



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
DEPARTAMENTO CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE LA
INFORMACIÓN
CHILLÁN

Propuesta de implementación de un Data Warehouse para el área de Soporte de Información, Rabie S.A.

Memoria para optar al Título de Ingeniero Civil en
Informática

Alumno : Felipe Andrés Orellana Sánchez

Profesor Guía : Sr. Gilberto Gutiérrez

CHILLÁN, 2013

Agradecimientos

Ya culmina una etapa en la cual obtuve valiosos conocimientos y habilidades que me ayudarán a desenvolverse en la vida laboral con suficientes fortalezas para enfrentar nuevos retos.

Por ende agradezco a la Universidad del Bío-Bío, gran institución que me dió la oportunidad para adquirir los conocimientos, los que me permitieron desarrollar este proyecto de título.

Asimismo, a los profesores que me brindaron su sabiduría en varios campos del conocimiento, en especial a mi profesor guía Sr. Gilberto Gutiérrez, a las profesoras Srta. María Soto Chico, sede Chillán y Sta. Mónica Caniupan sede Concepción, quienes me entregaron su valioso tiempo y apoyo en la confección de mi memoria de título.

Además, agradezco al personal del área de Soporte de Información de Rabie S.A. por haber confiado en mí para llevar a cabo este proyecto.

Agradezco principalmente a mi madre Elba Sánchez por su apoyo incondicional durante todas las etapas de mi vida y por su motivación en los momentos más duros. También a mi padre y familia por su constante presencia en todo éste periodo.

Finalmente quiero dar gracias a mi pareja, Yenny, por estar a mi lado en los momentos más importantes de mi carrera, sin mencionar su comprensión y entendimiento, entregándome así, sentimientos de alegría, amor y cariño.

Resumen

Las compañías u organizaciones han entendido la importancia de apoyar sus estrategias de negocios por medio de Tecnologías de la Información.

Es por ello que los actuales sistemas de información han evolucionado hacia los sistemas de gestión, apoyados por el Business Intelligence o Inteligencia de Negocios permitiendo, de esta manera, poder realizar tomas de decisión de carácter estratégico, logrando así, disminuir los riesgos que causaban estas decisiones hace algún tiempo atrás.

Una de estas Tecnologías de Información son los sistemas de Data Warehouse y que corresponden a colecciones de datos orientadas a un determinado ámbito; integrando información desde diversas fuentes, otorgan la funcionalidad de ayudar a la toma de decisiones en la entidad en que se implementa.

El objetivo del proyecto es entregar una propuesta de implementación de un sistema Data Warehouse, que preste apoyo a la labor del área de Soporte de Información de Rabie S.A., área en la que actualmente la información no se explota adecuadamente, debido a la sobrecarga de datos, lo cual dificulta generar la información relevante y oportuna que los directivos requieren para apoyar su proceso de toma de decisiones.

Este sistema incorporará los datos ya existentes en dicha área, conformando un compilado de información relevante, lo que generará a través de procesos de extracción, transformación y carga de los datos, un ambiente amigable, eficiente y eficaz para la entrega de información requerida, prestando de esta manera una útil herramienta al área de Soporte de Información, la cual en la actualidad no cuenta con esta operatividad.

...no podemos resolver problemas usando el mismo tipo de pensamiento que usamos cuando los creamos...

Albert Einstein

Tabla de Contenidos

Capítulo 1	12
Introducción.....	12
1.1 Objetivo del Proyecto	13
1.2 Objetivos Específicos del Proyecto	14
1.3 Alcance y Límites del Proyecto	15
Capítulo 2	17
Descripción de la Empresa	17
2.1 Identificación de la Empresa.....	17
2.2 Misión y Visión de Rabie S.A.	18
2.2.1 Misión.....	18
2.2.2 Visión	18
2.3 Área de Soporte de Información, Rabie S.A.....	19
Capítulo 3	21
Marco Teórico	21
3.1 Business Intelligence	21
3.1.1 Introducción.....	21
3.1.2 Definición	22
3.1.3 Proceso de Business Intelligence.....	23
3.1.4 Beneficios	24
3.2 DataMart	25
3.3 Data Warehouse	26
3.3.1 Introducción.....	26
3.3.2 Definición	26
3.3.3 Características.....	27
3.3.4 Cualidades	33
3.3.5 Objetivos generales.....	34
3.3.6 Ventajas	36
3.3.7 Desventajas.....	37
3.3.8 Redundancia	38

3.3.9	Estructura.....	39
3.3.10	Flujo de Datos.....	42
3.4	Metodologías de desarrollo de un Data Warehouse	43
3.4.1	Modelo top-down	43
3.4.2	Modelo Bottom-up	50
3.4.3	Comparativa de resumen	60
Capítulo 4	63
Entorno de la Aplicación	63
4.1	Situación actual.....	63
4.1.1	Área de la empresa a abordar	65
4.2	Forma de abordar el problema.....	67
4.3	Objetivo estratégico	67
Capítulo 5	68
Diseño de un Data Warehouse.....	68
5.1	Análisis de Requerimientos	68
5.1.2	Identificación de Preguntas	73
5.1.3	Identificación de indicadores y dimensiones o perspectivas de análisis	75
5.1.4	Diagrama Conceptual	77
5.2	Análisis de los procesos OLTP	78
5.2.1	Determinación de Indicadores	79
5.2.2	Definición de relaciones	81
5.2.3	Nivel de Granularidad	90
5.2.4	Diagrama conceptual ampliado	94
5.3	Modelo lógico del Data Warehouse.....	95
5.3.1	Tipo de Modelo Lógico	95
5.3.2	Tablas de Dimensiones.....	96
5.3.3	Tabla de Hechos	103
5.3.4	Uniones.....	104
5.4	Procesos ETL.....	105
5.4.1	Tabla de dimensión “Productos”	106
5.4.2	Tabla de dimensión “Vendedor”	108

5.4.3	Tabla de dimensión “Lugar Compra”	109
5.4.4	Tabla de dimensión “Sector Venta”	112
5.4.5	Tabla de dimensión “Tiempo”	113
5.4.6	Tabla de hechos “Venta”	116
5.5	Creación de Cubos Multidimensionales	122
5.5.1	Cubo de Ventas 1 (Producto-Fecha-Sector_Venta).....	122
5.5.2	Cubo de Ventas 2 (Vendedor-Fecha-Sector_Venta)	130
5.5.3	Cubo de Ventas 3 (Producto-Fecha-Sector_Venta).....	136
Capítulo 6	143
Tipos de herramientas de consulta y análisis..... 143		
6.1	Introducción	143
6.2	Consulta y Reportes	144
6.2.1	Crystal Reports	145
6.3	Herramientas OLAP	146
6.3.1	OlapCube	148
6.3.2	Mondrian	163
6.4	Data Mining	164
6.4.1	Orange	165
6.4.2	RapidMiner.....	166
6.4.3	Weka.....	168
6.4.4	jHepWork	169
6.4.5	KNIME	170
Capítulo 7	172
Conclusiones..... 172		
7.1	Introducción	172
7.2	Conclusiones por objetivos planteados.....	172
8.	Bibliografía.....	175

Índice de Figuras

Figura 1: Arquitectura del proyecto Data Warehouse [1]	16
Figura 2: Equipo de Trabajo Soporte de Información	20
Figura 3: Fases del proceso Business Intelligence [1].....	23
Figura 4: Data Warehouse Orientado al Negocio [14]	29
Figura 5: Data Warehouse Integrado [14]	30
Figura 6: Data Warehouse Histórico [14].....	31
Figura 7: Data Warehouse No Volátil [1]	32
Figura 8: Estructura de un Data Warehouse [1]	39
Figura 9: Flujo de Datos de un Data Warehouse [1]	42
Figura 10: Metodología HEFESTO.....	45
Figura 11: Metodología propuesta por Bill Inmon	48
Figura 12: Ejemplo de Metodologías Convencionales.....	50
Figura 13: Ejemplo Metodología de Inmon	50
Figura 14: Rapid Warehousing Methodology [17].....	51
Figura 15: Metodología Kimball – Ciclo de Vida [18]	54
Figura 16: Organigrama Rabie S.A.	64
Figura 17: Mapa Conceptual	78
Figura 18: Diagrama Entidad Relación	88
Figura 19: Relación entre el Modelo Relacional y el Diagrama propuesto.....	89
Figura 20: Diagrama Conceptual Ampliado.....	94
Figura 21: Tabla de dimensión "PRODUCTO"	97
Figura 22: Esquema de Jerarquía dimensión "PRODUCTO"	97
Figura 23: Tabla de dimensión "VENDEDOR"	98
Figura 24: Esquema de Jerarquía dimensión "VENDEDOR"	99
Figura 25: Tabla de dimensión "LUGAR_COMPRA"	99
Figura 26: Esquema de Jerarquía dimensión "LUGAR_COMPRA"	100
Figura 27: Tabla de dimensión "SECTOR_VENTA"	101
Figura 28: Esquema de Jerarquía dimensión "SECTOR_VENTA"	101
Figura 29: Tabla de dimensión "FECHA"	102
Figura 30: Esquema de Jerarquía dimensión "FECHA"	103
Figura 31: Tabla de Hechos "VENTA"	104
Figura 32: Modelo tipo Estrella del Data Warehouse	105
Figura 33: Caso práctico, datos de dimensión "Fecha"	115
Figura 34: Modelo lógico Cubo de Ventas 1.....	123
Figura 35: “PRODUCTO”, relación Padre-Hijo	127

Figura 36: “SECTOR_VENTA”, relación Padre-Hijo.....	127
Figura 37: “FECHA”, relación Padre-Hijo.....	128
Figura 38: Modelo lógico Cubo de Ventas 2.....	130
Figura 39: “VENDEDOR”, relación Padre-Hijo.....	134
Figura 40: “SECTOR_VENTA”, relación Padre-Hijo (2).....	134
Figura 41: “FECHA”, relación Padre-Hijo (2).....	135
Figura 42: Modelo lógico Cubo de Ventas 3.....	137
Figura 43: “PRODUCTO”, relación Padre-Hijo	140
Figura 44: “LUGAR_COMPRA”, relación Padre-Hijo.....	141
Figura 45: “FECHA”, relación Padre-Hijo (3).....	141
Figura 46: Pantalla de Bienvenida del Instalador de OlapCube.....	150
Figura 47: Pantalla de progreso y finalización de instalación.....	150
Figura 48: Pantalla de Inicio.....	151
Figura 49: Pantalla de Selección del origen de datos	151
Figura 50: Pantalla de Selección del origen de Datos (2).....	152
Figura 51: Pantalla de Selección del origen de Datos (3).....	152
Figura 52: Pantalla de Selección del origen de Datos (4).....	153
Figura 53: Pantalla de OLAP conectado con el Data Warehouse	153
Figura 54: Ventana de Características de la Dimensión y Atributos.....	156
Figura 55: Pantalla de Creación del Cubo.....	158
Figura 56: Pantalla de Análisis Dimensión FECHA	159
Figura 57: Pantalla de Análisis Dimensión LUGAR_COMPRA.....	159
Figura 58: Pantalla de Análisis Dimensión PRODUCTOS	160
Figura 59: Pantalla de exportación del cubo a Excel.....	160

Índice de Tablas

Tabla 1: Comparativa de Métodos y Metodologías.....	62
Tabla 2: Tabla “Maestra_Final_Diaria”	83
Tabla 3: Tabla “Maestra_Final_Mensual”	84
Tabla 4: Tabla "Detalle_Productos"	86
Tabla 5: Cantidad de Datos y Registros actuales	87
Tabla 6: Indicadores del “Cubo de Ventas 1”	124
Tabla 7: Atributos del “Cubo de Ventas 1”	125
Tabla 8: Indicadores del “Cubo de Ventas 2”	131
Tabla 9: Atributos del “Cubo de Ventas 2”	132
Tabla 10: Indicadores del “Cubo de Ventas 3”	138
Tabla 11: Atributos del “Cubo de Ventas 3”	139

Índice de Gráficos

Gráfico 1: Unidades Vendidas Abarrotes-Margarinas Agosto-Octubre 2012	162
Gráfico 2: Monto Total de Ventas Tabaquería Septiembre 2012.....	162
Gráfico 3: Ganancia de los Gerentes generada el primer día de cada mes por Canal 2	163

Capítulo 1

Introducción

Durante el transcurso del tiempo la informática ha incrementado el poder con el que se desenvuelve dentro de las organizaciones bien consolidadas, por lo que la incorporación de las nuevas Tecnologías de Información (TIs) es cada vez más apreciada por estas mismas.

Así mismo, las organizaciones a través de su participación activa en el mundo de las TIs, han ido evolucionando su productividad, con nuevos sistemas operacionales, permitiendo a los usuarios de esta forma realizar labores en periodos de tiempo más cortos e incluso en paralelo, obteniendo así mayor eficiencia en los procesos operacionales de una organización.

Debido a esto, las grandes organizaciones se han visto obligadas a mantener la información sobre sus operaciones, la cual ha sido respaldada en bases de datos u otras fuentes poco funcionales para los requerimientos actuales de una compañía.

Con el objetivo de ayudar en la toma de decisiones, se han ido incorporando bases de datos más complejas, que a su vez entreguen, mediante un análisis de datos, información de mayor valor para evaluar decisiones estratégicas y pronósticos sobre comportamientos futuros de una compañía. En atención a la necesidad planteada anteriormente surge la funcionalidad ofrecida por el sistema Data Warehouse, ya que reuniendo información de distintas fuentes, puede dar solución a interrogantes elaboradas con más complejidad y así satisfacer requerimientos más específicos de los potenciales usuarios de una compañía.

A continuación, se ahondará en cómo implementar este sistema en el área de Soporte de Información de la empresa Rabie S.A., señalando a su vez las variables que involucran su implementación.

1.1 Objetivo del Proyecto

El proyecto está enfocado a prestar apoyo a la empresa Rabie S.A., la cual cubre desde Arica a Chiloé y posee tres grandes Centros de Distribución ubicados en Chillán, Santiago y Antofagasta, equipados con la más alta tecnología para el almacenamiento, manejo y despacho de productos. Cabe destacar que la Administración Central de esta compañía, cuenta con modernas oficinas, ubicadas en el centro de la ciudad de Chillán.

El Centro de Distribución de Antofagasta cubre desde la XV región de Arica y Parinacota hasta la II región de Antofagasta, llegando de esta manera a unos 7.000 clientes, los que son atendidos por una fuerza de venta compuesta por 65 vendedores. El mix de productos está compuesto por unos 4.300 artículos, los que son despachados a los clientes, a más tardar dentro de 48 horas desde el momento de la compra.

El Centro de Distribución de Santiago posee una cobertura que se extiende desde la III hasta la VI región, incluyendo la región Metropolitana, llegando a más de 20.000 clientes, los que son atendidos por una fuerza de ventas superior a los 300 vendedores. Este centro ofrece a sus clientes, un mix de unos 6.000 artículos.

Por último, el Centro de Distribución Chillán, cubre una zona que se extiende entre la VII y X regiones, incluyendo la Isla Grande de Chiloé. Dicho centro abastece a más de 18.000 clientes, quienes son atendidos por alrededor de 230 vendedores, con un mix de productos compuesto por más de 5.000 artículos. Cabe destacar que en este Centro de Distribución se realiza aproximadamente un total de 50.000 ventas al día.

En la empresa podemos encontrar las áreas de Gestión Comercial, Gestión Logística, Gestión Ventas y Presupuesto, las cuales son abastecidas de información

directamente por el área de Soporte de Información, por lo que esta propuesta dará a esta última un mayor realce.

El área de Soporte de Información actualmente almacena la información, a través de una base de datos independiente, que se encuentra conformada por tablas de hechos, que contienen datos extraídos de fuentes externas al área, por lo que parte de éstos, son procesados para su almacenamiento. Las tablas se encuentran organizadas cronológicamente, en otras palabras de forma mensual, por lo que cada tabla contiene un exceso de atributos, además de un gran volumen de registros, los cuales se incrementan diariamente. Actualmente estas tablas contienen sobre un millón de registros, con una aproximación 100 GB procesados mensualmente.

A consecuencia de lo anterior, la información no se explota adecuadamente, debido a la sobrecarga de datos, lo cual dificulta generar la información relevante y oportuna que los directivos requieren para apoyar su proceso de toma de decisiones.

Con el siguiente proyecto se quiere dar solución al problema planteado anteriormente, proponiendo la implementación de un sistema Data Warehouse para apoyar la labor del área de Soporte de Información de la empresa Rabie S.A.

Este sistema incorporará los datos ya existentes en dicha área, conformando un compilado de información relevante, lo que se generará a través de procesos de extracción, transformación y carga de los datos, un ambiente amigable, eficiente y eficaz para la entrega de información requerida, prestando de esta manera una útil herramienta al área de Soporte de Información, la cual en la actualidad no cuenta con esta operatividad.

1.2 Objetivos Específicos del Proyecto

Los objetivos específicos son:

- Definir el marco teórico de un Data Warehouse, con objeto de reconocer el valor de esta tecnología y las posibilidades que ofrece a las organizaciones.

- Comparar las metodologías más utilizadas en la implementación de un Data Warehouse, con el propósito de que la organización pueda identificar la que más se ajusta a sus necesidades.
- Identificar los datos necesarios para conformar el Data Warehouse con sus respectivas fuentes (componente OLTP, On Line Transaction Processing)
- Establecer los procesos aplicados a los datos para manipularlos, integrarlos y transformarlos, de modo que, posteriormente, sea posible cargar los resultados obtenidos en el Data Warehouse (componente Load Manager)
- Proponer el modelo de la base de datos, al área de Soporte de Información, basándose en los tipos de modelado de un Data Warehouse (componente Data Warehouse Manager).

1.3 Alcance y Límites del Proyecto

En este proyecto se emplearán los pasos justificados que permiten desarrollar e implementar un sistema Data Warehouse, entregando de esta forma los procesos a considerar en el desarrollo de dicho sistema. Los datos relevantes estarán almacenados históricamente en sus respectivas tablas, y estas mismas contarán con la totalidad de los datos, no estando éstos repartidos en distintas tablas como ocurre actualmente. Lo propuesto facilitará su procesamiento, pudiendo de esta manera, obtener información valiosa de manera ágil. Además, permitirá a la organización, si lo desea, modificar su sistema de gestión de base de datos, apuntando al uso de tecnologías como Minería de Datos, entre otras.

Debido al tiempo limitado con que se cuenta para desarrollar este proyecto, se centrará en las etapas de diseño del Data Warehouse y las acciones de ETL.

A continuación se muestra la Figura 1 en la cual se encuentra demarcado en un recuadro, los componentes a abarcar en el proyecto, según la metodología Hefesto [1] propuesta para la construcción de un Data Warehouse:

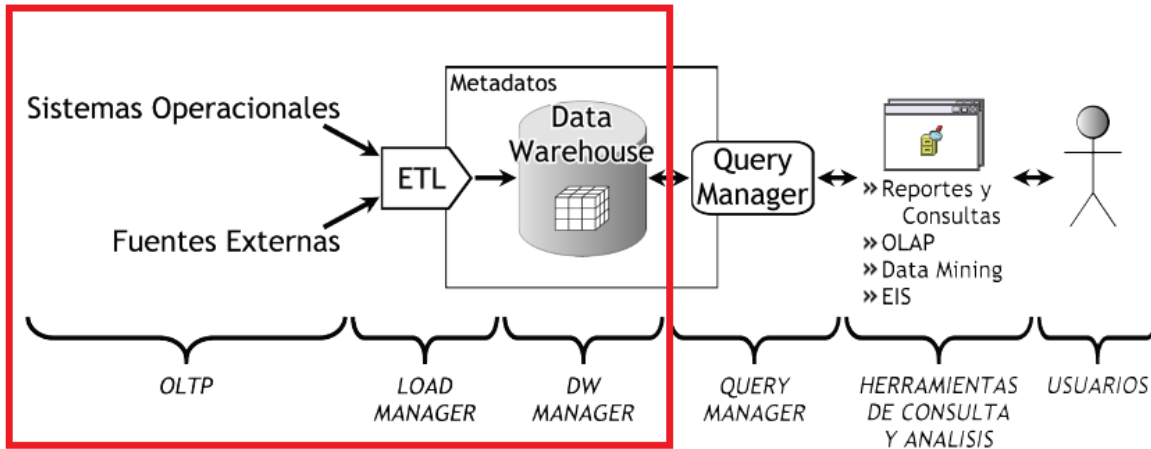


Figura 1: Arquitectura del proyecto Data Warehouse [1]

El resultado de este proyecto de título será la entrega de una comparación de las metodologías más utilizadas en la implementación de un Data Warehouse, para que la empresa pueda identificar la que más se ajuste a sus necesidades. Junto a esto se establecerán los procesos aplicados a los datos para manipularlos, integrarlos y transformarlos. Y, finalmente, se presentará un modelo de la base de datos que soporte tecnologías de procesamiento de datos, como lo son los data query y las herramientas de consulta y análisis de datos.

Capítulo 2

Descripción de la Empresa

En este capítulo se dará a conocer la empresa Rabié S.A., y sus respectivas áreas, indicando parte de su historia, su posición en el mercado, además de sus actividades. También se dará a conocer las distintas sucursales que posee a lo largo del país, entregando datos cuantitativos de sus empleados, proveedores, clientes entre otros.

2.1 Identificación de la Empresa

Rabié ha llegado a ser la empresa más grande, competitiva e innovadora de la industria a través de una historia de más de 100 años de esfuerzo y superación.

En 1925, el fundador estableció un sistema de distribución a puntos claves del sur y norte del país. En los años 30, Casa Rabié era ya la empresa comercial más grande de Chillán y con alcance nacional.

En 1996 Rabié S.A. construye el Mall Plaza El Roble. El mismo año, don Jorge Rabié recibe el Premio Icare en la “Categoría Empresario”.

Es inaugurado en 1999 el nuevo Centro de Distribución Chillán, ubicado en la naciente comuna de Chillán Viejo.

La empresa levanta su Centro de Distribución Antofagasta en el año 2002, para abastecer desde allí a la Primera Región. Su cobertura se extiende así desde Arica a Chiloé [2].

Rabie S.A. tiene como su actividad principal la Distribución de Productos al Comercio Minorista y Mayorista de Arica a Puerto Montt, con una posición en el mercado nacional de líder en su giro.

De forma interna Rabie S.A. cuenta con un total de 1.200 empleados y posee sobre 6.000 productos a nivel nacional. Y externamente cuenta con 1.500 transportistas externos y 200 proveedores, además de un total aproximado de 45.000 clientes a lo largo del país.

2.2 Misión y Visión de Rabie S.A.

2.2.1 Misión

"Ser el nexo entre la Industria y el Comercio Minorista"

El objetivo más amplio de Distribuidora Rabié es ser la solución logística más conveniente del país, tanto para los fabricantes como para los comerciantes. Es la proyección e imagen ideal que comparten sus propietarios, ejecutivos y trabajadores [2].

2.2.2 Visión

"Ser la solución logística más conveniente"

El gran propósito de Distribuidora Rabié es permitir que los fabricantes coloquen sus productos en cualquier lugar de Chile, y que los comerciantes tengan un abastecimiento seguro, confiable y con posibilidades de competir sin importar el lugar geográfico en el que se encuentren [2].

2.3 Área de Soporte de Información, Rabie S.A.

La función del área de Soporte de Información es generar y validar toda la información que circula dentro de la compañía, esto significa que hay mucha información que es generada, validada y liberada al resto de la compañía, aunque también existe una cantidad de informes que no son generados por esta área, esto se debe a que dicha área no tiene acceso a cierta información o por otros inconvenientes, por lo que en ese caso la función del área es sólo validar la información recibida.

En cuanto a la información generada, se pueden identificar dos informes relevantes para el área que serán especificados en el Capítulo 5: Solicitudes (Reportes), estos son:

- IGC (Informe de Gestión Comercial)
- B2B para proveedores (Comercial, Financiera, Logística)

Cabe señalar que actualmente el área Control Gestión donde se ubica Soporte de Información, se encuentra ejerciendo la labor de un área de Inteligencia de Negocios o Business Intelligent, la cual da mayor importancia a los procesos de recolectar y utilizar efectivamente la información, con el fin de mejorar la forma de operar de una organización, brindando a sus usuarios, el acceso a la información clave que necesitan para llevar a cabo sus tareas habituales y más precisamente, para poder tomar decisiones oportunas.

El área de Soporte de Información cuenta con personal que permitirá el desarrollo e implementación de un Data Warehouse, conformándose por el equipo de trabajo representado en la Figura 2.

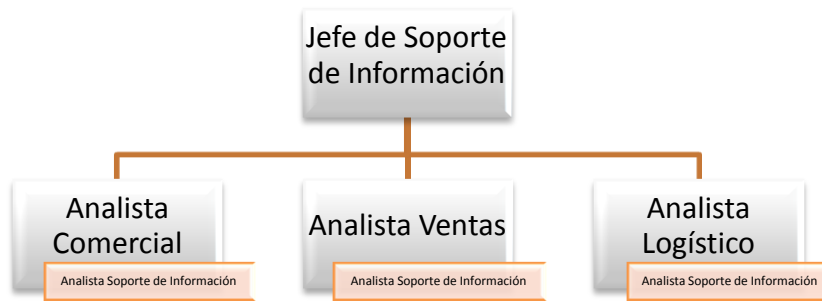


Figura 2: Equipo de Trabajo Soporte de Información

Este equipo de desarrollo es representado por el Jefe de Soporte de Información de la empresa Rabie S.A., el cual será el Jefe de Proyecto para una posterior implementación del sistema.

En conjunto a lo anterior, la empresa Rabie S.A. cuenta con dos tipos de clientes; está el cliente interno, que corresponde a las áreas de Control Gestión internas de la compañía, y por otro lado está el cliente externo que corresponde a los proveedores, transportistas y clientes de consumo. El cliente interno (Control Gestión) actualmente tiene acceso a la información por niveles, los cuales cubren los siguientes ámbitos:

- Informe Kpi
- Información operacional
- Información a pedido

El objetivo del área de Soporte de Información con respecto a la implementación de Data Warehouse es potenciar al cliente interno, de tal forma de optimizar las consultas, evitando la saturación del sistema, esto apunta al Control Gestión, ya que actualmente procesan alrededor de 100 GB de información diarios. También cabe destacar que el 85% de las consultas generadas corresponden directamente al sector de ventas.

Capítulo 3

Marco Teórico

En este capítulo se definirá un Data Warehouse indicando sus propiedades, características y su relación con los sistemas transaccionales. De la misma manera se abordará el tema de la integración y de las dificultades de su desarrollo. Además se tendrá una idea más clara con respecto a algunos conceptos relevantes a la hora de comprender un Data Warehouse como lo es Business Intelligence, indicando su relación e influencia en el desarrollo de este tipo de sistemas.

3.1 Business Intelligence

3.1.1 Introducción

En las bases de datos relacionales se almacenan y administran grandes cantidades de datos a través de sistemas transaccionales, los cuales son un tipo de sistema de información diseñado para recolectar, almacenar, modificar y recuperar todo tipo de información que es generada por las transacciones en una organización [3].

A consecuencia de los procesos involucrados en los sistemas transaccionales, se pretende obtener de los datos, información que enriquezca las decisiones de los usuarios, lo que apunta directamente a Business Intelligence, minimizando de esta forma el riesgo y la incertidumbre de las decisiones.

Lo que se espera de Business Intelligence, es que permita a las organizaciones traducir sus objetivos en indicadores de estudio para que así estos puedan ser analizados

desde diferentes perspectivas, haciendo posible de esta forma encontrar información que pueda responder a las preguntas no sólo del pasado y del presente, sino que también posibilite la construcción de modelos, para predecir eventos futuros [4].

3.1.2 Definición

Business Intelligence se puede describir como un concepto que integra por un lado el almacenamiento y por el otro el procesamiento de grandes cantidades de datos, con el principal objetivo de transformarlos en conocimiento y en decisiones en tiempo real, a través de un sencillo análisis y exploración [1].

La definición antes expuesta puede representarse a través de la siguiente fórmula:

$$\text{Datos} + \text{Análisis} = \text{Conocimiento}$$

Con esto se dice que Business Intelligence es el proceso de convertir datos en conocimiento y el conocimiento en acción, para la toma de decisiones [1].

Cabe señalar que Business Intelligence da mayor realce a los procesos de recolectar y utilizar efectivamente la información, con el fin de mejorar la forma de operar de una organización, brindando a sus usuarios, el acceso a la información clave que necesitan para llevar a cabo sus tareas habituales y más precisamente, para poder tomar decisiones oportunas basadas en datos correctos y certeros [4].

Al contar con la información exacta y en tiempo real, es posible, aparte de lo ya mencionado, identificar y corregir situaciones antes de que se conviertan en problemas y en potenciales pérdidas de control de la empresa, pudiendo conseguir nuevas oportunidades o readaptarse frente a la ocurrencia de sucesos inesperados [1].

Esto quiere decir que cuanto más relevante y útil sea la inteligencia que posea una organización sobre un negocio, sus clientes, proveedores, socios, operaciones, etc., mayor será su ventaja competitiva y se podrán tomar mejores decisiones. Esto se debe

simplemente a que, por ejemplo, cuanto más se conoce a los clientes, se logra satisfacer sus necesidades de mejor manera y, por supuesto, anticipar sus necesidades [1].

3.1.3 Proceso de Business Intelligence

Mediante un proceso aplicado a la organización es posible demostrar cómo se puede crear inteligencia de sus datos para ofrecer a los usuarios finales oportuna y acertadamente acceso a este conocimiento, para ello se presentará a continuación el proceso Business Intelligent, el cual está dividido en cinco fases, las que se pueden apreciar en la Figura 3.

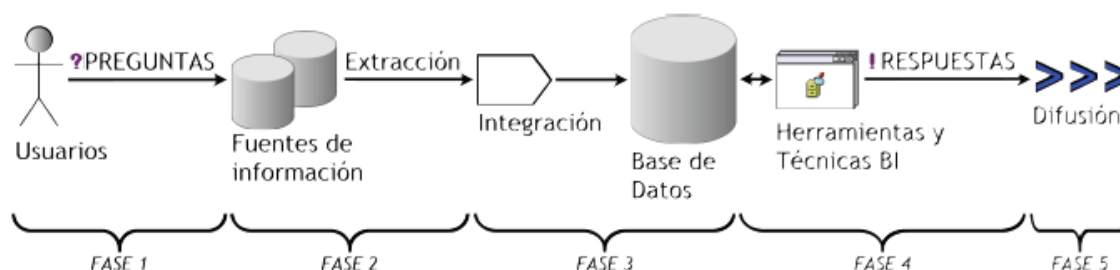


Figura 3: Fases del proceso Business Intelligence [1]

Fase 1 – Dirigir y Planear. En esta fase inicial es donde se deberán recolectar los requerimientos de información específicos de los diferentes usuarios, así como entender sus diversas necesidades, para que luego en conjunto con ellos se generen las preguntas que les ayudarán a alcanzar sus objetivos [1].

Fase 2 – Recolección de Información. Es aquí en donde se realiza el proceso de extraer desde las diferentes fuentes de información de la empresa, tanto internas como externas, los datos que serán necesarios para encontrar las respuestas a las preguntas planteadas en el paso anterior [1].

Fase 3 – Procesamiento de Datos. En esta fase es donde se integran y cargan los datos “en crudo” en un formato utilizable para el análisis. Esta actividad puede realizarse mediante la

creación de una nueva base de datos, agregando datos a una base de datos ya existente o bien consolidando la información.

Fase 4 – Análisis y Producción. Aquí, se procede a trabajar sobre los datos extraídos e integrados, utilizando herramientas y técnicas propias de la tecnología Business Intelligent, para crear inteligencia. Como resultado final de esta fase se obtendrán las respuestas a las preguntas, mediante la creación de reportes, indicadores de rendimiento, cuadros de mando, gráficos estadísticos, etc. [1]

Fase 5 – Difusión. Finalmente, se les entregará a los usuarios que lo requieran las herramientas necesarias, que les permitirán explorar los datos de manera sencilla e intuitiva [1].

3.1.4 Beneficios

Entre los beneficios más importantes que Business Intelligent proporciona a las organizaciones, vale la pena destacar los siguientes [4]:

- Reduce el tiempo mínimo que se requiere para recoger toda la información relevante de un tema en particular, ya que la misma se encontrará integrada en una fuente única de fácil acceso.
- Automatiza la asimilación de la información, debido a que la extracción y carga de los datos necesarios se realizará a través de procesos predefinidos.
- Proporciona herramientas de análisis para establecer comparaciones y tomar decisiones.
- Cierra el círculo que hace pasar de la decisión a la acción.
- Permite a los usuarios no depender de reportes o informes programados, porque los mismos serán generados de manera dinámica.
- Posibilita la formulación y respuesta de preguntas que son claves para el desempeño de la empresa.
- Permite acceder y analizar directamente los indicadores de éxito.

- Se pueden identificar cuáles son los factores que inciden en el buen o mal funcionamiento de la empresa.
- Se podrán detectar situaciones fuera de lo normal.
- Permitirá predecir el comportamiento futuro con un alto porcentaje de certeza, basado en el entendimiento del pasado.
- El usuario podrá consultar y analizar los datos de manera sencilla e intuitiva.

3.2 DataMart

Un DataMart es una base de datos departamental, especializada en el almacenamiento de datos de un área de negocio específica. Se caracteriza por disponer de una estructura óptima de datos, que se utiliza para analizar la información al detalle desde todas las perspectivas, que afecten a los procesos de un departamento. Un DataMart puede ser alimentado desde los datos de un sistema más complejo como lo son los Data Warehouses, o integrar por sí mismo un compendio de distintas fuentes de información [5].

También podemos definir que un DataMart es un subconjunto de un Data Warehouse para un propósito específico de una compañía [6], siendo éste 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 [7].

Un DataMart posee los siguientes componentes [1]:

- Fuentes de datos
- Procesos de extracción, transformación y carga de los datos (ETL)
- Data Warehouse
- Herramientas de exploración

3.3 Data Warehouse

3.3.1 Introducción

Cuando una organización genera datos en todos los ámbitos de su actividad diaria de negocios (compras, ventas, producción, etc.) y/o información externa relacionada, es necesario gestionar estos datos guardados en diversos formatos, fuentes y tipos, para que luego sean depurados e integrados, como también su almacenamiento debe ser centralizado permitiendo así su posterior análisis y exploración. Para ello existe un proceso que satisface estas exigencias, denominado Data Warehousing.

