

**CONSTRUCCIÓN DEL MODELO DE VENTAS MULTIDIMENSIONAL
BASADO EN LA INFORMACIÓN HISTÓRICA DE LA ORGANIZACIÓN
TDM TRANSPORTES S.A.S.**



**HAROLD DARIO JIMENEZ ARBELAEZ
DANNE ARLEY RAMIREZ ZAPATA**

ASESOR:

JOSE EUCARIO PARRA CASTRILLON

**UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN EN GESTION DE INFORMACION Y BASES DE DATOS
MEDELLÍN
2011**

CONTENIDO

1.	INFORMACIÓN GENERAL DEL PROYECTO.....	12
1.1.	TÍTULO.....	12
1.2.	INTEGRANTES.....	12
1.3.	TIPO DE INVESTIGACIÓN.....	12
1.4.	NIVEL DE INVESTIGACIÓN.....	12
1.5.	TIPO DE PROYECTO.....	13
1.6.	LÍNEA DE INVESTIGACIÓN USB.....	13
2.	DESCRIPCION DEL PROYECTO.....	14
2.1.	PLANTEAMIENTO DE LA PREGUNTA O PROBLEMA DE INVESTIGACIÓN Y SU JUSTIFICACIÓN EN TÉRMINOS DE NECESIDADES Y PERTINENCIA.....	14
2.2.	OBJETIVOS.....	15
2.2.1.	General.....	15
2.2.2.	Específicos.....	15
2.3.	MARCO TEÓRICO.....	16
2.3.1.	Definiciones de DataMart.....	18
2.3.2.	Modelo Multidimensional.....	18
2.3.3.	Componentes en la creación de un DataMart.....	21
2.3.4.	Procesos de extracción, transformación y carga de datos (ETL) ..	23
2.3.5.	DataWareHouse.....	26
2.3.6.	Herramientas de Explotación.....	27
2.4.	REFERENCIA GLOBAL.....	37

2.5.	DISEÑO METODOLÓGICO	41
2.5.1.	Planeación del DataMart	43
2.5.2.	Requerimientos y Análisis del Negocio	44
2.5.3.	Análisis Dimensional	45
2.5.4.	Diseño	46
2.5.5.	Construcción	46
2.5.6.	DataMart	47
2.5.7.	Pruebas	48
2.5.8.	Despliegue	48
2.5.9.	Capacitación de Usuario	48
2.6.	PRODUCTOS ESPERADOS	48
2.7.	ALCANCES E IMPACTOS ESPERADOS	49
2.7.1.	Impactos esperados	49
3.	DESARROLLO DE LA SOLUCION	52
3.1.	Requerimientos Funcionales	52
3.1.1.	Objetivos	52
3.1.2.	Análisis de Requerimientos	54
3.1.3.	Modelo conceptual	56
3.1.4.	Modelo Lógico	64
3.1.5.	Restricciones y/o consideraciones	65
3.2.	REQUERIMIENTOS TECNICOS Y DE ACCESO	66
3.2.1.	Objetivo	66
3.2.2.	Requerimientos de Hardware	66
3.2.3.	Requerimientos del Servidor	66
3.2.4.	Requerimientos del Cliente	67

3.2.5.	Componentes Técnicos.....	68
3.3.	ARQUITECTURA Y ESTRUCTURA DEL DATAMART	70
3.3.1.	Objetivos	70
3.3.2.	Arquitectura del DataMart	70
3.4.	DISEÑO FÍSICO DEL DATAMART	78
3.4.1.	Objetivos	78
3.4.2.	Diseño Físico	79
3.4.3.	Construcción de la MetaData para el Modelo Físico	87
3.5.	CONSTRUCCION ETL.....	95
3.5.1.	Objetivos	95
3.5.2.	Metodología	95
4.6.	DOCUMENTACIÓN OLAP DEL CUBO DE VENTAS	113
4.6.1.	Objetivos	113
4.6.2.	Documentación OLAP del Cubo de Ventas	113
4.7.	PRUEBAS	132
4.7.1.	Pruebas de Integración	132
4.7.2.	Pruebas Funcionales.....	134
4.8.	PLAN DE CAPACITACION	134
4.8.1.	Objetivo General	134
4.8.2.	Objetivos Específicos	134
4.8.3.	Población Objetivo	134
4.8.4.	Contenido.....	134
5.	RESULTADOS Y CONCLUSIONES.....	139
5.1.	RESULTADOS	139
5.2.	CONCLUSIONES.....	140

6.	BIBLIOGRAFIA	142
----	--------------------	-----

LISTA DE LAS FIGURAS

Figura 1. Modelo dimensional	20
Figura 2. Modelo Estrella.	21
Figura 3. Modelo Copo de Nieve.	21
Figura 4. Fuente de Datos.	23
Figura 5. Proceso ETL.....	26
Figura 6. Almacenes de Datos.....	27
Figura 7. Herramientas de Explotación.....	30
Figura 8. Modelo Conceptual Del DataMart de Ventas.....	56
Figura 9. Modelo Lógico Dimensional del DataMart de Ventas.	64
Figura 10. Componentes de la Arquitectura del DataMart de Ventas.....	71
Figura 11. Metadatos de la Dimensión Cliente del DataMart de Ventas.....	74
Figura 12. Metadatos de la Dimensión Fecha del DataMart de Ventas.....	75
Figura 13. Metadatos de la Dimensión Agencia y sus Dimensiones RolePlaying en el DataMart de Ventas.	76
Figura 14. Metadatos de la Dimensión Unidad de Cargue del DataMart de Ventas.....	76
Figura 15. Metadatos de la Dimensión Tipo de Transporte del DataMart de Ventas.....	76
Figura 16. Metadatos de la Dimensión Ciudad y sus Dimensiones RolePlaying en el DataMart de Ventas.	77
Figura 17. Metadatos de la Dimensión Vehículo del DataMart de Ventas.	77
Figura 18. Metadatos de la Dimensión Tráiler del DataMart de Ventas.....	77
Figura 19. Metadatos de la Tabla de Hechos del DataMart de Ventas.....	78
Figura 20. Modelo Físico del DataMart de Ventas.....	87
Figura 21. Arquitectura de Alto Nivel del proceso ETL	95

Figura 22. Proceso ETL para borrar los datos de las tablas del DataMart de Ventas.....	96
Figura 23. Proceso ETL para cargar la tabla dwd_Fecha del DataMart de Ventas.....	97
Figura 24. Proceso ETL para cargar la tabla dwd_Cliente del DataMart de Ventas.....	98
Figura 25. Proceso ETL para cargar la tabla dwd_Agencia del DataMart de Ventas.....	99
Figura 26. Proceso ETL para cargar la tabla dwd_UnidadCargue del DataMart de Ventas.....	100
Figura 27. Proceso ETL para cargar la tabla dwd_TipoTransporte del DataMart de Ventas.....	101
Figura 28. Proceso ETL para cargar la tabla dwd_Ciudad del DataMart de Ventas.....	102
Figura 29. Proceso ETL para cargar la tabla dwd_Vehiculo del DataMart de Ventas.....	103
Figura 30. Proceso ETL para cargar la tabla dwd_Trailer del DataMart de Ventas.....	104
Figura 31. Proceso ETL para cargar la tabla dwd_Ventas del DataMart de Ventas.....	105
Figura 32. Estructura de directorios del proceso de Carga de la ETL.....	106
Figura 33. Interface para Ejecutar Procedimiento Almacenado SpCargarETLVentas.....	110
Figura 34. Definición de Job en el Agente SQL Server para automatizar el Proceso de carga de la ETL de Ventas.	111
Figura 35. Parametrización básica del Job CargaETLVentas.....	111
Figura 36. Configuración del Comando SQL del Job CargaETLVentas.	112
Figura 37. Automatización del Job CargaETLVentas.	112
Figura 38. Estructura de Solución del Cubo OLAP de Ventas.....	113
Figura 39. Vista del Origen de datos de DWVentas.....	116
Figura 40. Grafico de Dependencias de la Dimensión Dwd Agencia.....	118

Figura 41. Grafico de Dependencias de la Dimensión Dwd Ciudad	120
Figura 42. Grafico de Dependencias de la Dimensión Dwd Cliente.....	121
Figura 43. Grafico de Dependencias de la Dimensión Dwd Fecha.....	124
Figura 44. Grafico de Dependencias de la Dimensión Dwd Tipo Transporte .	125
Figura 45. Grafico de Dependencias de la Dimensión Dwd Trailer.....	126
Figura 46. Grafico de Dependencias de la Dimensión Dwd Vehiculo	128
Figura 47. Grafico de Dependencias de la Dimensión Dwd Unidad Cargue...	128
Figura 48. Grafico de Dependencias de la Dimensión Dwd Unidad Cargue...	132
Figura 49. Escenario para Pruebas de Integración.....	133

LISTA DE TABLAS

Tabla 1. Fases Metodológicas.....	43
Tabla 2. Requerimientos del Servidor.....	67
Tabla 3. Requerimientos del Cliente.....	68
Tabla 4. Requerimientos de Software.....	70
Tabla 5. Dimensión Fecha.....	80
Tabla 6. Dimensión Cliente.....	80
Tabla 7. Dimensión Agencia.....	81
Tabla 8. Dimensión Unidad de Cargue.....	81
Tabla 9. Dimensión Tipo de Transporte.....	81
Tabla 10. Dimensión Ciudad.....	82
Tabla 11. Dimensión Vehículo.....	82
Tabla 12. Dimensión Tráiler.....	83
Tabla 13. Dimensión Role-Playing Agencia Despacho.....	84
Tabla 14. Dimensión Role-Playing Agencia Venta.....	84
Tabla 15. Dimensión Role-Playing Ciudad Origen.....	85
Tabla 16. Dimensión Role-Playing Ciudad Destino.....	85
Tabla 17. Tabla de Hechos - Ventas.....	86
Tabla 18. Atributos Dimensión Fecha.....	88
Tabla 19. Atributos Dimensión Cliente.....	89
Tabla 20. Atributos Dimensión Agencia.....	90
Tabla 21. Atributos Dimensión Unidad Cargue.....	90
Tabla 22. Atributos Dimensión Tipo Transporte.....	91
Tabla 23. Atributos Dimensión Ciudad.....	91
Tabla 24. Atributos Dimensión Vehículo.....	92
Tabla 25. Atributos Dimensión Trailer.....	93
Tabla 26. Atributos Tabla Hechos - Ventas.....	95
Tabla 27. Propiedades del Origen de Datos DWVentas.....	114
Tabla 28. Propiedades de la Vista de Origen de Datos DWVentas.....	115

Tabla 29. Esquema de la Tabla de Hechos - Ventas	115
Tabla 30. Dimensiones MOLAP	117
Tabla 31. Esquema de la Dimensión Agencia	118
Tabla 32. Jerarquías de la Dimensión Agencia	118
Tabla 33. Tabla de Convenciones de Colores	119
Tabla 34. Esquema de la Dimensión Ciudad	119
Tabla 35. Jerarquía de la Dimensión Ciudad	119
Tabla 36. Esquema de la Dimensión Cliente	120
Tabla 37. Esquema de la Dimensión Fecha	123
Tabla 38. Jerarquía de la Dimensión Fecha	124
Tabla 39. Esquema de la Dimensión Tipo Transporte	125
Tabla 40. Esquema de la Dimensión Trailer	126
Tabla 41. Esquema de la Dimensión Trailer	126
Tabla 42. Esquema de la Dimensión Vehículo	127
Tabla 43. Esquema de la Dimensión Vehículo	127
Tabla 44. Esquema de la Dimensión Unidad Cargue	128
Tabla 45. Propiedades del Cubo	129
Tabla 46. Dimensiones del Cubo OLAP	130
Tabla 47. Grupo de Medidas	130
Tabla 48. Medidas del Cubo OLAP	131

LISTA DE ANEXOS

- **ANEXO 1. Acta de Reunión Requerimientos TDM**
- **ANEXO 2. Pruebas Funcionales**

1. INFORMACIÓN GENERAL DEL PROYECTO

1.1. TÍTULO

Construcción del modelo de ventas multidimensional basado en la información histórica de la organización TDM Transportes S.A.S.

1.2. INTEGRANTES

Nombre: Danne Ramirez Zapata
Código: 71799741
Cédula: 71799741
Programa Académico: Especialización en Gestión de Información y Bases de Datos
Teléfono: 3136616567
Correo: danneramirez@hotmail.com
Nivel (semestre): 3

Nombre: Harold D. Jiménez Arbeláez
Código: 71758093
Cédula: 71758093
Programa Académico: Especialización en Gestión de Información y Bases de Datos
Teléfono: 3167418449
Correo: harolddja@hotmail.com
Nivel (semestre): 3

1.3. TIPO DE INVESTIGACIÓN

Exploratoria		Predicativa	
Descriptiva		Proyectiva	
Comparativa	X	Interactiva	
Analítica		Confirmativa	
Explicativa	X	Evaluativa	

1.4. NIVEL DE INVESTIGACIÓN

Exploratoria	
Descriptiva	
Explicativa	X

1.5. TIPO DE PROYECTO

Investigación en Ciencia básica	
Investigación aplicada	
Desarrollo tecnológico	X

1.6. LÍNEA DE INVESTIGACIÓN USB

Línea y grupo de investigación USB	
Trabajo empresarial	Construcción del modelo de ventas multidimensional basado en la información histórica de la organización TDM Transportes S.A.S.

2. DESCRIPCION DEL PROYECTO

2.1. PLANTEAMIENTO DE LA PREGUNTA O PROBLEMA DE INVESTIGACIÓN Y SU JUSTIFICACIÓN EN TÉRMINOS DE NECESIDADES Y PERTINENCIA.

TDM Transportes S.A.S. es una empresa de servicios dedicada al transporte de carga a nivel nacional y de los países del grupo andino. Utilizó hasta Febrero de 2011 el Sistema Administrativo de Transporte (SAT) para soportar toda la información transaccional relacionada con el Core del Negocio y proporcionaba el insumo necesario para la construcción y análisis de los indicadores de gestión y medición que apoyaban la toma de decisiones gerenciales.

Debido a las nuevas estrategias planteadas por la organización para impulsar el crecimiento en su gremio a nivel nacional, ha decidido invertir en una plataforma de TI basada en estándares internacionales (SAP) que soporte la planificación de los recursos empresariales y logísticos requeridos.

El cambio de plataforma de TI le ha implicado a la organización redefinir los indicadores de gestión para la medición y control por parte de la gerencia y a su vez apoyarse en la información histórica para realizar comparaciones y proyecciones mediante la dinámica de las variables del negocio y sus comportamientos a través del tiempo.

La ausencia de la información histórica en esta nueva plataforma de TI hace necesario diseñar una solución de inteligencia de negocios que permita conectar, extraer, procesar, transformar y estructurar la información almacenada transaccionalmente en años anteriores hacia un modelo multidimensional, el cual facilite el análisis y permita realizar comparaciones con la información actual proporcionada por la nueva plataforma.

El alcance de la solución tecnológica planteada estará enfocado principalmente en la construcción de un DataMart basado en la información de las ventas históricas relevantes a la línea tradicional de negocio conocida como VENTA DIRECTA (Solicitud de un cliente para transportar su mercancía desde un punto A hasta un punto B), generando las bases fundamentales necesarias para la explotación de la información y posterior elaboración de herramientas para la toma de decisiones.

¿Qué solución informática consideraría TDM Transportes como una herramienta útil para la toma de decisiones basada en las ventas de la organización, que ayude a los directivos a elaborar predicciones, determinar tendencias, analizar comportamientos y realizar estimaciones buscando como fin disminuir los costos y aumentar los ingresos apoyados en información precisa, relevante, oportuna y que ayude a la organización a definir estrategias para mejorar la calidad del servicio que prestan a sus clientes, incrementar la rentabilidad y su participación en el mercado?

2.2. OBJETIVOS

2.2.1. General

Diseñar una solución basada en tecnología de base de datos multidimensionales (DataMart), que permita explotar la información histórica de las ventas realizadas por TDM Transportes.

2.2.2. Específicos

- Definir la metodología para la construcción del DataMart de Ventas.
- Analizar y definir las fuentes de datos que alimentaran el DataMart de Ventas.
- Diseñar y Construir el modelo de datos, el metadatos y la Arquitectura del Modelo Multidimensional para el DataMart de Ventas.

- Estructurar las técnicas de extracción, las rutinas de limpieza, los procesos de transformación y los flujos de carga de los datos históricos que conformara el DataMart de Ventas a desarrollar para la organización.
- Explotar la información de ventas basado en un escenario específico definido por la organización.

2.3. MARCO TEÓRICO

Los primeros sistemas de información en la década de los 80's empezaron a desarrollarse utilizando el modelo relacional y la información almacenada en las bases de datos generalmente ha sido orientada al registro de transacciones, lo que comúnmente se conoce como sistemas OLPT. Como su nombre lo dice este tipo de sistemas están orientados exclusivamente a generar información a través de transacciones y no a la consulta y análisis de la información, ya que al aumentar el volumen de información en los sistemas transaccionales se dificulta la consulta de los datos generados. Como alternativa a esta situación surgió el concepto de DataWarehouse (DW) o Almacenes de datos como lo define Ralph Kimball *“una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis”* o *“la unión de todos los DataMarts de una entidad”*. [4]

El objetivo primordial de un DataWarehouse es almacenar los datos de tal manera que se facilita la extracción y consulta de los mismos sin importar el amplio volumen de información que pueda existir. Normalmente el alcance que tiene un DataWarehouse involucra toda la información generada en la empresa, y es por esto, la construcción de un DataWarehouse requiere una inversión en tiempo y esfuerzo considerable. Una estrategia o concepto alternativo al DataWarehouse que tiene el mismo fin pero con un alcance más limitado a un área o departamento de empresa es el DataMart. *“Una versión especial del DataWarehouse”*. Son subconjuntos de datos con el propósito de

ayudar a que un área específica dentro del negocio pueda tomar mejores decisiones.

Los datos existentes en este contexto pueden ser agrupados, explorados y propagados de múltiples formas para que diversos grupos de usuarios realicen la explotación de los mismos de la forma más conveniente según sus necesidades. (Wikipedia, 2010).

En los tiempos actuales las empresas necesitan depositar toda su confianza en la toma de decisiones, para lo cual se requieren fuentes de información fiables y oportunas, las cuales brinden a los empleados, líderes de sección, jefes administrativos, ejecutivos y también entes externos a la empresa (como organismos gubernamentales, bancos, fondos financieros, etc.) la facilidad de compartir, gestionar, procesar y utilizar los datos generados sobre toda la información que es procesada y almacenada por los sistemas de información de la compañía como fuente principal de apoyo a la toma de decisiones, marco del estado actual e indicador de los posibles estados futuros; para esto las empresas pueden valerse de los DataWarehouse.

Este proyecto pretende enfocarse en la implementación de una DataMart para un tema prioritario y de gran importancia para la organización, las ventas, las cuales son particularmente ventanas de información hacia los estados financieros de la compañía y factor primordial para la planificación de recursos empresariales y toma de decisiones acertadas para el logro de los objetivos de la organización.

Para tener un mejor control de la información y lograr el desarrollo del DataMart de Ventas es necesario contar con una base teórica dirigida hacia qué es un DataMart y cuáles son los componentes principales en la creación de un DataMart.

2.3.1. Definiciones de DataMart

Es un pequeño DataWareHouse, para un determinado número de usuarios, para un área funcional, específica de la compañía. También se puede definir que un DataMart es un subconjunto de una bodega de datos para un propósito específico.

DataMart es un modelo multidimensional basado en tecnología OLAP que representa a un área específica de la empresa, incluyendo las variables claves y los indicadores para el proceso de toma de decisiones.

Es una base de datos orientada a un tema específico. En otras palabras es un subconjunto del DataWareHouse Corporativo.

Un DataMart es una base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica. Se caracteriza por disponer la estructura óptima de datos para analizar la información al detalle desde todas las perspectivas que afecten a los procesos de dicho departamento. Un DataMart puede ser alimentado desde los datos de un DataWareHouse, o integrar por sí mismo un compendio de distintas fuentes de información.

Tomado de la fuente <http://todotecnology.blogspot.com/2009/09/datamart.html>

2.3.2. Modelo Multidimensional

Es considerado una de las técnicas favoritas en la construcción de un Data Warehouse, en la cual se constituyen modelos de tablas y relaciones con el propósito de optimizar la toma de decisiones, con base en las consultas hechas en una base de datos relacional que están ligadas con la medición o un conjunto de mediciones de los resultados de los procesos de negocio.

El modelo Dimensional (Dimensional Modeling) es una técnica de diseño lógico que tiene como objetivo presentar los datos dentro de un marco de trabajo

estándar e intuitivo, para permitir su acceso con un alto rendimiento. Cada Modelo Dimensional está compuesto por una tabla con una llave combinada, llamada tabla de hechos, y un conjunto de tablas más pequeñas llamadas tablas de dimensiones. Los elementos de estas tablas se pueden definir de la siguiente manera:

Hechos:

Colección de piezas de datos y datos de contexto. Cada hecho representa una parte del negocio, una transacción o un evento.

Los Hechos son aquéllos datos (los datos de la acción) que proporcionan una información cuantitativa sobre las características del Negocio que requiere analizar (Precios, Volumen, Ventas, etc.).

Su finalidad es proporcionar información necesaria para la gestión, facilitando el conocimiento del Negocio o Proceso a modelar, y fundamentar, entre otras, la toma de decisiones, facilitar los procesos de marketing (ofertas y promociones), fidelizar clientes, valorar el desempeño de los trabajadores, etc.

Dimensiones:

Una dimensión es una colección de miembros, unidades o individuos del mismo tipo. Son las áreas temáticas o sujetos del negocio. Proveen un método general para organizar la información corporativa.

Las Dimensiones buscan determinar un contexto para el análisis de los Hechos. Se trata de grupos homogéneos de elementos, en muchas ocasiones, jerarquizados. Su papel es promocionar la información contenida en los Hechos. Algunas dimensiones son Tiempo, Geografía, Clientes, Productos, etc.

Medidas:

Son atributos numéricos de un hecho que representan el comportamiento del negocio relativo a una dimensión. Cruzan todas las dimensiones en todos los niveles.

Una medida es una columna cuantitativa, numérica, en la tabla de hechos. Las medidas representan los valores que son analizados y éstos son las bases a través de las cuales el usuario puede realizar cálculos.

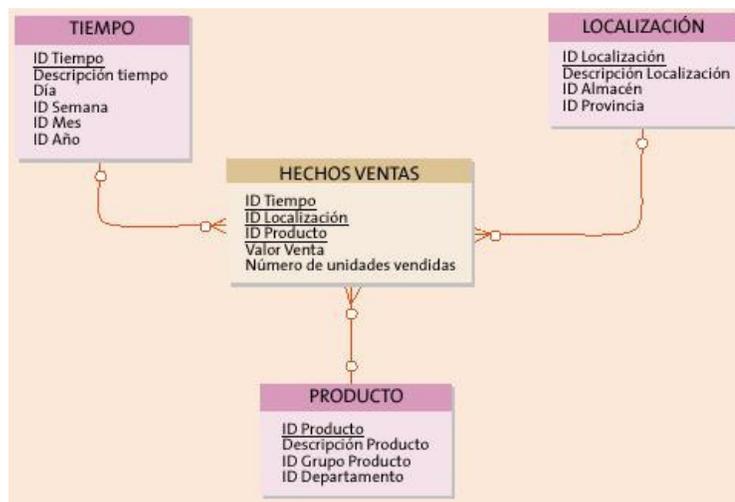


Figura 1. Modelo dimensional

Fuente: <http://www.profinmexico.com/boletines/JUN03.htm>

Fecha Consulta: Agosto 14 de 2011.

Dado lo anterior, se deriva que cada punto de entrada a la tabla de hechos está conectado a una dimensión, lo que permite determinar el contexto de los hechos.

Una base de datos dimensional se puede concebir como estructura de tres o cuatro dimensiones (OLAP), en el que los usuarios pueden acceder a una porción de la base de datos a lo largo de cualquiera de sus dimensiones.

Dado que es muy común representar a un modelo dimensional como una tabla de hechos rodeada por las tablas de dimensiones, frecuentemente se le denomina también como modelo estrella o esquema de estrella-uniión.

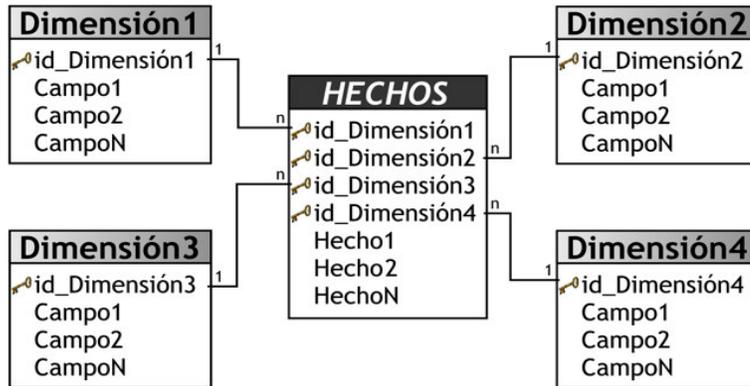


Figura 2. Modelo Estrella.

Fuente: <http://www.dataprix.com/datawarehouse-manager>

Consultado en Agosto 21 de 2011.

Otra variante es la que se conoce como Copo de Nieve (SnowFlake), en donde se presentan ramificaciones a partir de las tablas de dimensiones y no sólo a partir de la tabla de hechos. [5]

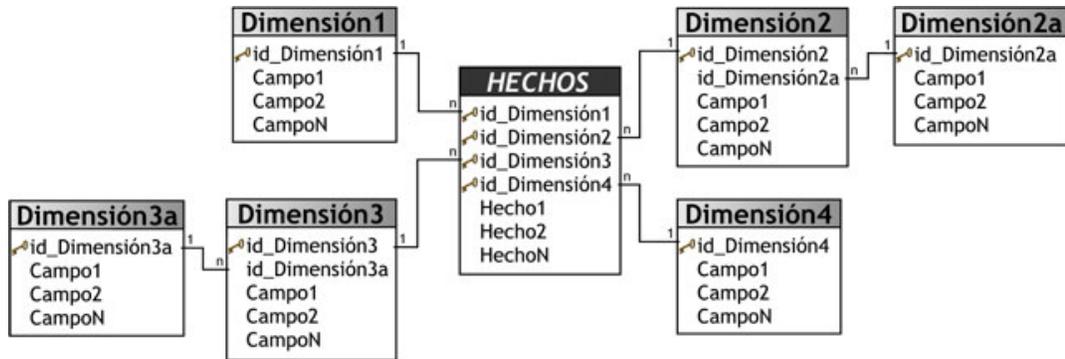


Figura 3. Modelo Copo de Nieve.

Fuente: <http://www.dataprix.com/datawarehouse-manager>

Consultado en Agosto 14 de 2011.

2.3.3. Componentes en la creación de un DataMart

Fuentes de Datos

Son los medios que alimentan de información al DataWareHouse o DataMart y están diseñadas para registrar grandes cantidades de transacciones. Entre ellas se tiene la base de datos OLTP (Una base de datos para soportar procesos transaccionales).

Características:

- Son pobladas por usuarios finales.
- Se optimizan en función a procesos transaccionales.
- Se actualizan constantemente.
- Contienen mucha información de detalle.

OLTP:

“Una base de datos para soportar procesos transaccionales en línea (OLTP), puede no ser adecuada para el DataWareHouse ya que ha sido diseñada para maximizar la capacidad transaccional de sus datos y típicamente tiene cientos de tablas la gran mayoría normalizadas. Su diseño también ha sido condicionado por los procesos operacionales que deberá soportar para la óptima actualización de sus datos, normalmente muchas de sus tablas en constantes y continuos cambios. Los sistemas DataWareHouse están orientados a procesos de consultas en contraposición con los procesos transaccionales.”¹

¹ Principios y técnicas prácticas para la implementación de un prototipo de Datawarehouse. Mideros Romero. 2007.

