdo interno del formulario, con una apariencia en tercera dimensión.

Caja de texto:

Dimensiones: alto = 285, ancho = 1575, se encuentra a una distancia de 2445 tw del borde superior y a 3840 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Lista despegable:

Dimensiones: alto = 315, ancho = 1335, se encuentra a una distancia de 2410 tw del borde superior y a 6480 del borde izquierdo interno del formulario, una apariencia de tercera dimensión.

Para numero\_sucursales, clase, tamaño, compra\_mensual\_queso, compra\_mensual\_crema, tipo\_clase, los valores de las dimensiones son:

Etiqueta:

Alto: 240

Ancho: varia dependiendo de la longitud de los caracteres en la etiqueta, por ejemplo, para numero\_sucursales = 1575 y para clase = 435

Distancia al borde izquierdo: 1680

Distancia al borde superior: varia de acuerdo a la posición de cada etiqueta, el espacio entre ellas es de 315.

Caja de texto:

Alto: 285

Ancho: varia dependiendo de la longitud de los caracteres en la caja de texto.

Distancia al borde izquierdo: 3840

Distancia al borde superior: varia de acuerdo a la posición de cada caja de texto, el espacio entre ellas es de 315.

**Listas despegables:**

Alto: 315

Ancho: depende de la longitud de los caracteres que se desplegaran.

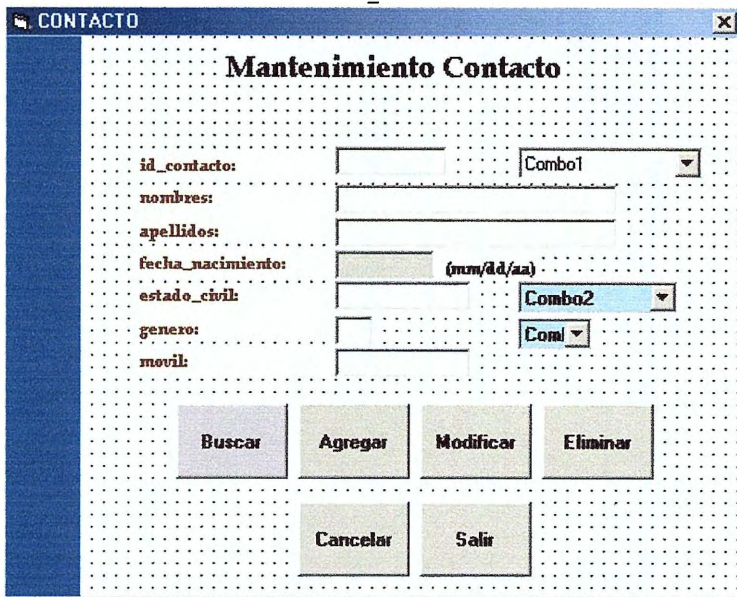
Distancia al borde izquierdo: 6480

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 4920 del borde superior interno del formulario y empezando a 2400 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 5880 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

**3.9.7.2 Formulario Mantenimiento Contacto.**

**Fig. 3.56 Formulario Mantenimiento Contacto**

**Dimensiones**

Alto: 5865

Ancho: 7305

Se muestra (Fig. 3.55) a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario aparece el texto CONTACTO.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 5535 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Contacto"), tiene un ancho de 3300, un alto de 345, esta a 2160 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado al centro, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y listas despegables para mantenimiento de contactos:*

Etiquetas del formulario para los datos del contacto:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: 1935

Distancia al borde izquierdo interno del formulario: 1320

Distancia al borde superior: varia de acuerdo a la posición de cada etiqueta, el espacio entre ellas es de 315.

Todas las cajas de texto del formulario para los datos del contacto tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: varía dependiendo de la longitud de los caracteres en la caja de texto.

Distancia al borde izquierdo: 3240

Distancia al borde superior: al mismo alto de cada etiqueta.

Las listas despegables (ComboBox) tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Ubicadas a 5040 tw del borde izquierdo interno del formulario.

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 3600 del borde superior interno del formulario y empezando a 1680 del borde izquierdo interno.

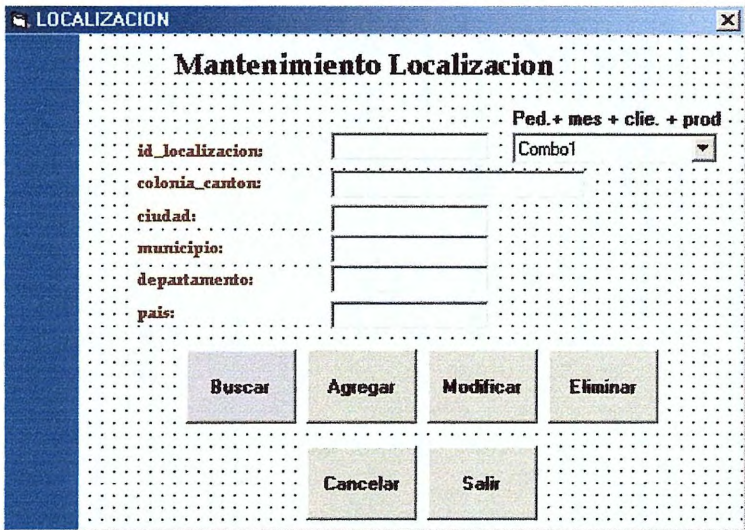
Los botones de Cancelar y Salir se encuentran a 4560 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.3 Formulario Mantenimiento Localización.



The screenshot shows a window titled "LOCALIZACION" with a sub-header "Mantenimiento Localización". The form contains several input fields and a dropdown menu. The fields are labeled: "id\_localizacion:", "colonia\_canton:", "ciudad:", "municipio:", "departamento:", and "pais:". To the right of the "id\_localizacion:" field is a dropdown menu with the text "Ped.+ mes + clie. + prod" and "Combo1" selected. Below the form are six buttons arranged in two rows: "Buscar", "Agregar", "Modificar", "Eliminar" in the top row, and "Cancelar", "Salir" in the bottom row.

Fig. 3.57 Formulario Mantenimiento Localización

### Dimensiones

Alto: 5280

Ancho: 7425

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario, que vemos en la figura 3.56, aparece el texto LOCALIZACION.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 4935 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Localizacion"), tiene un ancho de 3780, un alto de 345, esta a 1680 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y listas despegables para mantenimiento de localizacion:*

#### Etiquetas:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: 1815

Distancia al borde izquierdo interno del formulario: 1320

Distancia al borde superior: varia de acuerdo a la posición de cada etiqueta, el espacio entre ellas es de 315.

**Cajas de texto:**

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: 1575, con excepción del utilizado para la colonia o cantón, el cual es de 2535.

Distancia al borde izquierdo: 3240

Las listas despegables (ComboBox) tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Ubicadas a 5040 tw del borde izquierdo interno del formulario y un alto de 315 cada una

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 3720 del borde superior interno del formulario y empezando a 2640 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 4680 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.4 Formulario Mantenimiento Tiempo.

**Fig. 3.58 Formulario Mantenimiento Tiempo**

#### Dimensiones

Alto: 6000

Ancho: 9135

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario (ver figura 3.57) aparece el texto TIEMPO.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 5655 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Tiempo"), tiene un ancho de 3165, un alto de 345, esta a 3360 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y listas despegables para mantenimiento de Tiempo:***Etiquetas:**

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 240

Ancho: varía dependiendo del texto en la etiqueta.

Distancia al borde izquierdo interno del formulario: 1320

Distancia al borde superior: varía de acuerdo a la posición de cada etiqueta, el espacio entre ellas es de 315.

Todas las cajas de texto del formulario para los datos de tiempo tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: de acuerdo al número de caracteres que se pueden digitar en una caja de texto se ha dado un ancho común a cada caja de texto. Por ejemplo mes, día, hora soportan un máximo de 10 caracteres y el valor en twips es de 1215.

Distancia al borde izquierdo: 2520

Las listas despegables (ComboBox) tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Ubicadas a 5040 tw del borde izquierdo interno del formulario y a 1320 tw del borde superior la primera. Con un alto de 315 cada una.

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 3120 del borde superior interno del formulario y empezando a 1800 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 4080 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.5 Formulario Mantenimiento Datos Personales.

**Fig. 3.59 Formulario Mantenimiento Datos Personales**

Dimensiones

Alto: 6450

Ancho: 7890

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario (figura 3.58) aparece el texto DATOS PERSONALES.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 6135 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Datos Personales"), tiene un ancho de 4395, un alto de 345, esta a 1560 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado al centro, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y listas despegables:*

Etiquetas:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: 1815

Distancia al borde izquierdo interno del formulario: 1320

Distancia al borde superior: varía de acuerdo a la posición de cada etiqueta.

Cajas de texto:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: Varía de acuerdo al número de caracteres que se pueden digitar en ellas.

Distancia al borde izquierdo: 3360

Las listas despegables (ComboBox) tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Ubicadas a 5640 tw del borde izquierdo interno del formulario.

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 4320 del borde superior interno del formulario y empezando a 1440 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 5280 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.6 Formulario Mantenimiento Zona.



Fig. 3.60 Formulario Mantenimiento Zona

Dimensiones

Alto: 3885

Ancho: 6825

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario (figura 3.59) aparece el texto ZONA .

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 3495 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Zonas"), tiene un ancho de 2940, un alto de 345, esta a 2280 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

### *Etiquetas y cajas de texto:*

#### Etiquetas:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: 1815

Distancia al borde izquierdo interno del formulario: 1080

Distancia al borde superior: distancia de 315 tw entre cada etiqueta.

#### Cajas de texto:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: 615 para id\_zona y 3375 para descripción.

Distancia al borde izquierdo: 3000

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 1680 del borde superior interno del formulario y empezando a 1320 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 2640 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.7 Formulario Mantenimiento Vendedor.

**Fig. 3.61 Formulario Mantenimiento Vendedor**

**Dimensiones**

Alto: 4830

Ancho: 6375

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario aparece el texto VENDEDOR .

Tiene una etiqueta al lado izquierdo del formulario (ver la figura 3.60) de una combinación de colores que da un tono azul claro, con un alto de 4455 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Vendedor"), tiene un ancho de 3465, un alto de 345, esta a 1680 a la izquierda y 120 tw debajo de los bordes del

formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y lista despegable:*

Etiquetas:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: varia dependiendo del texto en la etiqueta

Distancia al borde izquierdo interno del formulario: 1800

Distancia al borde superior: distancia de 315 tw entre cada etiqueta.

Cajas de texto:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 1800

Ancho: 855

Distancia al borde izquierdo: 3360

La lista despegable (ComboBox) tiene:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Ubicadas a 4560 tw del borde izquierdo interno del formulario.

Alto: 315

Ancho: 1095

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 2400 del borde superior interno del formulario y empezando a 960 del borde izquierdo interno.

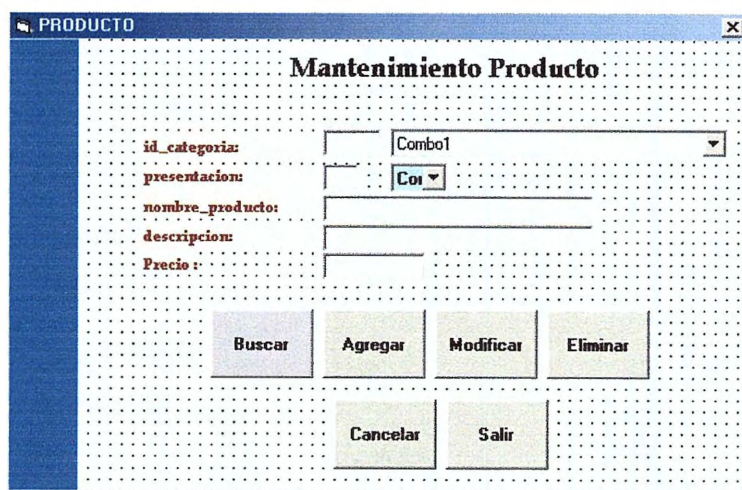
Los botones de Cancelar y Salir se encuentran a 3360 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.8 Formulario Mantenimiento Producto.



**Fig. 3.62 Formulario Mantenimiento Producto**

Dimensiones

Alto: 5175

Ancho: 7980

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario (Fig. 3.61) aparece el texto PRODUCTO.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 4815 y con un ancho de 735. Se dibuja en tiempo

de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Producto"), tiene un ancho de 3315, un alto de 345, esta a 3000 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antigua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y listas despegables:*

Etiquetas:

Fuente del texto: es del tipo Book Antigua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: 1815

Distancia al borde izquierdo interno del formulario: 1440

Distancia al borde superior: distancia de 315 tw entre cada etiqueta.

Cajas de texto:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: variante

Distancia al borde izquierdo: 3360

La lista despegable (ComboBox) tiene:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Ubicadas a 4080 tw del borde izquierdo interno del formulario.

Alto: 315

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 2880 del borde superior interno del formulario y empezando a 2160 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 3840 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.9 Formulario Mantenimiento Pagos.

**Fig. 3.63 Formulario Mantenimiento Pagos**

Dimensiones

Alto: 4455

Ancho: 7410

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario (Fig. 3.62) aparece el texto PAGOS.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 4095 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Pagos"), tiene un ancho de 2910, un alto de 345, esta a 2640 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

*Etiquetas, cajas de texto y listas despegables:*

Etiquetas:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: 1815

Distancia al borde izquierdo interno del formulario: 1200

Distancia al borde superior: distancia de 315 tw entre cada etiqueta.

Cajas de texto:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: variante

Distancia al borde izquierdo: 3480

La lista despegable (ComboBox) tiene:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Ubicadas a 5640 tw del borde izquierdo interno del formulario.

Alto: 315

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 2280 del borde superior interno del formulario y empezando a 1800 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 3240 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.10 Formulario Data Warehouse Ventas.

The screenshot shows a window titled "DATAWAREHOUSE\_VENTAS" with a close button in the top right corner. The main content area is titled "Mantenimiento Data Warehouse Ventas" and features a grid of input fields and buttons. The input fields are labeled as follows:

- id\_datawarehouse\_ventas:
- periodo\_venta:
- nombre\_producto:
- unidades\_vendidas\_mensu:
- precio\_producto:
- total\_venta\_mensual:

Below the input fields, there are six buttons arranged in two rows:

- Row 1: Buscar, Agregar, Modificar, Eliminar
- Row 2: Cancelar, Salir

Fig. 3.64 Formulario Data Warehouse Ventas

### Dimensiones

Alto: 5025

Ancho: 7005

El formulario (figura 3.63), se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario aparece el texto DATAWAREHOUSE\_VENTAS.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 4625 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Data Warehouse Ventas"), tiene un ancho de 5310, un alto de 345, esta a 1080 a la izquierda y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

### *Etiquetas y cajas de texto:*

#### Etiquetas:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 255

Ancho: variante dependiendo del texto que contiene la etiqueta.

Distancia al borde izquierdo interno del formulario: 1440

Distancia al borde superior: distancia de 315 tw entre cada etiqueta.

#### Cajas de texto:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: variante

Distancia al borde izquierdo: 3480

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 2880 del borde superior interno del formulario y empezando a 1440 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 3840 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

### 3.9.7.11 Formulario Mantenimiento Hechos\_Ventas.

The screenshot shows a software window titled "HECHOS VENTAS" with a sub-header "Mantenimiento Hechos\_Ventas". The form is set against a dotted grid background. On the left, there is a vertical blue bar. The form contains the following elements:

- Input fields for: `id_cliente`, `id_contacto`, `id_localizacion`, `id_tiempo`, `id_datawarehouse_ventas`, `id_categoria`, `id_pagos`, `id_vendedor`, `precio_venta`, `unidades_vendidas`, and `total_venta`.
- Three dropdown menus labeled "Combo1", "Combo2", and "Combo3". Above them is the text "(MES+AÑO+PRO)".
- A row of four buttons: "Buscar", "Agregar", "Modificar", and "Eliminar".
- A row of two buttons: "Cancelar" and "Salir".

Fig. 3.65 Formulario Mantenimiento Hechos\_Ventas

Dimensiones del formulario (Fig. 3.64)

Alto: 7635

Ancho: 9405

Se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación.

Barra de título del formulario aparece el texto HECHOS VENTAS.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un alto de 7455 y con un ancho de 735. Se dibuja en tiempo de ejecución con estilo 3D.

La etiqueta del título del formulario ("Mantenimiento Hechos\_Ventas"), tiene un ancho de 4170, un alto de 345, esta a 3120 a la derecha y 120 tw debajo de los bordes del formulario, se dibuja en tiempo de ejecución con estilo 3D.

El texto de la etiqueta está:

Justificado a la izquierda, estilo 3D, con un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

Se ha utilizado una línea horizontal negra a la par del id del tiempo y otra antes de los botones para darle el mantenimiento al formulario.

*Etiquetas, cajas de texto y listas despegables:*

Etiquetas:

Fuente del texto: es del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda.

Dimensiones:

Alto: 240

Ancho: variante dependiendo del texto que contiene la etiqueta.

Distancia al borde izquierdo interno del formulario: 1200

Cajas de texto:

Fuente del texto: es del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro.

Dimensiones:

Alto: 285

Ancho: variante

Distancia al borde izquierdo: 3600

Las listas despegables (ComboBox) tienen:

Fuente del texto: es del tipo MS Sans Serif, con estilo negrita y un tamaño de 8; de color negro.

Se utilizan 3 listas despegables, en las que el ancho y la ubicación con respecto al borde interior izquierdo y superior son diferentes, lo único que tienen en común es la altura, la cual es de 315

Los botones se han ubicado al centro del formulario, posicionando los 4 botones superiores a 5400 del borde superior interno del formulario y empezando a 2760 del borde izquierdo interno.

Los botones de Cancelar y Salir se encuentran a 6360 del borde superior interno del formulario.

Las dimensiones de los seis botones son de:

Alto = 735 y Ancho = 1095.

Las dimensiones de las imágenes de los botones son las mismas que la de los botones.

En general se puede decir que el diseño utilizado para los formularios del Datamart Ventas es el siguiente:

Un formulario de color blanco de fondo, que se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación, variando las dimensiones de cada formulario, en el cual, en la barra de título aparece texto, dándole un nombre.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un ancho de 735 y un alto variante para cada uno, el cual se dibuja en tiempo de ejecución con estilo 3D.

Una etiqueta para el título del formulario en la parte superior y centrado a 120 tw por debajo de los bordes del formulario. El texto de esta etiqueta tiene un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

Bajo la etiqueta que le da el título al formulario aparecen las etiquetas (al lado izquierdo del formulario) con la fuente del texto del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda, cajas de texto (al centro) con fuente del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro y listas despegables (al lado derecho, si las hay), con las mismas características que las cajas de texto para la fuente. Las dimensiones de estos tres objetos varían de formulario a formulario.

Los botones (Buscar, Agregar, Modificar, Eliminar, Cancelar, Salir) para darle el mantenimiento al formulario aparecen centrados y al final de él, con la ubicación con respecto a los bordes internos superiores e izquierdo dependiendo del tamaño del formulario.

El diseño que se aplica al Datamart Producción es el mismo que se ha explicado para el Datamart Ventas. Lo podemos apreciar en la siguiente figura 3.65, y la explicación es la siguiente:

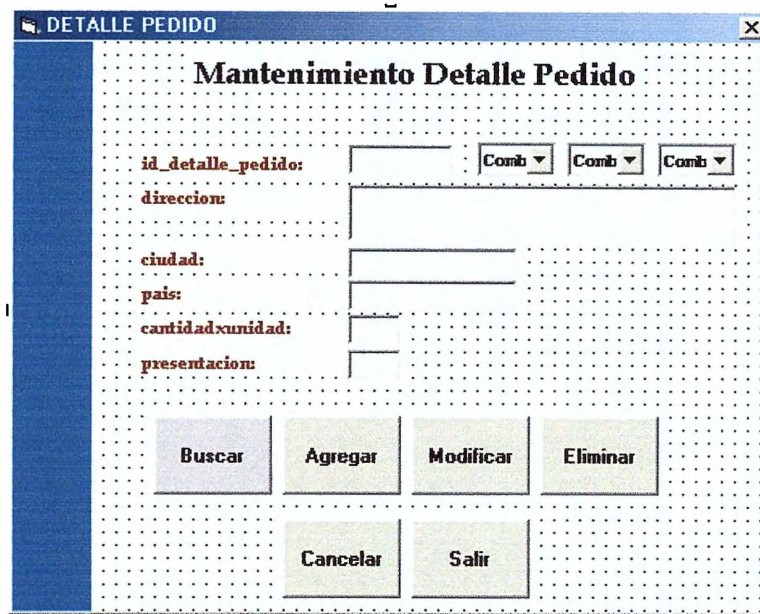
Un formulario de color blanco de fondo, que se muestra a una distancia de 1050 a la izquierda del formulario principal de la aplicación, variando las dimensiones de cada formulario, en el cual, en la barra de título aparece texto, dándole un nombre.

Tiene una etiqueta al lado izquierdo del formulario de una combinación de colores que da un tono azul claro, con un ancho de 735 y un alto variante para cada uno, el cual se dibuja en tiempo de ejecución con estilo 3D.

Una etiqueta para el título del formulario en la parte superior y centrado a 120 tw por debajo de los bordes del formulario. El texto de esta etiqueta tiene un fondo de color blanco, con fuente de tipo Book Antiqua, color negro, estilo negrita, tamaño 14.

Bajo la etiqueta que le da el título al formulario aparecen las etiquetas (al lado izquierdo del formulario) con la fuente del texto del tipo Book Antiqua, con estilo negrita y un tamaño de 8; de color rojo magenta con valor hexadecimal de H00000080, alineado a la izquierda, cajas de texto (al centro) con fuente del tipo MS Sans Serif, con estilo regular y un tamaño de 8; de color negro y listas despegables (al lado derecho, si las hay), con las mismas características que las cajas de texto para la fuente. Las dimensiones de estos tres objetos varían de formulario a formulario.

Los botones (Buscar, Agregar, Modificar, Eliminar, Cancelar, Salir) para darle el mantenimiento al formulario aparecen centrados y al final de él, con la ubicación con respecto a los bordes internos superiores e izquierdo dependiendo del tamaño del formulario.



**Fig. 3.66 Formulario Mantenimiento Detalle Pedido**

El diseño de los botones es el mismo en todos los formularios de la opción utilidades del menú principal, es decir, para los Datamart Ventas y Datamart Producción, las imágenes para los formularios se presentan en la figura 3.66.



**Fig. 3.67 Botones de mantenimiento en los formularios**

Los componentes utilizados para las imágenes de los botones, el ChameleonButton y el SexyControls, así como los otros componentes utilizados para los formularios y la aplicación completa, son explicados en la parte de anexos, en la sección A.8.1.

Las referencias o librerías utilizadas en la aplicación están en la parte A.8.2, de la sección de anexos A.

## CONCLUSIONES

Los Servidores OLAP permiten optimizar la información en las empresas, por lo tanto se puede desarrollar un sistema que muestre la aplicación de tecnologías multidimensionales y que tenga un impacto significativo en la economía de la zona.

Los esquemas multidimensionales facilitan tareas de búsqueda de información, permitiendo así la reducción de costos y tiempo, logrando resultados satisfactorios en los análisis de variables de marketing.

Evaluando tendencias de patrones ocultos en los datos podremos pronosticar situaciones que serán validas para cualquier variable analizada en el marketing, constituyéndose como información de utilidad para la toma de decisiones.

Proveer una herramienta que permita el análisis de servidores OLAP con interfaces graficas, permitiendo un entorno de trabajo mas amigable para el usuario.

Cuando se utiliza OLAP, la información relacional del Data Warehouse es transformada en estructuras multidimensionales, dando una mayor velocidad de consulta.

MDX esta diseñado para facilitar consultas en cubos multidimensionales, a diferencia que el lenguaje SQL que esta diseñado para realizar consultas en tablas de datos relacionales.

Es importante poseer las bases teóricas para una mejor comprensión de la estructura, funcionamiento y construcción de aplicaciones de este tipo de herramientas.

## BIBLIOGRAFIA

### Libros

Kendal & Kendal , *Análisis y Diseño de Sistemas*,5ta. Edición, Prentice Hall, 2002.

### Revistas

Fundación CORDES, *Oficina regional del bajo lempa de Tecoluca y Zacatecoluca*, El Salvador C.A.

### Internet

Metas de Minería de Datos  
<http://dns1.mor.itesm.mx>

WWW como fuente de recursos lingüísticos  
Departamento de Informática. Universidad de Jaén. España

Documento básico de minería de datos.  
DAEDALUS – Data, Decisions and Language, S.A.

Evaluación y selección de una técnica de Minería de Datos.  
Luis Daniel Pineda, Juan Pablo Vega y Andrés Dorado  
Facultad de Ingeniería, Pontificia Universidad Javeriana

Modelamiento Multidimensional.  
Carmen Gloria Wolf

Libros en Pantalla SQL Server 2000.

Librería MSDN.

MDX  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agmdxfunctintro\\_65iw.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agmdxfunctintro_65iw.asp)

ADO MD  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/htm/ammscadoapireference.asp>

## DSO

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/prabout\\_84a4.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/prabout_84a4.asp)

## PivotTable

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/pt\\_intro\\_49wl.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/pt_intro_49wl.asp)

<http://accessvbsqladvisor.com/>

<http://www.winnetmag.net/SQLServer/Article/ArticleID/6199/6199.html>

# ANEXOS

ANEXO A

A.1 Estructura DSO con sus colecciones, incluido el valor de la propiedad ClassType de los objetos contenidos en la colección MDStores.

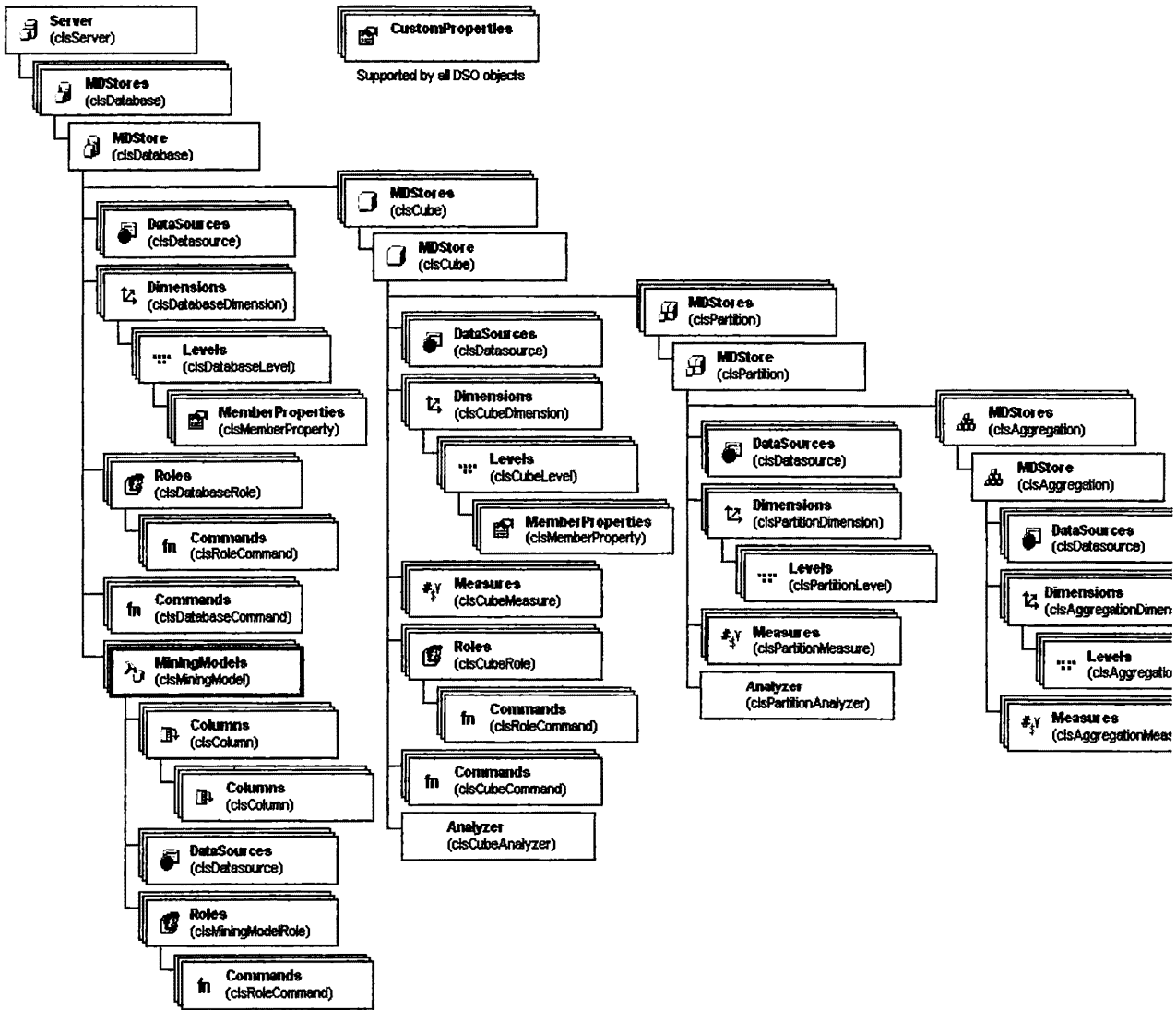


Figura A.1-Estructura DSO.

**A.2           Objetos DSO**

**Tabla A.2.1 - Objetos DSO que se pueden acceder directamente, por medio de la propiedad ClassType**

<b>Objeto</b>	<b>ClassType</b>
Column	<u>clsColumn</u>
Cube analyzer	<u>clsCubeAnalyzer</u>
Data mining model	<u>clsMiningModel</u>
Data source	<u>clsDataSource</u>
Member property	<u>clsMemberProperty</u>
Partition analyzer	<u>clsPartitionAnalyzer</u>
Server	<u>clsServer</u>

**Tabla A.2.2 – Categorías de los objetos DSO que son accedidos por medio de una interfaz.**

<b>Agregaciones</b>
Comandos
Cubos
Base de Datos
Dimensiones
Niveles
Medidas
Particiones
Roles

**A.3 Interfaces DSO.**

**Tabla A.3 – Interfaces DSO comunes y objetos que las implementan.**

<b>Interfaz</b>	<b>Descripción</b>	<b>Implementada por</b>
<u>Command</u>	Muestra la funcionalidad para definir e implementar consultas MDX que se ejecutan en el cliente y en el Análisis Service.	<u>clsDatabaseCommand</u> <u>clsCubeCommand</u> <u>clsRoleCommand</u>
<u>Dimension</u>	Esta interfaz define las propiedades, métodos y colecciones que se pueden usar para manipular los diferentes tipos de dimensiones: de base de datos, cubos, particiones y agregaciones.	<u>clsDatabaseDimension</u> <u>clsCubeDimension</u> <u>clsPartitionDimension</u> <u>clsAggregationDimension</u>
<u>Level</u>	Define objetos que especifican la jerarquía dimensión..	<u>clsDatabaseLevel</u> <u>clsCubeLevel</u> <u>clsPartitionLevel</u> <u>clsAggregationLevel</u>
<u>MDSStore</u>	Los objetos que implementan esta interfaz, son los que contienen las dimensiones de: base de datos, cubos, particiones, y agregaciones.	<u>clsDatabase</u> <u>clsCube</u> <u>clsPartition</u> <u>clsAggregation</u>
<u>Measure</u>	Los objetos que implementan esta interfaz describen los valores almacenados en cubos, particiones y agregaciones.	<u>clsCubeMeasure</u> <u>clsPartitionMeasure</u> <u>clsAggregationMeasure</u>
<u>Role</u>	Los objetos que implementan esta interfaz tiene permisos de accesos, sobre base de datos, cubos y modelos de minería de datos.	<u>clsDatabaseRole</u> <u>clsCubeRole</u> <u>clsMiningModelRole</u>

**A.4 Interfaz MDStore.**

**Tabla A.4.1 – Colecciones que soporta la interfaz MDStore.**

<b>Colección</b>	<b>Descripción</b>
<u>Commands</u>	Comandos definidos por el usuario o secuencia de comandos
<u>CustomProperties</u>	Colección de propiedades definidas por el usuario
<u>DataSources</u>	Colección de objetos especificando un proveedor de datos.
<u>Dimensions</u>	Mantiene las definiciones de las dimensiones de un objeto.
<u>MDStores</u>	Mantiene los objetos <b>MDStore</b> .
<u>Measures</u>	Mantiene las medidas
<u>MiningModels</u>	Colección de modelos de minería de datos contenida con una base de datos.
<u>Roles</u>	Mantiene los roles de usuarios definidos para una base de datos.

**Tabla A.4.2 – Tipos de clases de los objetos que pueden contener las colecciones MDStore**

<b>Colección</b>	<b>Tipo de Clase</b>
<b>Commands</b>	<u>clsCubeCommand</u> <u>clsDatabaseCommand</u>
<b>CustomProperties</b>	<u>Property</u>
<b>DataSources</b>	<u>clsDataSource</u>
<b>Dimensions</b>	<u>clsAggregationDimension</u> <u>clsCubeDimension</u> <u>clsDatabaseDimension</u> <u>clsPartitionDimension</u>
<b>MDStores</b>	<u>clsAggregation</u> <u>clsCube</u> <u>clsDatabase</u> <u>clsPartition</u>
<b>Measures</b>	<u>clsAggregationMeasure</u> <u>clsCubeMeasure</u>

	<u>clsPartitionMeasure</u>
<b>MiningModels</b>	<u>clsMiningModel</u>
<b>Roles</b>	<u>clsCubeRole</u> <u>clsDatabaseRole</u> <u>clsMiningModelRole</u>

**Tabla A.4.3 – Métodos que soporta la interfaz MDStore.**

<b>Método</b>	<b>Descripción</b>
<u>BeginTrans</u>	Inicia una transacción en una Base de Datos
<u>Clone</u>	Copia un objeto existente a otro del mismo tipo de clase.
<u>CommitTrans</u>	Compromete una transacción.
<u>CommitTransEx</u>	Compromete una transacción , dando control adicional sobre los procesos en reposo.
<u>LockObject</u>	Cierra un objeto para prevenir que los usuarios lo cambien.
<u>Merge</u>	Mezcla dos particiones..
<u>Process</u>	Procesa un objeto <b>MDStore</b> .
<u>Rollback</u>	Rebobina una transacción.
<u>UnlockObject</u>	Abre un objeto previamente cerrado.
<u>Update</u>	Actualiza las definiciones de un objeto en el repositorio de la meta dato.

**Tabla A.4.4 - Propiedades de interfaz MDStore.**

<b>Propiedad</b>
<u>AggregationPrefix</u>
<u>AllowDrillThrough</u>
<u>Analyzer</u>
<u>ClassType</u>
<u>DefaultMeasure</u>
<u>Description</u>
<u>DrillThroughColumns</u>
<u>DrillThroughFilter</u>

<u>DrillThroughFrom</u>
<u>DrillThroughJoins</u>
<u>EstimatedRows</u>
<u>EstimatedSize</u>
<u>FromClause</u>
<u>IsDefault</u>
<u>IsReadWrite</u>
<u>IsTemporary</u>
<u>IsValid</u>
<u>IsVisible</u>
<u>JoinClause</u>
<u>LastProcessed</u>
<u>LastUpdated</u>
<u>Name</u>
<u>OlapMode</u>
<u>Parent</u>
<u>ProcessingKeyErrorLimit</u>
<u>ProcessingKeyErrorLogFileName</u>
<u>ProcessOptimizationMode</u>
<u>RemoteServer</u>
<u>Server</u>
<u>SourceTable</u>
<u>SourceTableAlias</u>
<u>SourceTableFilter</u>
<u>State</u>
<u>SubClassType</u>

## A.5 Referencias API de ADO MD

**Tabla A.5.1 – Objetos ADO MD**

<u>Axis</u>	Representa coordenadas posicionales o filtros de coordenadas de un conjunto de celdas, tiene miembros seleccionados de una o mas dimensiones.
<u>Catalog</u>	Contiene información de un esquema multidimensional (cubos, dimensiones, jerarquías, niveles y miembros) específicos de un proveedor de datos multidimensionales.
<u>Cell</u>	Representa los datos de la intersección de un eje de coordenadas, contenidas en un cellset.
<u>Cellset</u>	Representa los resultados de una consulta multidimensional. Esto es, una colección de celdas seleccionadas de cubos u otros conjuntos de celdas.
<u>CubeDef</u>	Representa un cubo desde un esquema multidimensional, tiene un conjunto de dimensiones relacionadas.
<u>Dimension</u>	Representa una de las dimensiones de un cubo multidimensional, contiene una o más jerarquias de miembros.
<u>Hierarchy</u>	Representa una forma en la que los miembros de una dimensión pueden ser agregados o "rolled up." Una dimensión puede ser agregada a lo largo de una o más jerarquías.
<u>Level</u>	Contiene un conjunto de miembros, cada uno de los cuales tiene la misma categoría con una jerarquía.
<u>Member</u>	Representa un miembro de un nivel del cubo, el hijo del miembro de un nivel o un miembro de una posición a lo largo del eje de las de un conjunto de celdas
<u>Position</u>	Representa un conjunto de uno o más miembros de diferentes dimensiones, estos definen un punto a lo largo de un eje.