El Data Warehousing está encargado de extraer, transformar, consolidar, integrar y centralizar los datos, permitiendo de esta manera el acceso y exploración de la información requerida, esto ocurre gracias a una amplia gama de posibilidades de análisis multivariantes que ayudan al usuario a realizar un proceso de toma de decisiones estratégico y táctico. [8]

3.3.2 Definición

Un Data Warehouse es un sistema, no un producto, en el que se almacenan datos. Siendo una técnica para consolidar y administrar datos de variadas fuentes con el propósito de responder preguntas de negocios y tomar decisiones, de una forma oportuna. Además un Data Warehouse se vale de una base de datos relacional diseñada para el acceso rápido y análisis, no para procesar transacciones. Por lo que el Data Warehouse separa la carga del análisis y normalmente contiene datos históricos derivados de datos transaccionales [9].

Existen muchas definiciones para el Data Warehouse, la más conocida fue propuesta por William Inmon [10] - considerado el padre del Data Warehouse - en 1992:

"Un Data Warehouse es una colección de datos orientados a temas, integrados, no volátiles y variante en el tiempo, organizados para soportar necesidades empresariales".

En definitiva un Data Warehouse es aquel sistema que permite la extracción y filtro de distintos sistemas operacionales de la empresa, ya sean estas bases de datos

operacionales, hojas de cálculo, subsistemas operacionales, etc., donde los datos extraídos son sometidos a un proceso de transformación, integración y son almacenados en un depósito, para poder acceder a ellos mediante sistemas de aplicación orientados al usuario [11].

3.3.3 Características

William Inmon indicó que un Data Warehouse se caracterizaba por cumplir con los siguientes puntos [12]:

- **Orientada al negocio o a temas:** Los Data Warehouses están diseñados para ayudar a analizar los datos de un determinado tema o significado de negocio. Por ejemplo, para saber más sobre un Centro Hospitalario se puede construir un Data Warehouse que concentre las consultas. Utilizando este Data Warehouse se podrían hacer preguntas del tipo ¿Cuál ha sido la enfermedad más habitual este año?. Esta habilidad de localizar un tema prioritario hace que se cree un Data Warehouse orientado a un tema.
- **Integrado:** La integración está muy relacionada con el punto “temático”. Los Data Warehouses deben aunar datos de fuentes dispares de una forma consistente. Deben resolver problemas tales como el nombre de los campos, conflictos de inconsistencia en unidades de medidas antes de ser almacenados.
- **No volátil:** No volátil significa que una vez introducidos los datos en el Data Warehouse, estos datos no deben ser cambiados. Esto es lógico debido a que el propósito del Data Warehouse es ser capaz de analizar lo que ya ha ocurrido.
- **Variante en el tiempo:** El foco del Data Warehouse se centra en los cambios realizados a lo largo del tiempo, es decir, estudia cómo el tiempo hace evolucionar los diferentes elementos. Para esto se necesita una gran cantidad de datos almacenados a lo largo de mucho tiempo. En esto difiere mucho de un sistema transaccional, donde los datos históricos son archivados y poco accedidos [13].

A continuación se presentarán estos puntos de una forma más detallada.

3.3.3.1 *Orientada al negocio*

La primera característica de un Data Warehouse es que la clasificación de la información se basa en los aspectos de interés para la empresa. Por lo que el diseño y la implementación de los datos encontrados se ven directamente afectados por la clasificación definida con anterioridad, debido a la superioridad en cuanto a los clásicos procesos operacionales orientados a las aplicaciones.

Para ello se definen dos tipos de orientación, en el primero hace referencia al nivel de detalle de los datos, excluyendo de esta forma la información que no tendrán una participación en el proceso de toma de decisiones, por el contrario en los procesos orientados a las aplicaciones se incluyen los datos que satisfacen los requerimientos funcionales de la actividad en cuestión. Y el segundo se basa principalmente en la relación que existe entre los datos del Data Warehouse ya que son muy abundantes, debido a que cada tabla está conformada por la integración de otras tablas o fuentes del ambiente operacional con sus respectivas reglas de negocio.

Por lo que el depósito de datos se diseña para realizar consultas e investigaciones sobre las actividades de la organización y no para soportar los procesos que se realizan en ella, lo que significa que se requiere que la información este desnormalizada, es decir, con redundancia y que la misma esté dimensionada, para evitar tener que recorrer toda la base de datos cuando se necesite realizar algún análisis determinado, sino que simplemente la consulta sea enfocada por variables de análisis que permitan localizar los datos de manera rápida y eficaz, para poder de esta manera satisfacer una alta demanda de complejos exámenes en un mínimo tiempo de respuesta. [14]

A modo de ejemplo se presenta la combinación de elementos tales como datos de clientes, productos y cuentas, en una estructura que se acomoda a las necesidades de la aplicación que posee funciones tales como préstamos, ahorros, tarjetas bancarias y depósitos. Por lo que el Data Warehouse se organiza alrededor de sujetos tales como cliente, vendedor, producto y actividad, como se muestra en la Figura 4.

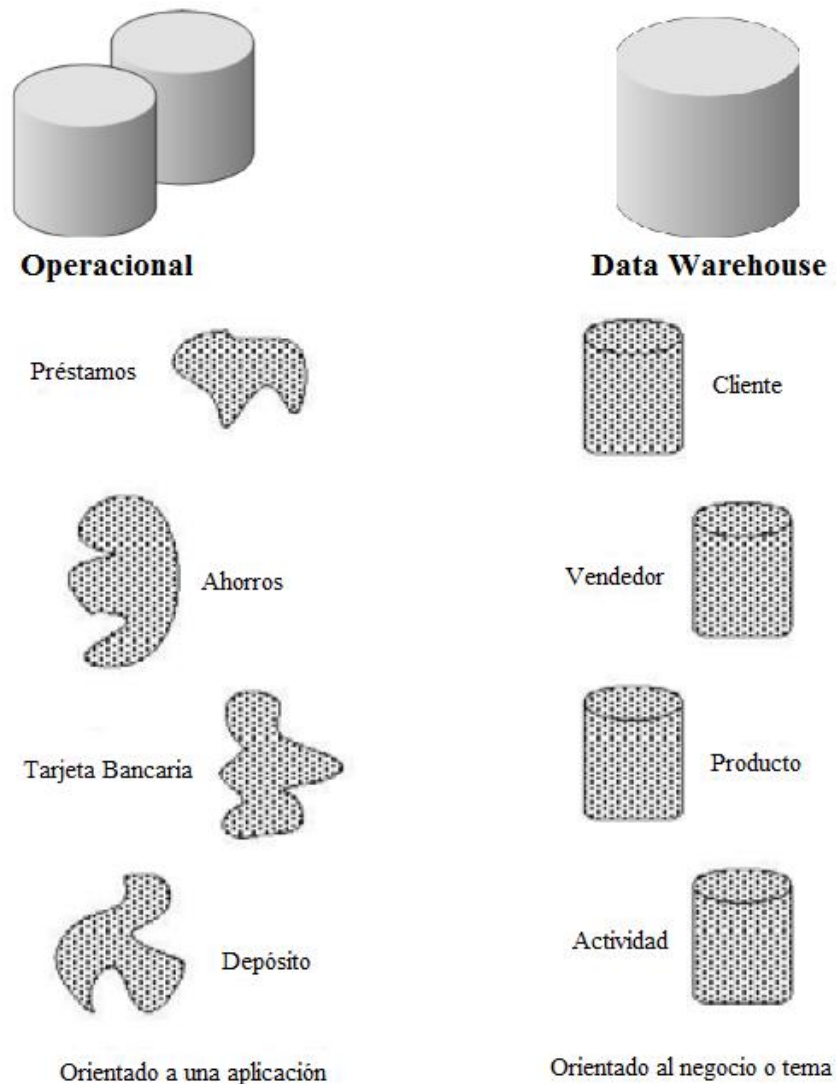


Figura 4: Data Warehouse Orientado al Negocio [14]

3.3.3.2 Integrada

A través del tiempo los diseñadores y programadores no han propuesto ningún estándar concreto para definir nombres de variables, tipos de datos, etc. Por lo que cada aplicación ha tenido la libertad de definir sus propios estilos personalizados, lo que afecta la relación de modelos entre sí por sus inconsistencias e incompatibilidades.

Para ello existe el proceso de Extracción, Transformación y Carga de los Datos (Extraction, Transformation and Load – ETL), que se aplica a todos los datos de diversas

fuentes producidos por los distintos departamentos, secciones y aplicaciones ya sea interna o externamente para ser consolidados en una instancia antes de ser agregados al Data Warehouse.

A modo de ejemplo, en la Figura 5 se puede apreciar que los diseñadores de aplicaciones codifican el campo GÉNERO en varias formas. Un diseñador representa GÉNERO como una “M” y una “F”, otros como un “1” o un “0”, otros como una “X” y una “Y” e inclusive como “masculino” y “femenino”. No importa mucho cómo el GÉNERO llega al Data Warehouse, lo importante es que sea de cualquier fuente de donde venga, el GÉNERO debe llegar el Data Warehouse en un estado uniforme.

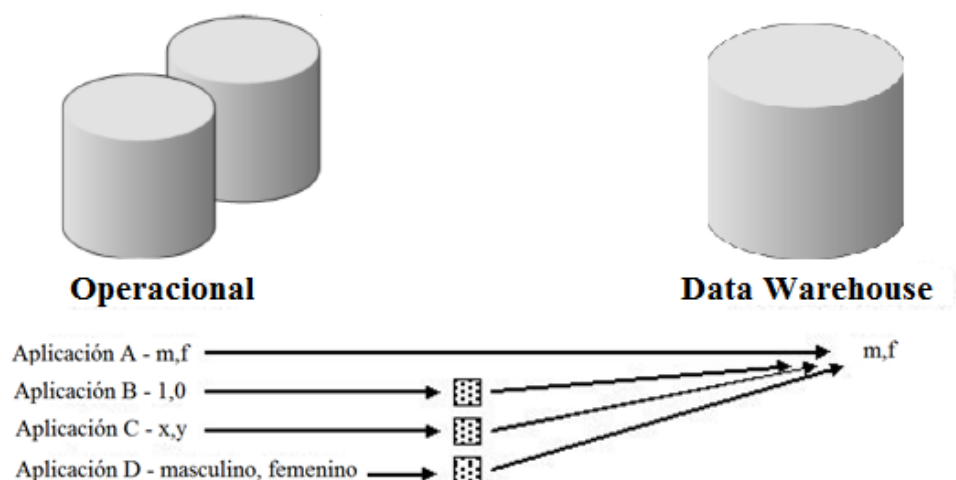


Figura 5: Data Warehouse Integrado [14]

Cuando los datos se mueven al Data Warehouse desde las aplicaciones orientadas al ambiente operacional, los datos se integran antes de entrar al depósito de datos.

3.3.3.3 Variante en el tiempo

Para acceder y procesar el gran volumen de información que contiene el Data Warehouse, se requieren cantidades de tiempos diferentes para obtener los resultados de la

consulta, por lo que esta cantidad de tiempo es normal en este ambiente y es por eso que la información que se encuentra dentro del depósito de datos se denomina, de tiempo variable, esto quiere decir que la información del Data Warehouse posee un elemento de tiempo asociado a sus datos.

Debido a lo anterior, se logra que el Data Warehouse pueda entregar los resultados de una consulta a un gran volumen de datos, almacenados históricamente, en un tiempo de respuesta relativamente igual a datos recientemente almacenados, siendo ésta una de sus principales características. Esta cualidad, que no se encuentra presente en fuentes de datos operacionales, garantiza poder desarrollar un análisis de la dinámica de la información [14].

A continuación, en la Figura 6 se presenta un Data Warehouse histórico de información, que representa los datos sobre un horizonte largo de tiempo – desde cinco a diez años. En cambio, el horizonte de tiempo representado para el ambiente operacional es mucho más corto – desde valores actuales hasta sesenta o noventa días.

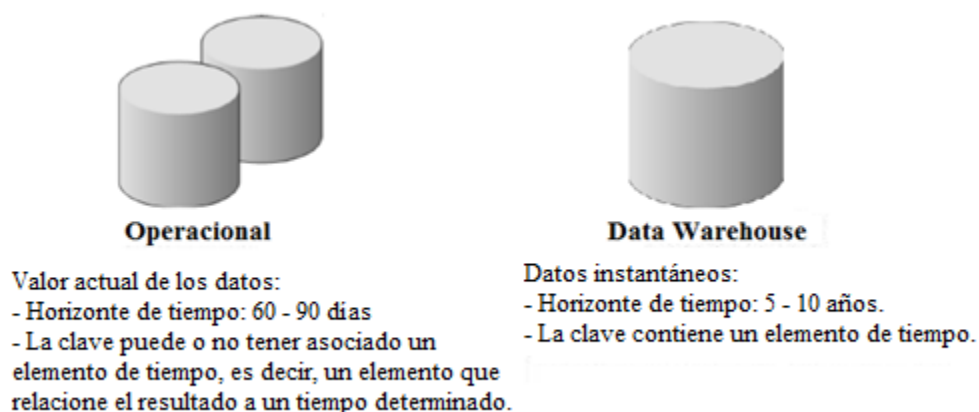


Figura 6: Data Warehouse Histórico [14]

El intervalo de tiempo y periodicidad de los datos debe definirse de acuerdo a la necesidad y requisitos de los usuarios.

Es elemental aclarar, que el almacenamiento de datos históricos, es lo que permite al Data Warehouse desarrollar pronósticos y análisis de tendencias y patrones, a partir de una base estadística de información.

3.3.3.4 No volátil

En los sistemas operacionales los datos cambian, o sea, se realizan con frecuencia procesos de actualización, inserción, eliminación, limpieza de datos, en cambio en el Data Warehouse los datos ingresados no deben cambiar, aunque en algunas ocasiones puede ocurrir algún cambio en específico.

Obteniendo de esta forma, una información útil para el análisis y la toma de decisiones debido a la estabilidad de los datos. Es por ello que sólo existe una manipulación básica de los datos, como lo son carga de los datos y acceso a los mismos.

Como ejemplo, en el accionar diario de una empresa, el Data Warehouse no debe preocuparse por las anomalías de actualización que se encuentran en el ambiente operacional, debido a esto existen diferencias en los procesos, permitiendo al Data Warehouse tomar libertades para optimizar el acceso a los datos, pudiendo ser en este caso a través de la normalización y desnormalización física de los datos [1].

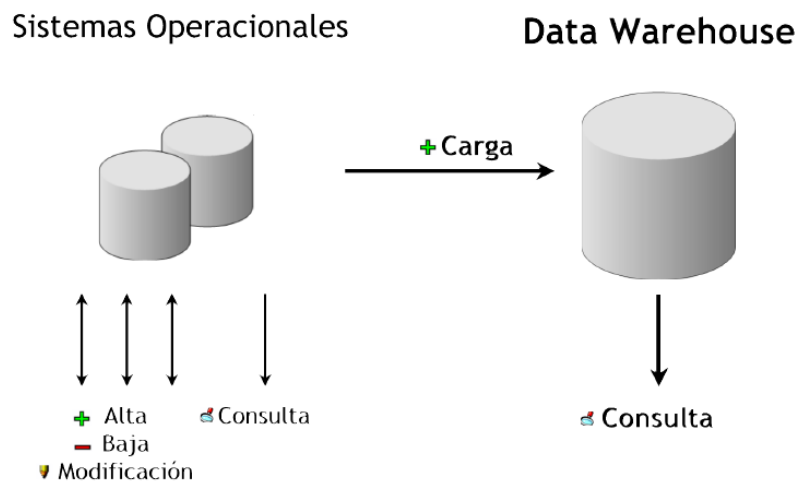


Figura 7: Data Warehouse No Volátil [1]

Por esta razón es que en el Data Warehouse no se requieren mecanismos de control de la concurrencia y recuperación.

3.3.4 Cualidades

Un Data Warehouse maneja un gran volumen de datos, esto se debe a que posee información histórica de la organización, extraída de diferentes fuentes y áreas, almacenadas en un sólo lugar centralizado. Esto permite que diversos medios de almacenamiento puedan soportar y mantener el depósito de datos.

Cabe señalar que el almacén de datos contiene información resumida y agregada de múltiples versiones.

Además posee la cualidad de organizar y almacenar los datos necesarios para efectuar consultas y procesos analíticos, con el objetivo de dar respuesta a preguntas complejas. Permitiendo de esta forma entregar la información de manera amigable, intuitiva y fácil de utilizar para la toma de decisiones. Debido a que la información se encuentra enfocada en torno al negocio los usuarios pueden tener un acceso más directo, en cuanto a la exploración de los datos y localizar relaciones complejas entre los mismos.

Otra característica es que no sólo es un depósito de datos, sino que además posee un conjunto de herramientas para consultar, analizar y presentar información, que permiten obtener o realizar análisis, extracción y explotación de los datos, con alta performance, para transformar dichos datos en información valiosa para la organización.

Con respecto a las tecnologías que son empleadas por los sistemas Data Warehouse, se pueden encontrar las siguientes:

- Arquitectura cliente/servidor.
- Técnicas avanzadas para replicar, refrescar y actualizar datos.
- Software front-end, para acceso y análisis de datos.
- Herramientas para extraer, transformar y cargar datos en el depósito, desde múltiples fuentes muy heterogéneas.
- Sistema de Gestión de Base de Datos (SGBD).

Todas las cualidades expuestas anteriormente son imposibles de cumplir en un típico ambiente operacional, y esto es una de las razones de ser del Data Warehousing [1].

3.3.5 Objetivos generales

Los objetivos generales en su mayoría intentan entregar una idea de lo que se pretende a grandes rasgos con un Data Warehouse, entregando información más global.

A continuación se detallarán los objetivos específicos que según Ralph Kimball [15] en su libro “The Data Warehouse Toolkit”, se pretenden alcanzar en el sistema Data Warehouse:

1. El Data Warehouse debe hacer la información de la organización fácilmente accesible.

El contenido del Data Warehouse debe ser comprensible, intuitivo y obvio para el usuario de negocio. La comprensibilidad implica legibilidad, por lo que el contenido del Data Warehouse necesita ser etiquetado de manera significativa. El usuario de negocio debe estar habilitado para extraer porciones del Data Warehouse y combinar esta información de todas las formas posibles, utilizando herramientas simples y fáciles de usar, con un tiempo de respuesta mínimo.

2. El Data Warehouse debe presentar la información de la organización consistentemente.

La información del Data Warehouse debe ser creíble. Los datos además tienen que ser cuidadosamente reunidos, de una variedad de orígenes en toda la organización. Estos deben ser limpiados, con calidad asegurada, y liberados cuando sean aptos para la utilización de los usuarios. La información de un proceso de negocio, debe coincidir con la información de otro proceso que se encuentre también dentro del Data Warehouse. Si dos métricas no significan lo mismo, entonces deben ser nombrados de forma distinta. Para que la información sea consistente, esta debe ser de alta calidad, esto significa que todos los datos sean correctos y estén completos. La consistencia también implica que las definiciones comunes del contenido de un Data Warehouse están disponibles para todos los usuarios.

3. El Data Warehouse debe ser adaptable y resistente a cambios.

Simplemente no se pueden evitar los cambios. Las necesidades de los usuarios, las condiciones del negocio, los datos, y la tecnología, están sujetos a los cambios en el transcurso del tiempo. El Data Warehouse debe estar diseñado para manejar estos inevitables cambios. Los cambios en el Data Warehouse no deben invalidar los datos o las aplicaciones existentes, además de no ser alterados o quebrantados cuando la comunidad de usuarios realice nuevas preguntas o se agreguen nuevos datos al Data Warehouse. Si los datos descriptivos del Data Warehouse son modificados, se debe tener en cuenta esos cambios adecuadamente.

4. El Data Warehouse debe ser un soporte que resguarde la información de la organización.

La información obtenida de la organización se encuentra almacenada en el Data Warehouse. Por lo que el Data Warehouse probablemente contiene información sobre qué se vendió, a quién y a qué precio – detalles potencialmente peligrosos en las manos de gente inapropiada. El Data Warehouse debe controlar efectivamente el acceso a la información confidencial de la organización.

5. El Data Warehouse debe servir como base para una toma de decisiones mejorada.

El Data Warehouse debe contener la información correcta para soportar los procesos que corresponden a la toma de decisiones. De esta forma permite generar decisiones que son consecuencia de los resultados entregados por el Data Warehouse. Estas decisiones entregan al negocio el impacto y el valor atribuible al Data Warehouse.

6. La organización debe aceptar al Data Warehouse para que se considere un éxito.

No importa que se haya construido una elegante solución usando los mejores productos y plataformas de su clase. Si los usuarios de la organización no aceptan el Data Warehouse y desiste de su uso luego de un tiempo posterior a la capacitación, entonces se ha fallado en la prueba de aceptación. A diferencia de las nuevas implantaciones de sistemas operacionales, donde los usuarios no tienen otra opción que usar el nuevo sistema, el uso del Data Warehouse es a veces opcional. La aceptación de los usuarios tiene más que ver con la simplicidad que con cualquier otra cosa.

Esto refleja claramente que la creación de un Data Warehouse no sólo tiene una alta demanda de conocimientos técnicos, sino que además involucra un dominio de los procesos de negocio, un cambio cultural de la organización y una visión muy clara, con objetivos bien definidos.

3.3.6 Ventajas

A continuación se enumerarán algunas de las ventajas más sobresalientes que trae aparejada la implementación de un Data Warehousing y que ejemplifican de mejor modo sus características y cualidades [1]:

- Transforma datos orientados a las aplicaciones en información orientada a la toma de decisiones.
- Integra y consolida diferentes fuentes de datos (internas y/o externas) y departamentos empresariales, que anteriormente formaban islas, en una única plataforma sólida y centralizada.
- Provee la capacidad de analizar y explotar las diferentes áreas de trabajo y de realizar un análisis inmediato de las mismas.
- Permite reaccionar rápidamente a los cambios del mercado.
- Aumenta la competitividad en el mercado.
- Elimina la producción y el procesamiento de datos que no son utilizados ni necesarios, producto de aplicaciones mal diseñadas o ya no utilizadas.
- Mejora la entrega de información, es decir, información completa, correcta, consistente, oportuna y accesible. Información que los usuarios necesitan, en el momento adecuado y en el formato apropiado.
- Logra un impacto positivo sobre los procesos de toma de decisiones. Cuando los usuarios tienen acceso a una mejor calidad de información, la empresa puede lograr por sí misma: aprovechar el enorme valor potencial de sus recursos de información y transformarlo en valor verdadero; eliminar los retardos de los procesos que resultan de información incorrecta, inconsistente y/o inexistente; integrar y

optimizar procesos a través del uso compartido e integrado de las fuentes de información; permitir al usuario adquirir mayor confianza acerca de sus propias decisiones y de las del resto, y lograr así, un mayor entendimiento de los impactos ocasionados.

- Aumento de la eficiencia de los encargados de tomar decisiones.
- Los usuarios pueden acceder directamente a la información en línea, lo que contribuye a su capacidad para operar con mayor efectividad en las tareas rutinarias. Además, pueden tener a su disposición una gran cantidad de valiosa información multidimensional, presentada coherentemente como fuente única, confiable y disponible en sus estaciones de trabajo. Así mismo, los usuarios tienen la facilidad de contar con herramientas que les son familiares para manipular y evaluar la información obtenida desde el Data Warehouse, tales como: hojas de cálculo, procesadores de texto, software de análisis de datos, software de análisis estadístico, reportes, tableros, etc.
- Permite la toma de decisiones estratégicas y tácticas.

3.3.7 Desventajas

A continuación se enumerarán algunas de las desventajas más comunes que se pueden presentar en la implementación de un Data Warehousing [1]:

- Requiere una gran inversión, debido a que su correcta construcción no es tarea sencilla y consume muchos recursos, además, su misma implementación implica desde la adquisición de herramientas de consulta y análisis, hasta la capacitación de los usuarios.
- Existe resistencia al cambio por parte de los usuarios.
- Los beneficios del almacén de datos son apreciados en el mediano y largo plazo.
- Este punto deriva del anterior, y básicamente se refiere a que no todos los usuarios confiarán en el Data Warehouse en una primera instancia, pero sí lo harán una vez que comprueben su efectividad y ventajas. Además, su correcta utilización surge de la propia experiencia.

- Si se incluyen datos propios y confidenciales de clientes, proveedores, etc., el depósito de datos atentará contra la privacidad de los mismos, ya que cualquier usuario podrá tener acceso a ellos.
- Subvaloración de los recursos necesarios para la captura, carga y almacenamiento de los datos.
- Subvaloración del esfuerzo necesario para su diseño y creación.
- Incremento continuo de los requerimientos del usuario.
- Subestimación de las capacidades que puede brindar la correcta utilización del Data Warehouse y de las herramientas de Business Intelligent en general.

3.3.8 Redundancia

Se dice erróneamente que el Data Warehouse almacena una redundancia de datos masiva entre el ambiente operacional y el Data Warehouse, debido a la información histórica recibida por diversas fuentes. Siendo lo anterior falso, a continuación se presentan los motivos que niegan lo antes propuesto:

- Para que los datos pertenezcan al Data Warehouse primero deben ser filtrados del ambiente operacional, por lo que solo pasan los datos que permitan obtener información relevante para la toma de decisiones.
- En segundo lugar, el horizonte de tiempo es completamente diferente entre el ambiente operacional y el Data Warehouse.
- El Data Warehouse contiene un resumen de la información que no existe en el ambiente operacional.
- Y por último los datos son transformados, para luego ser cargados al Data Warehouse. Lo que significa que la mayor parte de los datos son alterados para su selección, consolidándolos y posteriormente movidos al depósito.

En consecuencia, se puede afirmar que, la redundancia encontrada al cotejar los datos de ambos ambientes es mínima, ya que generalmente resulta en un porcentaje menor al 1% [1].

3.3.9 Estructura

Debido a la alta complejidad de un Data Warehouse, los datos se estructuran en diversos niveles de esquematización y detalle que los limitan.

En la Figura 8 se muestra de mejor forma la estructura.

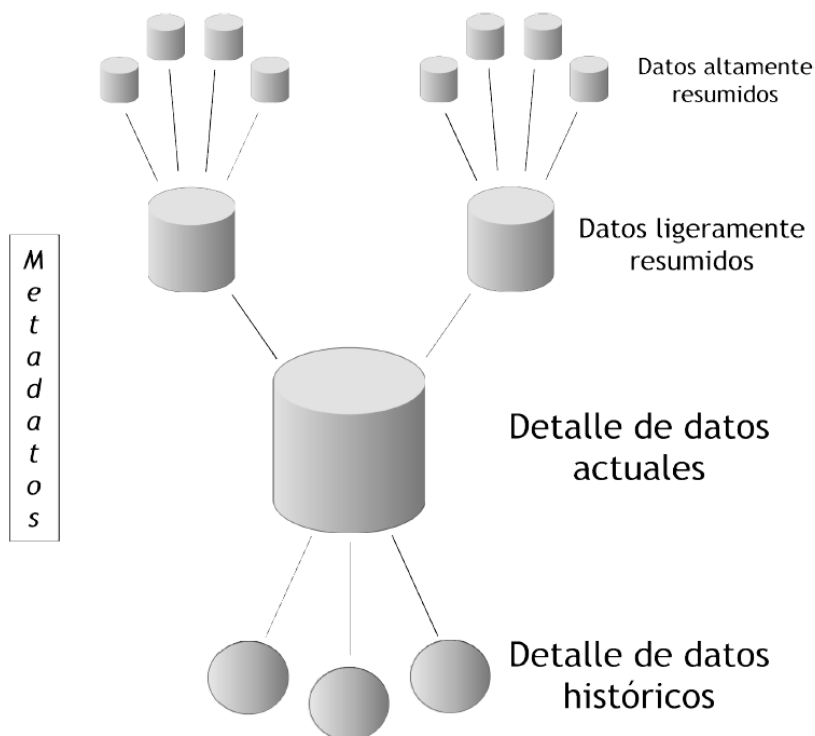


Figura 8: Estructura de un Data Warehouse [1]

Como se puede observar, los almacenes de datos están compuestos por diversos tipos de datos, que se organizan y dividen de acuerdo al nivel de detalle o granularidad que posean.

A continuación se explicarán cada uno de estos tipos de datos [1]:

- Detalle de datos actuales: son aquellos que reflejan las ocurrencias más recientes.
- Generalmente se almacenan en disco, aunque su administración sea costosa y compleja, con el fin de conseguir que el acceso a la información sea sencillo y

veloz, ya que son bastante voluminosos. Su gran tamaño se debe a que los datos residentes poseen el más bajo nivel de granularidad, o sea, se almacenan a nivel de detalle.

Por ejemplo, aquí es donde se guardaría el detalle de una venta realizada en tal fecha.

- **Detalle de datos históricos:** representan aquellos datos antiguos, que no son frecuentemente consultados. También se almacenan a nivel de detalle, normalmente sobre alguna forma de almacenamiento externa, ya que el volumen de los datos es demasiado alto y en adición a esto, no son requeridos con mucha periodicidad. Este tipo de datos son consistentes con los detalles de los datos actuales. Por ejemplo, en este nivel, al igual que en el anterior, se encontraría el detalle de una venta realizada en tal fecha, pero con la particularidad de que el día en que se registró la venta debe ser lo suficientemente antigua, para que se considere como histórica.
- **Datos ligeramente resumidos:** son los que provienen desde un bajo nivel de detalle y se agrupan estos datos bajo algún criterio o condición de análisis. Habitualmente son almacenados en disco. Por ejemplo, en este caso se almacenaría el resumen del detalle de las ventas realizadas en cada mes.
- **Datos altamente resumidos:** son aquellos que compactan aún más a los datos ligeramente resumidos. Se guardan en disco y son muy fáciles de acceder. Por ejemplo, aquí se encontraría el resumen de las ventas realizadas en cada año.
- **Metadatos:** representan la información acerca de los datos. De muchas maneras se sitúa en una dimensión diferente al de otros datos del Data Warehouse, ya que su contenido no es tomado directamente desde el ambiente operacional.

Estos diferentes niveles de detalle o granularidad, se obtienen a través de tablas de hechos agregadas y/o pre agregadas.

3.3.9.1 Tablas de hechos agregadas y pre agregadas

Un Data Warehouse contiene tablas de hechos agregadas y pre agregadas, que almacenan un resumen de los datos, esto quiere decir que se almacenan los datos en niveles de granularidad superior a los que inicialmente fueron obtenidos.

Para definir las tablas es necesario establecer un criterio con el cual realizar el resumen, cada vez que se requiere que los datos en una consulta se presenten en un nivel de granularidad superior al que se encuentran alojados en el Data Warehouse, se debe llevar a cabo un proceso de agregación.

En general los contenidos de este punto fueron resumidos de la metodología de Hefesto [1], ya que es quien mejor detalla los conceptos de las tablas agregadas y pre agregadas.

El objetivo general de las tablas de hechos agregadas y pre agregadas es similar, pero cada una de ellas tiene una manera de operar diferente:

- Tablas de hechos agregadas: se generan luego de que se procesa la consulta correspondiente a la tabla de hechos que se resumirá. En general, la agregación se produce dinámicamente a través de una instrucción SQL que incluya sumalizaciones.
- Tablas de hechos pre agregadas: se generan antes de que se procese la consulta correspondiente a la tabla de hechos que se resumirá. De esta manera, la consulta se realiza contra una tabla que ya fue previamente sumarizada. Habitualmente, estas sumalizaciones se calculan a través de procesos ETL.

Las tablas de hechos pre agregadas cuentan con los siguientes beneficios:

- Reduce la utilización de recursos de hardware que normalmente son incurridos en el cálculo de las sumalizaciones.
- Reduce el número de registros que serán analizados por el usuario.
- Reduce el tiempo utilizado en la generación de consultas por parte del usuario.

Las tablas de hechos pre agregadas son muy útiles en los siguientes casos generales:

- Cuando los datos a nivel detalle (menor nivel granular) son innecesarios y/o no son requeridos.
- Cuando una consulta sumariada a determinado nivel de granularidad es solicitado con mucha frecuencia.
- Cuando los datos son muy abundantes, y las consultas demoran en ser procesadas demasiado tiempo.

Como contrapartida, las tablas de hechos pre agregadas presentan una serie de desventajas:

- Requieren que se mantengan y gestionen nuevos procesos ETL.
- Demandan espacio de almacenamiento extra en el depósito de datos.

3.3.10 Flujo de Datos

El flujo de datos del Data Warehouse se puede observar en la Figura 9.

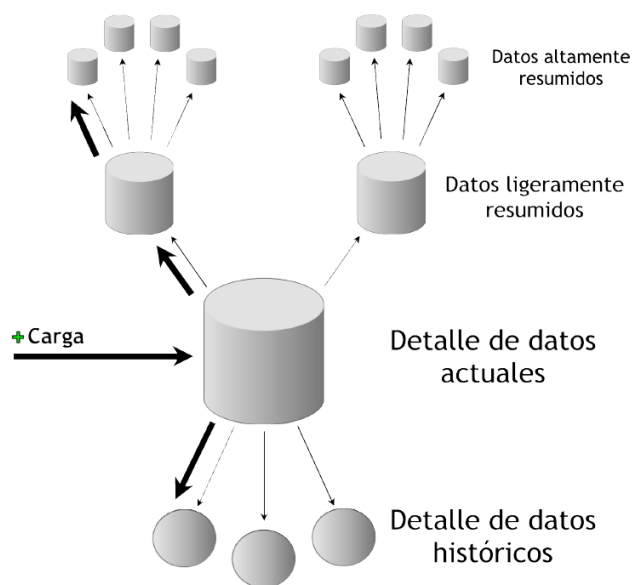


Figura 9: Flujo de Datos de un Data Warehouse [1]

El almacenamiento de la información ingresada a nivel de detalle de datos actuales en el Data Warehouse, depende directamente de la modificación de los siguientes eventos:

- Sean borrados del depósito de datos.
- Sean resumidos, ya sea a nivel de Datos ligeramente resumidos o a nivel de Datos altamente resumidos.
- Sean archivados a nivel de Detalle de datos históricos.