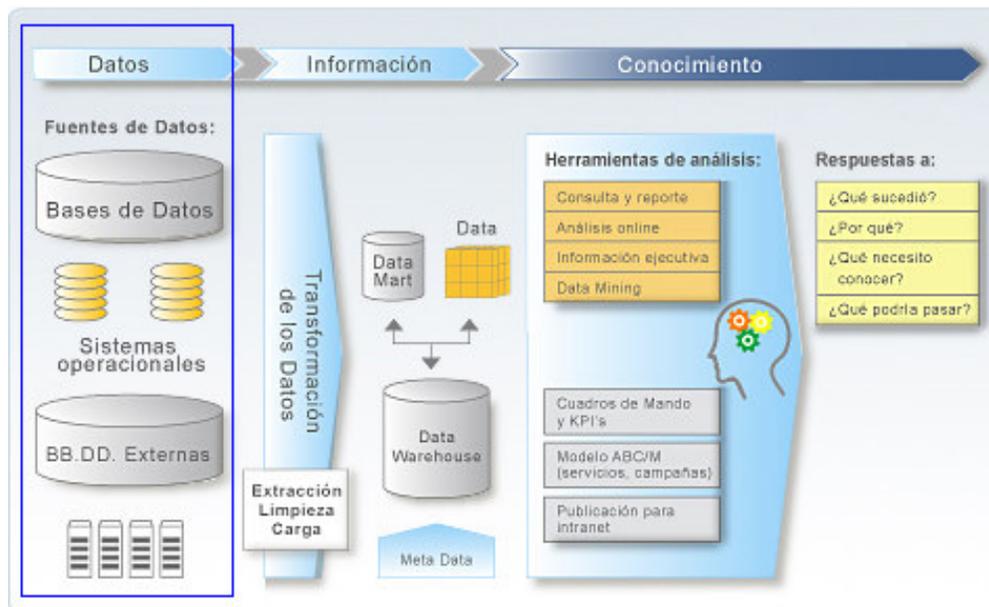


Figura 4. Fuente de Datos.

Fuente: <http://www.logica.com.es/web/guest/sistemas-datawarehouse>

Consultado en Agosto 21 de 2011.

2.3.4. Procesos de extracción, transformación y carga de datos (ETL)

ETL es el proceso que organiza el flujo de los datos entre diferentes sistemas en una organización y aporta los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos, DataMart o bodega de datos. ETL forma parte de la inteligencia empresarial (Business Intelligence), también llamado “Gestión de los datos” (Data Management). La idea es que una aplicación ETL lea los datos primarios de unas bases de datos de sistemas principales, realice transformación, validación, el proceso cualitativo, filtración, y al final escriba datos en el almacén y en este momento los datos son disponibles para analizar por el usuario.

Extracción

Se refiere al mecanismo por medio del cual los datos son leídos desde su fuente original.

Transformación

Es la etapa por la que puede atravesar una base de datos para estandarizar los datos de las distintas fuentes, normalizando y fijando una estructura para los datos.

Carga

Consiste básicamente en llevar los datos leídos y estandarizados a la bodega dimensional (puede ser remota o localmente). Si los requerimientos de datos incluyen datos históricos es necesario desarrollar tres conjuntos de programas de carga: una carga Inicial, una carga histórica, y una carga incremental.

- **Carga Inicial:** La carga inicial se asemeja mucho al proceso de conversión entre sistemas que se da en las organizaciones cuando pasan, por ejemplo, de sus viejos sistemas operacionales a un producto ERP, y consiste en la extracción, transformación y carga de los datos.
- **Carga Histórica:** Este proceso debe verse como una extensión de la carga inicial, pero la conversión aquí es un poco diferente porque los datos históricos son datos estáticos. A diferencia de los datos operacionales, los datos estáticos comúnmente ya se archivaron en dispositivos de almacenamiento offline. Es común que con el transcurso del tiempo se eliminen elementos de datos que ya no sirven, se agreguen nuevos, se modifiquen los tipos de ciertos datos o los formatos de los registros, lo que implica que los datos históricos no necesariamente se puedan sincronizar con los datos operacionales. Por lo tanto, los programas de conversión escritos para la carga inicial quizás no sean aplicables a la carga de datos históricos sin algunos cambios previos.

- **Carga Incremental:** Una vez que el almacén de datos está cargado con datos iniciales e históricos, hay que desarrollar otro proceso para la carga incremental, que se ejecutara mensual, semanal o diariamente. Existen dos formas de diseñar la carga incremental:
 1. Extraer todos los registros. Se extraen todos los registros operacionales, independientemente de los valores que hayan cambiado desde la última carga realizada. En general esta opción no es viable debido al volumen de los datos, por eso la mayoría opta por extraer deltas.
 2. Extraer Deltas solamente. Sólo se extraen registros nuevos o registros que contengan valores que cambiaron desde la última carga realizada. Diseñar programas para extraer deltas es más fácil cuando las fuentes consisten en bases de datos relacionales y se cuenta con una columna Timestamp para determinar los deltas.

“Los mismos elementos de datos, si son usados por aplicaciones diferentes o administrados por diferentes software DBMS (Sistemas Manejadores de Bases de Datos), pueden definirse al usar nombres de elementos inconsistentes, que tienen formatos inconsistentes y/o ser codificados de manera diferente. Todas estas inconsistencias deben resolverse antes que los elementos de datos sean almacenados en el DataWareHouse.

Uno de los desafíos de cualquier implementación de DataWareHouse, es el problema de transformar los datos. La transformación se encarga de las inconsistencias en los formatos de datos y la codificación, que pueden existir dentro de una base de datos única y que casi siempre existen cuando múltiples bases de datos contribuyen al DataWareHouse.

La transformación de datos también se encarga de las inconsistencias en el contenido de datos. Una vez que se toma la decisión sobre que reglas de

transformación serán establecidas, deben crearse e incluirse las definiciones en las rutinas de transformación.”²

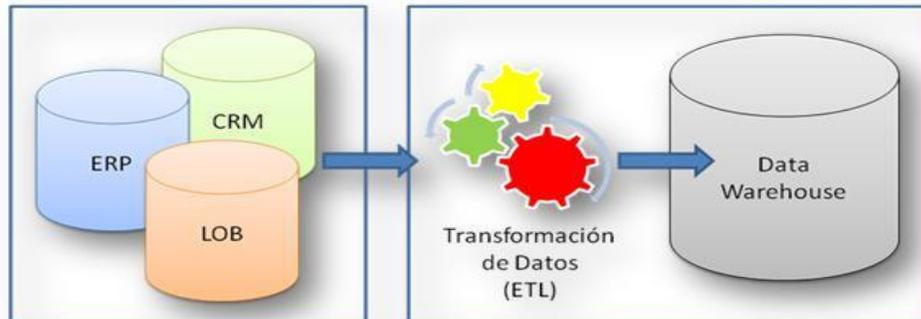


Figura 5. Proceso ETL

Fuente: <http://delfos.bligoo.com/content/view/195902/Data-Warehouse-Data-mart-y-Data-Mining.html>

Consultado en Agosto 21 de 2011.

2.3.5. DataWareHouse

Un DataWareHouse contiene la información de toda la empresa. Cualquier departamento puede acceder a la información de cualquier otro departamento mediante un único medio, así como obligar a que los mismos términos tengan el mismo significado para todos.

Un DataMart almacena la información de un área o departamento específico y un conjunto de DataMarts forman un DataWareHouse

Un DataMart es una solución que, compartiendo tecnología con el DataWareHouse (pero con contenidos específicos, volumen de datos más limitado y un alcance histórico menor), permita dar soporte a una empresa pequeña, un departamento o área de negocio de una empresa grande.

El DataMart cubre de manera óptima las necesidades de informes. No es conveniente efectuar consultas sobre los sistemas transaccionales, debido a que hay que integrar datos de diversas OLTP.

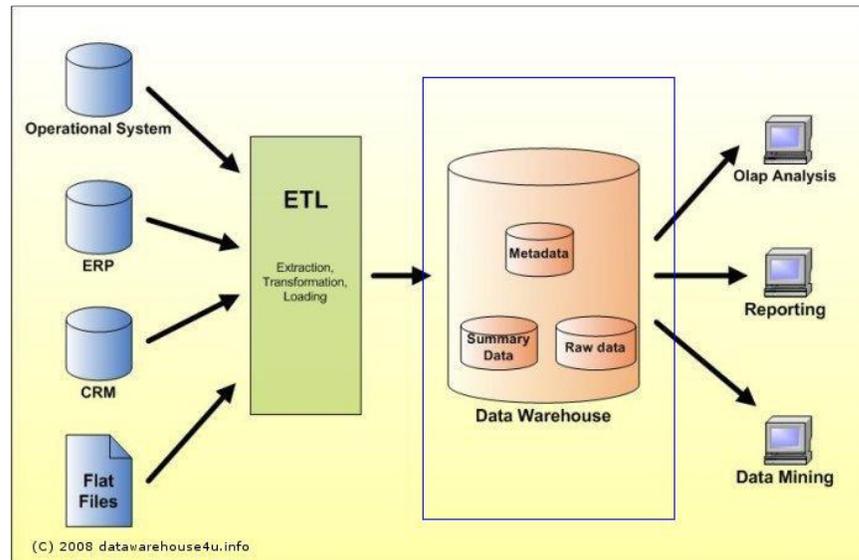


Figura 6. Almacenes de Datos.
Fuente: <http://tecnoxml.wikidot.com/oracle>
Consultado en Agosto 21 de 2011.

2.3.6. Herramientas de Explotación

El DataMart está orientado a la toma de decisiones. Un buen diseño de la base de datos favorece el análisis y la recuperación de datos para obtener una ventaja estratégica y para facilitar la toma de decisiones.

El DataMart no está orientado a procesos relacionados con la operatividad de la empresa. El DataMart está preparado para ser explotado mediante herramientas específicas que permiten la extracción de información significativa y patrones de comportamiento que permanecen ocultos en un enorme repositorio de datos.

Entre las herramientas software que existen:

- **Herramienta de consulta y reporte**

Las herramientas de consulta permiten especificar los elementos de datos, condiciones, criterios de agrupación y otros atributos de una solicitud de información. La herramienta de consulta genera entonces un llamado a una base de datos, extrae los datos pertinentes, efectúa cálculos adicionales, manipula los datos si es necesario y presenta los resultados en un formato claro.

El procesamiento estadístico se limita comúnmente a promedios, sumas, desviaciones estándar y otras funciones de análisis básicas. Aunque las capacidades varían de un producto a otro, las herramientas de consulta y reporte son más apropiadas cuando se necesita responder a la pregunta ¿"Qué sucedió"?

- **Herramientas de base de datos multidimensionales / OLAP**

Las Herramientas OLAP (Procesamiento analítico en línea), permiten efectuar Análisis Multidimensional de los Datos, extrayendo información de los Repositorios de Datos Corporativos, de cara a soportar consultas Ad Hoc adecuadas al Proceso de Toma de Decisiones. Estas Herramientas tienen la particularidad de trabajar bajo un lenguaje de negocios, abstrayendo al usuario de la misma, de cualquier aspecto técnico. Con esto, se facilita la incorporación de los ejecutivos, en la adquisición de la información deseada, mediante su propia gestión evitando intermediarios en el proceso de abastecimiento de la información.

Las Herramientas OLAP, implican un modelado de datos diferente al tradicional, optimizado para consultas de gran envergadura. Este nuevo modelo de estructura de datos, se nutre a partir de las Bases de Datos Transaccionales de la empresa. En este modelo, 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 formular consultas complejas, arreglar datos en un reporte, cambiar de datos resumidos a datos detallados y filtrar o rebanar los datos en subconjuntos significativos.

La tecnología OLAP permite un uso más eficaz de los almacenes de datos para el análisis en línea, lo que proporciona respuestas rápidas a consultas analíticas complejas e iterativas. Los modelos de datos multidimensionales de OLAP y las técnicas de agregados de datos organizan y resumen grandes cantidades de datos para que puedan ser evaluados con rapidez mediante el análisis en línea y las herramientas gráficas. Los sistemas OLAP proporcionan la velocidad y la flexibilidad necesarias para dar apoyo al analista en tiempo real.

- **Herramientas de DataMining**

El DataMining es el proceso que ayuda a descubrir los patrones y relaciones que puedan pasar desapercibidos en el análisis del negocio y de los clientes. Debe estar orientado a resolver un problema de negocios, y no debe requerir el ser un especialista en estadística para poder emplearlo. El proceso de Data Mining extrae los conocimientos guardados o información predictiva desde el DataWareHouse sin requerir pedidos o preguntas específicas. Las herramientas Mining usan algunas de las técnicas de computación más avanzadas para generar modelos y asociaciones como redes neurales, detección de desviación, modelamiento predictivo y programación genética. DataMining es un dato conducido, no una aplicación-conducida.

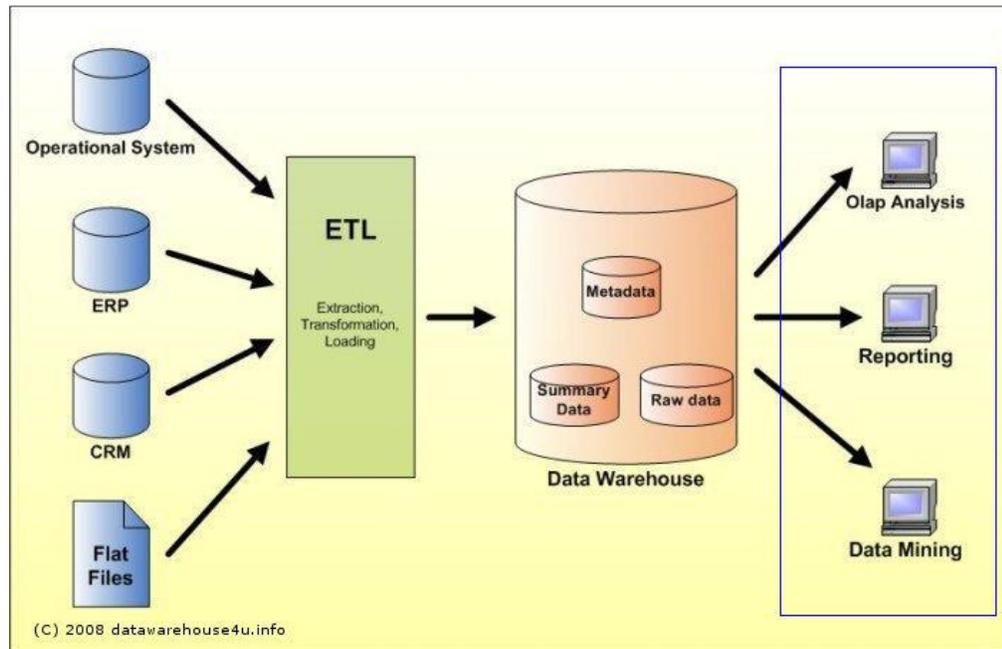


Figura 7. Herramientas de Explotación.
Fuente: <http://tecnoxml.wikidot.com/oracle>
Consultado en Agosto 21 de 2011.

- **Cubos OLAP**

El objeto principal dentro de la tecnología OLAP es el cubo. Un cubo es una representación multidimensional de un conjunto de datos, conteniendo detalle y datos de resumen. Proporcionan un mecanismo de consulta de datos con un tiempo de respuesta corto y uniforme sin importar la cantidad de datos almacenados en el cubo o la complejidad de la consulta. Una base de datos OLAP se conforma de uno o varios cubos, dependiendo de las necesidades de información en la organización y cada uno de éstos conforma el almacén de datos; Por ejemplo el cubo de Ventas, el cubo de Compras, etc.

Los componentes principales en la construcción de un cubo son las dimensiones, las medidas, el origen de datos y las particiones.

El origen de datos del cubo identifica y se conecta con la base de datos donde se encuentra el almacén de datos, que es el origen de los datos del cubo.

Una medida representa un valor, normalmente numérico, sobre un hecho específico que proporciona resúmenes en diferentes niveles de agregación dentro del contexto de la bodega de datos. Por ejemplo, el flete cobrado a un cliente o la cantidad despachada de un producto; mientras que una dimensión es la manera en la que se resumen los valores numéricos; por ejemplo: Tiempo, localización, productos, entre otros.

Las particiones son los contenedores multidimensionales de almacenamiento que guardan los datos del cubo. Cada cubo contiene, al menos, una partición y los datos de un cubo se pueden combinar a partir de varias particiones. Cada partición puede obtener sus datos de un origen de datos distinto y se pueden almacenar en una ubicación diferente. Los datos de una partición se pueden actualizar independientemente de los datos contenidos en otras particiones del mismo cubo. Por ejemplo, los datos de un cubo se pueden dividir mediante criterios de tiempo, con una partición para los datos del año actual, otra partición para los datos del año anterior y una tercera partición para los datos más antiguos. Las particiones de un cubo se pueden almacenar separadas en distintos modos de almacenamiento con diferentes grados de resumen.

- **Medidas**

Las medidas son datos numéricos de gran interés para los usuarios de los cubos. Las medidas que se seleccionen dependerán de los tipos de

información que solicitan los usuarios. Algunas medidas comunes son ventas, costos, gastos, cuenta de producción, etc.

Cada medida se almacena en una columna de la tabla de hechos en el almacén de datos. Debido a que un cubo puede contener únicamente una tabla de hechos, todas las medidas del cubo deben estar contenidas en ella.

Una medida puede contener varias columnas combinadas en una expresión. Por ejemplo, la medida Beneficios es la resta de dos columnas numéricas: Ventas y Costos.

Las columnas de la tabla de hechos pueden ser aditivas o no aditivas. Las columnas aditivas se pueden sumar. Por ejemplo, una columna monetaria es aditiva. Las columnas aditivas son apropiadas para utilizarse como medidas de un cubo. La suma de columnas no aditivas carece de significado. Por ejemplo, una columna numérica que contiene un identificador como Número de cuenta no es aditiva. Las columnas no aditivas no son apropiadas como medidas de un cubo, pero se pueden combinar con ayuda de ciertas funciones como Count (Contar). El resultado se puede utilizar después como una medida.

Para aplicar una función a una columna no aditiva de forma que el resultado se pueda utilizar como una medida, se debe crear un miembro calculado.

Los miembros calculados se pueden utilizar como medidas. Los valores de los miembros calculados se crean mediante fórmulas cuando se examina el contenido del cubo, pero estos valores no se almacenan. De esta forma, los miembros calculados ahorran espacio de almacenamiento en disco.

- **Dimensiones**

Las dimensiones son categorías descriptivas que se utilizan para facilitar el análisis de los datos numéricos (es decir, las medidas) contenidos en un cubo. Por ejemplo, si una medida de un cubo es Cuenta de producción y sus dimensiones son Fecha, Fábrica, Ubicación y Producto, los usuarios del cubo pueden dividir Cuenta de producción en varias categorías de Fecha, Fábrica, Ubicación y Producto.

Se puede crear una dimensión para utilizarla en un sólo cubo o en varios cubos. Una dimensión creada para un cubo individual se denomina dimensión privada. Una dimensión que varios cubos pueden utilizar recibe el nombre de dimensión compartida.

Las dimensiones compartidas pueden utilizarse en cualquier cubo de la base de datos. Al crear dimensiones compartidas y utilizarlas en varios cubos se ahorra tiempo ya que no se necesita crear las mismas dimensiones privadas en cada uno de los cubos. Las dimensiones compartidas también permiten la estandarización de medidas empresariales utilizadas entre cubos. Por ejemplo, las dimensiones compartidas estandarizadas para fechas y ubicación geográfica aseguran que los datos analizados desde cubos distintos tendrán una organización similar.

Cuando se defina una dimensión, se debe seleccionar una o más columnas de una tabla de dimensiones. Si selecciona varias columnas, todas ellas deben estar relacionadas entre sí de tal forma que sus valores se puedan organizar en una única jerarquía.

Cada columna de una dimensión constituye un nivel en la dimensión. Los niveles se ordenan por nivel de detalle y se organizarán en una

jerarquía que permita la creación de caminos lógicos para el aumento del nivel de detalle. Cada nivel contiene miembros. Los miembros son los valores de la columna que define el nivel.

Cada clave principal de una tabla de dimensiones debe combinarse con una clave foránea de la tabla de hechos de un cubo o de otra tabla de dimensiones. Para definir una dimensión no se necesitan las columnas de clave.

- **Modos de almacenamiento (MOLAP, ROLAP, HOLAP)**

Las opciones de almacenamiento físico afectan al rendimiento y a los requisitos de almacenamiento de los cubos. Se puede almacenar un cubo en una estructura MOLAP (OLAP multidimensional), en una base de datos ROLAP (OLAP relacional), o en una combinación HOLAP (OLAP híbrido) de estructura multidimensional y base de datos relacional. Cada estrategia de almacenamiento presenta ventajas y desventajas.

El almacenamiento MOLAP utiliza una estructura multidimensional para contener agregados y una copia de los datos base. El sistema de almacenamiento MOLAP proporciona los tiempos de respuesta a consultas más rápidos, que dependen únicamente del porcentaje y del diseño de los agregados del cubo. En general, MOLAP es más apropiado para cubos de uso frecuente y que necesitan tiempos de respuesta muy cortos.

El almacenamiento ROLAP utiliza tablas en las bases de datos relacionales del almacén de datos para almacenar los agregados de un cubo. A diferencia del almacenamiento MOLAP, ROLAP no almacena una copia de los datos base, sino que tiene acceso a la tabla de hechos

original cuando es necesario para responder a consultas. Las respuestas a consultas ROLAP suelen ser más lentas que aquellas que se realizan con otras dos estrategias de almacenamiento.

Un uso típico de ROLAP es el acceso a grandes conjuntos de datos consultados con poca frecuencia, tales como datos históricos de años no recientes.

El sistema de almacenamiento HOLAP combina atributos de MOLAP y ROLAP. Los datos de los agregados se almacenan en estructuras MOLAP mientras que los datos base se mantienen en la base de datos relacional del almacén de datos. Para las consultas que tienen acceso a los datos de resumen, HOLAP es el equivalente de MOLAP. Las consultas que tienen acceso a datos base, por ejemplo una consulta que aumenta el nivel de detalle hasta un hecho simple, deben recuperar los datos de la base de datos relacional y no se ejecutarán con tanta rapidez como cuando los datos base están almacenados en la estructura MOLAP.

Los cubos almacenados como HOLAP tienen un tamaño menor que los cubos MOLAP equivalentes y responden con mayor rapidez que los cubos ROLAP a consultas relativas a datos de resumen. El almacenamiento HOLAP suele ser adecuado para cubos que requieren tiempos cortos de respuesta para consultas realizadas en resúmenes basados en grandes cantidades de datos base.

- **Particiones**

Los cubos se pueden dividir en particiones, cada una de las cuales se puede almacenar mediante un modo distinto (MOLAP, ROLAP, HOLAP). Las particiones de un cubo son invisibles para el usuario, sin embargo, es importante que las particiones se definan de tal manera que

contengan datos mutuamente exclusivos. Un cubo puede proporcionar resultados incorrectos a algunas consultas si una parte de los datos del cubo está incluida en más de una partición.

Las particiones de un cubo se pueden almacenar en varios servidores para proporcionar un método de almacenamiento en cubos basado en clústeres.

Dos particiones de un cubo pueden mezclarse en una única partición que, a su vez, puede combinarse con otra partición y así sucesivamente hasta que quede una única partición. Por ejemplo, se puede mezclar cuatro particiones, cada una de las cuales contienen datos correspondientes a un trimestre, en una única partición que contenga los datos de todo el año.

- **Agregados**

Los agregados son resúmenes de datos precalculados que mejoran el tiempo de respuesta a las consultas por el simple hecho de tener preparadas las respuestas antes de que se planteen las preguntas. Por ejemplo, la respuesta a una consulta que solicita el total de ventas semanales de una determinada línea de productos y que se realiza en una tabla de hechos de un almacén de datos que contiene cientos de miles de filas de información, puede llevar mucho tiempo si hay que explorar la tabla de hechos para calcular la respuesta. Por el contrario, la respuesta podría ser casi inmediata si los datos de resumen para la respuesta a esta consulta se han calculado previamente. El cálculo previo de los datos de resumen es la clave para obtener respuestas rápidas en la tecnología OLAP. Si se calculan previamente todos los posibles agregados a un cubo, se obtiene el tiempo de respuesta más corto posible para todas las consultas. Sin embargo, el tiempo de

almacenamiento y el tiempo de proceso necesarios para todos los agregados puede ser sustancial.

Los requisitos de almacenamiento no sólo dependen del número de dimensiones y medidas, sino también del número de niveles de las dimensiones y del número de miembros de cada nivel.

Existe un equilibrio entre los requisitos de almacenamiento y el porcentaje de posibles agregados que se han precalculado. Si no se calcula previamente ningún agregado (0%), se necesitará poco espacio de almacenamiento además del necesario para almacenar los datos base. Sin embargo, en este caso, los tiempos de respuesta a consultas pueden variar y pueden llegar a ser muy elevados, ya que será necesario calcular todas las respuestas a partir de los datos de base en cada consulta. [6 - 7]

2.4. REFERENCIA GLOBAL

Hay que resaltar que este tipo de trabajos son realizados bajo un alto nivel de personalización. Esto debido a la naturaleza de los DataMart, ya que son usados como una base de análisis para los negocios. Cada negocio tiene reglas diferentes, pero sobre todo objetivos diferentes. Esto último hace que se tengan que ajustar los diferentes componentes de análisis de la Inteligencia de negocios en cada contexto en el que se usa esta herramienta.

A continuación se describen algunas referencias relacionadas con la implementación de un DataMart para apoyar el análisis estratégico y fortalecer las herramientas para una mejor toma de decisiones:

La Marina de Guerra del Perú – MGP.

La MGP es el órgano de ejecución del Ministerio de Defensa del Perú, encargado de la defensa marítima, fluvial y lacustre del país. Forma parte de las Fuerzas Armadas del Perú y como tal integra el Comando Conjunto de las Fuerzas Armadas del Perú.

La MGP planifica la realización de cursos de calificación y cursos de línea de carrera ya sea a nivel institucional y no institucional, con el fin que el personal superior (oficiales) y subalterno alcancen elevados niveles de preparación para el cumplimiento de las tareas que les puedan ser encomendadas. Ante lo cual se necesita tomar decisiones correctas para poder destacar al personal idóneo a las diferentes divisiones dentro del fuero militar; así mismo se requiere una debida planificación de que cursos se van a poner a disposición del personal de marina en sus diferentes grados.

Para llevar a cabo estos objetivos, se implementó un DataMart de Capacitación, el cual facilita la planificación, el análisis estratégico y así mismo, brinda una adecuada información que permite realizar planes referentes a cursos de capacitación o instrucción que servirán para una adecuada gestión y por ende una acertada toma de decisiones.

Este proyecto ha beneficiado a la Marina de Guerra del Perú en la realización de planes estratégicos para la realización de cursos de instrucción y capacitación, logrando de esta manera un personal debidamente capacitado en las diferentes especialidades. Mediante el DataMart se pudo contar con información consistente y disponible en tiempo real del personal que han llevado cursos ya sea de calificación o carrera. Esto ha permitido una toma de decisiones más rápidas respecto a donde asignar determinado personal. Del mismo modo ha fortalecido la capacidad de realizar comparaciones y obtener tendencias centrando la atención sobre los verdaderos puntos problemas.

El proyecto DATAMART CAPACITACION fue realizado en la Escuela Profesional de Computación y Sistemas, Facultad de Ingeniería y arquitectura, Universidad de San Martín de Porres. [1]

Entidad Aseguradora - Área de Marketing

La empresa aseguradora en la cual se implementó el DataMart de Clientes atravesaba por un proceso de crecimiento en el cual se busca aumentar sus ventas y asegurar la relación que mantiene con sus clientes.

El área de Gestión de Información junto con el área de Marketing implementó una solución integral para gestionar la información orientando el manejo de datos hacia una arquitectura propia de DataWarehouse. Este proyecto busca atender parte de las necesidades más imperantes del área de marketing, de este modo se busca usar la información de manera que se pueda controlar los siguientes aspectos:

- La eficiencia de sus procesos. A través de una gestión de la cobranza más efectiva.
- Retención de los clientes. A través de un seguimiento de clientes que tienen pólizas por renovar.
- Aumento de la adquisición de productos por cliente. Siguiendo el análisis de Cross y Up Selling (Venta Cruzada y Venta mejorada).
- Eficacia en la recuperación de Clientes perdidos (Ex - Clientes).

Dentro de este marco, el uso del DataMart de Clientes en el área de marketing ha permitido a los usuarios contar con la herramienta para monitorear la gestión del negocio y contar con una visión acerca del cumplimiento de sus objetivos. Además, este modelo de solución ha permitido de manera flexible afrontar el problema de duplicación de clientes existente en la organización.

El proyecto DATAMART DE CLIENTES fue realizado en la Escuela de Ingeniería Informática, Facultad de Ciencias e Ingeniería, Pontificia Universidad Católica de Perú. [2].

Instituto de Enfermedades Infecciosas (Infectious Diseases Institute) (IDI)

El instituto de enfermedades infecciosas (IDI) (UGANDA) tenía la gran necesidad de proporcionar acceso fácil y rápido a la información de los pacientes y llevar una pista del rendimiento de la clínica. Además, proporcionar a los médicos, los medios para ver los datos de las visitas previas en el tiempo requerido, poder identificar lo que ha cambiado y ver la condición del paciente actual en un contexto integro.

Para lograr este objetivo se construyó un sistema DataMart basado en web para integrar las diferentes fuentes de datos en un repositorio de datos central, de modo que los administradores ejecutivos monitoreen el rendimiento de la clínica IDI permitiéndoles tomar decisiones precisas y confiables.