**Tabla A.5.2 – Colecciones ADO MD.**

<u>Axes</u>	Contiene los objetos <b>Axis</b> que definen los cellset.
<u>CubeDefs</u>	Contiene los objetos <b>CubeDef</b> , que representan un cubo desde un catalogo multidimensional.
<u>Dimensions</u>	Contiene el objeto <b>Dimension</b> , que hace un cubo.
<u>Hierarchies</u>	Contiene el conjunto de objetos <b>Hierarchy</b> , desde una dimensión.
<u>Levels</u>	Contiene el objeto <b>Level</b> , que hace una jerarquía.
<u>Members</u>	Contiene el objeto <b>Member</b> desde un nivel o una posición, a lo largo de un eje.
<u>Positions</u>	Contiene el objeto <b>Position</b> , que define un punto sobre un eje.

**Tabla A.5.3 – Propiedades ADO MD.**

<u>ActiveConnection</u>	Indica a cual objeto de conexión ADO, esta asociado actualmente el catalogo o el cellset.
<u>Caption</u>	Indica el titulo usado cuando se despliega un objeto Nivel o Miembro
<u>ChildCount</u>	Indica el número de miembros por los cuales el actual objeto Miembro es padre en una jerarquía.
<u>Children</u>	Regresa una colección de miembros, para los cuales, el actual es el padre de una jerarquía.
<u>Count</u>	Indica el número de objetos de una colección.
<u>Depth</u>	Indica el número de niveles entre el Nivel y la raíz, del nivel de jerarquía.
<u>Description</u>	Descripción del objeto actual.
<u>DimensionCount</u>	Número de dimensiones sobre un eje.
<u>DrilledDown</u>	Indica si no hay hijos despues de el miembro sobre un eje..
<u>FilterAxis</u>	Información del filtro del actual conjunto de celdas.
<u>Item</u>	Retrieves a cell from a cellset using its coordinates.
<u>Item</u>	Regresa un especifico miembro de una colección por nombre o número ordinal.
<u>FormattedValue</u>	Indica el formato desplegado de un valor en una celda.
<u>LevelDepth</u>	Indica el número de niveles entre la raíz de la jerarquía y un miembro.
<u>LevelName</u>	EL nombre del nivel de un miembro.
<u>Name</u>	Nombre de un objeto.
<u>Ordinal (Cell)</u>	Identifica una celda por su posición dentro de un conjunto de

	celda.
<u>Ordinal (Position)</u>	Identifica una posición, a lo largo de un eje.
<u>Parent</u>	Indica el miembro que es padre del actual miembro en una jerarquía.
<u>ParentSameAsPrev</u>	Indica sí el padre del miembro posición es el mismo del miembro precedido inmediatamente.
<u>Source</u>	Indica la fuente de los datos en el cellset.
<u>State</u>	Estado actual del cellset.
<u>Type</u>	Tipo del miembro actual.
<u>UniqueName</u>	Indica un nombre sin ambigüedad para el actual objeto.
<u>Value</u>	Valor de la celda actual

### Tabla A.5.4 – Métodos ADO MD.

<u>Close</u>	Cierra un cellset abierto.
<u>GetSchemaObject</u>	Recuocera un objeto del esquema ADO MD (dimensión, jerarquía, nivel o miembro) por su nombre único.
<u>Open</u>	Recupera los resultados de una consulta multidimensional.

### A.5.3 – Ejemplo del objeto Cellset usando Visual Basic.

El siguiente proyecto de Visual Basic<sup>1</sup>, demuestra el uso básico de ADO MD, para acceder un cubo de datos. Despliega miembros de títulos para encabezados de columnas y filas, despliega valores formateados de celdas específicas dentro del cellset.

```
Private Sub cmdCellSettoDebugWindow_Click()  
Dim cat As New ADOMD.Catalog  
Dim cst As New ADOMD.Cellset  
Dim i As Integer  
Dim j As Integer  
Dim strServer As String  
Dim strSource As String  
Dim strColumnHeader As String
```

---

<sup>1</sup> El código necesita de Visual Basic 6.0 o Visual Basic 5.0 con Service Pack 3.

Dim strRowText As String

On Error GoTo Error\_cmdCellSettoDebugWindow\_Click

Screen.MousePointer = vbHourglass

'\*-----

'\* Coloca el servidor al LocalHost.

'\*-----

strServer = "LOCALHOST"

'\*-----

'\* Coloca la consulta MDX

'\*-----

strSource = strSource & "SELECT "

strSource = strSource & "[Measures].members ON COLUMNS,"

strSource = strSource & \_

"NON EMPTY [Store].[Store City].members ON ROWS"

strSource = strSource & " FROM Sales"

'\*-----

'\* Fija la conexión activa

'\*-----

cat.ActiveConnection = "Data Source=" & strServer & \_

";Provider=msolap;"

'\*-----

'\* Fija la fuente Cell Set a la consulta MDX

'\*-----

cst.Source = strSource

'\*-----

'\* Fija la conexión activa a la actual

'\*-----

```
Set cst.ActiveConnection = cat.ActiveConnection
```

```
***-----
```

```
*** Abre Cell Set
```

```
***-----
```

```
cst.Open
```

```
***-----
```

```
*** Permite espacios para encabezados de fila
```

```
***-----
```

```
strColumnHeader = vbTab & vbTab & vbTab & vbTab & vbTab & vbTab
```

```
***-----
```

```
*** Lazo que recorre el encabezado de columnas
```

```
***-----
```

```
For i = 0 To cst.Axes(0).Positions.Count - 1
    strColumnHeader = strColumnHeader & _
        cst.Axes(0).Positions(i).Members(0).Caption & vbTab & _
        vbTab & vbTab & vbTab
Next
Debug.Print vbTab & strColumnHeader & vbCrLf
```

```
***-----
```

```
*** Lazo que recorre el encabezado de filas y provee datos para cada una de ellas.
```

```
***-----
```

```
strRowText = ""
For j = 0 To cst.Axes(1).Positions.Count - 1
    strRowText = strRowText & _
        cst.Axes(1).Positions(j).Members(0).Caption & vbTab & _
        vbTab & vbTab & vbTab
    For k = 0 To cst.Axes(0).Positions.Count - 1
        strRowText = strRowText & cst(k, j).FormattedValue & _
```

vbTab & vbTab & vbTab & vbTab

Next

Debug.Print strRowText & vbCrLf

strRowText = ""

Next

Screen.MousePointer = vbDefault

Exit Sub

Error\_cmdCellSettoDebugWindow\_Click:

Beep

Screen.MousePointer = vbDefault

MsgBox "The Following Error has occurred:" & vbCrLf & \_

Err.Description, vbCritical, " Error!"

Exit Sub

End Sub

## A.6 Funciones MDX.

**Tabla A.6.1 – Funciones de dimensiones.**

<b>Función</b>	<b>Descripción</b>
<u>Dimension</u>	Regresa la dimensión que tiene una jerarquía, nivel o miembro específico.
<u>Dimensions</u>	Retorna la dimensión, en la cual, posición basada en cero dentro del cubo es especificada por una expresión numérica o nombres especificados por una cadena.

**Tabla A.6.2 – Funciones de jerarquía.**

<b>Función</b>	<b>Descripción</b>
<u>Hierarchy</u>	Regresa la jerarquía de un nivel o miembro.

**Tabla A.6.3 – Funciones de nivel.**

<b>Función</b>	<b>Descripción</b>
<u>Level</u>	Regresa el nivel de un miembro.
<u>Levels</u>	Retorna el nivel, en el cual, la posición en una dimensión es especificada por una expresión numérica o el nombre por una expresión de cadena.

## A.7 Propiedades del Servicio PivotTable.

Tabla A.7.1 – Propiedades PivotTable.

<b>Propiedad</b>
<u>ArtificialData Property</u>
<u>Authenticated User Property</u>
<u>Auto Synch Period Property</u>
<u>Cache Policy Property</u>
<u>Cache Ratio Property</u>
<u>Client Cache Size Property</u>
<u>CompareCaseNotSensitiveStringFlags Property</u>
<u>CompareCaseSensitiveStringFlags Property</u>
<u>Connect Timeout Property</u>
<u>CreateCube Property</u>
<u>Data Source Property</u>
<u>Datasource Connection Type Property</u>
<u>Default GUID Dialect Property</u>
<u>Default Isolation Mode Property</u>
<u>Default MDX Visual Mode Property</u>
<u>Distinct Measures By Key Property</u>
<u>Do Not Apply Commands Property</u>
<u>Execution Location Property</u>
<u>Initial Catalog Property</u>
<u>InsertInto Property</u>
<u>Large Level Threshold Property</u>
<u>Locale Identifier Property</u>
<u>Log File Property</u>
<u>MDX Calculated Members Mode Property</u>

<u>MDX Compatibility Property</u>
<u>MDX Object Qualification Property</u>
<u>MDX Unique Name Style Property</u>
<u>Mining Execution Location Property</u>
<u>Mining Location Property</u>
<u>Mining Persistence Format Property</u>
<u>OLE DB for OLAP Version Property</u>
<u>Password Property</u>
<u>Provider Property</u>
<u>Read Only Session Property</u>
<u>Restricted Client Property</u>
<u>Roles Property</u>
<u>Safety Options Property</u>
<u>Secured Cell Value Property</u>
<u>Show Hidden Cubes Property</u>
<u>Source DSN Property</u>
<u>Source DSN Suffix Property</u>
<u>SQL Compatibility Property</u>
<u>SSPI Property</u>
<u>UseExistingFile Property</u>
<u>User ID Property</u>
<u>Writeback Timeout Property</u>

## **A.8 Componentes y Referencias utilizadas en el OLAP Viewer.**

### **A.8.1 Componentes**

**Microsoft Chart Control 6.0 (OLEDB) - MSCHRT20.OCX**, Control que por si mismo es fácil de usar, utilizado para dibujar gráficos dependiendo de los datos de entrada.

**Microsoft Common Dialog Control 6.0 (SP3) – COMDLG32.OCX**, se utiliza para verificar si un archivo existe.

**Microsoft DataGrid Control 6.0 (SP5) –MSDATGRD.OCX**, usado para desplegar datos en las celdas de una tabla con una conexión ADO, recuperando la información de una base de datos.

**Microsoft FlexGrid Control 6.0 (SP3) –MSFLXGRD.OCX**, estableciendo una conexión a la base de datos y extrayendo la información de esta, los resultados se pueden desplegar en un flexgrid utilizando este control

**Microsoft Hierarchical FlexGrid Control 6.0 (SP3) –MSHFLXGD.OCX**, es un Grid jerárquico muy cómodo, permite hacer operaciones simples y rápidas, asignándoles recordsets directamente. No permite la edición sobre el Grid.

**Microsoft Masked Edit Control 6.0 –MSMASK32.OCX**, control que sirve para asignar mascarar a las entradas de un cuadro de texto

**Microsoft Windows Common Controls 5.0 (SP2) - Comdlg32.ocx**, utilizado para controlar la barra de herramientas(ToolBar), barra de estado(StatusBar), vistas de árbol(TreeView), listas(ListView), TabStrip.

**Microsoft Windows Common Controls 6.0 (SP6) - Comdlg32.ocx**, se usa para tener todos los controles comunes en los proyectos, viene con la versión 6.0 de Visual Basic.

**Microsoft Windows Common Controls-2 5.0 (SP2), Microsoft Windows Common Controls-2 6.0 (SP4), Microsoft Windows Common Controls-3 6.0**

Todos con OCX **Comdlg32.ocx**, utilizado para controlar barra de herramientas, barra de estado, vista de árbol, listas, todos los controles comunes en los proyectos, varía la versión del programa.

**MSFlexGrid Wizard – FlexWiz.ocx**, usado para desplegar datos en celdas de una tabla dinámica, estableciendo una conexión y recuperando la información de la base de datos.

**SexyControls – sexyControls.ocx**, control utilizado para añadir botones con diseños predeterminados dentro del programa. Componente de propietario particular utilizado para crear botones con aspecto multicolor.

**Microsoft Data Mining Model Browser**, componente utilizado para trabajar y proporcionar acceso hacia los modelos de minería de datos desde Visual Basic

**Microsoft XP Web Components**, colección de controles COM para publicar hojas de cálculos, gráficos y bases de datos para la web.

**ChameleonButton.**

Componente de propietario particular utilizado para crear botones con aspecto de Windows XP.

## A.8.2 Referencias (librerías)

**Visual Basic for Applications (Visual Basic para aplicaciones).** Subconjunto de Visual Basic que provee un lenguaje de macros común para aplicaciones de Microsoft. VBA, permite a usuarios y programadores poderosos ampliar la funcionalidad de programas como Word, Excel y Access.

**Microsoft Activex Data Objects 2.0 Library,** Para conectarse y acceder la base de datos por medio de ADO.

**Microsoft DAO 3.51 Object Library,** para acceder a la base de datos por medio de DAO.

**Microsoft Data Binding Collection,** la colección DataBinding es una propiedad extendida que Visual Basic proporciona para el control de los usuarios. Permite a los desarrolladores acceder a las listas de las propiedades enlazadas en sus controles. La colección Data binding, permite que cualquier control sea enlazado al entorno de datos (Data Enviroment) y a los controles ADO.

**Microsoft OLAP Services Add-Ins Manager,** esta librería contiene las clases, objetos, métodos, propiedades, enumeraciones y colecciones que se pueden usar en los programas.

**Microsoft Actives Data Objects (Multi-dimensional),** es un modelo de objeto del lado del cliente, que llama a las API de OLAP, proporcionado por MSADOMD.DLL, la fuente de la API, viene con Microsoft's Data Access Components (MDAC). Utilizado para acceder y trabajar con datos multidimensionales.

**Visual Basic runtime objects y procedures**

**Visual Basic objects y procedures**

**OLE Automation**

**Microsoft Graph 9.0 Object Library**

**Microsoft OLAP Designer Server Driver**

**Microsoft OLAP Designer Server Driver 8.0**

**OLAP Manager MDX function resource string**

**Microsoft ActiveX Plugin**

**Microsoft Decision Support Objects**

**Data Dynamics ActiveReports 2.0**

## **A.9 Transformación de los datos (DTS)**

Muchas organizaciones tienen que centralizar sus datos para mejorar la toma de decisiones. Sin embargo, los datos podrían estar almacenados en muchos formatos diferentes y en distintas ubicaciones. Los Servicios de transformación de datos (DTS) permiten satisfacer esta necesidad crucial de la empresa. Para ello, proporcionan un conjunto de herramientas que permiten extraer, transformar y consolidar datos de distintos orígenes en uno o varios destinos compatibles con la conectividad DTS. Para crear soluciones de transferencia de datos personalizadas para las necesidades específicas de la empresa.

### **A.9.1 Paquetes DTS**

Los paquetes DTS son conjuntos organizados de conexiones, tareas DTS, transformaciones DTS y restricciones de flujo de trabajo reunidas mediante programación o una herramienta de DTS y guardadas en Microsoft SQL Server, Meta Data Services de Microsoft SQL Server 2000, un archivo de almacenamiento estructurado o un archivo de Microsoft Visual Basic.

Cada paquete contiene uno o más pasos que se ejecutan de forma secuencial o en paralelo cuando se ejecuta el paquete. Cuando se ejecuta el paquete, se conecta con los orígenes de datos apropiados, copia los datos y los objetos de base de datos, transforma los datos y notifica los sucesos a otros usuarios o procesos.

### **A.9.2 Tareas DTS**

Una tarea DTS es un conjunto discreto de funcionalidad, ejecutada como un único paso de un paquete. Cada tarea define un elemento de trabajo que debe realizarse como parte de la transferencia de datos y el proceso de transformación de datos, o como un trabajo que debe ejecutarse.

DTS proporciona varias tareas que forman parte del modelo de objetos de DTS y a las que se puede tener acceso a través del Diseñador DTS o mediante programación. Estas tareas, que puede configurar de forma individual, cubren una amplia gama de situaciones de copia de datos, transformación de datos y notificación. Por ejemplo:

### **Importación y exportación de datos**

DTS puede importar datos de un archivo de texto o de un origen de datos OLE DB (por ejemplo, una base de datos de Microsoft Access 2000) en SQL Server. De forma alternativa, puede exportar datos desde SQL Server a un destino de datos OLE DB (por ejemplo, una hoja de cálculo Microsoft Excel 2000). DTS también permite la carga de datos de alta velocidad desde archivos de texto a tablas de SQL Server.

### **Transformación de datos**

El Diseñador DTS incluye la tarea Transformar datos, que permite seleccionar datos de una conexión de origen de datos, asignar las columnas de datos a un conjunto de transformaciones y enviar los datos transformados a una conexión de destino. El Diseñador DTS también contiene una tarea de consulta controlada por datos que permite asignar datos a consultas parametrizadas.

### **Copia de objetos de base de datos**

Con DTS, se puede transferir, además de los datos, índices, vistas, inicios de sesión, procedimientos almacenados, desencadenadores, reglas, valores predeterminados, restricciones y tipos de datos definidos por el usuario. Además, se puede generar secuencias de comandos para copiar los objetos de base de datos.

### **Intercambio de mensajes con otros usuarios y paquetes**

DTS incluye una tarea Enviar correo que permite enviar un mensaje de correo electrónico en caso de que se realice correctamente un paso de un paquete o si se produce un error. DTS también incluye una tarea Ejecutar paquete, que permite que un paquete ejecute otro paquete como un paso del paquete original, y una tarea Cola de mensajes, que permite utilizar Message Queue Server para el intercambio de mensajes entre paquetes.

### **Ejecución de un conjunto de instrucciones Transact-SQL o secuencias de comandos Microsoft ActiveX® en un origen de datos**

Las tareas de ejecución de SQL y de secuencia de comandos ActiveX permiten escribir instrucciones SQL y código de secuencia de comandos, y ejecutarlo como un paso del flujo de trabajo de un paquete.

### **A.9.3 Transformaciones DTS**

Una transformación DTS consiste en una o más funciones u operaciones aplicadas a un conjunto de datos antes de que éstos lleguen al destino. No se modifica el origen de datos. Por ejemplo, Se puede extraer una subcadena de una columna de datos de origen y copiarla a la tabla de destino. La función de subcadena concreta es la transformación asignada a la columna de origen. También se puede buscar filas con determinadas características (por ejemplo, valores de datos específicos en columnas) y aplicar las funciones únicamente a los datos de dichas filas. Las transformaciones facilitan la implementación de la validación de datos complejos, la limpieza de datos y las conversiones durante el proceso de importación y exportación.

Ejemplo para crear una tarea de transformación de datos (DTS)

```
Private Sub cmdCreateTaskAndPackage_Click()
On Error GoTo ProcErr

'Variables usadas para todas las tareas
Dim bSuccess As Boolean
Dim stp As DTS.Step2
Dim sBaseName As String
Dim sStepName As String
Dim msg As String

'Variables para especificar que tipo de Cliente es
Dim sSourceDataSource As String
Dim sDestDataSource As String
Dim sSourceCatalog As String
Dim sDestCatalog As String
Dim sSourceUserID As String
Dim sDestUserID As String
Dim sSourcePassword As String
Dim sDestPassword As String
Dim sSourceTableName As String
Dim sDestTableName As String
Dim sExistingSource As String
Dim sExistingDest As String

'Asignar el nombre de la base de datos
sBaseName = txtBaseName.Text

'Seleccionar el servidor donde el paquete se guardara
msServer = txtServer.Text

msUserID = TxtUserID.Text
msPassword = TxtPassword.Text

'Creando un nuevo paquete
Set pkg = CreateObject("DTS.Package2")

'Creando un nombre unico para el paquete
sBaseName = fctFindUniquePackageName(sBaseName, msServer, msUserID,
msPassword)
If sBaseName = "" Then

    GoTo ProcExit
Else
    pkg.Name = sBaseName & " Package"
End If
```

```
'Fijar las características que son específicas para esta clase de tarea
sSourceUserID = txtSourceUserID.Text
sDestUserID = txtDestUserID.Text

sSourcePassword = txtSourcePassword.Text
sDestPassword = txtDestPassword.Text

sSourceDataSource = txtSourceDataSource.Text      'Nombre del servidor
sDestDataSource = txtDestDataSource.Text

sSourceCatalog = txtSourceCatalog.Text           'Nombre de la base datos
sDestCatalog = txtDestCatalog.Text

sSourceTableName = txtSourceTableName.Text       'Nombre de las tablas
sDestTableName = txtDestTableName.Text

sExistingSource = "" 'txtExistingSource.Text
sExistingDest = "" 'txtExistingDest.Text

'Llamar el procedimiento para crear la nueva tarea
sStepName = fctCreateTransformDataTask(pkg, sBaseName, sSourceDataSource,
sDestDataSource, sSourceCatalog, sDestCatalog, _
    sSourceUserID, sDestUserID, sSourcePassword, sDestPassword,
sSourceTableName, sDestTableName, _
    sExistingSource, sExistingDest)
'Guardar el nuevo paquete si la tarea y los pasos fueron creados satisfactoriamente
If sStepName <> "" Then
    'Guardar el paquete
    If msUserID = "" Then
        pkg.SaveToSQLServer msServer, , , 256
    Else
        pkg.SaveToSQLServer msServer, msUserID, msPassword
    End If

    Set stp = pkg.Steps(sStepName)
    Set tsk = pkg.Tasks(stp.TaskName)

    cmdAddTransformations.Enabled = True
    Set stp = Nothing
Else
    MsgBox "Hubo un error al crear esta tarea. El paquete no fue guardado."
End If
ProcExit:
Exit Sub
ProcErr:
MsgBox Err.Number & " - " & Err.Description
GoTo ProcExit
End Sub
```

## **ANEXO B**

- ❖ **Manual de Instalacion**
- ❖ **Manual del Usuario**
- ❖ **Manual del Programador**

**OLAP**



**Viewer**  
version 1.0



**Manual de Instalación**



## INDICE

Introducción.....	I
1. Requerimientos de Hardware.....	1
2. Requerimientos de Software.....	1
3. Base de Datos (Datamarts ).....	1
3.1 Cargar SQL Enterprise Manager.....	1
3.2 Restauración de Base de Datos.....	2
4. Componentes.....	5
5. Librerías DLL.....	7
6. Instalación del Sistema.....	8
7. Acceso al Sistema.....	10

## INDICE DE IMÁGENES

### ❖ FIGURAS

Figura 1. Recuperar Base de Datos.....	2
Figura 2. Restaurar Base de Datos.....	3
Figura 3. Cambiar Dispositivo.....	4
Figura 4. Especificación de Ruta.....	4
Figura 5. Instalación Paso 1.....	8
Figura 6. Instalación Paso 2.....	8
Figura 7. Instalación Paso 3.....	9
Figura 8. Instalación Paso 4.....	9
Figura 9. Acceso al Sistema.....	10

## **INTRODUCCIÓN**

Bienvenidos al Manual de instalación del Sistema "OLAP Viewer". Este manual contiene información útil desde restauración de las bases de datos utilizadas en el sistema, como la preparación para la instalación, al procedimiento de instalación paso a paso.

Este manual lo guiará a través de una instalación típica, una vez que se haya completado la instalación como se describe aquí, se tendrá un sistema completamente funcional.

## 1. Requerimientos de Hardware.

Los requerimientos mínimos de hardware para el funcionamiento de la herramienta OLAP Viewer se describen a continuación.

- ✓ Microprocesador Pentium o Celeron de 750 Mhz.
- ✓ 128 M de RAM.
- ✓ Disco de duro de 20 G (dependiendo de la cantidad de información).

## 2. Requerimientos de Software.

Los requerimientos de software para el buen funcionamiento de OLAP Viewer se detallan a continuación.

- ✓ Microsoft Windows 2000 Server.
- ✓ Microsoft SQL Server 2000.
- ✓ Analysis Service (SQL Server 2000).
- ✓ Microsoft Visual Basic 6.0
- ✓ Resolución de 800x600 Píxeles.

## 3. Bases de Datos (Datamarts).

Las Bases de Datos que se utilizan en el sistema son:

- ✓ DATAMART VENTAS
- ✓ DATAMART PRODUCCIÓN

Para poder trabajar con los Datamarts es necesario recuperar los Backups, la carpeta donde se encuentran los archivos se llama BACKUP DATAMARTS/FINALES.

Para poder realizar una recuperación siga los siguientes pasos:

### 3.1 Cargar SQL Enterprise Manager.

#### Crear nueva base de datos.

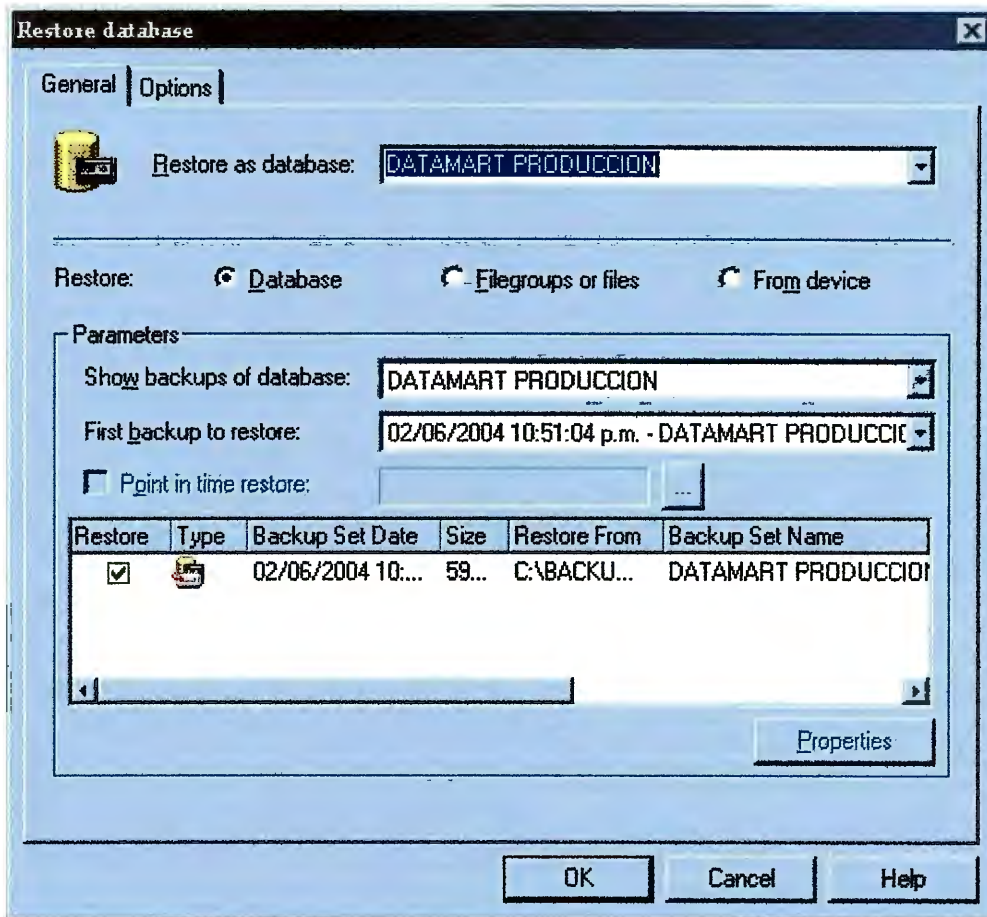
Clic derecho en la carpeta de Bases de datos, Nueva Base de datos.

En esta parte es importante nombrar la base de datos tal y como se llamaba originalmente, para el caso de los Datamarts los nombres son DATAMART VENTAS y DATAMART PRODUCCIÓN, revisar este paso ya que de lo contrario no funcionara la restauración de base de datos.

### 3.2 Restauración de Base de Datos.

He aquí los pasos para recuperar una base de datos.

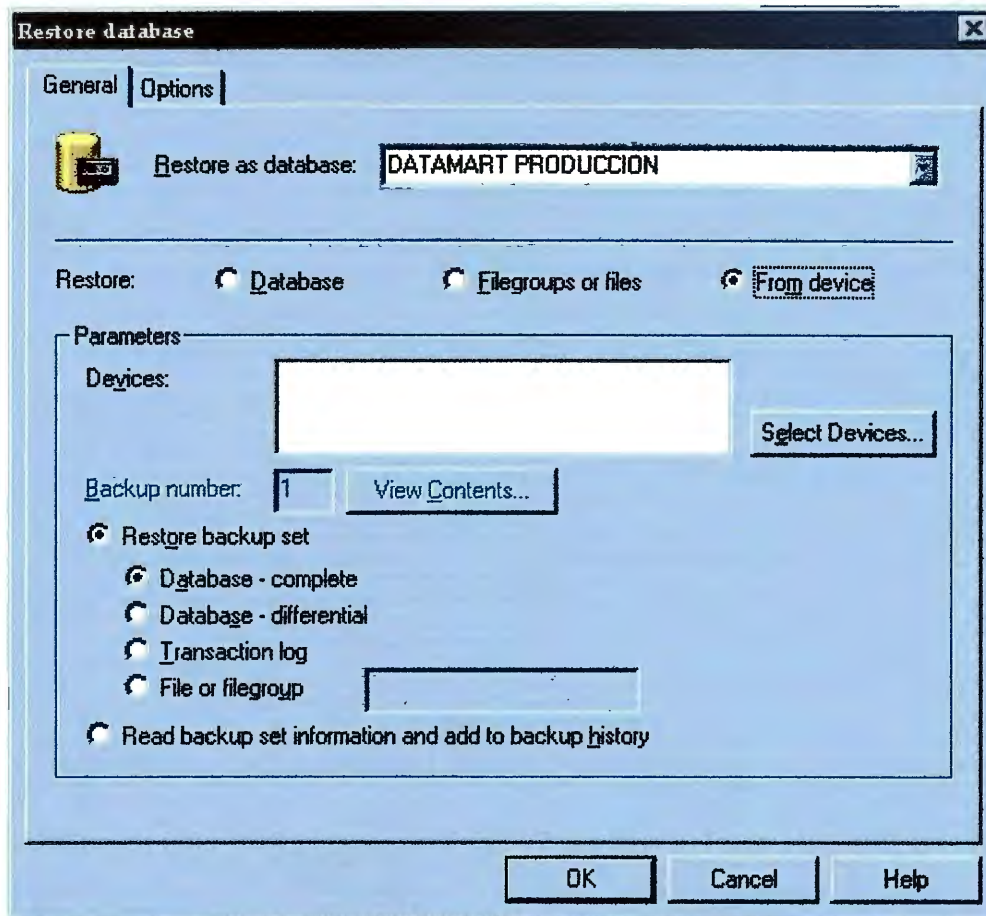
Clic derecho en la base de datos, Clic en la opción todas las tareas, escoger Recuperar base de datos.



**Figura 1. Recuperar Base de Datos**

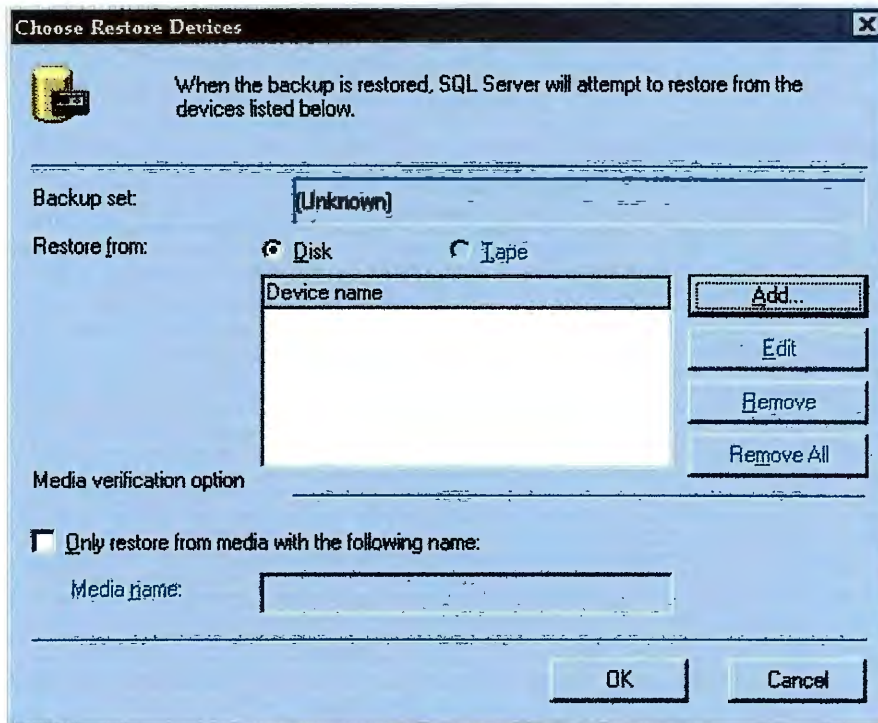
Proceder a recuperar la base de datos desde un dispositivo de almacenamiento, en este caso seleccionar disco duro o un CD.

Elegimos la opción de Recuperar la Base de Datos Completa.



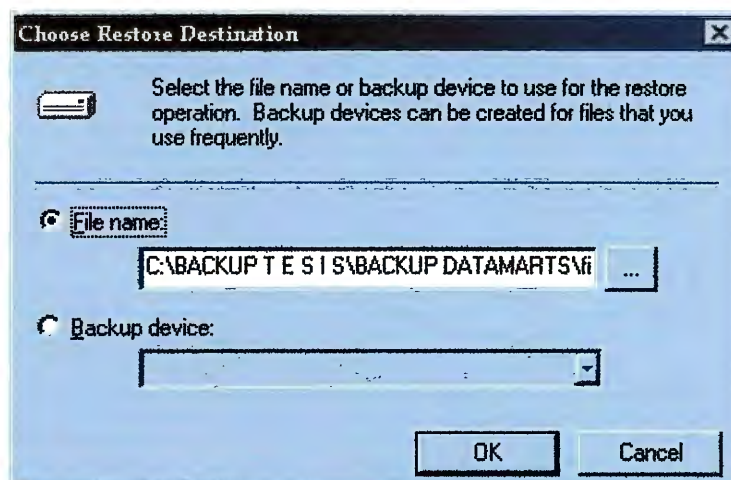
**Figura 2. Restaurar Base de Datos**

Para poder realizar la recuperación es necesario especificar el dispositivo de almacenamiento. Clic en **Add** para agregar dispositivo.



**Figura 3. Cambiar Dispositivo**

Finalmente especificar la ruta donde se encuentran los archivos de recuperación.



**Figura 4. Especificación de Ruta**

## 4. Componentes

Microsoft Visual Basic 6.0 posee una diversidad de componentes que se utilizan para el desarrollo de aplicaciones orientadas a objetos. Los componentes utilizados en la Herramienta OLAP Viewer se detallan a continuación:

### **Picture Box**

Componente que se utiliza para agregar imágenes a los formularios.

### **Label**

Componente que se utiliza para agregar etiquetas a diversidad de objetos en los formularios.

### **TextBox**

Componente que se utiliza para agregar cajas de texto a los formularios.