3.4 Metodologías de desarrollo de un Data Warehouse

Para obtener los resultados esperados de un Data Warehouse, los procesos de negocio se seleccionan con el objetivo de modelarlos, para así establecer una granularidad de cada uno de ellos. Por este motivo es necesario comprender correctamente los datos de los diferentes sistemas dentro de una organización y sus respectivas relaciones. La gestión de estas relaciones durante la carga de almacenamiento de datos es esencial. Es por esto que para el desarrollo de un Data Warehouse no existe una única metodología en la que se basará el diseño, dependiendo exclusivamente del contexto en el que se encuentra la empresa y los objetivos que persiga.

Estas diferentes metodologías se pueden englobar dentro de dos grandes bloques:

- Top-down
- Bottom-up

3.4.1 Modelo top-down

Este modelo fue propuesto por Bill Inmon [10], el enfoque top-down se utiliza cuando la tecnología y los problemas del negocio se conocen con anterioridad. Este enfoque logra la sinergia entre los problemas de negocio alcanzando los objetivos

buscados. Se trata de un método sistémico, que minimiza los problemas de integración, pero es costoso, debido a la gran cantidad de datos y su poca flexibilidad.

Además, en este método se formula un resumen del sistema, sin especificar detalles. Cada parte del sistema se refina diseñándola con mayor detalle. Después, cada parte nueva se vuelve a refinar, cada vez con mayor detalle, hasta que la especificación completa es lo suficientemente detallada para validar el modelo. Este modelo se diseña con frecuencia con la ayuda de "cajas negras" que hacen más fácil cumplir requerimientos, aunque estas cajas negras no expliquen en detalle los componentes individuales.

El enfoque top-down considera que el almacén de datos debe responder a las necesidades de todos los usuarios en la organización, y no sólo de un determinado grupo.

3.4.1.1 Metodología más empleada dentro del modelo top-down

3.4.1.1.1 Metodología propuesta por Hefesto [1]

La metodología HEFESTO, permite la construcción de Data Warehouse de forma sencilla, ordenada e intuitiva. Su nombre fue inspirado en el Dios griego de la construcción y el fuego.

HEFESTO es una metodología propia, cuya propuesta está fundamentada en una muy amplia investigación, comparación de metodologías existentes y experiencias propias en procesos de confección de almacenes de datos.

La idea principal, es comprender cada paso que se realizará, para no caer en el tedio de tener que seguir un método al pie de la letra sin saber exactamente qué se está haciendo, ni por qué.

La construcción e implementación de un Data Warehouse puede adaptarse muy bien a cualquier ciclo de vida de desarrollo de software, con la salvedad de que para algunas fases en particular, las acciones que se han de realizar serán muy diferentes. Lo que se debe tener muy en cuenta, es no entrar en la utilización de metodologías que requieran fases

extensas de reunión de requerimientos y análisis, fases de desarrollo monolítico que conlleve demasiado tiempo y fases de despliegue muy largas. Lo que se busca, es entregar una primera implementación que satisfaga una parte de las necesidades, para demostrar las ventajas del Data Warehouse y motivar a los usuarios.

La metodología HEFESTO, puede ser embebida en cualquier ciclo de vida que cumpla con la condición antes declarada.

Con el fin de que se llegue a una total comprensión de cada paso o etapa, se acompañará con la implementación en una empresa real, para demostrar los resultados que se deben obtener y ejemplificar cada concepto.

3.4.1.1.1 Descripción

La metodología HEFESTO puede resumirse a través de la Figura 10.

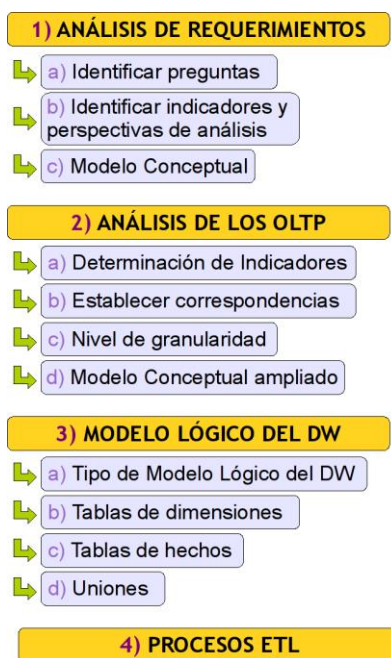


Figura 10: Metodología HEFESTO

Como se puede apreciar en la Figura 10, se comienza recolectando las necesidades de información de los usuarios y se obtienen las preguntas claves del negocio. Luego, se deben identificar los indicadores resultantes de los interrogativos y sus respectivas perspectivas o dimensiones de análisis, mediante las cuales se construirá el modelo conceptual de datos del Data Warehouse.

Después, se analizarán los OLTP para determinar cómo se construirán los indicadores como por ejemplo el total de las unidades vendidas o de ganancias de uno o varios productos, señalar las correspondencias con los datos fuentes y para seleccionar los campos de estudio de cada perspectiva.

Una vez hecho esto, se pasará a la construcción del modelo lógico del depósito, en donde se definirá cuál será el tipo de esquema que se implementará. Seguidamente, se confeccionarán las tablas de dimensiones que son elementos que contienen atributos (o campos) que se utilizan para restringir y agrupar los datos almacenados en una tabla de hechos cuando se realizan consultas sobre dicho datos en un entorno Data Warehouse o DataMart. Finalmente, se definirá la tabla de hechos la cual es la tabla central de un esquema dimensional (en estrella o en copo de nieve) que contiene los valores de las medidas de negocio, para luego efectuar sus respectivas uniones.

Por último, se definirán los procesos de extracción, transformación y carga de los datos fuente, que poblarán y actualizarán el Data Warehouse.

3.4.1.1.1.2 Características

Esta metodología cuenta con las siguientes características:

- Los objetivos y resultados esperados en cada fase se distinguen fácilmente y son sencillos de comprender.
- Se basa en los requerimientos del usuario, por lo cual su estructura es capaz de adaptarse con facilidad y rapidez ante los cambios en el negocio.

- Reduce la resistencia al cambio, ya que involucra al usuario final en cada etapa para que tome decisiones respecto al comportamiento y funciones del Data Warehouse.
- Utiliza modelos conceptuales y lógicos, los cuales son sencillos de interpretar y analizar.
- Es independiente del tipo de ciclo de vida que se emplee para contener la metodología.
- Es independiente de las herramientas que se utilicen para su implementación.
- Es independiente de las estructuras físicas que contengan el Data Warehouse y de su respectiva distribución.
- Cuando se culmina con una fase, los resultados obtenidos se convierten en el punto de partida para llevar a cabo el paso siguiente.
- Se aplica tanto para Data Warehouse como para DataMart.

3.4.1.1.2 Metodología propuesta por Bill Inmon [10]

Esta metodología propone los mecanismos necesarios para llevar a cabo la correcta realización de un Data Warehouse. Para Bill Inmon, el diseño de un Data Warehouse comienza con los procesos que definirán el almacenamiento de los datos en el sistema, debido a que se almacenarán grandes volúmenes de datos, por lo que depende de este proceso la eficiencia en el acceso a los datos.

Además, la definición de Inmon sustenta uno de los principios fundamentales del desarrollo de un Data Warehouse, principio en el que el ambiente de origen de los datos y el ambiente de acceso de los datos, deben estar físicamente en diferentes bases de datos y en equipos separados.

Por último, los actuales sistemas tienen gran cantidad de datos, lo que hace poco realista el intentar hacer cargas cada poco tiempo. Si el volumen de datos no es cuidadosamente gestionado y condensado, dicho volumen de datos impide que los objetivos del Data Warehouse se alcancen.

A Inmon se le asocia frecuentemente con los Data Warehouse a nivel empresarial, que involucran desde un inicio todo el ámbito corporativo, sin centrarse en un incremento específico hasta después de haber terminado completamente el diseño del Data Warehouse. En su filosofía, un DataMart es sólo una de las capas del Data Warehouse y los DataMart son dependientes del depósito central de datos o Data Warehouse Corporativo y, por lo tanto, se construyen después de él.

En el enfoque de Inmon, es necesario desarrollar una estrategia de Data Warehouse e identificar las principales áreas desde el inicio del proyecto, conociendo de esta forma con antelación y bastante exactitud la estructura que presentarán los principales núcleos de desarrollo en la empresa, para asegurar una solución integral, evitando la aparición de situaciones inesperadas que puedan poner en peligro el proyecto.

Inmon es defensor de utilizar el modelo relacional para el ambiente en el que se implementará el Data Warehouse Corporativo, ya que como él mismo afirma, la creación de una base de datos relacional con una ligera normalización, son la base de los DataMart.

El desarrollo de la metodología propuesta por Inmon se aprecia en la Figura 11.

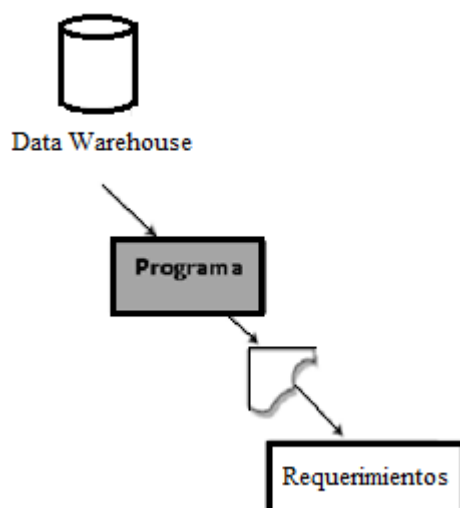


Figura 11: Metodología propuesta por Bill Inmon

La metodología de Inmon tiene un enfoque a modo de explosión en el sentido de que en cierto modo no viene acompañada del ciclo de vida normal de las aplicaciones, sino que los requisitos irán acompañando al proyecto según vaya comprobándose su necesidad. Esta visión de Inmon puede traer consigo mucho riesgo a la compañía, ya que invierte grandes esfuerzos en el desarrollo del Data Warehouse y no es hasta la aparición de los DataMart cuando se empieza a explotar la inversión y obtener beneficios. Esta estrategia se contempla en el marco de que es imposible conocer cuáles son las necesidades concretas de información de una empresa, el ambiente dinámico en que se mueve la organización, el cambio de estructura que conlleva el desarrollo de la nueva plataforma y los consiguientes cambios a los sistemas transaccionales que implica su introducción. Esto hace muy probable que después de la gran inversión en tiempo y recursos en el desarrollo del Data Warehouse, se haga patente la necesidad de cambios fundamentales que traen consigo altos costos de desarrollo para la organización, poniendo en evidente peligro el éxito de todo el proyecto en sí y que podían ser evitados con una pronta detección en una temprana puesta en explotación de un primer avance del Data Warehouse.

Esta metodología para la construcción de un sistema de este tipo, es frecuente a la hora de diseñar un sistema de información, utilizando las herramientas habituales como el esquema Entidad/Relación, pero al tener un enfoque global, es más difícil de desarrollar en un proyecto sencillo, pues estamos intentando abordar el “todo”, a partir del cual luego iremos al “detalle”. Esta es otra de las restricciones que trabajan en contra de la metodología de Inmon ya que implica un consumo de tiempo mayor, teniendo como consecuencia que muchas empresas se inclinen por usar metodologías con las que obtengan resultados tangibles en un espacio menor de tiempo.

A modo de resumen, la metodología de Inmon es distinta a las otras, ya que en estas metodologías convencionales generalmente se habla de conformar un Data Warehouse a partir de distintas fuentes, generando de esta forma los DataMarts, que en este caso, estarían todos en un mismo repositorio, también respetando el modelo como se aprecia en la Figura 12.

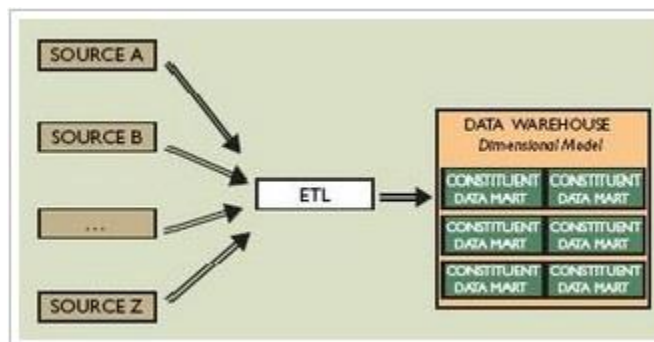


Figura 12: Ejemplo de Metodologías Convencionales

En cambio en la metodología propuesta por Inmon, en este caso el Data Warehouse no está modelado dimensionalmente, sino que está en tercera forma normal (3NF). Así, el creador de este modelo entiende que esta forma es mucho más rica y adaptable que el modelo anterior. Una vez que se tiene el Data Warehouse generado, se pueden crear los DataMarts para las áreas de negocio que se necesiten, y además se puede utilizar para cualquier otro tipo de sistema decisional como por ejemplo sistemas expertos, o minería de datos. A continuación la Figura 13, ilustra el modelo propuesto por Inmon.

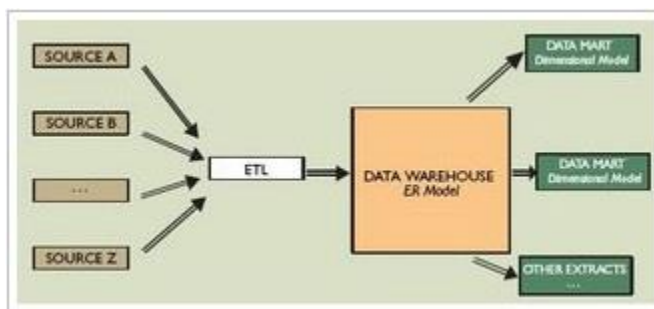


Figura 13: Ejemplo Metodología de Inmon

3.4.2 Modelo Bottom-up

Es un modelo que se caracteriza por ser rápido ya que se basa en experimentos y prototipos propuesta por Ralph Kimball [15]. Siendo este un método flexible que permite a la organización ir más lejos con menores costos. La idea es construir DataMart independientes para evaluar las ventajas del nuevo sistema a medida que avanzamos. En él,

las partes individuales se diseñan con detalle y luego se enlazan para formar componentes más grandes, que a su vez se enlazan hasta que se forma el sistema completo.

Las estrategias basadas en el flujo de información Bottom-up pueden ser potencialmente necesarias y suficientes, porque se basan en el conocimiento de todas las variables que pueden afectar a los elementos del sistema.

3.4.2.1 Metodologías más empleadas dentro del modelo Bottom-up

3.4.2.1.1 Rapid Warehousing Methodology (RWN) [16]

Dicha metodología es iterativa y está basada en el desarrollo incremental del proyecto de Data Warehouse dividido en seis fases como se aprecia en la Figura 14.

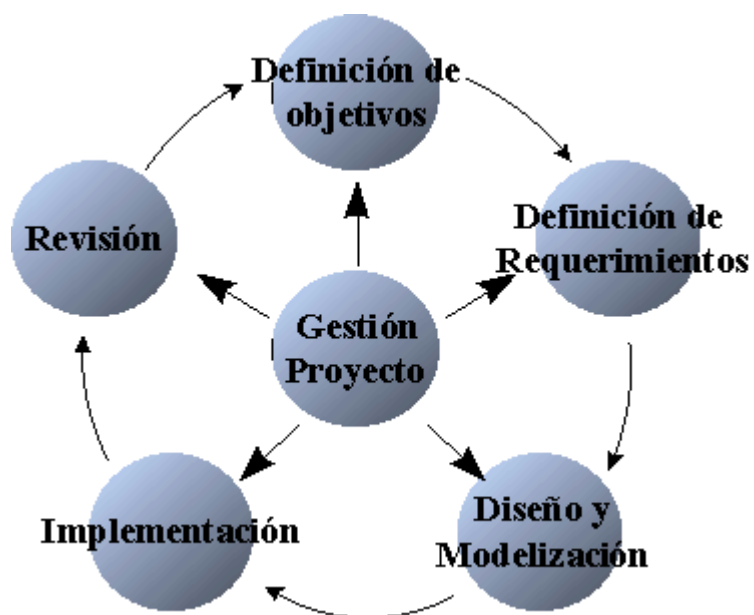


Figura 14: Rapid Warehousing Methodology [17]

A continuación se describe con mayor detalle cada una de las fases de la metodología [1] [17]:

3.4.2.1.1.1 Definición de los objetivos

En esta fase se definirá el equipo de proyecto que debe estar compuesto por representantes del departamento informático y de los departamentos usuarios del Data Warehouse, además de la figura de jefe de proyecto. Se definirá el alcance del sistema y cuáles son las funciones que el Data Warehouse realizará como suministrador de información de negocio estratégica para la empresa. Se definirán así mismo, los parámetros que permitan evaluar el éxito del proyecto.

3.4.2.1.1.2 Definición de los requerimientos de información

Durante esta fase se mantendrán sucesivas entrevistas entre los representantes del departamento usuario final y los representantes del departamento de informática. Se realizará el estudio de los sistemas de información existentes, que ayudarán a comprender las carencias actuales y futuras que deben ser resueltas en el diseño del Data Warehouse. Asimismo, en esta fase el equipo de proyecto debe ser capaz de validar el proceso de entrevistas y reforzar la orientación de negocio del proyecto. Al finalizar esta fase se obtendrá el documento de definición de requerimientos en el que se reflejarán no sólo las necesidades de información de los usuarios, sino cuál será la estrategia y arquitectura de implantación del Data Warehouse.

3.4.2.1.1.3 Diseño y modelización

Los requerimientos de información identificados durante la anterior fase proporcionarán las bases para realizar el diseño y la modelización del Data Warehouse. En esta fase se identificarán las fuentes de los datos (sistema operacional, fuentes externas, etc.) y las transformaciones necesarias para, a partir de dichas fuentes, obtener el modelo lógico de datos del Data Warehouse. Este modelo estará formado por entidades y relaciones que permitirán resolver las necesidades de negocio de la organización.

3.4.2.1.1.4 Implementación

La implantación de un Data Warehouse lleva implícitos los siguientes pasos:

- Extracción de los datos del sistema operacional y transformación de los mismos.
- Carga de los datos validados en el Data Warehouse. Esta carga deberá ser planificada con una periodicidad que se adaptará a las necesidades de refresco detectadas durante la fase de diseño del nuevo sistema.
- Explotación del Data Warehouse mediante diversas técnicas dependiendo del tipo de aplicación que se dé a los datos. Entre las técnicas más habituales podemos encontrar las siguientes:
 - Query & Reporting
 - On-line analytical processing (OLAP)
 - Executive Information System (EIS) o Información de gestión
 - Decision Support Systems (DSS)
 - Visualización de la información

3.4.2.1.1.5 Revisión

La construcción del Data Warehouse no finaliza con la implantación del mismo, sino que es una tarea iterativa en la que se trata de incrementar su alcance aprendiendo de las experiencias anteriores. Después de implantarse, debería realizarse una revisión del Data Warehouse planteando preguntas que permitan, después de los seis o nueve meses posteriores a su puesta en marcha, definir cuáles serían los aspectos a mejorar o potenciar en función de la utilización que se haga del nuevo sistema.

3.4.2.1.1.6 Gestión del Proyecto

La gestión del proyecto debe encargarse de la coordinación y ejecución de las distintas fases que conforman la construcción e implantación de un Data Warehouse. Este proceso se tiene que apoyar en una metodología específica para este tipo de trabajos, si bien es más importante que la elección de la mejor de las metodologías, el realizar un control para asegurar el seguimiento de la misma. En las fases que se establezcan es fundamental incluir una fase de formación en la herramienta utilizada, para un máximo aprovechamiento de la aplicación. Seguir los pasos de la metodología y comenzar el Data Warehouse por un área específica de la empresa permitirá obtener resultados tangibles en un corto espacio de tiempo.

3.4.2.1.2 Metodología Kimball – Ciclo de Vida [18]

La Figura 15 muestra de forma esquemática las fases que componen la metodología propuesta por Kimball y las siguientes secciones resumen el contenido de cada una de las fases.

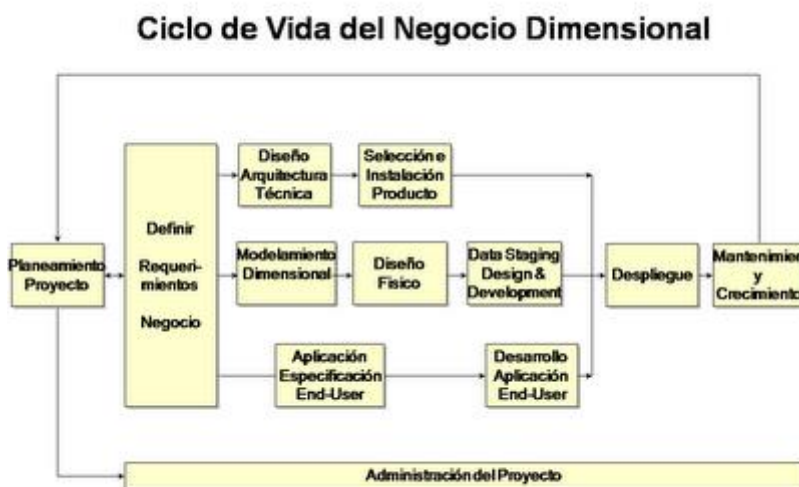


Figura 15: Metodología Kimball – Ciclo de Vida [18]

3.4.2.1.2.1 Planificación del Proyecto

La planificación busca identificar la definición y el alcance del proyecto de Data Warehouse, incluyendo las justificaciones del negocio y las evaluaciones de factibilidad. Esta etapa se concentra sobre la definición del proyecto. Según sentencia Kimball [18]:

“Antes de comenzar un proyecto de Data Warehouse o DataMart, hay que estar seguro si existe la demanda y de dónde proviene. Si no se tiene un usuario sólido, posponga el proyecto”.

Como metodología, en esta etapa propone identificar el alcance preliminar basándose en los requerimientos del negocio y no en fechas límites, construyendo la justificación del proyecto en términos del negocio. A nivel de planificación del proyecto se establece la identidad del mismo, el personal (los usuarios, gerentes del proyecto, equipo del proyecto), desarrollo del plan del proyecto, el seguimiento y la monitorización.

3.4.2.1.2.2 Definición de los Requerimientos del Negocio

Un factor determinante en el éxito de un proceso de Data Warehouse es la interpretación correcta de los diferentes niveles de requerimientos expresados por los distintos grupos de usuarios. La técnica utilizada para revelar los requerimientos de los analistas del negocio difiere de los enfoques tradicionales guiados por los datos. Los diseñadores de los Data Warehouse deben entender los factores claves que guían el negocio para determinar efectivamente los requerimientos y traducirlos en consideraciones de diseño apropiadas. Los usuarios finales y sus requerimientos impactan siempre en la implementación de un Data Warehouse. Según la perspectiva de Kimball [18], los requerimientos del negocio se posicionan en el centro del “universo del Data Warehouse”. Como destaca siempre el autor, los requerimientos del negocio deben determinar el alcance del Data Warehouse (qué datos debe contener, cómo deben estar organizados, cada cuánto tiempo debe actualizarse, quiénes y desde dónde accederán, etc.).

3.4.2.1.2.3 Modelado Dimensional

La definición de los requerimientos del negocio determina los datos necesarios para cumplir los requerimientos analíticos de los usuarios. Diseñar los modelos de datos para soportar estos análisis requiere un enfoque diferente al usado en los sistemas operacionales. Básicamente, se comienza con una matriz donde se determina la dimensionalidad de cada indicador y luego se especifican los diferentes grados de detalle dentro de cada concepto del negocio, así como la granularidad de cada indicador y las diferentes jerarquías que dan forma al Modelo Dimensional del Negocio (MDN) o mapa dimensional.

3.4.2.1.2.4 Diseño Físico

El diseño físico de la base de datos se focaliza sobre la selección de las estructuras necesarias para soportar el diseño lógico. Un elemento principal de este proceso es la definición de estándares del entorno de la base de datos. La indexación y las estrategias de particionamiento se determinan también en esta etapa. En la estrategia de particionamiento o agregación, el Data Warehouse tiene, y debe tener, todo el detalle de información en su nivel atómico. Así, por poner algún ejemplo, en los sectores de telecomunicaciones o banca es habitual encontrarse con Data Warehouse con miles de millones de registros. Sin embargo, la mayoría de consultas no necesitan acceder a un nivel de detalle demasiado profundo.

3.4.2.1.2.5 Diseño y Desarrollo de la Presentación de Datos

Esta etapa es típicamente la más subestimada de las tareas en un proyecto de Data Warehouse. Las principales actividades de esta fase del ciclo de vida son: la extracción, la transformación y la carga (ETL). Se definen como procesos de extracción aquellos requeridos para obtener los datos que permitirán efectuar la carga del Modelo Físico diseñado. Así mismo, se definen como procesos de transformación los procesos para

convertir o recodificar los datos fuente a fin de poder efectuar la carga efectiva del Modelo Físico. Por otra parte, los procesos de carga de datos son los procesos requeridos para poblar el Data Warehouse. Todas estas tareas son altamente críticas pues tienen que ver con la materia prima del Data Warehouse: los datos. La desconfianza y pérdida de credibilidad del Data Warehouse provocará efectos inmediatos e inevitables si el usuario se encuentra con información inconsistente. Es por ello que la calidad de los datos es un factor determinante en el éxito de un proyecto de Data Warehouse. Es en esta etapa donde deben sanearse todos los inconvenientes relacionados con la calidad de los datos fuente. Para cumplir con estas premisas es necesario tener en cuenta ciertos parámetros a la hora de desarrollar las tablas de dimensión y la tabla de hechos.

3.4.2.1.2.6 Diseño de la Arquitectura Técnica

Los entornos de Data Warehouse requieren la integración de numerosas tecnologías. Se deben tener en cuenta tres factores: los requerimientos del negocio, los actuales entornos técnicos y las directrices técnicas y estrategias futuras planificadas por la compañía para poder establecer el diseño de la arquitectura técnica del entorno de Data Warehouse.

Algunos equipos de trabajo no entienden las ventajas de una arquitectura y tienen la sensación de que las tareas son demasiado opacas, por lo que entienden su diseño como una distracción y un obstáculo para el progreso del Data Warehouse, así que optan por omitir el diseño de la arquitectura. Sin embargo, hay otros equipos de trabajo que dedican un tiempo demasiado grande para el diseño arquitectónico. El autor Ralph Kimball [18] recomienda no irse a ninguno de los dos extremos para hacerlo de una manera intermedia. Para ello propone un proceso de 8 pasos para asegurar un correcto diseño arquitectónico sin extenderse demasiado en el tiempo.

1. Establecer un Grupo de Trabajo de Arquitectura
2. Establecer requisitos relacionados con la arquitectura
3. Establecer documento de requisitos arquitectónicos
4. Desarrollar de un modelo arquitectónico de alto nivel

5. Diseñar y especificar los subsistemas
6. Determinar las fases de aplicación de la Arquitectura
7. Documentar la Arquitectura Técnica
8. Revisar y finalizar la Arquitectura Técnica

El plan de la arquitectura se debe comunicar con diferentes niveles de detalle: equipo de proyecto, patrocinador y director del proyecto. Tras la revisión, la documentación debe ser actualizada y utilizada inmediatamente en el proceso de selección del producto.

3.4.2.1.2.7 Selección de Productos e Instalación

Utilizando el diseño de arquitectura técnica como marco es necesario evaluar y seleccionar los componentes específicos de la arquitectura, como la plataforma de hardware, el motor de base de datos, la herramienta de ETL, las herramientas de acceso, etc. Una vez evaluados y seleccionados los componentes determinados se procede con la instalación y prueba de los mismos en un ambiente integrado de Data Warehouse. Para ello es necesario tener en cuenta una serie de puntos a considerar que son recomendados por el autor de esta metodología:

- Comprender el proceso de compras corporativas.
- Elaborar una matriz de evaluación del producto.
- Realizar investigación de mercados.
- Filtrar opciones y realizar evaluaciones más detalladas.
- Manejar un prototipo.
- Seleccionar el producto, instalar y negociar.

3.4.2.1.2.8 Especificación de Aplicaciones para Usuarios Finales

No todos los usuarios del Data Warehouse necesitan el mismo nivel de análisis. Es por ello que en esta etapa se identifican los roles o perfiles de usuarios para los diferentes tipos de aplicaciones necesarias en base al alcance de los perfiles detectados (gerencial, analista del negocio, vendedor, etc.)

3.4.2.1.2.9 Desarrollo de Aplicaciones para Usuarios Finales

A continuación del punto anterior, el desarrollo de las aplicaciones de los usuarios finales involucra configuraciones de los metadatos y construcción de reportes específicos. Los usuarios acceden al Data Warehouse por medio de herramientas de productividad basadas en GUI (Graphical User Interface). De hecho existen multitud de estas herramientas con las que proveer a los usuarios. Las herramientas pueden incluir software de consultas, generadores de reportes, procesamiento analítico en línea o herramientas de Datamining dependiendo de los tipos de usuarios y sus requerimientos particulares. Sin embargo, una sola herramienta puede no satisfacer todos los requerimientos, por lo que quizás sea necesario la integración de herramientas hechas bajo petición expresa de los usuarios para satisfacer sus necesidades de consulta sobre el Data Warehouse.

3.4.2.1.2.10 Implementación

La implementación representa la convergencia de la tecnología, los datos y las aplicaciones de usuarios finales accesibles para el usuario del negocio. Hay varios factores extras que aseguran el correcto funcionamiento de todos estos elementos, entre ellos se encuentran la capacitación, el soporte técnico, la comunicación y las estrategias de feedback. Todas estas tareas deben tenerse en cuenta antes de que cualquier usuario pueda tener acceso al Data Warehouse.

3.4.2.1.2.11 Mantenimiento y crecimiento

Como se remarca siempre, la creación de un Data Warehouse es un proceso (de etapas bien definidas, con comienzo y fin, pero de naturaleza espiral) que acompaña a la evolución de la organización durante toda su historia. Se necesita continuar con las actualizaciones de forma constante para poder seguir la evolución de las metas por conseguir. Al contrario de los sistemas tradicionales, los cambios en el desarrollo deben ser vistos como signos de éxito. Es importante establecer las prioridades para poder manejar los nuevos requerimientos de los usuarios y de esa forma poder evolucionar y crecer.

Una vez que se ha construido e implantado el Data Warehouse es necesario administrar su crecimiento y mantenimiento. Si bien las tareas pueden llegar a parecer similares a las tratadas en otras etapas del ciclo de vida, existe una diferencia clave: los usuarios están ahora accediendo al Data Warehouse.

3.4.2.1.2.12 Gestión del Proyecto

La gestión del proyecto asegura que las actividades del ciclo de vida se lleven a cabo de manera sincronizada. Entre sus actividades principales se encuentra la monitorización del estado del proyecto y el acoplamiento entre los requerimientos del negocio y las restricciones de los sistemas de información para poder manejar correctamente las expectativas en ambos sentidos

3.4.3 Comparativa de resumen

La Tabla 1 presenta un resumen comparativo de los diferentes tipos de modelos con sus respectivas metodologías, con el objetivo de obtener una visión más general.

Modelo	Metodologías	Ventajas	Características
Top-Down	Hefesto	Permite la construcción de Data Warehouse de forma sencilla, ordenada e intuitiva	<ul style="list-style-type: none"> • Objetivos y resultados de fácil comprensión • Estructura adaptable • Reduce la resistencia al cambio • Utiliza modelos sencillos de interpretar y analizar. • Es independiente del tipo de ciclo de vida, de las herramientas, estructuras físicas que se utilicen en la implementación. • Es secuencial en su desarrollo.
	Inmon	Se aborda la totalidad del problema para luego llegar a los detalles	<ul style="list-style-type: none"> • El ambiente de origen y de acceso de los datos, debe estar físicamente separado en diferentes bases de datos, y en equipos separados. • Se aplica a grandes volúmenes de datos, de nivel empresarial. • Se involucra desde un inicio todo el ámbito corporativo • Evita la aparición de situaciones inesperadas que puedan poner en peligro el proyecto.
Bottom-Up	Rapid Warehousing	Es iterativo y está basado en el desarrollo incremental del proyecto	<ul style="list-style-type: none"> • Establece el equipo del proyecto en conjunto con los parámetros de evaluación del proyecto. • Realiza un estudio de los sistemas de información existentes, para comprender las carencias actuales y futuras. • Realiza el diseño y la modelización del Data Warehouse en base a los

			<p>requerimientos.</p> <ul style="list-style-type: none"> • Luego de la implantación del sistema se realizan revisiones iterativas.
	Kimball – Ciclo de Vida	Es una metodología con un Ciclo de Vida más estructurado	<ul style="list-style-type: none"> • Se basa en los requerimientos de negocio no en fechas límites. • Evalúa la factibilidad del proyecto. • Posee diferentes niveles de requerimientos. • Utiliza un enfoque diferente al usado en los sistemas operacionales. • Define estándares del entorno de la base de datos. • Diseña la arquitectura técnica del entorno del Data Warehouse. • Es una metodología de tipo espiral.

Tabla 1: Comparativa de Métodos y Metodologías

Capítulo 4

Entorno de la Aplicación

En este capítulo se presentará la empresa Rabie S.A., definiendo sus áreas relacionadas con el sistema Data Warehouse, abordando el tema, como su situación actual, para definir las necesidades reales y actuales que tiene esta empresa con relación al problema planteado. Además de ver la forma en que el sistema pueda ser implementado y señalar la metodología a utilizar en conjunto con los objetivos de esta memoria de título.

4.1 Situación actual

Rabie S.A. corresponde a una empresa de nivel nacional, siendo una de las más importantes en el ámbito de Distribución de Productos al Comercio Minorista y Mayorista, teniendo como objetivo ser la solución logística más conveniente del país, tanto para los fabricantes como para los comerciantes. Permitiendo así a los fabricantes ubicar sus productos en cualquier lugar de Chile, y que los comerciantes tengan un abastecimiento seguro, confiable y con posibilidades de competir sin importar el lugar geográfico en el que se encuentren.