Esta solución ha permitido a los médicos acceder a los registros médicos completos, simplificando su trabajo de diagnosticar enfermedades correctamente para evitar pruebas costosas o riesgosas y para diseñar planes efectivos de tratamiento que tengan en cuenta muchos factores de complicación.

El proyecto DATAMART BASADO EN WEB PARA CUIDADO DE PACIENTES (A Web Based Data Mart System For Patient Care) fue desarrollado en Makerere University [3].

En el contexto internacional el desarrollo de DataMarts se lleva en gran escala. Esta herramienta es muy usada en la mayoría de empresas que gestionan su información a

través de una herramienta de inteligencia de negocios. Mencionar los trabajos realizados fuera del país llevaría a enumerar un sinnúmero de trabajos realizados, cada uno

de ellos con estructuras y reglas de negocio propias de cada organización.

2.5. DISEÑO METODOLÓGICO

Para el desarrollo e implementación de este proyecto, se enfoca en los hitos de la metodología planteada Ralph Kimball (experto en el diseño de DataWareHouse) y la cual se encamina principalmente en el diseño de la base de datos que almacenará la información para la toma de decisiones.

Un factor muy relevante en el desarrollo e implantación de un proyecto de DataWarehousing, es la adopción de una forma de trabajo bien estructurada y ordenada; y apoyados en el uso de una metodología reconocida y estandarizada, lograr recortar los tiempos de desarrollo, programar la inversión de recursos de manera eficiente y proporcionar un lenguaje común que permita mejorar la comunicación y productividad en el equipo de trabajo.

Nuestra metodología consta de seis fases:

- Planeación
- Requerimientos y Análisis del Negocio
- Diseño
- Construcción
- Pruebas
- Despliegue

Fase	Etapa	Contenido
PLANEACION	Plan del proyecto	Visión del Negocio
		Equipo de Trabajo
		Cronograma
REQUERIMIENTOS Y ANALISIS DEL	Requerimientos	Reunion(es) para levantamiento de

NEGOCIO		requisitos.
		Revisión Reportes Estadísticos
		Datos Transaccionales
	Análisis Dimensional	Cuadro Dimensiones y Jerarquías
		Cuadro de Medidas - Dimensiones
		Análisis Dimensional Final
DISEÑO	Diseño Dimensional	Definir la granularidad
		Diseñar Dimensiones y Medidas
		Escoger Tabla Hecho
		Documentación del Modelo
CONSTRUCCION	DataMart	Implementar Dimensiones
		Implementar Tabla Hecho
		Crear Relaciones
	ETL (Procesos de Extracción. Transformación y Carga)	Realizar Mapeo
		Diagrama General de Poblamiento.
		Poblamiento de Dimensiones.
		Poblamiento de Tabla

		Hecho.
		Crear Programación de Tareas
PRUEBAS	Creación de Cubo de pruebas	Identificación de Orígenes
		Vistas de la Conexión
		Creación de Cubos
		+ Dimensiones
	+ Tabla Hecho	
	Despliegue del Cubo de pruebas	Aplicación: Usando herramientas de oficina
DESPLIEGUE	Implementación	Capacitación de Usuarios

Tabla 1. Fases Metodológicas.

A continuación se describen las fases que se llevarán a cabo para implementar el DataMart de Ventas en la organización TDM Transportes.

2.5.1. Planeación del DataMart

En esta fase se identifica las etapas requeridas por el proyecto para lograr los objetivos definidos. Así mismo, se define el alcance de los entregables de la implementación.

- **Objetivo del Proyecto**

Se describe de forma clara y precisa los objetivos del proyecto, los cuales definirán el horizonte y servirán como punto de referencia para lograr los resultados esperados.

- **Definición y alcance de la Aplicación**

Se describe en qué consiste la implementación del DataMart de Ventas, el alcance y los beneficios para la organización.

- **Equipo de Trabajo**

Se define claramente los roles y responsabilidades de cada miembro del equipo.

2.5.2. Requerimientos y Análisis del Negocio

El objetivo de esta fase es dar amplitud y profundidad a los requerimientos de la organización y evaluar detalladamente las necesidades técnicas necesarias para lograr una exitosa implementación.

- **Requerimientos Técnicos**

Se elabora un documento detallado con las especificaciones técnicas mínimas requeridas para la implementación del DataMart de ventas de la organización.

- **Requerimientos Funcionales**

Basado en el alcance del proyecto, se identifica y se define las necesidades a nivel funcional requeridas y que apoyaran la toma de decisiones de la organización.

- **Requerimientos de Acceso**

Se define los procedimientos y las funciones necesarias para otorgar acceso al DataMart de Ventas.

- **Arquitectura y Estructura de la Solución**

Se diseña la arquitectura que soportará la solución y sus componentes siguiendo correctamente los procedimientos.

2.5.3. Análisis Dimensional

Cuadro de Tablas Dimensionales

Se define las tablas dimensionales basadas en los requerimientos definidos en la fase de requerimientos.

- **Cuadro de Medidas – Dimensiones**

Se establece la relación medida – dimensión según lo especificado en los requerimientos y el cuadro de dimensiones.

- **Cuadro Tabla de Hechos**

Se elabora un documento definiendo las tablas de hechos a partir de las relaciones medida – dimensiones definidas.

- **Análisis Dimensional Final**

Esquema final del análisis dimensional, se revisará cada punto del análisis y se documentará.

2.5.4. Diseño

En esta fase se define el plan de construcción del DataMart de Ventas basado en las variables identificadas en el análisis.

Tablas Dimensionales

Se diseñan las tablas dimensionales, describiendo detalladamente el diccionario de datos del modelo

- **Tablas de Hechos**

Se diseñan las tablas de Hechos.

- **Crear Relaciones**

Se define y diseñan las relaciones entre las tablas dimensionales y de hechos.

- **Procedimientos Almacenados**

Se define y diseñan los procedimientos que servirán en el proceso de carga del DataMart.

- **Diagrama General**

Se realiza el diagrama general del DataMart.

2.5.5. Construcción

Estrategia de Extracción de Datos – ETL:

Este proceso permitirá mover, limpiar y cargar datos desde diferentes fuentes para poblar el DataMart.

- **Paquetes Carga de Dimensiones**

A través de la herramienta SQL Server Integration Services (SSIS) construimos los diferentes procesos para la carga de dimensiones.

- **Paquetes Carga Tablas de Hechos**

A través de la herramienta SQL Server Integration Services (SSIS) construimos los diferentes procesos para la carga de las tablas de hechos.

- **Paquetes para Procesamiento**

Se construirá los paquetes que servirán para el monitoreo de la carga de datos del DataMart.

2.5.6. DataMart

- **Implementar Tablas Dimensionales**

Se cargarán las dimensiones que fueron diseñadas en la etapa de diseño.

- **Implementar Tabla de Hechos**

Se cargarán la tabla de hechos que fue diseñada en la etapa de diseño.

- **Doblamiento de Dimensiones**

Se establecerán las dimensiones que pertenecen a las tablas de hechos.

- **Doblamiento de Tabla de Hechos**

Se establecerán las principales tablas que se definen para el DataMart.

- **Crear Programación de Tareas**

Se crearán los paquetes de extracción de datos haciendo uso de la herramienta SQL Server Integration Services (SSIS).

2.5.7. Pruebas

Se desarrollarán las pruebas funcionales con los diferentes escenarios establecidos, para garantizar las especificaciones funcionales requeridas por la organización.

Una vez realizadas todas las pruebas exitosamente, se realizará una reunión de validación para obtener aprobación y así poder realizar el despliegue del proyecto en la organización.

2.5.8. Despliegue

Se procede a realizar el despliegue en producción del proyecto.

2.5.9. Capacitación de Usuario

Se desarrolla un plan de formación para socializar y proporcionar al personal el conocimiento necesario para operar y hacer uso correctamente del DataMart.

2.6. PRODUCTOS ESPERADOS

El proyecto tiene como objetivo cumplir con la elaboración de los siguientes entregables:

- Modelo de datos para el DataMart de Ventas, el cual será un documento entregable con la descripción detallada de los almacenamientos orígenes, las entidades clave de datos, la forma en que se utilizaran, como será traídos y transportados los datos, y como los datos serán transformados para asegurar la integridad y la calidad de la información del DataMart.
- Componentes de extracción, transformación y carga para el modelo de datos, desarrollado en aplicaciones de bases de datos, entregados con su código fuente y documentación.
- Documento de Pruebas funcionales al DataMart que garantice su aplicabilidad para la organización y cumpla con los requisitos esperados.

2.7. ALCANCES E IMPACTOS ESPERADOS

El alcance de este proyecto se enfocara fundamentalmente en el análisis, diseño, construcción y entrega de una solución multidimensional basada en un DataMart que gestionara la información de las ventas históricas de la organización TDM Transportes, proporcionara las estructuras requeridas para la explotación de la información y ayudara en la elaboración de instrumentos para una mejor toma de decisiones.

Para llevar a cabo el desarrollo de este proyecto se hará uso de la arquitectura para el manejo de soluciones de Inteligencia de Negocios de Microsoft – SQL SERVER Business Intelligence Suite 2008.

2.7.1. Impactos esperados

El conjunto de actividades del proyecto se enmarca principalmente dentro de las siguientes fases:

- **Gestión del Proyecto:**

Aplicar las técnicas de Gestión de Proyectos de TI para abordar una solución multidimensional de apoyo a la toma de decisiones dentro de un entorno empresarial real.

- **Evaluación del negocio:**

Obtener un conocimiento tácito del área de negocio específica que se espera impactar con el DataMart de ventas dentro de la organización.

- **Definición de requerimientos:**

Aplicar las técnicas de recolección de requerimientos para identificar y documentar las necesidades funcionales que debe suplir el DataMart de ventas dentro de la organización.

- **Análisis de datos:**

Conocer el sistema transaccional OLTP que posee la información histórica de la organización y que permitirá definir el modelo dimensional del DataMart de Ventas.

- **Diseño de la base de datos:**

Elaborar la estructura relacional y dimensional que soportara al DataMart de ventas.

- **Diseño y desarrollo del ETL:**

Elaborar los procesos estructurados que permitirán la extracción, transformación y carga de los datos desde el sistema transaccional hacia estructura de datos del DataMart.

- **Desarrollo de la aplicación:**

Diseñar las interfaces necesarias utilizando herramientas software para explotar el DataMart.

- **Pruebas y certificación:**

Validar el correcto funcionamiento del modelo del DataMart con base a los requerimientos definidos en el análisis, y obtener la aprobación por parte de la organización.

- **Despliegue:**

Publicar la solución dentro de la infraestructura tecnológica de la organización, y capacitar a los usuarios líderes que harán uso del DataMart.

3. DESARROLLO DE LA SOLUCION

3.1. Requerimientos Funcionales

La compañía TDM Transportes S.A.S. dedicada al servicio de transporte de carga en Colombia, hasta el mes de Febrero del año 2011 trabajó con el sistema de información CORE del negocio llamado SAT, desarrollado al interior de la Compañía, debido a esto se crea la necesidad de transformar la información transaccional histórica a una solución de base de datos multidimensional que permita a las áreas directivas elaborar los análisis necesarios para la toma de decisiones.

En el acta de reunión definida en el ANEXO 1, se analizaron los requerimientos generales y específicos para la extracción de la información relevante a las ventas de la organización, este documento es el resultado de dicho análisis y el insumo principal para la elaboración del DataMart de Ventas como modelo Multidimensional.

La empresa es una empresa consolidada a nivel nacional y apuesta por la tecnología en todos los ámbitos de actuación, desde el propiamente relacionado con su sector (servicios de transporte terrestre), como a nivel de sistemas de información, donde trabajan con el ERP de SAP.

3.1.1. Objetivos

Desarrollar un proyecto de BI para sacar el mayor partido a la información de ventas que dispone la organización. El director comercial actúa como patrocinador de este proyecto, que además es un proyecto piloto dentro de la compañía y que se ampliara a otras áreas en el caso de obtener el éxito esperado.

Disponer de una herramienta de consulta centralizada de toda la información referente a ventas, servicios, asesores, clientes y agencias en un solo almacén de datos a partir del cual la organización podría extraer la información (en caso de requerirla) de la siguiente manera:

1. A nivel de reportes y consultas ad-hoc.
2. Navegación dimensional por los datos utilizando herramientas OLAP.
3. Establecer las bases para el análisis y Minería de Datos en los clientes de TDM Transportes, buscando como fin tomar mejores decisiones que ayuden a aumentar la rentabilidad.

El modelado de información de las ventas históricas de la organización se apoya en una solución tipo OLAP.

Debido a que la información histórica de las ventas de TDM Transportes se encuentra almacenada en un sistema de información transaccional OLTP (Base de datos Relacional) donde la información, aunque se encuentra bien normalizada y estructurada para facilitar la elaboración de consultas y reportes que ayudan a visualizar el panorama real de la organización en un periodo de tiempo determinado, carece de herramientas analíticas fundamentales presentes en los sistemas de apoyo gerencial, que favorezcan el análisis, la interpretación y explotación de los datos críticos necesarios para reconocer tendencias importantes en las ventas y especialmente ayudar a la toma acertada de decisiones. Entre otras características relevantes consideradas para realizar el proyecto basado en una solución OLAP sobre otra propuesta, se destaca la agilidad para procesar grandes volúmenes de datos, la capacidad para responder ágil y rápidamente a consultas complejas, la facilidad para manejar información histórica eficientemente y la flexibilidad para realizar el cruce de variables por dos o más dimensiones, en caso de ser requerido y resumir los datos de múltiples maneras y perspectivas en tiempos cortos.

Peculiaridad y ventaja del diseño de una base de datos multidimensional.

Las bases de datos multidimensionales están rigurosamente diseñadas para que la información que almacenan sea consistente y bien estructurada, con dimensiones y medidas bien definidas, estructuradas lógicamente según el tipo de negocio y/o usabilidad, y que garanticen que los datos no serán alterados

durante su utilización. Todas estas particularidades de las bases de datos OLAP hacen la diferencia y la ventaja sobre las bases de datos transaccionales OLTP.

3.1.2. Análisis de Requerimientos

Después de la reunión con el Director Comercial, el Asesor Comercial y el director de Información y Tecnología, y a partir de los requerimientos de información definidos en el acta # 1 (Ver ANEXO 1), se decide:

- a) **Ámbito:** El ámbito del proyecto serán los procesos de negocio relacionados con ventas. El objetivo es tener un mejor conocimiento de la casuística de ventas de la modalidad de negocio conocida como Venta Directa, que permita analizar la información desde todas las perspectivas posibles de interés para el negocio.
- b) **Granularidad de la Información:** A partir de los requerimientos de información, se decide que el dato más granular para el modelo de ventas es la transacción a nivel de la Remesa con el fin de garantizar máxima dimensionalidad y flexibilidad.
- c) **Dimensiones:** Las perspectivas por las que se quiere analizar la información en la empresa TDM Transportes son:
 - **Dimensión Fecha:** Dimensión que permitirá analizar y visualizar la información de las ventas por tiempo (Periodos de fecha).
 - **Dimensión Cliente:** Dimensión que permitirá analizar y visualizar la información de las ventas agrupada por los clientes.
 - **Dimensión Agencias:** Dimensión que permitirá analizar y visualizar la información de las ventas agrupada por la agencia donde se elabora el despacho de la mercancía o donde se causa contablemente.

- **Dimensión Unidad Cargue:** Dimensión que permitirá analizar y visualizar la información de las ventas agrupada por las unidades de cargue o empaque de la mercancía.
 - **Dimensión Tipo Transporte:** Dimensión que permitirá analizar y visualizar la información de las ventas agrupada por los tipos de transporte que maneja la organización.
 - **Dimensión Ciudad:** Dimensión que permitirá analizar y visualizar la información de las ventas agrupada por la ciudad de origen o de destino asociada a la venta.
 - **Dimensión Vehículo:** Dimensión que permitirá analizar y visualizar la información de las ventas agrupada por los vehículos en los cuales es transportada la mercancía.
 - **Dimensión Tráiler:** Dimensión que permitirá analizar y visualizar la información de las ventas agrupada por los tráiler en los cuales es cargada la mercancía.
- d) **Hechos:** Estructura que contiene todos los valores o medidas de negocio asociadas a las ventas históricas y que serán medio de análisis para las diferentes perspectivas o dimensiones.

3.1.3. Modelo conceptual

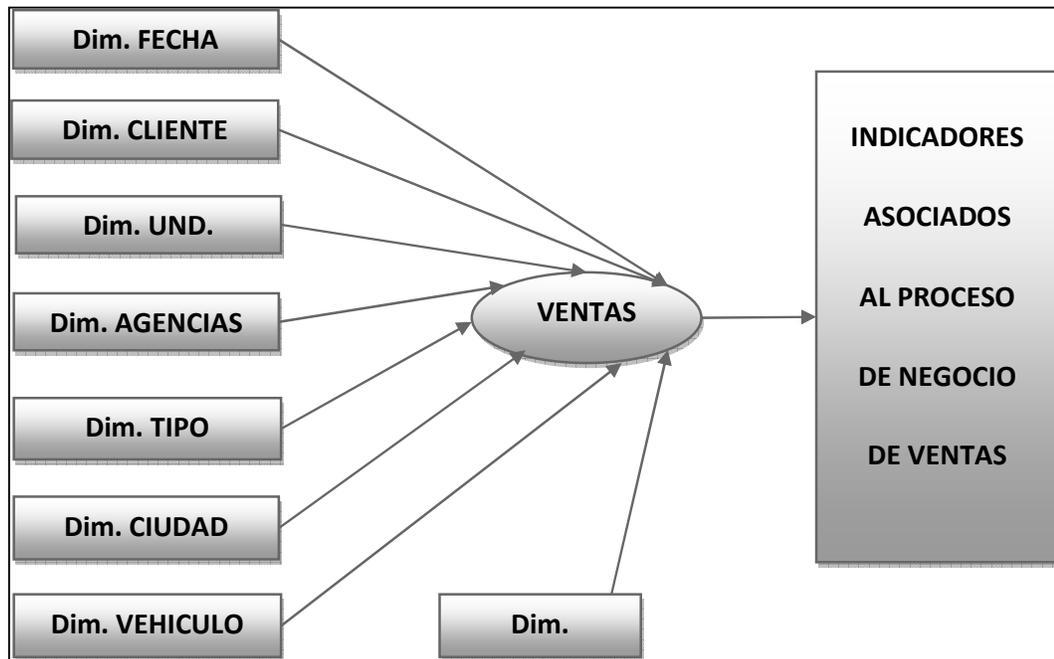


Figura 8. Modelo Conceptual Del DataMart de Ventas.

El desarrollo de cada una de las dimensiones de análisis y de los indicadores del modelo (Hechos) es el siguiente:

Dimensión Fecha

La dimensión Fecha es la única dimensión casi garantizada para estar en cada DataMart y además siempre proporciona una de las perspectivas por las que se analizará la información. La información que conformara esta dimensión será generada a partir de un periodo de tiempo determinado (Incluirá los años 2009, 2010, 2011 hasta el mes de Febrero).

Los atributos que conformaran esta dimensión serán:

- **Clave Fecha (PK):** Define la clave principal de la dimensión en el formato numérico – YYYYMMDD.

- **Fecha:** Fecha en el formato en que se realiza la transacción (DD/MM/YYYY)
- **Descripción de la fecha:** Define la fecha en su formato (Día de Mes de Año: 01 de enero de 2009).
- **Día de la semana:** Nombre del día de la semana. (Lunes, martes, miércoles, etc).
- **Numero día de la semana:** Numero que representa el día de la semana (1, 2, 3...7).
- **Día del Mes:** Número del día en el mes calendario basado en la fecha (1, 15, 31).
- **Día del Año:** Numero de día en el año calendario basado en la fecha (32, 101, 365).
- **Semana:** Numero de la semana en el año basada en la fecha (semana 15).
- **Semana del año:** Formato año – semana (YYYY-SS, 2010-15) basado en el fecha.
- **Mes:** Número del mes calendario en el año basado en la fecha. (Enero = 1, Febrero = 2, etc.).
- **Nombre del Mes:** Nombre del mes basado en la fecha. (Enero, Febrero, Marzo, etc.).
- **Mes del Año:** Formato Año-Mes basado en la fecha (YYYY-MM, 2009-03).
- **Trimestre:** Trimestre del año basado en la fecha (1T, 2T, 3T o 4T).
- **Formato Trimestre del año:** Formato Año - Trimestre (YYYY-TT, 2010-1T).
- **Cuatrimestre:** Cuatrimestre del año basado en la fecha (1Q, 2Q, 3Q).
- **Formato Cuatrimestre del año:** Formato Año - Cuatrimestre (YYYY-QT, 2010-3Q).
- **Semestre:** Semestre del año basado en la fecha.
- **Semestre del año:** Formato semestre – Año (YYYY-SS, 2010-1S).

- **Año:** Año calendario basado en la fecha. Formato en 4 dígitos (2009, 2010, 2011).
- **Festivo:** Indicador que tomará los valores de FESTIVO o NOFESTIVO basado en la fecha.
- **Fin de Semana:** Indicador que tomará los valores de DIASEMANA o FINSEMANA basado en la fecha. Obviamente para el sábado y el domingo sería asignado el valor FINSEMANA.

Dimensión Cliente

Esta dimensión hace parte de los componentes críticos del modelo y es la piedra angular para realizar análisis centrado en todos los atributos relacionados con el Cliente. Así mismo, nos van a permitir analizar quien nos compra, bajo que agrupaciones y clasificaciones.

Los atributos que conformaran esta dimensión serán:

- **Clave Cliente (PK):** Define la clave principal de la dimensión Cliente en formato numérico.
- **Identificación Cliente:** Información que identifica al cliente que solicita un servicio (transacción de Venta). Sera la clave natural de la dimensión Cliente y corresponde a la Cédula o NIT del cliente.
- **Nombre Cliente:** Nombre del Cliente.
- **Grupo Cliente:** Es el código que agrupa a un conjunto de clientes en un nivel superior, los cuales comparten una serie de características comunes. Ej. (Grupo Mundial).
- **Asesor Comercial:** Responsable comercial que tiene asignada la relación con el cliente.
- **Ramo:** Información relacionada con la actividad económica que desarrolla el cliente a nivel comercial.

Dimensión Agencia

Esta dimensión permitirá realizar análisis de la información desde la perspectiva de las agencias en donde se registran las ventas y de donde se despacha la mercancía.

La dimensión Agencias será modelada como una dimensión Role-Playing, la cual ocurre cuando una dimensión aparece simultáneamente varias veces en la misma tabla de hecho - y será usada para manejar los roles de Agencia de Ventas y Agencias de Despacho en la tabla hechos

Los atributos que conformaran esta dimensión serán:

- **Agencia ID (PK):** Define la clave principal de la dimensión Agencias en formato numérico.
- **Agencia:** Agencia de venta y / o despacho. Clave Natural de la dimensión Agencias.
- **Nombre de la Agencia:** Nombre de la agencia de venta y/o despacho.
- **Ciudad de la Agencia:** Atributo relacionado con el ámbito geográfico de la agencia en términos de ciudad.
- **Departamento de la Agencia:** Atributo relacionado con el ámbito geográfico de la agencia en términos del departamento.

Dimensión Unidad de Cargue

Esta dimensión permitirá realizar análisis de la información desde la perspectiva de la unidad de Cargue en la que se embarca la mercancía.

Los atributos que conformaran esta dimensión serán:

- **Unidad Cargue ID (PK):** Define la clave principal de la dimensión Unidad Cargue en formato numérico.

- **Unidad Cargue:** Descripción de la unidad de cargue de la mercancía del cliente.

Dimensión Tipo Transporte

Esta dimensión permitirá realizar análisis de la información desde la perspectiva de los tipos de transporte en el cual se despacha la mercancía.

Los atributos que conformaran esta dimensión serán:

- **Tipo Transporte ID (PK):** Define la clave principal de la dimensión Tipo Transporte en formato numérico.
- **Tipo Transporte:** Define la Clave Natural de la dimensión Tipo de Transporte.
- **Descripción del Tipo de Transporte:** Identificación del Tipo de Transporte requerido para despachar la mercancía del cliente.

Dimensión Ciudad

Esta dimensión permitirá realizar análisis de la información desde la perspectiva de la ruta Origen – Destino de la mercancía, teniendo en cuenta que punto de origen o destino (Ciudad, Municipio, pueblo, etc.) será tratado como una ciudad.

La dimensión Ciudad será modelada como una dimensión Role-Playing y será usada para manejar los roles de punto de Origen y Destino de la ruta en la tabla hechos.

Los atributos que conformaran esta dimensión serán:

- **Ciudad ID (PK):** Define la clave principal de la dimensión ruta en formato numérico.
- **Ciudad:** Atributo relacionado con el código de la ciudad con la cual se identifica en la base de datos transaccional.
- **Nombre de la Ciudad:** Atributo relacionado al nombre de la ciudad como punto de referencia geográfico.
- **Departamento de la Ciudad:** Atributo relacionado con el nombre del departamento asociado al ámbito geográfico de la ciudad.

Dimensión Vehículo

Esta dimensión permitirá realizar análisis de la información desde la perspectiva del Vehículo en el cual se desplaza la mercancía.

Los atributos que conformaran esta dimensión serán:

- **Vehículo ID (PK):** Define la clave principal de la dimensión vehículo en formato numérico.
- **Placa Vehículo:** Define la Clave Natural (Placa del Vehículo) de la dimensión Vehículo.
- **Tipo vehículo:** Descripción del Tipo de vehículo requerido para desplazar la mercancía (Tractomula, Minimula, Sencillo, etc.).
- **Modelo Vehículo:** Año que identifica el modelo del vehículo.
- **Documento Propietario Vehículo:** Atributo que describe el número del documento de identificación del propietario del vehículo.
- **Nombre Propietario Vehículo:** Atributo que describe el nombre propietario del vehículo, puede ser persona natural o jurídica.
- **Propiedad Vehículo:** Atributo que describe si el vehículo es propiedad o no de la organización.

Dimensión Tráiler

En esta dimensión se agrupará la información desde la perspectiva del Tráiler en el cual se carga la mercancía del cliente.

Los atributos que conformaran esta dimensión serán:

- **Tráiler ID (PK):** Define la clave principal de la dimensión Tráiler en formato numérico.
- **Placa Tráiler:** Define la Clave Natural (Placa del Tráiler o remolque) de la dimensión Tráiler.
- **Tipo Tráiler:** Define el número de ejes del tráiler (S1, S2, S3)
- **Grupo Tráiler:** Descripción del Grupo de remolque utilizado para cargar la mercancía (Plancha, tanque, etc.).
- **Propiedad Tráiler:** Atributo que describe si el tráiler es propiedad o no de la organización.

Medidas de las Ventas (Hechos)

Los valores o medidas de negocio que se analizarán para las diferentes perspectivas o dimensiones serán los siguientes:

- **Flete Cobrado:** Valor del flete cobrado al cliente para movilizar la mercancía. Este valor incluye los valores complementarios al transporte de la remesa como (Cargues, Descargues, CiudadAdicional, ClienteAdicional, ValorAdicional, ValorVaciarContenedor, ValorUrbano, CostosFacturadosCliente).
- **Flete Pagado:** Valor del flete pagado al tenedor del vehículo utilizado para movilizar la mercancía. Este valor incluye los valores complementarios al transporte de la remesa como (Cargues, Descargues, CostosRemesa, CostosCobradosTercero).
- **Cantidad:** Cantidad de unidades transportadas en la remesa.

- **Peso Pagado:** Peso pagado al propietario del vehículo que transporta la mercancía en la remesa.
- **Peso Cobrado:** Peso cobrado al cliente de los productos que contiene la remesa.
- **Peso Real:** Peso real de los productos que contiene la remesa.
- **Acompañante Pagado:** Valor pagado al tenedor del vehículo por escoltar la mercancía.
- **Acompañante Cobrado:** Valor cobrado al cliente por escoltar la mercancía.
- **Intermediación:** Valor de utilidad que arroja la venta en base al flete. Este valor es calculado de la siguiente manera: (Flete Cobrado – Flete Pagado).
- **Margen Intermediación:** Porcentaje de utilidad que arroja la venta en base al flete. Este valor es calculado de la siguiente manera: $((\text{Flete Cobrado} - \text{Flete Pagado}) / \text{Flete Cobrado}) * 100$.
- **Dimensión Remesa (Dimensión Degenerada):** La dimensión Remesa, que es clave en la tabla de hechos, es una dimensión degenerada. Esto significa que la clave se identifica a sí mismo y no tiene ningún atributo adicional, por lo que no es necesario construir una tabla de dimensión para ella. En la tabla de hecho, el número de remesa (número de control operacional) será identificada con la notación DD y será usada para realizar análisis a nivel de transacción, estará compuesta por la agencia concatenada con el número de la remesa.
- **Dimensión Kilómetros Ruta (Dimensión Degenerada):** En la tabla de hecho se llevará el número de kilómetros que recorrió el vehículo que llevó la remesa.
- **Dimensión Manifiesto (Dimensión Degenerada):** En la tabla de hecho se llevará el número de manifiesto que contiene la remesa.