### **Frame**

Componente que se utiliza para agregar Frames, es decir campos de separación de datos en los formularios.

### **CommandButton**

Componente que se utiliza para agregar botones, estos objetos son los que se encargan de realizar diversidad de acciones dentro de los formularios.

### **ComboBox**

Componente que se utiliza para agregar combos de información en los formularios.

### **TreeView**

Componente que se utiliza para manipular datos jerárquicamente, tal como lo hace el explorador de Windows.

### **StatusBar**

Componente que se utiliza para agregar campos de información en los formularios MDI.

### **ImageList**

Componente que se utiliza para programar otros componentes como el StatusBar y TreeView.

### **Line**

Componente que se utiliza para agregar líneas en los formularios.

### **Datagrid**

Componente que se utiliza para mostrar información grillada.

**MaskedTextBox**

Componente que se utiliza para programar objetos como TextBox, utilizado para definir formatos de fecha.

**ChameleonButton**

Componente de propietario particular utilizado para crear botones con aspecto de Windows XP.

**PivotTable**

Componente que se descarga desde Microsoft Office XP, se utiliza para generar tablas dinámicas y se relaciona con otros componentes para mostrar información.

**ChartSpace**

Componente que se utiliza para generar gráficos dinámicos en combinación con el componente PivotTable.

**DMMBrowser**

Componente que se utiliza para analizar modelos de minería de datos previamente creados en Microsoft SQL Server 2000 en combinación con el componente PivotTable.

**MSFlexgrid**

Componente que se utiliza para mostrar datos grillados.

**SexyCmdButton**

Componente de propietario particular utilizado para crear botones con aspecto multicolor.

**DTPicker**

Componente utilizado para mostrar la fecha.

## 5. Librerías (dll)

Microsoft Visual Basic 6.0 posee una diversidad de librerías que se utilizan para el desarrollo de aplicaciones orientadas a objetos. Las librerías utilizadas en la Herramienta OLAP Viewer se detallan a continuación:

### **DSO.**

Librería que interactúa con las API de Microsoft SQL Server 2000 y que se combina con otro tipo de librerías. Esta librería permite la iteración con variables multidimensionales alojadas en el SQL Análisis Manager.

### **MDStore.**

Librería que se combina con DSO (Objetos de toma de decisión) para la manipulación de datos como cubos multidimensionales y bases de datos entre otros.

### **Microsoft Active Data Object (Multi-dimensional) 2.6**

Librería que se encarga de proveer componentes de objetos de datos DSO y MDStore.

### **Data Dynamics ActiveReport.**

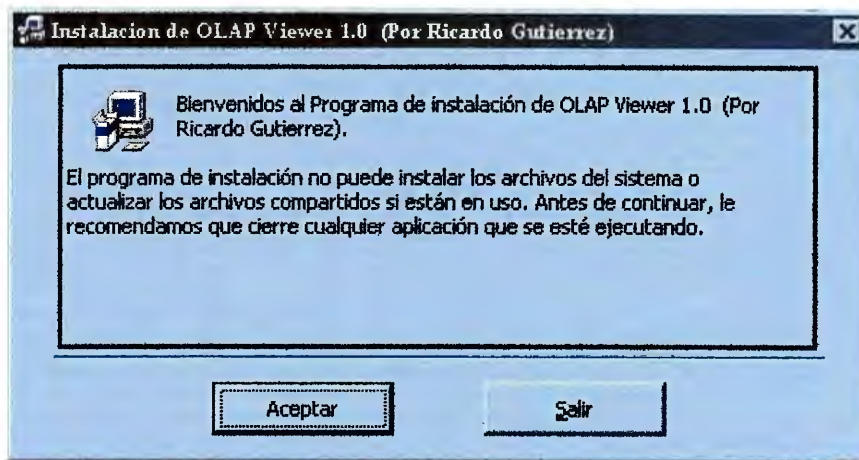
Librería utilizada para generar reportes multidimensionales dinámicos, se agrega por separado e incluye un formulario para reporte de datos.

## 6. Instalación

El programa OLAP Viewer ha sido empaquetado de tal forma que se creo un archivo ejecutable en donde están almacenados los componentes y librerías requeridos para su funcionamiento, además de facilitar y brindar una mejor disponibilidad de reubicación de archivos a los usuarios.

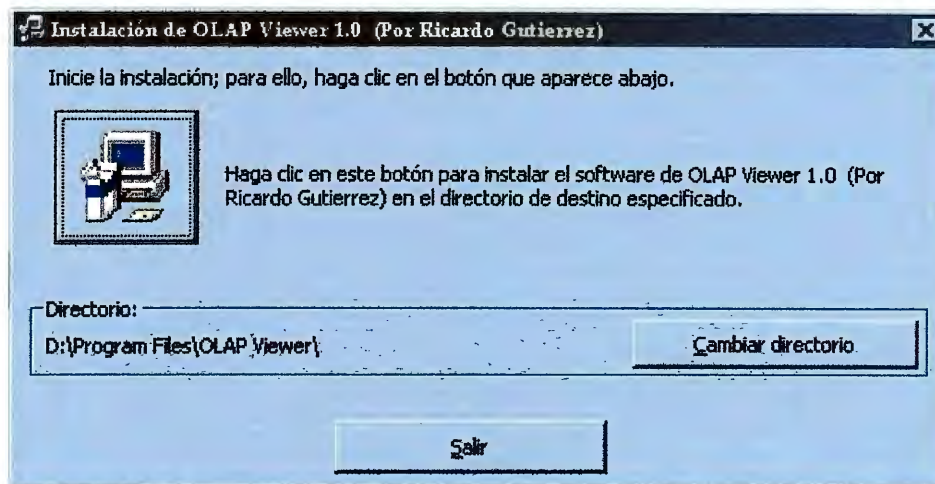
Hacer Clic en **SETUP.exe**, archivo ubicado en la carpeta **Instalador OLAP Viewer**, de esta manera se comenzara a desempaquetar todos los componentes necesarios para su correcto funcionamiento.

Aparecerá una pantalla como la siguiente.



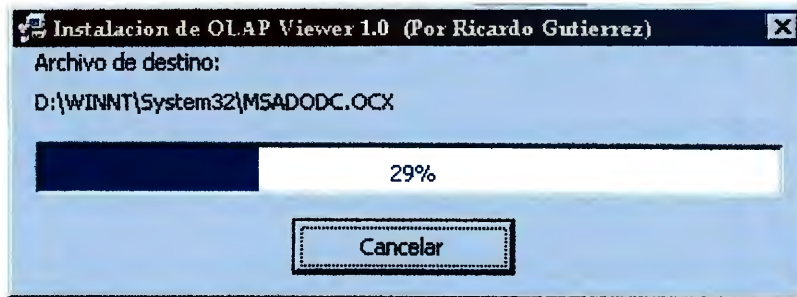
**Figura 5. Instalación Paso 1**

Clic en aceptar para empezar a instalar el programa.



**Figura 6. Instalación Paso 2**

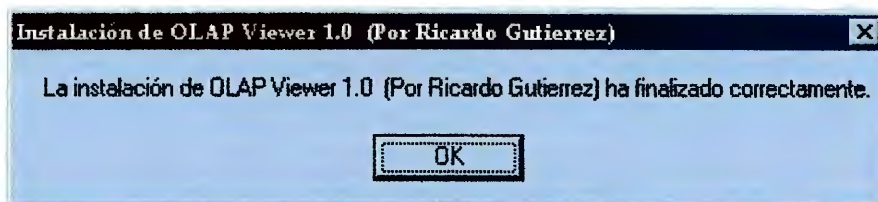
El instalador permite cambiar la localización de los archivos a descargar. Clic en el botón para empezar a instalar el programa OLAP Viewer.



**Figura 7. Instalación Paso 3**

Las librerías se instalan en el directorio previamente especificado.

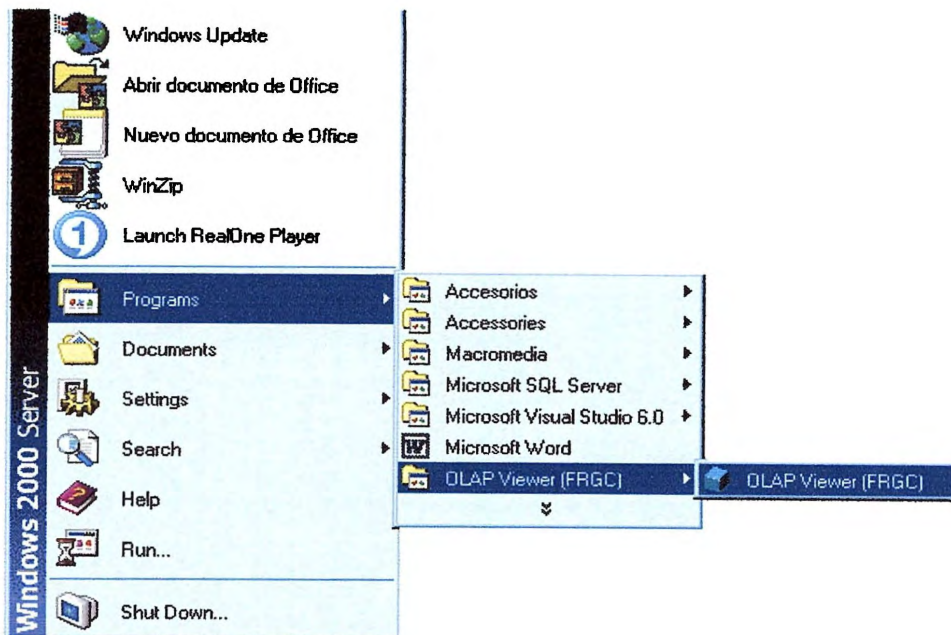
Una vez terminada la instalación de datos aparece una pantalla como la siguiente, especificando que todo se ha realizado correctamente.



**Figura 8. Instalación Paso 4**

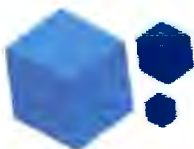
## 7. Acceso al sistema

Finalmente para acceder a OLAP Viewer hacer clic en inicio, programas, OLAP Viewer, OLAP Viewer. Recuerde que tiene que haber iniciado la cuenta en Microsoft SQL Server 2000 para que todo funcione a la perfección, en caso contrario el sistema mandara un mensaje de error y se cerrara automáticamente.



**Figura 9. Acceso al sistema**

**OLAP**



**Viewer**

version 1.0



**Manual del Usuario**



## INDICE

Introducción.....	I
1. Instalación del programa.....	1
2. Inicio de sesión.....	1
3. Menú principal.....	2
3.1 Archivo.....	3
3.2 Edición.....	4
3.3 Utilidades.....	5
3.4 Ayuda.....	10
4. Nuevo Cubo (MOLAP).....	11
4.1 Agregar Cubo.....	11
4.2 Identificación de Tabla de Hechos.....	12
4.3 Agregar medidas.....	13
4.4 Agregar dimensiones.....	14
4.5 Relacionar Tablas.....	15
4.6 Procesar Cubo.....	16
5. Crear Dimensión.....	17
5.1 Elegir Base de datos.....	17
5.2 Elegir tabla de hechos.....	18
5.3 Elegir niveles de Dimensión.....	19
6. MDX.....	20
6.1 Reportes.....	22

7.	Consulta SQL.....	23
8.	Administrador de Cubos.....	24
8.1	Barra de Herramientas.....	24
8.2	Refrescar.....	25
8.3	Explorador de Minería de datos.....	25
8.4	Grafico Dinámico de datos.....	26
8.5	Explorador Dinámico de datos.....	27
8.6	Procesar Cubo.....	28
8.7	Agregar Dimensión.....	28
8.8	Borrar Cubo.....	28
8.9	Nuevo Cubo.....	28
9.	Utilidades.....	29

#### Mantenimiento de Datos – Datamart Ventas

9.1	Por Clientes – Clientes.....	29
9.2	Por Clientes – Contacto.....	31
9.3	Por Clientes – Localización.....	33
9.4	Por Clientes – Tiempo.....	35
9.5	Por Clientes – Datos Personales.....	36
9.6	Por Vendedor y zona – Zona.....	38
9.7	Por Vendedor y zona – Vendedor.....	39
9.8	Producto.....	41
9.9	Pagos.....	42
9.10	DataWarehouse Ventas.....	44
9.11	Hechos ventas.....	46

## Mantenimiento de Datos – Datamart Producción

9.12	Por Pedido – Detalle Pedido.....	48
9.13	Por Pedido – Pedido.....	50
9.14	Por Pedido – Inventario.....	52
9.15	Por Categoría – Categoría.....	54
9.16	Por Categoría – Producto.....	56
9.17	Por Proveedor – Proveedor.....	58
9.18	Por Proveedor – Contacto.....	60
9.19	Por Proveedor – Materia Prima.....	62
9.20	Compañía de Envió.....	64
9.21	Forma de Envió.....	66
9.22	Periodo de Producción .....	68
9.23	DataWarehouse Producción .....	70
9.24	Hechos Producción.....	72

## INDICE DE IMÁGENES

### ❖ FIGURAS

Figura 1. Inicio de Sesión.....	1
Figura 2. Menú principal.....	2
Figura 3. Menú Archivo.....	3
Figura 4. Menú Edición.....	4
Figura 5. Menú Utilidades.....	5
Figura 6. Menú Utilidades – Datamart Ventas.....	5
Figura 7. Menú Utilidades – Datamart Ventas - Clientes.....	6
Figura 8. Menú Utilidades – Datamart Ventas – Vendedor y Zona.....	7
Figura 9. Menú Utilidades – Datamart Producción - Pedido.....	7
Figura 10. Menú Utilidades – Datamart Producción – Detalle Pedido.....	8
Figura 11. Menú Utilidades – Datamart Producción - Categoría.....	9
Figura 12. Menú Utilidades – Datamart Producción - Proveedor.....	9
Figura 13. Ayuda.....	10
Figura 14. Agregar Cubo.....	11
Figura 15. Identificación de Tabla de Hechos.....	12
Figura 16. Agregar medidas.....	13
Figura 17. Agregar dimensiones.....	14
Figura 18. Relacionar Tablas.....	15
Figura 19. Mensaje Procesar Cubo.....	16
Figura 20. Crear Dimensión - Elegir Base de datos.....	17
Figura 21. Crear Dimensión - Elegir tabla de hechos.....	18
Figura 22. Crear Dimensión - Elegir niveles de Dimensión.....	19
Figura 23. Mensaje Dimensión Agregada.....	19
Figura 24. Aplicación MDX.....	20
Figura 25. Reportes MDX.....	22
Figura 26. Consulta SQL.....	23
Figura 27. Administrador de Cubos.....	24

Figura 28. Selección de Minería de Datos.....	25
Figura 29. Examinador de Minería de Datos.....	26
Figura 30. Grafico Dinámico.....	26
Figura 31. Explorador Dinámico de Datos.....	27
Figura 32. Mantenimiento de Clientes.....	29
Figura 33. Búsqueda de Clientes.....	30
Figura 34. Mantenimiento de Contacto.....	31
Figura 35. Búsqueda de Contacto.....	32
Figura 36. Mantenimiento de Localización.....	33
Figura 37. Búsqueda de Localización.....	33
Figura 38. Mantenimiento de Tiempo.....	35
Figura 39. Mantenimiento Datos Personales.....	36
Figura 40. Búsqueda de Datos Personales.....	37
Figura 41. Mantenimiento de Zonas.....	38
Figura 42. Búsqueda de Zonas.....	38
Figura 43. Mantenimiento de Vendedor.....	39
Figura 44. Búsqueda de Vendedor.....	40
Figura 45. Datamart Ventas - Mantenimiento de Productos.....	41
Figura 46. Datamart Ventas - Búsqueda de Productos.....	41
Figura 47. Mantenimiento de Pagos.....	42
Figura 48. Búsqueda de Pagos.....	43
Figura 49. Mantenimiento DataWarehouse Ventas.....	44
Figura 50. Búsqueda DataWarehouse Ventas.....	45
Figura 51. Mantenimiento Hechos Ventas.....	46
Figura 52. Agregar Hechos Ventas.....	47
Figura 53. Mantenimiento de Detalle Pedido.....	48
Figura 54. Búsqueda de Detalle Pedido.....	49
Figura 55. Mantenimiento de Pedido.....	50
Figura 56. Búsqueda de Pedido.....	51
Figura 57. Mantenimiento de Inventario.....	52

Figura 58. Búsqueda de Inventario.....	53
Figura 59. Mantenimiento de Categoría.....	54
Figura 60. Búsqueda de Categoría.....	55
Figura 61. Datamart Producción - Mantenimiento de Producto.....	56
Figura 62. Datamart Producción - Búsqueda de Producto.....	57
Figura 63. Mantenimiento de Proveedor.....	58
Figura 64. Búsqueda de Proveedor.....	59
Figura 65. Mantenimiento de Contacto.....	60
Figura 66. Búsqueda de Contacto.....	61
Figura 67. Mantenimiento de Materia Prima.....	62
Figura 68. Búsqueda de Materia Prima.....	63
Figura 69. Mantenimiento de Compañía de Envió.....	64
Figura 70. Búsqueda de Compañía de Envió.....	65
Figura 71. Mantenimiento de Envió.....	66
Figura 72. Búsqueda de Envió.....	67
Figura 73. Mantenimiento de Producción.....	68
Figura 74. Búsqueda de Producción.....	69
Figura 75. Mantenimiento DataWarehouse Producción.....	70
Figura 76. Búsqueda de DataWarehouse Producción.....	71
Figura 77. Mantenimiento Hechos Producción.....	72
Figura 78. Agregar Hechos Producción.....	73

## INTRODUCCIÓN

En el presente Manual del Usuario se trata de representar las funciones básicas del Sistema **“OLAP Viewer”** , desde como instalar el programa ya que una vez este instalado el programa podemos hacer uso del Sistema para eso necesitamos iniciar sesión de usuario los cuales ya deben estar creados previamente en SQL Server 2000.

De ahí podemos utilizar todas las funciones del Menú Principal del sistema entre las cuales se encuentran:

En la opción Archivo podemos encontrar las siguientes funciones:

- ✓ Nuevo Inicio de Sesión.
- ✓ Nuevo Cubo (Molap).
- ✓ Nueva Dimensión.
- ✓ Salir

En la opción Edición se encuentran las siguientes funciones:

- ✓ Administración de Cubos.
- ✓ Consulta MDX.
- ✓ Consulta SQL.

En estas opciones el usuario puede administrar los cubos que ya existen en el Sistema, también puede realizar consultas multidimensionales o consultas SQL.

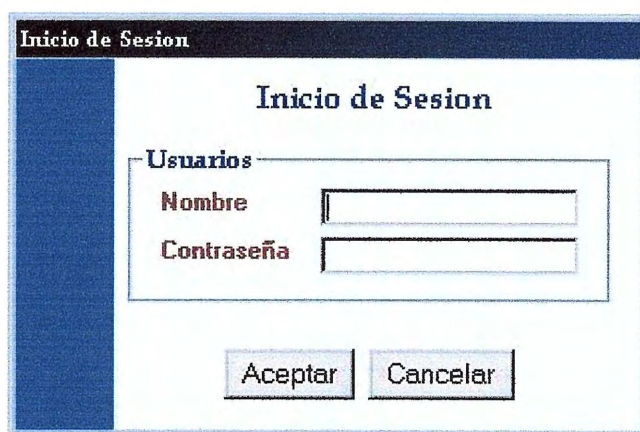
En la opción Utilidades se encuentran todos los mantenimientos para la Base de Datos Datamart Ventas y Datamart Producción que son las bases de datos utilizadas en el sistema.

El presente manual supone que el usuario conoce o posee conocimientos básicos de SQL Server 2000 para la creación de la Base de Datos, así como también conocimiento de Análisis Service y Minería de Datos para la creación de los cubos y modelos de minería de datos.

## 1. Instalación

- Dar clic en **Setup**
- Después dar clic en Aceptar
- Iniciar - Escoger directorio
- Escoger Grupo de Programas (OLAP Viewer)
- Cargar la dll y ocx y copiarlas en c: system32

## 2. Inicio de Sesión



**Figura 1. Inicio de Sesión**

Para iniciar la sesión al programa OLAP Viewer es necesario tener en consideración los siguientes aspectos:

- ✓ Insertar correctamente el login de Usuario previamente creado en SQL Server 2000.
- ✓ Los usuarios creados corresponden a Administrador, Producción y Ventas.  
Para Administrador el login y password respectivamente es: **sa, sa** .  
Para Producción es: **producción, pro**.  
Para Ventas es: **ventas, ven**.
- ✓ Para poder entrar al programa es necesario insertar una clave de usuario. En caso de que no exista el usuario o el password sea incorrecto el sistema volverá a cargar la pantalla de inicio de sesión.

Una vez introducidos de manera correcta el usuario y el password podrá acceder al sistema.

### 3. Menú Principal

OLAP Viewer esta elaborado para realizar tareas con datos multidimensionales, los procesos que lo componen están detallados a continuación.



**Figura 2. Menú Principal**

**Menú principal.** Conformado por los siguientes ítems:

- ✓ Archivo.
- ✓ Edición.
- ✓ Utilidades.
- ✓ Ayuda.

Clasificando de esta manera los procesos del sistema conseguimos agilizar el flujo de operaciones con los almacenes de datos residentes en el servidor, haciendo más eficiente la realización de procesos de análisis de información.

### 3.1 Archivo.

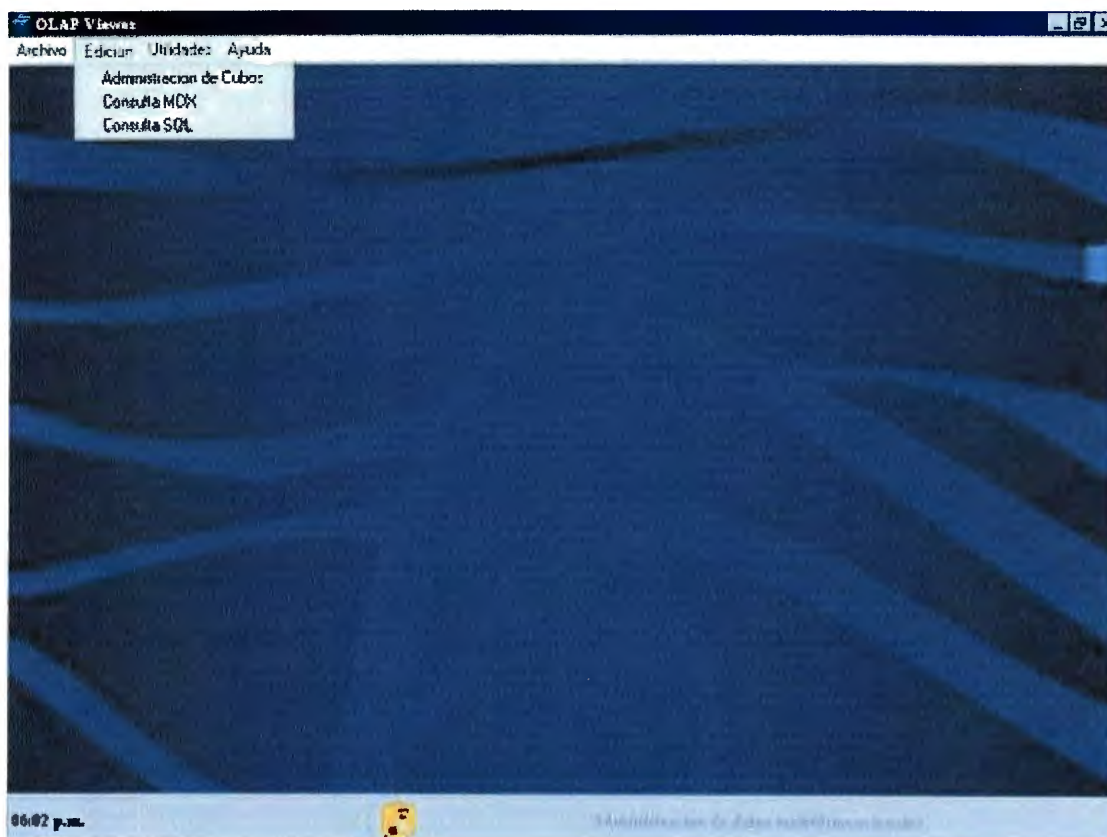


**Figura 3. Menú Archivo**

Presenta información como:

- ✓ Nuevo Inicio de Sesión.
- ✓ Nuevo Cubo (Molap).
- ✓ Nueva Dimensión.
- ✓ Salir.

## 3.2 Edición.



**Figura 3. Menú Edición**

Presenta información como:

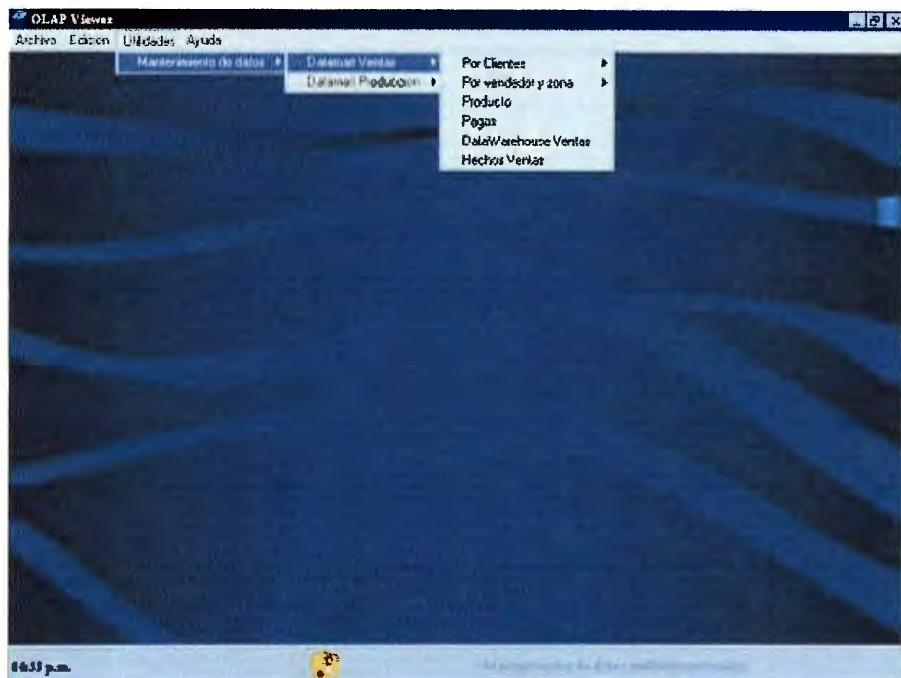
- ✓ Administración de Cubos.
- ✓ Consulta MDX.
- ✓ Consulta SQL.

### 3.3 Utilidades.



**Figura 5. Menú Utilidades**

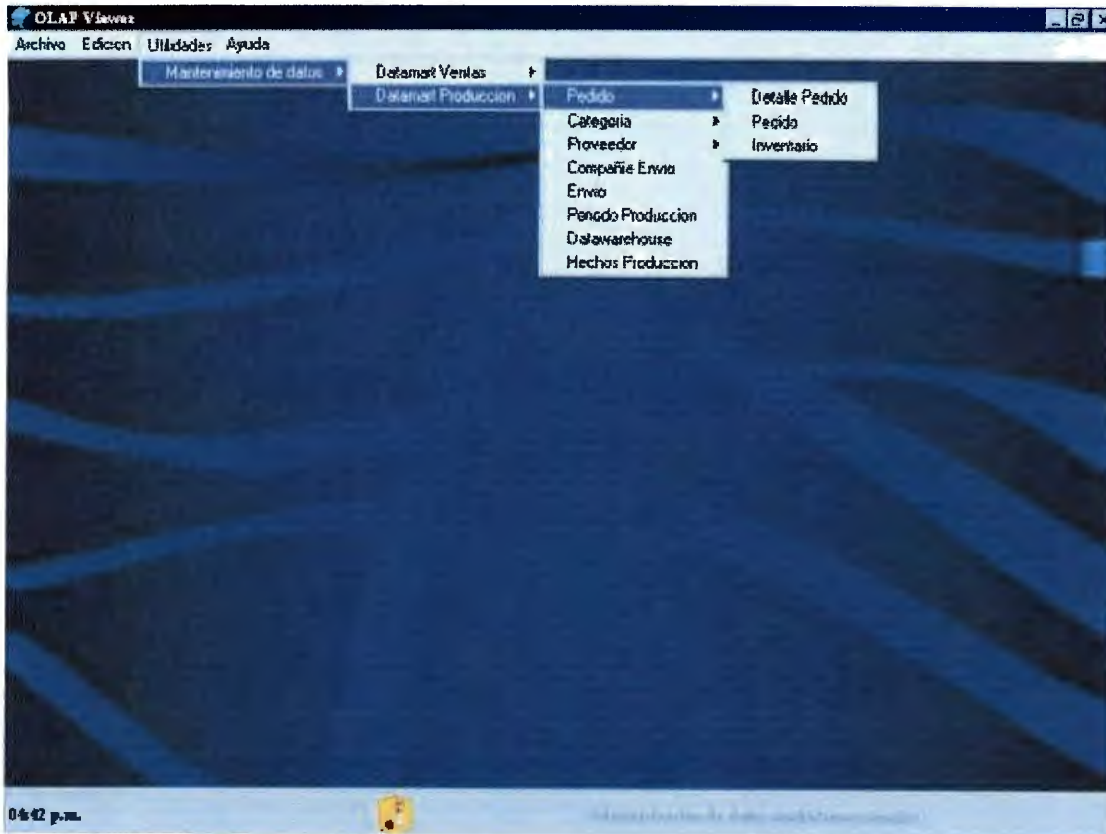
El mantenimiento de la base de datos Datamart Ventas presenta los siguientes formularios:



**Figura 6. Menú Utilidades – Datamart Ventas**

Pedido, Categoría y Proveedor también presentan una característica muy importante: Que es un registro continuo.

Para Pedido tenemos:



**Figura 10. Menú Utilidades – Datamart Producción – Pedidos – Detalle pedido**

Para Vendedor y Zona:



**Figura 8. Menú Utilidades – Datamart Ventas – Vendedor y zona**

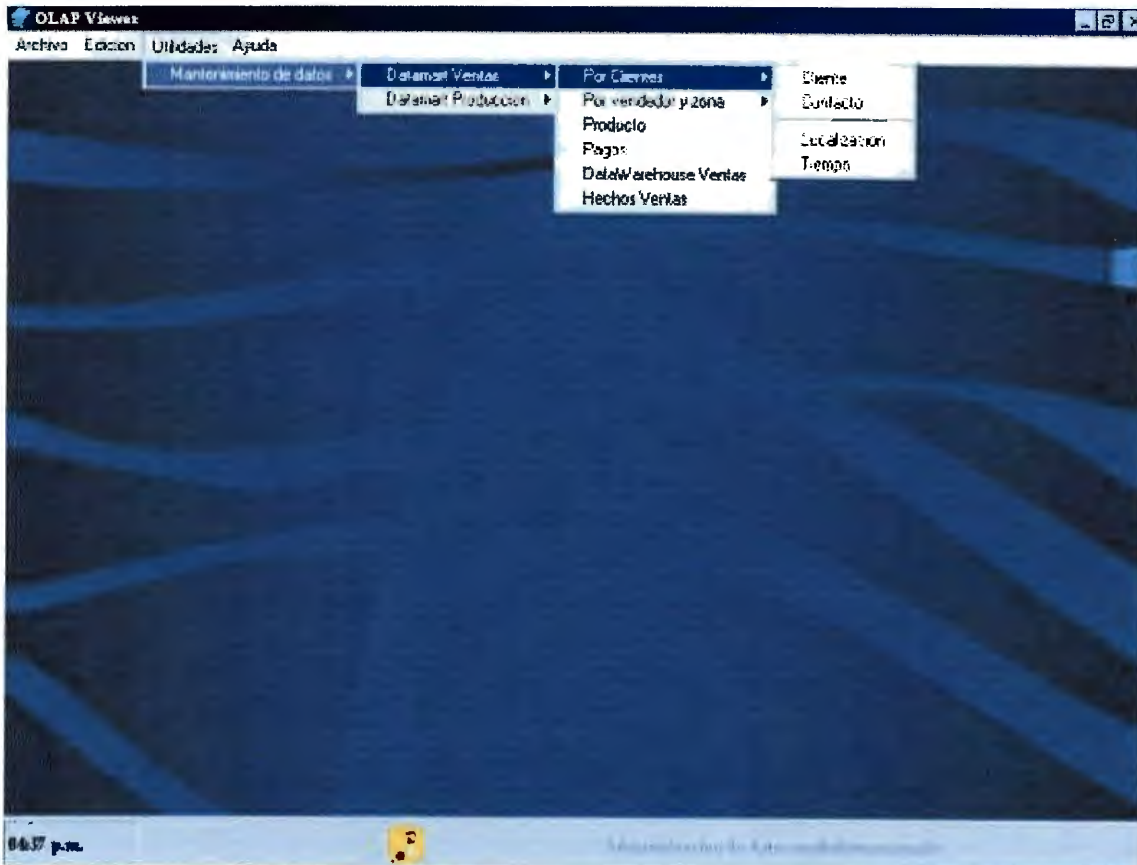
El mantenimiento de la base de datos Datamart Producción presenta los siguientes formularios:



**Figura 9. Menú Utilidades – Datamart Producción – Pedido**

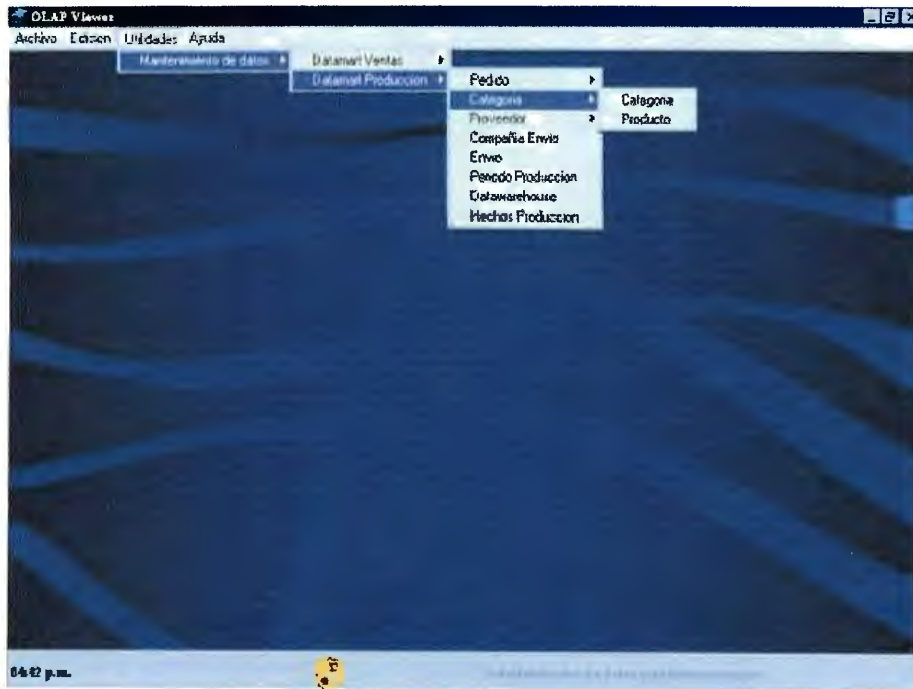
Cliente al igual que Vendedor y zona presentan una característica muy importante que es un registro continuo, es decir completar varios procesos en diferentes tablas para lograr completar un registro en específico, la razón de esto se debe a la estructura en estrella que presentan las bases de datos multidimensionales.