Para ello Rabie S.A. cuenta con un control de gestión compuesto por dos tipos de clientes, los cuales corresponden al cliente externo y al cliente interno como se mencionaba en capítulos anteriores, siendo el último quién está en constante acceso a la información de la empresa. El cliente interno está compuesto por cuatro áreas:

- Soporte de Información

Esta área está encargada de generar, validar y liberar la información relacionada con la compañía. Y está compuesta por 5 personas.

- Gestión comercial

Esta área maneja precios de producto, márgenes y revisa los ingresos, estando compuesta por 3 personas.

- Gestión Ventas

Esta área trata con información de vendedores, indicadores decisivos para venta, históricos de venta, históricos de cobertura, entre otros, y está compuesta por 3 personas.

- Gestión Logística

Esta área contiene el manejo operacional de bodegas, cómo responden los proveedores, tiempos de demora que toma la entrega de productos a repartir. Y está compuesta por 2 personas.

En general, se puede decir que Rabie S.A. está organizada en una serie de unidades administrativas (según se puede visualizar en la Figura 16) y cuenta con alrededor de 950 empleados.

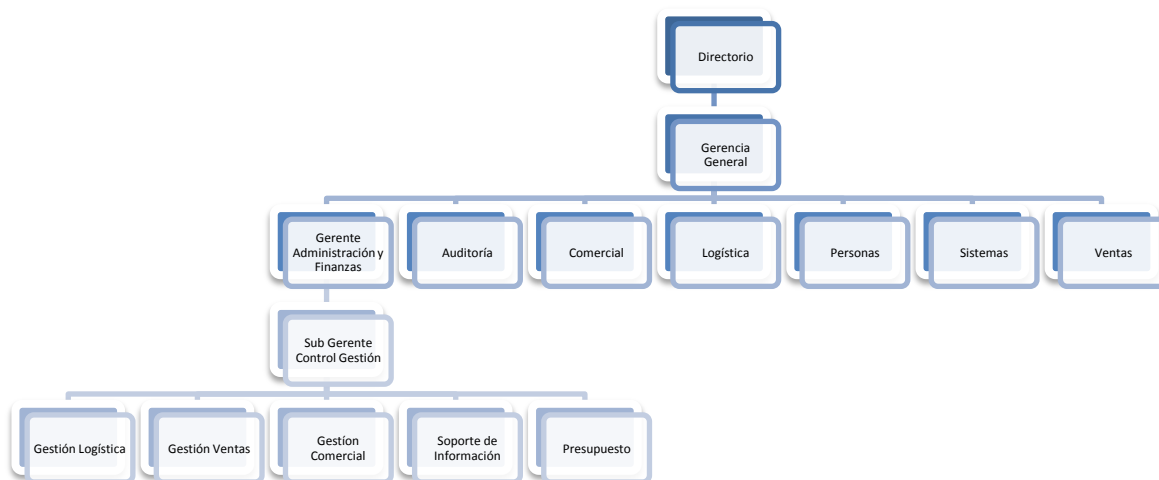


Figura 16: Organigrama Rabie S.A.

4.1.1 Área de la empresa a abordar

El área a analizar de la empresa Rabie S.A. será Soporte de Información quien necesita optimizar la entrega de información que provee diariamente a las áreas de Gestión Comercial, Gestión Logística, Gestión Ventas y Presupuesto, esto se debe a dos razones; primero, el procesamiento de los datos se realiza de forma manual incluyendo la generación de reportes e informes, además esto se ve involucrado por la carga diaria de los datos, ya que en la actualización, éstos son remplazados por los datos nuevos, lo que exige una mayor carga en el sistema, afectando directamente al rendimiento con respecto a consultas realizadas ya sea para generar informes o por consultas personalizadas. Actualmente el área procesa alrededor de 100 Gb de información diarios los que mantienen una carga constante en el sistema, afectando hasta la consulta más simple.

En segundo lugar, los datos se almacenan de forma histórica en distintas tablas siendo éstas mensuales y diarias, aumentando considerablemente el volumen de los mismos, por lo que podemos encontrar distintas tablas mensuales generadas a partir del año 2010, de las cuales su procesamiento en conjunto produce una saturación del sistema debido al gran volumen de datos que contiene cada una de ellas. Dificultando de esta forma que las consultas sean realizadas correcta y eficientemente.

Además, cabe destacar que algunos de los datos no se pueden extraer directamente de la base de datos, por lo que el área de Soporte de Información debe responder a estas solicitudes de manera personalizada, realizando un procesamiento del dato para su entrega. Esto también se ve afectado por el nivel de preparación de los usuarios finales, ya que no poseen los conocimientos sobre los procesos de extracción de los datos, viéndose esto reflejado en el tipo de consultas que realizan.

Todo lo anterior dificulta generar la información relevante y oportuna con respecto a diferentes dimensiones que permiten un análisis de mayor calidad de la información, como lo son las dimensiones de tiempo, geografía, productos o ventas entre otras, lo cual permite realizar análisis de distintas perspectivas del negocio, información que los directivos requieren para apoyar su proceso de toma de decisiones, aumentando de esta forma las expectativas de la empresa. Este tipo de análisis se facilita a través de herramientas

montadas sobre un entorno de un Data Warehouse, de tal forma de poder dar un mejor uso y obtener un mayor manejo de los datos para que estos entreguen información valiosa que posteriormente sea transformada en conocimiento.

Debido a que Rabie S.A. ya cuenta con sistemas operacionales aplicados a sus bases de datos, es factible la implementación de un sistema Data Warehouse, lo que permite en gran medida abordar el problema de una manera más sencilla y exitosa.

Para desarrollar el Data Warehouse en el área de Soporte de Información, Rabie S.A., se eligió la Metodología propuesta por Hefesto en el Capítulo 3, la cual explica la formación del modelo multidimensional a partir de definir los esquemas de hechos, los cuales darán inicio al desarrollo del sistema, para luego formar las dimensiones con sus respectivos atributos. Se seleccionó esta metodología debido a que permite la construcción de Data Warehouse de forma sencilla, ordenada e intuitiva, además que el área de Soporte de Información desconoce los procedimientos y el funcionamiento de un Data Warehouse por lo que es fundamental utilizar una metodología que ofrece un mayor detalle de la implementación del sistema.

En el caso del área de Soporte de Información, los desarrolladores están en constante trabajo, lo que dificulta utilizar una metodología que mantenga la totalidad del tiempo en el desarrollo del Data Warehouse, por ende se eligió una metodología que puede adaptarse muy bien a cualquier ciclo de vida de desarrollo de software, de esta forma no se seleccionaron metodologías que requieran fases extensas de reunión de requerimientos y análisis, fases de desarrollo monolítico que conlleve demasiado tiempo y fases de despliegue muy largas. Permitiendo entregar una primera implementación que satisfaga una parte de las necesidades, para demostrar las ventajas del Data Warehouse y motivar a los usuarios gerenciales o encargados de la toma de decisiones.

4.2 Forma de abordar el problema

La solución propuesta es la implementación de un Data Warehouse, el cual reducirá la carga en el sistema, permitiendo realizar consultas automatizadas con un menor tiempo de respuesta. Los datos estarán almacenados históricamente en sus respectivas tablas, esto quiere decir que cada tabla tendrá la totalidad de los datos, no estando éstos repartidos en distintas tablas como ocurre actualmente, lo anterior facilitará su procesamiento, permitiendo obtener información valiosa de manera ágil. Además, permitirá a la organización, si lo desea, modificar su sistema de gestión de base de datos, apuntando al uso de tecnologías como Minería de Datos, entre otras. A través de la implementación de un Data Warehouse, se debe satisfacer como primera instancia el área de Soporte de Información de esa forma se podrán satisfacer las otras áreas que dependen de esta, siendo las áreas de Gestión Comercial, Gestión Logística, Gestión Ventas y Presupuesto. Cabe destacar que algunas de estas áreas físicamente se encuentran alejadas del Área Soporte de Información, lo que provoca que al existir alguna falla no puedan satisfacer consultas importantes directamente con ésta.

4.3 Objetivo estratégico

Debido a que la existencia de bases de datos transaccionales centralizadas en donde se almacena la información operacional de Soporte de Información, Rabie S.A., se ha optado por la creación de un Data Warehouse Centralizado o Central, capaz de rescatar dicha información de una manera rápida y una baja utilización de recursos del sistema.

Junto con lo anterior, el universo de usuarios que accederá al Data Warehouse estará acotado por el área de Soporte de Información, ya que son el área encargada de la generación de reportes, e informes diarios y mensuales, destinados a las áreas de Gestión Comercial, Gestión Logística, Gestión Ventas y Presupuesto, lo que facilita la distribución de éste, en cuanto a la carga de datos operacionales sobre las bases de datos transaccionales que realizan sus ingresos a través de sistemas centralizados a lo largo de todo el país.

Capítulo 5

Diseño de un Data Warehouse

5.1 Análisis de Requerimientos

Mediante entrevistas con el personal de Soporte de Información, se pudieron identificar las necesidades actuales de la empresa que en parte se pretende dar solución con la implementación de un Data Warehouse en cuanto a resultados consistentes y utilizables.

5.1.1.1 *Requerimientos Generales*

1. Entregar la información que permita generar la totalidad o parte de la carga automática de los principales informes del área:

Informes Diarios

- Los informes operacionales
 - **Quiebres**

Muestra las cantidades y valores por productos que la empresa no fue capaz de abastecer a la demanda de los usuarios. A este informe tiene acceso las áreas de Logística y Ventas. Se pretende cubrir con el Data Warehouse parte del informe, en la sección correspondiente a Ventas.
 - **Informe Cobertura**

Corresponde a la revisión de las cantidades de clientes atendidos, por zona y por proveedor, teniendo acceso a esto solo el área de Ventas, a lo cual pretende cubrir el Data Warehouse la totalidad del informe.

- **GenJef Ventas Productos**

Informe maestro con ventas históricas, con apertura a nivel de productos, lo que corresponde desde proveedor, categoría y producto, teniendo acceso a este informe sólo el área comercial. Lo que se pretende cubrir con el Data Warehouse, es casi la totalidad del informe.

Informes Mensuales

- **Cubo de venta**

Está orientado principalmente a los productos, vendedores y clientes, siendo esta información abierta, la cual se pretende cubrir con el Data Warehouse.

- **Cubo de venta tradicional**

Filtra solamente el canal tradicional que corresponde a la venta no realizada por los vendedores especialistas de Proyecto. Cubriéndose en su totalidad por el Data Warehouse.

- **Maestra Rentabilidad Comercial**

Informe que muestra la rentabilidad por proveedor, categoría y producto de forma histórica, información solo para el Departamento de Gestión, la cual se pretende cubrir en su totalidad con el Data Warehouse.

- **Sellout Comercial**

Corresponde al resumen de la venta por producto del mes, información de carácter publica, la cual se pretende cubrir con el Data Warehouse.

2. Debe proveer la base sobre la que estarán montados los cubos de información de cada área, esto se refiere a que se utilizará como soporte para la generación de los informes mencionados en el punto anterior.
3. El Data Warehouse debe conectarse a las bases de datos actuales, permitiendo de esta forma la actualización automatizada de los datos existentes en ellas.

4. Obtener información histórica, con respecto a los datos almacenados actualmente en la base de datos en ubicaciones diferentes.
5. Permitir a los usuarios finales poder realizar consultas dinámicas, de forma particular.
6. Disminuir el tiempo de respuesta, a consultas personalizadas por parte de usuarios.
7. Disminuir el tiempo dedicado por parte del personal de Soporte de Información a la generación de consultas solicitadas por usuarios recurrentemente, esto permite que el personal de soporte pueda realizar otras actividades.

Con la idea de satisfacer los puntos presentados anteriormente, se realizará un análisis detallado de requisitos a través de preguntas que expliciten los objetivos de la organización. Luego, se analizarán éstas preguntas para identificar cuáles serán los indicadores y perspectivas que se utilizarán en la construcción del Data Warehouse. Y para finalizar, se confeccionará un diagrama conceptual que permite visualizar el resultado obtenido en este primer paso.

5.1.1.2 Requerimientos específicos

El área de Soporte de Información da respuesta a solicitudes o reportes que son solicitados de forma diaria y mensual, por lo que es relevante para ellos suplir ciertos campos de información que contienen estos reportes con los datos entregados por el Data Warehouse. Además se señala que todas las consultas generadas por los reportes e informes acceden a las tablas Maestras Finales que posee el área de Soporte de Información, las cuales contienen información procesada de venta, detalles comerciales y logísticos, donde se han aplicado una serie de criterios desarrollados por las áreas de gestión. Estas tablas se diferencian unas de otras debido a que están organizadas mensualmente.

Por lo que a continuación se presentan más detalladamente las solicitudes e informes que acceden a los datos que estarán incluidos en el Data Warehouse, dando también a conocer los tiempos que toma actualmente la consulta.

5.1.1.2.1 Solicitudes (Reportes)

5.1.1.2.1.1 Venta neta de producto

Este reporte contiene información de las ventas netas generadas por algún producto en particular, los datos dependen directamente de la necesidad de información, en otras palabras, se genera a solicitud del cliente interno. Con el objetivo de obtener el monto total de ventas por producto.

Este reporte posee las siguientes características anexas:

- Ingresa a la tabla Maestra_Final_MesAño.
- Se generan 13 reportes mensuales aproximadamente.
- Duración de la consulta entre 10 a 30 minutos aproximadamente.

5.1.1.2.1.2 Venta en unidades de productos

Este reporte contiene las ventas en unidades de productos, esto quiere decir que muestra información de las cantidades de uno o varios productos vendidos. De tal forma obtener el total de unidades vendidas por producto.

Este reporte posee las siguientes características anexas:

- Ingresa a la tabla Maestra_Final_MesAño.
- Se generan 12 reportes mensuales aproximadamente
- Duración de la consulta de 10 a 30 minutos aproximadamente.

5.1.1.2.1.3 Cobertura de clientes

Este reporte contiene información de la zona geográfica de donde corresponden los clientes que realizan compras a la empresa. Identificando de esta forma el sector y el cliente relacionado a la venta.

Este reporte posee las siguientes características anexas:

- Ingresa a la tabla Maestra_Final_MesAño.
- Se generan 15 reportes mensuales aproximadamente.
- Duración de la consulta de 15 a 35 minutos aproximadamente.

5.1.1.2.1.4 Venta por vendedor

Este reporte muestra en detalle las ventas realizadas por cada vendedor. Información que se genera a pedido. De tal forma obtener el monto total de ventas por vendedor.

Este reporte posee las siguientes características anexas:

- Ingresa a la tabla Maestra_Final_MesAño.
- Se generan 20 reportes mensuales aproximadamente.
- Duración de la consulta de 10 a 30 minutos aproximadamente.

5.1.1.2.1.5 Venta por zona

Este reporte contiene información de los montos de ventas por zonas, las que corresponden a los diferentes grupos geográficos al cual pertenece un vendedor. Con el objetivo de generar el monto total de ventas por zona.

Este reporte posee las siguientes características:

- Ingresa a la tabla Maestra_Final_MesAño.

- Se generan 10 reportes mensuales aproximadamente.
- Duración de la consulta es de 15 minutos aproximadamente.

5.1.2 Identificación de Preguntas

En este paso se recopilan las necesidades de información, lo cual se llevó a cabo a través de entrevistas, cuestionarios y observaciones.

El análisis de los requerimientos de los diferentes usuarios y administradores, es el punto de partida de la metodología de HEFESTO, ya que en cierto modo permite guiar la investigación hacia un desarrollo que refleje claramente lo que se espera del Data Warehouse, en relación a sus funciones y cualidades.

El objetivo principal de esta fase, es la de obtener e identificar las necesidades de información clave de alto nivel, que es esencial para llevar a cabo las metas y estrategias de la empresa, y que facilitará una eficaz y eficiente toma de decisiones.

Debe tenerse en cuenta que dicha información, es la que proveerá el soporte para desarrollar los pasos sucesivos, por lo cual, es muy importante que se preste especial atención al relevar los datos [1].

Para ello, se entrevistó a los principales usuarios en busca de sus necesidades de información, identificando de esta forma la actividad principal de Rabie y del área de Soporte de Información, siendo éste “Ventas”, ya que diariamente se realizan aproximadamente un 85% de consultas con respecto a esta área.

Luego se procedió a identificar las necesidades más relevantes con respecto a este proceso y además de las variables o perspectivas que deben tenerse en cuenta para poder tomar decisiones basadas en él.

A posterior se identificaron cuáles eran los indicadores que representan de mejor modo el proceso de Ventas y qué es exactamente lo que se desea analizar del mismo. A modo de respuesta se obtuvo que era relevante conocer las ganancias, los montos de ventas

y sus respectivos costos, además de los kilos vendidos; logrando de esta forma obtener las dimensiones desde las cuales se consultarán dichos indicadores. Cabe señalar que una dimensión de base de datos es una colección de objetos relacionados, denominados atributos, que se pueden usar para proporcionar información sobre los datos de hechos de uno o varios cubos multidimensionales. Estos objetos están enlazados a una o varias columnas de una o varias tablas de una vista de origen de datos. De manera predeterminada, estos atributos están visibles como jerarquías de atributo y se pueden utilizar para comprender los datos de hechos en un cubo multidimensional.

Para simplificar esta tarea se les presentó una serie de ejemplos concretos de otros casos similares.

El resultado por área obtenido fue el siguiente:

Área de Venta:

1. Se desea conocer cuál fue el monto de las ventas de categorías de productos por zona en un tiempo determinado.
2. Se desea conocer cuál fue el monto de las ventas de las zonas por vendedor en un tiempo determinado.
3. Se desea conocer cuál fue el monto de las ventas, el monto total de costos y la ganancia por sector de la compra en un tiempo determinado.

Área Comercial:

4. Se desea conocer cuál fue el monto de las ventas, el monto de los costos y la ganancia de productos por vendedor en un tiempo determinado.
5. Se desea conocer cuál fue el monto de las ventas, el monto de los costos y la ganancia de categorías de productos por zona en un tiempo determinado.

Área Logística:

6. Se desea conocer la cantidad de mt3 utilizados en las categorías de productos en un tiempo determinado

7. Se desea conocer la cantidad de kilos presentes en las categorías de productos en un tiempo determinado
8. Se desea conocer la cantidad de unidades vendidas de los productos en un tiempo determinado.

Cabe señalar que se dio un mayor énfasis a la incorporación de la dimensión Tiempo, además, a través de la exposición a los usuarios de ejemplos prácticos, se pudo formular las consultas antes mencionadas, con el objetivo de obtener resultados que permitan un correcto análisis posterior. Permitiendo así generar un nuevo ámbito para la toma de decisiones de la organización.

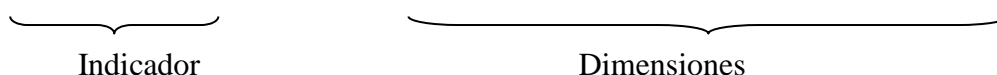
5.1.3 Identificación de indicadores y dimensiones o perspectivas de análisis

Luego de haber establecido las preguntas claves propuestas en el punto anterior, se procede a su descomposición con el objeto de identificar los indicadores que se utilizarán y las perspectivas de análisis.

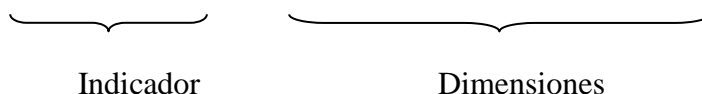
A continuación, se analizarán las preguntas obtenidas en el paso anterior y se detallarán cuáles son sus respectivos indicadores y dimensiones.

Área de Venta:

1. Monto de ventas de categorías de productos por zona en un tiempo determinado.



2. Monto de ventas de las zonas por vendedor en un tiempo determinado.



3. Monto de ventas, el monto de costos y la ganancia categorías de

Indicador

productos por lugar de compra en un tiempo determinado.

Dimensiones

Área Comercial:

4. Monto de ventas, el monto de costo y la ganancia de

Indicadores

productos por vendedor en un tiempo determinado.

Dimensiones

5. Monto de ventas, el monto de costos y la ganancia de categorías de

Indicadores

productos por zona en un tiempo determinado.

Dimensiones

Área Logística:

6. Cantidad de mt3 utilizados en los productos en un tiempo determinado

Indicador

Dimensiones

7. Cantidad de kilos presentes en los productos en un tiempo determinado

Indicador

Dimensiones

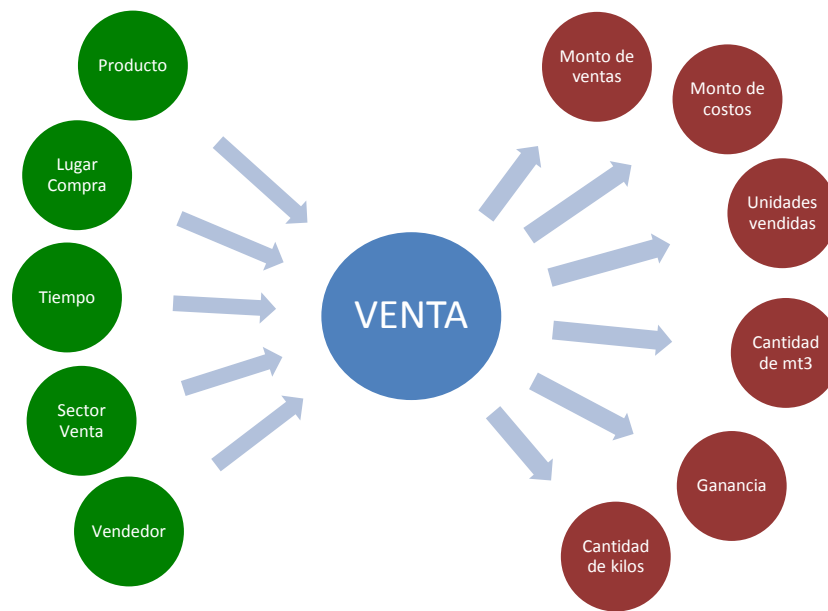


Figura 17: Mapa Conceptual

A modo de explicación, se puede observar a la izquierda las dimensiones seleccionadas, unidas a un óvalo central que representa y lleva el nombre de la relación que existe entre ellas, la cual corresponde a la Venta. La relación, constituye el proceso o área de estudio elegida en el proyecto, desprendiéndose de ésta los indicadores, que se ubican a la derecha del esquema.

5.2 Análisis de los procesos OLTP

Se identificarán las fuentes que alimentarán el Data Warehouse, para ello se tomará como referencia los procesos OLTP (On Line Transaction Processing), los cuales representan toda aquella información transaccional que genera la empresa en su accionar diario, además, de las fuentes externas con las que puede llegar a disponer.

Como ya se ha mencionado, estas fuentes de información, son de características muy disímiles entre sí, en formato, procedencia, función, etc.

Entre los OLTP más habituales que pueden existir en cualquier organización se encuentran:

- Archivos de textos.
- Hipertextos.
- Hojas de cálculos.
- Informes semanales, mensuales, anuales, etc.
- Bases de datos transaccionales.

Actualmente no existen registros sobre algún documento que contenga los modelos relacionales, ni tampoco un registro de las bases de datos actuales implementadas, esto se debe a la gran cantidad de tablas existentes, aproximadamente 250 tablas.

Por lo que la extracción de los datos se realizará sobre las Tablas Maestras existentes, las que contienen una recopilación de los atributos necesarios ya procesados, siendo estos datos extraídos de las tablas que se generan en las distintas áreas donde se actualizan los datos diariamente. Por lo que estas tablas siempre estarán actualizadas con datos consistentes.

5.2.1 Determinación de Indicadores

En este paso se definirán los cálculos a realizar para obtener los indicadores, como se especifica a continuación:

- “Monto de ventas”:
 - Hechos: Venta
 - Función de sumarización: SUM.

Aclaración: el indicador “Monto total de ventas” representa la sumatoria de las ventas de un producto en particular.

- “Monto de costos”:
 - Hechos: Costo

- Función de sumarización: SUM.

Aclaración: el indicador “Monto total de costos” representa la sumatoria de los costos de la adquisición de un producto en particular.

- Ganancia

- Hechos: $(\text{Venta}) - (\text{Costo})$
- Función de sumarización: SUM.

Aclaración: el indicador “Ganancia” representa la sumatoria de las ganancias que generó la venta de cada producto, y se obtiene al restar la venta neta, menos su respectivo costo.

- Unidades vendidas.

- Hechos: Unidades
- Función de sumarización: SUM.

Aclaración: el indicador “Unidades vendidas” representa la sumatoria de las unidades de un producto en particular en una venta.

- Cantidad de mt3

- Hechos: $(\text{Volumen}) * (\text{Unidades})$
- Función de sumarización: SUM.

Aclaración: el indicador “Cantidad de mt3” representa la sumatoria de los mt3 multiplicados por las unidades correspondientes a la venta de un producto en particular.

- Cantidad de kilos

- Hechos: $(\text{Weight}) * (\text{Unidades})$
- Función de sumarización: SUM.

Aclaración: el indicador “Cantidad de kilos” representa la sumatoria de los kilos multiplicados por las unidades correspondientes a la venta de un producto en particular en una venta.

5.2.2 Definición de relaciones

Se procede a la examinación los OLTP disponibles que contengan la información requerida, como así también sus características, para poder identificar las relaciones entre el diagrama conceptual y las fuentes de datos.

La idea es que todos los elementos del diagrama conceptual estén correspondidos en los OLTP.

5.2.2.1 Fuentes Internas

Corresponden a las fuentes internas que actualmente satisfacen casi la totalidad de los requerimientos de los usuarios, esto permitirá evaluar los campos relevantes para el Data Warehouse que pertenecen al sistema, para ello se utilizarán las tablas Maestras que contienen datos procesados de una gran cantidad de tablas.

El proceso actual de la extracción de datos consta de una tabla que se actualiza diariamente y otra que se actualiza de forma mensual, a continuación se detallará las características y atributos de cada tabla.

5.2.2.1.1 Base de Datos Gestión Maestra

5.2.2.1.1.1 Tabla MAESTRA_FINAL_DIARIA

La tabla MAESTRA_FINAL_DIARIA contiene una recopilación diaria de diversos datos que se generan durante el día, de distintas tablas ubicadas en las diversas áreas de la empresa.

La tabla MAESTRA_FINAL_DIARIA contiene los siguientes atributos:

Campo	Dato ejemplo	Campo	Dato ejemplo
NOT_FECHA_VENTA	17/08/2012 14:42	NOM_SUPERVISOR	VEGA SALINAS LUIS HUMBERT
NOT_FECHA_ENTREGA	18/08/2012 14:37	NOM_JEFE_REGIONAL	CUADRA CHARME PABLO
NOT_DIFERIDO	0	NOM_SUBGERENTE	NAZAR CLARCK JUAN ESTEBAN
NOT_NUM_NOTA_VENTA	14440114	COD_ZONA	63
FAC_NUMERO	42469302	ZONA	RM 6
FAC_FECHA_FACTURACION	18/08/2012 0:00	NOT_PROYECTO_ARTICULO	205002
NOT_COD_CD	2	LISTA	PROYECTO CIGARRILLOS PHILIP MORRIS
TRU_CODIGO	0	STATUS	G
ART_CODIGO	4400320	EBITDA	TRADICIONAL
COD_EAN	0000078011656	PROYECTO	TRADICIONAL
INVTID	00000078011656	CANAL	TRADICIONAL
DESCR_PRODUCTO	CIG.PHILIP MORRIS AZUL KS BOX 20 UND	TIPO	20
COD_PROVEEDOR	0776198307000	CANTIDAD	18968
NOM_PROVEEDOR	PHILIP MORRIS CHILE COMER. LTD	PRECIO_NETO	18326
KILOS_TOTALES	0,6	COSTO_NETO	0
M_CUBICOS_TOTALES	0,00192	DSCTO	0
CONTE	1	RAPEL	0
CATEGORIA	CIGARRILLOS	NETO_NCT5	0
CLI_CODIGO	009348973K400	CTO_NCT5	18474,832
CLI_CODIGO_INTERNO	080288-2	PRECIO_NETO_GESTION	17849,524
NOM_CLIENTE	FUENTES SILVA JOSE	COSTO_NETO_GESTION	0
DIR_CODIGO	005	OI_DC	0
SEC_CODIGO	0	OI_RENT	0
NOMBRE_SECTOR	0	OI_PER	0
PUEBLO	131447	OI_ING_M	0
NOM_PUEBLO	PE#ALOLEN	OI_MG	0
NOM_COMUNA	PEÑALOLEN		

NOM_PROVINCIA	SANTIAGO	OI_BS	0
COD_REGION	13	SSAA	49
COD_PERSONA	7649	FLETE	5701
VEN_CODIGO	4400	RUTA	18/08/2012 9:00
NOM_VENDEDOR	MALDONADO SUAZO JORGE	FECHA_SALIDA	0
	CARLOS	LEAD_TIME	

Tabla 2: Tabla “Maestra_Final_Diaria”

Características:

- Esta tabla contiene los registros que se generan diariamente en distintas áreas, siendo estas Gestión Comercial, Gestión Logística, Gestión Ventas y Presupuesto.
- Esta tabla tiene un total de 67 atributos.
- La tabla en cada actualización diaria no incorpora los nuevos valores, sino que elimina los datos y los reemplaza por los datos nuevos. Estos datos son los que contenía anteriormente más los datos nuevos.
- Se actualizan diariamente aproximadamente 50 mil registros.
- La actualización toma un tiempo aproximado de 20 minutos a principio de mes y 40 minutos al finalizar el mes.

5.2.2.1.1.2 Tabla MAESTRA_FINAL_MESAÑO

Las tablas MAESTRA_FINAL_MESAÑO se diferencian una de otra según el mes y año, ya que contiene una recopilación del mes, estos datos son fruto de la tabla MAESTRA_FINAL_DIARIA entregando de esta forma gran cantidad de los atributos necesarios para la carga de información en el Data Warehouse.

La tabla MAESTRA_FINAL_MESAÑO contiene los siguientes atributos:

Campo	Dato ejemplo	Campo	Dato ejemplo
NOT_FECHA_VENTA	03/07/2012 12:33	LISTA	LISTA BASE COBERTURA SANTIAGO
NOT_FECHA_ENTREGA		STATUS	G
NOT_DIFERIDO	0	PROYECTO	TRADICIONAL
NOT_NUM_NOTA_VENTA (PK)	14227194	CANAL	TRADICIONAL
FAC_NUMERO	42340874	TIPO	GNV
FAC_FECHA_FACTURACION	03/07/2012 0:00	EBITDA	TRADICIONAL

NOT_COD_CD	2	CANTIDAD	25
TRU_CODIGO (PK)	0	PRECIO_NETO	2572
ART_CODIGO	4151002	COSTO_NETO	1975
INVTID (PK)	07806500241324	RAPEL	0
COD_EAN	7806500241324	NETO_NCT5	0
DESCR_PRODUCTO	SERV.ABOLENGO MINI 2413-6 40 UND	CTO_NCT5	0
COD_PROVEEDOR	0965293108000	SSAA	140
NOM_PROVEEDOR	CMPC TISSUE S.A. (TRADICIONAL)	FLETE	698,8354619
KILOS_TOTALES	1,25	DSCTO_ESP	0
M_CUBICOS_TOTALES	0,0252	NC_MDC	0
CONTE	1	CTO_MDC	0
CATEGORIA	SERVILLETAS	PRECIO_NETO_GESTION	2572
CLI_CODIGO	0079066397400	COSTO_NETO_GESTION	1975
CLI_CODIGO_INTERNO	108496-2	GTO_FIN	0
NOM_CLIENTE	VARAS RODRIGUEZ MARIA INES	GTO_FVTA	48
DIR_CODIGO	001	GTO_VVTA	38
SEC_CODIGO	0	GTO_REEMB	53
NOMBRE_SECTOR	0	GTO_ESTR	8
PUEBLO	131529	GTO_DR	41
NOM_PUEBLO	LAMPA	OI_DC	0
NOM_COMUNA	LAMPA	OI_PE	0
NOM_PROVINCIA	CHACABUCO	OI_MG	0
COD_REGION	13	OI_BS	197,5
COD_PERSONA	6655	OI_RT	0
VEN_CODIGO	3880	OI_Ot	3,2691E-13
NOM_VENDEDOR	SANCHEZ MUÑOZ JOSE MIGUEL	OI_T	197,5
NOM_SUPERVISOR	QUIJADA CANEO FRANCISCO A	RAMPLA	0
NOM_JEFE_REGIONAL	GANGAS ARANDA FREDY CRIST	RUTA	1709
NOM_SUBGERENTE	NAZAR CLARCK JUAN ESTEBAN	FECHA_SALIDA	03/07/2012 19:45
COD_ZONA	57	LEAD_TIME	0
NOT_PROYECTO_ARTICULO	2002	FLETE_x_CLI	288
ZONA	RM 1		

Tabla 3: Tabla “Maestra_Final_Mensual”

Características:

- Esta tabla es el resultado de la recopilación diaria de registros de la tabla MAESTRA_FINAL_DIARIA, que contiene en resumen todos los registros del mes.

- Estas tablas existen desde Enero del 2010, generándose mensualmente una nueva, por lo que se tiene información histórica desde esa fecha, en distintas tablas.
- A la fecha de septiembre del 2012, fecha en la cual se realizó el análisis en la empresa, se detectó un total de 23 tablas.
- En base a estas tablas se responde la mayor cantidad de consultas que generan los usuarios finales.
- Esta tabla contiene un total de 78 atributos.
- Contiene un atributo OI_T destacado con color verde en la tabla, que contiene la suma de los OI, atributos contenidos en la misma tabla.
- Contiene 11 atributos que son ingresados de forma manual mes a mes, que se ven resaltados con el color amarillo en la tabla, y corresponden a los siguientes atributos:
 - NC_MDC
 - CTO_MDC
 - GTO_FIN
 - GTO_FVTA
 - GTO_VVTA
 - GTO_REEMB
 - GTO_ESTR
 - GTO_DR
 - RAMPLA
 - FLETE_x_CLI

5.2.2.2 Fuentes Externas

Hace referencia a las fuentes que aportan datos desde otras áreas, siendo en este caso las tablas que se encuentran en el área de Sistemas.