- **Dimensión Factura (Dimensión Degenerada):** En la tabla de hecho se llevará la agencia concatenada con el número de Factura que contiene la remesa.
- **Estado Remesa:** En la tabla de Hecho se llevará el estado de la remesa.

El modelo lógico del DataMart de Ventas, incluyendo todas las dimensiones e indicadores es el siguiente:

3.1.4. Modelo Lógico

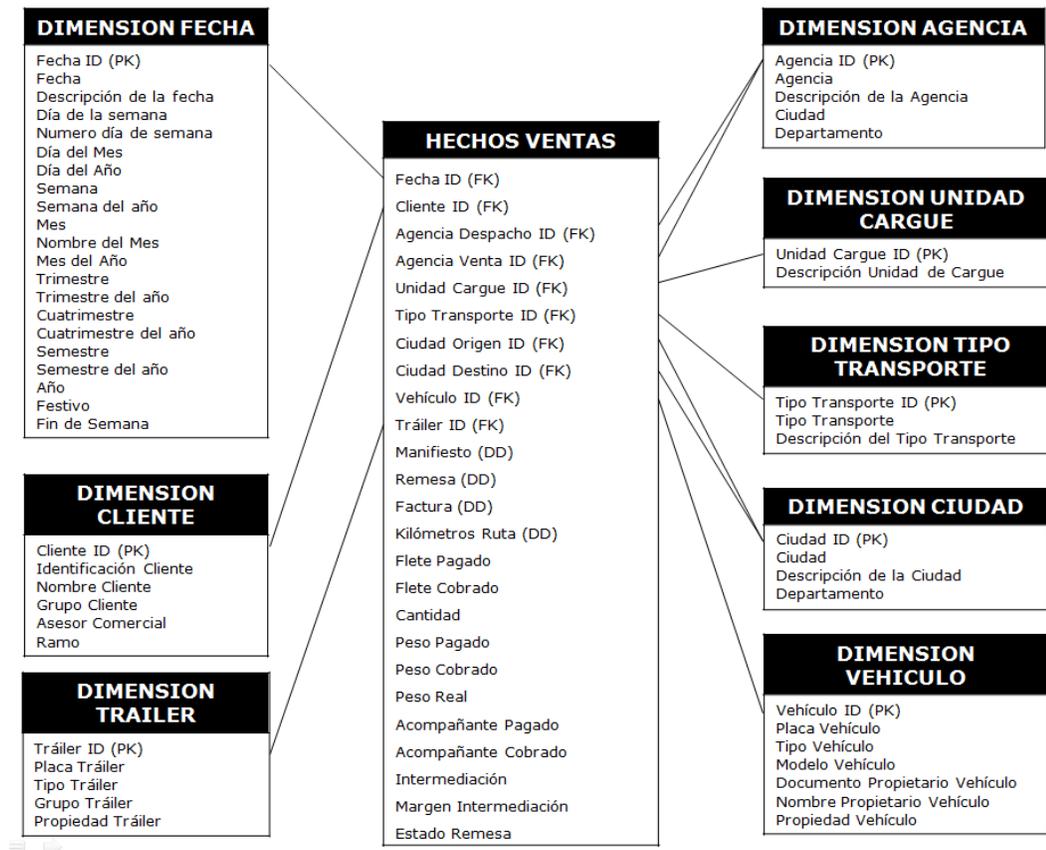


Figura 9. Modelo Lógico Dimensional del DataMart de Ventas.

3.1.5. Restricciones y/o consideraciones

Para la elaboración del modelo se tendrán en cuenta las siguientes restricciones y/o consideraciones:

1. No se cargarán las remesas con estado 'A' (Anulado) en la tabla de hechos.
2. Solo se cargarán las remesas que se encuentren entre el rango de fechas de 1º de Enero de 2009 y el 1º de Marzo de 2011.
3. Las remesas que estén con flete incluido no se cargarán en la tabla de hechos, debido que otra remesa ya contiene el flete.
4. Las remesas de reexpedición no se incluirán en la tabla de hechos debido a que su flete a cobrar es cero, el flete a pagar será adicionado a la remesa reexpedida.
5. Las remesas sin número de manifiesto o sin número de factura se incluirán en la tabla de hechos, pero identificando en cada campo la carencia del dato con valor en 'NOT_FOUND'. Las remesas sin manifiesto impiden la extracción de los kilómetros de la ruta del manifiesto, para el cual tomará valor de cero (0).
6. Los kilómetros de la ruta que se toman del manifiesto y son asignados a la remesa no hacen referencia exacta a los kilómetros recorridos por cada remesa.
7. En la dimensión Vehículo se incluirá el vehículo tantas veces como propietarios haya tenido éste en el tiempo.
8. Debido a que la información de los propietarios del tráiler durante la venta no se almacena en el SAT, en la dimensión Tráiler no se incluirán los propietarios que haya tenido éste en el tiempo.
9. Las remesas que tienen flete Incluido se encuentran dentro del sistema transaccional desde el 7 de abril de 2007, por lo tanto, el paquete construido para la carga de las remesa no aplicará en fechas anteriores para el alcance definido en el DataMart de Ventas.

3.2. REQUERIMIENTOS TECNICOS Y DE ACCESO

3.2.1. Objetivo

Proveer la información necesaria para determinar las características de hardware y software requeridos para implementar el DataMart de Ventas histórico en la organización TDM transportes.

3.2.2. Requerimientos de Hardware

Un elemento importante a tener en cuenta al momento de brindar recomendaciones de hardware es la escalabilidad de la propia infraestructura. La infraestructura de hardware sigue avanzando en la dirección de ofrecer equipos con gran capacidad de escalabilidad, que debe ser considerada al momento de hacer una recomendación para un propósito concreto. La mejor inversión en software y hardware es aquella que acompaña con elasticidad las necesidades concretas que se van planteando con el tiempo. El sobre-dimensionamiento de un equipo en etapas iniciales va a afectar el ROI sin brindar un resultado concreto.

Otros factores que deben ser analizados y que también influyen en la determinación de los requerimientos son:

- La cantidad de información efectivamente almacenada en los DataMart.
- La complejidad de los modelos multidimensionales.
- La modalidad de uso por parte de los usuarios.
- Otros factores directamente dependientes de los datos.

La siguiente información detalla los requerimientos para la máquina servidor utilizada para implementar el DataMart:

3.2.3. Requerimientos del Servidor

Los parámetros que se muestran en la siguiente tabla, representan una guía para establecer los requerimientos mínimos.

CARACTERÍSTICAS	CONFIGURACIÓN MÍNIMA RECOMENDADA
Arquitectura	64 bits
Procesador	Intel Core 2 Duo 2.0Ghz (64bits) AMD 5200
Memoria	4Gb en 64 bits
Espacio en Disco	4Gb Configuración y Logs
Sistema Operativo	Windows Server 2003, Windows Server 2008
Otros características	Discos SATA II o SAS

Tabla 2. Requerimientos del Servidor

3.2.4. Requerimientos del Cliente

Se debe considerar que los requerimientos a nivel del “Cliente” dependen de la cantidad de aplicaciones simultáneas que el usuario utiliza para su trabajo.

Para el caso “Cliente” los requerimientos mínimos son:

Características	Configuración Mínima Recomendada
Arquitectura	32/64bits
Procesador	Intel Pentium IV 1.7GHz
Memoria	512 MB / 768 MB
Espacio en Disco	1 GB

Sistema Operativo	Windows XP / Windows 7
Navegador	Internet Explorer 7.0 - 8.0 / Firefox 3.X

Tabla 3. Requerimientos del Cliente

3.2.5. Componentes Técnicos

- **Seguridad.** El sistema debe permitir el acceso (autenticación) a través de la integración con el servidor LDAP de TDM Transportes.
- **Infraestructura.** La solución de BI deberá desplegarse en dos ambientes independientes así:
 - a. Pruebas:** se realizará en máquinas existentes de TDM Transportes, las cuales permitirán realizar pruebas funcionales.
 - b. Producción:** se realizará en máquinas existentes de TDM Transportes que permiten dimensionar la infraestructura de acuerdo al análisis del volumen de datos reales.
- **Software:** La solución se deberá implantar considerando los siguientes componentes:
 - Plataforma en la que se encuentran los datos transaccionales SQLSERVER 2008 R2.
 - Se realizará la instalación de la bodega de datos en el motor SQLSERVER 2008 R2.
 - Ambiente para plataforma de ETL. SQL Server 2008 R2 Integration Services
 - Plataforma de inteligencia de negocios Business Intelligence Suite SQLSERVER 2008 R2.
 - Explotación del cubo de información. Microsoft Excel 2007.

El desarrollo e implantación de la solución debe ser compatible con el ambiente técnico actual de TDM Transportes, del cual se detallan cada una de las versiones de todos sus componentes:

COMPONENTE	ESPECIFICACIÓN	VERSIÓN	SISTEMA OPERATIVO
Base de datos	Microsoft SQL SERVER	2008 R2	Windows Server 2008
Servidor Web	Microsoft IIS (Internet Information Server)	7.0.x	Windows Server 2008
Mensajería	Microsoft Outlook	2003/2007	Windows XP/7
Directorio de Servicios	Microsoft Active Directory	Ver. 6	Windows Server 2008 (Professional)
Protocolo de Comunicación	TCP/IP	4	Cualquier Sistema Operativo
Sistema Operativo de Servidores	Windows Server	2008	Windows Server 2008 (Professional)
Sistema Operativo de Estación	Microsoft Windows	XP	Professional Service Pack 3
		7	Professional Service Pack 1
Navegadores	Microsoft Internet Explorer	7.0	Windows XP/7

Web		8.0	
	Mozilla Firefox	2.x.x 3.6.x	Windows XP/7
Condiciones mínimas de Canales de Telecomunicaciones entre ciudades remotas y oficina central			
Ancho de Banda (BW)	MPLS, IP De 128 Kbps	Retardo (Delay)	Hasta 300 ms (Internet)

Tabla 4. Requerimientos de Software

3.3. ARQUITECTURA Y ESTRUCTURA DEL DATAMART

3.3.1. Objetivos

El objetivo principal de la arquitectura del DataMart es identificar los componentes fundamentales que permitirán agrupar los datos de las ventas históricas de la organización con el fin de facilitar su análisis, de forma que sean útiles para acceder y analizar información y ayudar en la toma de mejores decisiones.

3.3.2. Arquitectura del DataMart

La arquitectura del DataMart establece el marco de trabajo, estándares y procedimientos que permiten integrar en él las necesidades de información a nivel organizacional.

A continuación se ilustra gráficamente la arquitectura del DataMart de Ventas

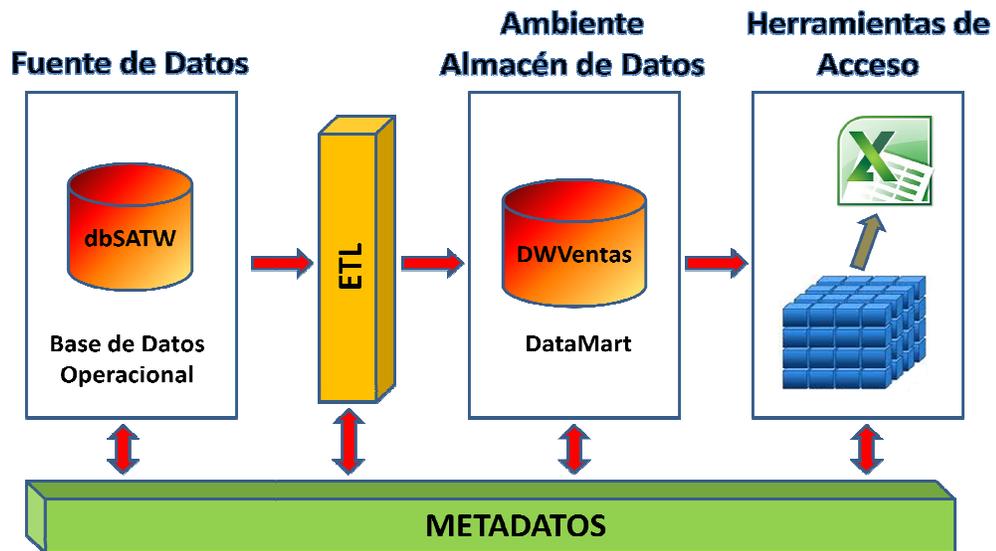


Figura 10. Componentes de la Arquitectura del DataMart de Ventas.

Fuente de Datos

Este componente es el que normalmente está presente en las organizaciones, y a partir del cual se realiza la captura de datos que se contemplará en el DataMart. La fuente de datos del cual se obtendrán los datos históricos significativos de la operación de la organización es el sistema operacional SAT (Base de datos transaccional dbSATW).

Extracción, transformación y carga (ETL)

Para poblar el DataMart se deben mover bloques de datos desde la base de datos operacional dbSATW, mediante procesos programados que se ejecutan frecuentemente fuera del horario de trabajo para no entorpecer la operación normal de la organización.

Las interfaces o subsistemas que usaremos para poblar el DataMart serán construidos utilizando la herramienta Microsoft SQLServer Integration Services.

Al momento de diseñar y construir estas interfaces para poblar el DataMart, debemos considerar la posibilidad de que éstas permitan el crecimiento

evolutivo del DataMart (En caso de que la organización lo requiera), brindando escalabilidad y soporte para grandes cantidades de datos y consultas complejas.

- **Extracción:** El propósito principal de la fase de extracción es capturar y copiar los datos requeridos de nuestra fuente de datos. Los datos que se extraen son colocados en un archivo intermedio con un formato definido, que luego será utilizado por la siguiente fase del proceso.

Los registros que sean rechazados en el proceso deben ser registrados en un archivo o log de rechazos para que puedan ser analizados posteriormente y así tener la posibilidad de cargarlos en el DataMart correctamente. Además, esto permite descubrir los errores que han ocurrido en los procesos de creación de los datos operacionales. Ejemplos de estos errores son violaciones de integridad, claves duplicadas, formatos de datos incorrectos y datos inválidos como campos vacíos, fechas futuras, entre otros.

Hay que tener en cuenta que después de la fase inicial de carga del DataMart sólo es necesario cargar los datos nuevos y que han sido modificados. Es importante anotar que para el DataMart de Ventas Histórico sólo se realizara la fase inicial de carga del DataMart.

- **Transformación:** Las funciones básicas a ser realizadas en esta fase consisten en leer los archivos intermedios generados por la fase de extracción, realizar las transformaciones necesarias, construir los registros en el formato del DataMart y en caso de ser necesario, crear un archivo de salida conteniendo todos los registros nuevos a ser cargados en el DataMart. La mayor parte del trabajo en esta fase involucra el efectuar las transformaciones necesarias.

Algunas de las transformaciones son: Combinar campos múltiples de nombres y apellidos en un solo campo, fusionar campos o datos

homónimos, separar un campo de fecha en campos de año, mes y día, convertir el tipo de dato de la identificación, entre otros.

- **Carga:** El objetivo de esta fase consiste en tomar los registros formateados por la fase de transformación y cargarlos en el DataMart, que es el contenedor para todos los datos informativos, en el caso de los datos históricos, requeridos por las operaciones del DataMart Ventas.

Ambiente Almacén de Datos:

Es el bloque donde se almacenan los datos informativos, utilizado principalmente para usos estratégicos. Corresponde al DataMart DWVentas que permitirá gestionar la información histórica de las ventas de la organización.

Herramientas de acceso y uso:

Sin las herramientas adecuadas de acceso y análisis, el DataMart se puede convertir en una mezcla de datos sin ninguna utilidad. Es necesario poseer técnicas que capturen los datos importantes de manera rápida y puedan ser analizados desde diferentes puntos de vista. También deben transformar los datos capturados en información útil para el negocio.

Actualmente a este tipo de herramientas se las conocen como Herramientas de Inteligencia de Negocio y están situadas conceptualmente sobre el DataMart. Para el proyecto se usará la herramienta para análisis multidimensional Microsoft SQLSERVER Analysis Services.

Así mismo, se utilizara el software Microsoft Office EXCEL para mostrar la información y emitir reportes, gráficos y diagramas para el análisis y presentación.

Metadatos:

Los metadatos son datos acerca de los datos. En una base de datos, los metadatos son la representación de los diversos objetos que definen una base de datos, por ejemplo, ubicación y descripción de la base de datos, las descripciones lógicas y físicas de tablas, columnas y atributos.

Uno de los problemas con el que pueden encontrarse los usuarios de un DataMart es saber lo que hay en él y cómo pueden acceder a lo que quieren. A fin de proveer el acceso a los datos universales, es absolutamente necesario mantener los metadatos. Un componente llamado repositorio les ayuda a conseguirlo. Los metadatos son sólo una de las utilidades del repositorio, pero éste tiene muchas funcionalidades: catalogar y describir la información disponible, especificar el propósito de la misma, indicar las relaciones entre los distintos datos, establecer quién es el propietario de la información, relacionar las estructuras técnicas de datos con la información de negocio, establecer las relaciones con los datos operacionales y las reglas de transformación, y limitar la validez de la información.

A continuación se describen los metadatos de las dimensiones del modelo.

DWD_CLIENTE					
CAMPO	TIPO DATO	NULOS	TIPO CLAVE	OBJETO	FUENTE
Cliente_id	int	NOT NULL	PK	Clave	Entero Autoincrementable
IdentificacionCliente	varchar(15)	NOT NULL	IX	Tabla	tblClientes.strNumeroDoc
NombreCliente	varchar(80)	NOT NULL		Tabla	tblClientes.strNombreCliente
GrupoCliente	varchar(30)	NOT NULL		Tabla	tblGruposClientes.strDescripcionGrupoCliente
AsesorComercial	varchar(80)	NOT NULL		Tabla	tblUsuarios.strNombre1 tblUsuarios.strNombre2 tblUsuarios.strApellido1 tblUsuarios.strApellido2
Ramo	varchar(30)	NOT NULL		Tabla	tblCIU.strRamo

Figura 11. Metadatos de la Dimensión Cliente del DataMart de Ventas.

DWD_FECHA					
CAMPO	TIPO DATO	NULOS	TIPO CLAVE	OBJETO	FUENTE
Fecha_id	int	NOT NULL	PK	Clave	Construido en SQL
Fecha	date	NOT NULL	IX	Condicion	Construido en SQL
NombreFecha	varchar(30)	NOT NULL		Condicion	Construido en SQL
NombreDiaSemana	varchar(10)	NOT NULL		Condicion	Construido en SQL
DiaSemana	smallint	NOT NULL		Condicion	Construido en SQL
DiaMes	smallint	NOT NULL		Condicion	Construido en SQL
DiaAño	smallint	NOT NULL		Condicion	Construido en SQL
Semana	smallint	NOT NULL		Condicion	Construido en SQL
SemanaAño	varchar(7)	NOT NULL		Condicion	Construido en SQL
Mes	smallint	NOT NULL		Condicion	Construido en SQL
NombreMes	varchar(10)	NOT NULL		Condicion	Construido en SQL
MesAño	varchar(7)	NOT NULL		Condicion	Construido en SQL
Trimestre	smallint	NOT NULL		Condicion	Construido en SQL
TrimestreAño	varchar(7)	NOT NULL		Condicion	Construido en SQL
Cuatrimestre	smallint	NOT NULL		Condicion	Construido en SQL
CuatrimestreAño	varchar(7)	NOT NULL		Condicion	Construido en SQL
Semestre	smallint	NOT NULL		Condicion	Construido en SQL
SemestreAño	varchar(7)	NOT NULL		Condicion	Construido en SQL
Año	smallint	NOT NULL		Condicion	Construido en SQL
Festivo	varchar(10)	NOT NULL		Condicion	Construido en SQL
FinSemana	varchar(10)	NOT NULL		Condicion	Construido en SQL

Figura 12. Metadatos de la Dimensión Fecha del DataMart de Ventas.



Figura 13. Metadatos de la Dimensión Agencia y sus Dimensiones Role-Playing en el DataMart de Ventas.

DWD_UNIDADCARGUE					
CAMPO	TIPO DATO	NULOS	TIPO CLAVE	OBJETO	FUENTE
UnidadCargue_id	int	NOT NULL	PK	Clave	Entero Autoincrementable
UnidadCargue	varchar(30)	NOT NULL	IX	Tabla	tblUnidadesCargue.strDescripcionUnidadCargue tblServiciosSolicitud.strDescripcionServicio

Figura 14. Metadatos de la Dimensión Unidad de Cargue del DataMart de Ventas.

DWD_TIPOTRANSPORTE					
CAMPO	TIPO DATO	NULOS	TIPO CLAVE	OBJETO	FUENTE
TipoTransporte_id	int	NOT NULL	PK	Clave	Entero Autoincrementable
TipoTransporte	varchar(2)	NOT NULL	IX	Tabla	tblTiposTransporte.strCodigoTipoTransporte
DescripcionTipoTransporte	varchar(50)	NOT NULL		Tabla	tblTiposTransporte.strDescripcionTipoTransporte

Figura 15. Metadatos de la Dimensión Tipo de Transporte del DataMart de Ventas.



Figura 16. Metadatos de la Dimensión Ciudad y sus Dimensiones Role-Playing en el DataMart de Ventas.

DWD_VEHICULO					
CAMPO	TIPO DATO	NULOS	TIPO CLAVE	OBJETO	FUENTE
Vehiculo_id	int	NOT NULL	PK	Clave	Entero Autoincrementable
PlacaVehiculo	varchar(10)	NOT NULL	IX	Tabla	tblVehiculos.strPlacaVehiculo
TipoVehiculo	varchar(30)	NOT NULL		Tabla	tblTiposEquipos.strDescripcionTipoEquipo
ModeloVehiculo	int	NOT NULL		Tabla	tblVehiculos.intModelo, tblVehiculos.intModeloTransformado
DocumentoPropietarioVehiculo	varchar(15)	NOT NULL		Tabla	tblManifiestos.strNumeroDocPoseedor
PropietarioVehiculo	varchar(60)	NOT NULL		Tabla	tblTerceros.strNombreTercero
PropiedadVehiculo	varchar(7)	NOT NULL		Condicion	tblManifiestos.strNumeroDocPoseedor {PROPIO - TERCERO}

Figura 17. Metadatos de la Dimensión Vehículo del DataMart de Ventas.

DWD_TRAILER					
CAMPO	TIPO DATO	NULOS	TIPO CLAVE	OBJETO	FUENTE
Trailer_id	int	NOT NULL	PK	Clave	Entero Autoincrementable
PlacaTrailer	varchar(10)	NOT NULL	IX	Tabla	tblTrailers.strPlacaTrailer
TipoTrailer	varchar(2)	NOT NULL		Tabla	tblTrailers.strTipoTrailer
GrupoTrailer	varchar(30)	NOT NULL		Tabla	tblGruposTrailers.strDescripcionGrupoTrailer
PropiedadTrailer	varchar(7)	NOT NULL		Condicion	tblTrailers.strNumeroDocPropietario {PROPIO - TERCERO}

Figura 18. Metadatos de la Dimensión Tráiler del DataMart de Ventas.

DWD_VENTAS					
CAMPO	TIPO DATO	NULOS	TIPO CLAVE	OBJETO	FUENTE
Fecha_id	int	NOT NULL	PK	Clave	Dimension Fecha
Cliente_id	int	NOT NULL	PK	Clave	Dimension Cliente
AgenciaDespacho_id	int	NOT NULL	PK	Clave	Dimension Agencia
AgenciaVenta_id	int	NOT NULL	PK	Clave	Dimension Agencia
UnidadCargue_id	int	NOT NULL	PK	Clave	Dimension UnidadCargue
TipoTransporte_id	int	NOT NULL	PK	Clave	Dimension TipoTransporte
CiudadOrigen_id	int	NOT NULL	PK	Clave	Dimension Ciudad
CiudadDestino_id	int	NOT NULL	PK	Clave	Dimension Ciudad
Vehiculo_id	int	NOT NULL	PK	Clave	Dimension Vehiculo
Trailer_id	int	NOT NULL	PK	Clave	Dimension Trailer
Remesa	varchar(10)	NOT NULL	PK, IX	Tabla	tblRemesas.strCodigoAgencia tblRemesas.intCons
Manifiesto	varchar(10)	NOT NULL		Tabla	tblRemesas.intNumeroManifiesto
Factura	varchar(10)	NOT NULL		Tabla	tblRemesas.strCodigoAgenciaFactura tblRemesas.intNumeroFactura
KilometrosRuta	int	NOT NULL		Tabla	tblRutas.dblKms
FletePagado	Money	NOT NULL		Funcion Scalar	fnFletePagadoRemesa
FleteCobrado	Money	NOT NULL		Funcion Scalar	fnFleteCobradoRemesa
Cantidad	int	NOT NULL		Funcion Scalar	CantidadRemesaPorRemesa
PesoPagado	float	NOT NULL		Tabla	tblRemesas.dblKgsProp
PesoCobrado	float	NOT NULL		Tabla	tblRemesas.dblKgsRealesCobro
PesoReal	float	NOT NULL		Funcion Scalar	PesoRemesasPorRemesa
AcompanantePagado	Money	NOT NULL		Tabla	tblDetallesAcompañamientos.dblValor
AcompananteCobrado	Money	NOT NULL		Tabla	tblRemesas.dblTarifaEscoltaCobro
Intermediacion	Money	NOT NULL		Funcion Scalar	Intermediacion
MargenIntermediacion	float	NOT NULL		Funcion Scalar	fnMargenIntermediacion
EstadoRemesa	varchar(10)	NOT NULL		Tabla	tblRemesas.strEstado

Figura 19. Metadatos de la Tabla de Hechos del DataMart de Ventas.

3.4. DISEÑO FÍSICO DEL DATAMART

3.4.1. Objetivos

El objetivo es definir las estructuras necesarias a nivel físico para soportar las necesidades de información que han sido identificadas en la etapa de levantamiento y análisis y las cuales se han reflejado en un modelo lógico dimensional. Así mismo, abarcar la nomenclatura de organización del DataMart como Entidades/Tablas, Atributos/Campos, Restricciones de Integridad y Campos Estándar en la definición de las dimensiones.

3.4.2. Diseño Físico

El siguiente paso en el desarrollo del DM según la metodología de Ralph Kimball es el diseño físico del mismo. El modelo lógico dimensional desarrollado debe convertirse ahora en un diseño físico que incluya los nombres de columna físicos, los tipos de los datos, las declaraciones de clave (si procede), la posibilidad de incluir valores nulos, la creación de índices y tablas agregadas como alternativas para garantizar un mejor rendimiento y optimizar las consultas.

Durante el proceso de diseño físico se trasladan los esquemas lógicos previstos a las estructuras reales. En este momento hay que transformar las entidades en tablas, crear las relaciones entre las dimensiones y la tabla de hechos mediante claves foráneas, transformar atributos en columnas y transformar los identificadores únicos primarios y alternativos en claves primarias y alternativas.

A continuación se muestra el diseño físico de cada una de las dimensiones:

Dimensión Fecha

Nombre de la tabla: **dwd_Fecha**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
Fecha_id (PK)	Integer	NO
Fecha (*)	Date	NO
NombreFecha	Varchar(30)	NO
NombreDiaSemana	Varchar(10)	NO
DiaSemana	Smallint	NO
DiaMes	Smallint	NO
DiaAño	Smallint	NO
Semana	Smallint	NO
SemanaAño	Varchar(7)	NO
Mes	Smallint	NO
Mes	Varchar(10)	NO
MesAño	Varchar(7)	NO
Trimestre	Smallint	NO

TrimestreAño	Varchar(7)	NO
Cuatrimestre	Smallint	NO
CuatrimestreAño	Varchar(7)	NO
Semestre	Smallint	NO
SemestreAño	Varchar(7)	NO
Año	Smallint	NO
Festivo	Varchar(10)	NO
FinSemana	Varchar(10)	NO

Tabla 5. Dimensión Fecha

(*) Se define el índice **IX_dwd_Fecha** para optimizar e indexar la información por el campo Fecha.

Dimensión Cliente

Nombre de la tabla: **dwd_Cliente**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
Cliente_id (PK)	Integer	NO
IdentificacionCliente (*)	Varchar(15)	NO
NombreCliente	Varchar(80)	NO
GrupoCliente	Varchar(30)	NO
AsesorComercial	Varchar(80)	NO
Ramo	Varchar(30)	NO

Tabla 6. Dimensión Cliente

(*) Se define el índice **IX_dwd_IdentificacionCliente** para optimizar e indexar la información por el campo IdentificacionCliente.