Para Cliente tenemos:



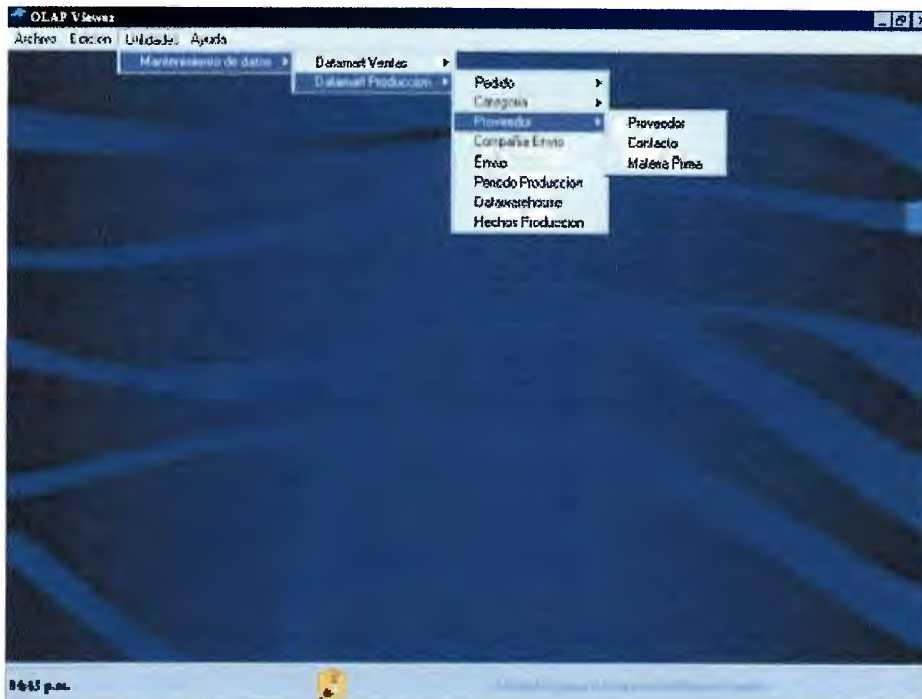
**Figura 7. Menú Utilidades – Datamart Ventas – Clientes**

Para Categoría:



**Figura 11. Menú Utilidades – Datamart Producción – Categoría**

Para Proveedor:



**Figura 12. Menú Utilidades – Datamart Producción – Proveedor**

### 3.4 Ayuda.



**Figura 13. Menú Ayuda**

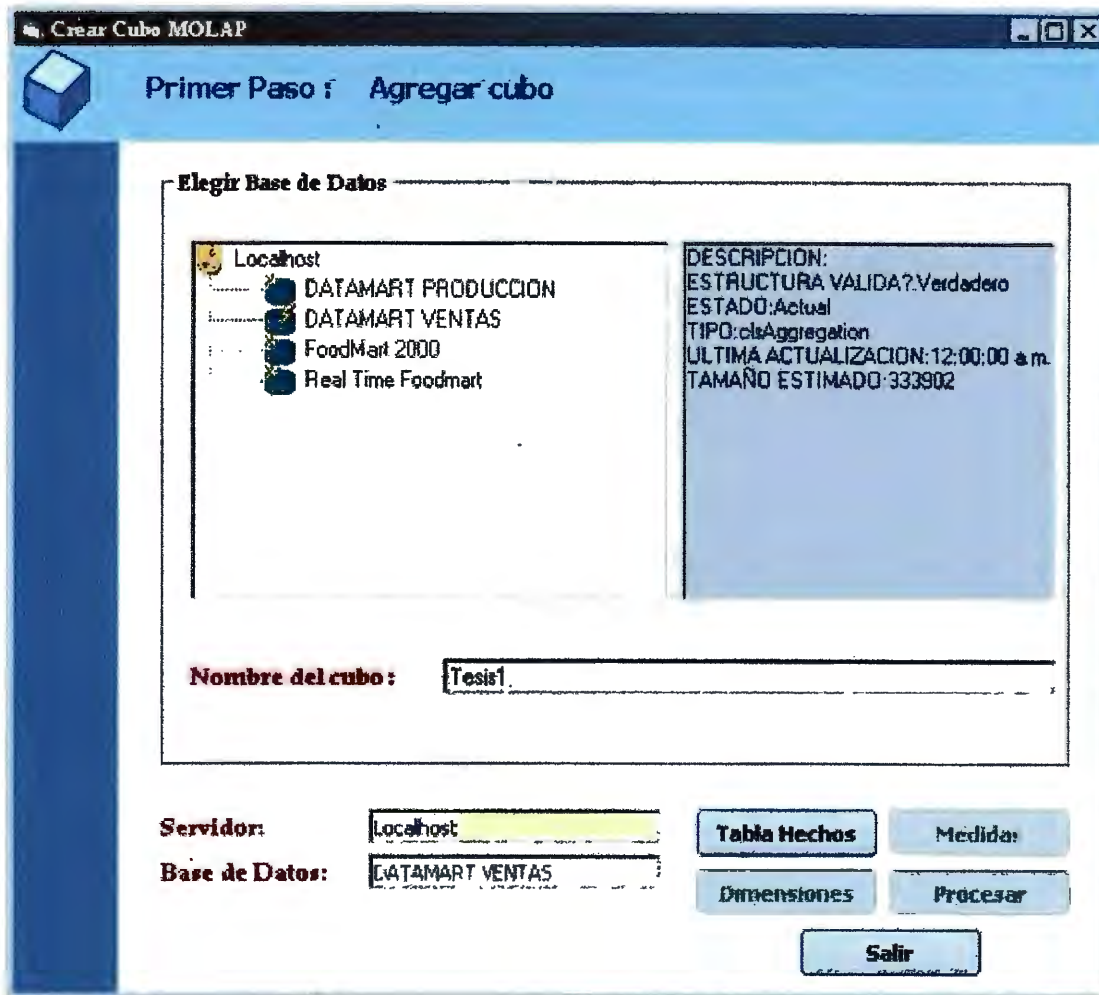
Presenta información como:

- ✓ Ayuda F1
- ✓ Acerca de

## 4. Nuevo cubo (MOLAP)

Este formulario crea un cubo multidimensional simulando un asistente, para poder lograrlo es necesario completar diversos procesos o pasos que se describen a continuación:

### 4.1 Agregar Cubo.



**Figura 14. Agregar Cubo**

Para poder agregar un cubo es necesario seguir las siguientes especificaciones:

- ✓ Doble Clic en el servidor para mostrar las Bases de Datos
- ✓ Elegir Base de Datos
- ✓ Asignar un nombre al cubo

Una vez completados estos pasos se activara el botón de Tabla Hechos.

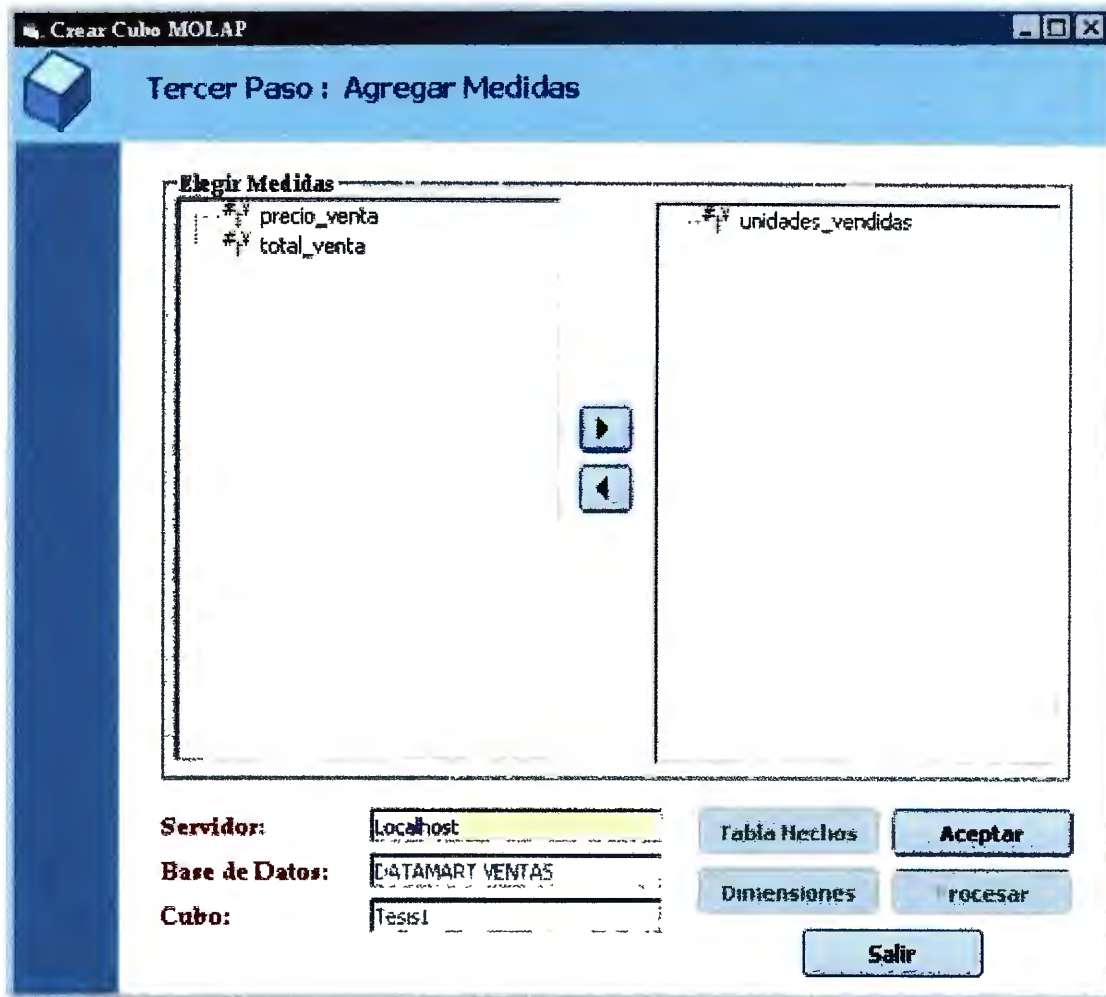
## 4.2 Identificación de Tabla Hechos.



Figura 15. Identificación Tabla de Hechos

En este paso elegimos la tabla hechos, una vez conseguido esto dar clic en Aceptar. Si todo esta bien aparecerá un mensaje emergente que nos advertirá que el cubo fue creado vacío y que se necesitara agregar medidas y dimensiones por lo cual se activara el botón Medidas, para continuar hacer clic en Medidas.

### 4.3 Agregar Medidas.



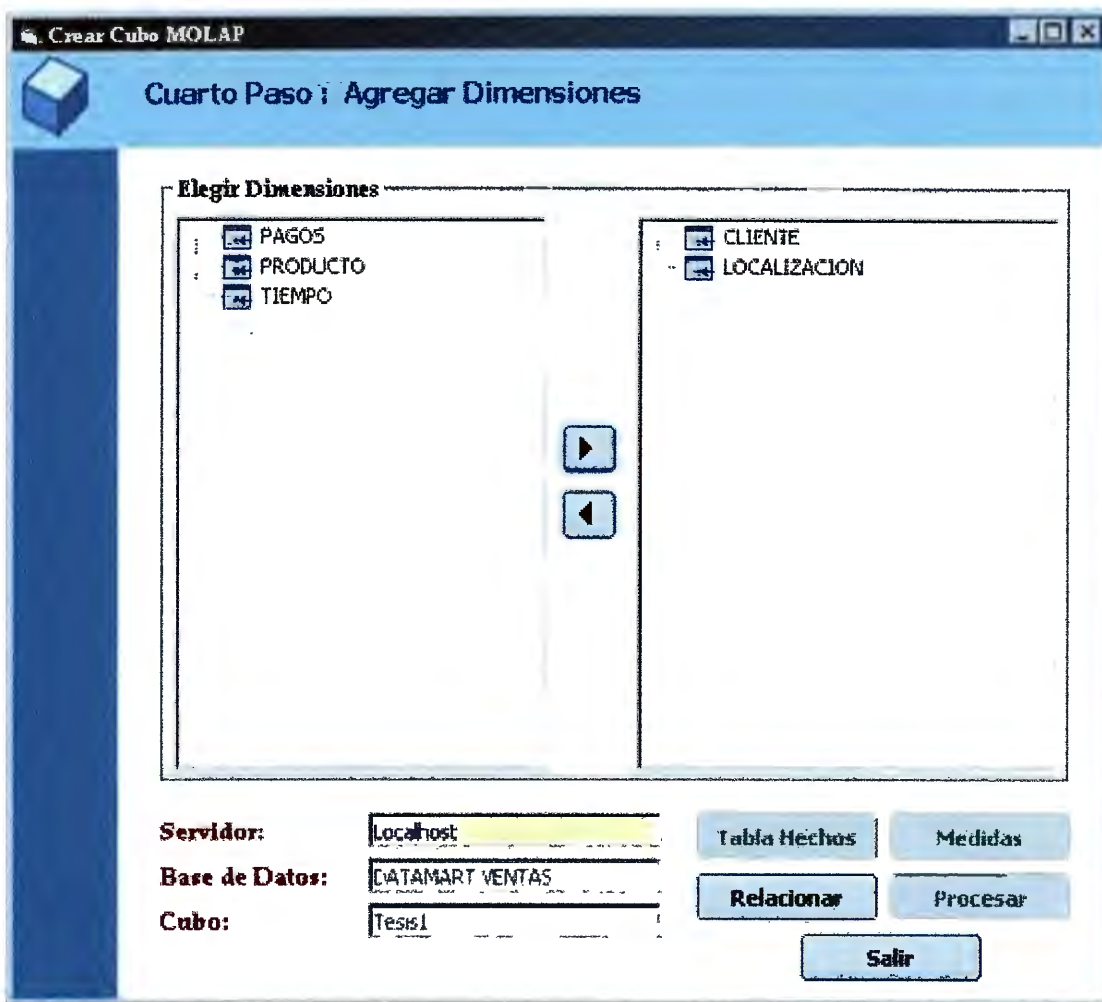
**Figura 16. Agregar Medidas**

Procederemos a elegir las medidas, recordando que las medidas son los valores numéricos que analizaremos con respecto a una o más variables denominadas dimensiones.

Una vez realizada esta operación dar clic en Aceptar, aparecerá otro mensaje advirtiéndolo que las medidas fueron agregadas y que necesitaremos agregar algunas dimensiones.

Si todo está bien se activará el botón Dimensiones al cual procederemos a dar clic.

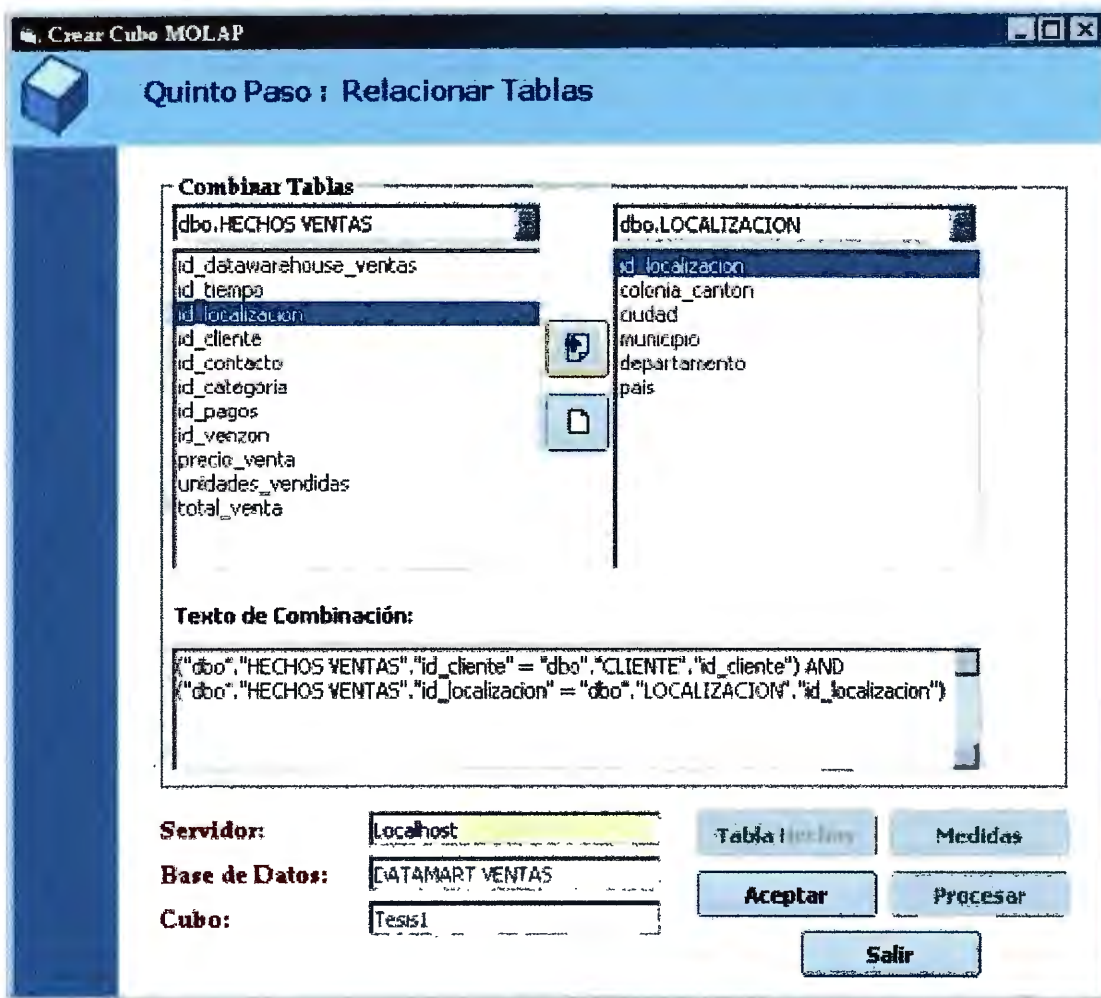
#### 4.4 Agregar Dimensiones.



**Figura 17. Agregar Dimensiones**

En este paso agregaremos las dimensiones que creamos convenientes para la creación del cubo, recordando que las dimensiones son categorías descriptivas que enriquecerán el contenido del cubo.

## 4.5 Relacionar Tablas.

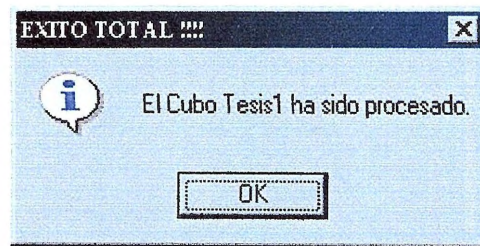


**Figura 18. Relacionar tablas**

En este paso relacionaremos las llaves de las dimensiones y la tabla de hechos de nuestra base de datos, esto se logra posicionando el cursor en los cuadros de lista, eligiendo los índices o llaves y finalmente haciendo clic en el botón agregar combinación.

Finalizado esto dar clic en Aceptar, aparecerá un mensaje advirtiéndole que el cubo fue creado con éxito y que está listo para ser procesado.

## 4.6 Procesar Cubo



**Figura 19. Mensaje Cubo Procesado**

Una vez terminado todo el procedimiento aparecerá un mensaje advirtiéndole que el cubo ha sido procesado y está listo para poder analizarlo.

Para salir del Asistente dar clic en Salir.

## 5. Crear Dimensión

Este formulario crea una dimensión simulando a un asistente, para poder lograrlo es necesario completar diversos procesos o pasos que se describen a continuación:

### 5.1 Elegir Base de Datos.

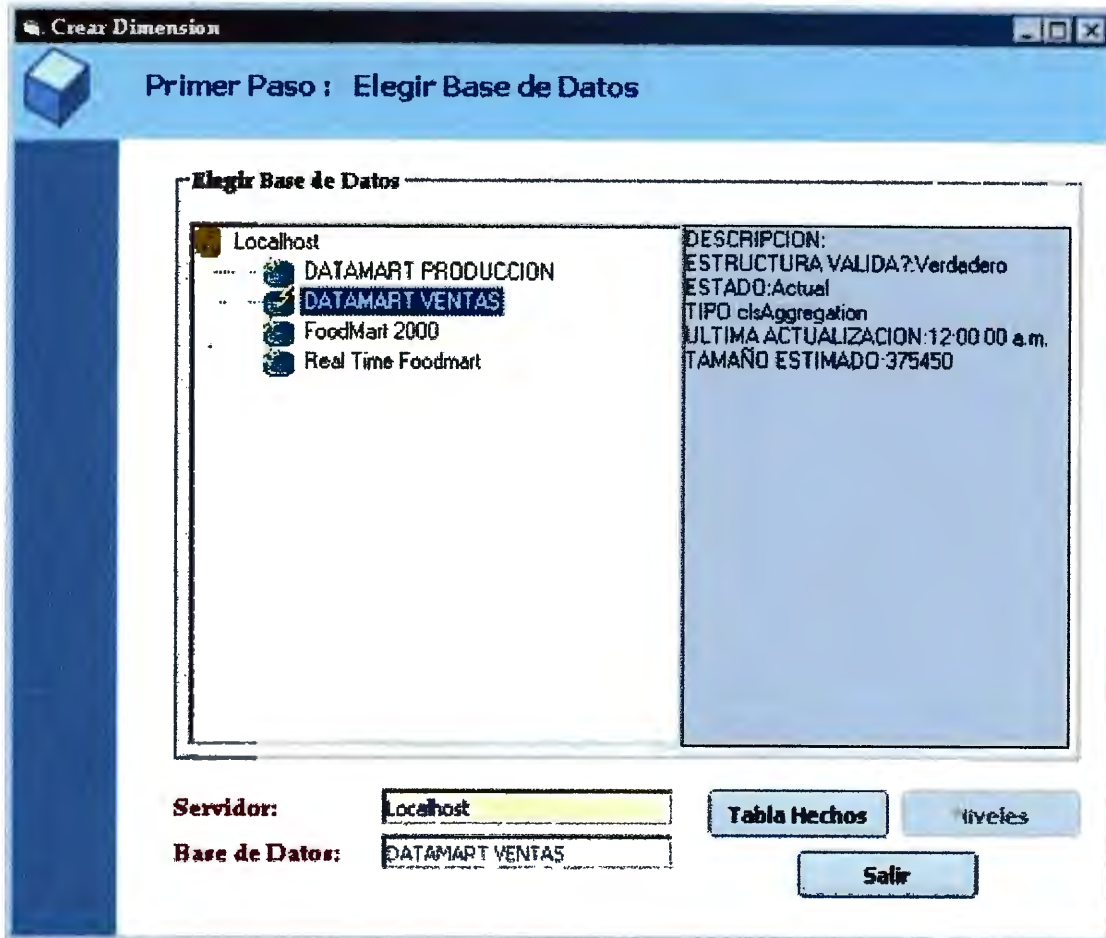


Figura 20. Crear Dimensión – Elegir Base de Datos

En este paso se elige la Base de Datos.

## 5.2 Elegir Tabla de Hechos.

En este paso identificaremos la tabla de hechos, una vez hecho esto procederemos a asignarle un nombre a la dimensión.

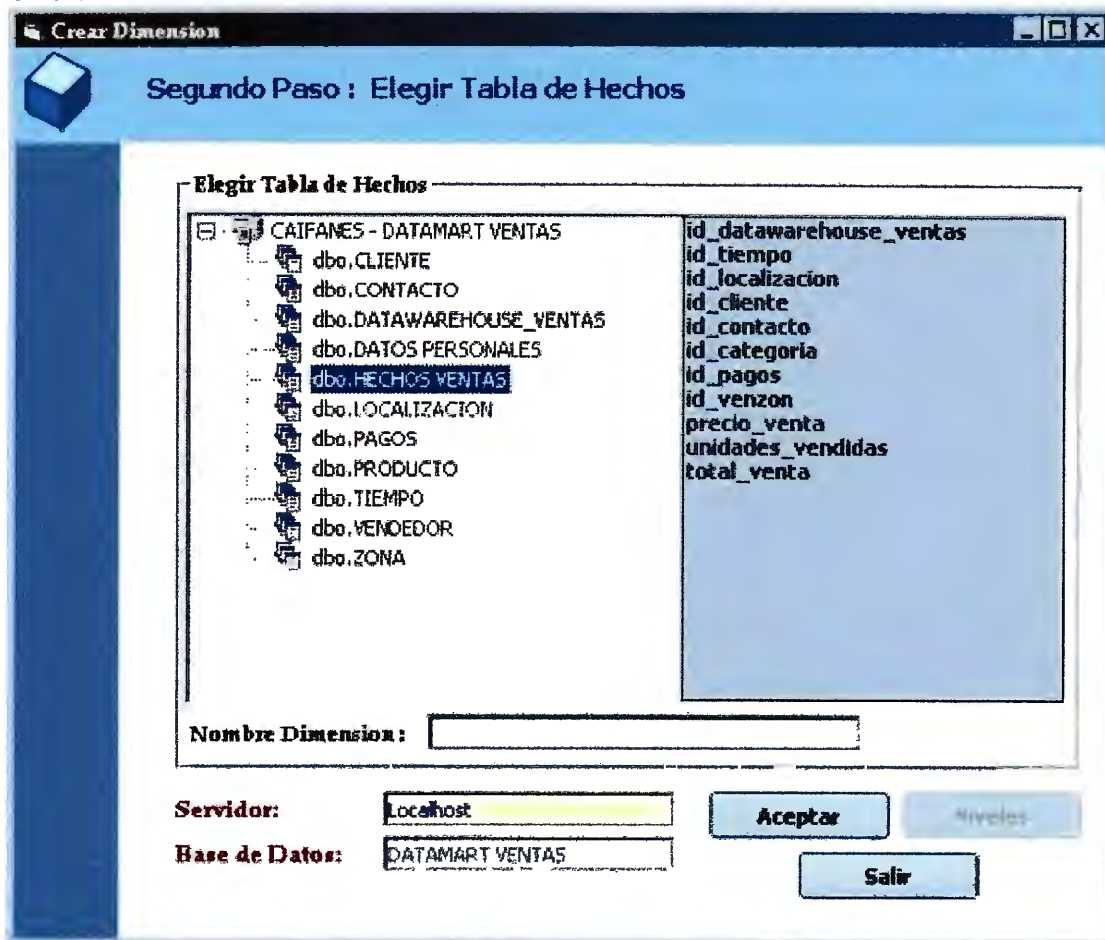
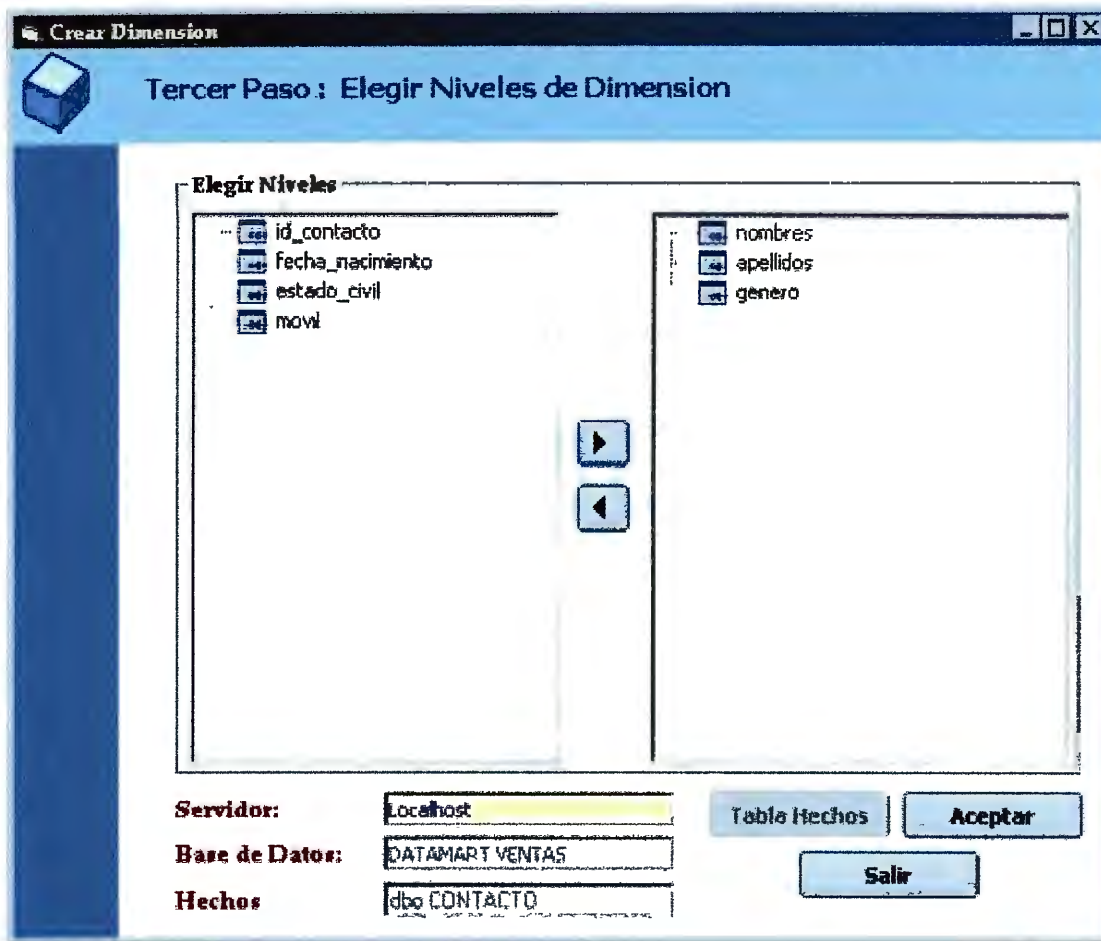


Figura 21. Crear Dimensión – Elegir Tabla de Hechos

Clic en Aceptar. Para finalizar este paso hacer clic en Niveles.

### 5.3 Elegir Niveles de Dimensión.



**Figura 22. Crear Dimensión – Elegir Niveles de Dimensión**

En este paso se elige el número de niveles que el usuario estime conveniente, luego de esto procederá a hacer clic en Aceptar

Si todo ha salido bien aparecerá un mensaje como el siguiente que nos advertirá que la dimensión ha sido creada.

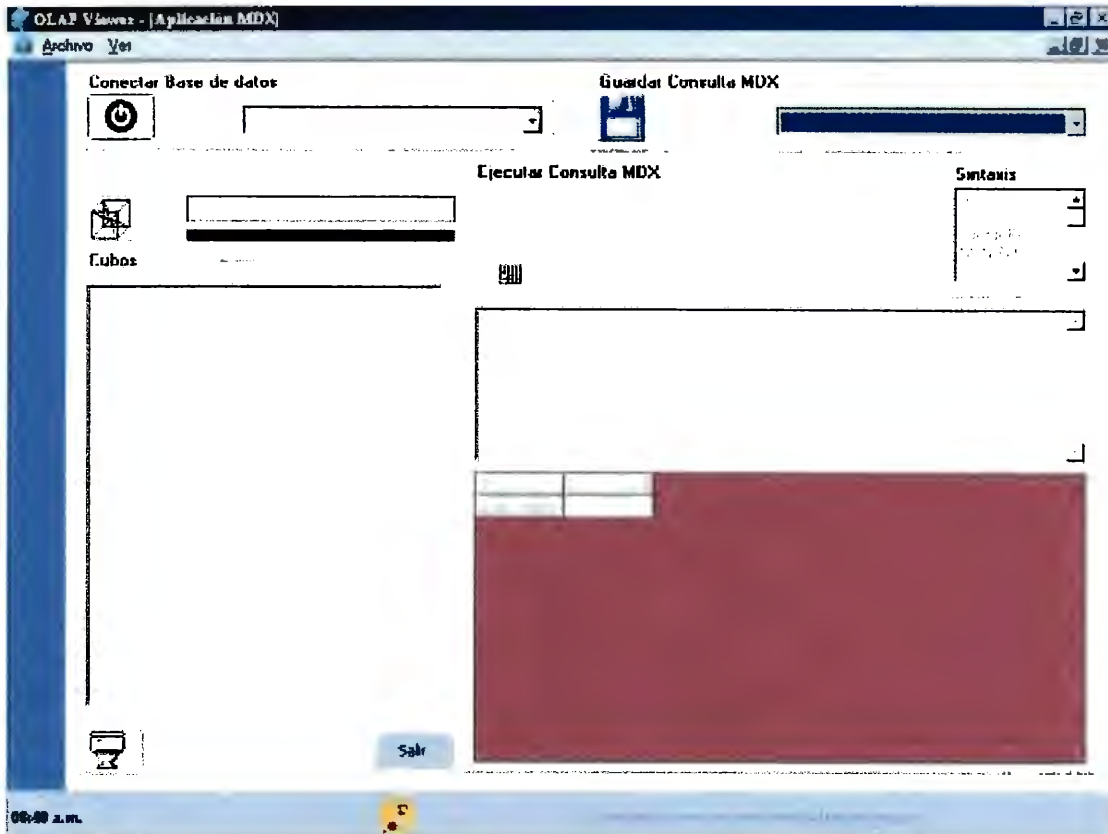


**Figura 23. Mensaje Dimensión Agregada**

Para salir del asistente dar clic en el botón Salir.

## 6. MDX


Aplicación MDX para el manejo de consultas de datos multidimensionales almacenados en cubos OLAP.



**Figura 24. Aplicación MDX**

El reporte multidimensional puede ser impreso presionando  **Print...**

### ¿Como funciona el formulario?

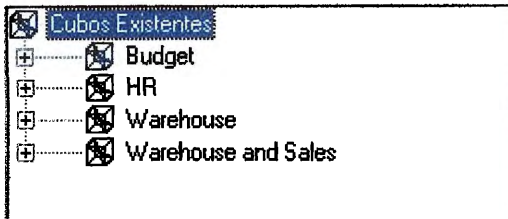
1. Conectarse al Servidor haciendo clic en el botón . Se habilitara el Combo que muestra las bases de datos existentes.
2. Elegir base de datos



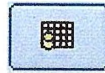
3. Ver cubos existentes haciendo clic en el botón



4. Elegir los niveles para la creación automática de la consulta



5. Ejecutar la consulta haciendo clic en el botón



Para limpiar los datos del formulario hacer clic en el botón



Para Salir hacer clic en  o en el menú Archivo-Salir.

### Consulta Temporal.

Una consulta podrá ser almacenada temporalmente haciendo clic en :



Esto permite al usuario no estar escribiendo o recuperando una consulta repetidamente.

### Abrir y Salvar consulta.

Para abrir una consulta:

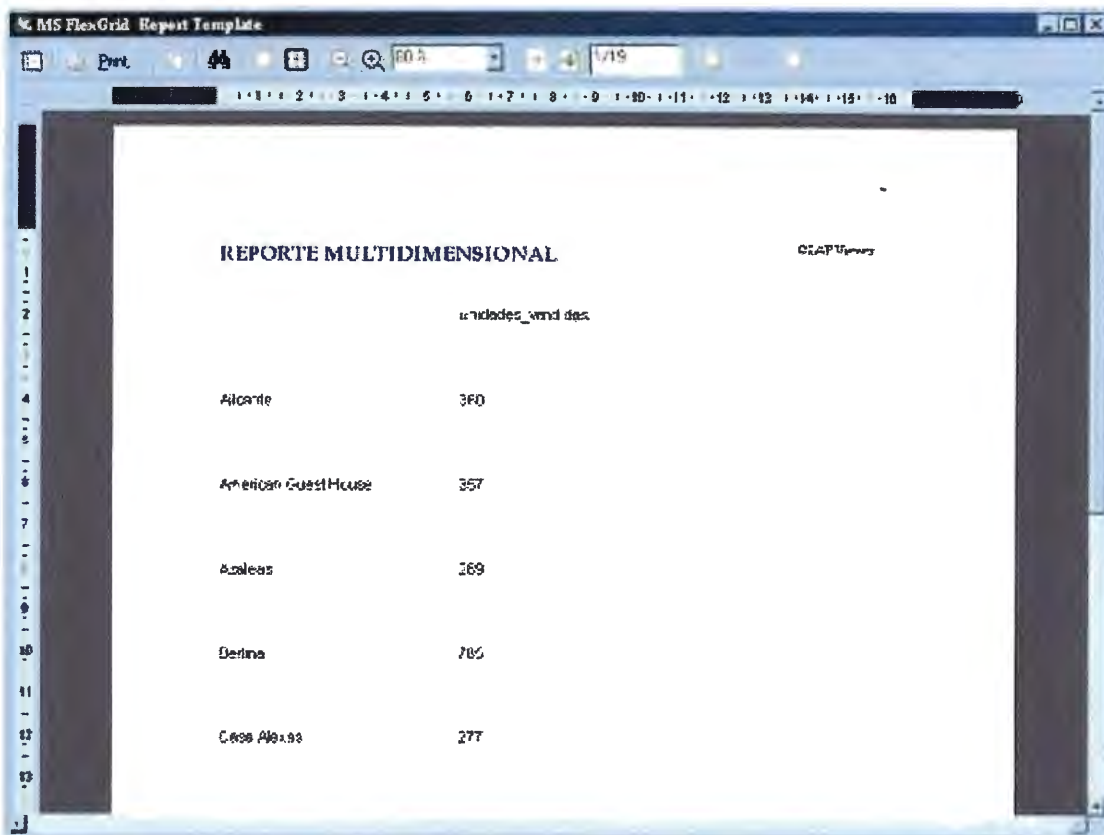
- Menú Archivo, Abrir consulta MDX. Especificar ruta.