5.2.2.2.1 Base de Datos Relaciones

5.2.2.2.1.1 Tabla Detalle_Producto

Se extraerán los siguientes atributos:

Campo	Dato ejemplo
INVTID (PK)	07791293991054
ENV	UND
COD_MARCA	R015
DESCR_MARCA	REXONA
COD_SECCION	6
DESCR_SECCION	CUIDADO PERSONAL
COD_EJECUTIVO	002689
NOMBRE_EJECUTIVO	BRAVO VILLEGAS ALLISON ESTRELL
WEIGHT	150
VOLUME	402,04

Tabla 4: Tabla "Detalle_Productos"

5.2.2.3 Volúmenes de datos de las fuentes Internas y Externas

Analizando los datos de las 23 tablas de Maestra_Final_MesAño, que se han generado a partir del año 2010 hasta el mes de septiembre de 2012, fecha en la cual se realizó el estudio.

Total registros por tabla mensual = Total Diario aprox. x 23 días hábiles

$$1.150.000 = 50.000 \times 23$$

Ahora si tenemos un total de 78 atributos por registro.

Total de datos mensuales = Total de Registros x Cantidad de atributos

$$89.700.000 = 1.150.000 \times 78$$

Entonces, por tabla se extraerán aproximadamente 89.700.000 datos, con aproximadamente un total de 1.150.000 registros.

Tabla	Cantidad de Tablas	Registros totales (aprox.)	Datos totales (aprox.)	Almacenamiento
Maestra_Final_Diaria	1	50.000	3.900.000	55 MB diarios
Maestra_Final_MesAño	1	1.150.000	89.700.000	1,2 GB

Tabla 5: Cantidad de Datos y Registros actuales

Cabe destacar que los cálculos realizados son evaluados con datos aproximados, por lo que los resultados pueden diferir con los actuales.

5.2.2.4 Relaciones

En el OLTP de la empresa analizada, el proceso principal de venta está representado por el diagrama entidad relación de la Figura 18 que se presenta a continuación:

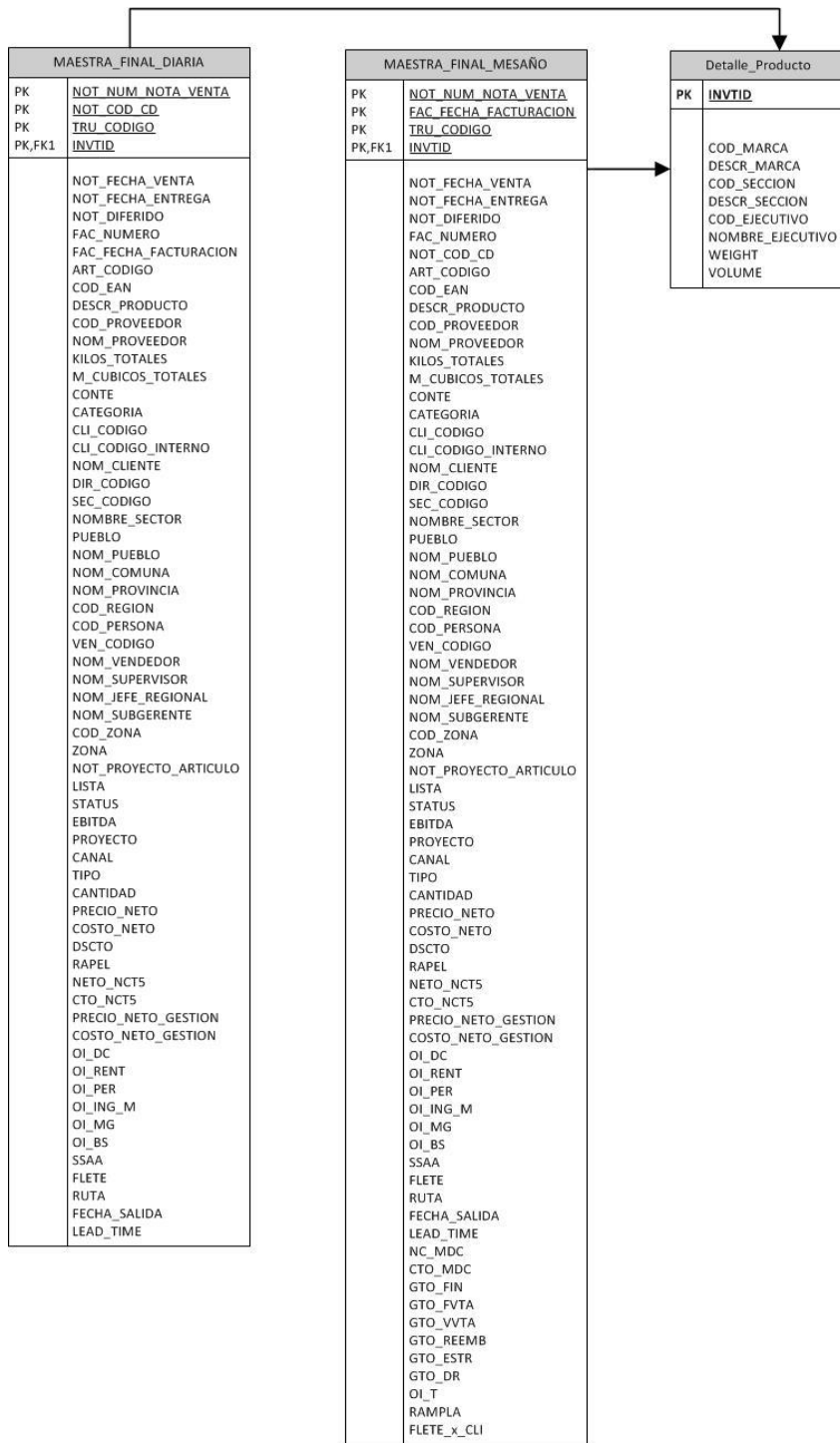


Figura 18: Diagrama Entidad Relación

A continuación, en la Figura 19, se expondrá la relación entre los atributos del modelo actual de la base de datos y el diagrama presentado anteriormente:

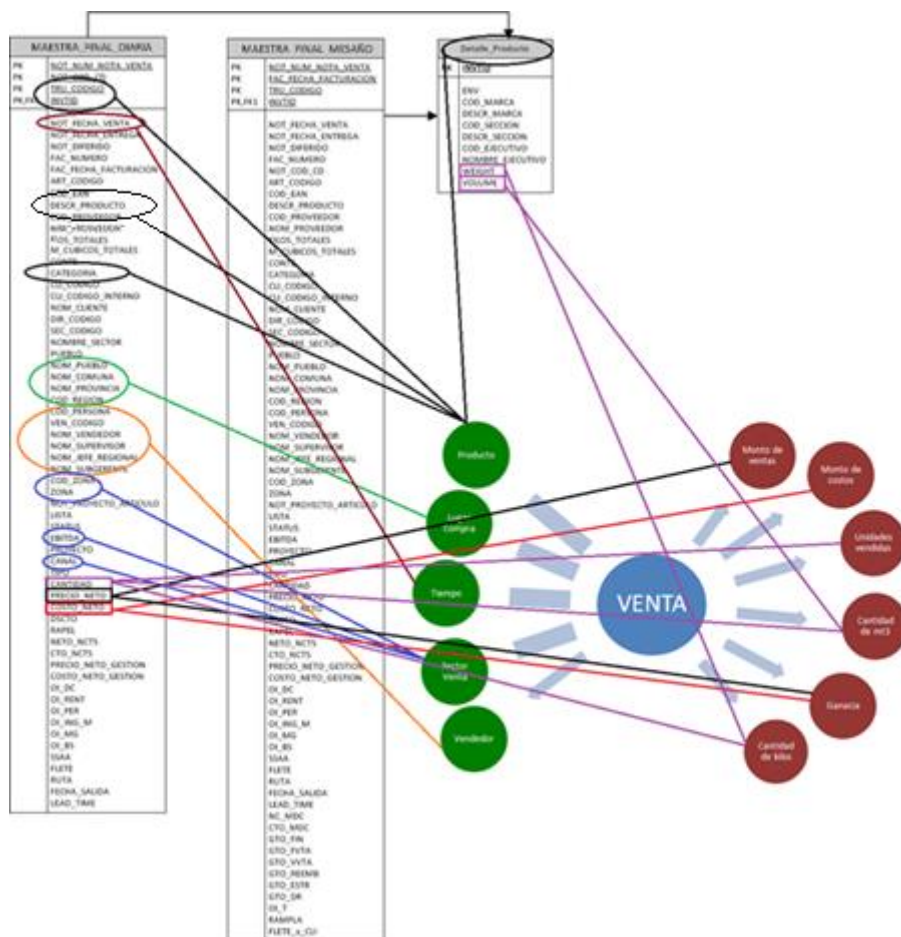


Figura 19: Relación entre el Modelo Relacional y el Diagrama propuesto

Las relaciones identificadas fueron las siguientes:

- Cabe señalar que los campos de la tabla “MAESTRA_FINAL_DIARIA” seleccionados corresponden de igual forma a la tabla “MAESTRA_FINAL_MENSUAL”, los que no se señalaron por razones de orden en el gráfico. Por lo que más abajo se referirá solamente a la tabla “MAESTRA” que hace referencia a las dos tablas antes mencionadas.

- La tabla “MAESTRA” y la tabla “Detalle_Productos” se relaciona con la perspectiva “Productos”.
- La tabla “MAESTRA” con la perspectiva “Vendedores”.
- La tabla “MAESTRA” con la perspectiva “Lugar de Compra”.
- La tabla “MAESTRA” con la perspectiva “Sector de Venta”.
- El campo “fecha” de la tabla “MAESTRA” con la perspectiva “Tiempo” (debido a que es la fecha principal en el proceso de venta).
- El campo "precio_net" de la tabla “MAESTRA” con el indicador “Monto de ventas”.
- El campo “costo_net” de la tabla “MAESTRA” con el indicador “Monto de costos”.
- El campo "precio_net" restado del campo “costo_net” de la tabla “MAESTRA” con el indicador “Ganancia”.
- El campo "cantidad” de la tabla “MAESTRA” con el indicador “Unidades vendidas”.
- El campo "volume” de la tabla “Detalles_Producto” multiplicado por el campo “unidades” de la tabla “MAESTRA” con el indicador “Cantidad de mt3”.
- El campo "weight” de la tabla “Detalles_Producto” multiplicado por el campo “unidades” de la tabla “MAESTRA” con el indicador “Cantidad de kilos”.

5.2.3 Nivel de Granularidad

En este paso se presenta la examinación y selección de los campos que contendrá cada perspectiva, presentándose en detalle el significado de cada campo y/o valor de los datos pertenecientes a los OLTP.

El significado de los datos se obtuvo directamente de los usuarios, identificando valores posibles y características, con el objeto de decidir cuáles son los datos considerados relevantes para realizar consultas y cuáles no.

Con respecto a la perspectiva “Tiempo”, es muy importante definir el ámbito mediante el cual se agruparán o resumirán los datos.

Cabe destacar que la selección de los campos que integrarán cada perspectiva, se realizó con el respectivo cuidado, ya que esta acción determinará la granularidad de la información encontrada en el Data Warehouse.

De acuerdo a las relaciones establecidas, se analizaron los campos residentes en cada tabla a la que se hacía referencia, a través de dos métodos diferentes. Primero se examinó la base de datos para intuir los significados de cada campo, y luego se consultó con el encargado del sistema sobre algunos aspectos de los cuales no se comprendía su sentido.

A consecuencia de esto se generó el diagrama representado por la Figura 19 del punto anterior, el que presenta las relaciones entre el modelo y el diagrama, permitiendo apreciar los nombres de los campos que son bastante explícitos y se deducen con facilidad, pero aun así fue necesario investigarlos para evitar cualquier tipo de inconsistencia.

5.2.3.1 Perspectiva Productos

Corresponde a los productos asociados a una venta en la organización.

Los datos disponibles son los siguientes:

- **Invtid:** es una de las claves primarias pertenecientes a la Tabla MAESTRA que, en conjunto con otras claves, identifican un producto único.
- **tru_codigo:** es una de las claves primarias pertenecientes a la Tabla MAESTRA que, en conjunto con otras claves, identifican un producto único, la cual corresponde al código de la bodega en la cual se encuentra el producto,
- **Descr_Producto:** nombre o descripción del producto.
- **Cod_Marca:** código interno perteneciente a una marca específica.
- **Descr_Marca:** nombre o descripción de la marca.

- **Categoría:** nombre o descripción de la categoría.
- **Cod_Seccion:** clave interna que identifica únicamente a una sección de productos.
- **Descr_Seccion:** nombre o descripción de la sección.
- **Weight:** valor numérico correspondiente al peso total del producto.
- **Volume:** valor numérico correspondiente al volumen total del producto.

5.2.3.2 *Perspectiva Vendedor*

Corresponde al vendedor asociado a una venta en la organización.

Los datos que se pueden utilizar son los siguientes:

- **ven_codigo:** código interno único designado a un vendedor.
- **cod_persona:** código interno único correspondiente a una persona de la organización.
- **nom_vendedor:** nombre del vendedor.
- **nom_supervisor:** nombre del supervisor de un vendedor.
- **nom_jefe_regional:** nombre del jefe regional de un vendedor.
- **nom_subgerente:** nombre del subgerente de un vendedor.

5.2.3.3 *Perspectiva Lugar Compra*

Corresponde a lugar geográfico en que se realizó una venta en la organización.

Los datos que se pueden utilizar son los siguientes:

- **cod_region:** corresponde al número de una región en particular.
- **nom_provincia:** corresponde al nombre de la provincia.
- **nom_comuna:** corresponde al nombre de la comuna.
- **nom_pueblo:** corresponde al nombre de la ciudad o pueblo.

5.2.3.4 *Perspectiva Sector Venta*

Corresponde al sector o área de venta de la organización.

Los datos que se pueden utilizar son los siguientes:

- **ebitda:** nombre del segmento al cual pertenece el vendedor, cobertura, mayorista, Instituciones, SSSL, etc.
- **Canal:** es el nombre del segmento más detallado al cual pertenece el vendedor, se ocupa principalmente en SSSL Ejemplo: SSSL -> Lever VIII o Lever RM
- **cod_zona:** código que corresponde a una zona única.
- **Zona:** es el nombre del grupo geográfico al cual pertenece un vendedor

5.2.3.5 *Perspectiva Tiempo*

Corresponde a la fecha en que se realiza una venta.

Es la que determinará la granularidad del depósito de datos, los datos más típicos que pueden emplearse son los siguientes:

- **Año:** número del año
- **mes_nombre:** nombre del mes
- **mes:** número de un mes en un año
- **semestre:** número de semestre
- **trimestre:** número de trimestre
- **semana:** número de la semana en un año.
- **dia_nombre:** nombre del día de la semana, Ejemplo: Lunes.
- **dia:** número del día
- **not_fecha_venta:** fecha completa con el formato original Ejemplo: 23/11/2010

5.2.4 Diagrama conceptual ampliado

Se presentará un diagrama que graficará los resultados de los pasos anteriores, el cual se ampliará para agregar bajo cada dimensión los campos elegidos y bajo cada indicador su respectiva fórmula de cálculo, esto se puede apreciar en la Figura 20.

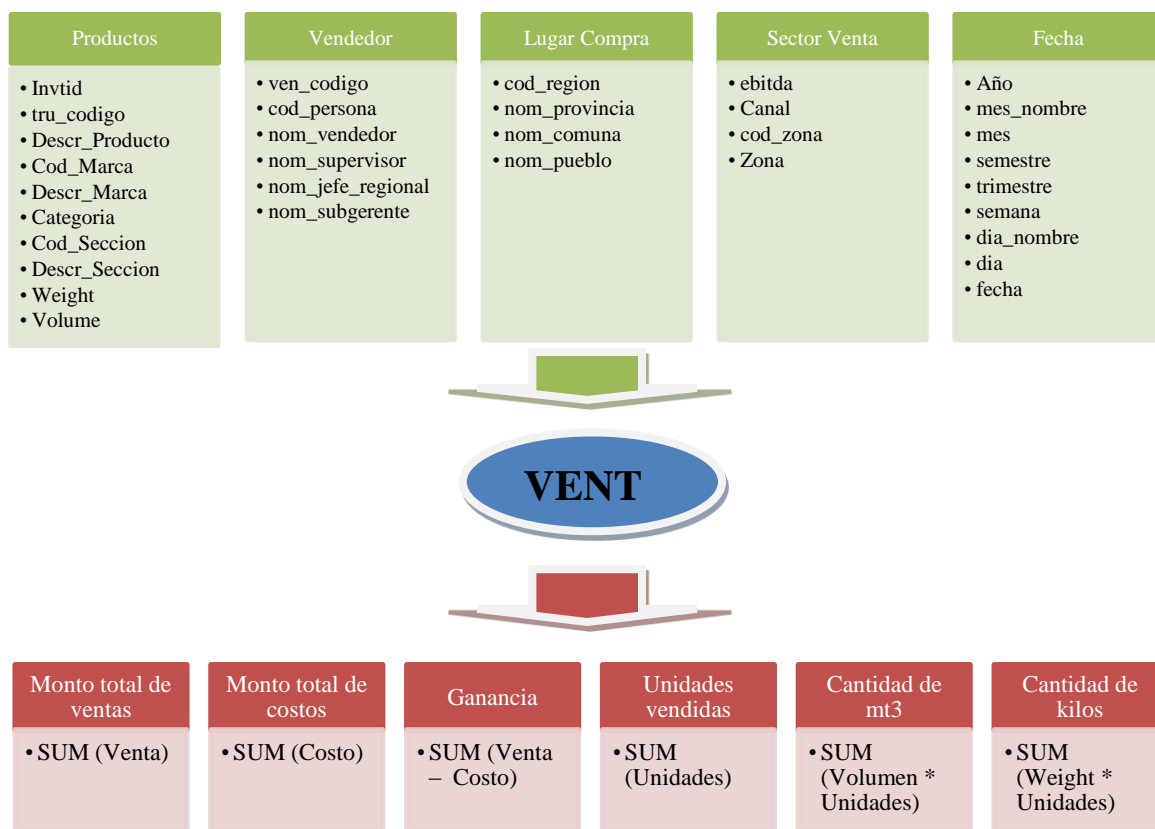


Figura 20: Diagrama Conceptual Ampliado

5.3 Modelo lógico del Data Warehouse

Para generar el modelo lógico, se debe tener presente que los atributos de las respectivas dimensiones se deben organizar en jerarquías definidas por el usuario, que proporcionan rutas de exploración para ayudar a los usuarios a examinar los datos de un cubo.

Luego de obtener como base el diagrama conceptual, se procede a crear el modelo lógico de la estructura del Data Warehouse, definiendo el tipo de modelo que se utilizará, de esta forma se podrán aplicar las acciones propias al modelo, para diseñar las tablas de dimensiones y de hechos. Finalmente, se realizarán las uniones pertinentes entre estas tablas.

5.3.1 Tipo de Modelo Lógico

Mediante un análisis aplicado a la estructura de datos que actualmente posee la empresa, se propone el modelo estrella (star), que consiste de una gran tabla central que contiene información sobre los hechos, y tablas más pequeñas (relacionadas a la tabla de hechos) con información sobre las dimensiones.

Esto se debe principalmente a que los datos almacenados en el área de Soporte de Información no se encuentran normalizados, por lo que lo más adecuado es utilizar el esquema estrella.

Las ventajas que trae aparejada la desnormalización, son las de obviar uniones (Join) entre las tablas cuando se realizan consultas, procurando así un mejor tiempo de respuesta y una mayor sencillez con respecto a su utilización.

El esquema en estrella es el más simple de interpretar y es soportado por casi todas las herramientas de consulta y análisis, y los metadatos son fáciles de documentar y mantener.

5.3.2 Tablas de Dimensiones

Se procede a diseñar las tablas de dimensiones que formarán parte del Data Warehouse. El procedimiento consta en tomar cada dimensión con sus campos relacionados y realizar el siguiente proceso:

- Se elegirá un nombre que identifique la tabla de dimensión.
- Se añadirá un campo que represente su clave principal.
- Se redefinirán los nombres de los campos si es que no son lo suficientemente intuitivos.
- Se definirán las jerarquías de los atributos, esto es sumamente importante para la creación de cubos multidimensionales

El último proceso mencionado anteriormente, se debe a que en muchas ocasiones interesa disponer de los datos a varios niveles de granularidad, en estos casos se crea una jerarquía con la dimensión, debido a que se tienen varios niveles de asociación de los datos.

5.3.2.1 Dimensión “Productos”

- La nueva tabla de dimensión tendrá el nombre “PRODUCTO”.
- Se le agregará una clave principal con el nombre “id_Producto”.
- No se modificará ningún campo de la dimensión productos en la tabla.

Se puede apreciar el resultado de estas operaciones en la Figura 21.

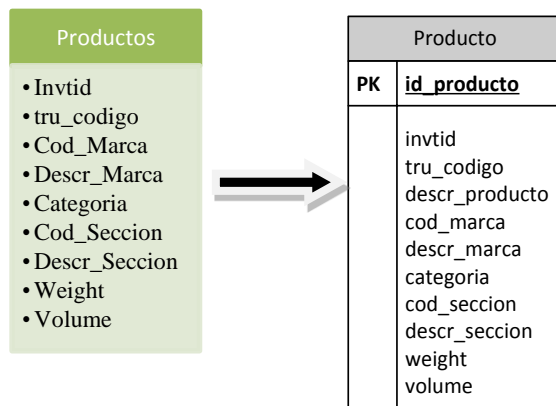


Figura 21: Tabla de dimensión "PRODUCTO"

Y en la Figura 22, se define la jerarquía de la dimensión “Productos”, en donde se definen las siguientes agregaciones:

- Un producto en especial pertenece sólo a una marca. Una marca puede tener uno o más productos.
- Una marca en especial pertenece sólo a una categoría. Una categoría puede tener una o más marcas.
- Una categoría en especial pertenece sólo a una sección. Y una sección puede tener una o más categorías.

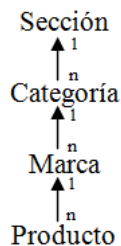


Figura 22: Esquema de Jerarquía dimensión "PRODUCTO"

5.3.2.2 Dimensión “Vendedor”

- La nueva tabla de dimensión tendrá el nombre “VENDEDOR”.
- Se le agregará una clave principal con el nombre “id_vendedor”.
- Se cambiará el nombre del campo de “ven_codigo” por “cod_vendedor”.
- Se cambiará el nombre del campo de “nom_supervisor” por “supervisor”.
- Se cambiará el nombre del campo de “nom_vendedor” por “vendedor”.
- Se cambiará el nombre del campo de “nom_subgerente” por “subgerente”.

Se puede apreciar el resultado de estas operaciones en la Figura 23.

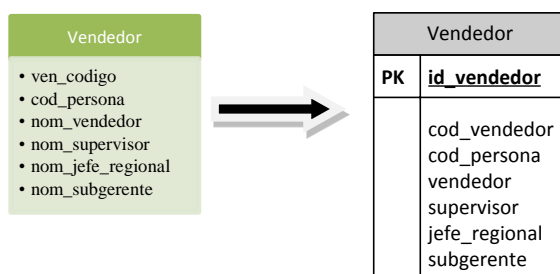


Figura 23: Tabla de dimensión "VENDEDOR"

Y en la Figura 24 se define la jerarquía de la dimensión “Vendedor”, en donde se definen las siguientes agregaciones:

- Un vendedor en especial pertenece solo a un supervisor. Un supervisor puede tener uno o más vendedores.
- Un supervisor en especial pertenece solo a un jefe regional. Un jefe regional puede tener una o más supervisores.
- Un jefe regional en especial pertenece solo a una gerente. Y un gerente puede tener un o más jefes regionales.

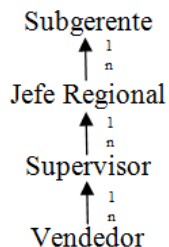


Figura 24: Esquema de Jerarquía dimensión "VENDEDOR"

5.3.2.3 Dimensión "Lugar Compra"

- La nueva tabla de dimensión tendrá el nombre "LUGAR_COMPRA".
- Se le agregará una clave principal con el nombre "id_lugar_compra".
- Se agregará un campo calculado del campo "cod_region" con el nombre de "region", que contendrá el nombre de la región correspondiente.
- Se cambiará el nombre del campo de "cod_region" por "n_region".
- Se cambiará el nombre del campo de "nom_provincia" por "provincia".
- Se cambiará el nombre del campo de "nom_comuna" por "comuna".
- Se cambiará el nombre del campo de "nom_pueblo" por "ciudad".

Se puede apreciar el resultado de estas operaciones en la Figura 25.

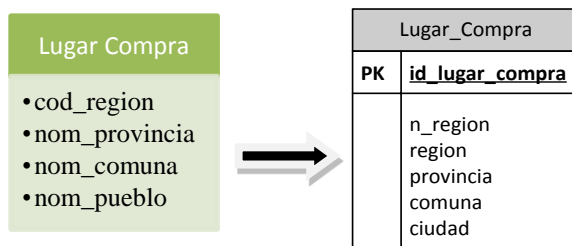


Figura 25: Tabla de dimensión "LUGAR_COMPRA"

Y en la Figura 26 se define la jerarquía de la dimensión “Lugar_Compra”, en donde se definen las siguientes agregaciones:

- Una ciudad en especial pertenece solo a una comuna. Una comuna puede tener una o más ciudades.
- Una comuna en especial pertenece solo a una provincia. Una provincia puede tener una o más comunas.
- Una provincia en especial pertenece solo a una región. Y una región puede tener una o más comunas.

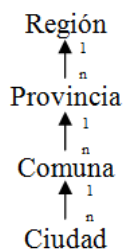


Figura 26: Esquema de Jerarquía dimensión "LUGAR_COMPRA"

5.3.2.4 Dimensión “Sector Venta”:

- La nueva tabla de dimensión tendrá el nombre “SECTOR_VENTA”.
- Se le agregará una clave principal con el nombre “id_sector_venta”.
- Se cambiará el nombre del campo de “ebitda” por “canal_1”.
- Se cambiará el nombre del campo de “canal” por “canal_2”.

Se puede apreciar el resultado de estas operaciones en la Figura 27.

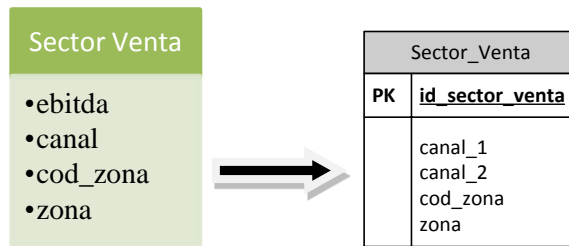


Figura 27: Tabla de dimensión "SECTOR_VENTA"

Y en la Figura 28 se define la jerarquía de la dimensión “Sector_Venta”, en donde se definen las siguientes agregaciones:

- Una zona en especial pertenece solo a un canal 1. Un Canal 1 puede tener uno o más zonas.
- Una zona en especial pertenece solo a un canal 2. Un Canal 2 puede tener uno o más zonas.

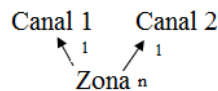


Figura 28: Esquema de Jerarquía dimensión "SECTOR_VENTA"

5.3.2.5 Dimensión “Tiempo”:

- La nueva tabla de dimensión tendrá el nombre “FECHA”.
- Se cambiará el nombre del campo de “not_fecha_venta” por “fecha_completa”.
- Se generará a partir del campo “not_fecha_venta” el campo “año”.
- Se generará a partir del campo “not_fecha_venta” el campo “mes_nombre”.
- Se generará a partir del campo “not_fecha_venta” el campo “mes”.
- Se generará a partir del campo “not_fecha_venta” el campo “semestre”.
- Se generará a partir del campo “not_fecha_venta” el campo “trimestre”.
- Se generará a partir del campo “not_fecha_venta” el campo “semana”.
- Se generará a partir del campo “not_fecha_venta” el campo “dia_nombre”.

- Se generará a partir del campo “not_fecha_venta” el campo “dia”.

Se puede apreciar el resultado de estas operaciones en la Figura 29.

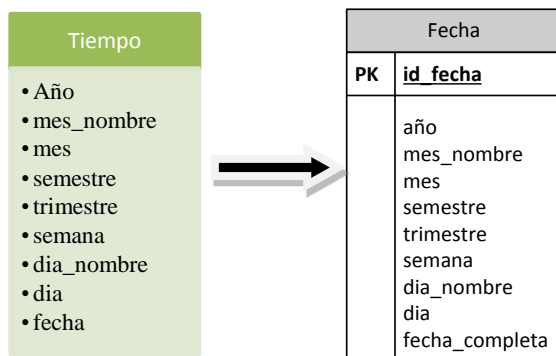


Figura 29: Tabla de dimensión "FECHA"

Y en la Figura 30 se define la jerarquía de la dimensión “**Fecha**”, en donde se definen las siguientes agregaciones:

- Un día de la semana pertenece solo a una semana del mes. Una semana del mes tiene uno o más días de la semana.
- Una semana del año pertenece solo a un mes. Un mes tiene una o más semanas del año.
- Un mes del año pertenece solo a un semestre del año. Un semestre del año tiene uno o más meses del año.
- Un mes del año pertenece solo a un trimestre del año. Un trimestre del año tiene uno o más meses del año.
- Un semestre del año pertenece solo a un año. Un año tiene uno o más semestres del año.
- Un trimestre del año pertenece solo a un año. Un año tiene uno o más trimestres del año.

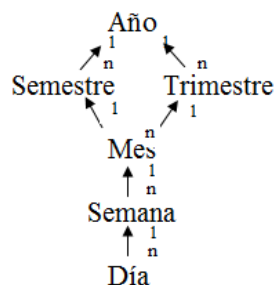


Figura 30: Esquema de Jerarquía dimensión "FECHA"

5.3.3 Tabla de Hechos

Se definirán las tablas de hechos, las cuales contendrán los hechos a través de los cuales se construirán los indicadores de estudio, cumpliendo con lo siguiente:

- Se debe asignar un nombre a la tabla de hechos que represente la información analizada.
- Se definirá su clave primaria, que se compone de la combinación de las claves primarias de cada tabla de dimensión relacionada.
- Se crearán tantos campos de hechos como indicadores se hayan definido en el diagrama conceptual, asignándoles los mismos nombres que estos o podrán ser renombrados para su comprensión.

5.3.3.1 Tabla de Hechos "VENTA"

- La tabla de hechos tendrá el nombre "VENTA".
- Su clave principal será la combinación de las claves primarias de las tablas de dimensiones antes definidas: "id_producto", "id_vendedor", "id_lugar_compra", "id_sector_venta", "id_fecha", en conjunto con los atributos "NUM_NOTA_VENTA" y "NOT_COD_CD" de la tabla

“Maestra_Final_MesAño”, estos dos atributos permiten diferenciar una venta de otra.

- Se crearán 6 hechos, que se corresponden con los dos indicadores y serán renombrados:
 - “Monto de ventas” por “MontoVenta”
 - “Monto de costos” por “MontoCosto”
 - “Ganancia” no sufrirá cambios.
 - “Unidades Vendidas” por “Unidades”
 - “Cantidad de mt3” por “Volumen”
 - “Cantidad de kilos” por “Kilos”

A continuación se presenta en la Figura 31 la tabla de hechos.

Venta	
PK	<u>id_producto</u>
PK	<u>id_lugar_compra</u>
PK	<u>id_sector_venta</u>
PK	<u>id_vendedor</u>
PK	<u>id_fecha</u>
PK	<u>num_nota_venta</u>
PK	<u>not_cod_cd</u>
	MontoVenta MontoCosto Ganancia Unidades Volumen Kilos

Figura 31: Tabla de Hechos "VENTA"

5.3.4 Uniones

En la Figura 32 se presenta el esquema tipo estrella propuesto anteriormente, en el cual se aprecian las uniones correspondientes entre sus tablas de dimensiones y la tabla de hechos.

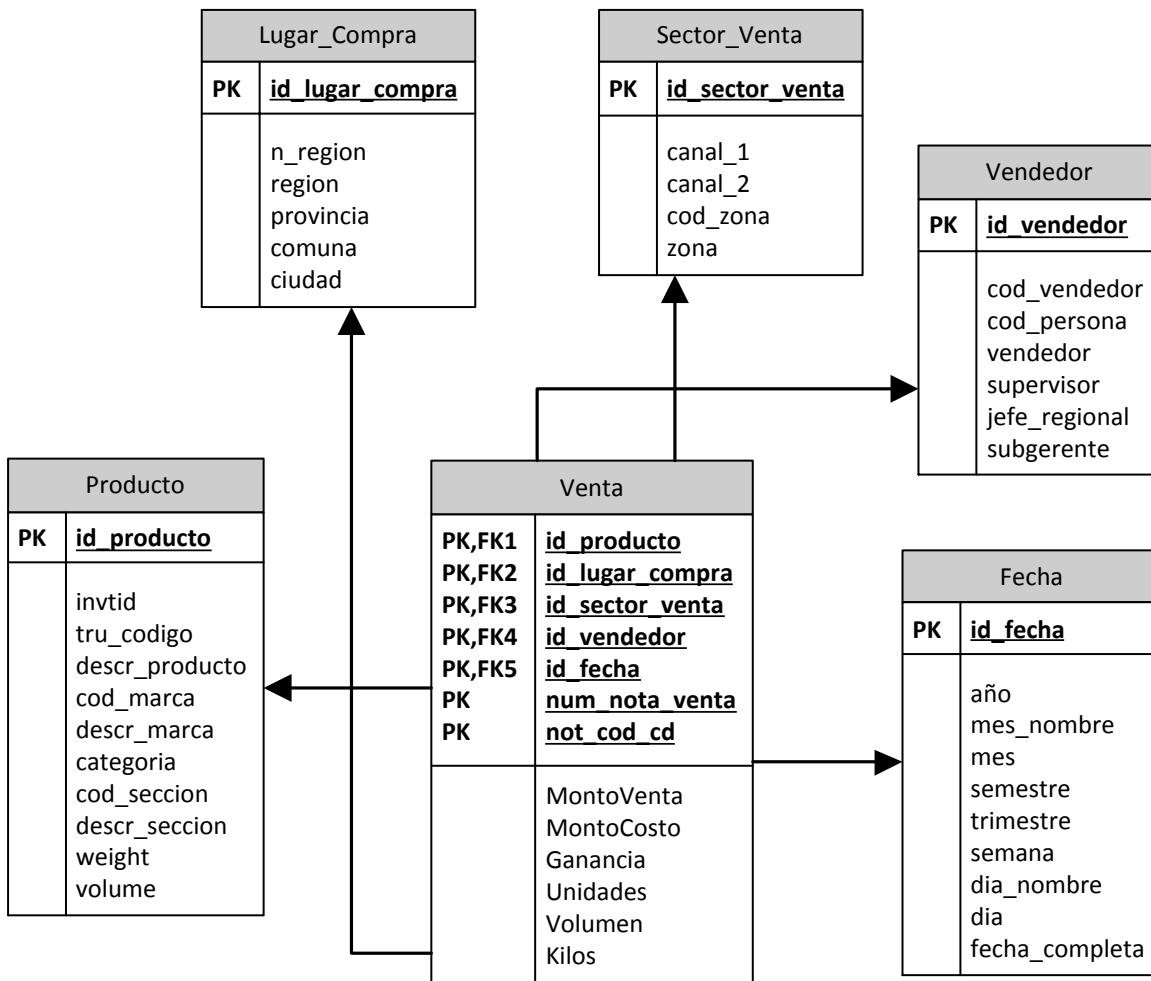


Figura 32: Modelo tipo Estrella del Data Warehouse

5.4 Procesos ETL

Una vez definido el modelo lógico, comienza la fase de prueba con datos, a través de los procesos de ETL, esto quiere decir que se extraerán datos de diferentes fuentes, para luego integrarlos, filtrarlos y depurarlos, para ello se utilizarán sólo sentencias SQL que contendrán los datos que serán de interés.