Dimensión Agencia

Nombre de la tabla: **dwd_Agencia**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
Agencia_id (PK)	Integer	NO
Agencia (*)	Varchar(2)	NO
NombreAgencia	Varchar(30)	NO
CiudadAgencia	Varchar(30)	NO
DepartamentoAgencia	Varchar(50)	NO

Tabla 7. Dimensión Agencia

(*) Se define el índice **IX_dwd_Agencia** para optimizar e indexar la información por el campo Agencia.

Dimensión Unidad de Cargue

Nombre de la tabla: **dwd_UnidadCargue**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
UnidadCargue_id (PK)	Integer	NO
UnidadCargue (*)	Varchar(30)	NO

Tabla 8. Dimensión Unidad de Cargue

(*) Se define el índice **IX_dwd_UnidadCargue** para optimizar e indexar la información por el campo UnidadCargue.

Dimensión Tipo Transporte

Nombre de la tabla: **dwd_TipoTransporte**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
TipoTransporte_id (PK)	Integer	NO
TipoTransporte (*)	Varchar(2)	NO
DescripcionTipoTransporte	Varchar(50)	NO

Tabla 9. Dimensión Tipo de Transporte

(*) Se define el índice **IX_dwd_TipoTransporte** para optimizar e indexar la información por el campo TipoTransporte.

Dimensión Ciudad

Nombre de la tabla: **dwd_Ciudad**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
Ciudad_id (PK)	Integer	NO
Ciudad (*)	Varchar(4)	NO
NombreCiudad	Varchar(30)	NO
DepartamentoCiudad	Varchar(50)	NO

Tabla 10. Dimensión Ciudad

(*) Se define el índice **IX_dwd_Ciudad** para optimizar e indexar la información por el campo Ciudad.

Dimensión Vehículo

Nombre de la tabla: **dwd_Vehiculo**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
Vehiculo_id (PK)	Integer	NO
PlacaVehiculo (*)	Varchar(10)	NO
TipoVehiculo	Varchar(30)	NO
ModeloVehiculo	Integer	NO
DocumentoPropietarioVehiculo	Varchar(15)	NO
NombrePropietarioVehiculo	Varchar(60)	NO
PropiedadVehiculo	Varchar(7)	NO

Tabla 11. Dimensión Vehículo

(*) Se define el índice **IX_dwd_PlacaVehiculo** para optimizar e indexar la información por el campo placa del vehiculo: PlacaVehiculo

Dimensión Trailer

Nombre de la tabla: **dwd_Trailer**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
Trailer_id (PK)	Integer	NO
PlacaTrailer (*)	Varchar(10)	NO
TipoTrailer	Varchar(2)	NO
GrupoTrailer	Varchar(30)	NO
PropiedadTrailer	Varchar(7)	NO

Tabla 12. Dimensión Tráiler

(*) Se define el índice **IX_dwd_PlacaTrailer** para optimizar e indexar la información por el campo PlacaTrailer.

Dimensiones Role-Playing (RP)

Debido a que manejamos en el modelo dimensional varias agencias y ciudades asociadas a cada Remesa, tales como Agencia de Despacho, Agencia de Venta, Ciudad Origen y Ciudad Destino, es importante destacar que cada una estas agencias y ciudades deberían ser una clave foránea en la tabla de hecho. Sin embargo, no podemos simplemente unir dos claves foráneas (Agencias de Despacho y Agencia de Venta) a la misma tabla de dimensión Agencia. El SQL requeriría que ambas agencias fueran idénticas, lo cual no es muy probable, para que la sentencia JOIN operara correctamente. Con el fin de solucionar este inconveniente, usamos objetos SQL tipo vista (View) para crear la ilusión de dos tablas de Agencia y etiquetamos unívocamente las columnas en cada una de ellas. Con estas vistas tenemos dos dimensiones de Agencia únicas

que pueden ser usadas como si fueran copias físicas independientes con Constraints completamente no relacionados y que desempeñan simultáneamente diferentes roles en la tabla de hechos. El mismo escenario sucede con la Ciudad Origen y Ciudad Destino a la dimensión Ciudad.

- **Dimension Role-Playing – Vista Agencia Despacho**

Se definen la dimensión Role-Playing Agencia para simular la dimensión de AgenciaDespacho.

Nombre de la vista: **viAgenciaDespacho**

CAMPO DIMENSION AGENCIA	ALIAS DIM AGENCIA VENTA - RP
Agencia_id	AgenciaDespacho_id
Agencia	AgenciaDespacho
NombreAgencia	NombreAgenciaDespacho
CiudadAgencia	CiudadAgenciaDespacho
DepartamentoAgencia	DepartamentoAgenciaDespacho

Tabla 13. Dimensión Role-Playing Agencia Despacho

- **Dimension Role-Playing – Vista Agencia Venta**

Se definen la dimensión Role-Playing Agencia para simular la dimensión de AgenciaVenta.

Nombre de la vista: **viAgenciaDespacho**

CAMPO DIMENSION AGENCIA	ALIAS DIM AGENCIA VENTA - RP
Agencia_id	AgenciaVenta_id
Agencia	AgenciaVenta
NombreAgencia	NombreAgenciaVenta
CiudadAgencia	CiudadAgenciaVenta
DepartamentoAgencia	DepartamentoAgenciaVenta

Tabla 14. Dimensión Role-Playing Agencia Venta

- **Dimension Role-Playing – Vista Ciudad Origen**

Se definen la dimensión Role-Playing Ciudad para simular la dimensión de CiudadOrigen.

Nombre de la vista: **viCiudadOrigen**

CAMPO DIMENSION AGENCIA	ALIAS DIM AGENCIA VENTA - RP
Ciudad_id	CiudadOrigen_id
Ciudad	CiudadOrigen
NombreCiudad	NombreCiudadOrigen
DepartamentoCiudad	DepartamentoCiudadOrigen

Tabla 15. Dimensión Role-Playing Ciudad Origen

- **Dimension Role-Playing – Vista Ciudad Destino**

Se definen la dimensión Role-Playing Ciudad para simular la dimensión de CiudadDestino.

Nombre de la vista: **viCiudadDestino**

CAMPO DIMENSION AGENCIA	ALIAS DIM AGENCIA VENTA - RP
Ciudad_id	CiudadDestino_id
Ciudad	CiudadDestino
NombreCiudad	NombreCiudadDestino
DepartamentoCiudad	DepartamentoCiudadDestino

Tabla 16. Dimensión Role-Playing Ciudad Destino

Tabla De Hechos

Nombre de la tabla: **dwh_Ventas**

NOMBRE DEL CAMPO	TIPO	ADMITE NULOS
Fecha_id	Integer	NO
Cliente_id	Integer	NO
AgenciaDespacho_id	Integer	NO
AgenciaVenta_id	Integer	NO
UnidadCargue_id	Integer	NO
TipoTransporte_id	Integer	NO

CiudadOrigen_id	Integer	NO
CiudadDestino_id	Integer	NO
Vehiculo_id	Integer	NO
Trailer_id	Integer	NO
Remesa (*)	Varchar(10)	NO
Manifiesto	Varchar(10)	NO
Factura	Varchar(10)	NO
KilometrosRuta	Integer	NO
FletePagado	Money	NO
FleteCobrado	Money	NO
Cantidad	Integer	NO
PesoPagado	Float	NO
PesoCobrado	Float	NO
PesoReal	Float	NO
AcompanantePagado	Money	NO
AcompananteCobrado	Money	NO
Intermediacion	Money	NO
MargenIntermediacion	Float	NO
EstadoRemesa	Varchar(10)	NO

Tabla 17. Tabla de Hechos - Ventas

(*) Se define el índice **IX_dwd_Remesa** para optimizar e indexar la información por el campo Remesa.

El modelo físico definitivo sería el siguiente:

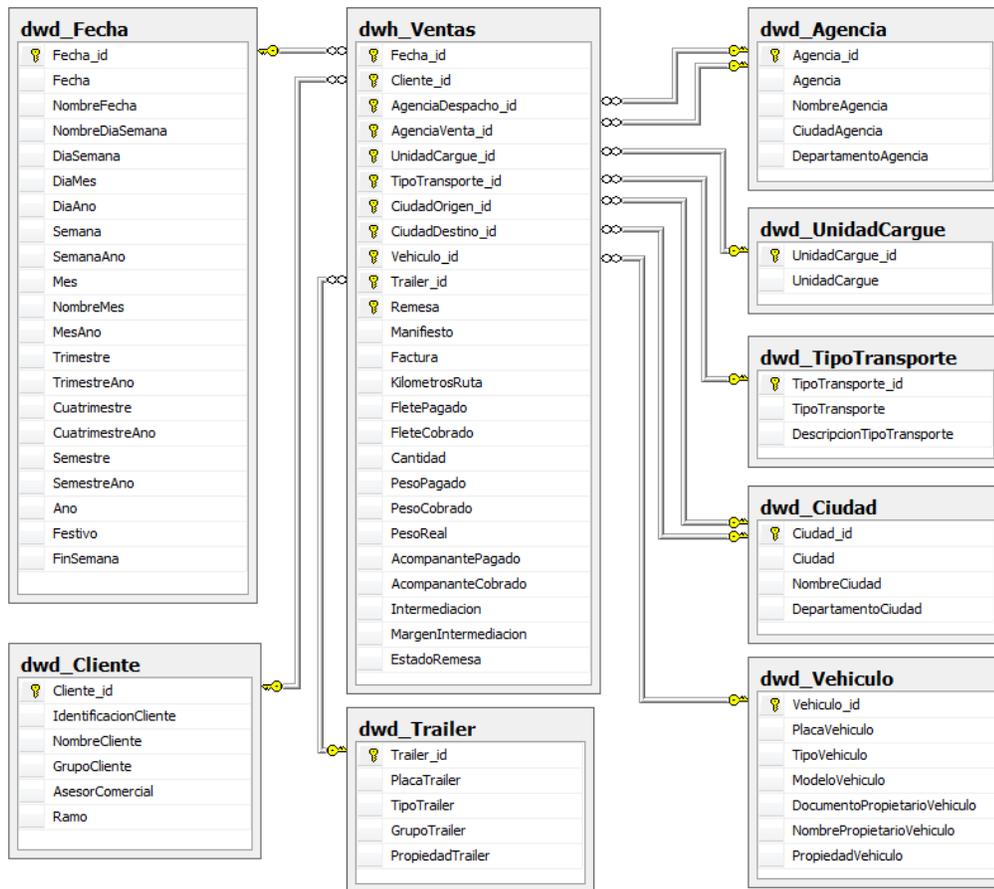


Figura 20. Modelo Físico del DataMart de Ventas.

3.4.3. Construcción de la MetaData para el Modelo Físico

La metodología que usaremos para la construcción de la MetaData y que permitirá identificar cada uno de los componentes del DataMart, consiste en revisar de manera detallada cada una de las dimensiones y sus atributos, así como la tabla de hechos, y para cada componente, describir en qué lugar del sistema origen se encuentra, cuáles son sus características, que tipo de transformaciones se deben realizar sobre ellos, y considerar todas las observaciones importantes que apoyen la elaboración de los procesos de ETL.

Dimensión Fecha

La dimensión Fecha es una dimensión ficticia, que no existe en el sistema operacional SAT, y que construiremos a partir de los calendarios (generaremos los datos correspondientes a los años 2009, 2010, 2011 hasta el mes de febrero). Para la construcción de la dimensión y de todos sus atributos, construiremos un proceso ETL para la carga inicial de esta dimensión.

TRATAMIENTO A REALIZAR EN LOS ATRIBUTOS
Fecha_id: Fecha convertida a numero con el formato YYYYMMDD.
Fecha: Fecha en el formato de fecha (DD/MM/YYYY)
NombreFecha: Fecha convertida a texto en el formato Día de Mes de Año (Ejemplo: 01 de enero de 2009).
NombreDiaSemana: Nombre del día de la semana (Lunes, martes, miércoles, etc.).
DiaSemana: Numero del día de la semana (1 al 7, El Lunes es el primer día de la semana).
DiaMes: Número del día en el mes (día 1, día 29, día 31).
DiaAño: Numero de día en el año (día 32, día 101, día 365).
Semana: Numero de la semana en la que se encuentra la fecha (semana 15, Semana 54).
SemanaAño: Numero de la semana incluyendo el año al que pertenece (Formato YYYY-SS, 2009-35 sería la semana 35 del año 2009)
Mes: Número del mes (mes 1, mes 12).
NombreMes: Nombre del mes (Enero, Febrero, Diciembre).
MesAño: Numero del mes incluyendo el año al que pertenece (Formato YYYY-MM, 2009-12 sería el mes 12 del año 2009)
Trimestre: Indicador del trimestre basado en la fecha (1T, 2T, 3T o 4T).
TrimestreAño: Indicador del trimestre incluyendo el año (Formato YYYY-TT, 2009-03 sería el tercer trimestre del año 2009)
Cuatrimestre: Indicador del cuatrimestre basado en la fecha (1Q, 2Q, 3Q).
CuatrimestreAño: Indicador del cuatrimestre incluyendo el año (Formato YYYY-QQ, 2009-01 sería el primer Cuatrimestre del año 2009)
Semestre: Indicador del semestre basado en la fecha (1S y 2S).
SemestreAño: Indicador del semestre incluyendo el año (Formato YYYY-SS, 2009-01 sería el primer semestre del año 2009)
Año: Año basado en la fecha. Formato en 4 dígitos (2009, 2010, 2011).
Festivo: Indicador si una fecha es festivo o no. Tomará los valores de FESTIVO o NOFESTIVO.
FinSemana: Indicador de si una fecha es fin de semana o no. Tomará los valores de DIASEMANA o FINSEMANA. Para el día sábado y el domingo será asignado el valor FINSEMANA.

Tabla 18. Atributos Dimensión Fecha

Dimensión Cliente

Los datos principales de la dimensión Cliente se encuentran en el sistema transaccional SAT en la tabla tblClientes (Maestro de Clientes). El origen de cada uno de los atributos y las transformaciones a realizar vienen descritos en la siguiente tabla:

ATRIBUTO	ORIGEN DATOS	TIPO	TRANSFORMACIÓN	OBSERVACIONES
Cliente_id		Integer		Este campo debe definirse como campo auto numérico
IdentificacionCliente	tblClientes.strNumeroDoc	Varchar(15)		
NombreCliente	tblClientes.strNombreCliente	Varchar(100)		
GrupoCliente	tblGruposClientes.strDescripcionGrupoCliente	Varchar(30)		Recuperar de la tabla tblGruposClientes usando el campo tblClientes.strCodigoGrupoCliente
AsesorComercial	tblUsuarios.strNombre1 tblUsuarios.strNombre2 tblUsuarios.strApellido1 tblUsuarios.strApellido2	Varchar(80)	Concatenar en una cadena de caracteres, los valores tblUsuarios.strNombre1, tblUsuarios.strNombre2, tblUsuarios.strApellido1, tblUsuarios.strApellido2	Recuperar de la tabla tblUsuarios usando el campo tblClientes.strUsuarioCcial
Ramo	tblCIIU.strRamo	Varchar(30)		

Tabla 19. Atributos Dimensión Cliente

Dimensión Agencia

Los datos de la dimensión Agencia se encuentran en el SAT en la tabla tblAgencias. El origen de cada uno de los atributos y las transformaciones a realizar se describen en la siguiente tabla:

ATRIBUTO	ORIGEN DATOS	TIPO	TRANSFORMACIÓN	OBSERVACIONES
Agencia_id		Integer		Este campo debe definirse como campo auto numérico
Agencia	tblAgencias.strCodigoAgencia	Varchar(2)		

Nombre_Agencia	tblAgencias. strDescripcionAgen cia	Varchar(30)		
CiudadAgencia	tblCiudades. strDescripcionCiud ad	Varchar(30)		Recuperar de la tabla tblCiudades usando el campo tblAgencias. strCodigoCiudad
DepartamentoAgen cia	tblCiudades. strDepartamento	Varchar(50)		Recuperar de la tabla tblCiudades usando el campo tblAgencias. strCodigoCiudad

Tabla 20. Atributos Dimensión Agencia

Dimensión Unidad Cargue

Los datos principales de la dimensión Unidad Cargue se encuentran en el SAT en la tabla tblUnidadCargue. El origen de cada uno de los atributos y las transformaciones a realizar se describen en la siguiente tabla:

ATRIBUTO	ORIGEN DATOS	TIPO	TRANSFORMACI ÓN	OBSERVACIONES
UnidadCargue_id		Integer		Este campo debe definirse como campo auto numérico
UnidadCargue	tblUnidadCargue. strDescripcionUnid adCargue tblServiciosSolicitu d.strDescripcionSe rvicio	Varchar(30)		Este campo debe ser el resultado de la unión entre la unidad de cargue de la remesa unido al servicio solicitado desde la solicitud (si existe), dándole prioridad a los servicios de Isotanke y Flexitanque

Tabla 21. Atributos Dimensión Unidad Cargue

Dimensión Tipo Transporte

Los datos de los Tipos de Transporte se encuentran en el SAT en la tabla tblTiposTransporte. El origen de cada uno de los atributos y las transformaciones a realizar se describen en la siguiente tabla:

ATRIBUTO	ORIGEN DATOS	TIPO	TRANSFORMACIÓN	OBSERVACIONES
TipoTransporte_id		Integer		Este campo debe definirse como campo auto numérico.
TipoTransporte	tblTiposTransporte.strCodigoTipoTransporte	Varchar(2)		
DescripciónTipoTransporte	tblTiposTransporte.strDescripcionTipoTransporte	Varchar(50)		

Tabla 22. Atributos Dimensión Tipo Transporte

Dimensión Ciudad

Los principales datos de la dimensión Ciudad están en el SAT en la tabla tblCiudades. El origen de cada uno de los atributos y las transformaciones a realizar se describen en la siguiente tabla:

Atributo	Origen Datos	Tipo	Transformación	Observaciones
Ciudad_id		Integer		Este campo debe definirse como campo auto numérico
Ciudad	tblCiudades.strCodigoCiudad	Varchar(4)		
NombreCiudad	tblCiudades.strDescripcionCiudad	Varchar(30)		
Departamento	tblCiudades.strDepartamento	Varchar(50)		

Tabla 23. Atributos Dimensión Ciudad

Dimensión Vehículo

Los datos de la dimensión vehículo se encuentran en el SAT en la tabla tblVehiculos. El origen de cada uno de los atributos y las transformaciones a realizar se describen en la siguiente tabla:

ATRIBUTO	ORIGEN DATOS	TIPO	TRANSFORMACIÓN	OBSERVACIONES
----------	--------------	------	----------------	---------------

Vehiculo_id		Integer		Este campo debe definirse como campo auto numérico
PlacaVehiculo	tblVehiculos. strPlacaVehiculo	Varchar(10)		
TipoVehiculo	tblTiposEquipos. strDescripcionTipo Equipo	Varchar(30)		Recuperar de la tabla tblTiposEquipos usando el campo tblVehiculos.strCodigoTipoEquipo
ModeloVehiculo	tblVehiculos. intModelo	Integer		Si el campo tblVehiculos.intModeloTransformado tiene un valor distinto de NULL entonces se toma este campo, si no se toma el valor del campo tblVehiculos.intModelo.
DocumentoPropietarioVehiculo	tblTerceros.strNumeroDoc	Varchar(15)		Recuperar de la tabla tblTerceros.strNumeroDoc usando el campo tblManifiestos.strTipoDocPoseedor y tblManifiestos.strNumeroDocPoseedor
NombrePropietarioVehiculo	tblTerceros.strNombreTercero	Varchar(60)		Recuperar de la tabla tblTerceros.strNombreTercero usando el campo tblManifiestos.strTipoDocPoseedor y tblManifiestos.strNumeroDocPoseedor
PropiedadVehículo	Campo Calculado	Varchar(7)		Campo que almacenara el valor "PROPIO" cuando el NIT del poseedor sea de TDM, y "TERCERO" cuando no corresponda.

Tabla 24. Atributos Dimensión Vehículo

Dimensión Trailer

Los datos de la dimensión Tráiler están en el SAT en la tabla tblTrailers. El origen de cada uno de los atributos y las transformaciones a realizar se describen en la siguiente tabla:

Atributo	Origen Datos	Tipo	Transformación	Observaciones
Tráiler_id		Integer		Este campo debe definirse como campo auto numérico
PlacaTrailer	tblTrailers. strPlacaTrailer	Varchar(10)		

TipoTrailer	tblTrailers. strTipoTrailer	Varchar(2)		
GrupoTrailer	tblGruposTrailers. strDescripcion GrupoTrailer	Varchar(30)		Recuperar de la tabla tblGruposTrailers usando el campo tblTrailers.strCodigoGrupoTrailer
PropiedadTráiler	Campo Calculado	Varchar(7)		Campo que almacenara el valor "PROPIO" cuando el NIT del poseedor sea de TDM, y "TERCERO" cuando no corresponda.

Tabla 25. Atributos Dimensión Trailer

Tabla de Hechos (Ventas)

Los datos de negocio de las operaciones de venta se encuentran registrados en las tablas y tblManifiestos, tblRemesas y tblDetallesRemesas del SAT. El origen de las claves de la tabla de hechos y de los indicadores de negocio, así como las transformaciones a realizar serán descritas en la siguiente tabla:

Indicador	Origen	Tipo	Transformación	Observaciones
Fecha_id		Integer		Fecha de la remesa en formato numérico. Clave foránea de la Dimensión Fecha
Cliente_id		Integer		Clave foránea de la dimensión Cliente
AgenciaDespacho_id		Integer		Clave foránea de la dimensión Agencia
AgenciaVenta_id		Integer		Clave foránea de la dimensión Agencia
UnidadCargue_id		Integer		Clave foránea de la dimensión UnidadCargue
TipoTransporte_id		Integer		Clave foránea de la dimensión TipoTransporte
CiudadOrigen_id		Integer		Clave foránea de la dimensión Ciudad
CiudadDestino_id		Integer		Clave foránea de la dimensión Ciudad
Vehiculo_id		Integer	En caso de no existir la placa	Clave foránea de la dimensión Vehículo

			del vehículo en el manifiesto, se asignara el valor 'NOT_FOUND'.	
Trailer_id		Integer	En caso de no existir la placa del tráiler, se asignara el valor 'NOT_FOUND'.	Clave foránea de la dimensión Tráiler
Remesa	tblRemesas.strCodigoAgencia y tblRemesas.intCons	Varchar(10)	Concatenar en una cadena de caracteres, los valores tblRemesas.strCodigoAgencia y tblRemesas.IntCons	
Manifiesto	tblRemesas.intNumeroManifiesto	Varchar(10)	En caso de no existir el número del manifiesto, se asignara el valor 'NOT_FOUND'.	
Factura	tblRemesas.strCodigoAgenciaFactura y tblRemesas.intNumeroFactura	Varchar(10)	En caso de no existir el número de la factura, se asignara el valor 'NOT_FOUND'.	
KilometrosRuta	tblRutas.dbIKms	Integer	En caso de no existir los kilómetros del manifiesto, se asignara el valor cero (0).	Recuperar de la tabla tblRutas usando el campo tblManifiestos.strCodigoRuta
FletePagado	Campo Calculado	Money		Función Scalar – fnFletePagadoRemesa
FleteCobrado	Campo Calculado	Money		Función Scalar – fnFleteCobradoRemesa
Cantidad	Campo Calculado	Integer		Función escalar – CantidadRemesaPorRemesa
PesoPagado	tblRemesas.dbIKgs Prop	Float		
PesoCobrado	tblRemesas.dbIKgs RealesCobro	Float		
PesoReal	Campo Calculado	Float		Función Scalar – PesoRemesasPorRemesa
AcompanantePagado	Campo Calculado	Money		Funcion Scalar – fnAcompanamientoPagadoRemesa
AcompananteCobrado	tblRemesas.dbITarifaEscoltaCobro	Money		
Intermediacion	Campo Calculado	Money		Funcion Scalar – fnValorIntermediacion
MargenIntermediacion	Campo Calculado	Float		Funcion Scalar – fnMargenIntermediacion
EstadoRemesa	tblRemesas.strEstado	Varchar(10)	Se asignará el valor al estado	

			dependiendo de siguientes condiciones: P: PENDIENTE I: IMPRESA F: FACTURADA	
--	--	--	--	--

Tabla 26. Atributos Tabla Hechos - Ventas

3.5. CONSTRUCCION ETL

3.5.1. Objetivos

El objetivo es construir las estructuras ETL necesarias para la carga de datos en las dimensiones y la tabla de hechos definidas en el diseño.

3.5.2. Metodología

Teniendo en cuenta que el Modelo de Ventas para TDM Transportes S.A.S. tiene como objetivo albergar los datos de las ventas de forma histórica, no cambiantes en el tiempo y los cuales se encuentran almacenados en la base de datos transaccional dbSATW (única fuente de datos transaccional) que dejó de ser operativa a partir del 1 de Marzo/2011, se desarrollarán todos los procesos de Extracción, Transformación y Carga necesarios para poblar directa y/o eficientemente las tablas de dimensión y la tabla de hechos (FACT) que conformaran el DataMart.

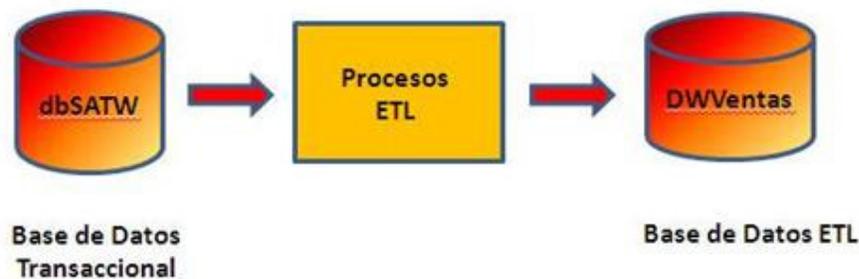


Figura 21. Arquitectura de Alto Nivel del proceso ETL

A continuación se muestra el diseño físico de cada una de las dimensiones:

Borrado de las Tablas de Dimensiones y Hechos

El paquete de borrado de las tablas que conforman el DataMart de Ventas está formada por una consulta SQL elaborada a través de un componente Execute SQL Task del SQL Server Integration Services (SSIS), la cual elimina los registros de todas las tablas en cascada iniciando por la tabla de Hechos (Ventas) y continuando con las tablas de las dimensiones. Posterior al borrado se hace un 'Reset' de los índices de cada una de las tablas con semilla AutoIncremental.

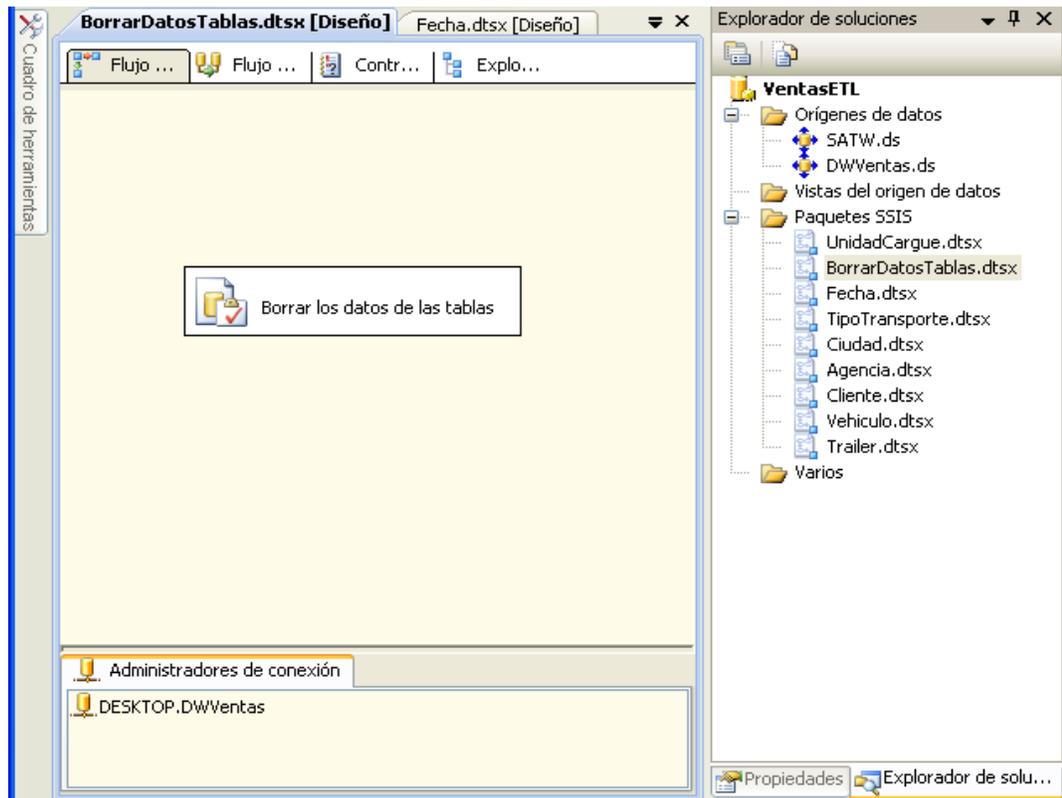


Figura 22. Proceso ETL para borrar los datos de las tablas del DataMart de Ventas.