Para guardar una consulta:

- Menú Archivo, Guardar consulta MDX. Especificar ruta.

## 6.1 Reportes.

En el menú ver existe la opción reporte, esta opción nos permite hacer un reporte dinámico y multidimensional a partir de los datos proporcionados en la consulta. Los resultados pueden ser impresos de acuerdo a los datos que se necesiten estudiar.



The screenshot shows a window titled "MS FlexGrid Report Template" displaying a multidimensional report. The report has a title "REPORTE MULTIDIMENSIONAL" and a subtitle "OLAP Viewer". The main content is a table with the following data:

	Inclodes_venta des
Alicante	360
American Guest House	357
Azeleas	269
Delina	705
Casa Alexsa	277

Figura 25. Reporte MDX

## 7. Consulta SQL

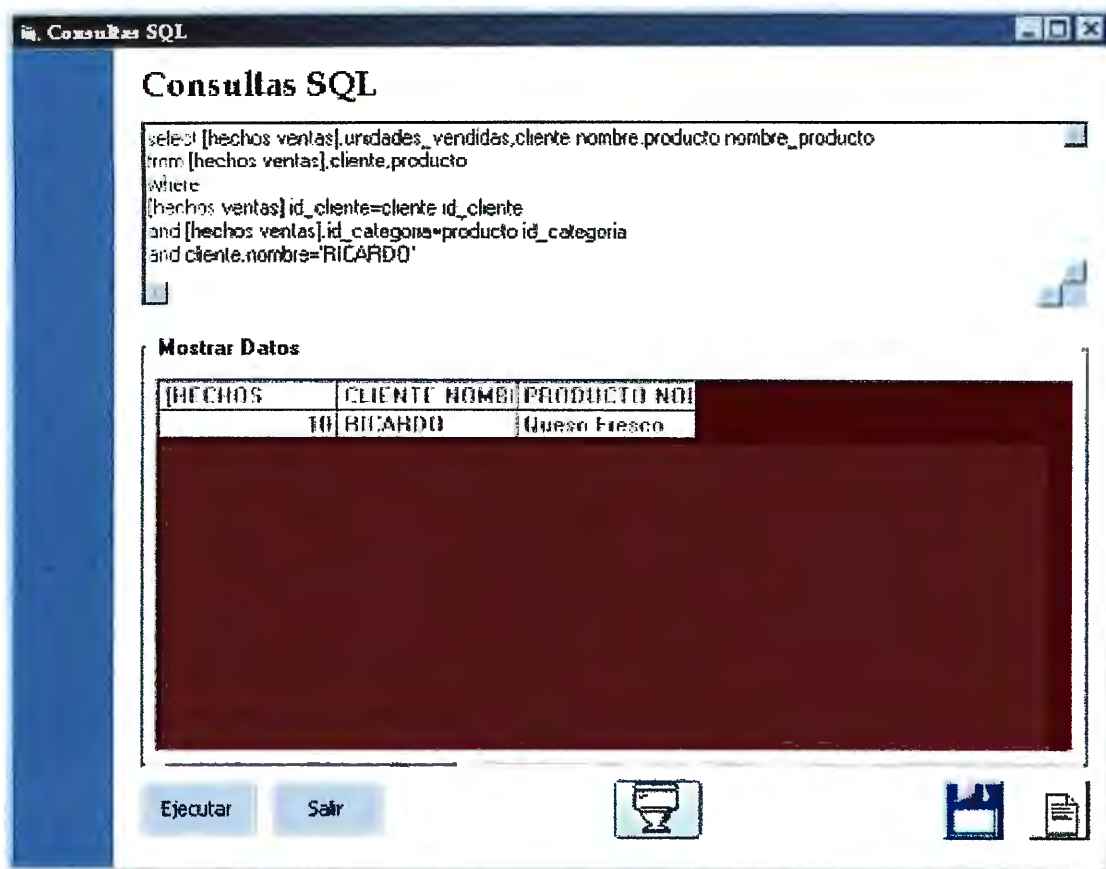


Figura 26. Consulta SQL

Este formulario permite manejar consultas SQL.

### Manejo:

Para abrir una consulta hacer clic en el botón Recuperar consulta.

Para guardar consulta clic en botón Guardar consulta.

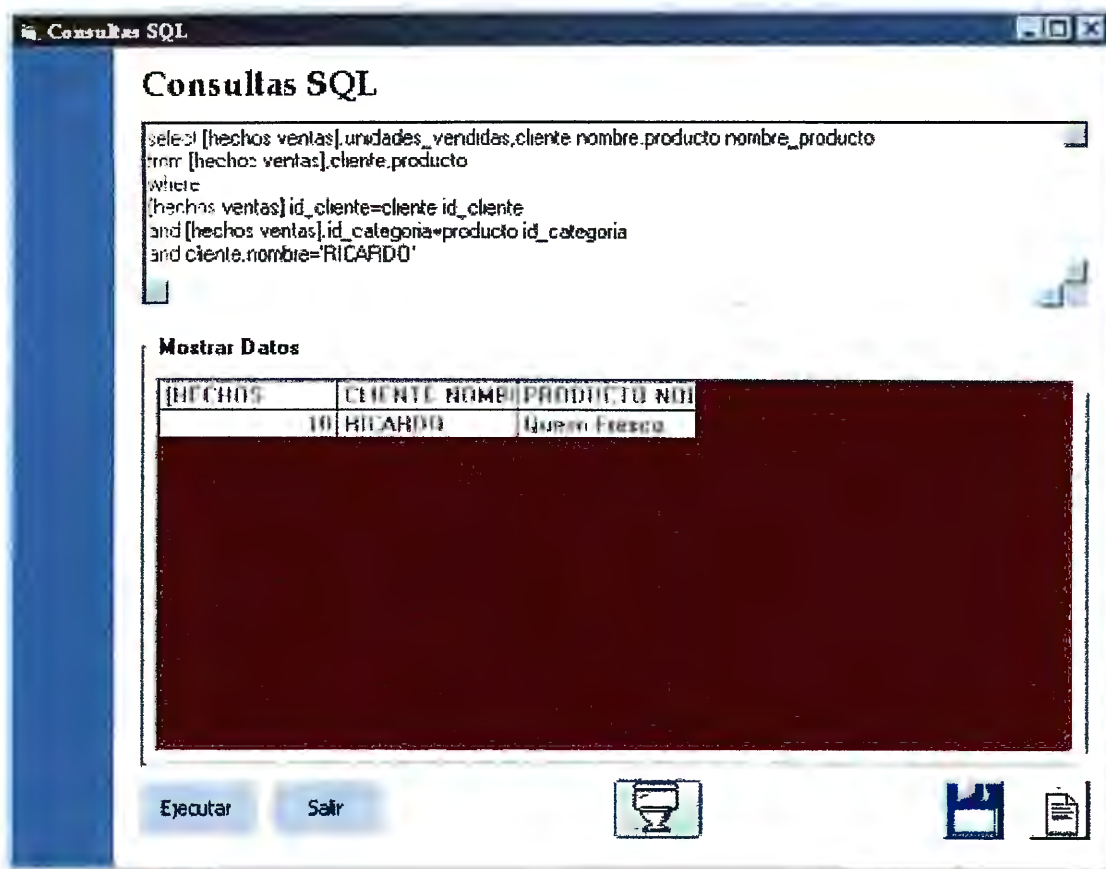
Para ejecutar hacer clic en Ejecutar.

Para Salir clic en Salir.

Para limpiar hacer clic en Limpiar datos.

En caso de error por favor verifique la sintaxis de la consulta.

## 7. Consulta SQL



**Figura 26. Consulta SQL**

Este formulario permite manejar consultas SQL.

### Manejo:

Para abrir una consulta hacer clic en el botón Recuperar consulta.

Para guardar consulta clic en botón Guardar consulta.

Para ejecutar hacer clic en Ejecutar.

Para Salir clic en Salir.

Para limpiar hacer clic en Limpiar datos.

En caso de error por favor verifique la sintaxis de la consulta.

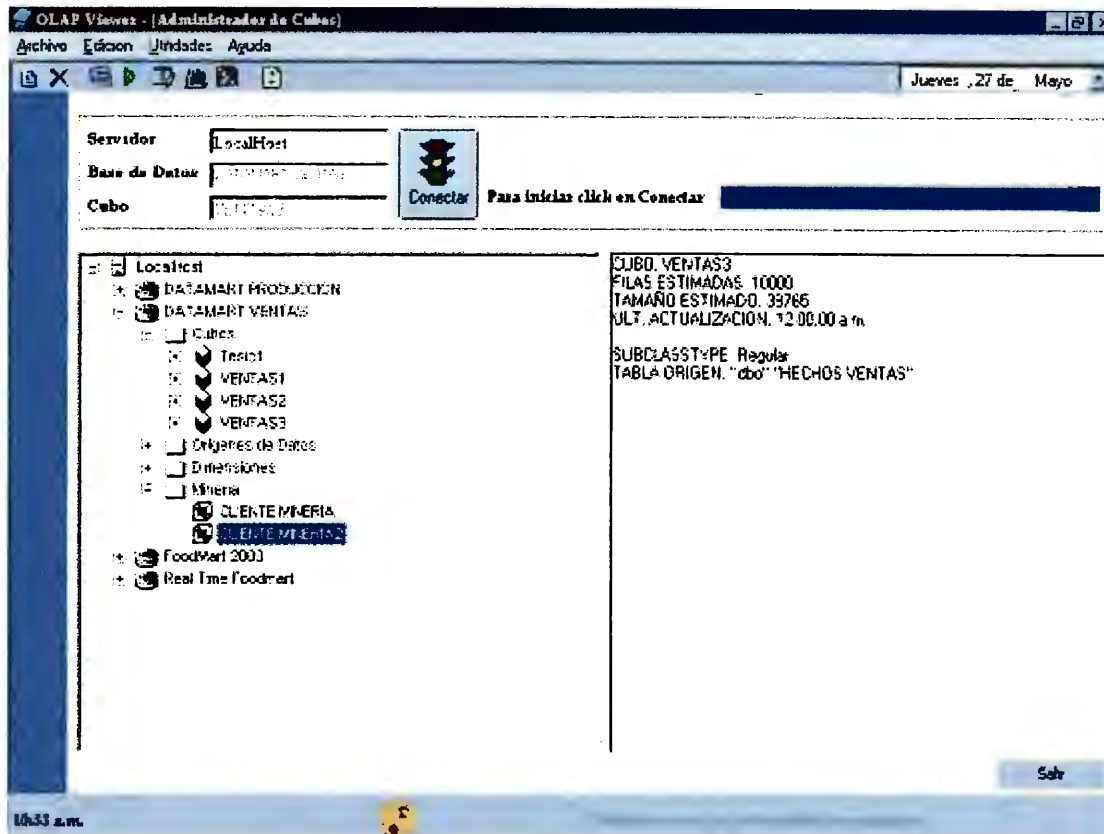
### 8.1.1 Refrescar.

Actualiza las Bases de Datos del Servidor.

### 8.1.2 Explorador de Minería de Datos.

Se utiliza para explorar cubos con información procesada por modelos y algoritmos de minería de datos.

El primer paso es elegir el modelado de Minería.



**Figura 28. Administrador de Cubos – Selección de Minería de datos**

El segundo y último paso es interactuar con el Browser de Minería de Datos. Recordando que el modelo de minería se crea en el Analysis Service de SQL Server 2000.

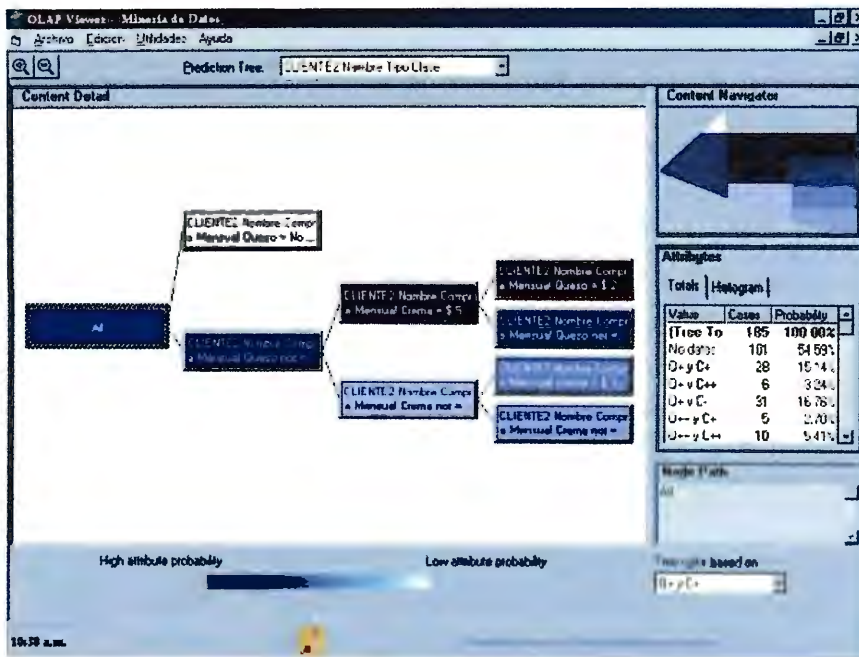
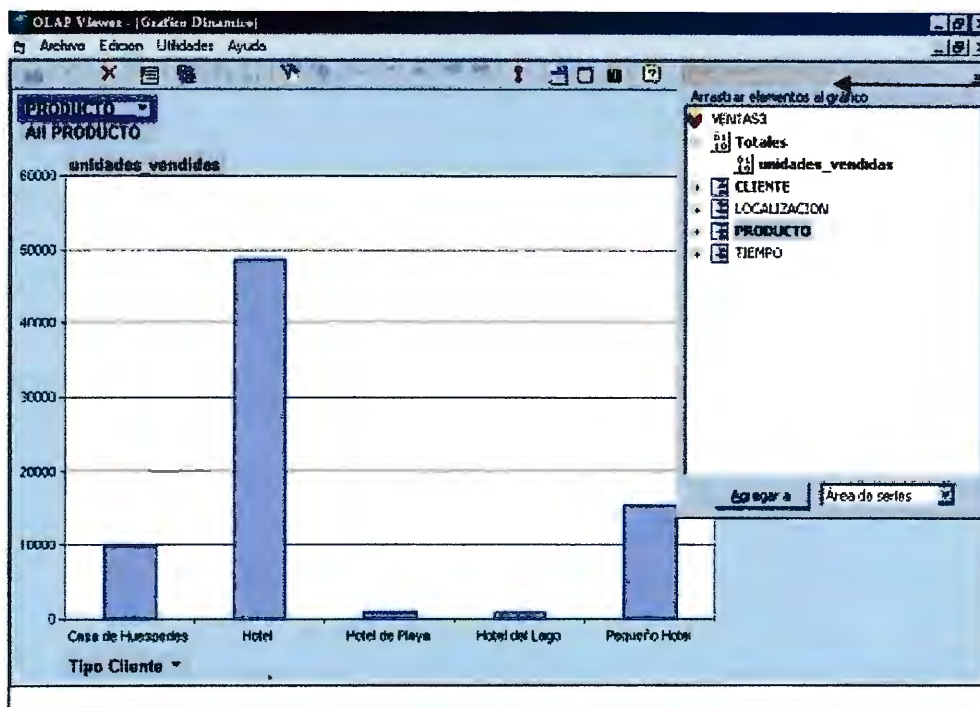


Figura 29. Examinador de Minería de datos

Para salir del formulario hacer clic en

### 8.1.3 Grafico Dinámico de Datos.

Genera un grafico a partir de los datos almacenados en un cubo. Una vez elegido el cubo tendremos el siguiente formulario.



Lista de campos

Figura 27. Grafico Dinámico

Para poder suministrar información al grafico hacer clic en el botón Lista de Campos y luego arrastrar la información al área del grafico dependiendo de las variables a analizar por un usuario.

- Para salir del formulario hacer clic en

### 8.1.4 Explorador Dinámico de Datos.

Permite interactuar con los datos almacenados en un cubo.

Una vez elegido el cubo tendremos el siguiente formulario.

		El Salvador	Total general
Presentacion	Nombre Producto	unidades	unidades
G	Jugo de Guayaba Grande	2600	2600
	Jugo de Mango Grande	2460	2460
	Jugo de Manzana Grande	1200	1200
	Jugo de Marañon Grande	1440	1440
	Total	7760	7700
M	Jugo de Guayaba Mediano	200	1200
	Jugo de Mango Mediano	1700	1700
	Jugo de Manzana Mediano	2140	2140
	Jugo de Marañon Mediano	1760	1760
	Total	6800	6800
P	Jugo de Guayaba Pequeño	200	1200
	Jugo de Mango Pequeño	740	740
	Jugo de Manzana Pequeño	1560	1560
	Jugo de Marañon Pequeño	1800	1800
	Total	5300	5300
U		56229	
Total general		76029	76029

Figura 31. Explorador Dinámico de datos

Para poder interactuar con los datos del cubo es necesario hacer clic en el botón Lista de Campos y arrastrar las dimensiones hacia el área del analizador de datos, proceso similar al del Grafico Dinámico.

Para salir del formulario hacer clic en

**8.1.5 Procesar Cubo.**

Procesamiento completo de un cubo.

**8.1.6 Agregar Dimensión.**

Asistente para agregar dimensiones a un cubo.

**8.1.7 Borrar Cubo.**

Borra un cubo en el servidor.

**8.1.8 Nuevo Cubo.**

Asistente para la creación de un cubo.

Para salir del Administrador de Cubos hacer clic en

Salir

## 9. Utilidades

### DATAMART VENTAS

#### 9.1 Mantenimiento de Datos – Datamart Ventas – Por Clientes - Cliente.

Mantenimiento de la base de datos Datamart ventas, Por Clientes es parte de un registro en línea que relaciona otras tablas de la base como lo son Contacto, Localización y tiempo. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.

CLIENTE

### Mantenimiento Clientes

id\_cliente:

tipo\_cliente:

telefono:

fax:

nombre:

compra\_mensual:

numero\_sucursales:

clase:

tamaño:

compra\_mensual\_queso:

compra\_mensual\_crema:

tipo\_clase:

Buscar Agregar Modificar Eliminar

Cancelar Salir

Figura 32. Mantenimiento de Clientes

## 9. Utilidades

### DATAMART VENTAS

#### 9.1 Mantenimiento de Datos – Datamart Ventas – Por Clientes - Cliente.

Mantenimiento de la base de datos Datamart ventas, Por Clientes es parte de un registro en línea que relaciona otras tablas de la base como lo son Contacto, Localización y tiempo. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.

The screenshot shows a window titled 'CLIENTE' with a sub-header 'Mantenimiento Clientes'. The form contains the following fields and buttons:

- id\_cliente: [text input]
- tipo\_cliente: [text input]
- telefono: [text input]
- fax: [text input]
- nombre: [text input]
- compra\_mensual: [text input]
- numero\_sucursales: [text input]
- clase: [text input]
- tamaño: [text input]
- compra\_mensual\_queso: [text input]
- compra\_mensual\_crema: [text input]
- tipo\_clase: [text input]

Buttons at the bottom:

- Buscar (magnifying glass icon)
- Agregar (folder icon)
- Modificar (document with pencil icon)
- Eliminar (document with trash icon)
- Cancelar (red circle with slash icon)
- Salir (red circle with white square icon)

Figura 32. Mantenimiento de Clientes

## 9.2 Mantenimiento de Datos – Datamart Ventas – Por Clientes - Contacto.

Mantenimiento de la base de datos Datamart ventas, Por Clientes es parte de un registro en línea que relaciona otras tablas de la base como son Cliente, Localización y tiempo. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



The screenshot shows a web application window titled 'CONTACTO' with a sub-header 'Mantenimiento Contacto'. The form contains the following fields and controls:

- id\_contacto:** A text input field followed by a dropdown menu.
- nombres:** A text input field.
- apellidos:** A text input field.
- fecha\_nacimiento:** A date input field with slashes ( / / ) and a label '(mm/dd/aa)'.
- estado\_civil:** A text input field.
- genero:** A small text input field.
- movil:** A text input field.

Below the form are six buttons with icons and labels:

- Buscar:** Icon of a magnifying glass over a document.
- Agregar:** Icon of a yellow folder.
- Modificar:** Icon of a document with a red arrow pointing to it.
- Eliminar:** Icon of a document with a red 'X' over it.
- Cancelar:** Icon of a red circle with a diagonal slash.
- Salir:** Icon of a red circle with a white dot in the center.

Figura 34. Mantenimiento de Contacto

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.



The screenshot shows a web application window titled 'CONTACTO' with a sub-header 'Mantenimiento Contacto'. The form contains several input fields: 'id\_contacto' with a dropdown menu showing 'CCEN01C2', 'nombres' with 'Ana Maria', 'apellidos' with 'Escobar Soto', 'fecha\_nacimiento' with '15/06/1 72' and a date format '(mm/dd/aa)', 'estado\_civil' with 'Soltera', 'genero' with 'F', and 'movil' with '8795648'. Below the form are six buttons: 'Buscar' (with a magnifying glass icon), 'Aceptar' (with a folder icon), 'Modificar' (with a document and arrows icon), 'Eliminar' (with a document and trash icon), 'Cancelar' (with a red circle and slash icon), and 'Salir' (with a red circle and arrow icon).

**Figura 35. Búsqueda de Contacto**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

### 9.3 Mantenimiento de Datos – Datamart Ventas – Por Clientes - Localización.

Mantenimiento de la base de datos Datamart ventas, Por Clientes es parte de un registro en línea que relaciona otras tablas de la base como son Cliente, Contacto y tiempo. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.

The screenshot shows a window titled 'LOCALIZACION' with a sub-header 'Mantenimiento Localización'. On the right side, there is a label 'Ped. + mes + cli. + prod' above a dropdown menu. Below this are several input fields for the following labels: 'id\_localizacion', 'colonia\_canton', 'ciudad', 'municipio', 'departamento', and 'pais'. At the bottom of the window, there are six buttons: 'Buscar', 'Agregar', 'Modificar', 'Eliminar', 'Cancelar', and 'Salir'.

Figura 32. Mantenimiento de Localización

#### Buscar.

Utilizamos este proceso para buscar un registro en específico.

This screenshot shows the same 'Mantenimiento Localización' window, but with the 'Buscar' button highlighted with a red border. The input fields now contain the following data: 'id\_localizacion' is '101CCEN01GM', 'colonia\_canton' is 'Miramonte', 'ciudad' is 'San Salvador', 'municipio' is 'San Salvador', 'departamento' is 'San Salvador', and 'pais' is 'El Salvador'. The 'Ped. + mes + cli. + prod' dropdown menu also shows '101CCEN01GM'.

Figura 37. Búsqueda de Localización

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.4 Mantenimiento de Datos – Datamart Ventas – Por Clientes – Tiempo.

Mantenimiento de la base de datos Datamart ventas, Por Clientes es parte de un registro en línea que relaciona otras tablas de la base como son Cliente, Contacto y Localización. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.

**Figura 35. Mantenimiento de Tiempo**

### **Agregar.**

Utilizaremos este proceso para agregar registros a la base.

### **Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

### **Eliminar.**

Este proceso borrara un registro en específico.

### **Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

### **Salir.**

Salir del formulario de mantenimiento.

## 9.5 Mantenimiento de Datos – Datamart Ventas – Por vendedor y zona – Datos personales.

Mantenimiento de la base de datos datamart ventas, Por vendedor y zona es parte de un registro en línea que relaciona otras tablas de la base como son Zona y Vendedor. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.

**DATOS PERSONALES**

### Mantenimiento Datos Personales

**id\_vendedor:**

**nombres:**

**apellidos:**

**fecha\_nacimiento:**

**genero:**

**direccion:**

**telefono:**

**movil:**

**salario:**

**documento\_identidad:**

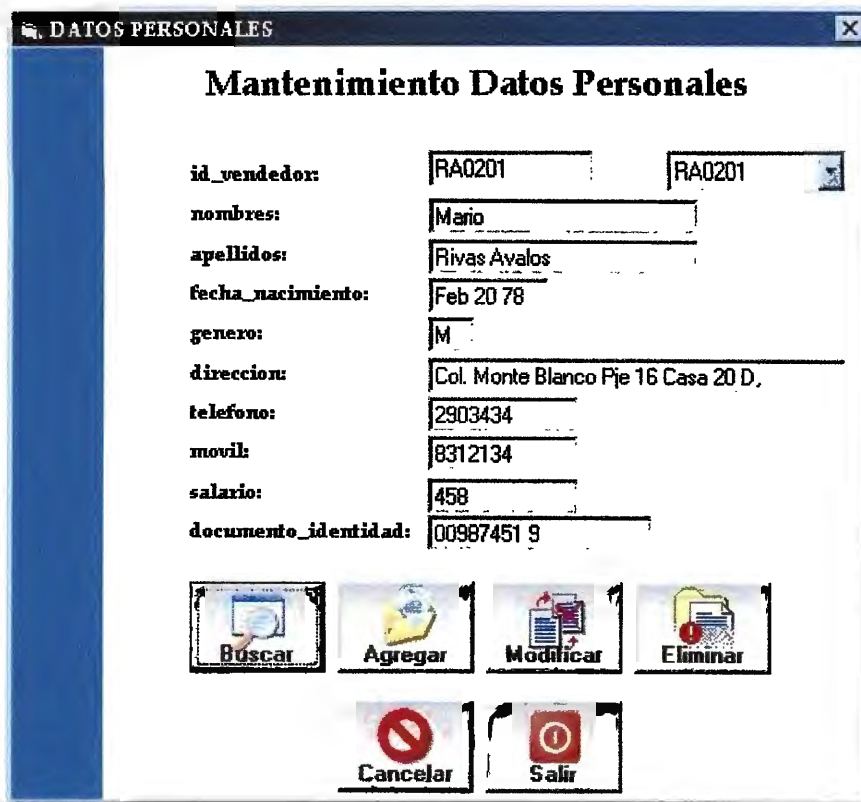
**Buscar** **Agregar** **Modificar** **Eliminar**

**Cancelar** **Salir**

Figura 32. Mantenimiento de Datos personales

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.



The screenshot shows a window titled "DATOS PERSONALES" with a sub-header "Mantenimiento Datos Personales". It contains a search form with the following fields and values:

id_vendedor:	RA0201	RA0201
nombres:	Mario	
apellidos:	Rivas Avalos	
fecha_nacimiento:	Feb 20 78	
genero:	M	
direccion:	Col. Monte Blanco Pje 16 Casa 20 D,	
telefono:	2903434	
movil:	8312134	
salario:	458	
documento_identidad:	00987451 9	

Below the form are six buttons: "Buscar" (with a magnifying glass icon), "Agregar" (with a plus icon), "Modificar" (with a document and pencil icon), "Eliminar" (with a trash can icon), "Cancelar" (with a red circle and slash icon), and "Salir" (with a red circle and arrow icon).

**Figura 40. Búsqueda de Datos personales**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.6 Mantenimiento de Datos – Datamart Ventas – Por vendedor y zona – Zona.

Mantenimiento de la base de datos Datamart ventas, Por vendedor y zona es parte de un registro en línea que relaciona otras tablas de la base como son Datos personales y Vendedor. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.

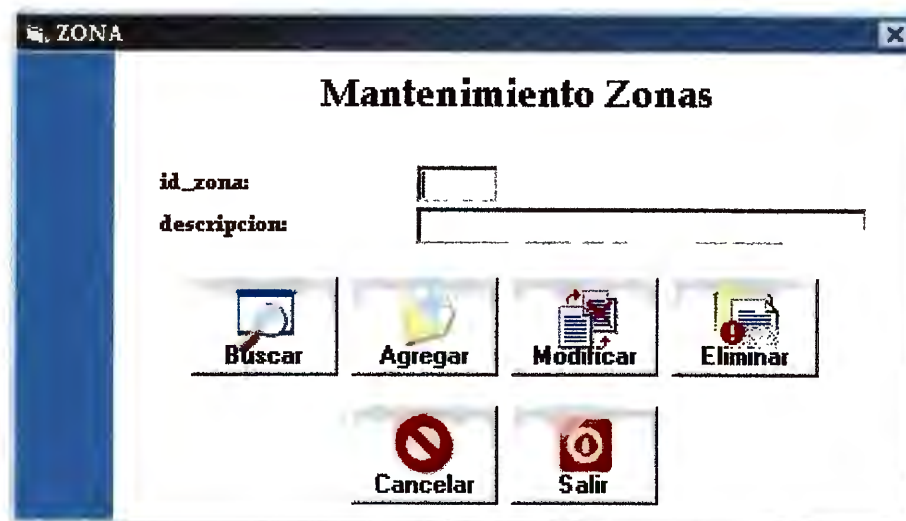


Figura 41. Mantenimiento de Zonas

### Buscar.

Utilizamos este proceso para buscar un registro en específico.



Figura 42. Búsqueda de Zonas

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

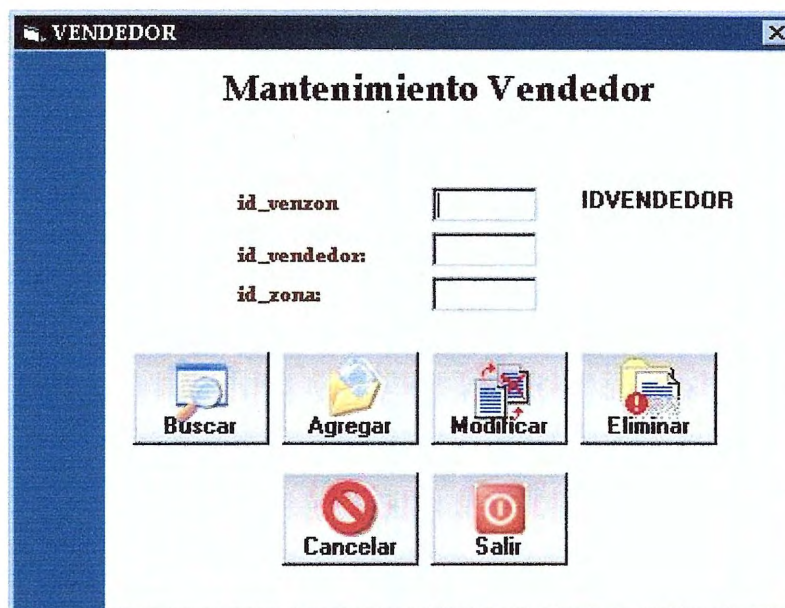
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.7 Mantenimiento de Datos – Datamart Ventas – Por vendedor y zona – Vendedor.

Mantenimiento de la base de datos Datamart ventas, Por vendedor y zona es parte de un registro en línea que relaciona otras tablas de la base como son Datos personales y Zona. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



The screenshot shows a software window titled "VENDEDOR" with a sub-header "Mantenimiento Vendedor". The window contains three input fields: "id\_venzon" (a dropdown menu), "id\_vendedor", and "id\_zona". Below these fields is a row of four buttons: "Buscar" (with a magnifying glass icon), "Agregar" (with a folder icon), "Modificar" (with a document and arrows icon), and "Eliminar" (with a trash can icon). Below this row are two more buttons: "Cancelar" (with a red circle and slash icon) and "Salir" (with a red power button icon).

**Figura 43. Mantenimiento de Vendedor**

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.



VENDEDOR

### Mantenimiento Vendedor

id\_vencion: AC0206 IDVENDEDOR

id\_vendedor: AC0206

id\_zona: OCC

Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 44. Búsqueda de vendedor**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

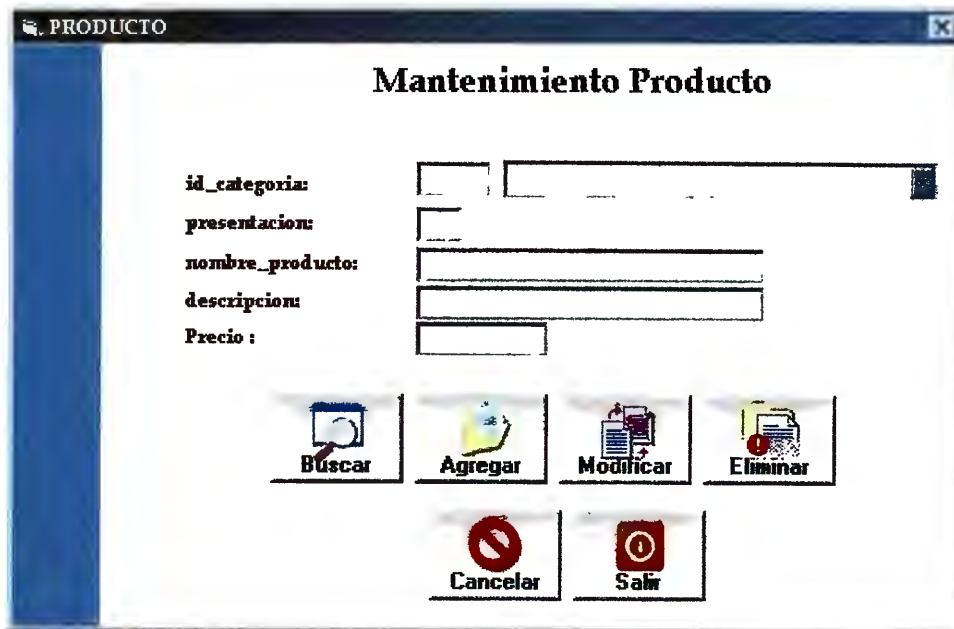
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.8 Mantenimiento de Datos – Datamart Ventas – Producto.

Mantenimiento de la base de datos Datamart ventas, se encarga de actualizar datos que tienen relación con el producto que se distribuye en el mercado.



PRODUCTO

### Mantenimiento Producto

id\_categoria: [ ] [ ]

presentacion: [ ]

nombre\_producto: [ Queso Curado ]

descripcion: [ 1 Kilogramo ]

Precio : [ 10 ]

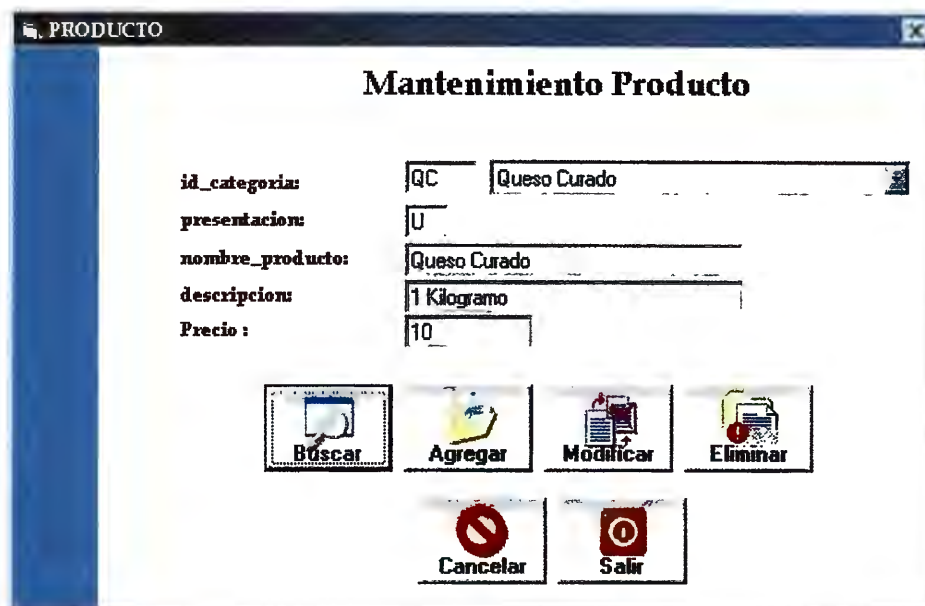
Buscar Agregar Modificar Eliminar

Cancelar Salir

Figura 41. Datamart Ventas - Mantenimiento de Producto

### Buscar.

Utilizamos este proceso para buscar un registro en específico.