La información que estará almacenada en el depósito de datos tendrá condiciones adicionales y restricciones, siendo éstas analizadas con anticipación para evitar pérdidas de datos importantes.

A modo de resumen, el proceso consta de las siguientes partes; Primero se cargarán los datos de las dimensiones y luego los de la tabla de hechos, siempre consciente de la correcta correspondencia entre cada elemento. Para finalmente, luego de la carga de los datos, establece las políticas de actualización o refresco de los datos.

A continuación, se generarán las sentencias SQL para cargar las diferentes tablas de dimensiones y la tabla de hechos.

5.4.1 Tabla de dimensión “Productos”

Se tomará como fuente de entrada las tablas “Maestra_Final_MesAño”, “Maestra_Final_Diaria” unidas cada una por si sola a la tabla “Detalle_Producto” del OLTP mencionado anteriormente.

Se consultó a Soporte de Información y se averiguó que se deseaba tener todos los productos que estuvieran relacionados a una venta y que no contengan ningún campo con valor NULL.

Se presenta a continuación la consulta SQL para la creación de la tabla “Productos”:

```
CREATE TABLE [dbo].[Producto] (
  [id_producto] [bigint] IDENTITY(1,1) NOT NULL,
  [invtid] [nvarchar](50) NOT NULL,
  [tru_codigo] [int] NOT NULL,
  [Descr_Producto] [nvarchar](255) NOT NULL,
  [Cod_Marca] [nvarchar](4) NOT NULL,
  [Descr_Marca] [nvarchar](30) NOT NULL,
  [Categoria] [nvarchar](255) NOT NULL,
  [Cod_Seccion] [nvarchar](6) NOT NULL,
  [Descr_Seccion] [nvarchar](100) NOT NULL,
  [Weight] [float] NOT NULL,
  [Volume] [float] NOT NULL,
  CONSTRAINT [PK_Producto] PRIMARY KEY CLUSTERED
  (
    [id_producto] ASC
  ) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Ahora para la carga y actualización de los datos provenientes de las tablas “Maestra_Final_MesAño” y la tabla “Maestra_Final_Diaria”, se utilizó la siguiente sentencia SQL, la cual inserta solo productos que no estén presentes en el Data Warehouse:

```

INSERT INTO [Data Warehouse].[dbo].[Producto]
    ([invtid]
    , [tru_codigo]
    , [Cod_Marca]
    , [Descr_Marca]
    , [Categoria]
    , [Cod_Seccion]
    , [Descr_Seccion]
    , [Weight]
    , [Volume])
SELECT
    [Maestra_Final_MesAño].[invtid],
    [Maestra_Final_MesAño].[tru_codigo],
    [Maestra_Final_MesAño].[Descr_Producto],
    [Maestra_Final_MesAño].[Cod_Marca],
    [Maestra_Final_MesAño].[Descr_Marca],
    [Maestra_Final_MesAño].[Categoria],
    [Maestra_Final_MesAño].[Cod_Seccion],
    [Maestra_Final_MesAño].[Descr_Seccion],
    [Detalle_Producto].[Weight],
    [Detalle_Producto].[Volume]
FROM [Rabie].[dbo].[Maestra_Final_MesAño] INNER JOIN
[Maestra_Final_MesAño].[invtid]= [Detalle_Producto].[invtid]
WHERE
    [Maestra_Final_MesAño].[invtid] is not null and
    [Maestra_Final_MesAño].[tru_codigo] is not null and
    [Maestra_Final_MesAño].[Descr_Producto] is not null and
    [Maestra_Final_MesAño].[Descr_Marca] is not null and
    [Maestra_Final_MesAño].[Categoria] is not null and
    [Maestra_Final_MesAño].[Cod_Seccion] is not null and
    [Maestra_Final_MesAño].[Descr_Seccion] is not null and
    [Detalle_Producto].[Weight] is not null and
    [Detalle_Producto].[Volume] is not null
EXCEPT
SELECT
    [Maestra_Final_MesAño].[invtid],
    [Maestra_Final_MesAño].[tru_codigo],
    [Maestra_Final_MesAño].[Descr_Producto],
    [Maestra_Final_MesAño].[Cod_Marca],
    [Maestra_Final_MesAño].[Descr_Marca],
    [Maestra_Final_MesAño].[Categoria],
    [Maestra_Final_MesAño].[Cod_Seccion],
    [Detalle_Producto].[Weight],
    [Detalle_Producto].[Volume]
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
INNER JOIN [Data Warehouse].[dbo].[producto] ON
[Maestra_Final_MesAño].[invtid]=[Producto].[invtid] and
[Maestra_Final_MesAño].[tru_codigo]=[Producto].[tru_codigo] and

[Maestra_Final_MesAño].[Cod_Marca]=[Producto].[Cod_Marca] and
[Maestra_Final_MesAño].[Descr_Marca]= [Producto].[Descr_Marca] and
[Maestra_Final_MesAño].[Categoria]=[Producto].[Categoria] and
[Maestra_Final_MesAño].[Cod_Seccion]=[Producto].[Cod_Seccion] and

```

```
INNER JOIN [Rabie].[dbo].[Detalle_Producto] ON
[Maestra_Final_MesAño].[invtid]=[Detalle_Producto].[invtid]
[Detalle_Producto].[Weight]=[Producto].[Weight] and
[Detalle_Producto].[Volume]=[Producto].[Volume]
GROUP BY      [Maestra_Final_MesAño].[invtid],
              [Maestra_Final_MesAño].[tru_codigo],
              [Maestra_Final_MesAño].[Descr_Producto],
              [Maestra_Final_MesAño].[Cod_Marca],
              [Maestra_Final_MesAño].[Descr_Marca],
              [Maestra_Final_MesAño].[Categoria],
              [Maestra_Final_MesAño].[Cod_Seccion],
              [Maestra_Final_MesAño].[Descr_Seccion],
```

5.4.2 Tabla de dimensión “Vendedor”

Se tomará como fuente de entrada la tabla “Maestra_Final_MesAño” del OLTP mencionado anteriormente.

Se consultó a Soporte de Información y se averiguó que se deseaba tener todos los vendedores que estuvieran relacionados a una venta y que no contengan ningún campo con valor NULL.

Se presenta a continuación la consulta SQL para la creación de la tabla “Vendedor”:

```
CREATE TABLE [dbo].[Vendedor] (
  [id_vendedor] [bigint] IDENTITY(1,1) NOT NULL,
  [cod_vendedor] [int] NOT NULL,
  [cod_persona] [int] NOT NULL,
  [supervisor] [nvarchar](50) NOT NULL,
  [vendedor] [nvarchar](50) NOT NULL,
  [jefe_regional] [nvarchar](50) NOT NULL,
  [subgerente] [nvarchar](50) NOT NULL,

  CONSTRAINT [PK_Vendedor] PRIMARY KEY CLUSTERED
(
  [id_vendedor] ASC
) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Ahora para la carga y actualización de los datos provenientes de las tablas “Maestra_Final_MesAño” y la tabla “Maestra_Final_Diaria”, se utilizó la siguiente sentencia SQL, la cual inserta solo productos que no estén presentes en el Data Warehouse:

```
INSERT INTO [Data Warehouse].[dbo].[Vendedor]
([cod_vendedor]
,[cod_persona]
,[supervisor]
```

```

, [vendedor]
, [jefe_regional]
, [subgerente])
SELECT      [Maestra_Final_MesAño].[ven_codigo],
            [Maestra_Final_MesAño].[cod_persona],
            [Maestra_Final_MesAño].[nom_supervisor],
            [Maestra_Final_MesAño].[nom_vendedor],
            [Maestra_Final_MesAño].[nom_jefe_regional],
            [Maestra_Final_MesAño].[nom_subgerente]
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
WHERE      [Maestra_Final_MesAño].[ven_codigo] is not null and
            [Maestra_Final_MesAño].[cod_persona] is not null and
            [Maestra_Final_MesAño].[nom_supervisor] is not null and
            [Maestra_Final_MesAño].[nom_vendedor] is not null and
            [Maestra_Final_MesAño].[nom_jefe_regional] is not null and
            [Maestra_Final_MesAño].[nom_subgerente] is not null

EXCEPT
SELECT      [Maestra_Final_MesAño].[ven_codigo],
            [Maestra_Final_MesAño].[cod_persona],
            [Maestra_Final_MesAño].[nom_supervisor],
            [Maestra_Final_MesAño].[nom_vendedor],
            [Maestra_Final_MesAño].[nom_jefe_regional],
            [Maestra_Final_MesAño].[nom_subgerente]
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
INNER JOIN [Data Warehouse P1].[dbo].[Vendedor] ON
            [Maestra_Final_MesAño].[ven_codigo]=[Vendedor].[cod_vendedor] and
            [Maestra_Final_MesAño].[cod_persona]=[Vendedor].[cod_persona] and
            [Maestra_Final_MesAño].[nom_supervisor]=[Vendedor].[supervisor] and
            [Maestra_Final_MesAño].[nom_vendedor]=[Vendedor].[vendedor] and
            [Maestra_Final_MesAño].[nom_jefe_regional]=[Vendedor].[jefe_regional] and
            [Maestra_Final_MesAño].[nom_subgerente]=[Vendedor].[subgerente]
GROUP BY   [Maestra_Final_MesAño].[ven_codigo],
            [Maestra_Final_MesAño].[cod_persona],
            [Maestra_Final_MesAño].[nom_supervisor],
            [Maestra_Final_MesAño].[nom_vendedor],
            [Maestra_Final_MesAño].[nom_jefe_regional],
            [Maestra_Final_MesAño].[nom_subgerente]

```

5.4.3 Tabla de dimensión “Lugar Compra”

Se tomará como fuente de entrada la tabla “Maestra_Final_MesAño” del OLTP mencionado anteriormente.

Se consultó a Soporte de Información y se averiguó que se deseaba tener todos los lugares geográficos en los cuales se realizan compras por los clientes externos y que no contengan ningún campo con valor NULL.

Se presenta a continuación la consulta SQL para la creación de la tabla “Lugar_Compra”:

```
CREATE TABLE [dbo].[Lugar_Compra] (
    [id_lugar_compra] [bigint] IDENTITY(1,1) NOT NULL,
    [n_region] [int] NOT NULL,
    [region] [nvarchar](50) NOT NULL,
    [provincia] [varchar](50) NOT NULL,
    [comuna] [varchar](50) NOT NULL,
    [ciudad] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Lugar_Compra] PRIMARY KEY CLUSTERED
(
    [id_lugar_compra] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Ahora para la carga y actualización de los datos provenientes de las tablas “Maestra_Final_MesAño” y la tabla “Maestra_Final_Diaria”, se insertaron dos códigos de tipo CASE, el primero permite que el atributo “ciudad” contenga datos entendibles para los usuarios, para ello se cambió el simbolo “#” por la letra “Ñ”, y el segundo permite utilizar el atributo “cod_region” para definir los nombres de las regiones para el atributo “region” de la tabla Lugar_Compra. Para lo cual se utilizó la siguiente sentencia SQL, la cual inserta solo lugares geográficos que no estén presentes en el Data Warehouse:

```
INSERT INTO [Data Warehouse].[dbo].[Lugar_Compra]
    ([n_region]
    ,[provincia]
    ,[comuna]
    ,[ciudad]
    ,[region])
SELECT
    [Maestra_Final_MesAño].[cod_region]
    ,[Maestra_Final_MesAño].[nom_provincia]
    ,[Maestra_Final_MesAño].[nom_comuna]
    , CASE
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#ANCUL' THEN 'ÑANCUL'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#IPAS' THEN 'ÑIPAS'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#IQUEN' THEN 'ÑIQUEN'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA' THEN 'ÑUÑO A'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA 35' THEN 'ÑUÑO A
35'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA 36' THEN 'ÑUÑO A
36'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA 37' THEN 'ÑUÑO A
37'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CA#ETE' THEN 'CAÑETE'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CHA#ARAL' THEN 'CHAÑARAL'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CHA#ARAL ALTO' THEN 'CHAÑARAL
ALTO'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CO#ARIPE' THEN 'COÑARIPE'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'DO#IHUE' THEN 'DOÑIHUE'
```

```

WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#AFLOR' THEN 'PEÑAFLOR'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN' THEN 'PEÑALOLEN'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      80' THEN
'PEÑALOLEN      80'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      81' THEN
'PEÑALOLEN      81'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      82' THEN
'PEÑALOLEN      82'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      /ES' THEN
'PEÑALOLEN      /ES'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR' THEN 'VIÑA DEL
MAR'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR  1' THEN 'VIÑA
DEL MAR  1'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR  2' THEN 'VIÑA
DEL MAR  2'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR  3' THEN 'VIÑA
DEL MAR  3'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR  4' THEN 'VIÑA
DEL MAR  4'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR  5' THEN 'VIÑA
DEL MAR  5'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR  6' THEN 'VIÑA
DEL MAR  6'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR  7' THEN 'VIÑA
DEL MAR  7'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR /01' THEN 'VIÑA
DEL MAR /01'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR /02' THEN 'VIÑA
DEL MAR /02'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR /03' THEN 'VIÑA
DEL MAR /03'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VICU#A' THEN 'VICUÑA'
WHEN [Maestra_Final_MesAño].[nom_pueblo] is not null THEN
[Maestra_Final_MesAño].[nom_pueblo]
END
,CASE
WHEN [Maestra_Final_MesAño].[COD_REGION]=1 THEN 'TARAPACÁ'
WHEN [Maestra_Final_MesAño].[COD_REGION]=2 THEN 'ANTOFAGASTA'
WHEN [Maestra_Final_MesAño].[COD_REGION]=3 THEN 'ATACAMA'
WHEN [Maestra_Final_MesAño].[COD_REGION]=4 THEN 'COQUIMBO'
WHEN [Maestra_Final_MesAño].[COD_REGION]=5 THEN 'VALPARAISO'
WHEN [Maestra_Final_MesAño].[COD_REGION]=6 THEN 'OHIGGINS'
WHEN [Maestra_Final_MesAño].[COD_REGION]=7 THEN 'MAULE'
WHEN [Maestra_Final_MesAño].[COD_REGION]=8 THEN 'BIOBÍO'
WHEN [Maestra_Final_MesAño].[COD_REGION]=9 THEN 'ARAUCANÍA'
WHEN [Maestra_Final_MesAño].[COD_REGION]=10 THEN 'LOS LAGOS'
WHEN [Maestra_Final_MesAño].[COD_REGION]=11 THEN 'AYSÉN'
WHEN [Maestra_Final_MesAño].[COD_REGION]=12 THEN 'MAGALLANES'
WHEN [Maestra_Final_MesAño].[COD_REGION]=13 THEN 'RM'
WHEN [Maestra_Final_MesAño].[COD_REGION]=14 THEN 'LOS RÍOS'
WHEN [Maestra_Final_MesAño].[COD_REGION]=15 THEN 'ARICA Y PARINACOTA'
END
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
WHERE      [Maestra_Final_MesAño].[cod_region] is not null and
           [Maestra_Final_MesAño].[nom_provincia] is not null and
           [Maestra_Final_MesAño].[nom_comuna] is not null and

```

```

[Maestra_Final_MesAño].[nom_pueblo] is not null and
[Maestra_Final_MesAño].[cod_region] is not null

EXCEPT
SELECT      [Maestra_Final_MesAño].[cod_region]
           , [Maestra_Final_MesAño].[nom_provincia]
           , [Maestra_Final_MesAño].[nom_comuna]
           , [Maestra_Final_MesAño].[nom_pueblo]
           , [Maestra_Final_MesAño].[cod_region]
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
INNER JOIN [Data Warehouse P1].[dbo].[Lugar_Compra] ON
[Maestra_Final_MesAño].[cod_region]=[Lugar_Compra].[n_region] and
[Maestra_Final_MesAño].[nom_provincia]=[Lugar_Compra].[provincia] and
[Maestra_Final_MesAño].[nom_comuna]=[Lugar_Compra].[comuna] and
[Maestra_Final_MesAño].[nom_pueblo]=[Lugar_Compra].[ciudad]
GROUP BY   [Maestra_Final_MesAño].[cod_region]
           , [Maestra_Final_MesAño].[nom_provincia]
           , [Maestra_Final_MesAño].[nom_comuna]
           , [Maestra_Final_MesAño].[nom_pueblo]

```

5.4.4 Tabla de dimensión “Sector Venta”

Se tomará como fuente de entrada la tabla “Maestra_Final_MesAño” del OLTP mencionado anteriormente.

Se consultó a los clientes y se averiguó que se deseaba tener todos los sectores internos de venta que posee la organización y que no contengan ningún campo con valor NULL.

Se presenta a continuación la consulta SQL para la creación de la tabla “Sector_Venta”:

```

CREATE TABLE [dbo].[Sector_Venta] (
    [id_Sector_Venta] [bigint] IDENTITY(1,1) NOT NULL,
    [cod_zona] [varchar](12) NOT NULL,
    [zona] [nvarchar](50) NOT NULL,
    [canal_1] [nvarchar](255) NOT NULL,
    [canal_2] [nvarchar](255) NOT NULL,
    CONSTRAINT [PK_Sector_Venta] PRIMARY KEY CLUSTERED
    (
        [id_Sector_Venta] ASC
    ) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

Ahora para la carga y actualización de los datos provenientes de las tablas “Maestra_Final_MesAño” y la tabla “Maestra_Final_Diaria”, se insertó un código que permite agrupar el valor “Otros” con el valor “Otro” del atributo “canal” de la tabla

Maestra_Final_MesAño, con el objetivo de realizar las agrupaciones de forma correcta para los cálculos realizados por el Data Warehouse:

```

INSERT INTO [Data Warehouse].[dbo].[Sector_Venta]
    ([cod_zona]
    , [zona]
    , [canal_1]
    , [canal_2])
SELECT
    [Maestra_Final_MesAño].[cod_zona],
    [Maestra_Final_MesAño].[zona],
    [Maestra_Final_MesAño].[ebitda],
CASE
WHEN [Maestra_Final_MesAño].[canal]= 'Otro' THEN 'OTROS'
WHEN [Maestra_Final_MesAño].[canal]<> 'Otro' THEN
[Maestra_Final_MesAño].[canal]
END
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
WHERE
    [Maestra_Final_MesAño].[cod_zona] is not null and
    [Maestra_Final_MesAño].[zona] is not null and
    [Maestra_Final_MesAño].[EBITDA] is not null and
    [Maestra_Final_MesAño].[canal] is not null

EXCEPT
SELECT
    [Maestra_Final_MesAño].[cod_zona],
    [Maestra_Final_MesAño].[zona],
    [Maestra_Final_MesAño].[EBITDA],
    [Maestra_Final_MesAño].[canal]
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
INNER JOIN [Data Warehouse P1].[dbo].[Sector_Venta] ON
    [Maestra_Final_MesAño].[cod_zona]=[Sector_Venta].[cod_zona]
and
[Maestra_Final_MesAño].[zona]=[Sector_Venta].[zona] and
[Maestra_Final_MesAño].[EBITDA]=[Sector_Venta].[canal_1] and
[Maestra_Final_MesAño].[canal]=[Sector_Venta].[canal_2]
GROUP BY
    [Maestra_Final_MesAño].[cod_zona],
    [Maestra_Final_MesAño].[zona],
    [Maestra_Final_MesAño].[EBITDA],
    [Maestra_Final_MesAño].[canal]

```

5.4.5 Tabla de dimensión “Tiempo”

Se tomará como fuente de entrada la tabla “Maestra_Final_MesAño” del OLTP mencionado anteriormente.

A continuación se presenta la consulta SQL para la creación de la tabla “Tiempo”:

```

CREATE TABLE [dbo].[Fecha] (
    [id_fecha] [bigint] NOT NULL,
    [año] [int] NOT NULL,
    [mes_nombre] [char](10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [mes] [int] NOT NULL,
    [semestre] [int] NOT NULL,
    [trimestre] [int] NOT NULL,
    [semana] [int] NOT NULL,
    [dia_nombre] [char](10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [dia] [char](10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [fecha_completa] [nvarchar](50) COLLATE Modern_Spanish_CI_AS NOT
NULL,
    CONSTRAINT [PK_Fecha] PRIMARY KEY CLUSTERED
(
    [id_fecha] ASC
) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

Ahora para la carga y actualización de los datos de las tablas “Maestra_Final_MesAño” y la tabla “Maestra_Final_Diaria”, lo que se hizo, fue utilizar una fórmula que permitiera generar la clave primaria de la fecha, para ello se utilizó la fecha del primer movimiento de ventas almacenado en la base de datos, correspondiente a la fecha, en este caso 01/01/2010, hasta la fecha extraída de la venta, en conjunto con otras fórmulas para los semestres, trimestres, días del año, etc, utilizando la siguiente consulta SQL:

```

INSERT INTO [Data Warehouse].[dbo].[Fecha]
    ([id_fecha]
    , [año]
    , [mes_nombre]
    , [mes]
    , [semestre]
    , [trimestre]
    , [semana]
    , [dia_nombre]
    , [dia]
    , [fecha_completa])
SELECT
    DATEDIFF(day, 01/01/2010,
    [Maestra_Final_MesAño].[not_fecha_venta])
    , DATENAME(year, [Maestra_Final_MesAño].[not_fecha_venta])
    , DATENAME(month, [Maestra_Final_MesAño].[not_fecha_venta])
    , DATEPART(month, [Maestra_Final_MesAño].[not_fecha_venta])
, CASE
    WHEN DATEPART(month, [Maestra_Final_MesAño].[not_fecha_venta])
    between 1 and 6 THEN '1'
    WHEN DATEPART(month, [Maestra_Final_MesAño].[not_fecha_venta])
    between 7 and 12 THEN '2'
END
    , DATENAME(quarter, [Maestra_Final_MesAño].[not_fecha_venta])
    , DATENAME(week, [Maestra_Final_MesAño].[not_fecha_venta])
    , DATENAME(weekday, [Maestra_Final_MesAño].[not_fecha_venta])

```

```

        , DATEPART (day, [Maestra_Final_MesAño].[not_fecha_venta])
        , [Maestra_Final_MesAño].[not_fecha_venta]
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
WHERE not_fecha_venta is not null

EXCEPT
SELECT      DATEDIFF ( day, 01/01/2010 ,
               [Maestra_Final_MesAño].[not_fecha_venta] )
           , DATENAME (year, [Maestra_Final_MesAño].[not_fecha_venta])
           , DATENAME (month, [Maestra_Final_MesAño].[not_fecha_venta])
           , DATEPART (month, [Maestra_Final_MesAño].[not_fecha_venta])
, CASE
WHEN DATEPART (month, [Maestra_Final_MesAño].[not_fecha_venta])
between 1 and 6 THEN '1'
WHEN DATEPART (month, [Maestra_Final_MesAño].[not_fecha_venta])
between 7 and 12 THEN '2'
END
           , DATENAME (quarter, [Maestra_Final_MesAño].[not_fecha_venta])
           , DATENAME (week, [Maestra_Final_MesAño].[not_fecha_venta])
           , DATENAME (weekday, [Maestra_Final_MesAño].[not_fecha_venta])
           , DATEPART (day, [Maestra_Final_MesAño].[not_fecha_venta])
           , [Maestra_Final_MesAño].[not_fecha_venta]
FROM [Rabie].[dbo].[Maestra_Final_MesAño]
INNER JOIN [Data Warehouse P1].[dbo].[Fecha] ON
           [Maestra_Final_MesAño].[not_fecha_venta]=[Fecha].[fecha_completa]
GROUP BY  [Maestra_Final_MesAño].[not_fecha_venta]

```

A continuación, en la Figura 33, se puede apreciar una muestra de los datos generados.

id_fecha	año	mes_nombre	mes	semestre	trimestre	semana	dia_nombre	dia	fecha_completa
41120	2012	Agosto	8	2	3	32	Miércoles	1	01/08/2012
41121	2012	Agosto	8	2	3	32	Jueves	2	02/08/2012
41122	2012	Agosto	8	2	3	32	Viernes	3	03/08/2012
41151	2012	Septiembre	9	2	3	36	Sábado	1	01/09/2012
41153	2012	Septiembre	9	2	3	37	Lunes	3	03/09/2012
41181	2012	Octubre	10	2	4	41	Lunes	1	01/10/2012
41182	2012	Octubre	10	2	4	41	Martes	2	02/10/2012
41183	2012	Octubre	10	2	4	41	Miércoles	3	03/10/2012

Figura 33: Caso práctico, datos de dimensión "Fecha"

Como se puede observar, la primera columna representa la clave primaria, la cual fue calculada con la función DATEDIFF que entrega la diferencia de días entre la fecha 01/01/2010 y la fecha a comparar, función que se representa a continuación:

```
DATEDIFF( day, 01/01/2010 , [fecha a comparar] )
```

5.4.6 Tabla de hechos “Venta”

Para la confección de la tabla de hechos, se tuvieron que tomar como fuente las tablas “Maestra_Final_MesAño” y “Maestra_Final_Diaria”. Al igual que en las tablas de dimensiones, se recolectaron las condiciones que deben cumplir los datos para considerarse de interés.

Debido a que la Tabla de hechos accede a las dimensiones, es necesario que éstas estén creadas con anticipación, ya que utiliza las claves primarias.

```
CREATE TABLE [dbo].[Venta] (
    [id_vendedor] [bigint] NOT NULL,
    [id_producto] [bigint] NOT NULL,
    [id_sector_venta] [bigint] NOT NULL,
    [id_fecha] [bigint] NOT NULL,
    [id_lugar_compra] [bigint] NOT NULL,
    [num_nota_venta] [nvarchar] (255) NOT NULL,
    [not_cod_cd] [smallint] NOT NULL,
    [MontoVenta] [float] NOT NULL,
    [MontoCosto] [float] NOT NULL,
    [Ganancia] AS ([MontoVenta]-[MontoCosto]),
    [Unidades] [int] NOT NULL,
    [Volumen] [float] NOT NULL,
    [Kilos] [float] NOT NULL,
    CONSTRAINT [PK_Venta_1] PRIMARY KEY CLUSTERED
(
    [id_vendedor] ASC,
    [id_producto] ASC,
    [id_sector_venta] ASC,
    [id_fecha] ASC,
    [id_lugar_compra] ASC,
    [num_nota_venta] ASC,
    [not_cod_cd] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Venta] WITH CHECK ADD CONSTRAINT [FK_Venta_Fecha]
FOREIGN KEY([id_fecha])
```

```

REFERENCES [dbo].[Fecha] ([id_fecha])
GO
ALTER TABLE [dbo].[Venta] CHECK CONSTRAINT [FK_Venta_Fecha]
GO
ALTER TABLE [dbo].[Venta] WITH CHECK ADD CONSTRAINT
[FK_Venta_Lugar_Compra] FOREIGN KEY([id_lugar_compra])
REFERENCES [dbo].[Lugar_Compra] ([id_lugar_compra])
GO
ALTER TABLE [dbo].[Venta] CHECK CONSTRAINT [FK_Venta_Lugar_Compra]
GO
ALTER TABLE [dbo].[Venta] WITH CHECK ADD CONSTRAINT [FK_Venta_Producto]
FOREIGN KEY([id_producto])
REFERENCES [dbo].[Producto] ([id_producto])
GO
ALTER TABLE [dbo].[Venta] CHECK CONSTRAINT [FK_Venta_Producto]
GO
ALTER TABLE [dbo].[Venta] WITH CHECK ADD CONSTRAINT
[FK_Venta_Sector_Venta] FOREIGN KEY([id_sector_venta])
REFERENCES [dbo].[Sector_Venta] ([id_Sector_Venta])
GO
ALTER TABLE [dbo].[Venta] CHECK CONSTRAINT [FK_Venta_Sector_Venta]
GO
ALTER TABLE [dbo].[Venta] WITH CHECK ADD CONSTRAINT [FK_Venta_Vendedor]
FOREIGN KEY([id_vendedor])
REFERENCES [dbo].[Vendedor] ([id_vendedor])
GO
ALTER TABLE [dbo].[Venta] CHECK CONSTRAINT [FK_Venta_Vendedor]

```

Luego de creada la tabla de hechos es necesario agregar 4 atributos a la tabla Maestra_Final_MesAño, estos campos reducirán la carga del sistema a la hora de cargar los datos a la tabla de hechos. Para esto fue necesario añadir los siguientes atributos:

- id_producto
- id_vendedor
- id_lugar_compra
- id_sector_venta

El atributo id_fecha no fue necesario ya que se puede calcular con facilidad.

Para la agregación de estos cuatro atributos se utilizó el siguiente código:

```

ALTER TABLE [Rabie].[dbo].[Maestra_Final_MesAño]
ADD id_producto INT null;
ALTER TABLE [Rabie].[dbo].[Maestra_Final_MesAño]
ADD id_vendedor INT null;
ALTER TABLE [Rabie].[dbo].[Maestra_Final_MesAño]
ADD id_lugar_compra INT null;
ALTER TABLE [Rabie].[dbo].[Maestra_Final_MesAño]
ADD id_sector_venta INT null;

```

Y para la agregación de los valores de los atributos, con respecto a los datos almacenados en el Data Warehouse, manteniendo de esta forma una consistencia en los datos, se utilizaron los siguientes códigos por atributo:

Agregación de atributo “id_producto”:

```
UPDATE [Rabie].[dbo].[Maestra_Final_MesAño]
SET [id_producto] = [Producto].[id_producto]
FROM [Data Warehouse P1].[dbo].[Producto]
INNER JOIN [Rabie].[dbo].[Maestra_Final_MesAño] ON
[Producto].[invtid]=[Maestra_Final_MesAño].[invtid] and
[Producto].[tru_codigo]=[Maestra_Final_MesAño].[tru_codigo] and
[Producto].[Descr_Producto]=[Maestra_Final_MesAño].[Descr_Producto] and
[Producto].[Cod_Marca]=[Maestra_Final_MesAño].[Cod_Marca] and
[Producto].[Descr_Marca]=[Maestra_Final_MesAño].[Descr_Marca] and
[Producto].[Categoria]=[Maestra_Final_MesAño].[Categoria] and
[Producto].[Cod_Seccion]=[Maestra_Final_MesAño].[Cod_Seccion] and
[Producto].[Descr_Seccion]=[Maestra_Final_MesAño].[Descr_Seccion] and
[Producto].[Weight]=[Maestra_Final_MesAño].[Weight] and
[Producto].[Volume]=[Maestra_Final_MesAño].[Volume]
WHERE [Maestra_Final_MesAño].[id_producto] is null
```

Agregación de atributo “id_vendedor”:

```
UPDATE [Rabie].[dbo].[Maestra_Final_MesAño]
SET [id_vendedor] = [Vendedor].[id_vendedor]
FROM [Data Warehouse P1].[dbo].[Vendedor]
INNER JOIN [Rabie].[dbo].[Maestra_Final_MesAño] ON
[Vendedor].[cod_vendedor]=[Maestra_Final_MesAño].[ven_codigo] and
[Vendedor].[cod_persona]=[Maestra_Final_MesAño].[cod_persona] and
[Vendedor].[supervisor]=[Maestra_Final_MesAño].[nom_supervisor] and
[Vendedor].[vendedor]=[Maestra_Final_MesAño].[nom_vendedor] and
[Vendedor].[jefe_regional]=[Maestra_Final_MesAño].[nom_jefe_regional] and
[Vendedor].[subgerente]=[Maestra_Final_MesAño].[nom_subgerente]
WHERE [Maestra_Final_MesAño].[id_vendedor] is null
```

Agregación del atributo “id_lugar_compra”, en cual se utilizó un CASE debido a que los valores del atributo “ciudad” de la dimensión Lugar_Compra estaban modificados anteriormente, lo que generaba inconsistencias en la comparación con el atributo “nom_pueblo” de la tabla Maestra_Final_MesAño en cuanto a la asignación de la clave primaria correspondiente:

```
UPDATE [Rabie].[dbo].[Maestra_Final_MesAño]
SET [id_lugar_compra]=[Lugar_Compra].[id_lugar_compra]
FROM [Data Warehouse P1].[dbo].[Lugar_Compra]
INNER JOIN [Rabie].[dbo].[Maestra_Final_MesAño] ON
[Lugar_Compra].[n_region]=[Maestra_Final_MesAño].[cod_region] and
[Lugar_Compra].[provincia]=[Maestra_Final_MesAño].[nom_provincia] and
```