Dimensión Fecha

La carga de la dimensión fecha tiene como origen de datos una sentencia SQL elaborada a través de un componente Execute SQL Task del SSIS, la cual

genera las columnas de la dimensión Fecha con los respectivos formatos definidos en el documento de requerimientos. El proceso realiza un ciclo que itera las fechas desde Enero 1 de 2009 hasta el Marzo 1 de 2011, insertando cada uno de los registros en la tabla de dimensión `dwd_Fecha`.

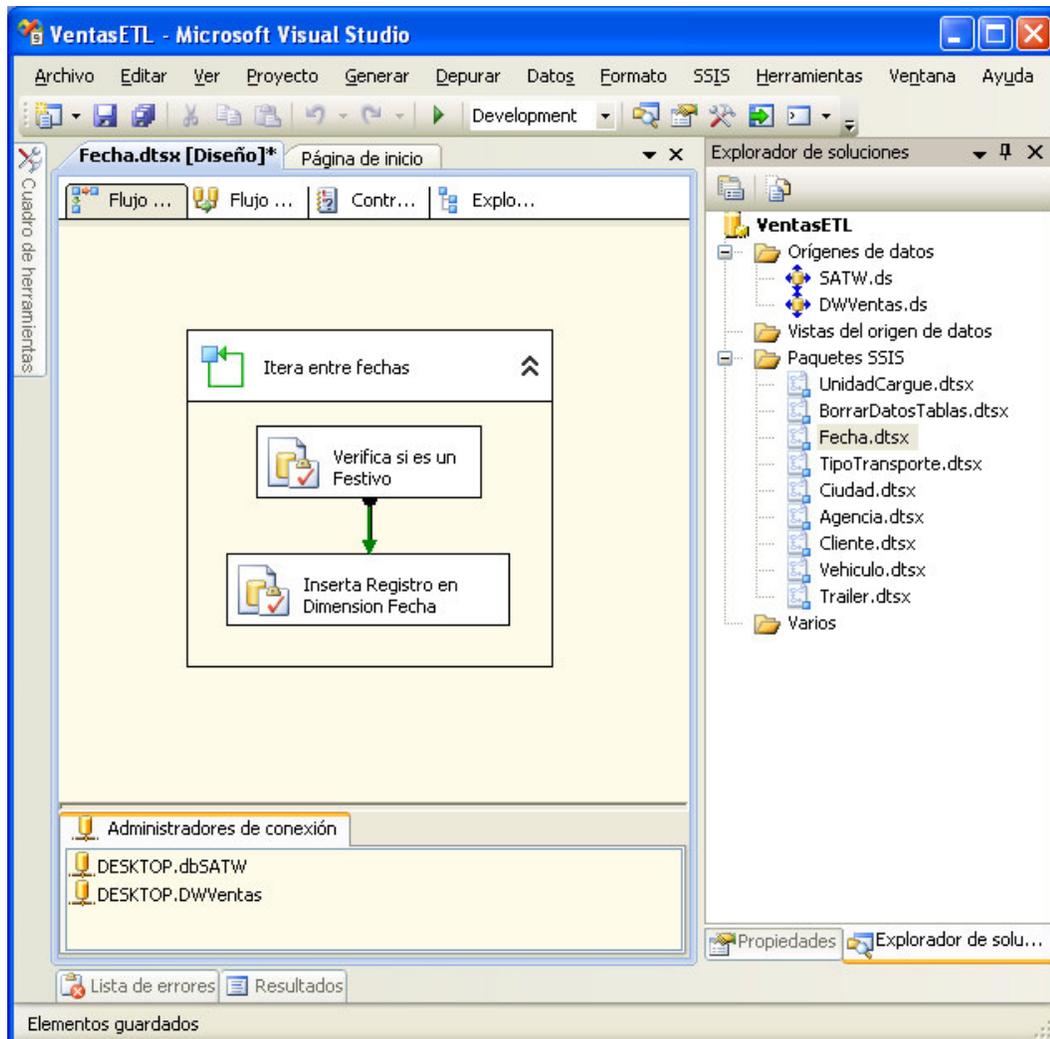


Figura 23. Proceso ETL para cargar la tabla `dwd_Fecha` del DataMart de Ventas

Dimensión Cliente

La carga de la dimensión Cliente tiene como origen de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, el cual

selecciona todos los clientes de la tabla maestra de clientes, extrae las columnas requeridas por la dimensión, establece el formato definido para cada columna y finaliza cargando los registros en la tabla `dwd_Cliente`.

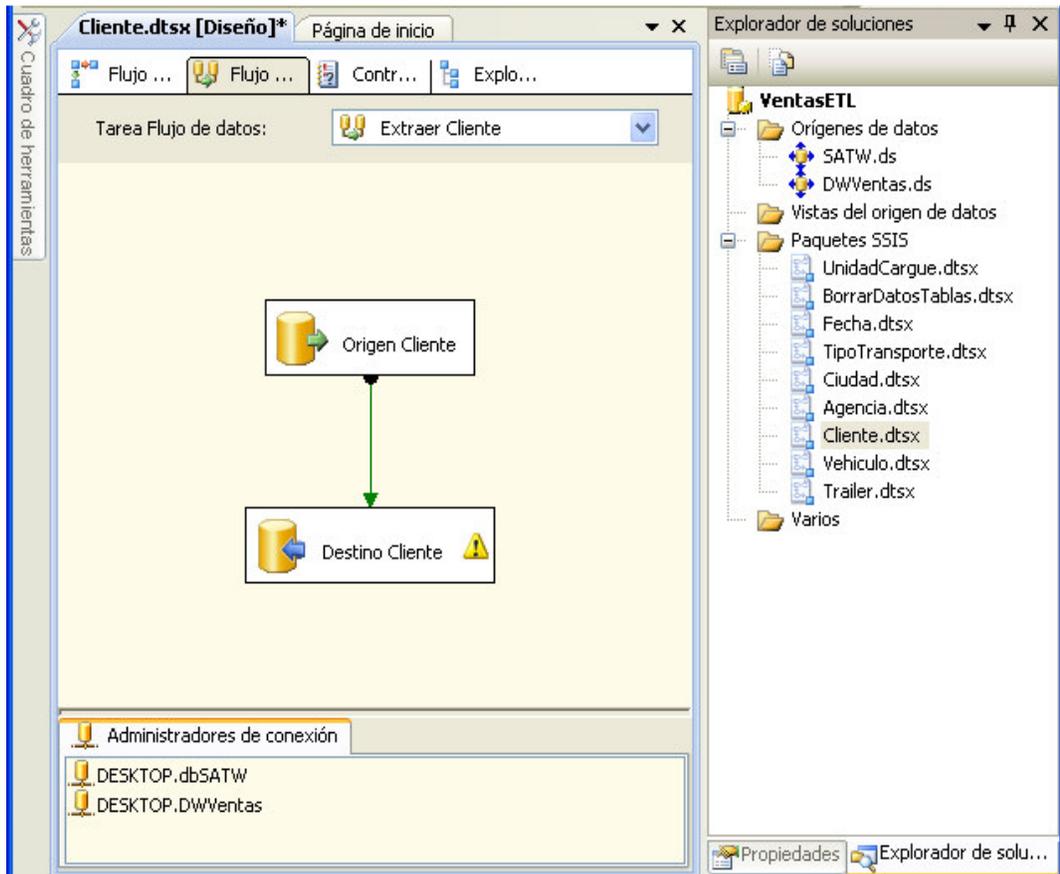


Figura 24. Proceso ETL para cargar la tabla `dwd_Cliente` del DataMart de Ventas

Dimensión Agencia

La carga de la dimensión Agencia tiene como origen de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, la cual selecciona todas las Agencias de la tabla maestra de Agencias, extrae las columnas necesarias para la carga a la dimensión, establece el formato requerido para cada columna y finaliza cargando los registros en la tabla `dwd_Agencia`.

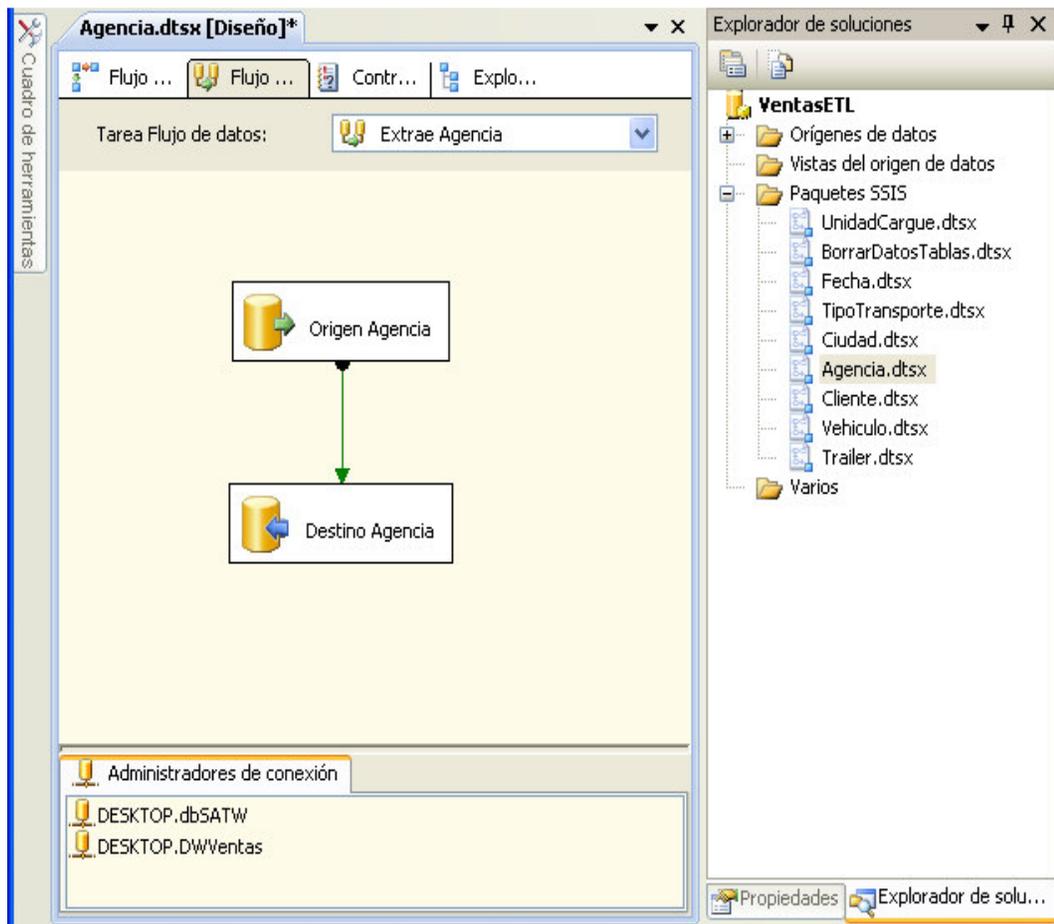


Figura 25. Proceso ETL para cargar la tabla dwd_Agencia del DataMart de Ventas

Dimensión Unidad de Cargue

La carga de la dimensión Unidad de Cargue tiene como origen de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, la cual selecciona todas las Unidades de Cargue unido a los servicios Solicitados en la Solicitud de Servicio, agrupa diferentes unidades de cargue en una sola Unidad de acuerdo a las reglas del negocio y, finaliza cargando los registros en la tabla dwd_UnidadCargue.

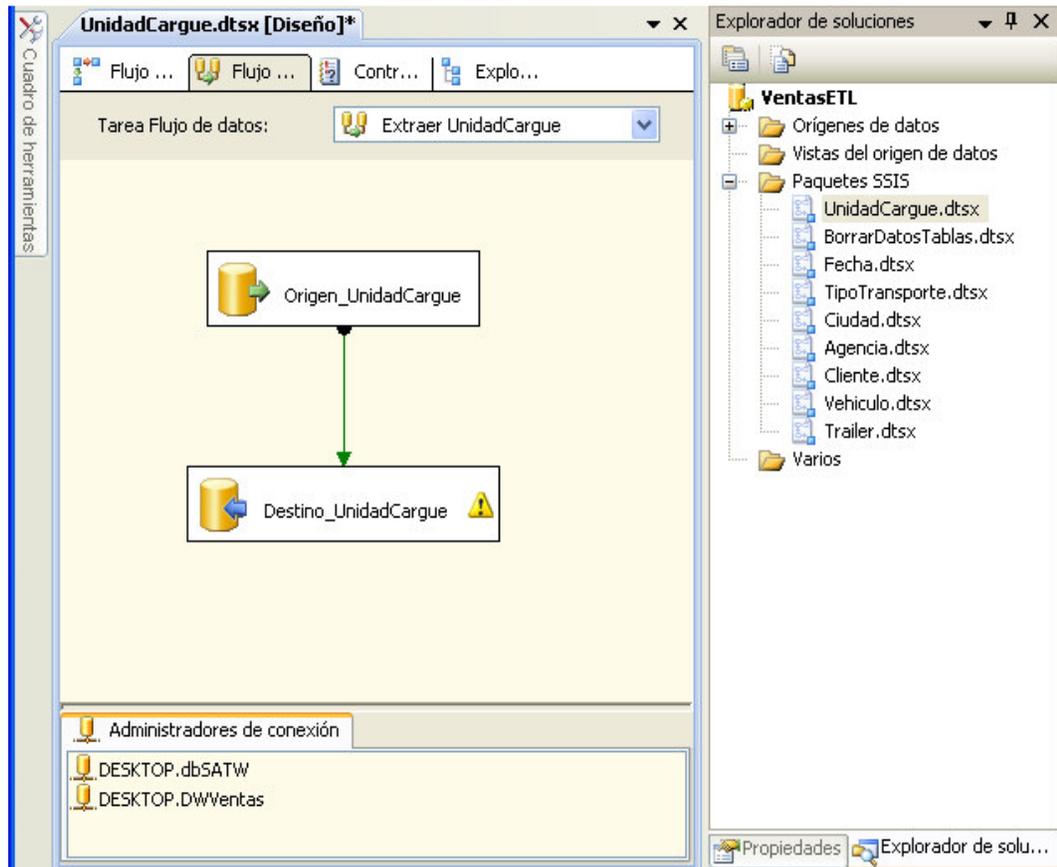


Figura 26. Proceso ETL para cargar la tabla dwd_UnidadCargue del DataMart de Ventas

Dimensión Tipo Transporte

La carga de la dimensión Tipo Transporte tiene como origen de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, la cual selecciona todos los Tipos de Transporte de la tabla maestra de Tipos de Transporte, extrae las columnas necesarias para la carga a la dimensión, establece el formato requerido para cada columna y finaliza cargando los registros en la tabla dwd_TipoTransporte.

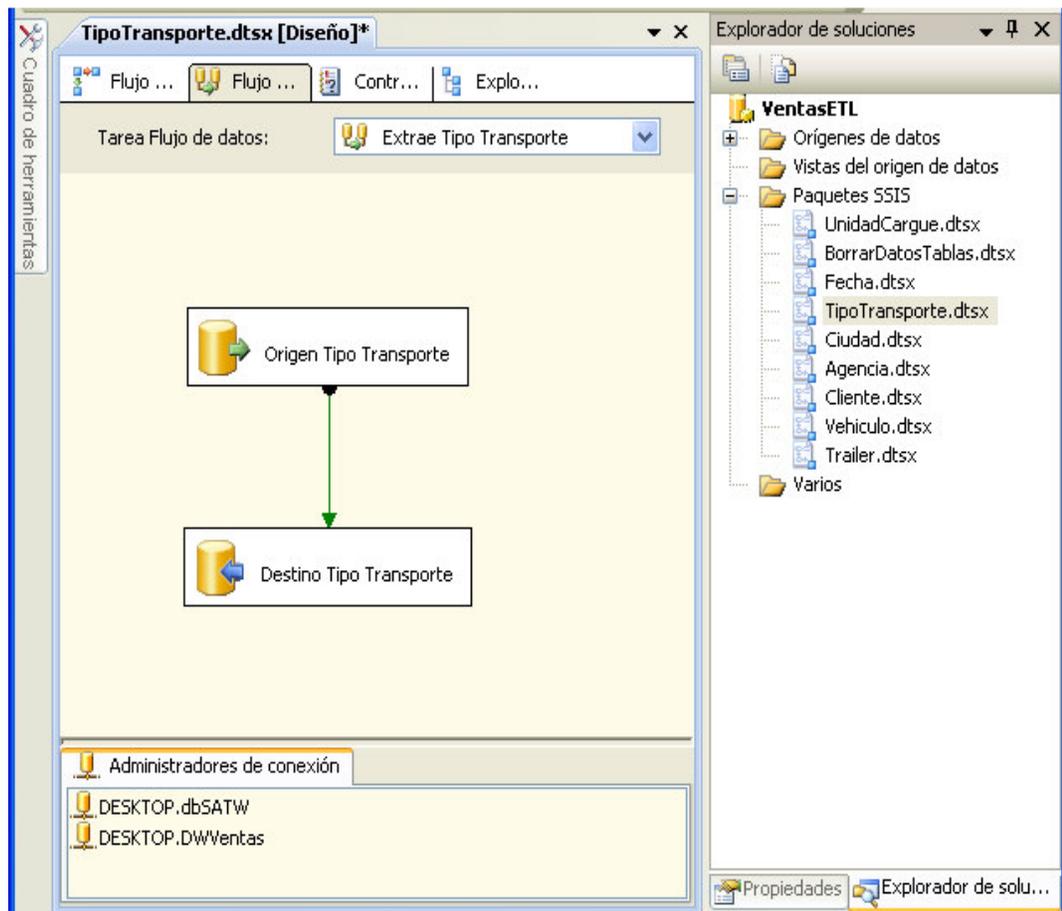


Figura 27. Proceso ETL para cargar la tabla dwd_TipoTransporte del DataMart de Ventas

Dimensión Ciudad

La carga de la dimensión Ciudad tiene como origen de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, la cual selecciona todas las Ciudades de la tabla maestra de Ciudades, extrae las columnas necesarias para la carga a la dimensión, establece el formato requerido para cada columna y finaliza cargando los registros en la tabla dwd_Ciudad.

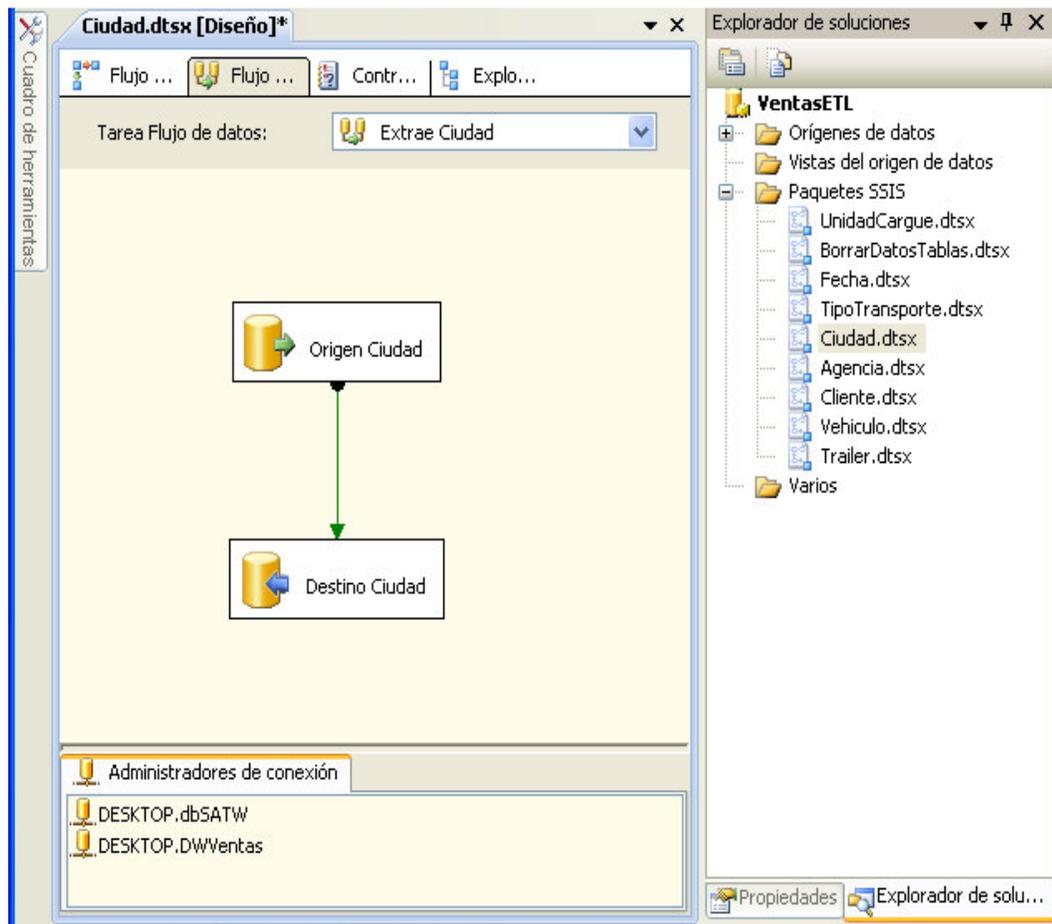


Figura 28. Proceso ETL para cargar la tabla dwd_Ciudad del DataMart de Ventas

Dimensión Vehículo

La carga de la dimensión Vehículo tiene como origen de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, la cual selecciona todos los Vehículos de la tabla maestra de Vehículos, extrae las columnas necesarias para la carga a la dimensión, establece el formato requerido para cada columna y finaliza cargando los registros en la tabla dwd_Vehiculo.

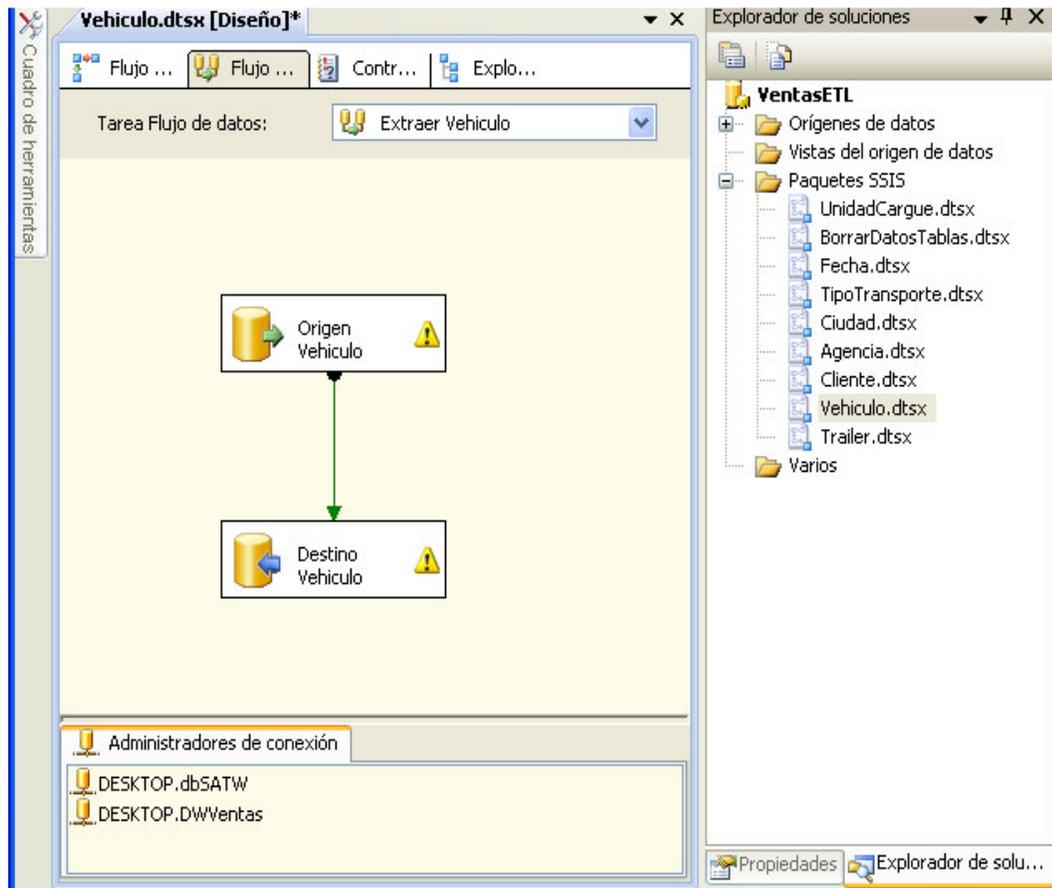


Figura 29. Proceso ETL para cargar la tabla dwd_Vehiculo del DataMart de Ventas

Dimensión Tráiler

La carga de la dimensión Tráiler tiene como origen de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, la cual selecciona todos los Tráiler de la tabla maestra de Tráiler, extrae las columnas necesarias para la carga a la dimensión, establece el formato requerido para cada columna y finaliza cargando los registros en la tabla dwd_Trailer.

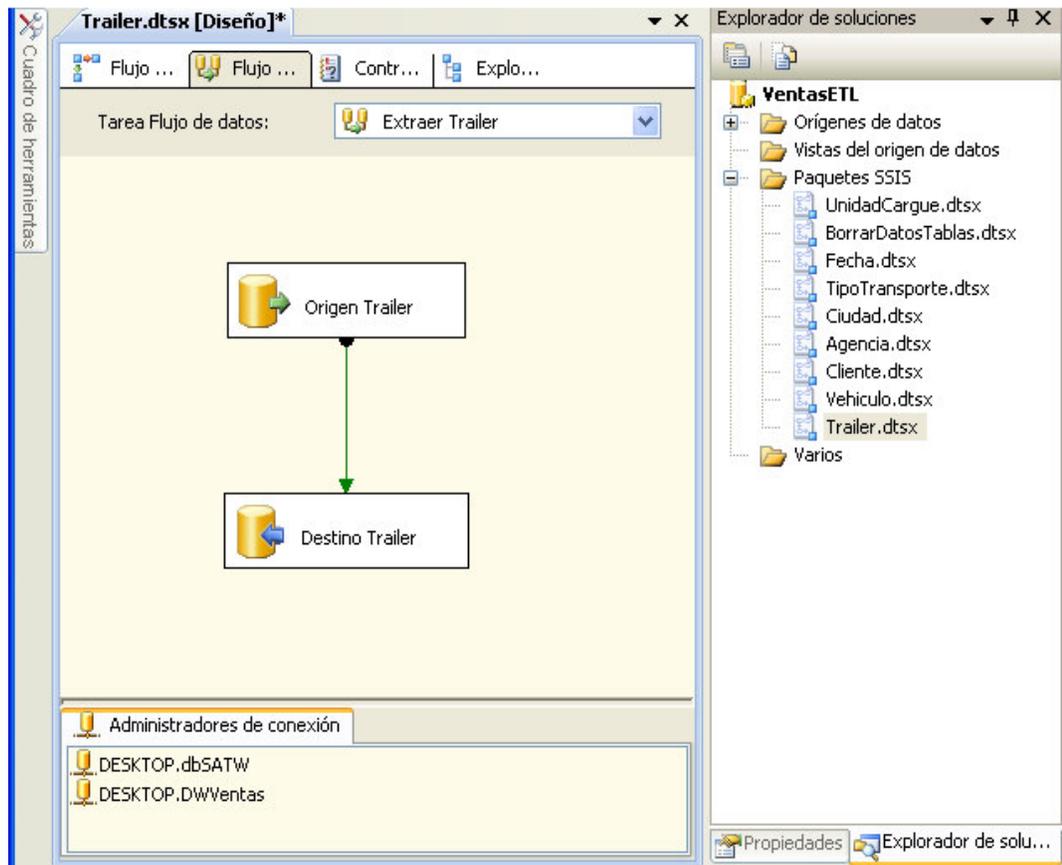


Figura 30. Proceso ETL para cargar la tabla dwd_Trailer del DataMart de Ventas

Tabla de Hechos de Ventas

La carga de la tabla de hechos de ventas tiene como fuente de datos una sentencia SQL elaborada a través de un componente OLE DB Source del SSIS, la cual selecciona todas los campos claves que conforman la remesa, también extrae las claves de las dimensiones, los campos de las dimensiones degeneradas y valores de medidas, establece el formato requerido para cada columna, utiliza sentencias de búsqueda de las claves en las dimensiones y en caso de no existir lleva los registros con problemas a un archivo plano en formato Excel y finaliza cargando los registros en la tabla dwh_Ventas.