PRODUCTO

### Mantenimiento Producto

id\_categoria: [ QC ] [ Queso Curado ]

presentacion: [ U ]

nombre\_producto: [ Queso Curado ]

descripcion: [ 1 Kilogramo ]

Precio : [ 10 ]

Buscar Agregar Modificar Eliminar

Cancelar Salir

Figura 46. Datamart Ventas - Búsqueda de Producto

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

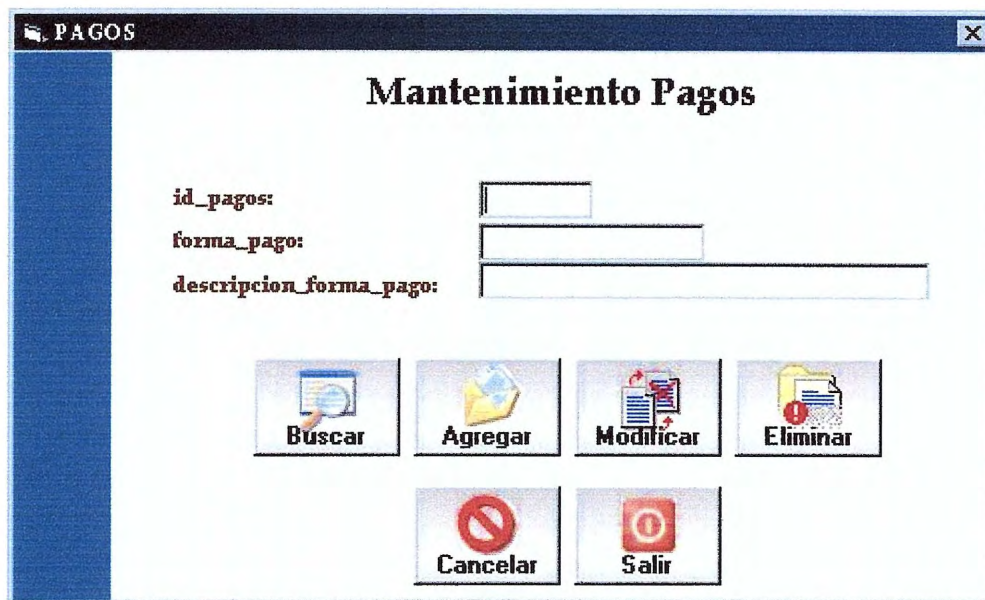
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

**9.9 Mantenimiento de Datos – Datamart Ventas – Pagos.**

Mantenimiento de la base de datos Datamart ventas, se encarga de actualizar datos que tienen relación con la forma de pago que realiza el cliente.

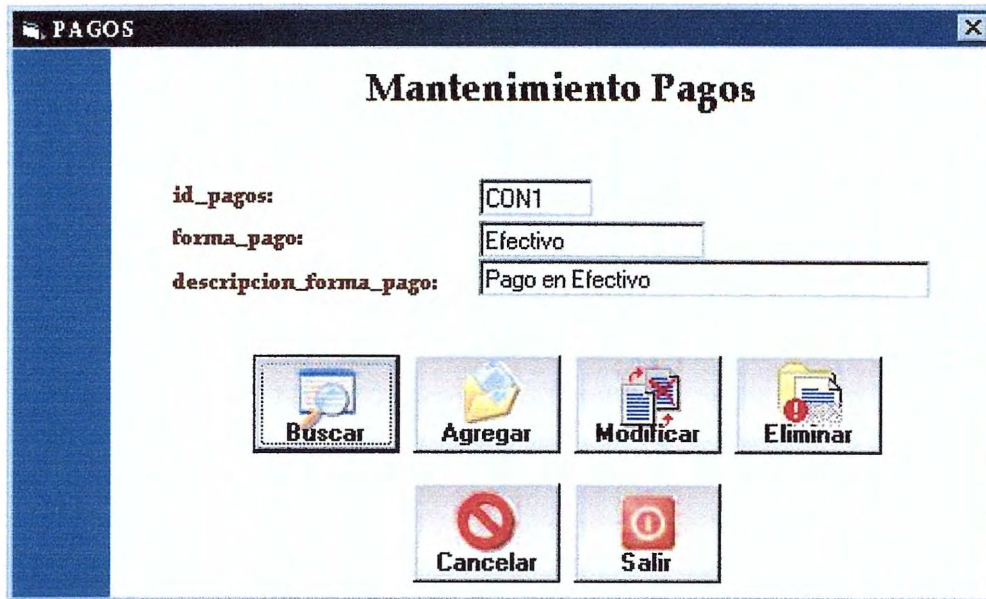


The screenshot shows a software window titled "PAGOS" with a sub-header "Mantenimiento Pagos". The window contains three text input fields labeled "id\_pagos:", "forma\_pago:", and "descripcion\_forma\_pago:". Below these fields is a row of four buttons: "Buscar" (with a magnifying glass icon), "Agregar" (with a folder icon), "Modificar" (with a document and red arrows icon), and "Eliminar" (with a document and red 'X' icon). Below this row are two more buttons: "Cancelar" (with a red circle and slash icon) and "Salir" (with a red circle and white arrow icon).

**Figura 47. Mantenimiento de Pagos**

**Buscar.**

Utilizamos este proceso para buscar un registro en específico. CON1 que representa pago en efectivo, CON2 pago en Cheque y CRE1 pago en Crédito.



The screenshot shows a window titled 'Mantenimiento Pagos' with a search form. The form contains three input fields: 'id\_pagos' with the value 'CON1', 'forma\_pago' with the value 'Efectivo', and 'descripcion\_forma\_pago' with the value 'Pago en Efectivo'. Below the form are six buttons: 'Buscar' (with a magnifying glass icon), 'Agregar' (with a folder icon), 'Modificar' (with a document and arrows icon), 'Eliminar' (with a document and trash icon), 'Cancelar' (with a red circle and slash icon), and 'Salir' (with a red circle and power icon).

**Figura 48. Búsqueda de Pagos**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.10 Mantenimiento de Datos – Datamart Ventas – DataWarehouse Ventas.

Mantenimiento de la base de datos Datamart ventas, se encarga de llevar un resumen que puede ser mensual o dependiendo del periodo que le interese estudiar al analista de ventas.



The screenshot shows a web application window titled "DATAWAREHOUSE\_VENTAS". The main heading is "Mantenimiento Data Warehouse Ventas". Below the heading, there are six input fields with the following labels: "id\_datawarehouse\_ventas:", "periodo\_venta:", "nombre\_producto:", "unidades\_vendidas\_mensu:", "precio\_producto:", and "total\_venta\_mensual:". Below the input fields, there are six buttons: "Buscar", "Agregar", "Modificar", "Eliminar", "Cancelar", and "Salir".

**Figura 49. Mantenimiento Data Warehouse Ventas**

### Buscar.

Utilizamos este proceso para buscar un registro en específico. La sintaxis del `id_datawarehouse_ventas` es: Mes + Año + Producto, Ejemplo: ENE02GGG.

**DATAWAREHOUSE\_VENTAS**

### Mantenimiento Data Warehouse Ventas

id\_datawarehouse\_ventas: ABR02CFU

periodo\_venta: Abril 2002

nombre\_producto: Crema Fresca

unidades\_vendidas\_mensu: 450

precio\_producto: 1.25

total\_venta\_mensual: 562.5

Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 50. Búsqueda de Data Warehouse Ventas**

#### **Agregar.**

Utilizaremos este proceso para agregar registros a la base.

#### **Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

#### **Eliminar.**

Este proceso borrara un registro en específico.

#### **Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

#### **Salir.**

Salir del formulario de mantenimiento.

## 9.11 Mantenimiento de Datos – Datamart Ventas – Hechos Ventas.

Mantenimiento de la base de datos Datamart ventas, es el mas importante de todos, resultado del registro en línea que como se ha explicado anteriormente es indispensable para el manejo de las ventas en la base de datos.

**Figura 50. Mantenimiento Hechos Ventas**

### Buscar.

Utilizamos este proceso para buscar un registro en específico.

### Agregar.

Utilizaremos este proceso para agregar registros a la base, para poder agregar el registro en línea es necesario dar clic en Agregar y luego completar una serie de datos para poder tener un registro completo de la venta. El único dato que hay que introducir es el de Datawarehouse ya que no esta directamente relacionado con la tabla de hechos ventas.

**HECHOS VENTAS**

**Mantenimiento Hechos\_Ventas**

id\_cliente: 1

id\_contacto: 1

id\_localizacion: 1

id\_tiempo: 1

id\_datawarehouse\_ventas: (MES+AÑO+PRO)

id\_categoria: CF Crema Especial

id\_pagos: CONT Efectivo

id\_vendedor: FM0207 Franco Mendoza

precio\_venta: 1.25

unidades\_vendidas: 4

total\_venta: 5

Aceptar

Cancelar

**Figura 52. Agregar Hechos ventas**

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

**HECHOS VENTAS**

**Mantenimiento Hechos\_Ventas**

id\_cliente: 1

id\_contacto: 1

id\_localizacion: 1

id\_tiempo: 1

id\_datawarehouse\_ventas: (MES+AÑ+PRO)

id\_categoria: CF Crema Especial

id\_pagos: CON1 Efectivo

id\_vendedor: FM0207 Franco Mendoza

precio\_venta: 125

unidades\_vendidas: 4

total\_venta: 5

Buttons: Aceptar, Cancelar

**Figura 52. Agregar Hechos ventas**

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

**DATAMART PRODUCCIÓN.****9.12 Mantenimiento de Datos – Datamart Producción – Por Pedido - Detalle Pedido.**

Mantenimiento de la base de datos Datamart producción, Por Pedido es parte de un registro en línea que relaciona otras tablas de la base como son Pedido e Inventario. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



**DETALLE PEDIDO**

### Mantenimiento Detalle Pedido

**id\_detalle\_pedido:**

**direccion:**

**ciudad:**

**pais:**

**cantidad x unidad:**

**presentacion:**

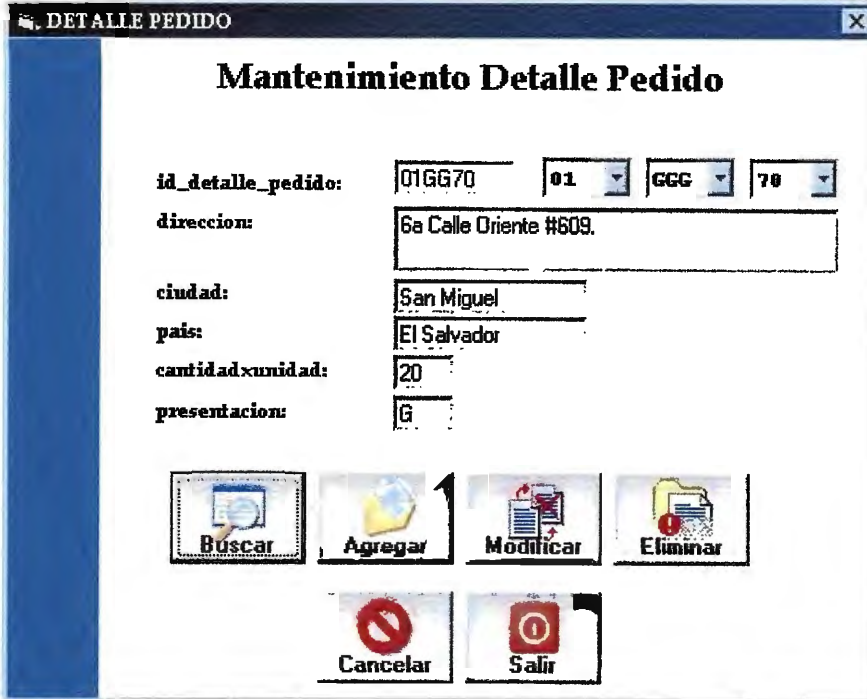
**Buscar** **Agregar** **Modificar** **Eliminar**

**Cancelar** **Salir**

**Figura 53. Mantenimiento de Detalle Pedido**

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.



The screenshot shows a window titled "DETALLE PEDIDO" with a sub-header "Mantenimiento Detalle Pedido". The form contains the following fields and values:

id_detalle_pedido:	01GG70	01	GGG	70
direccion:	6a Calle Oriente #609.			
ciudad:	San Miguel			
pais:	El Salvador			
cantidadxunidad:	20			
presentacion:	G			

Below the form are six action buttons: "Buscar" (with a magnifying glass icon), "Agregar" (with a plus icon), "Modificar" (with a pencil icon), "Eliminar" (with a trash can icon), "Cancelar" (with a red circle and slash icon), and "Salir" (with a red circle and arrow icon).

**Figura 54. Búsqueda de Detalle Pedido**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

### 9.13 Mantenimiento de Datos – Datamart Producción – Por Pedido - Pedido.

Mantenimiento de la base de datos Datamart producción, Por Pedido es parte de un registro en línea que relaciona otras tablas de la base como son Detalle Pedido e Inventario. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.

The screenshot shows a software window titled "PEDIDO" with a sub-header "Mantenimiento Pedido". The window contains the following fields and controls:

- id\_correlativo:** A text input field followed by three dropdown menus.
- id\_pedido:** A text input field.
- id\_detalle\_pedido:** A text input field followed by a dropdown menu.
- fecha\_pedido:** A date input field with the format "(mm/dd/aa)".
- fecha\_envio:** A date input field with the format "(mm/dd/aa)".
- fecha\_entrega:** A date input field with the format "(mm/dd/aa)".
- nombre\_cliente:** A text input field.

Below the input fields, there are six buttons arranged in two rows:

- Row 1: "Buscar" (with a magnifying glass icon), "Agregar" (with a folder icon), "Modificar" (with a document and arrows icon), and "Eliminar" (with a document and trash icon).
- Row 2: "Cancelar" (with a red circle and slash icon) and "Salir" (with a red circle and arrow icon).

Figura 55. Mantenimiento de Pedido

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.

The screenshot shows a window titled 'PEDIDO' with a sub-header 'Mantenimiento Pedido'. It contains several input fields for search criteria:

- id\_correlativo:** 301221, 3, 01, 221
- id\_pedido:** 47
- id\_detalle\_pedido:** 01YP47
- fecha\_pedido:** 02/01/2003 (mm/dd/aa)
- fecha\_envio:** 03/01/2003 (mm/dd/aa)
- fecha\_entrega:** 03/01/2003 (mm/dd/aa)
- nombre\_cliente:** Hotel Victoria

Below the fields are six buttons: 'Buscar' (with a magnifying glass icon), 'Agregar' (with a folder icon), 'Modificar' (with a document and red arrows icon), 'Eliminar' (with a document and red X icon), 'Cancelar' (with a red circle and slash icon), and 'Salir' (with a red circle and power icon).

**Figura 56. Búsqueda de Pedido**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**


Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.14 Mantenimiento de Datos – Datamart Producción – Por Pedido – Inventario.

Mantenimiento de la base de datos Datamart producción, Por Pedido es parte de un registro en línea que relaciona otras tablas de la base como lo son Detalle Pedido y Pedido. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



The screenshot shows a window titled "INVENTARIO" with a sub-header "Mantenimiento Inventario". The form includes the following fields and controls:

- id\_inventario:** A text input field followed by three small, empty square buttons.
- unidades\_pedidas:** A text input field.
- unidades\_enviadas:** A text input field.
- unidades\_existencia:** A text input field.
- suspendido:** A checkbox.
- produccion\_mensual:** A wide text input field.

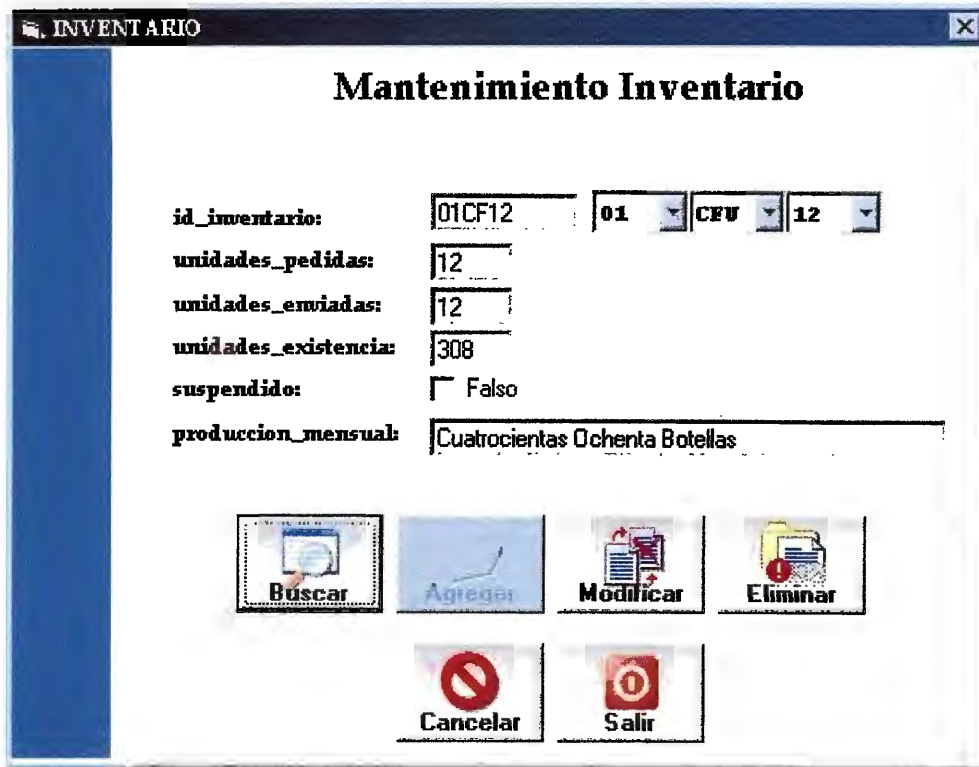
Below the form are six buttons arranged in two rows:

- Row 1: "Buscar" (with a magnifying glass icon), "Agregar" (with a plus icon), "Modificar" (with a document and arrows icon), "Eliminar" (with a document and minus icon).
- Row 2: "Cancelar" (with a red circle and slash icon), "Salir" (with a red circle and dot icon).

Figura 57. Mantenimiento de Inventario

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.



The screenshot shows a software window titled 'INVENTARIO' with a sub-header 'Mantenimiento Inventario'. It contains several input fields and buttons. The fields are: 'id\_inventario:' with a text box containing '01CF12' and three dropdown menus with values '01', 'CFW', and '12'; 'unidades\_pedidas:' with a text box containing '12'; 'unidades\_enviadas:' with a text box containing '12'; 'unidades\_existencia:' with a text box containing '308'; 'suspendido:' with a checkbox labeled 'Falso'; and 'produccion\_mensual:' with a text box containing 'Cuatrocientas Ochenta Botellas'. Below the fields are six buttons: 'Buscar' (with a magnifying glass icon), 'Agregar' (with a plus icon), 'Modificar' (with a document and arrow icon), 'Eliminar' (with a trash can icon), 'Cancelar' (with a red circle and slash icon), and 'Salir' (with a red circle and dot icon).

**Figura 58. Búsqueda de Inventario**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**


Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.15 Mantenimiento de Datos – Datamart Producción – Por Categoría – Categoría.

Mantenimiento de la base de datos Datamart producción, Por Categoría es parte de un registro en línea que relaciona otras tablas de la base como lo es Producto. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



**Mantenimiento Categoría**

id\_categoria:

nombre:

descripcion:

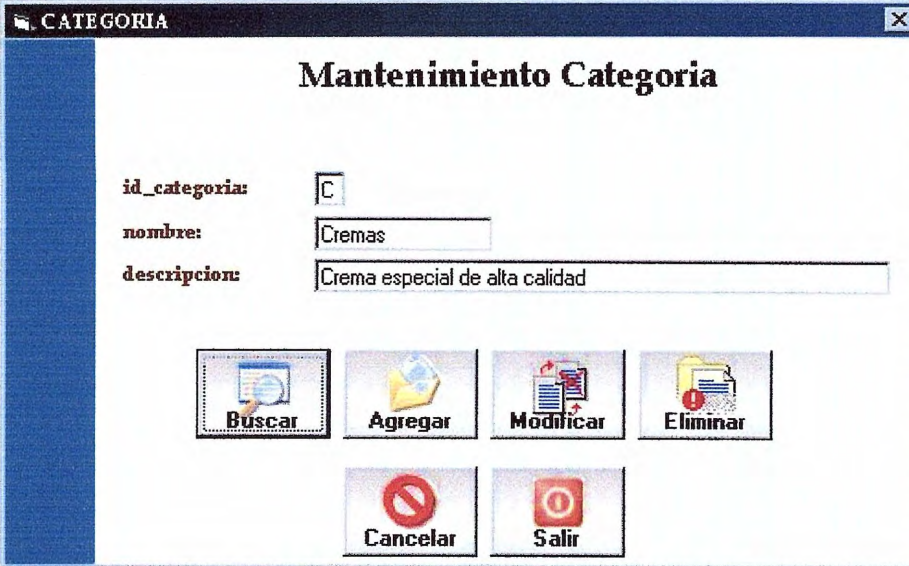
Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 53. Mantenimiento de Categoría**

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.



The screenshot shows a window titled 'CATEGORIA' with a sub-header 'Mantenimiento Categoría'. It contains three input fields: 'id\_categoria:' with the value 'C', 'nombre:' with the value 'Cremas', and 'descripcion:' with the value 'Crema especial de alta calidad'. Below the fields are six buttons: 'Buscar' (magnifying glass icon), 'Agregar' (folder icon), 'Modificar' (document with pencil icon), 'Eliminar' (document with trash icon), 'Cancelar' (red circle with slash icon), and 'Salir' (red circle with power icon).

**Figura 60. Búsqueda de categoría**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

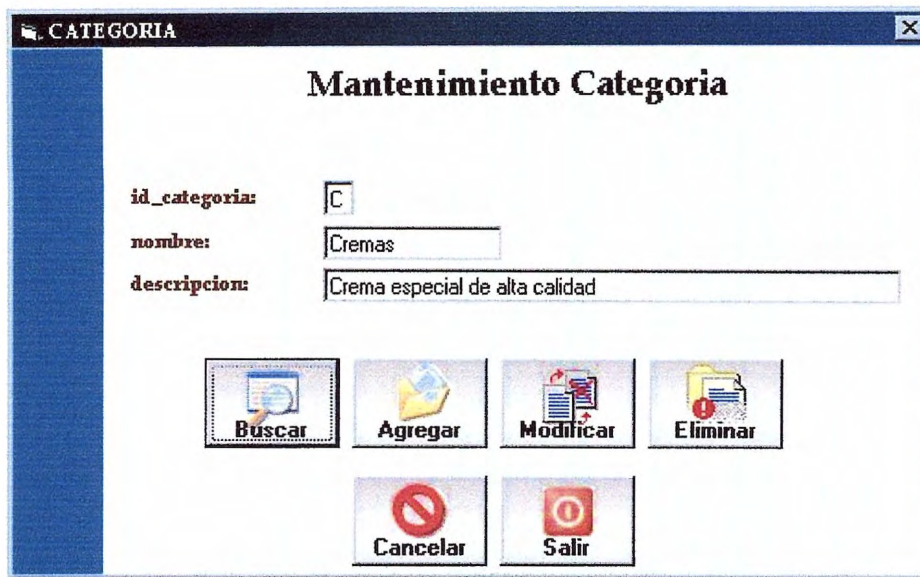
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

**Buscar.**

Utilizamos este proceso para buscar un registro en específico.



The screenshot shows a window titled 'CATEGORIA' with a sub-header 'Mantenimiento Categoría'. It contains three input fields: 'id\_categoria:' with the value 'C', 'nombre:' with the value 'Cremas', and 'descripcion:' with the value 'Crema especial de alta calidad'. Below the fields are six buttons: 'Buscar' (magnifying glass icon), 'Agregar' (folder icon), 'Modificar' (document with pencil icon), 'Eliminar' (document with trash icon), 'Cancelar' (red circle with slash icon), and 'Salir' (red circle with white dot icon).

**Figura 60. Búsqueda de categoría**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.16 Mantenimiento de Datos – Datamart Producción – Por Categoría – Producto.

Mantenimiento de la base de datos Datamart producción, Por Categoría es parte de un registro en línea que relaciona otras tablas de la base como lo es Categoría. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



The screenshot shows a software window titled "PRODUCTO" with a close button in the top right corner. The main content area is titled "Mantenimiento Producto". It features a list of labels on the left, each followed by an input field:

- id\_producto: [input field]
- id\_categoria: [input field] [dropdown menu]
- id\_presentacion: [input field]
- nombre\_producto: [input field]
- descripcion: [input field]
- precio\_venta: [input field]
- costo: [input field]
- unidades\_producidas: [input field]

At the bottom of the window, there are two rows of buttons:

- Row 1: "Buscar" (magnifying glass icon), "Agregar" (plus icon), "Modificar" (document with arrows icon), "Eliminar" (trash can icon).
- Row 2: "Cancelar" (red circle with slash icon), "Salir" (red circle with white dot icon).

Figura 61. Datamart Producción - Mantenimiento de Producto

### Buscar.

Utilizamos este proceso para buscar un registro en específico.

The screenshot shows a window titled 'PRODUCTO' with a sub-header 'Mantenimiento Producto'. The form contains the following fields and values:

id_producto:	CFU
id_categoria:	C
id_presentacion:	U
nombre_producto:	Crema Especial
descripcion:	1 Botella
precio_venta:	1.25
costo:	0.85
unidades_producidas:	500

Below the form are six buttons: 'Buscar', 'Agregar', 'Modificar', 'Eliminar', 'Cancelar', and 'Salir'.

**Figura 62. Datamart Producción - Búsqueda de Producto**

### **Agregar.**

Utilizaremos este proceso para agregar registros a la base.

### **Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

### **Eliminar.**

Este proceso borrara un registro en específico.

### **Cancelar.**

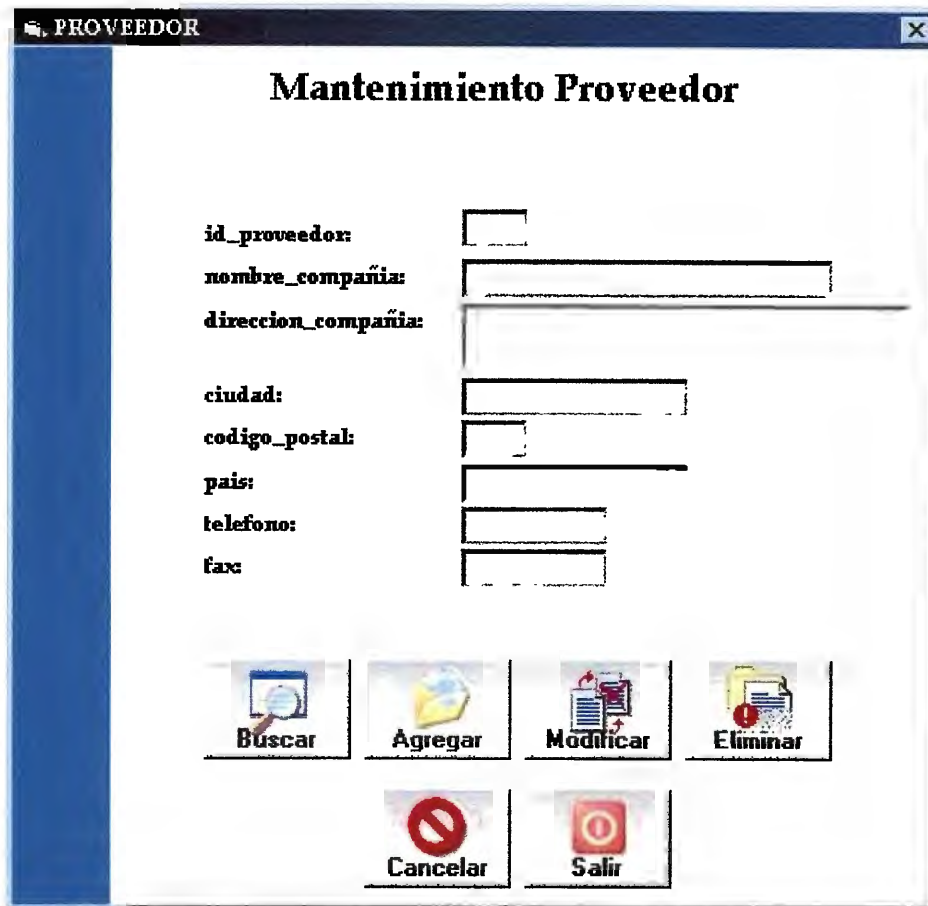
Este proceso cancela cualquier actividad dentro de este formulario.

### **Salir.**

Salir del formulario de mantenimiento.

## 9.17 Mantenimiento de Datos – Datamart Producción – Por Proveedor – Proveedor.

Mantenimiento de la base de datos Datamart producción, Por Proveedor es parte de un registro en línea que relaciona otras tablas de la base como son Contacto y Materia Prima. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



The screenshot shows a window titled "PROVEEDOR" with a sub-header "Mantenimiento Proveedor". The form contains the following fields and buttons:

- id\_proveedor:
- nombre\_compañia:
- direccion\_compañia:
- ciudad:
- codigo\_postal:
- pais:
- telefono:
- fax:

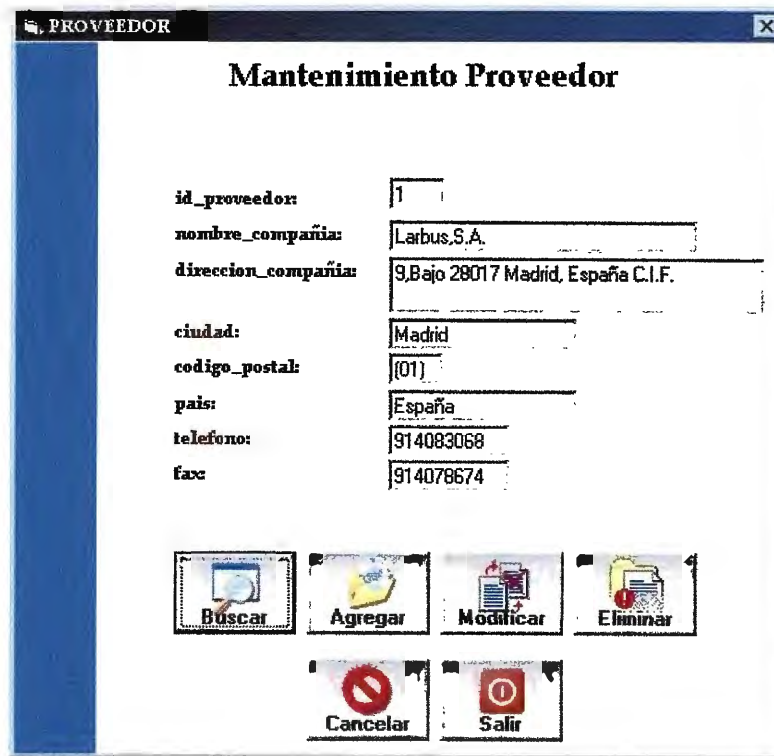
Buttons at the bottom:

- Buscar (Magnifying glass icon)
- Agregar (Folder icon)
- Modificar (Document with arrows icon)
- Eliminar (Document with trash icon)
- Cancelar (Red circle with slash icon)
- Salir (Red circle with power icon)

Figura 63. Mantenimiento de Proveedor

**Buscar.**

Utilizamos este proceso para buscar un registro en específico. El id \_ proveedor es un numero correlativo que comienza de 1.



The screenshot shows a window titled 'PROVEEDOR' with a sub-header 'Mantenimiento Proveedor'. It contains a form with the following fields and values:

id_proveedor:	1
nombre_compañia:	Larbus, S.A.
direccion_compañia:	9, Bajo 28017 Madrid, España C.I.F.
ciudad:	Madrid
codigo_postal:	(01)
pais:	España
telefono:	914083068
fax:	914078674

Below the form are six buttons: 'Buscar' (with a magnifying glass icon), 'Agregar' (with a plus icon), 'Modificar' (with a pencil icon), 'Eliminar' (with a trash can icon), 'Cancelar' (with a red circle and slash icon), and 'Salir' (with a red circle and power icon).

**Figura 64. Búsqueda de Proveedor**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

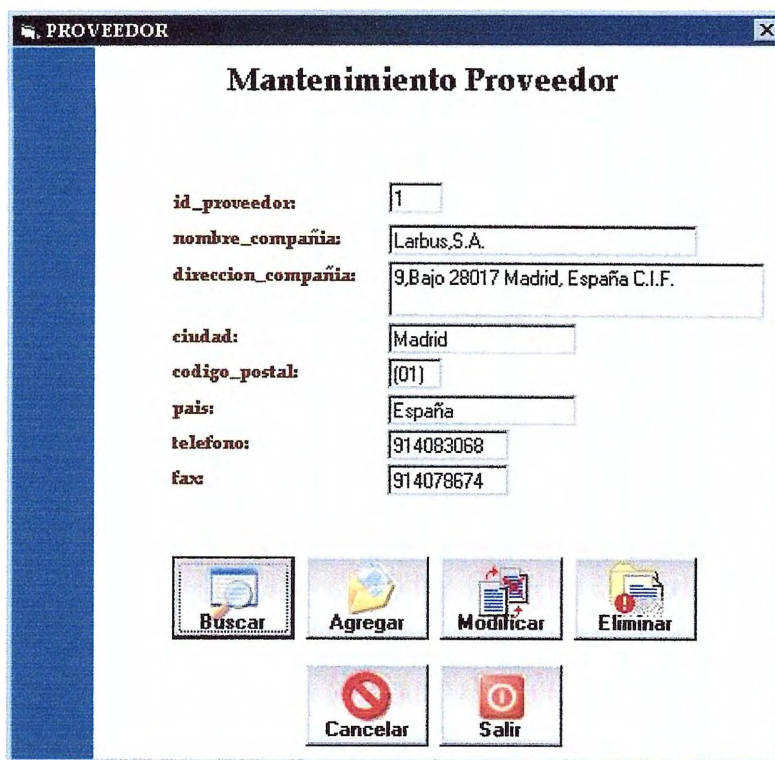
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

**Buscar.**

Utilizamos este proceso para buscar un registro en específico. El id \_ proveedor es un numero correlativo que comienza de 1.



The screenshot shows a web application window titled "PROVEEDOR" with a sub-header "Mantenimiento Proveedor". The form contains the following fields and values:

id_proveedor:	1
nombre_compañia:	Larbus,S.A.
direccion_compañia:	9,Bajo 28017 Madrid, España C.I.F.
ciudad:	Madrid
codigo_postal:	(01)
pais:	España
telefono:	914083068
fax:	914078674

Below the form are six buttons: "Buscar" (magnifying glass icon), "Agregar" (folder icon), "Modificar" (document with pencil icon), "Eliminar" (document with trash icon), "Cancelar" (red circle with slash icon), and "Salir" (red circle with power icon).