```

[Lugar_Compra].[comuna]=[Maestra_Final_MesAño].[nom_comuna] and
[Lugar_Compra].[ciudad]= CASE
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#ANCUL' THEN 'ÑANCUL'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#IPAS' THEN 'ÑIPAS'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#IQUEN' THEN 'ÑIQUEN'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA' THEN 'ÑUÑOA'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA      35' THEN 'ÑUÑOA
35'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA      36' THEN 'ÑUÑOA
36'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = '#U#OA      37' THEN 'ÑUÑOA
37'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CA#ETE' THEN 'CAÑETE'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CHA#ARAL' THEN 'CHAÑARAL'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CHA#ARAL ALTO' THEN 'CHAÑARAL
ALTO'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'CO#ARIPE' THEN 'COÑARIPE'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'DO#IHUE' THEN 'DOÑIHUE'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#AFLOR' THEN 'PEÑAFLOR'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN' THEN 'PEÑALOLEN'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      80' THEN
'PEÑALOLEN      80'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      81' THEN
'PEÑALOLEN      81'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      82' THEN
'PEÑALOLEN      82'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'PE#ALOLEN      /ES' THEN
'PEÑALOLEN      /ES'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR' THEN 'VIÑA DEL
MAR'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR      1' THEN 'VIÑA
DEL MAR      1'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR      2' THEN 'VIÑA
DEL MAR      2'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR      3' THEN 'VIÑA
DEL MAR      3'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR      4' THEN 'VIÑA
DEL MAR      4'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR      5' THEN 'VIÑA
DEL MAR      5'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR      6' THEN 'VIÑA
DEL MAR      6'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR      7' THEN 'VIÑA
DEL MAR      7'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR /01' THEN 'VIÑA
DEL MAR /01'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR /02' THEN 'VIÑA
DEL MAR /02'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VI#A DEL MAR /03' THEN 'VIÑA
DEL MAR /03'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = 'VICU#A' THEN 'VICUÑA'
WHEN [Maestra_Final_MesAño].[nom_pueblo] = [Lugar_Compra].[ciudad] THEN
[Maestra_Final_MesAño].[nom_pueblo]
END
WHERE [Maestra_Final_MesAño].[id_lugar_compra] is null

```

Agregación del atributo “id_sector_venta”, en cual se utilizó un CASE debido a que los valores del atributo “canal_2” de la dimensión Sector_Venta estaban modificados anteriormente, lo que generaba inconsistencias en la comparación con el atributo “canal” de la tabla Maestra_Final_MesAño en cuanto a la asignación de la clave primaria correspondiente:

```
UPDATE [Rabie].[dbo].[Maestra_Final_MesAño]
    SET [id_sector_venta]=[Sector_Venta].[id_sector_venta]
FROM [Data Warehouse P1].[dbo].[Sector_Venta]
INNER JOIN [Rabie].[dbo].[Maestra_Final_MesAño] ON
[Sector_Venta].[cod_zona]=[Maestra_Final_MesAño].[cod_zona] and
[Sector_Venta].[zona]=[Maestra_Final_MesAño].[zona] and
[Sector_Venta].[canal_1]=[Maestra_Final_MesAño].[ebitda] and
[Sector_Venta].[canal_2]= CASE
WHEN [Maestra_Final_MesAño].[canal] = 'Otro' THEN 'OTROS'
WHEN [Maestra_Final_MesAño].[canal] = [Sector_Venta].[canal_2] THEN
[Maestra_Final_MesAño].[canal]END
WHERE [Maestra_Final_MesAño].[id_sector_venta] is null
```

Finalmente, luego de la agregación de las claves primarias de las dimensiones mencionadas anteriormente, se procede a la carga y actualización de la tabla de hechos “Venta” con los datos de las tablas “Maestra_Final_MesAño” y la tabla “Maestra_final_Diaria”, utilizando el siguiente código SQL:

```
INSERT INTO [Data Warehouse P1].[dbo].[Venta]
    ([id_producto]
    ,[id_vendedor]
    ,[id_fecha]
    ,[id_sector_venta]
    ,[id_lugar_compra]
    ,[num_notas_venta]
    ,[not_cod_cd]
    ,[MontoVenta]
    ,[MontoCosto]
    ,[Unidades]
    ,[Volumen]
    ,[Kilos])
SELECT
    [Maestra_Final_MesAño].[id_producto]
    ,[Maestra_Final_MesAño].[id_vendedor]
    ,[DATEDIFF ( day, 01/01/2010 , [Maestra_Final_MesAño].[
not_fecha_venta] )
    ,[Maestra_Final_MesAño].[id_sector_venta]
    ,[Maestra_Final_MesAño].[id_lugar_compra]
    ,[Maestra_Final_MesAño].[not_num_notas_venta]
    ,[Maestra_Final_MesAño].[not_cod_cd]
    ,[Maestra_Final_MesAño].[venta]
    ,[Maestra_Final_MesAño].[costo]
    ,[Maestra_Final_MesAño].[cantidad]
    ,([Producto].[volume]*[Maestra_Final_MesAño].[cantidad])
```



```

, ([Producto].[Weight]*[Maestra_Final_MesAño].[cantidad])
FROM [rabie].[dbo].[Maestra_Final_MesAño]
INNER JOIN [Data Warehouse P1].[dbo].[Producto] ON
[Maestra_Final_MesAño].[id_producto]=[Producto].[id_producto]
WHERE
[Maestra_Final_MesAño].[id_producto] is not null and
[Maestra_Final_MesAño].[id_vendedor] is not null and
[Maestra_Final_MesAño].[id_sector_venta] is not null and
[Maestra_Final_MesAño].[id_lugar_compra] is not null and
[Maestra_Final_MesAño].[not_fecha_venta] is not null

EXCEPT
SELECT
[Maestra_Final_MesAño].[id_producto]
, [Maestra_Final_MesAño].[id_vendedor]
, DATEDIFF ( day, 01/01/2010 , [Maestra_Final_MesAño].[fecha] )
, [Maestra_Final_MesAño].[id_sector_venta]
, [Maestra_Final_MesAño].[id_lugar_compra]
, [Maestra_Final_MesAño].[not_num_nota_venta]
, [Maestra_Final_MesAño].[not_cod_cd]
, [Maestra_Final_MesAño].[venta]
, [Maestra_Final_MesAño].[costo]
, [Maestra_Final_MesAño].[cantidad]
, ([Producto].[volume]*[Maestra_Final_MesAño].[cantidad])
, ([Producto].[Weight]*[Maestra_Final_MesAño].[cantidad])
FROM [rabie].[dbo].[Maestra_Final_MesAño]
INNER JOIN [Data Warehouse P1].[dbo].[Producto] ON
[Maestra_Final_MesAño].[id_producto]=[Producto].[id_producto]
INNER JOIN [Data Warehouse P1].[dbo].[Venta] ON
[Maestra_Final_MesAño].[id_producto]=[Venta].[id_producto] and
DATEDIFF(day,01/01/2010,[Maestra_Final_MesAño].[not_fecha_venta])=[Venta]
.[id_fecha] and
[Maestra_Final_MesAño].[id_sector_venta]=[Venta].[id_sector_venta] and
[Maestra_Final_MesAño].[id_lugar_compra]=[Venta].[id_lugar_compra] and

[Maestra_Final_MesAño].[not_num_nota_venta]=[Venta].[num_nota_venta] and
[Maestra_Final_MesAño].[not_cod_cd]=[Venta].[not_cod_cd] and
[Maestra_Final_MesAño].[venta]=[Venta].[MontoVenta]and
[Maestra_Final_MesAño].[costo]=[Venta].[MontoCosto]and
[Maestra_Final_MesAño].[cantidad]=[Venta].[Unidades]and
([Producto].[volume]*[Maestra_Final_MesAño].[cantidad])=[Venta].[Volumen]
and
([Producto].[Weight]*[Maestra_Final_MesAño].[cantidad])=[Venta].[Kilos]
WHERE
[Maestra_Final_MesAño].[id_producto] is not null and
[Maestra_Final_MesAño].[id_vendedor] is not null and
[Maestra_Final_MesAño].[id_sector_venta] is not null and
[Maestra_Final_MesAño].[id_lugar_compra] is not null and
[Maestra_Final_MesAño].[not_fecha_venta] is not null