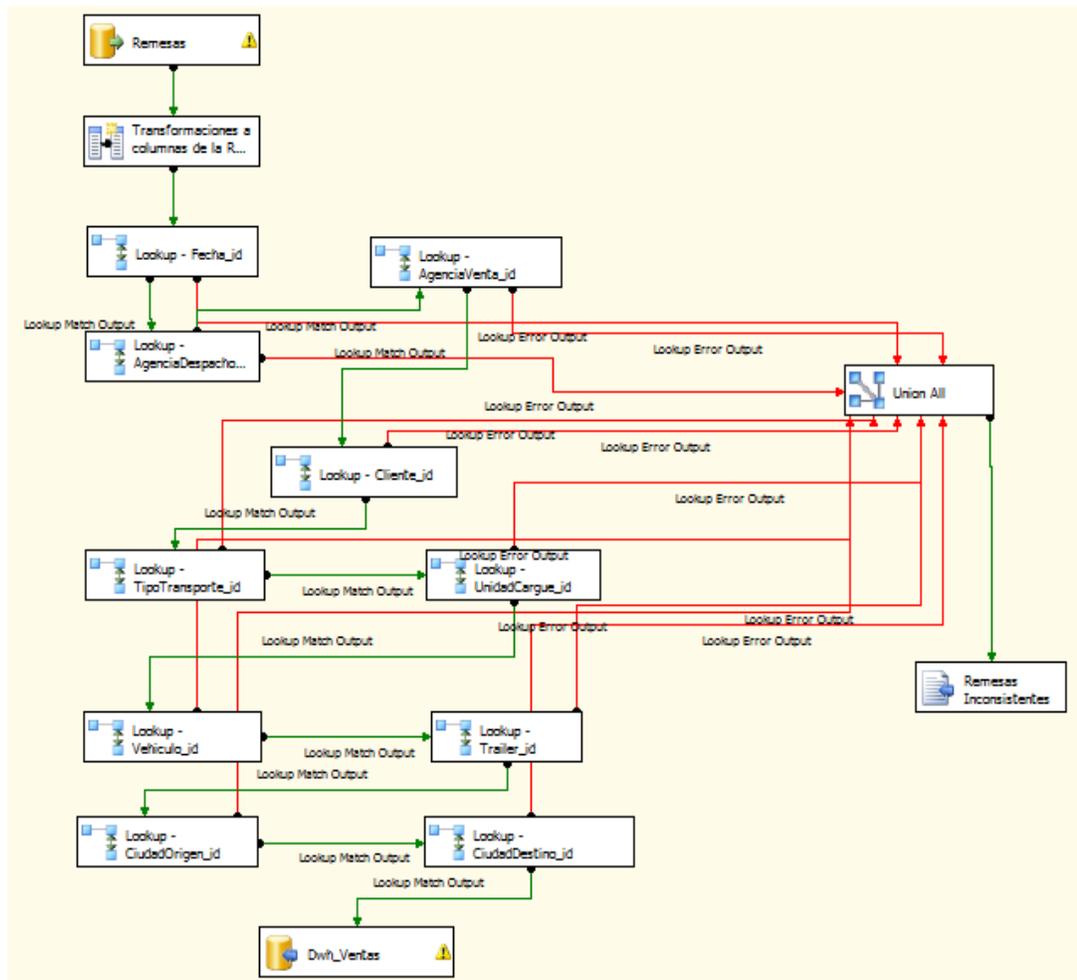


Figura 31. Proceso ETL para cargar la tabla dwd_Ventas del DataMart de Ventas

Proceso de Carga de la ETL

El proceso sincronizado y completo encargado de poblar la tabla de hecho y todas las tablas de dimensión que conformaran el DataMart de Ventas, se realiza basado en la siguiente estructura:



Figura 32. Estructura de directorios del proceso de Carga de la ETL.

1. Archivos XML de Configuración

En la estructura de directorios de la solución de carga de las ETL de Ventas, reside el directorio \Configuracion XML, el cual contiene los archivos de configuración que permiten parametrizar las variables y las cadenas de conexión requeridas por los paquetes de SSIS para facilitar la conexión hacia las diferentes fuentes de datos y garantizar la ejecución correcta del componente. Así mismo, se encuentra presente el archivo XML que describe la lista de paquetes o componentes asociados al proceso.

[VentasETL]\Configuracion XML

ListaDePaquetes.XML: Archivo XML que contiene la lista de todos los paquetes que deben ser ejecutados, el orden de sincronización y su respectivo archivo de configuración.

```
<?xml version="1.0" ?>
<ETLConfiguracion>
  <Paquete Nombre="BorrarDatosTablas.dtsx" Orden="1"
  ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="Fecha.dtsx" Orden="2" ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="Agencia.dtsx" Orden="3" ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="Ciudad.dtsx" Orden="4" ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="Cliente.dtsx" Orden="5" ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="TipoTransporte.dtsx" Orden="6" ArchivoXML="GeneralXMLConfig.dtsConfig"
  />
  <Paquete Nombre="UnidadCargue.dtsx" Orden="7" ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="Vehiculo.dtsx" Orden="8" ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="Trailer.dtsx" Orden="9" ArchivoXML="GeneralXMLConfig.dtsConfig" />
  <Paquete Nombre="CargaTablaHechos.dtsx" Orden="10"
  ArchivoXML="TablaDeHechoXMLConfig.dtsConfig" />
</ETLConfiguracion>
```

GeneralXMLConfig.dtsConfig: Archivo XML que contiene la información de configuración necesaria para la ejecución de los paquetes asociados al borrado de las tablas y a la carga de las tablas de dimensión.

```
<?xml version="1.0" ?>
<DTSConfiguration>
  <Configuration ConfiguredType="Property"
    Path="\Package.Connections[Conexion.dbSATW].Properties[ConnectionString]"
    ValueType="String">
    <ConfiguredValue>Data Source=SERVER;Initial
      Catalog=dbSATW;Provider=SQLNCLI10.1;Integrated Security=SSPI;Auto
      Translate=False;Application Name=SSIS-Fecha-{B1A15BD7-CEF6-4B20-8536-
      20073D4B7382} dbSATW;</ConfiguredValue>
  </Configuration>
  <Configuration ConfiguredType="Property"
    Path="\Package.Connections[Conexion.DWVentas].Properties[ConnectionString]"
    ValueType="String">
    <ConfiguredValue>Data Source=SERVER;Initial
      Catalog=DWVentas;Provider=SQLNCLI10.1;Integrated Security=SSPI;Auto
      Translate=False;Application Name=SSIS-Fecha-{F4B236E4-5A92-4EA5-AB20-
      90F919DA0188} DWVentas;</ConfiguredValue>
  </Configuration>
  <Configuration ConfiguredType="Property"
    Path="\Package.Variables[User::FechaFinal].Properties[Value]" ValueType="DateTime">
    <ConfiguredValue>3/1/2011</ConfiguredValue>
  </Configuration>
  <Configuration ConfiguredType="Property"
    Path="\Package.Variables[User::FechaInicial].Properties[Value]" ValueType="DateTime">
    <ConfiguredValue>1/1/2009</ConfiguredValue>
  </Configuration>
  <Configuration ConfiguredType="Property"
    Path="\Package.Variables[User::Festivo].Properties[Value]" ValueType="Int32">
    <ConfiguredValue>0</ConfiguredValue>
  </Configuration>
</DTSConfiguration>
```

TablaDeHechoXMLConfig.dtsConfig: Archivo XML que contiene la información de configuración necesaria para la ejecución del paquete que realiza la carga de la tabla de Hechos. Su estructura es similar a la definida previamente en el archivo GeneralXMLConfig.dtsConfig.

2. Paquetes

En el directorio Paquetes de la solución, se encuentran todos los componentes o paquetes desplegados desde el SSIS y que realizarán la carga completa de la ETL de Ventas.

Los archivos que residen en el directorio Paquetes son:

- Agencia.dtsx
- BorrarDatosTablas.dtsx
- CargaTablaHechos.dtsx
- Ciudad.dtsx
- Cliente.dtsx
- Fecha.dtsx
- TipoTransporte.dtsx
- Trailer.dtsx
- UnidadCargue.dtsx
- Vehiculo.dtsx

Es importante anotar que la información del archivo ListaDePaquetes.XML se relaciona directamente con los paquetes que residen en este directorio.

3. Archivo Inconsistencias

En este directorio residen los archivos en los cuales se reportan las inconsistencias generadas a nivel de datos al momento de ejecutarse los paquetes.

En el archivo Error_Output_Ventas.txt quedan registradas todas las remesas que presentaron inconsistencias al relacionarse con las dimensiones.

4. Automatización del Proceso

Para automatizar el proceso de carga de la ETL de Ventas se hace uso de los procedimientos almacenados SpEjecutarPaqueteSSIS y SpCargarETLVentas.

El Procedimiento Almacenado SpEjecutarPaqueteSSIS ejecuta un paquete de SSIS en forma local y genera una bitácora con la información detallada de seguimiento de la ejecución del proceso.

Este procedimiento recibe como parámetros la ruta del paquete a ejecutar y su archivo de configuración XML.

```
EXEC [dbo].[SpEjecutarPaqueteSSIS]
    @ssisPkgFilePath = N'C:\VentasETL\Paquetes\Agencia.dtsx',
    @ssisPkgXMLConfigFilePath = N'C:\VentasETL\Configuracion XML
\GeneralXMLConfig.dtsConfig'
```

El procedimiento almacenado SpCargarETLVentas, lee del archivo de configuración ListaDePaquetes.XML, la información de los paquetes a ejecutar, su orden y su respectivo archivo de configuración.

Posteriormente, invoca de forma iterativa el procedimiento SpEjecutarPaqueteSSIS para ejecutar independiente y secuencialmente cada uno de los paquetes que conforman la ETL de Ventas. A medida que se ejecutan los paquetes, se va dejando una bitácora detallada de su ejecución.

Este procedimiento recibe como parámetros la ruta donde se encuentran los paquetes a ejecutar y la ruta donde se encuentran los archivos de configuración XML.

```
EXEC [dbo].[SpCargarETLVentas]
    @ssisPkgPath = N'C:\VentasETL\Paquetes\' ,
    @ssisPkgXMLConfigPath = N'C:\VentasETL\Configuracion XML\'
```

La ejecución del Procedimiento almacenado SpCargarETLVentas, el cual realiza todo el proceso de carga de las tablas de dimensión y de la tabla de Hecho, se puede realizar de dos formas:

1. Desde el SQL Server Management Studio (SSMS), se selecciona el procedimiento almacenado (Stored Procedure) SpCargarETLVentas y se ejecuta ingresando los parámetros solicitados.

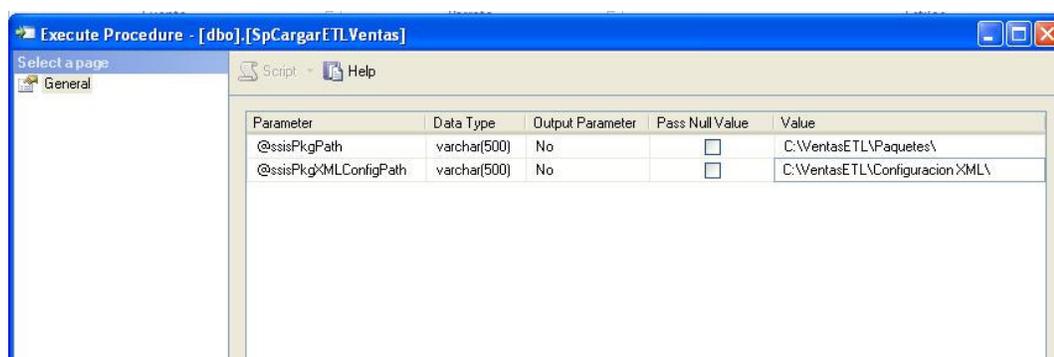


Figura 33. Interface para Ejecutar Procedimiento Almacenado SpCargarETLVentas.

2. Desde el SQL Server Agent en el SSMS, se define un Job para automatizar el proceso de carga de la ETL de Ventas, si es requerido.

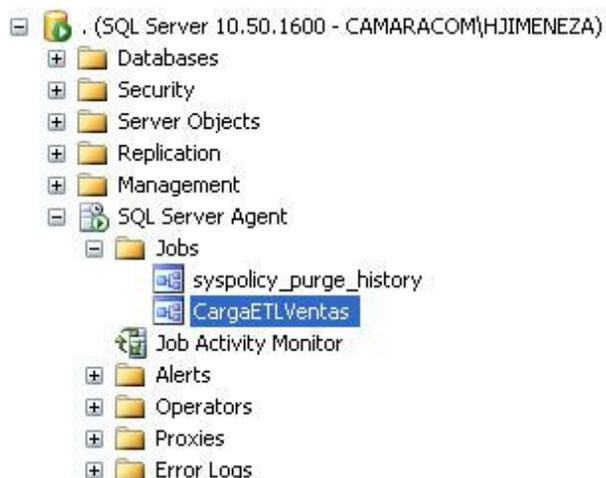


Figura 34. Definición de Job en el Agente SQL Server para automatizar el Proceso de carga de la ETL de Ventas.

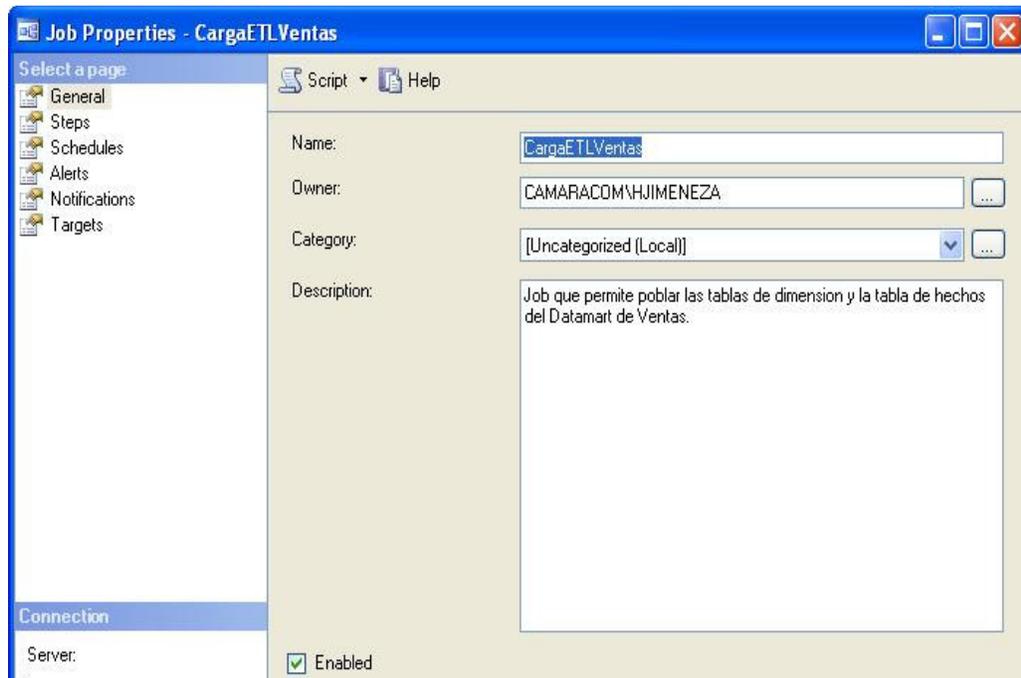


Figura 35. Parametrización básica del Job CargaETLVentas.

Se programa el procedimiento almacenado SpCargarETLVentas con sus respectivos parámetros.

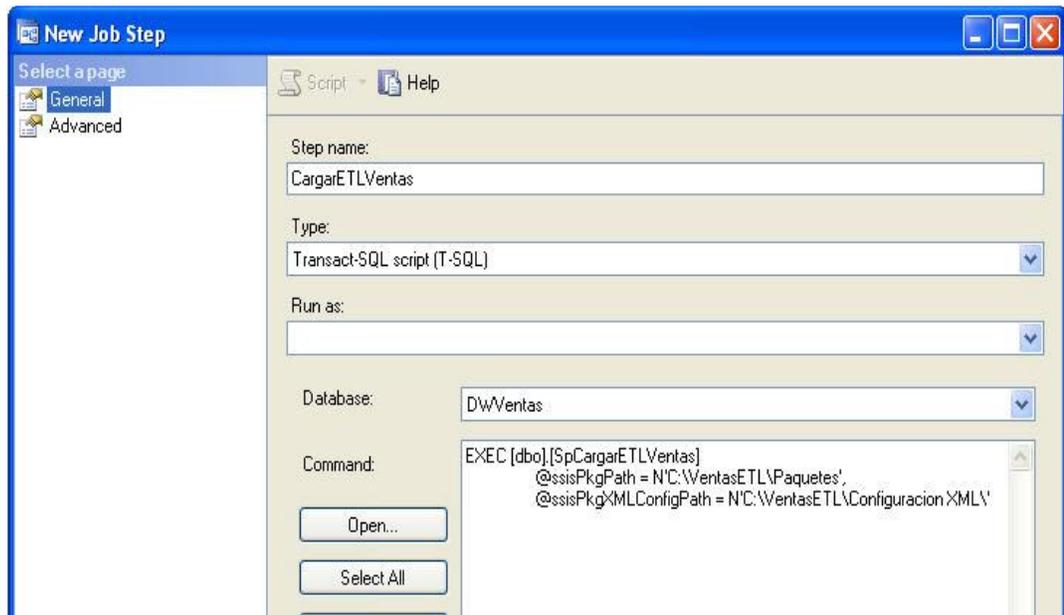


Figura 36. Configuración del Comando SQL del Job CargaETLVentas.

En este paso, se puede establecer opcionalmente (Advanced), si la bitácora de ejecución se lleva a una tabla de Log o a un archivo de salida.

Por último, si el Job va a manejar algún horario especial de ejecución, entonces se procede a su programación.

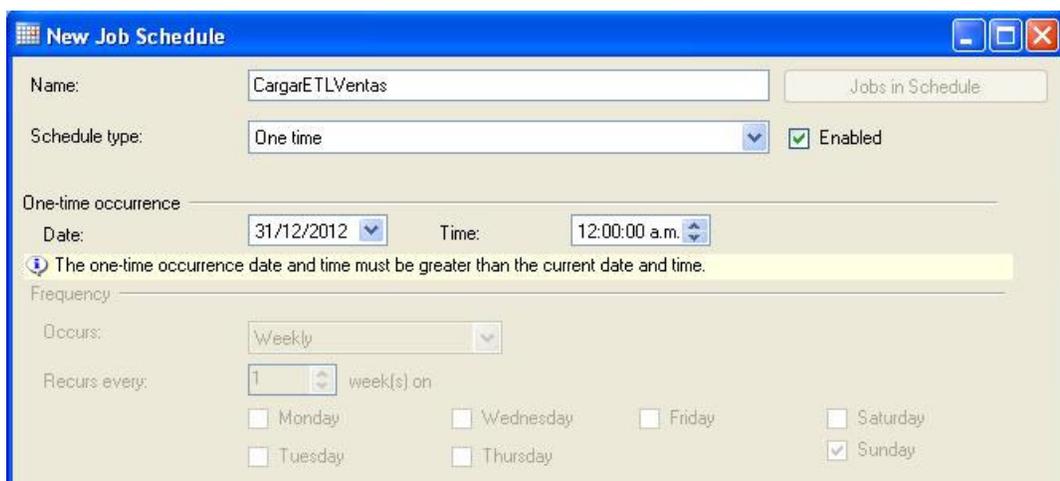


Figura 37. Automatización del Job CargaETLVentas.

4.6. DOCUMENTACIÓN OLAP DEL CUBO DE VENTAS

4.6.1. Objetivos

El objetivo es realizar en forma detallada la documentación técnica de los componentes que conforman el cubo OLAP de Ventas.

4.6.2. Documentación OLAP del Cubo de Ventas

La documentación técnica detallada del cubo OLAP de Ventas sigue la estructura de solución del SQL Server Analysis Server (SSAS) que a continuación se define:

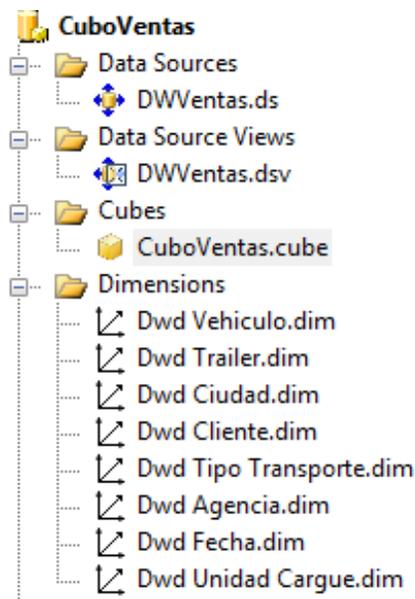


Figura 38. Estructura de Solución del Cubo OLAP de Ventas.

Orígenes de Datos

El origen de datos proporciona toda la información que el cubo requiere para conectarse exitosamente a la base de datos donde residen los esquemas en estrella.

Origen de Datos: DWVentas.ds

Propiedades del origen de datos DWVentas.ds

Nombre	Valor
Tipo	Origen De Datos Relacional
Nombre	DWVentas
Descripción	Origen de Datos del Cubo OLAP de Ventas
Cadena de Conexión	Provider=SQLNCLI10.1;Data Source=;<<SERVER>>;Integrated Security=SSPI;Initial Catalog=DWVentas
Modo Suplantación	ImpersonateServiceAccount (Utiliza cuenta de servicio)

Tabla 27. Propiedades del Origen de Datos DWVentas

DataSources View (Vista de los orígenes de datos)

La vista del origen de datos es un modelo de datos lógico de tablas, vistas y consultas relacionadas de uno o varios orígenes de datos que proporcionan la información para un cubo particular. Permite combinar tablas de tantos orígenes de datos como sea necesario para reunir los datos que el cubo necesita.

Vista de Origen de Datos: DWVentas.dsv

Propiedades de la vista de origen de datos DWVentas.dsv

Nombre	Valor
Nombre	DWVentas
Descripción	Modelo de datos lógico de tablas.
Origen de Datos (DataSource)	DWVentas

Nombre	Valor
Esquema	Estrella

Tabla 28. Propiedades de la Vista de Origen de Datos DWVentas

Tablas

Nombre	Nombre Esquema db	Nombre Tabla	Tipo
dbo_dwd_Agencia	dbo	dwd_Agencia	Tabla
dbo_dwd_Ciudad	dbo	dwd_Ciudad	Tabla
dbo_dwd_Cliente	dbo	dwd_Cliente	Tabla
dbo_dwd_Fecha	dbo	dwd_Fecha	Tabla
dbo_dwd_TipoTransporte	dbo	dwd_TipoTransporte	Tabla
dbo_dwd_Trailer	dbo	dwd_Trailer	Tabla
dbo_dwd_UnidadCargue	dbo	dwd_UnidadCargue	Tabla
dbo_dwd_Vehiculo	dbo	dwd_Vehiculo	Tabla
dbo_dwh_Ventas	dbo	dwh_Ventas	Tabla
dbo_viAgenciaDespacho	dbo	viAgenciaDespacho	Vista
dbo_viAgenciaVenta	dbo	viAgenciaVenta	Vista
dbo_viCiudadDestino	dbo	viCiudadDestino	Vista
dbo_viCiudadOrigen	dbo	viCiudadOrigen	Vista

Tabla 29. Esquema de la Tabla de Hechos - Ventas

La siguiente figura ilustra el esquema de la vista del origen de datos generado por el SSIS.

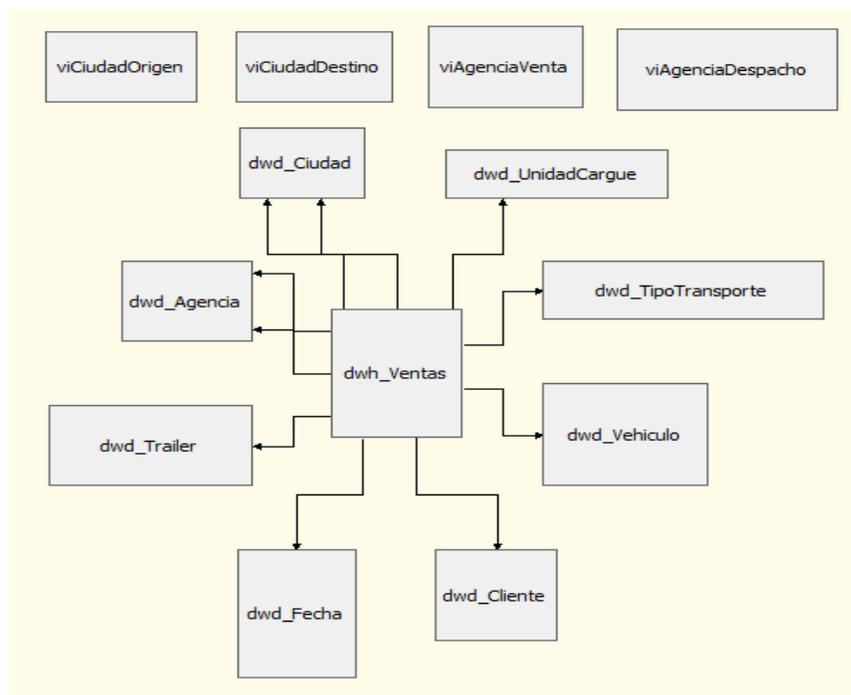


Figura 39. Vista del Origen de datos de DWVentas

Dimensiones

Cada cubo tiene una o más dimensiones, cada una de ellas basada en una o más tablas de dimensión. Una dimensión en el cubo representa una categoría para analizar datos del negocio.

El cubo OLAP de Ventas está conformado por las siguientes 8 dimensiones MOLAP:

Dimensiones

Nombre

Nombre
Dwd Agencia
Dwd Ciudad
DDwd Cliente
Dwd Fecha
Dwd Tipo Transporte
<u>Dwd Trailer</u>
Dwd Unidad Cargue
Dwd Vehiculo

Tabla 30. Dimensiones MOLAP

A continuación se detalla de manera independiente cada una de las dimensiones:

Dimensión Dwd Agencia

Atributos

Nombre	Uso	Jerarquías
Agencia		
Agencia Id	Key	
Ciudad Agencia		HI_Agencia_Dpto, HI_Agencia_Nombre
Departamento Agencia		HI_Agencia_Dpto, HI_Agencia_Nombre
Nombre Agencia		HI_Agencia_Dpto, HI_Agencia_Nombre

Tabla 31. Esquema de la Dimensión Agencia

Las Jerarquías se definen por una secuencia de campos y a menudo se usan para determinar las filas o columnas de una tabla dinámica (Pivot Table) cuando se consulta un cubo.