**Figura 64. Búsqueda de Proveedor**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

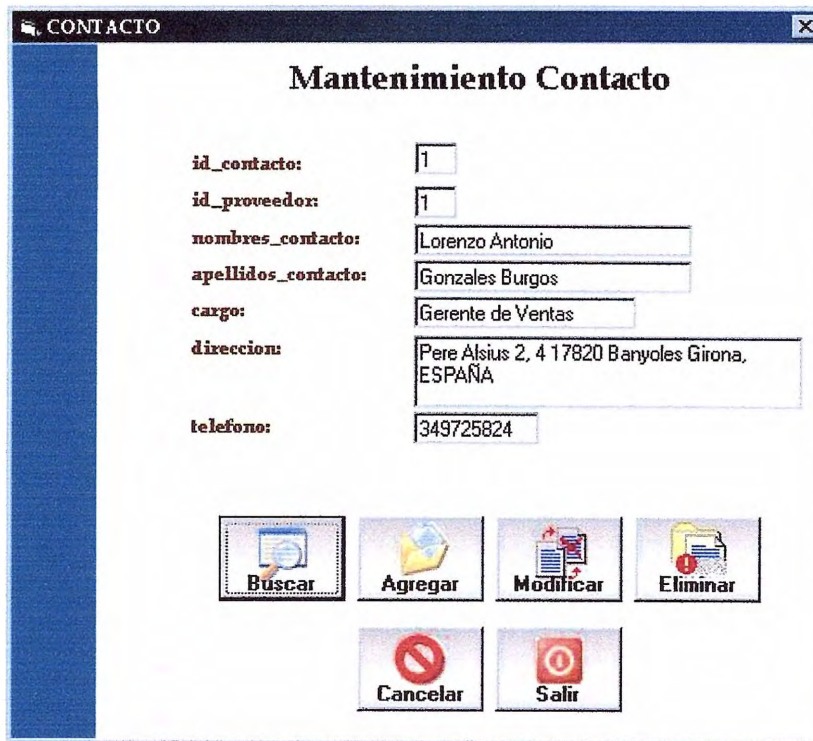
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

**Buscar.**

Utilizamos este proceso para buscar un registro en específico. El id\_contacto es un correlativo que comienza de 1.



The screenshot shows a window titled 'CONTACTO' with a sub-header 'Mantenimiento Contacto'. The form contains the following fields and values:

id_contacto:	1
id_proveedor:	1
nombres_contacto:	Lorenzo Antonio
apellidos_contacto:	Gonzales Burgos
cargo:	Gerente de Ventas
direccion:	Pere Alsius 2, 4 17820 Banyoles Girona, ESPAÑA
telefono:	349725824

Below the form are six buttons: 'Buscar' (with a magnifying glass icon), 'Agregar' (with a folder icon), 'Modificar' (with a document and arrows icon), 'Eliminar' (with a trash can icon), 'Cancelar' (with a red circle and slash icon), and 'Salir' (with a red circle and power icon).

**Figura 66. Búsqueda de Contacto**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

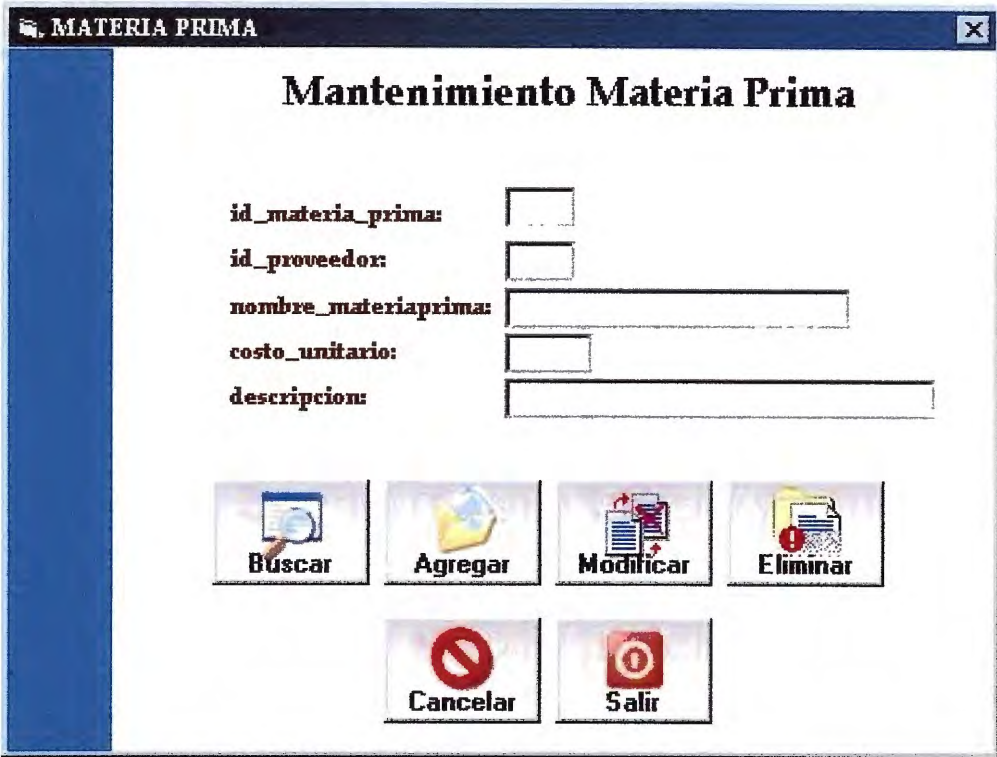
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.19 Mantenimiento de Datos – Datamart Producción – Por Proveedor – Materia Prima.

Mantenimiento de la base de datos Datamart producción, Por Proveedor es parte de un registro en línea que relaciona otras tablas de la base como son Proveedor y Contacto. El registro en línea es necesario para poder llenar con datos clave la tabla de hechos, esto es necesario debido al esquema de estrella utilizado en las bases de datos Datamart Ventas y Datamart Producción.



**MATERIA PRIMA**

### Mantenimiento Materia Prima

id\_materia\_prima:

id\_proveedor:

nombre\_materiaprima:

costo\_unitario:

descripcion:

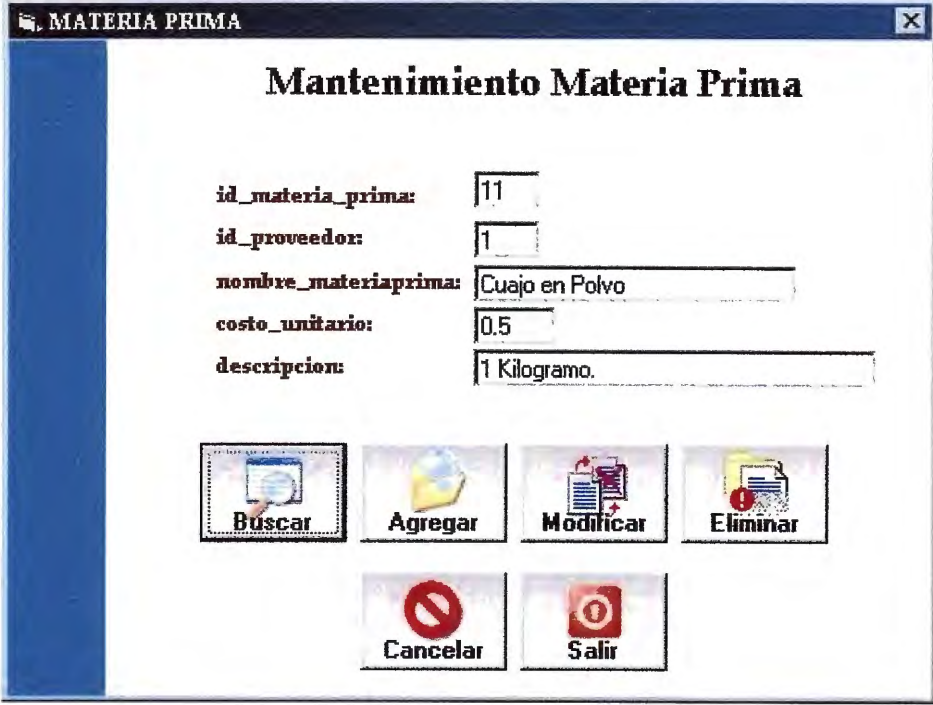
**Buscar** **Agregar** **Modificar** **Eliminar**

**Cancelar** **Salir**

**Figura 67. Mantenimiento de Materia Prima**

**Buscar.**

Utilizamos este proceso para buscar un registro en específico. El id\_materia\_prima es un código combinado que se explica a continuación: numero de producto + id \_proveedor



MATERIA PRIMA

### Mantenimiento Materia Prima

id\_materia\_prima: 11

id\_proveedor: 1

nombre\_materiaprima: Cuajo en Polvo

costo\_unitario: 0.5

descripcion: 1 Kilogramo.

Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 68. Búsqueda de Materia Prima**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

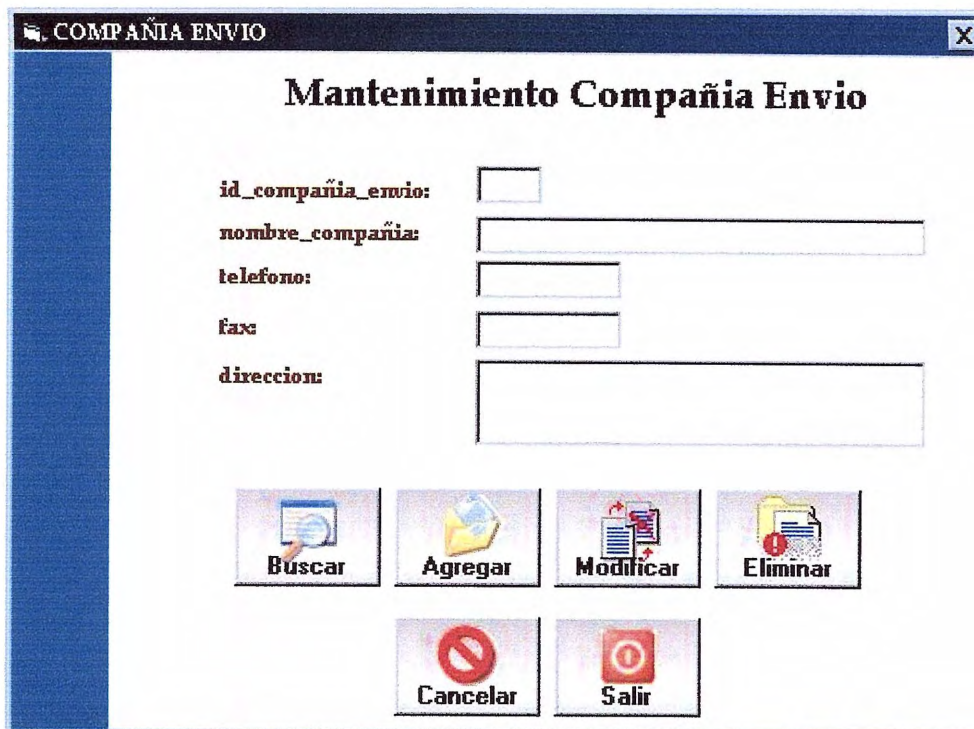
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

## 9.20 Mantenimiento de Datos – Datamart Producción – Compañía Envío.

Mantenimiento de la base de datos Datamart producción, se encarga de llevar un control sobre la compañía que se encarga de transportar el producto hacia su destino.



COMPAÑIA ENVIO

### Mantenimiento Compañía Envío

id\_compañía\_envio:

nombre\_compañía:

telefono:

fax:

dirección:

Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 69. Mantenimiento de Compañía de Envío**

### Buscar.

Utilizamos este proceso para buscar un registro en específico. El id\_compañía\_envio es un número correlativo que comienza de 1.

COMPANIA ENVIO

### Mantenimiento Compañía Envío

id\_compañía\_envio: 1

nombre\_compañía: Biolact Transportes

telefono: 3050270

fax: 3050272

dirección: Carretera del Litoral, Km 80, 300 Mts al Norte de Gasolinera ESSO, Municipio de Tecoluca

Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 70. Búsqueda de Compañía de Envío**

#### **Agregar.**

Utilizaremos este proceso para agregar registros a la base.

#### **Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

#### **Eliminar.**

Este proceso borrara un registro en específico.

#### **Cancelar.**

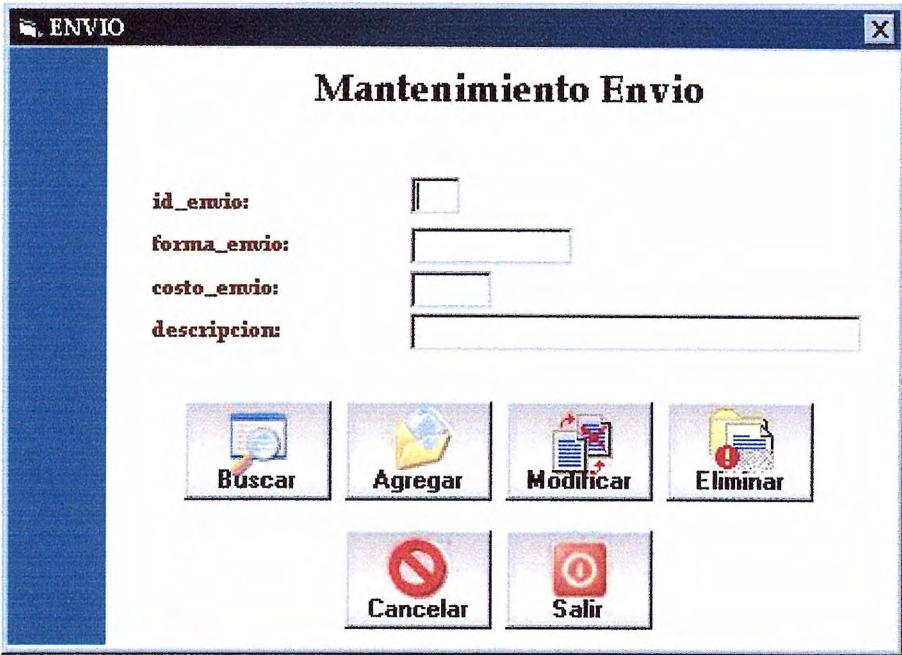
Este proceso cancela cualquier actividad dentro de este formulario.

#### **Salir.**

Salir del formulario de mantenimiento.

## 9.21 Mantenimiento de Datos – Datamart Producción – Forma de envió.

Mantenimiento de la base de datos Datamart producción, se encarga de llevar un control sobre la forma de envió y lo que esto representa en el costo de envió.



The screenshot shows a window titled "ENVIO" with a sub-header "Mantenimiento Envío". The window contains a form with the following fields:

- id\_envio: [input field]
- forma\_envio: [input field]
- costo\_envio: [input field]
- descripcion: [input field]

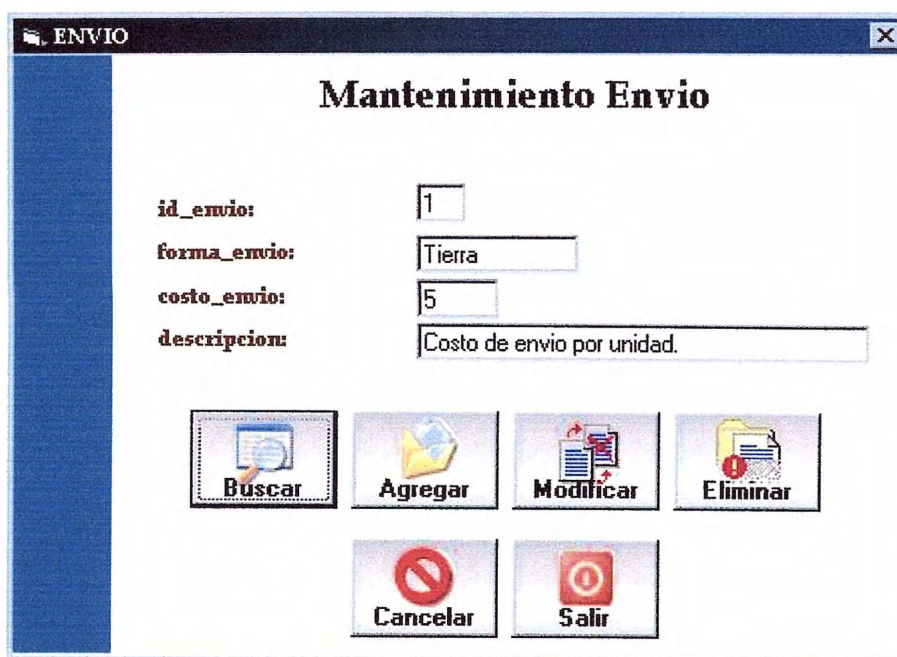
Below the form, there are six buttons arranged in two rows:

- Row 1: Buscar, Agregar, Modificar, Eliminar
- Row 2: Cancelar, Salir

Figura 71. Mantenimiento de Envío

### Buscar.

Utilizamos este proceso para buscar un registro en específico. El id\_envio es un correlativo que comienza de 1.



ENVIO

### Mantenimiento Envío

id\_envio:

forma\_envio:

costo\_envio:

descripcion:

Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 72. Búsqueda de Envío**

#### **Agregar.**

Utilizaremos este proceso para agregar registros a la base.

#### **Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

#### **Eliminar.**

Este proceso borrara un registro en específico.

#### **Cancelar.**

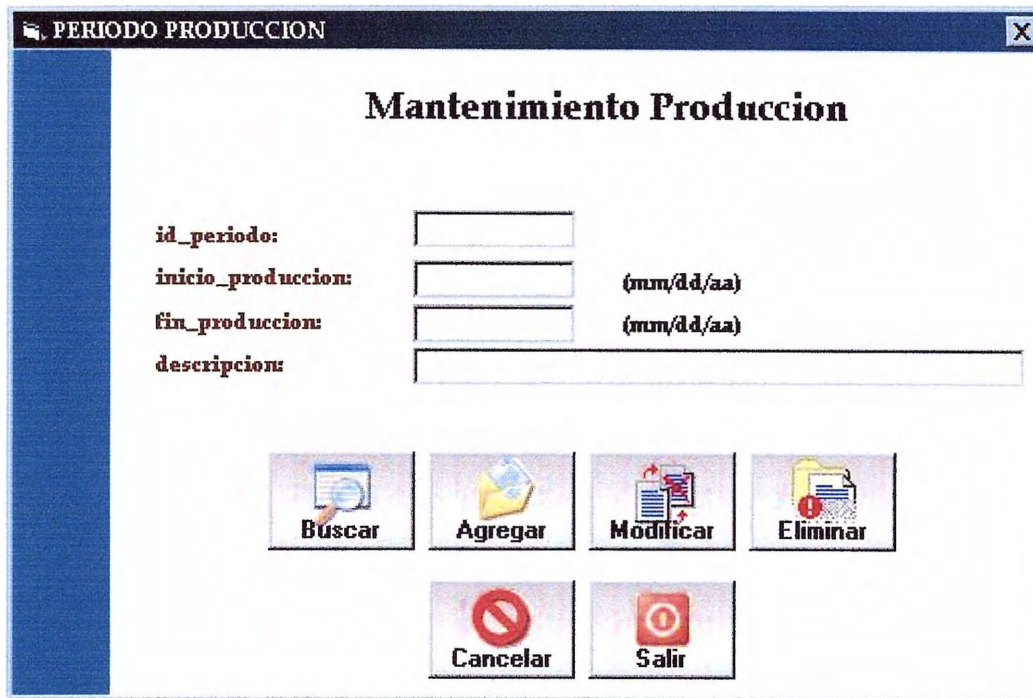
Este proceso cancela cualquier actividad dentro de este formulario.

#### **Salir.**

Salir del formulario de mantenimiento.

## 9.22 Mantenimiento de Datos – Datamart Producción – Periodo de Producción.

Mantenimiento de la base de datos Datamart producción, se encarga de llevar un control sobre los periodos de producción.



The screenshot shows a window titled "PERIODO PRODUCCION" with a sub-header "Mantenimiento Produccion". The form includes the following fields and controls:

- id\_periode:** A text input field.
- inicio\_produccion:** A date input field with the format "(mm/dd/aa)".
- fin\_produccion:** A date input field with the format "(mm/dd/aa)".
- descripcion:** A larger text input field.
- Buttons:** A row of four buttons: "Buscar" (magnifying glass icon), "Agregar" (folder icon), "Modificar" (document with red plus icon), and "Eliminar" (document with red minus icon). Below this row are two more buttons: "Cancelar" (red circle with slash icon) and "Salir" (red square with white circle icon).

**Figura 73. Mantenimiento de Producción**

### Buscar.

Utilizamos este proceso para buscar un registro en específico. El id\_envio forma parte de un mnemónico que se describe a continuación.

Mes de la producción siguiente + código de producto + 1 si es producto único o 2 si es producto variado

PERIODO PRODUCCION

### Mantenimiento Produccion

id\_periodes: 01QP1

inicio\_produccion: 01/12/2002 (mm/dd/aa)

fin\_produccion: 29/12/2002 (mm/dd/aa)

descripcion: Periodo de Produccion de Queso Probiotico

Buscar Agregar Modificar Eliminar

Cancelar Salir

**Figura 74. Búsqueda de Producción**

**Agregar.**

Utilizaremos este proceso para agregar registros a la base.

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

**Cancelar.**

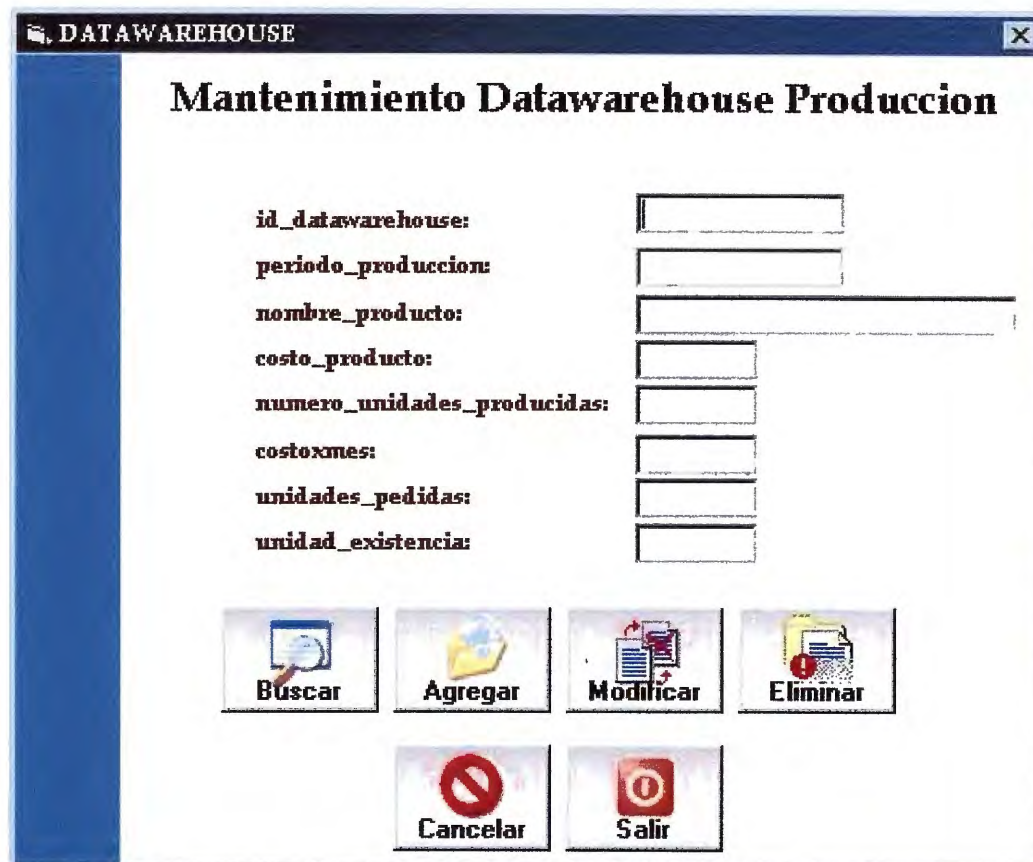
Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento.

### 9.23 Mantenimiento de Datos – Datamart Producción – Datawarehouse Producción.

Mantenimiento de la base de datos Datamart producción, se encarga de llevar un resumen que puede ser mensual o dependiendo del periodo que le interese estudiar al analista de ventas.



The screenshot shows a window titled "DATAWAREHOUSE" with a sub-header "Mantenimiento Datawarehouse Produccion". The form contains the following fields:

- id\_datawarehouse: [input field]
- periodo\_produccion: [input field]
- nombre\_producto: [input field]
- costo\_producto: [input field]
- numero\_unidades\_producidas: [input field]
- costoxmes: [input field]
- unidades\_pedidas: [input field]
- unidad\_existencia: [input field]

Below the form are six buttons:

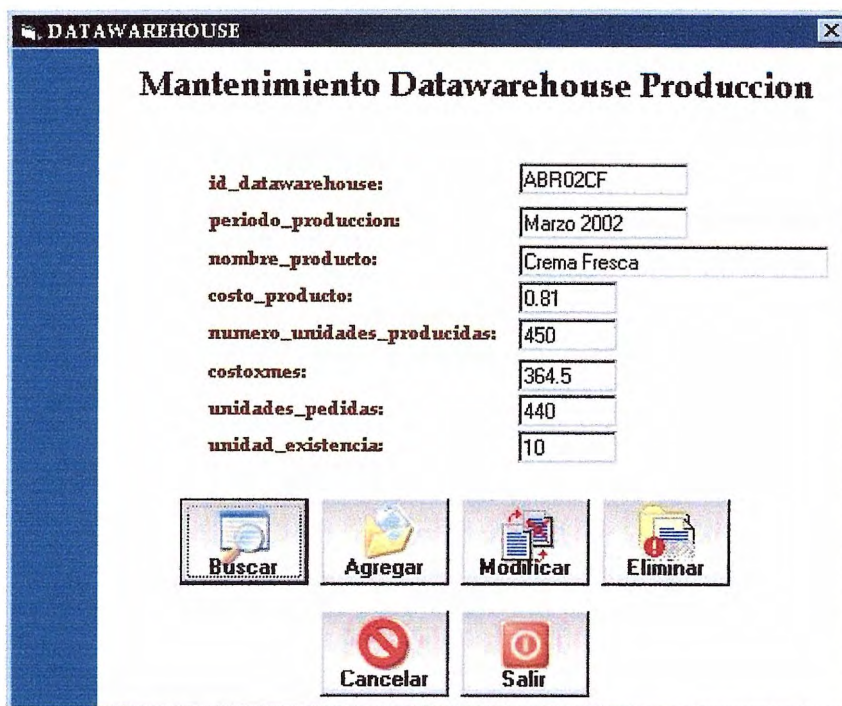
- Buscar (Search)
- Agregar (Add)
- Modificar (Modify)
- Eliminar (Delete)
- Cancelar (Cancel)
- Salir (Exit)

Figura 75. Mantenimiento DataWarehouse Producción

## Buscar

Utilizamos este proceso para buscar un registro en específico. El id\_envio forma parte de un mnemónico que se describe a continuación.

Tres letras para el mes de producción + 2 dígitos para el año + código del producto.



id_datawarehouse:	ABR02CF
periodo_produccion:	Marzo 2002
nombre_producto:	Crema Fresca
costo_producto:	0.81
numero_unidades_producidas:	450
costoxmes:	364.5
unidades_pedidas:	440
unidad_existencia:	10

Buttons: Buscar, Agregar, Modificar, Eliminar, Cancelar, Salir

**Figura 76. Búsqueda DataWarehouse Producción**

### **Agregar.**

Utilizaremos este proceso para agregar registros a la base.

### **Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

### **Eliminar.**

Este proceso borrara un registro en específico.

### **Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

### **Salir.**

Salir del formulario de mantenimiento.

## 9.24 Mantenimiento de Datos – Datamart Producción – Hechos Producción.

Mantenimiento de la base de datos Datamart producción, es el mas importante de todos, es el resultado del registro en línea que como se ha explicado anteriormente es indispensable para el manejo de las operaciones en el área de producción de la base de datos.

The screenshot shows a software window titled "HECHOS PRODUCCION" with a sub-header "Mantenimiento Hechos Produccion". On the left, there is a list of fields with their respective input boxes:

- id\_datawarehouse: [ ]
- id\_correlativo: [ 1 ]
- id\_categoria: [ ]
- id\_compania\_emisor: [ ]
- id\_proveedor: [ ]
- id\_inventaria: [ ]
- id\_emisor: [ ]
- id\_periodes: [ ]
- coste\_producto: [ ]
- numero\_unidades\_producidas: [ ]
- coste\_mmc: [ ]

At the bottom of the window, there are several buttons: "Buscar", "Agregar", "Modificar", "Eliminar", "Cancelar", and "Salir".

Figura 77. Mantenimiento Hechos Producción

### Buscar.

Utilizamos este proceso para buscar un registro en específico.

**Agregar.**

Utilizaremos este proceso para agregar registros a la base, para poder agregar el registro en línea es necesario dar clic en Agregar y luego completar una serie de datos para poder tener un registro completo de la venta. El dato que hay que introducir es el de Datawarehouse ya que no esta directamente relacionado con la tabla de hechos ventas.

The screenshot shows a web application window titled "HECHOS PRODUCCION" with a sub-header "Mantenimiento Hechos Produccion". The form contains the following fields and controls:

- id\_datawarehouse:** Text input field.
- id\_correlativo:** Text input field with value "1".
- id\_categoria:** Text input field with value "C".
- id\_compania\_emisor:** Text input field with value "1".
- id\_proveedor:** Text input field with value "1".
- id\_inventario:** Text input field.
- id\_emisor:** Text input field with value "1".
- id\_periodo:** Text input field with value "01CF1".
- costo\_producto:** Text input field with value "0.85".
- numero\_unidades\_producidas:** Text input field with value "500".
- costoxmes:** Text input field with value "A25".
- Product Selection:** A series of dropdown menus:
  - Dropdown 1: "Cremas" (selected), "Crema Especial" (shown).
  - Dropdown 2: "Biolact Transportes" (shown).
  - Dropdown 3: "Larbus, S. R." (shown).
  - Dropdown 4: "01CF12" (shown).
  - Dropdown 5: "Tierra" (shown).
  - Dropdown 6: "01CF1" (shown).

At the bottom of the form, there are six buttons: "Buscar", "Aceptar", "Modificar", "Eliminar", "Cancelar", and "Salir".

**Figura 78. Agregar Hechos Producción**

**Modificar.**

Utilizaremos este proceso para modificar los datos de un registro erróneo.

**Eliminar.**

Este proceso borrara un registro en específico.

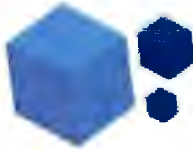
**Cancelar.**

Este proceso cancela cualquier actividad dentro de este formulario.

**Salir.**

Salir del formulario de mantenimiento

**OLAP**



**Viewer**

version 1.0



**Manual del Programador**

## INDICE

Introducción.....	I
1. Definición de las Bases de Datos.....	1
2. Formularios.....	11
3. Modulos.....	26
4. Funciones y Procedimientos.....	27

## INDICE DE IMÁGENES

### ❖ FIGURAS

Figura 1. dbo.Cliente.....	1
Figura 2. dbo.Contacto.....	1
Figura 3. dbo.Datawarehouse_Ventas.....	2
Figura 4. dbo.Datos Personales.....	2
Figura 5. dbo.Hechos Ventas.....	3
Figura 6. dbo.Localizacion.....	3
Figura 7. dbo.Pagos.....	4
Figura 8. dbo.Producto.....	4
Figura 9. dbo.Tiempo.....	4
Figura 10. dbo.Vendedor.....	5
Figura 11. dbo.Zona.....	5
Figura 12. dbo.Categoria.....	6
Figura 13. dbo.Compañía Envío.....	6
Figura 14. dbo.Contacto.....	7
Figura 15. dbo.Datawarehouse.....	7
Figura 16. dbo.Detalle Pedido.....	7
Figura 17. dbo.Envío.....	8
Figura 18. dbo.Hechos Produccion.....	8
Figura 19. dbo.Inventario.....	9
Figura 20. dbo.Materia Prima.....	9
Figura 21. dbo.Pedido.....	9
Figura 22. dbo.Periodo produccion.....	10
Figura 23. dbo.Producto.....	10
Figura 24. dbo.Proveedor.....	10

## INTRODUCCIÓN

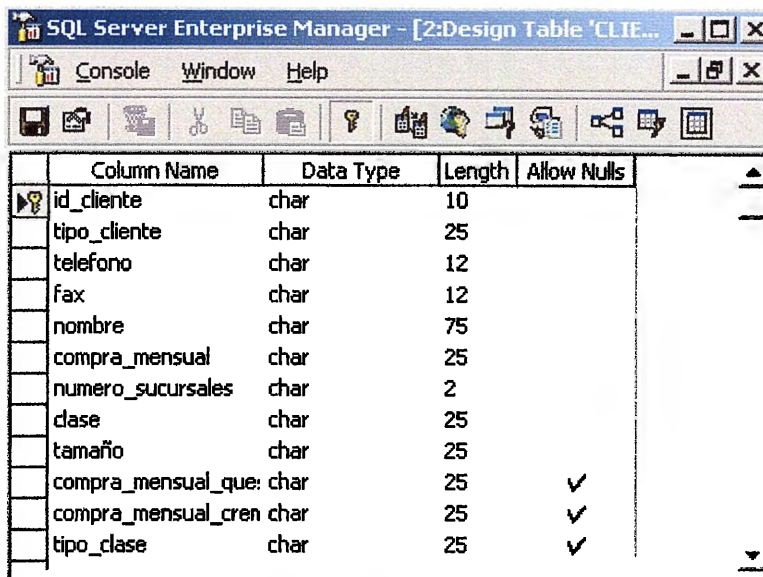
El manual del programador tiene como objetivo brindar información referente al desarrollo de la aplicación **“Sistema OLAP Viewer”**.

Ya que este será una guía para el programador que quiera actualizarlo. Y así seguir utilizando lo que es Servidor OLAP Y Minería de Datos, ya que estas herramientas pueden ser de gran utilidad a la toma de decisión.

## 1. DEFINICIÓN DE LAS BASES DE DATOS

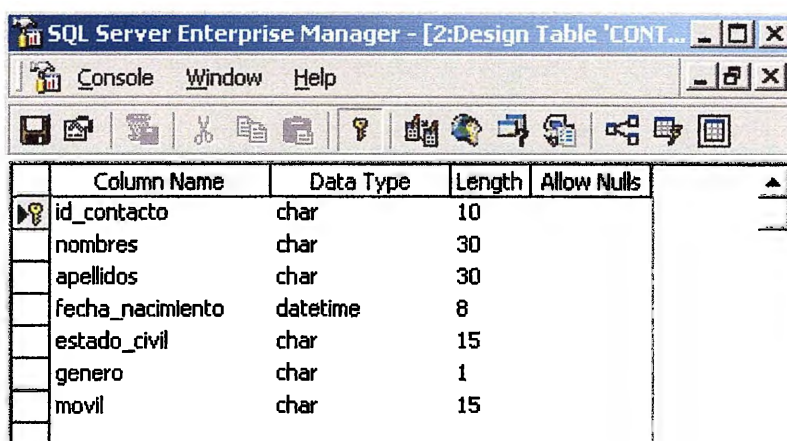
<b>Tecnología utilizada:</b>	Microsoft SQL Server 2000
<b>Nombre de Base de Datos:</b>	<b>DATAMART VENTAS</b>

### Diseño de las tablas:



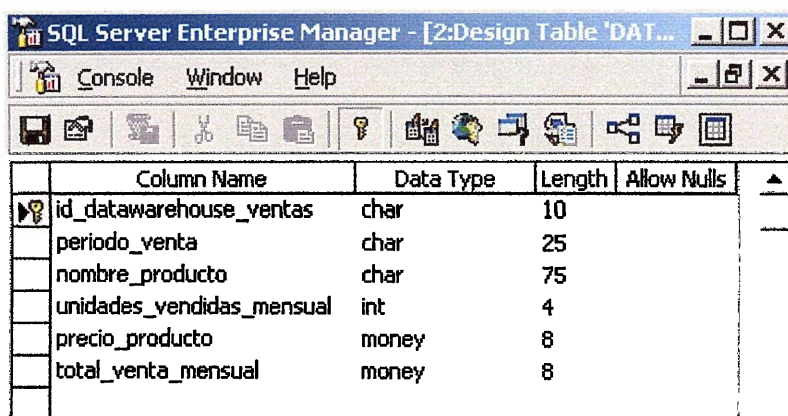
Column Name	Data Type	Length	Allow Nulls
id_cliente	char	10	
tipo_cliente	char	25	
telefono	char	12	
fax	char	12	
nombre	char	75	
compra_mensual	char	25	
numero_sucursales	char	2	
clase	char	25	
tamaño	char	25	
compra_mensual_que	char	25	✓
compra_mensual_cren	char	25	✓
tipo_clase	char	25	✓

**Figura 1. dbo.Cliente**



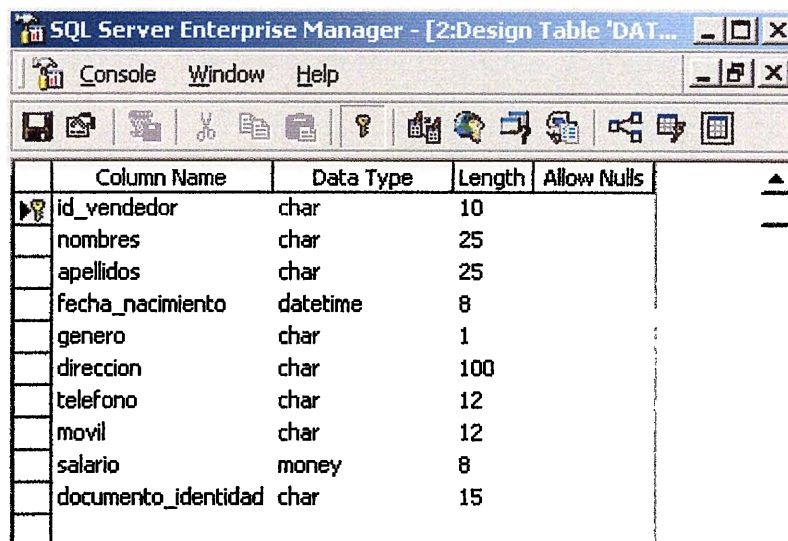
Column Name	Data Type	Length	Allow Nulls
id_contacto	char	10	
nombres	char	30	
apellidos	char	30	
fecha_nacimiento	datetime	8	
estado_civil	char	15	
genero	char	1	
movil	char	15	

**Figura 2. dbo.Contacto**



The screenshot shows the 'SQL Server Enterprise Manager' window with the 'Design Table' view for 'DataWarehouse\_Ventas'. The table structure is as follows:

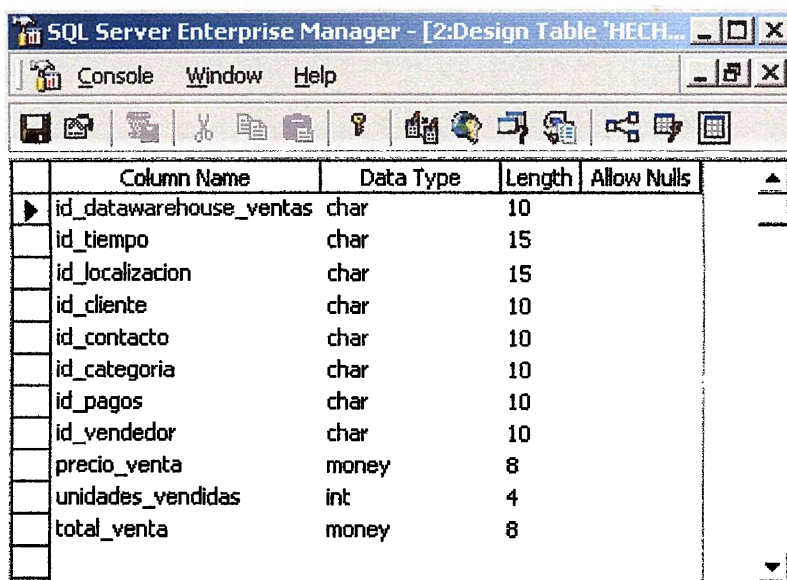
Column Name	Data Type	Length	Allow Nulls
id_datawarehouse_ventas	char	10	
periodo_venta	char	25	
nombre_producto	char	75	
unidades_vendidas_mensual	int	4	
precio_producto	money	8	
total_venta_mensual	money	8	

**Figura 3. dbo.DataWarehouse\_Ventas**

The screenshot shows the 'SQL Server Enterprise Manager' window with the 'Design Table' view for 'Datos Personales'. The table structure is as follows:

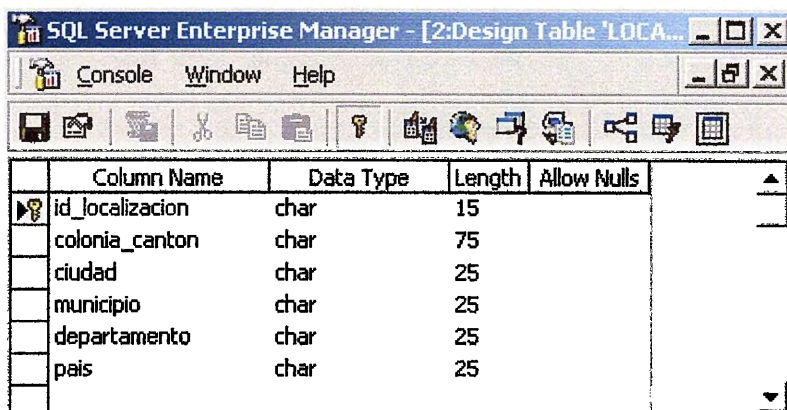
Column Name	Data Type	Length	Allow Nulls
id_vendedor	char	10	
nombres	char	25	
apellidos	char	25	
fecha_nacimiento	datetime	8	
genero	char	1	
direccion	char	100	
telefono	char	12	
movil	char	12	
salario	money	8	
documento_identidad	char	15	

**Figura 4. dbo.Datos Personales**



The screenshot shows the 'Design Table' view for a table named 'HECHOS'. The table has the following columns:

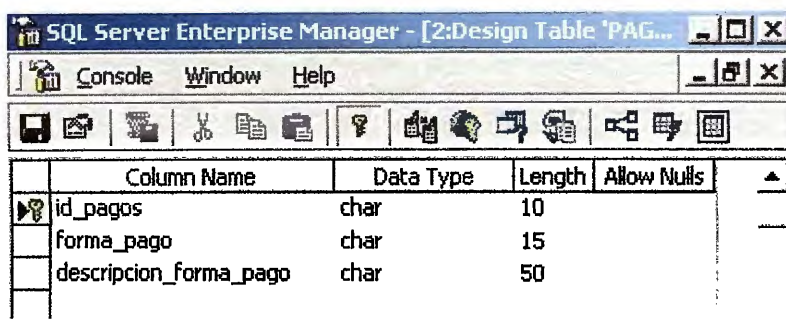
Column Name	Data Type	Length	Allow Nulls
id_datawarehouse_ventas	char	10	
id_tiempo	char	15	
id_localizacion	char	15	
id_cliente	char	10	
id_contacto	char	10	
id_categoria	char	10	
id_pagos	char	10	
id_vendedor	char	10	
precio_venta	money	8	
unidades_vendidas	int	4	
total_venta	money	8	

**Figura 5. dbo.Hechos Ventas**

The screenshot shows the 'Design Table' view for a table named 'LOCALIZACION'. The table has the following columns:

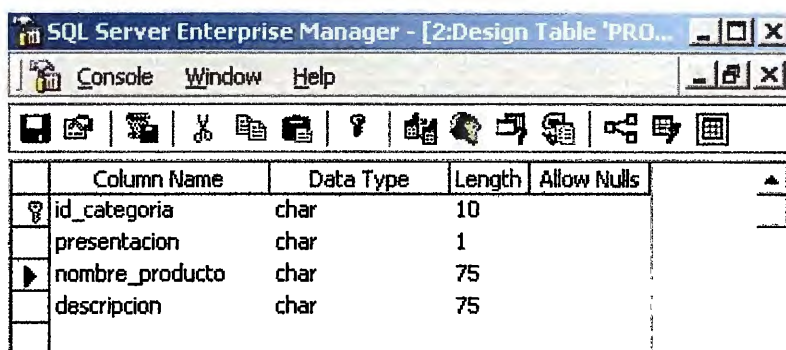
Column Name	Data Type	Length	Allow Nulls
id_localizacion	char	15	
colonia_canton	char	75	
ciudad	char	25	
municipio	char	25	
departamento	char	25	
pais	char	25	

**Figura 6. dbo.Localización**



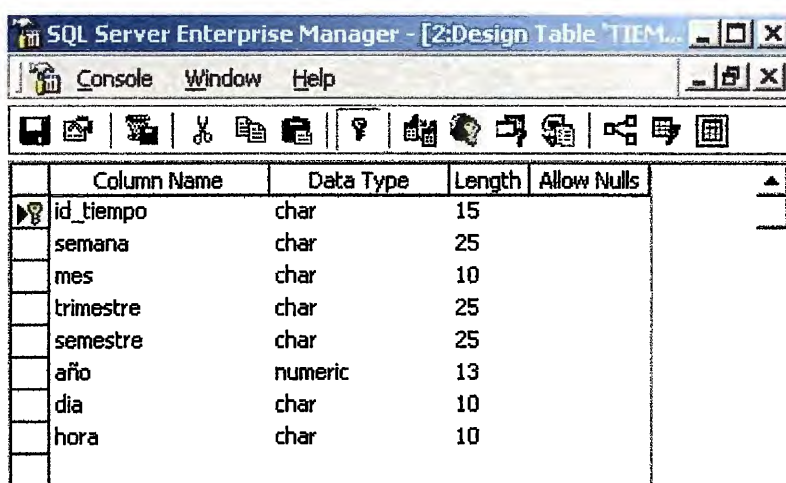
Column Name	Data Type	Length	Allow Nulls
id_pagos	char	10	
forma_pago	char	15	
descripcion_forma_pago	char	50	

Figura 7. dbo.Pagos



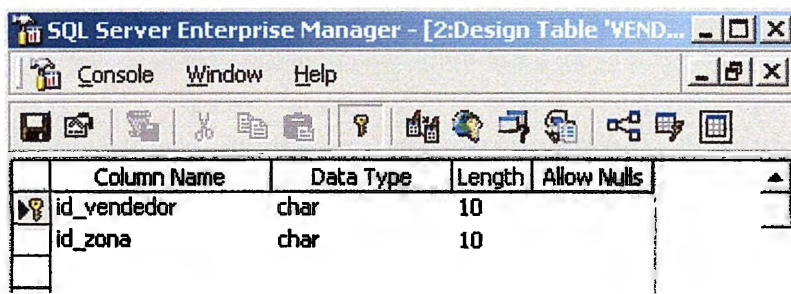
Column Name	Data Type	Length	Allow Nulls
id_categoria	char	10	
presentacion	char	1	
nombre_producto	char	75	
descripcion	char	75	

Figura 8. dbo.Producto



Column Name	Data Type	Length	Allow Nulls
id_tiempo	char	15	
semana	char	25	
mes	char	10	
trimestre	char	25	
semestre	char	25	
año	numeric	13	
dia	char	10	
hora	char	10	

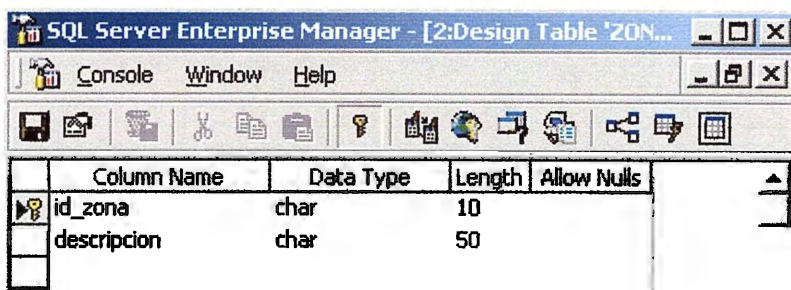
Figura 9. dbo.Tiempo



SQL Server Enterprise Manager - [2:Design Table 'VEND...]

Console Window Help

	Column Name	Data Type	Length	Allow Nulls
	id_vendedor	char	10	
	id_zona	char	10	

**Figura 10. dbo.Vendedor**

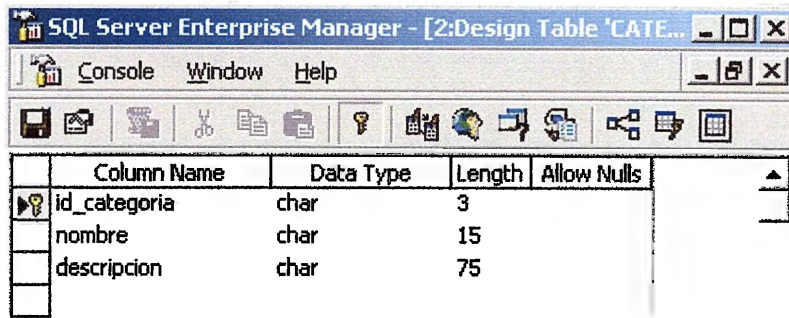
SQL Server Enterprise Manager - [2:Design Table 'ZON...]

Console Window Help

	Column Name	Data Type	Length	Allow Nulls
	id_zona	char	10	
	descripcion	char	50	

**Figura 11. dbo.Zona**

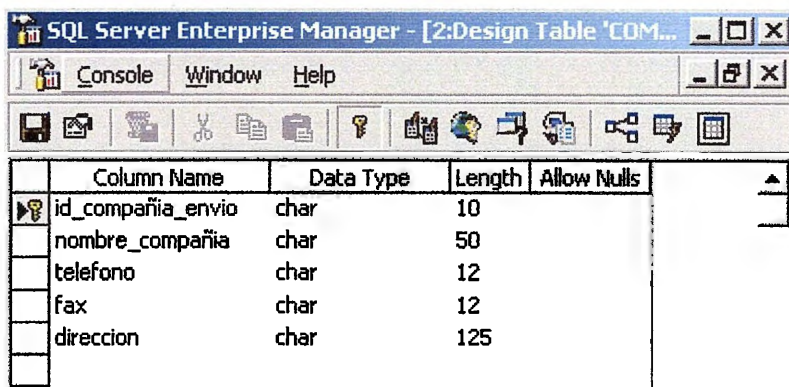
<b>Tecnología utilizada:</b>	Microsoft SQL Server 2000
<b>Nombre de Base de Datos:</b>	<b>DATAMART PRODUCCION</b>



The screenshot shows the 'Design Table' view for the 'CATEGORIA' table in SQL Server Enterprise Manager. The table has three columns: 'id\_categoria' (char, length 3), 'nombre' (char, length 15), and 'descripcion' (char, length 75). The 'id\_categoria' column is marked as the primary key.

Column Name	Data Type	Length	Allow Nulls
id_categoria	char	3	
nombre	char	15	
descripcion	char	75	

**Figura 12. dbo.Categoría**



The screenshot shows the 'Design Table' view for the 'COMPANIA ENVIOS' table in SQL Server Enterprise Manager. The table has five columns: 'id\_compañia\_envio' (char, length 10), 'nombre\_compañia' (char, length 50), 'telefono' (char, length 12), 'fax' (char, length 12), and 'direccion' (char, length 125). The 'id\_compañia\_envio' column is marked as the primary key.

Column Name	Data Type	Length	Allow Nulls
id_compañia_envio	char	10	
nombre_compañia	char	50	
telefono	char	12	
fax	char	12	
direccion	char	125	

**Figura 13. dbo.Compañía Envío**

The screenshot shows the 'Design Table' view for 'dbo.Contacto' in SQL Server Enterprise Manager. The table has the following columns:

Column Name	Data Type	Length	Allow Nulls
id_contacto	numeric	9	
id_proveedor	char	10	
nombres_contacto	char	25	
apellidos_contacto	char	25	
cargo	char	50	
direccion	char	100	
telefono	char	13	

Figura 14. dbo.Contacto

The screenshot shows the 'Design Table' view for 'dbo.DataWarehouse' in SQL Server Enterprise Manager. The table has the following columns:

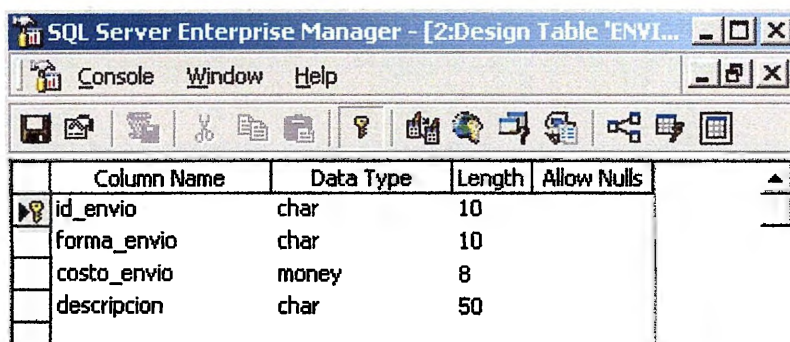
Column Name	Data Type	Length	Allow Nulls
id_datawarehouse	char	10	
periodo_produccion	char	50	
nombre_producto	char	75	
costo_producto	money	8	
numero_unidades_prc	numeric	9	
costoxmes	money	8	
unidades_pedidas	numeric	9	
unidad_existencia	numeric	9	

Figura 15. dbo.DataWarehouse

The screenshot shows the 'Design Table' view for 'dbo.Detalle Pedido' in SQL Server Enterprise Manager. The table has the following columns:

Column Name	Data Type	Length	Allow Nulls
id_detalle_pedido	char	10	
direccion	char	150	
ciudad	char	50	
pais	char	50	
cantidadxunidad	numeric	9	
presentacion	char	3	

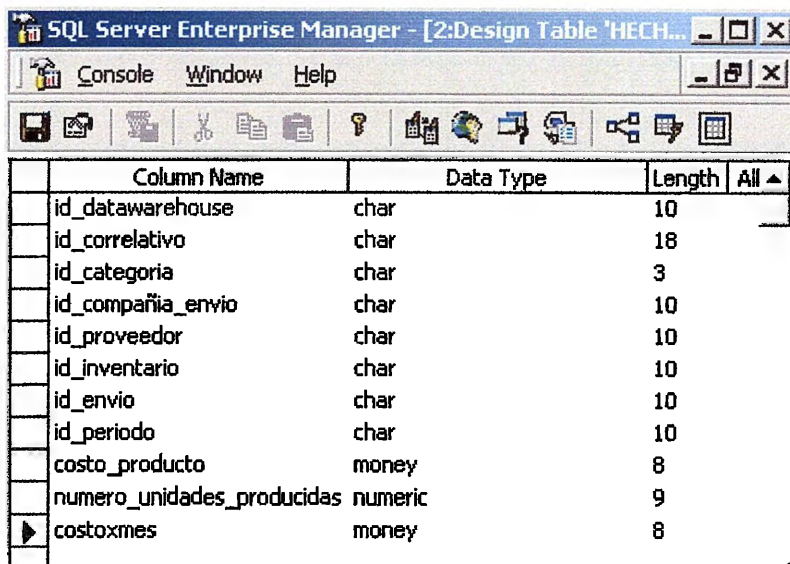
Figura 16. dbo.Detalle Pedido



The screenshot shows the 'Design Table' view for 'dbo.Envio' in SQL Server Enterprise Manager. The table has four columns: 'id\_envio' (char, 10), 'forma\_envio' (char, 10), 'costo\_envio' (money, 8), and 'descripcion' (char, 50). The 'id\_envio' column is marked as a primary key.

Column Name	Data Type	Length	Allow Nulls
id_envio	char	10	
forma_envio	char	10	
costo_envio	money	8	
descripcion	char	50	

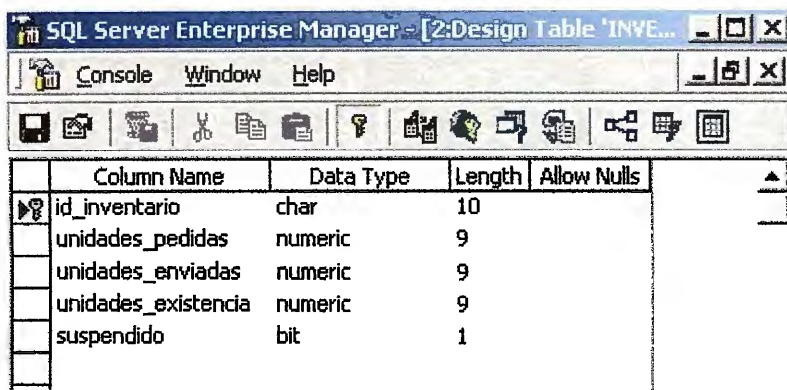
Figura 17. dbo.Envio



The screenshot shows the 'Design Table' view for 'dbo.Hechos Producción' in SQL Server Enterprise Manager. The table has 13 columns: 'id\_datawarehouse' (char, 10), 'id\_correlativo' (char, 18), 'id\_categoria' (char, 3), 'id\_compañia\_envio' (char, 10), 'id\_proveedor' (char, 10), 'id\_inventario' (char, 10), 'id\_envio' (char, 10), 'id\_periodo' (char, 10), 'costo\_producto' (money, 8), 'numero\_unidades\_producidas' (numeric, 9), and 'costoxmes' (money, 8). The 'id\_correlativo' column is marked as a primary key.

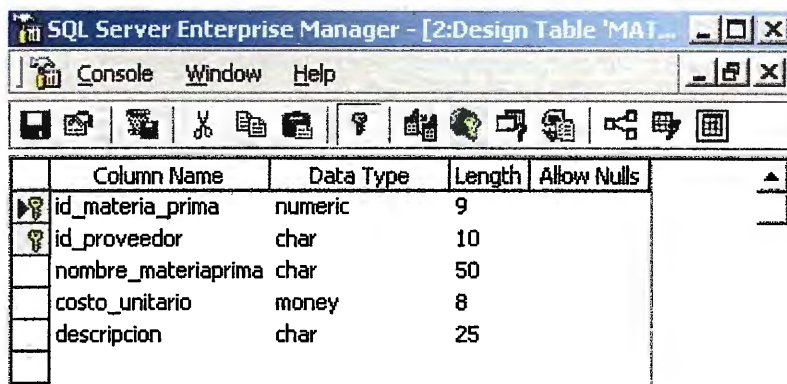
Column Name	Data Type	Length	All
id_datawarehouse	char	10	
id_correlativo	char	18	
id_categoria	char	3	
id_compañia_envio	char	10	
id_proveedor	char	10	
id_inventario	char	10	
id_envio	char	10	
id_periodo	char	10	
costo_producto	money	8	
numero_unidades_producidas	numeric	9	
costoxmes	money	8	

Figura 18. dbo.Hechos Producción



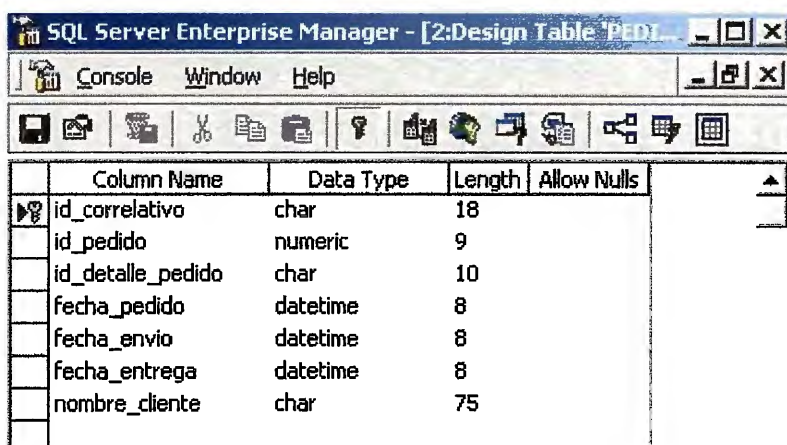
Column Name	Data Type	Length	Allow Nulls
id_inventario	char	10	
unidades_pedidas	numeric	9	
unidades_enviadas	numeric	9	
unidades_existencia	numeric	9	
suspendido	bit	1	

Figura 19. dbo.Inventario



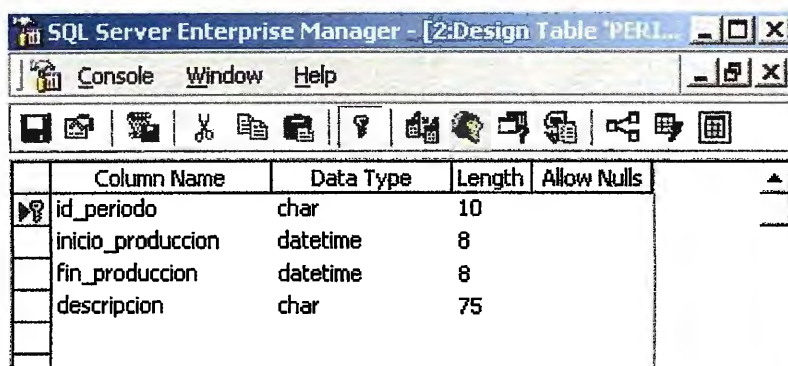
Column Name	Data Type	Length	Allow Nulls
id_materia_prima	numeric	9	
id_proveedor	char	10	
nombre_materiaprima	char	50	
costo_unitario	money	8	
descripcion	char	25	

Figura 20. dbo.Materia Prima



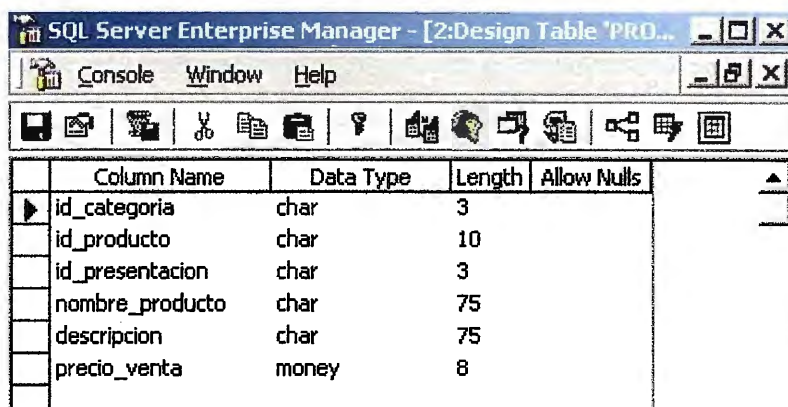
Column Name	Data Type	Length	Allow Nulls
id_correlativo	char	18	
id_pedido	numeric	9	
id_detalle_pedido	char	10	
fecha_pedido	datetime	8	
fecha_envio	datetime	8	
fecha_entrega	datetime	8	
nombre_cliente	char	75	

Figura 21. dbo.Pedido



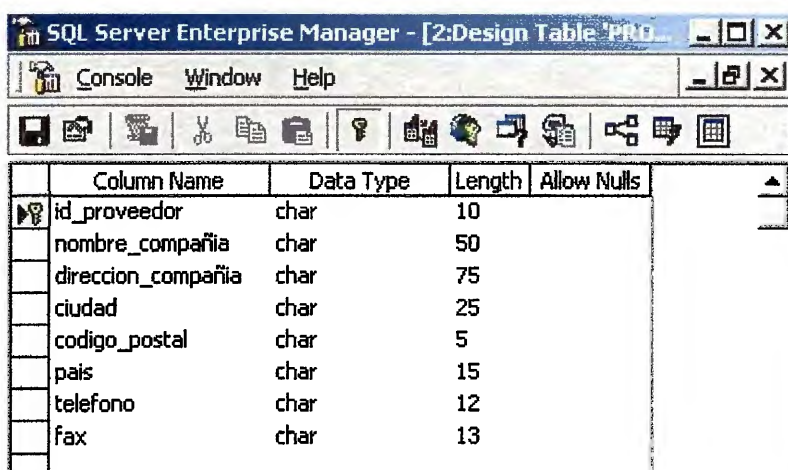
Column Name	Data Type	Length	Allow Nulls
id_periodo	char	10	
inicio_produccion	datetime	8	
fin_produccion	datetime	8	
descripcion	char	75	

Figura 22. dbo.Periodo Producción



Column Name	Data Type	Length	Allow Nulls
id_categoria	char	3	
id_producto	char	10	
id_presentacion	char	3	
nombre_producto	char	75	
descripcion	char	75	
precio_venta	money	8	

Figura 23. dbo.Producto



Column Name	Data Type	Length	Allow Nulls
id_proveedor	char	10	
nombre_compania	char	50	
direccion_compania	char	75	
ciudad	char	25	
codigo_postal	char	5	
pais	char	15	
telefono	char	12	
fax	char	13	

Figura 24. dbo.Proveedor

## 2. FORMULARIOS

A continuación se detallan los nombres y descripciones de cada uno de los formularios que comprenden la aplicación:

<b>Lenguaje de programación:</b>	Microsoft Visual Basic 6.0
<b>Nombre de proyecto:</b>	<b>OLAPViewer.vbp</b>

<b>NOMBRE:</b>	Administrador
<b>ARCHIVO:</b>	Administrador.frm
<b>DESCRIPCION:</b>	Contiene todos los cubos, dimensiones y modelos de minería de datos que se han creado con la aplicación

<b>NOMBRE:</b>	Exploradordedatos
<b>ARCHIVO:</b>	Exploradordedatos.frm
<b>DESCRIPCION:</b>	Permite interactuar con los datos almacenados en un cubo

<b>NOMBRE:</b>	FormMDX
<b>ARCHIVO:</b>	MDX.frm
<b>DESCRIPCION:</b>	Aplicación MDX para el manejo de consultas de datos multidimensionales almacenados en cubos OLAP

<b>NOMBRE:</b>	FrmAcerca
<b>ARCHIVO:</b>	Acerca2.frm
<b>DESCRIPCION:</b>	Muestra el acerca de la aplicación

<b>NOMBRE:</b>	Grafico Dinámico
<b>ARCHIVO:</b>	Graficos.frm
<b>DESCRIPCION:</b>	Genera un grafico a partir de los datos almacenados en un cubo

<b>NOMBRE:</b>	Inicio Sesión
<b>ARCHIVO:</b>	Inicio_Sesion.frm
<b>DESCRIPCION:</b>	<p>Permite el ingreso a la aplicación, es necesario tener en consideración los siguientes aspectos:</p> <p>Insertar correctamente el login de Usuario previamente creado en SQL Server 2000.</p> <p>Los usuarios creados corresponden a Administrador, Producción y Ventas.</p> <p>Para Administrador el login y password respectivamente es: <b>sa, sa</b></p> <p>Para Producción es: <b>producción, pro</b></p> <p>Para Ventas es: <b>ventas, ven</b></p>

<b>NOMBRE:</b>	Menu Principal
<b>ARCHIVO:</b>	MDIForm1.frm
<b>DESCRIPCION:</b>	Este formulario contiene la interfaz local de la aplicación y contiene la barra de Menú y de esta manera conseguimos agilizar el flujo de operaciones con los almacenes de datos residentes en el servidor

<b>NOMBRE:</b>	Cliente
<b>ARCHIVO:</b>	Mtto_Cliente.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla cliente de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Contacto
<b>ARCHIVO:</b>	Mtto_Contacto.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla contacto de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Datos Personales
<b>ARCHIVO:</b>	Mtto_DatosPersonales.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Datos Personales de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Datawarehouse_Ventas
<b>ARCHIVO:</b>	Mtto_DWV.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Datawarehouse_ventas de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Hechos Ventas
<b>ARCHIVO:</b>	Mtto_Hechos.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Hechos Ventas de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Localización
<b>ARCHIVO:</b>	Mtto_Localizacion.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Localizacion de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Pagos
<b>ARCHIVO:</b>	Mtto_Pagos.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Pagos de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Producto
<b>ARCHIVO:</b>	Mtto_Producto.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Producto de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Tiempo
<b>ARCHIVO:</b>	Mtto_Tiempo.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Tiempo de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Vendedor
<b>ARCHIVO:</b>	Mtto_Vendedor.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Vendedor de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Zona
<b>ARCHIVO:</b>	Mtto_Zona.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Zona de la base de datos Datamart Ventas y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Categoría
<b>ARCHIVO:</b>	Prod_Categoria.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Categoría de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Compañía Envió
<b>ARCHIVO:</b>	Prod_Compenvio.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Compañía Envió de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Contacto
<b>ARCHIVO:</b>	Prod_Contacto.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Contacto de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Datawarehouse
<b>ARCHIVO:</b>	Prod_Datawarehouse.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Datawarehouse de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Detalle Pedido
<b>ARCHIVO:</b>	Prod_Detallepedido.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Detalle Pedido de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Envio
<b>ARCHIVO:</b>	Prod_Envio.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Envio de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Hechos Producción
<b>ARCHIVO:</b>	Prod_.HechosProd.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Hechos Producción de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Inventario
<b>ARCHIVO:</b>	Prod_Inventario.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Inventario de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Materia Prima
<b>ARCHIVO:</b>	Prod_Materiaprime.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Materia Prima de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Pedido
<b>ARCHIVO:</b>	Prod_Pedido.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Pedido de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

<b>NOMBRE:</b>	Periodo Producción
<b>ARCHIVO:</b>	Prod_Periodo.frm
<b>DESCRIPCION:</b>	<p>Este formulario se utiliza para el mantenimiento de la tabla Periodo Producción de la base de datos Datamart Producción y contiene los siguientes procedimientos:</p> <ul style="list-style-type: none"><li>• Buscar</li><li>• Agregar</li><li>• Modificar</li><li>• Eliminar</li><li>• Cancelar</li><li>• Salir</li></ul>

### 3. MODULOS

La aplicación hace uso de un modulo

<b>Nombre de Modulo:</b>	moduloOLAP.bas
--------------------------	----------------

Este modulo contiene la función con la cual un usuario puede conectarse a SQL Server 2000 utilizando el tipo de conexión ADODB.

La función es:

**Fuction Conectaru (USER As string, PSW As string)**

## Funciones y Procedimientos

para la generación de la aplicación se utilizaron las siguientes funciones o procedimientos las cuales están alojadas en ciertos objetos de Visual Basica

- **Borrar\_Cubo**

Tiene como objetivo eliminar un cubo que pertenece a una base de datos

- **ConectarServidor**

Crea una instancia la cual hace una conexión al servidor que se esta utilizando, usando una conexión DSO

- **ListarBases**

En este procedimiento se muestra los Cubos, Medidas, Dimensiones y Modelos de Minería de datos que pertenecen a una Base de datos en un objeto de Visual Basic llamado TreeView1

- **Agregar\_Cubo**

Funcion que se utiliza para agregar un nuevo cubo

- **Procesar\_Cubo**

Procesa un cubo que pertenece a una Base de datos para que se guarden los nuevos cambios del cubo utilizando una conexión DSO

- **VerEstado**

Muestra el estado del Servidor cuando se ha realizado una conexión

- **EjecutarMDX**

Ejecuta una consulta MDX y los resultados se muestran en un objeto MSFlexGrid

- **hechos**

Función que se utiliza para mostrar una tabla de hechos

- **niveles**

Procedimiento que se utiliza para agregar niveles a las dimensiones

- **Agregar\_Dimension**

Procedimiento que se utiliza para agregar dimensiones a un cubo que existe en una base de datos

- **Mnuabrirmdx**

Abre una consulta con extensión .mdx la cual ya esta almacenada

- **Mnuguardarmdx**

Guarda una consulta MDX

- **Mnureporte**

Abre un reporte dinámico y multidimensional a partir de los datos proporcionados en una consulta MDX

- **LlenarMSFlexGrid1**

Esta función tiene el objetivo de llenar un objeto MSFlexGrid por medio de una consulta MDX

- **VerCubos**

Muestra los cubos que pertenecen a una base de datos los cuales se visualizan en un objeto treeview1 usando una conexión Adomd

- **Cmdacep**

Llama a la función que se encuentra en el modulo con la cual un usuario puede iniciar sesión

- **Agregar**

Función que se utiliza para ingresar registros a las tablas que pertenecen una base de Datos

- **Buscar**

Realiza una búsqueda por medio del código a una tabla que pertenece a una Base de Datos

- **Cancelar**

Función que se utiliza para cancelar cualquier acción

- **Eliminar**

Función que se utiliza para eliminar un registro

- **Modificar**

Función que se utiliza para modificar los campos de un registro