Jerarquías

Nombre	Niveles
HI_Agencia_Dpto	1. Departamento Agencia 2. Ciudad Agencia 3. Nombre Agencia
HI_Agencia_Nombre	1. Nombre Agencia 2. Ciudad Agencia 3. Departamento Agencia

Tabla 32. Jerarquías de la Dimensión Agencia

Gráfico de Dependencias



Figura 40. Gráfico de Dependencias de la Dimensión Dwd Agencia

Tabla de Convenciones de colores

	Cubo OLAP
	Base de Datos

	Vista de Origen de Datos
	Dimensión
	Origen de Datos Relacional

Tabla 33. Tabla de Convenciones de Colores

Dimensión Dwd Ciudad

Atributos

Nombre	Uso	Jerarquías
Ciudad		
Ciudad Id	Key	
Departamento Ciudad		HI_Ciudad
Nombre Ciudad		HI_Ciudad

Tabla 34. Esquema de la Dimensión Ciudad

Jerarquías

Nombre	Niveles
HI_Ciudad	<ol style="list-style-type: none"> 1. Departamento Ciudad 2. Nombre Ciudad

Tabla 35. Jerarquía de la Dimensión Ciudad

Gráfico de Dependencias

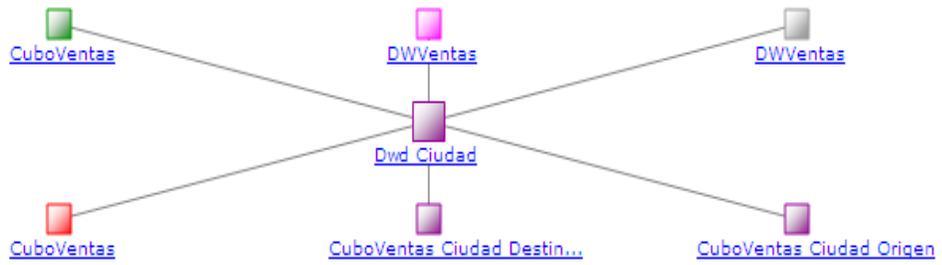


Figura 41. Gráfico de Dependencias de la Dimensión Dwd Ciudad

Dimensión Dwd Cliente

Atributos

Nombre	Uso	Jerarquías
Asesor Comercial		
Cliente Id	Key	
Grupo Cliente		
Identificacion Cliente		
Nombre Cliente		
Ramo		

Tabla 36. Esquema de la Dimensión Cliente

Gráfico de Dependencias

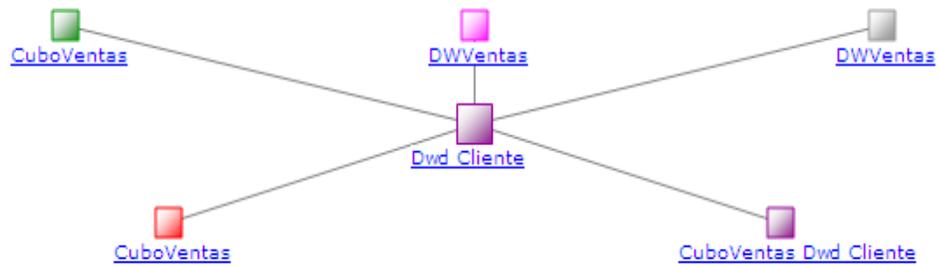


Figura 42. Gráfico de Dependencias de la Dimensión Dwd Cliente

Dimensión Dwd Fecha

Atributos

Nombre	Tipo (SSIS)	Uso	Jerarquías
Ano	Years		HI_Calendario_Ano_Mes, HI_Calendario_Ano_Semestre, HI_Calendario_Ano_Cuatrimestre, HI_Calendario_Ano_Semestre_Mes, HI_Calendario_Ano_Cuatrimestre_Me s, HI_Calendario_Ano_Trimestre, HI_Calendario_Ano_Trimestre_Mes, HI_Calendario_Ano_Semana, HI_Calendario_Ano_DiaSemana
Cuatrimestre	Quarters		
Cuatrimestre Ano	QuarterOfYear		HI_Calendario_Ano_Cuatrimestre, HI_Calendario_Ano_Cuatrimestre_Me s
Dia Ano	DayOfYear		
Dia Mes	DayOfMonth		

Nombre	Tipo (SSIS)	Uso	Jerarquías
Dia Semana	DayOfWeek		
Fecha	Days		
Fecha Id		Key	
Festivo	IsHoliday		
Fin Semana	IsWeekDay		
Mes	Months		
Mes Ano	MonthOfYear		
Nombre Dia Semana	ReportingDayOfWeek		HI_Calendario_Ano_DiaSemana
Nombre Fecha	ReportingDate		
Nombre Mes	ReportingMonths		HI_Calendario_Ano_Mes, HI_Calendario_Ano_Semestre_Mes, HI_Calendario_Ano_Cuatrimestre_Mes, HI_Calendario_Ano_Trimestre_Mes
Semana	Weeks		
Semana Ano	WeekOfYear		HI_Calendario_Ano_Semana
Semestre	HalfYears		
Semestre Ano	HalfYearOfYear		HI_Calendario_Ano_Semestre, HI_Calendario_Ano_Semestre_Mes
Trimestre	Trimesters		

Nombre	Tipo (SSIS)	Uso	Jerarquías
Trimestre Ano	TrimesterOfYear		HI_Calendarario_Ano_Trimestre, HI_Calendarario_Ano_Trimestre_Mes

Tabla 37. Esquema de la Dimensión Fecha

Inteligencia Dimensional en la dimensión Dwd Fecha

Uno de las formas más comunes a través de la cual los datos se resumen en un cubo es por Tiempo y es muy importante destacar que los cubos saben mucho acerca de él.

El SQL Server Analysis Services (SSIS) para responder mejor a las preguntas relacionadas con el Tiempo, necesita conocer cuáles de las dimensiones almacena la información del tiempo y cuales campos en la dimensión de tiempo corresponde a cuales unidades de tiempo (Ver Tipo SSIS en la dimensión Dwd Fecha). En el SSIS, a través de la inteligencia de dimensión (dimension intelligence), se define la dimensión de tiempo - Dwd fecha y se mapean sus atributos con los tipos de atributos de la dimensión Tiempo (Time).

Jerarquías

Nombre	Jerarquía
HI_Calendarario_Ano_Mes	1. Ano 2. Nombre Mes
HI_Calendarario_Ano_Semestre	1. Ano 2. Semestre Ano
HI_Calendarario_Ano_Cuatrimestre	1. Ano 2. Cuatrimestre Ano

Nombre	Jerarquía
HI_Calendarario_Ano_Semestre_Mes	1. Ano 2. Semestre Ano 3. Nombre Mes
HI_Calendarario_Ano_Cuatrimestre_Mes	1. Ano 2. Cuatrimestre Ano 3. Nombre Mes
HI_Calendarario_Ano_Trimestre	1. Ano 2. Trimestre Ano
HI_Calendarario_Ano_Trimestre_Mes	1. Ano 2. Trimestre Ano 3. Nombre Mes
HI_Calendarario_Ano_Semana	1. Ano 2. Semana Ano
HI_Calendarario_Ano_DiaSemana	1. Ano 2. Nombre Dia Semana

Tabla 38. Jerarquía de la Dimensión Fecha

Gráfico de Dependencias

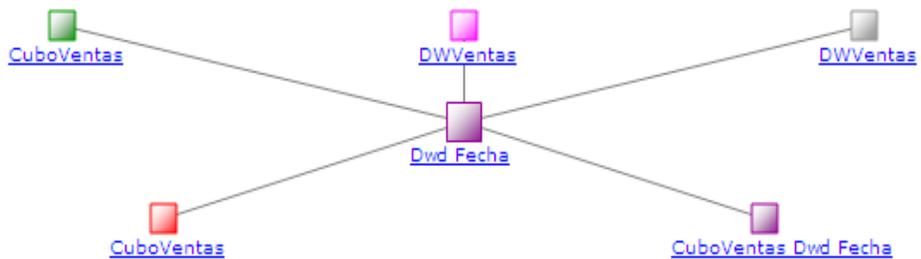


Figura 43. Gráfico de Dependencias de la Dimensión Dwd Fecha

- **Dimensión Dwd Tipo Transporte**

Atributos

Nombre	Uso	Jerarquías
Descripcion Transporte	Tipo	
Tipo Transporte		
Tipo Transporte Id	Key	

Tabla 39. Esquema de la Dimensión Tipo Transporte

Gráfico de Dependencias

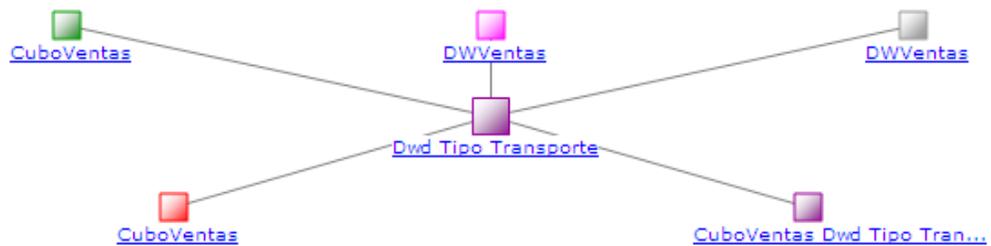


Figura 44. Gráfico de Dependencias de la Dimensión Dwd Tipo Transporte

Dimensión Dwd Trailer

Atributos

Nombre	Uso	Jerarquías
Grupo Trailer		
Placa Trailer		HI_PropiedadTrailer
Propiedad Trailer		HI_PropiedadTrailer

Nombre	Uso	Jerarquías
Tipo Trailer		HI_PropiedadTrailer
Trailer Id	Key	

Tabla 40. Esquema de la Dimensión Trailer

Jerarquías

Nombre	Niveles
HI_PropiedadTrailer	<ol style="list-style-type: none"> 1. Propiedad Trailer 2. Tipo Trailer 3. Placa Trailer

Tabla 41. Esquema de la Dimensión Trailer

Gráfico de Dependencias

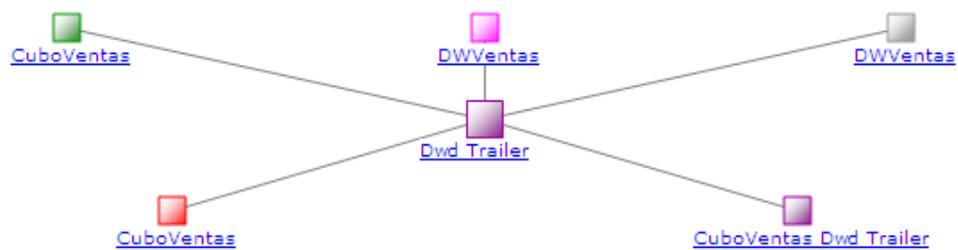


Figura 45. Gráfico de Dependencias de la Dimensión Dwd Trailer

Dimensión Dwd Vehículo

Atributos

Nombre	Uso	Jerarquías
--------	-----	------------

Nombre	Uso	Jerarquías
Documento Propietario Vehiculo		
Modelo Vehiculo		
Nombre Propietario Vehiculo		HI_PropiedadVehiculoTercero
Placa Vehiculo		HI_PropiedadVehiculoTercero, HI_PropiedadVehiculo
Propiedad Vehiculo		HI_PropiedadVehiculoTercero, HI_PropiedadVehiculo
Tipo Vehiculo		HI_PropiedadVehiculoTercero, HI_PropiedadVehiculo
Vehiculo Id	Key	

Tabla 42. Esquema de la Dimensión Vehículo

Jerarquías

Nombre	Niveles
HI_PropiedadVehiculoTercero	<ol style="list-style-type: none"> 1. Propiedad Vehiculo 2. Nombre Propietario Vehiculo 3. Tipo Vehiculo 4. Placa Vehiculo
HI_PropiedadVehiculo	<ol style="list-style-type: none"> 1. Propiedad Vehiculo 2. Tipo Vehiculo 3. Placa Vehiculo

Tabla 43. Esquema de la Dimensión Vehículo

Gráfico de Dependencias

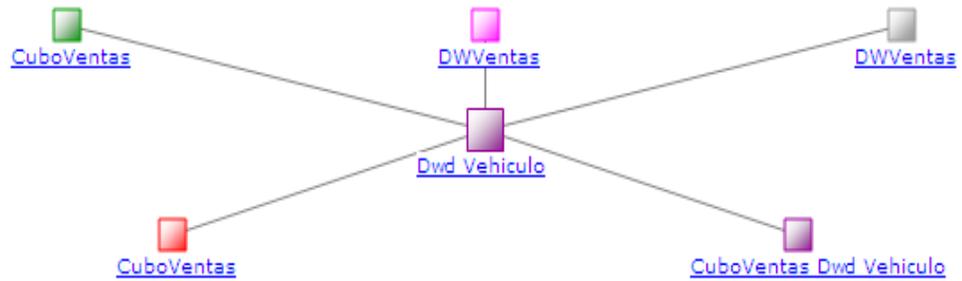


Figura 46. Gráfico de Dependencias de la Dimensión Dwd Vehículo

Dimensión Dwd Unidad Cargue

Atributos

Nombre	Uso	Jerarquías
Unidad Cargue		
Unidad Cargue Id	Key	

Tabla 44. Esquema de la Dimensión Unidad Cargue

Gráfico de Dependencias

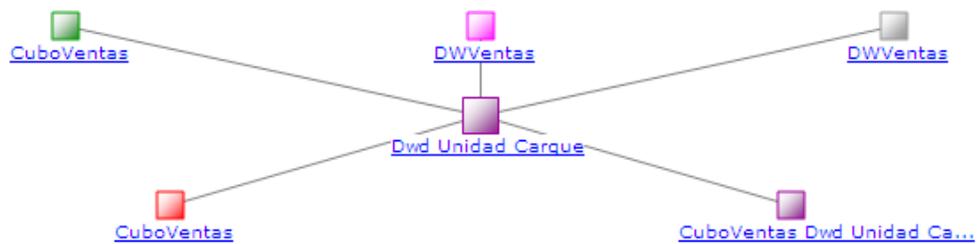


Figura 47. Gráfico de Dependencias de la Dimensión Dwd Unidad Cargue

Cubo OLAP

La unidad básica de almacenamiento en Analysis Services es el Cubo. Esta colección de datos agregados permiten que las consultas (Queries) retornen información de manera ágil y rápida.

Los cubos son ordenados en dimensiones y medidas. Las dimensiones en el cubo vienen de las tablas de dimensión, mientras que las medidas vienen de las tablas de hecho de la base de datos, algunas veces llamada, base de datos de esquema estrella.

Cubo: CuboVentas.cube

Propiedades del Cubo

Nombre	Valor
Nombre	CuboVentas
Descripción	Estructura analítica multidimensional que maneja la información histórica de las ventas.
Vista del Origen de Datos	DWVentas

Tabla 45. Propiedades del Cubo

Dimensiones

Dimensión del Cubo	Dimensión Base Datos	Alias
CuboVentas Agencia Despacho	Dwd Agencia	Agencia Despacho
CuboVentas Agencia Venta	Dwd Agencia	Agencia Venta
CuboVentas Ciudad Destino	Dwd Ciudad	Ciudad Destino
CuboVentas Ciudad Origen	Dwd Ciudad	Ciudad Origen

Dimensión del Cubo	Dimensión Base Datos	Alias
CuboVentas Dwd Cliente	Dwd Cliente	Dwd Cliente
CuboVentas Dwd Fecha	Dwd Fecha	Dwd Fecha
CuboVentas Dwd Tipo Transporte	Dwd Transporte Tipo	Dwd Transporte Tipo
CuboVentas Dwd Trailer	Dwd Trailer	Dwd Trailer
CuboVentas Dwd Unidad Cargue	Dwd Unidad Cargue	Dwd Unidad Cargue
CuboVentas Dwd Vehiculo	Dwd Vehiculo	Dwd Vehiculo

Tabla 46. Dimensiones del Cubo OLAP

Grupo de Medidas

Nombre	Estado	Modo Procesamiento	Modo almacenamiento
Dwh Ventas	Procesado	Regular	MOLAP

Tabla 47. Grupo de Medidas

Medidas de Dwh Ventas

Nombre	Tipo Dato	Fuente	Función Agregada
Kilometros Ruta	Integer	dbo_dwh_Ventas.KilometrosRuta	Sum
Flete Pagado	Double	dbo_dwh_Ventas.FletePagado	Sum
Flete Cobrado	Double	dbo_dwh_Ventas.FleteCobrado	Sum

Nombre	Tipo Dato	Fuente	Función Agregada
Cantidad	Integer	dbo_dwh_Ventas.Cantidad	Sum
Peso Pagado	Double	dbo_dwh_Ventas.PesoPagado	Sum
Peso Cobrado	Double	dbo_dwh_Ventas.PesoCobrado	Sum
Peso Real	Double	dbo_dwh_Ventas.PesoReal	Sum
Acompanante Pagado	Double	dbo_dwh_Ventas.AcompanantePagado	Sum
Acompanante Cobrado	Double	dbo_dwh_Ventas.AcompananteCobrado	Sum
Intermediacion	Double	dbo_dwh_Ventas.Intermediacion	Sum
Margen Intermediacion	Double	dbo_dwh_Ventas.MargenIntermediacion	Sum
Dwh Ventas Count	Integer	dbo_dwh_Ventas.	Count

Tabla 48. Medidas del Cubo OLAP

Gráfico de Dependencias

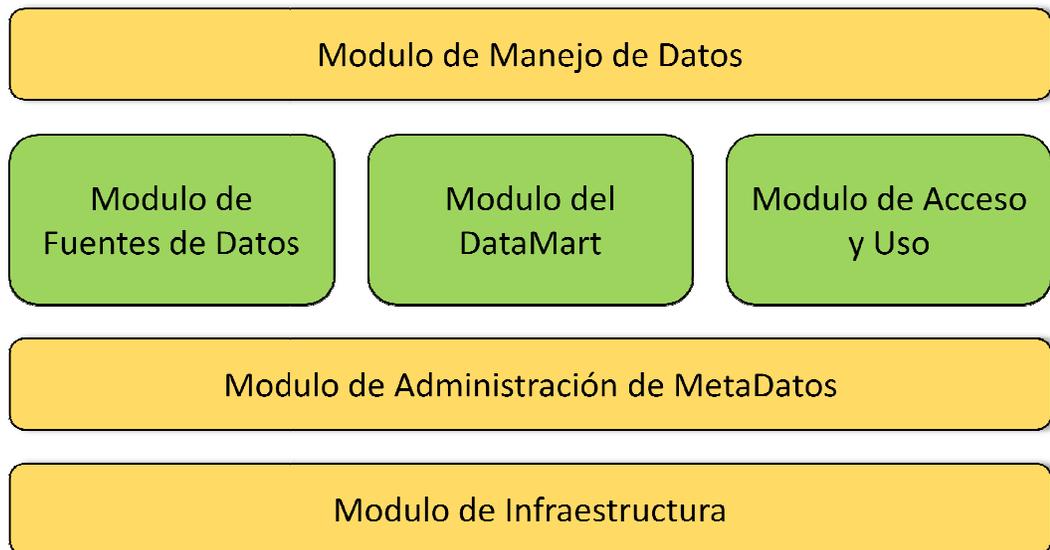


Figura 49. Escenario para Pruebas de Integración

Debido a que los módulos de Manejo de Datos, Fuentes de Datos y del DataMart se desarrollaron e implementaron bajo las herramientas de Microsoft SQL Server 2008 R2; esto garantiza la integración entre ellos, dado que pertenecen a la misma plataforma tecnológica.

La integración entre el módulo de Infraestructura, los tres módulos mencionados previamente y el módulo de Acceso y Uso, queda asegurada debido a que comparten la misma plataforma tecnológica y el módulo de Acceso y Uso que se realizara bajo la herramienta Microsoft Excel Versión 2007 o 2010, tiene una infraestructura diseñada específicamente para interactuar con Microsoft SQL Server 2008 R2.

El módulo de Administración de Metadatos también pasa las pruebas de integración debido a que se opta por el uso del Repositorio propio de Microsoft SQL Server 2008 R2.

4.7.2. Pruebas Funcionales

Estas pruebas con sus respectivos reportes de resultados, se especifican en el ANEXO 2; donde se abordan los requerimientos funcionales para el DataMart solicitados por el usuario y los cuales determinan los criterios de aceptación del proyecto.

4.8. PLAN DE CAPACITACION

4.8.1. Objetivo General

Proponer a la organización TDM transportes un plan de capacitación sobre el manejo de Tablas Dinámicas en Excel que le proporcione a los usuarios, los conocimientos necesarios para interpretar, analizar y administrar los datos con facilidad, reconocer tendencias importantes en los datos y tomar acertadas decisiones sobre datos críticos de la empresa.

4.8.2. Objetivos Específicos

- Analizar y resumir grandes bases de datos utilizando tablas dinámicas.
- Realizar cálculos en las tablas dinámicas con fines analíticos y de presentación.
- Aprender a graficar con tablas dinámicas para mejorar la visualización de los datos y poder realizar proyecciones y tomar decisiones empresariales.

4.8.3. Población Objetivo

El contenido de esta capacitación será dirigido principalmente al personal de Marketing y soluciones logísticas de la organización.

4.8.4. Contenido

Modulo I. Conceptos básicos

1. ¿Qué es una tabla Dinámica?

2. Aspectos importantes a considerar antes de crear una tabla dinámica.

Módulo II. Crear un informe de tabla dinámica

Módulo III. Modificar el diseño de la tabla dinámica

1. Cambiar la posición de las variables.
2. Modificar la posición de los datos.
3. Evaluar otras opciones de diseño.

Módulo IV. Diseño y formato

1. Aplicar Totales y filtros.
2. Cambiar el Diseño de la Tabla.
3. Formatos en la tabla dinámica.

Módulo V. Cómo realizar cálculos en una tabla dinámica

1. Obtener promedios, máximos, mínimos, entre otros.
2. Calcular porcentajes.
3. Colocar subtotales.
4. Crear y configurar nuevos campos.
5. Elementos calculados y Aplicación de fórmulas en tablas dinámicas.

Módulo VI. Filtros y Subtablas

1. Filtros.
2. SubTablas.

Módulo VII. Agrupar datos

1. Agrupar fechas.

2. Datos numéricos.
3. Errores al agrupar fechas.
4. Vínculos entre las tablas dinámicas.

Módulo VIII. Gráficos dinámicos

1. Opciones de gráficos dinámicos.
2. Generación de gráfico a partir de una tabla dinámica.
3. Actualización de gráfico.
4. Líneas de tendencia.

Módulo IX. Trabajar con datos externos

1. Fuentes de datos.
2. Importar datos.
3. Filtrar datos importados.
4. Diseñar tablas dinámicas desde el origen.
5. Caso de estudio: Análisis y Explotación del Cubo OLAP de Ventas a través de una tabla dinámica.

Modalidad de la Propuesta

La capacitación debe incluir actividades coordinadas por los capacitadores quienes deben estar en contacto con el personal en entrenamiento, intercambiando con ellos materiales didácticos, actividades en la herramienta de trabajo, consultas y otros aspectos relevantes que fortalezcan el desarrollo del programa.

Actividades Propuestas

- **Interacción en línea:** Se deben utilizar herramientas de comunicación como correo electrónico donde el personal pueda realizar consultas y/o intercambiar ideas relacionadas con la temática planteada. Así mismo, hacer uso de la mesa de ayuda con la que cuenta la organización para solicitar servicios de soporte y obtener solución al requerimiento oportunamente.
- **Seguimiento al personal en entrenamiento:** El seguimiento por parte de los capacitadores debe ser permanente, con el fin de acompañar y sostener el éxito del proceso en la formación tecnológica.

Material de Estudio

El estudio que propone la modalidad, se debe llevar a cabo por medio de un material didáctico organizado por módulos, que reúna los contenidos y actividades, en forma articulada, para garantizar la calidad de la capacitación. Este material debe constituir una herramienta de trabajo indispensable para el desarrollo de cada uno de los módulos y permitir estructurar los contenidos desde una propuesta teórica – practica.

Infraestructura

Disponer de una sala de estudio que reúna las condiciones necesarias para el adecuado desarrollo del programa de capacitación.

- **Hardware**

Disponer de mínimo de 5 computadores en Red.

Conexión a Internet

- **Software**

Microsoft Office Excel versión 2007 o 2010

Microsoft Office Outlook versión 2007 o 2010.

- **Evaluación**

El programa de capacitación debe ser evaluado mediante la realización de ejercicios o trabajos prácticos. Así mismo, es importante llevar a cabo un registro de asistencia por parte del personal a la capacitación.

5. RESULTADOS Y CONCLUSIONES

5.1. RESULTADOS

Los resultados obtenidos durante el desarrollo de este trabajo de grado se muestran a continuación:

1. Con los resultados obtenidos del análisis de requerimientos se pudo determinar el mejor modelo de datos que se adaptará a las necesidades del proyecto.
2. Se identificó que solo existía una fuente de datos desde la cual se analizaría y se extraería la información de las ventas históricas de la organización.
3. Como resultado de la búsqueda de una plataforma que se adaptara a las especificaciones del proyecto, se seleccionó SQL Server Business Intelligence 2008 como herramienta de diseño y construcción del DataMart por su amplia gama de funcionalidades y servicios para administrar la base de datos, para realizar los procesos de extracción, transformación y carga de los datos y para la construcción y despliegue del cubo de información.
4. Se diseñó e implementó el DataMart de acuerdo a las necesidades del proyecto con la herramienta de inteligencia de negocios previamente definida.
5. Se determinó que no era necesario realizarle modificaciones a la base de datos transaccional SAT.
6. Se seleccionaron los indicadores o variables que se requieren para satisfacer las necesidades de análisis de datos.
7. Al analizar los datos históricos de las transacciones en la base de datos operacional se pudo observar que los mismos son variables y no presentan una tendencia significativa.
8. Para realizar análisis de los datos por parte del usuario final, se definió el uso de la herramienta de Excel – tablas dinámicas como sistema de reportes que ayude a la planificación y al proceso de toma de decisiones.

5.2. CONCLUSIONES

1. Las herramientas analíticas OLAP son parte importante en las organizaciones dado que permiten evaluar y monitorear la situación real de una empresa desde una perspectiva que facilita la exploración de los diversos escenarios de una manera sencilla. Adicionalmente, proporciona una visión más profunda de todos los niveles de una organización.
2. Los modelos multidimensionales permiten ver las medidas del negocio desde diferentes perspectivas y ayudan a identificar fácilmente patrones y tendencias.
3. Para que este proyecto de DataWareHousing tenga éxito es necesario que exista una cultura de información en la organización. Si las decisiones son tomadas basadas en intuiciones y suposiciones y no se apalancan en un modelo analítico de información, ningún proyecto de esta característica cumplirá su objetivo en la organización.
4. La realización exitosa del proceso de implantación de una solución de DataWareHouse debe contar fuertemente con el patrocinio al interior de la organización. Si no se cuenta con patrocinadores en el área del negocio y en el área técnica es preferible no arriesgar y posponer el proyecto.
5. Es esencial el manejo de roles en el equipo de desarrollo de un proyecto de DataWareHouse, de esta manera hay fluidez y especialización en los miembros del equipo que implica una mayor eficiencia y calidad del trabajo.
6. Uno de los procesos con mayor dificultad en un proyecto de DataMart es la identificación de los requerimientos del usuario y el análisis de los sistemas fuente. En las organizaciones es muy común la falta de documentación de los sistemas de información y el volumen de inconsistencias en los datos históricos tiende a ser alto. Es de anotar que para la empresa TDM transportes el porcentaje de inconsistencias en sus datos fue realmente pequeño, debido a la estructura de datos y controles de validación en el sistema operacional SAT están bien definidos y diseñados.

7. La metodología de desarrollo utilizada para la realización de este proyecto, aborda de manera detallada todas las etapas y fortalecen el proceso de implementación de la solución DataMart.
8. Una adecuada identificación y comprensión de patrones históricos en los datos permiten al usuario realizar planificaciones basadas en condiciones futuras que rodean a una situación particular.
9. Es importante para el usuario final que el DataMart se convierta en una herramienta que permita visualizar de manera sencilla, ágil y clara los datos de la organización dado que éstos forman parte importante para la toma de decisiones.
10. La estructura de una base de datos OLAP es fundamental para el éxito de un proyecto de este tipo. Aunque la estructura ER (Entidad – Relación) funciona correctamente para almacenar datos operativos, es totalmente impropia e inadecuada para estas soluciones. El modelo utilizado refleja las necesidades del negocio en términos de soporte a la toma de decisiones.
11. Este proyecto tiene el objetivo de mostrar a la organización las bondades que presenta una herramienta como el DataMart para realizar análisis de la información y se espera que se convierta en corto tiempo en una herramienta de uso crítico y abra las puertas para el desarrollo de posteriores proyectos al interior de la organización.

6. BIBLIOGRAFIA

[1] FLORES QUIROZ, Fair, et al. DATAMART CAPACITACION (Seguridad Nacional). Facultad de Ingenieria y Arquitectura. Universidad de San Martin de Porres. Peru. 2009.

[2] FERNANDEZ OCHOA, Eddy. Análisis, diseño e implementación de un Datamart de Clientes para el Area de Marketing de una Entidad Aseguradora. Facultad de Ciencias e Ingenieria. Especialidad de Ingenieria Informatica. Pontificia Universidad Catolica del Peru. Lima (Peru). 2009.

[3] RONALDLEE, Ejalu. A Web Based Data Mart System For Patient Care. Department of Information Systems. Faculty of Computing and Information Technology. Department of Information Systems. Makerere University. Uganda – Africa. 2006

[4] KIMBALL, Ralph, et al. The Data Warehouse Lifecycle Toolkit. Wiley. Segunda Edicion en Ingles. Estados Unidos. 672 paginas. 2010.

[5] KIMBALL, Ralph. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition). Wiley. 464 Paginas. 2002.

[6] THOMSEN, EriK, et al. Microsoft OLAP Solutions (First Edtion). Wiley. 497 Paginas. 1999.

[7] INMON, W.H. Building the Data Warehouse (Third Edtion). Wiley. 356 Paginas. 2002.

Paginas Web

VIKLUND, Andreas. Blog de WordPress.com.

<http://churriwifi.wordpress.com/category/business-intelligence/>

ACCELEBRATE. SSAS Tutorial: SQL Server 2008 Analysis Services Tutorial.

http://www.accelebrate.com/sql_training/ssas_2008_tutorial.htm

ACCELEBRATE. SSAS Tutorial: SQL Server 2008 Integration Services Tutorial.

http://www.accelebrate.com/sql_training/ssis_2008_tutorial.htm

Edgewood Solutions, LLC. Using XML Package Configurations with Integration Services SSIS.

<http://www.mssqltips.com/sqlservertip/1434/using-xml-package-configurations-with-integration-services-ssis/>

Soe Tun. Execute SSIS Package from Stored Procedure with Parameters using DTEXEC Utility.

<http://geekswithblogs.net/stun/archive/2010/02/24/execute-ssis-package-from-stored-procedure-with-parameters-using-dtexec-utility.aspx>