GROUP BY
[Maestra_Final_MesAño].[id_producto]
, [Maestra_Final_MesAño].[id_vendedor]
, [Maestra_Final_MesAño].[not_fecha_venta]
, [Maestra_Final_MesAño].[id_sector_venta]
, [Maestra_Final_MesAño].[id_lugar_compra]
, [Maestra_Final_MesAño].[not_num_nota_venta]
, [Maestra_Final_MesAño].[not_cod_cd]
, [Maestra_Final_MesAño].[venta]
, [Maestra_Final_MesAño].[costo]
, [Maestra_Final_MesAño].[cantidad]
, [Producto].[volume]
, [Producto].[Weight]

```

5.5 Creación de Cubos Multidimensionales

Los cubos multidimensionales o cubos OLAP, contienen datos resumidos de grandes Bases de datos o Sistemas Transaccionales (OLTP). Se utilizan principalmente en informes de negocios de ventas, marketing, informes de dirección, minería de datos y áreas similares.

Es necesario el uso de estos cubos debido a la rápida respuesta para las consultas aplicadas a grandes volúmenes de datos, como los que almacena Rabie S.A.

Los cubos multidimensionales se componen de hechos numéricos llamados medidas, que se clasifican por dimensiones. El cubo de metadatos para este caso es creado a partir de un esquema tipo estrella, el cual es definido para una base de datos relacional específica. Las medidas se obtienen de los registros de una tabla de hechos y las dimensiones se derivan de la dimensión de los cuadros.

Los cubos estarán basados en el modelo lógico diseñado, siguiendo los pasos de la metodología Hefesto [1], además se presentará la correcta distinción entre:

- Hechos de una tabla de hechos e indicadores de un cubo.
- Campos de una tabla de dimensión y atributos de un cubo.

A continuación se definirán tres cubos de venta.

5.5.1 Cubo de Ventas 1 (Producto-Fecha-Sector_Venta)

Se creará un cubo multidimensional de ejemplo, basado en el modelo lógico propuesto por Hefesto [1], diseñado para el área de Soporte de Información de Rabie S.A., el cubo tendrá el nombre “Cubo de Ventas 1”, el cual estará conformado por las dimensiones de Producto, Sector_Venta y Fecha, éste cubrirá lo planteado en el Capítulo 5 en el punto 5.1.2 en el cuál se definen las interrogantes para las distintas áreas que a continuación se presentarán, en las cuales los indicadores estarán subrayados con rojo y las dimensiones con verde:

Área de Venta:

1. Monto de ventas de categorías de productos por zona en un tiempo determinado.

Área Comercial:

2. Monto de ventas, el monto de costos y la ganancia de categorías de productos por zona en un tiempo determinado.

Área Logística:

3. Cantidad de mt3 utilizados en los productos en un tiempo determinado
4. Cantidad de kilos presentes en los productos en un tiempo determinado
5. Cantidad de unidades vendidas de los productos en un tiempo determinado.

Por lo que el modelo lógico queda representado en la Figura 34: Modelo lógico Cubo de Ventas 1:

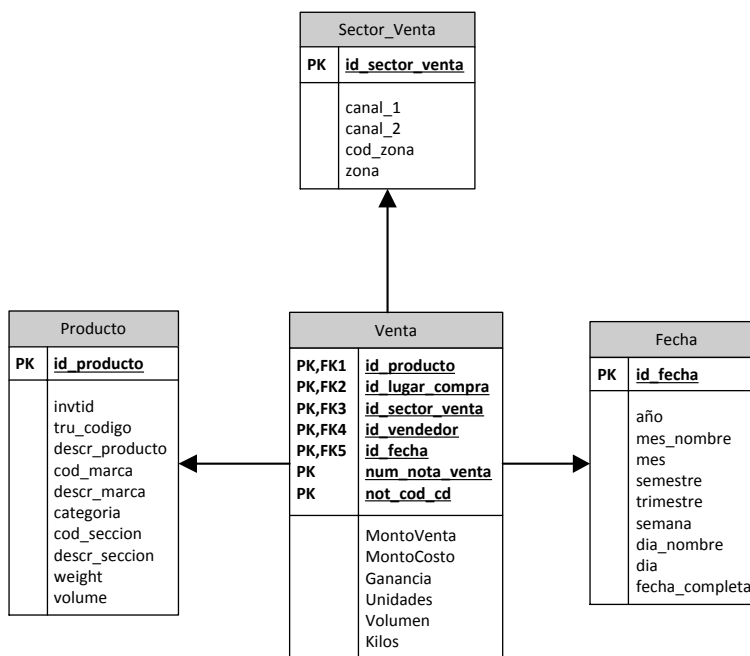


Figura 34: Modelo lógico Cubo de Ventas 1

5.5.1.1 Creación de Indicadores

En este momento se crearán seis indicadores que serán incluidos en el cubo “Cubo de Ventas 1”, los que son representados en la Tabla 6.

Indicador	Descripción	Fórmula
1. Monto Total Ventas	De la tabla de hechos “VENTAS”, se sumará el hecho “MontoVenta” para la creación de este indicador	SUM(VENTAS.MontoVenta)
2. Monto Total Costos	De la tabla de hechos “VENTAS”, se sumará el hecho “MontoCosto” para la creación de este indicador.	SUM(VENTAS.MontoCosto)
3. Ganancias	De la tabla de hechos “VENTAS”, se sumará el hecho “Ganancia” para la creación de este indicador.	SUM(VENTAS.Ganancia)
4. Cantidad Total mt3	De la tabla de hechos “VENTAS”, se sumará el hecho “Volumen” para la creación de este indicador.	SUM(VENTAS.Volumen)
5. Cantidad Total kilos	De la tabla de hechos “VENTAS”, se sumará el hecho “Kilos” para la creación de este indicador.	SUM(VENTAS.kilos)
6. Unidades Vendidas	De la tabla de hechos “VENTAS”, se sumará el hecho “Unidades” para la creación de este indicador.	SUM(VENTAS.Unidades)

Tabla 6: Indicadores del “Cubo de Ventas 1”

A continuación se aprecia los indicadores del Cubo de Ventas 1:



Cubo de Ventas 1

- Monto Total Ventas | SUM ([VENTAS.MontoVenta])
- Monto Total Costos | SUM ([VENTAS.MontoCosto])
- Ganancias | SUM ([VENTAS.Ganancia])
- Cantidad Total mt3 | SUM ([VENTAS.Volumen])
- Cantidad Total kilos | SUM ([VENTAS.Kilos])
- Unidades Vendidas | SUM ([VENTAS.Unidades])

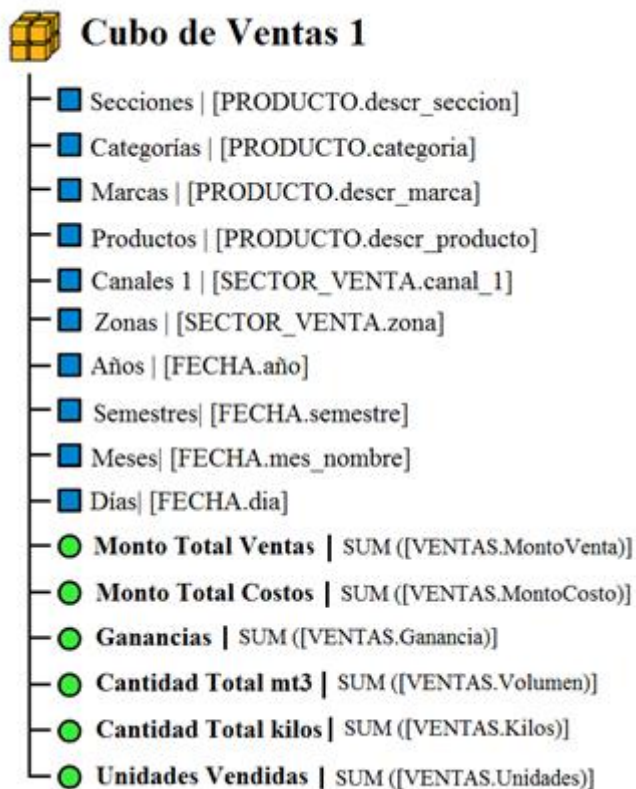
5.5.1.2 Creación de Atributos

Ahora se crearán y agregarán al cubo once atributos como se definen en la Tabla 7.

Atributo	Descripción
1. Secciones	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Descr_Seccion” para la creación del atributo
2. Categorías	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Categoria” para la creación del atributo
3. Marcas	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Descr_Marca” para la creación del atributo
4. Productos	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Descr_Producto” para la creación del atributo
5. Canales 1	De la tabla de dimensión “SECTOR_VENTA”, se tomará el campo “Canal_1” para la creación del atributo
6. Zonas	De la tabla de dimensión “SECTOR_VENTA”, se tomará el campo “Zona” para la creación del atributo
7. Años	De la tabla de dimensión “FECHA”, se tomará el campo “Año” para la creación del atributo
8. Semestres	De la tabla de dimensión “FECHA”, se tomará el campo “Semestre” para la creación del atributo
9. Meses	De la tabla de dimensión “FECHA”, se tomará el campo “Mes_Nombre” para la creación del atributo
10. Días	De la tabla de dimensión “FECHA”, se tomará el campo “Dia” para la creación del atributo denominado

Tabla 7: Atributos del “Cubo de Ventas 1”

Entonces, el cubo quedaría conformado de la siguiente manera:



5.5.1.3 Creación de Jerarquías

Finalmente se crearán y agregarán al cubo tres jerarquías.

Se definió la jerarquía “Jerarquía Productos”, que se aplicará sobre los atributos recientemente creados, “Secciones”, “Categorías”, “Marcas” y “Productos”, en donde:

- Un producto en especial pertenece sólo a una marca. Una marca puede tener uno o más productos.
- Una marca en especial pertenece sólo a una categoría. Una categoría puede tener una o más marcas.
- Una categoría en especial pertenece sólo a una sección. Y una sección puede tener una o más categorías.

Se expresa gráficamente en la Figura 35.

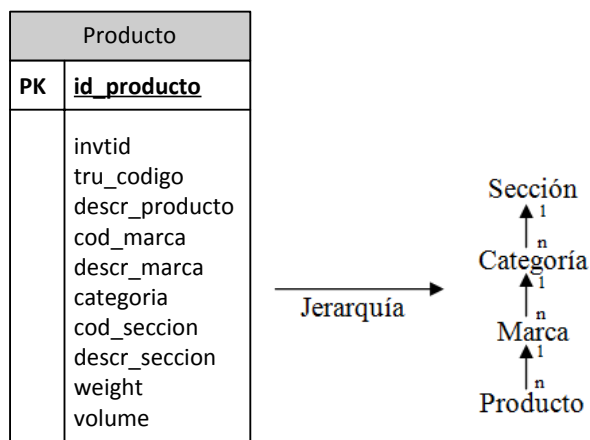


Figura 35: “PRODUCTO”, relación Padre-Hijo

Se definió la jerarquía “Jerarquía Zonas”, que se aplicará sobre los atributos recientemente creados, “Canales 1”, y “Zonas”.

Una zona en especial pertenece solo a un canal 1. Un Canal 1 puede tener uno o más zonas.

Se expresa gráficamente, en la Figura 36.

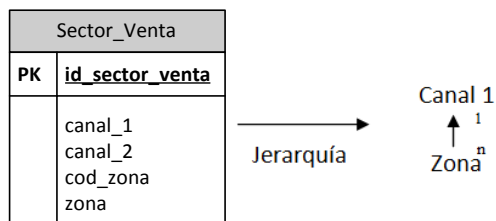


Figura 36: “SECTOR_VENTA”, relación Padre-Hijo

Se definió la jerarquía “Jerarquía Fechas”, que se aplicará sobre los atributos recientemente creados, “Años”, “Semestres”, “Meses” y “Días”.

- Un día del mes pertenece solo a un mes del año. Un mes del año tiene uno o más días del mes.
- Un mes del año pertenece solo a un semestre del año. Un semestre del año tiene uno o más meses del año.
- Un semestre del año pertenece solo a un año. Un año tiene uno o más semestres del año.

Se expresa gráficamente, en la Figura 37.

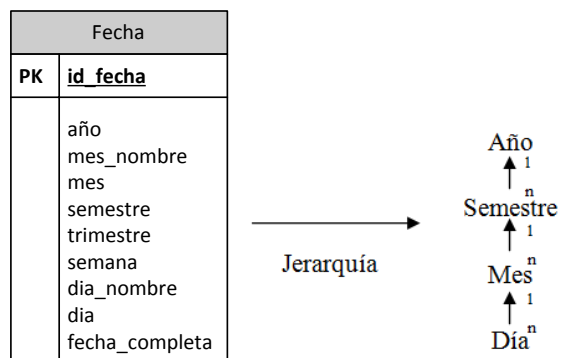


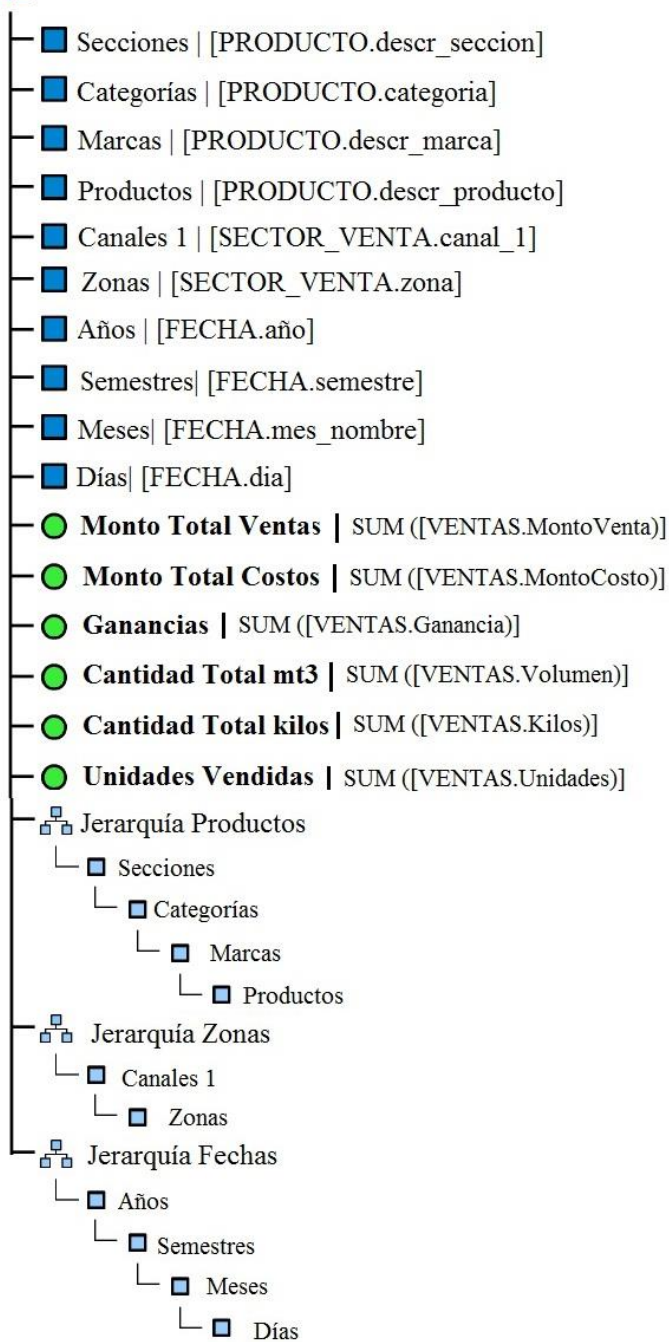
Figura 37: “FECHA”, relación Padre-Hijo

Cabe destacar que el atributo Proveedores no posee jerarquías para este caso. Esto se debe a que en el cubo irá solo, sin descendencia ni procedencia.

Entonces, el cubo quedaría conformado de la siguiente manera:



Cubo de Ventas 1



5.5.2 Cubo de Ventas 2 (Vendedor-Fecha-Sector_Venta)

Se creará otro cubo multidimensional a modo de ejemplo, basado en el modelo lógico propuesto por Hefesto [1], diseñado para el área de Soporte de Información de Rabie S.A., el cubo tendrá el nombre “Cubo de Ventas 2”, el cual estará conformado por las dimensiones de Producto, Proveedor y la Fecha, éste cubrirá lo planteado en el Capítulo 5 en el punto 5.1.2 en el cual se definen las interrogantes para las distintas áreas. A continuación se encuentra la necesidad del área de venta a satisfacer con el cubo, los indicadores estarán subrayados con rojo y las dimensiones con verde, como se aprecia a continuación:

Área de Venta:

1. Monto de ventas de las zonas por vendedor en un tiempo determinado.

Por lo que el modelo lógico queda representado en la Figura 38.

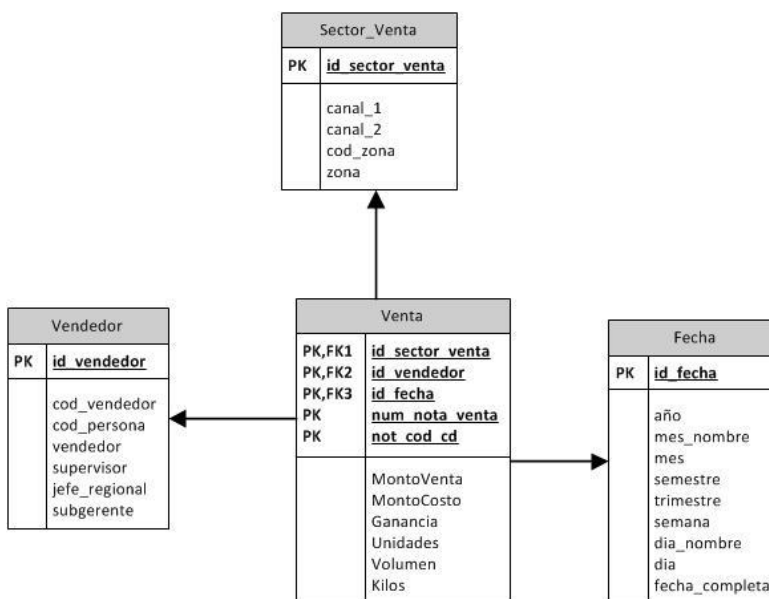


Figura 38: Modelo lógico Cubo de Ventas 2

5.5.2.1 Creación de Indicadores

En este momento se crearán dos indicadores que serán incluidos en el cubo “Cubo de Ventas 2”, los que son representados en la Tabla 8.

Indicador	Descripción	Formula
1. Monto Total Ventas	De la tabla de hechos “VENTAS”, se sumará el hecho “MontoVenta” para la creación de este indicador	SUM(VENTAS.MontoVenta)
2. Ganancias	De la tabla de hechos “VENTAS”, se sumará el hecho “Ganancia”, éste hecho se agregó al cubo para que éste entregue información de las ganancias que pueden ser de utilidad para los directivos de la empresa	SUM(VENTAS.Ganancia)

Tabla 8: Indicadores del “Cubo de Ventas 2”

A continuación se aprecia los indicadores del Cubo de Ventas 2:



Cubo de Ventas 2

- Monto Total Ventas | SUM ([VENTAS.MontoVenta])
- Ganancias | SUM ([VENTAS.Ganancia])

5.5.2.2 Creación de Atributos

Ahora se crearán y agregarán al cubo diez atributos, como se definen en la Tabla 9.

Atributo	Descripción
1. Vendedores	De la tabla de dimensión “VENDEDOR”, se tomará el campo “vendedor” para la creación del atributo
2. Supervisores	De la tabla de dimensión “VENDEDOR”, se tomará el campo “supervisor” para la creación del atributo
3. Jefes Regionales	De la tabla de dimensión “VENDEDOR”, se tomará el campo “jefe_regional” para la creación del atributo
4. Subgerentes	De la tabla de dimensión “VENDEDOR”, se tomará el campo “subgerente” para la creación del atributo
5. Canales 2	De la tabla de dimensión “SECTOR_VENTA”, se tomará el campo “Canal_2” para la creación del atributo
6. Zonas	De la tabla de dimensión “SECTOR_VENTA”, se tomará el campo “Zona” para la creación del atributo
7. Años	De la tabla de dimensión “FECHA”, se tomará el campo “Año” para la creación del atributo
8. Trimestres	De la tabla de dimensión “FECHA”, se tomará el campo “Trimestre” para la creación del atributo
9. Meses	De la tabla de dimensión “FECHA”, se tomará el campo “Mes_Nombre” para la creación del atributo
10. Días	De la tabla de dimensión “FECHA”, se tomará el campo “Dia” para la creación del atributo

Tabla 9: Atributos del “Cubo de Ventas 2”

Entonces, el cubo quedaría conformado de la siguiente manera:



Cubo de Ventas 2



5.5.2.3 Creación de Jerarquías

Finalmente se crearán y agregarán al cubo cuatro jerarquías.

Se definió la jerarquía “Jerarquía Vendedores”, que se aplicará sobre los atributos recientemente creados, “Gerentes”, “Jefes Regionales”, “Supervisores” y “Vendedores”, en donde:

- Un vendedor en especial pertenece solo a un supervisor. Un supervisor puede tener uno o más vendedores.
- Un supervisor en especial pertenece solo a un jefe regional. Un jefe regional puede tener una o más supervisores.
- Un jefe regional en especial pertenece solo a una gerente. Y un gerente puede tener un o más jefes regionales.

Se expresa gráficamente en la Figura 39.

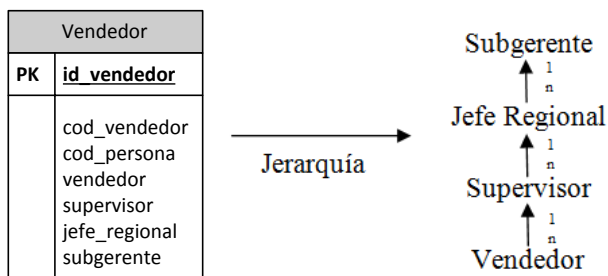


Figura 39: “VENDEDOR”, relación Padre-Hijo

Se definió la jerarquía “Jerarquía Zonas”, que se aplicará sobre los atributos recientemente creados, “Canales 2”, y “Zonas”.

Una zona en especial pertenece solo a un canal 2. Un Canal 2 puede tener uno o más zonas.

Se expresa gráficamente, en la Figura 40.

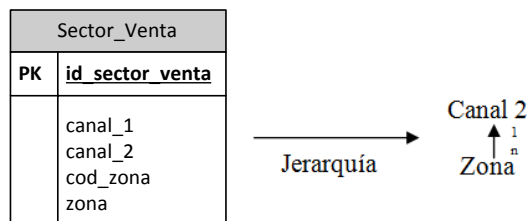


Figura 40: “SECTOR_VENTA”, relación Padre-Hijo (2)

Se definió la jerarquía “Jerarquía Fechas”, que se aplicará sobre los atributos recientemente creados, “Años”, “Trimestres”, “Meses” y “Días”.

- Un día del mes pertenece solo a un mes del año. Un mes del año tiene uno o más días del mes.
- Un mes del año pertenece solo a un trimestre del año. Un trimestre del año tiene uno o más meses del año.
- Un trimestre del año pertenece solo a un año. Un año tiene uno o más trimestres del año.

Se expresa gráficamente, en la Figura 41.

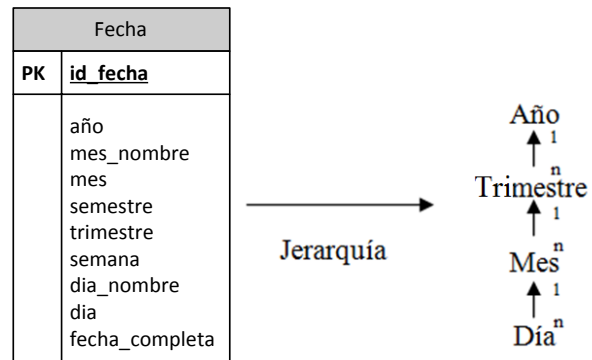
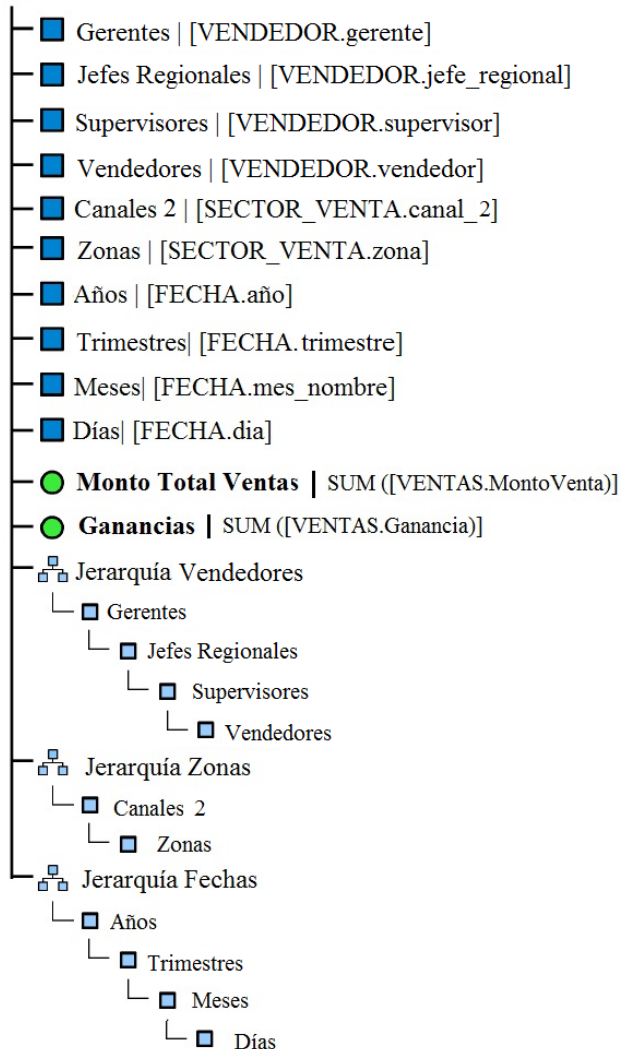


Figura 41: "FECHA", relación Padre-Hijo (2)

Entonces, el cubo quedaría conformado de la siguiente manera:



Cubo de Ventas 2



5.5.3 Cubo de Ventas 3 (Producto-Fecha-Sector_Venta)

Se creará un cubo multidimensional de ejemplo, basado en el modelo lógico propuesto por Hefesto [1], diseñado para el área de Soporte de Información de Rabie S.A., el cubo tendrá el nombre “Cubo de Ventas 1”, el cual estará conformado por las dimensiones de Producto, Sector_Venta y Fecha, éste cubrirá lo planteado en el Capítulo 5 en el punto 5.1.2 en el cual se definen las interrogantes para las distintas áreas que a continuación se presentarán, en las cuales los indicadores estarán subrayados con rojo y las dimensiones con verde.

Área de Venta:

1. Monto de ventas, el monto de costos y la ganancia categorías de productos por lugar de compra en un tiempo determinado.

Por lo que el modelo lógico queda representado en la Figura 42.

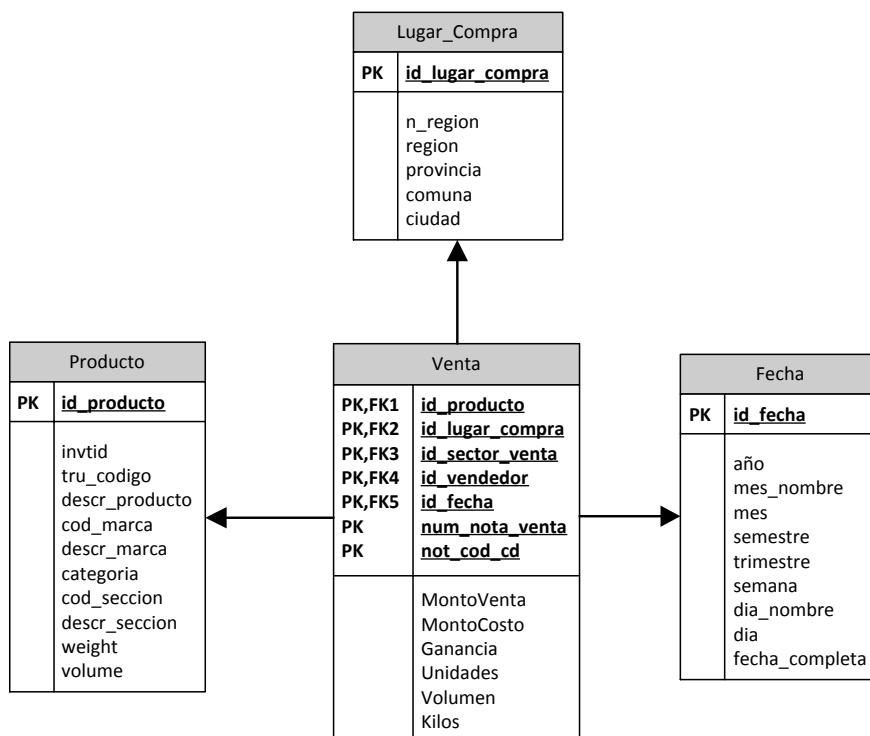


Figura 42: Modelo lógico Cubo de Ventas 3

5.5.3.1 Creación de Indicadores

En este momento se crearán tres indicadores que serán incluidos en el cubo “Cubo de Ventas 3”, los que son representados en la Tabla 10.

Indicador	Descripción	Fórmula
1. Monto Total Costos	De la tabla de hechos “VENTAS”, se sumará el hecho “MontoCosto” para la creación de este indicador.	SUM(VENTAS.MontoCosto)

2. Ganancias	De la tabla de hechos “VENTAS”, se SUM(VENTAS.Ganancia) sumará el hecho “Ganancia” para la creación de este indicador.
3. Unidades Vendidas	De la tabla de hechos “VENTAS”, se SUM(VENTAS.Unidades) sumará el hecho “Unidades”, el indicador fue agregado al cubo con el objetivo de obtener un análisis más completo de la información almacenada en el cubo

Tabla 10: Indicadores del “Cubo de Ventas 3”

A continuación se aprecia los indicadores del Cubo de Ventas 3:



Cubo de Ventas 3

- **Monto Total Ventas** | SUM ([VENTAS.MontoVenta])
- **Ganancias** | SUM ([VENTAS.Ganancia])
- **Unidades Vendidas** | SUM ([VENTAS.Unidades])

5.5.3.2 Creación de Atributos

Ahora se crearán y agregarán al cubo once atributos, como se definen en la Tabla 11.

Atributo	Descripción
1. Secciones	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Descr_Seccion” para la creación del atributo
2. Categorías	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Categoria” para la creación del atributo
3. Marcas	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Descr_Marca” para la creación del atributo

4. Productos	De la tabla de dimensión “PRODUCTO”, se tomará el campo “Descr_Producto” para la creación del atributo
5. Regiones	De la tabla de dimensión “LUGAR_COMPRA”, se tomará el campo “region” para la creación del atributo
6. Provincias	De la tabla de dimensión “LUGAR_COMPRA”, se tomará el campo “provincia” para la creación del atributo
7. Comunas	De la tabla de dimensión “LUGAR_COMPRA”, se tomará el campo comuna” para la creación del atributo
8. Ciudades	De la tabla de dimensión “LUGAR_COMPRA”, se tomará el campo “ciudad” para la creación del atributo
9. Años	De la tabla de dimensión “FECHA”, se tomará el campo “Año” para la creación del atributo
10. Meses	De la tabla de dimensión “FECHA”, se tomará el campo “Mes_Nombre” para la creación del atributo
11. Días	De la tabla de dimensión “FECHA”, se tomará el campo “Dia” para la creación del atributo

Tabla 11: Atributos del “Cubo de Ventas 3”

Entonces, el cubo quedaría conformado de la siguiente manera:



Cubo de Ventas 3

- Secciones | [PRODUCTO.descr_seccion]
- Categorías | [PRODUCTO.categoria]
- Marcas | [PRODUCTO.descr_marca]
- Productos | [PRODUCTO.descr_producto]
- Regiones | [LUGAR_COMPRA.region]
- Provincias | [LUGAR_COMPRA.provincia]
- Comunas | [LUGAR_COMPRA.comuna]
- Ciudades | [LUGAR_COMPRA.ciudad]
- Años | [FECHA.año]
- Meses | [FECHA.mes_nombre]
- Días | [FECHA.día]
- **Monto Total Ventas** | SUM ([VENTAS.MontoVenta])
- **Ganancias** | SUM ([VENTAS.Ganancia])
- **Unidades Vendidas** | SUM ([VENTAS.Unidades])

5.5.3.3 Creación de Jerarquías

Finalmente se crearán y agregarán al cubo tres jerarquías.

Se definió la jerarquía “Jerarquía Productos”, que se aplicará sobre los atributos recientemente creados, “Secciones”, “Categorías”, “Marcas” y “Productos”, en donde:

- Un producto en especial pertenece solo a una marca. Una marca puede tener uno o más productos.
- Una marca en especial pertenece solo a una categoría. Una categoría puede tener una o más marcas.
- Una categoría en especial pertenece solo a una sección. Y una sección puede tener una o más categorías.

Se expresa gráficamente en la Figura 35.

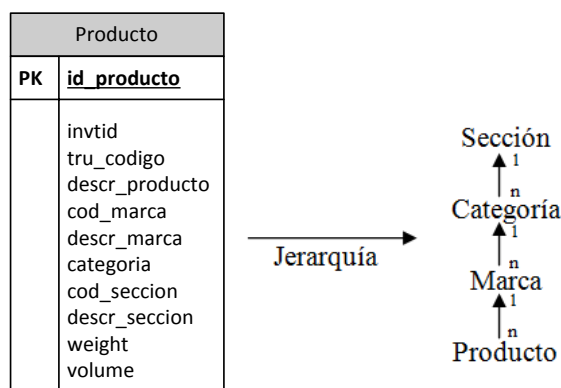


Figura 43: “PRODUCTO”, relación Padre-Hijo

Se definió la jerarquía “Jerarquía Geográfica”, que se aplicará sobre los atributos recientemente creados, “Regiones”, “Provincias”, “Comunas” y “Ciudades”.

- Una ciudad en especial pertenece solo a una comuna. Una comuna puede tener una o más ciudades.
- Una comuna en especial pertenece solo a una provincia. Una provincia puede tener una o más comunas.

- Una provincia en especial pertenece solo a una región. Y una región puede tener una o más comunas.

Se expresa gráficamente, en la Figura 44.

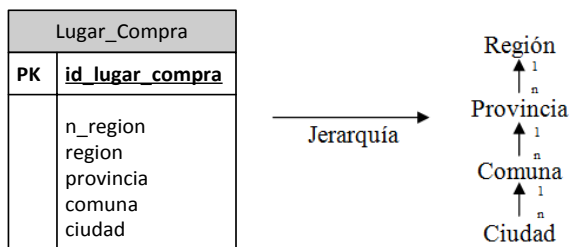


Figura 44: “LUGAR_COMPRA”, relación Padre-Hijo

Se definió la jerarquía “Jerarquía Fechas”, que se aplicará sobre los atributos recientemente creados, “Años”, “Meses” y “Días”.

- Un día del mes pertenece solo a un mes del año. Un mes del año tiene uno o más días del mes.
- Un mes del año pertenece solo a un año. Un año tiene uno o más meses del año.

Se expresa gráficamente, en la Figura 45.

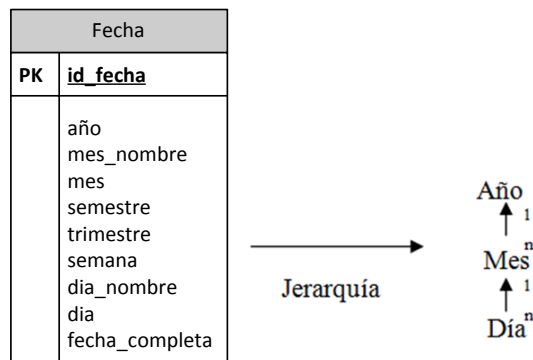
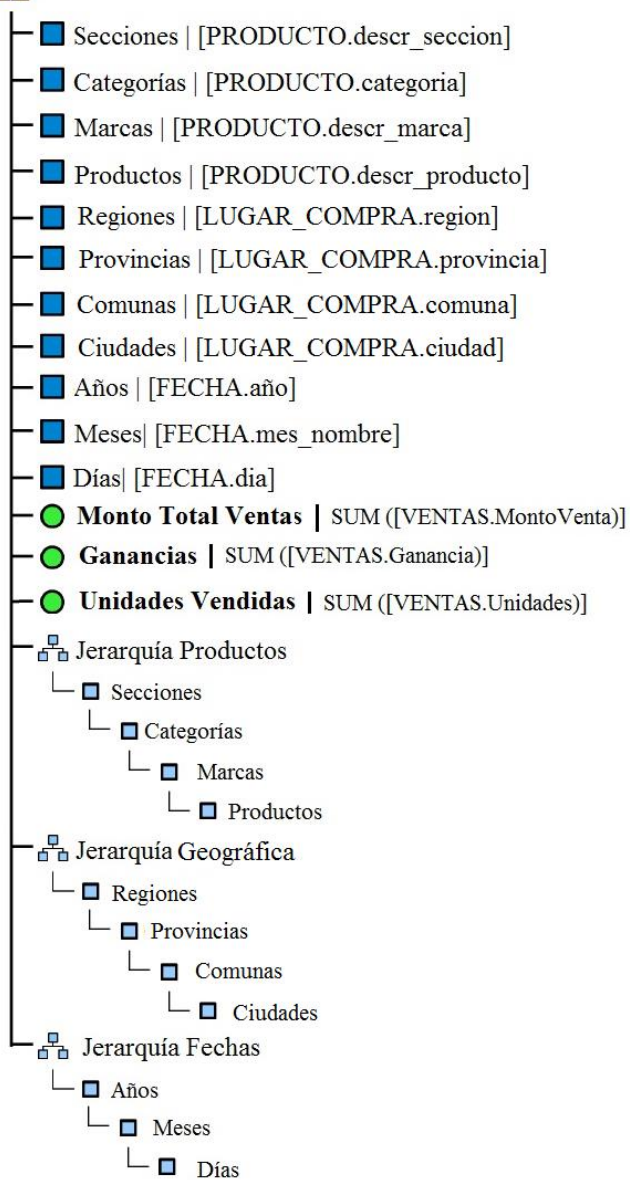


Figura 45: “FECHA”, relación Padre-Hijo (3)

Entonces, el cubo quedaría conformado de la siguiente manera:



Cubo de Ventas 3



Capítulo 6

Tipos de herramientas de consulta y análisis

Finalmente en este capítulo, se expondrán herramientas adicionales que son utilizadas para la consulta y análisis de la información almacenada en un Data Warehouse, las cuales complementan la operatividad del funcionamiento de este sistema. Se utilizará como ejemplo una de éstas, de la cual se describirá su proceso de utilización para la creación de cubos OLAP.

6.1 Introducción

Las herramientas de consulta y análisis son sistemas que permiten al usuario realizar la exploración de datos del Data Warehouse. Básicamente constituyen el nexo entre el depósito de datos y los usuarios.

A través de una amigable interfaz gráfica y una serie de pasos simples, el usuario genera consultas que son enviadas desde la herramienta de consulta y análisis al Query Manager, este a su vez realiza la extracción de información al Data Warehouse Manager y devuelve los resultados obtenidos a la herramienta que se los solicitó. Luego, estos resultados son expuestos ante el usuario en formatos que le son familiares [19].

Básicamente el proceso de consulta y análisis se lleva a cabo a través de seis pasos sucesivos [1]:

1. El usuario selecciona o establece qué datos desea obtener del Data Warehouse, mediante las interfaces de la herramienta que utilice.

2. La herramienta recibe el pedido del usuario, construye la consulta y la envía al Query Manager.
3. El Query Manager ejecuta la consulta sobre el cubo multidimensional con que se esté trabajando (todas las consultas al Data Warehouse se hacen a través de algún cubo multidimensional).
4. El Query Manager obtiene los resultados de la consulta.
5. El Query Manager envía los datos a las herramientas de consulta y análisis.
6. Las herramientas presentan al usuario la información requerida.

Cabe señalar que una de las principales ventajas para el área de Soporte de Información al utilizar estas herramientas, es que los usuarios no se tienen que preocupar por conocer cuál es la estructura de los datos, ni por saber emplear el lenguaje SQL, solo se deben enfocar en el análisis, lo que permitiría aumentar los niveles de conocimiento de las áreas que se alimentan de la información generada por dicha área.

A continuación se presentarán diferentes tipos de herramientas de consulta y análisis, con sus características actuales. Además de la factibilidad de adquisición de éstas.

6.2 Consulta y Reportes

Se han desarrollado para los usuarios una gran cantidad de poderosas herramientas de consulta y reporte en el mercado a través de pantallas gráficas intuitivas, la posibilidad de generar informes avanzados y detallados del área de interés del negocio que se esté analizando.

Para hacer consultas más accesibles a usuarios no avanzados en el uso de estas tecnologías, con herramientas que ofrecen interfaces gráficas para seleccionar, arrastrar y pegar, se presentan algunas de las herramientas más comunes utilizadas por los usuarios:

- Crystal Reports de SAP
- Impromptu de Cognos, Reportsmith de Borland

- Intelligent Query de IQ Software
- Esperant de Software AG

Es por ello que a continuación se presentarán detalles más técnicos de una de las herramientas más utilizadas para realizar consultas y reportes, con el objetivo de entregar mayor conocimiento al área de Soporte de Información con respecto al uso de estas tecnologías.

6.2.1 Crystal Reports

Es una aplicación del ámbito Business Intelligent utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos. Esta herramienta está orientada para potenciar a los usuarios de pequeñas y medianas empresas.

SAP Crystal Reports entrega además una solución integral de generación de informes que ayuda a los diseñadores de informes, a los profesionales de TI, a los usuarios avanzados y a los desarrolladores de aplicaciones a diseñar, explorar, visualizar y proporcionar informes a través de Internet o de manera integrada en aplicaciones empresariales.

Con SAP Crystal Reports, los usuarios finales pueden tener acceso a informes con una presentación impecable, realizar modelados empresariales en los informes y tomar decisiones de manera inmediata desde el mismo informe, reduciendo de este modo la dependencia de las TI y de los desarrolladores. A continuación se presentan las principales características del producto [20].

- Integración de SAP Crystal Dashboard Design, Adobe Flex y Adobe Flash.
- Visualización de informes interactiva.
- Controlador de datos de servicios Web mejorado.
- Funciones de diseño mejoradas.
- Controlador salesforce.com integrado.
- Exportación de XML mejorada.

- SDK de modificación de informes NET.
- Publicación de informes avanzada.

Información adicional:

- Página Oficial del Producto: <http://www.sap.com>
- Idioma de la aplicación: español, inglés.
- Año de última versión: 2008.
- Permite descargar una versión de prueba de 30 días.
- Precio: 479,00 EUR.
- Precio en Chile: 298.615,94 pesos.

6.3 Herramientas OLAP

Las herramientas OLAP, se conocen como sistemas analíticos, estos sistemas pueden encontrarse integrados tanto en suites Business Intelligent o ser simplemente una aplicación independiente.

Se entiende por OLAP, o proceso analítico en línea, al método ágil y flexible para organizar datos, específicamente metadatos, sobre un objeto o jerarquía de objetos, como un sistema u organización multidimensional, y cuyo objetivo es recuperar y manipular datos y combinaciones de los mismos a través de consultas o incluso informes.

Una herramienta OLAP está formada por un motor y un visor. El motor es, en realidad, justo el concepto que acabamos de definir. El visor OLAP es una interfaz que permite consultar, manipular, reordenar y filtrar datos existentes en una estructura OLAP mediante una interfaz gráfica de usuario [21].

Las estructuras OLAP permiten realizar consultas que serían sumamente complejas en el lenguaje SQL. Lo que aporta verdaderos beneficios al aplicarlas al Data Warehouse propuesto en este proyecto, debido a que proporciona libertad a los usuarios finales para realizar consultas de forma independiente del área de Soporte de Información.

Además de las características ya descritas, se pueden enumerar las siguientes [1]:

- Permite recolectar y organizar la información analítica necesaria para los usuarios y disponer de ella en diversos formatos, tales como tablas, gráficos, reportes, dashboards, etc.
- Soporta análisis complejos de grandes volúmenes de datos.
- Complementa las actividades de otras herramientas que requieran procesamiento analítico en línea.
- Presenta al usuario una visión multidimensional de los datos para cada tema de interés del negocio.
- Es transparente al tipo de tecnología que soporta el Data Warehouse, ya sea ROLAP, MOLAP u HOLAP.
- No tiene limitaciones con respecto al número máximo de dimensiones permitidas.
- Permite a los usuarios, analizar la información basándose en más criterios que un análisis de forma tradicional.
- Al contar con muestras grandes, se pueden explorar mejor los datos en busca de respuestas.
- Permiten realizar agregaciones y combinaciones de los datos de maneras complejas y específicas, con el fin de realizar análisis más estratégicos.

Dentro de las herramientas gratuitas para generar reportes OLAP se destacan las tres siguientes:

- OlapCube
- JPivot
- Mondrian

A continuación se presentará una herramienta de este tipo, de nombre OlapCube, herramienta con la que se detallará su proceso de utilización para la creación de cubos OLAP, basados en el Data Warehouse propuesto en esta memoria. El objetivo es guiar y demostrar al área de Soporte de Información la efectividad de esta herramienta en el análisis del negocio desde diferentes escenarios históricos, en un ambiente multidimensional, o sea, mediante la combinación de diferentes dimensiones o temas de interés.

6.3.1 OlapCube

OlapCube es una poderosa y sencilla herramienta para analizar datos y luego compartirlos en línea. Esta herramienta permite crear cubos multidimensionales localmente en el equipo.

Dentro de un resumen de sus características avanzadas, se pueden encontrar las siguientes.

- No necesita tener preinstalado SQL Server u otro software OLAP, para su funcionamiento.
- Los cubos generados son de carácter estándar, se almacenan en archivos con una extensión .CUB, la cual se puede abrir desde un Excel, lo cual es muy común en este entorno.
- Cubre todas las principales bases de datos y archivos de datos, como lo son SQL Server, Oracle, IBM DB2, MySQL, Sybase, Informix, PostgreSQL, IBM UniVerse, IBM UniData, Progress, Firebird, Interbase, Paradox, Ingres, Mimer SQL, Pervasive, SQLBase, Caché, Teradata, DBMaker, Valentina, Excel, Access, Archivos de texto, Visual FoxPro, SQLite, Filemaker.
- Posee un lector de cubos que ofrece una manera muy flexible e intuitiva para navegar por los datos.
- Posee óptimos tiempos de respuesta y permite que los datos estén disponibles con la base de datos desconectada.
- Incluye la función Dashboard, la cual es una aplicación de carácter Business Intelligence que permite a una organización visualizar la información más importante para monitorear, analizar y administrar el desempeño del negocio de manera más efectiva, ayudando a las organizaciones a optimizar su desempeño y alcanzar sus objetivos estratégicos.
- Permite enviar Dashboards por email, cargarlos en un servidor FTP, o simplemente publicarlos en un blog.

- En Excel, los cubos se cargan instantáneamente, permitiendo una visual parecida a las tablas dinámicas con sus respectivos gráficos. No es necesario cargar los datos ni organizarlos.

Información adicional:

- Página Oficial del Producto: <http://www.olapcube.com/>
- Idioma de la aplicación: inglés.
- Tamaño: 8.0 MB
- Año de última versión: 2010.
- No se requiere licencia para navegar por los cubos o para crear cuadros de mando.
- Una licencia es necesaria únicamente para los usuarios que crean los cubos, aunque de igual manera se pueden crear cubos en conjunto con un aviso que indica que es necesaria la licencia.
- Precio licencia: \$ 99.00 USD
- Precio en Chile: 46.830,65 pesos.

Debido a lo útil, eficiente y económica que es esta herramienta, se seleccionó para realizar una implementación de prueba, con datos reales de la empresa Rabie S.A., implementación que se detallará paso a paso en el siguiente punto.

6.3.1.1 Implementación de OlapCube en un Data Warehouse

Para la implementación de la herramienta OlapCube, se utilizaron datos reales de la empresa Rabie S.A. facilitados por el área de Soporte de Información, datos que corresponden a los tres primeros días, de los meses agosto, septiembre y octubre del año 2012. El motivo de solo tener información de los días mencionados anteriormente se debe al gran volumen de datos que maneja esta área. Estos datos pasaron por todos los procesos de ETL, necesarios para poblar el Data Warehouse propuesto en el Capítulo 5.

Por lo que se procederá a indicar los pasos desde la instalación y conexión del OlapCube con el Data Warehouse ya instalado y cargado de datos, hasta los resultados obtenidos y procesados por dicha herramienta.

Para ello, es necesario llevar a cabo cinco pasos que se detallan a continuación.

6.3.1.1.1 Paso 1: Instalación del programa

Abrir el instalador descargado del siguiente enlace:

http://www.olapcube.com/tr_download.asp.

Seguido de lo anterior, saldrá la pantalla de instalación, como lo muestra la Figura 46, en la cual se debe hacer clic en Next.

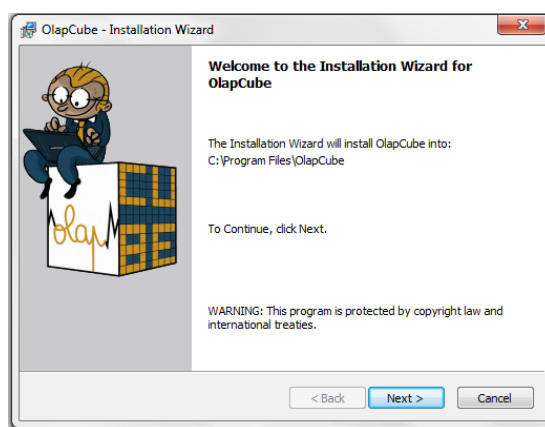


Figura 46: Pantalla de Bienvenida del Instalador de OlapCube

Luego se instalará el programa y finalizará la instalación como se muestra en la Figura 47.

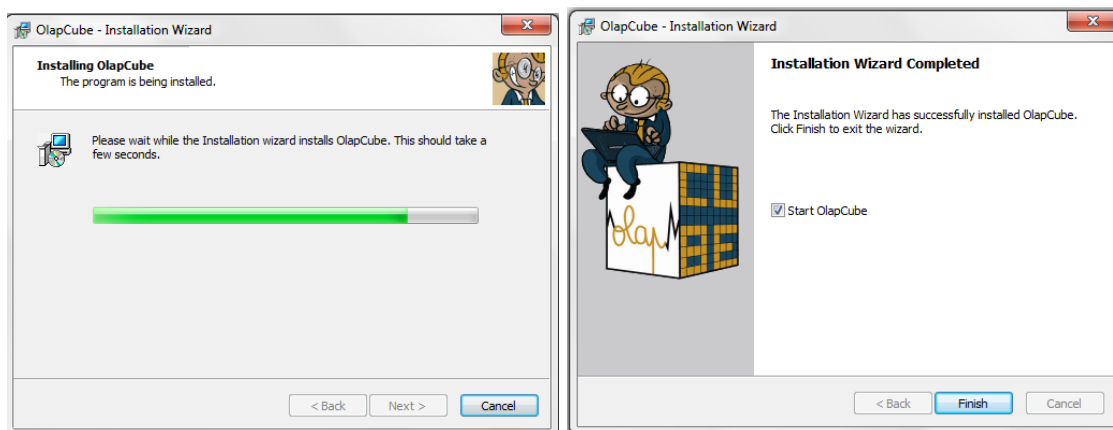


Figura 47: Pantalla de progreso y finalización de instalación

6.3.1.1.2 Paso 2: Conectar a fuente de datos

Luego de abrir el programa ya instalado, se debe hacer clic en el círculo rojo que se muestra en la Figura 48.

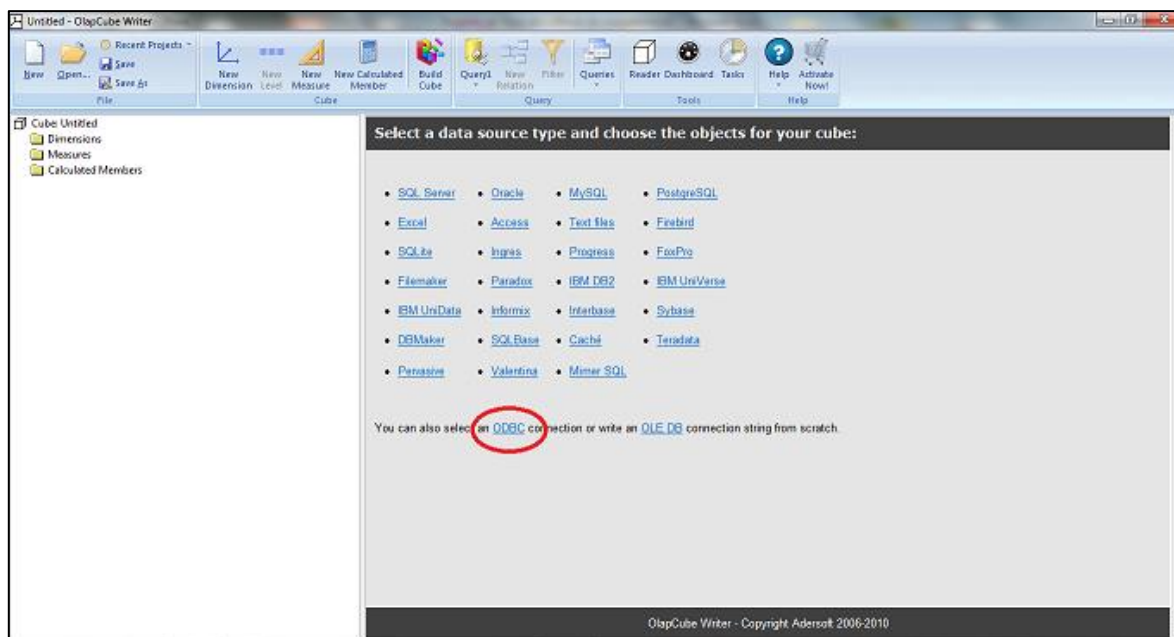


Figura 48: Pantalla de Inicio

Se debe hacer clic en Data Source Administrator, luego en Agregar y elegir en la lista el tipo de base de datos a conectar, en este caso SQL Native Client y clic en finalizar. Proceso que se puede apreciar en la Figura 49.

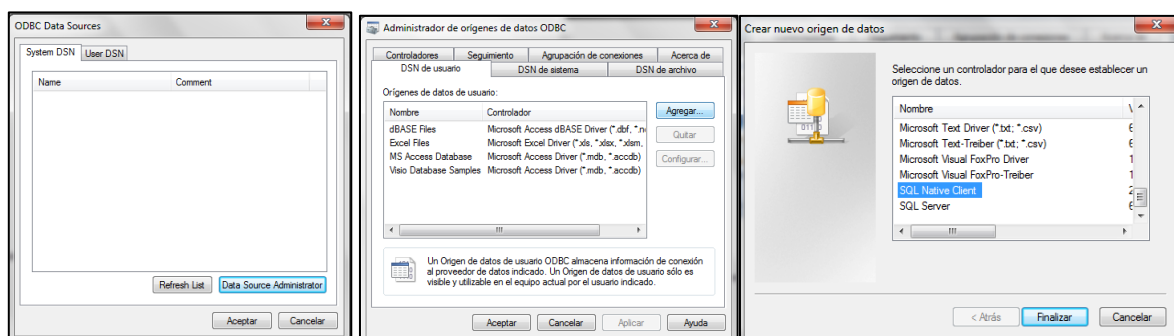


Figura 49: Pantalla de Selección del origen de datos

Aparecerá una pantalla en la cual es necesario ingresar el nombre con el que se referirá al origen de la base de datos y el servidor, luego se ingresa manualmente el nombre de Data Warehouse y en este caso el servidor EQUIPO\SQLEXPRESS. Luego se presiona siguiente y siguiente, como lo muestra la Figura 50.

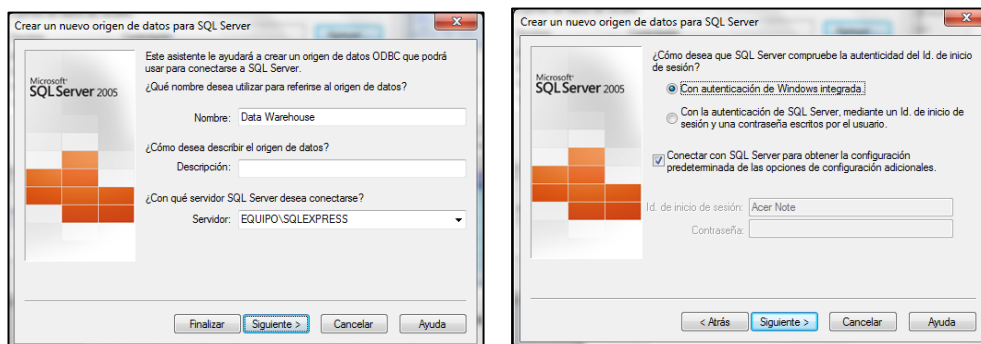


Figura 50: Pantalla de Selección del origen de Datos (2)

A continuación se debe marcar la primera opción para seleccionar la base de datos predeterminada, en este caso Data Warehouse, y siguiente. Luego finalizar como lo indica la Figura 51.

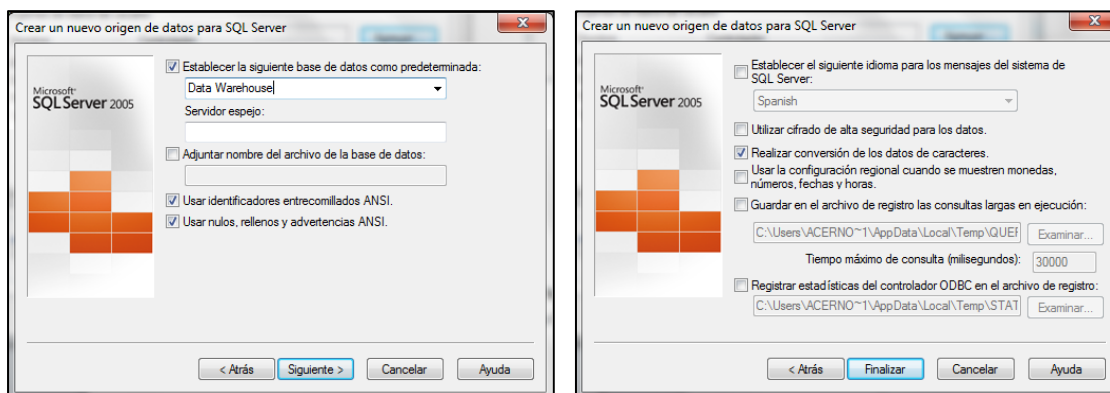


Figura 51: Pantalla de Selección del origen de Datos (3)

Para finalizar, se vuelve a la primera pantalla, en la cual seleccionaremos la pestaña User DNS. Se puede apreciar que ya está el Data Warehouse en la lista, se selecciona y acepta, luego de esto, se marcan las tablas del Data Warehouse a cargar y se acepta, como se ve en la Figura 52.

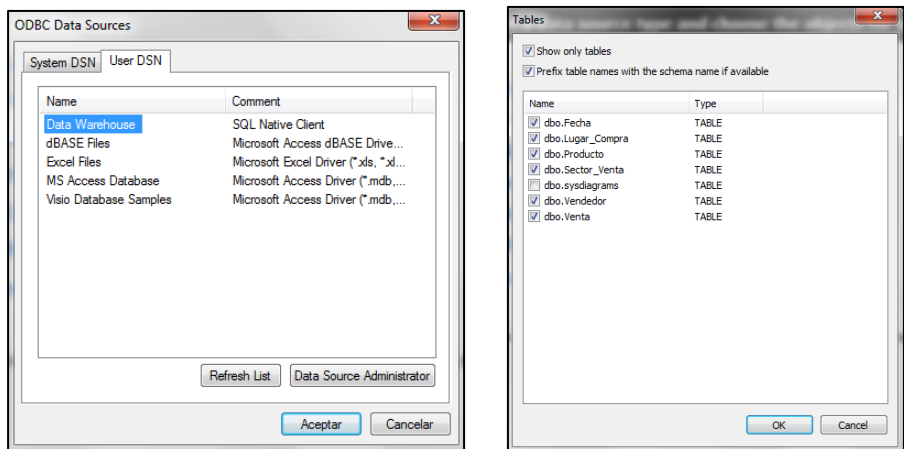


Figura 52: Pantalla de Selección del origen de Datos (4)

De tal forma obtenemos el Data Warehouse conectado al OLAPCUBE con sus respectivas relaciones, donde ya se muestra una vista previa de los datos en la parte inferior derecha, como se ve en la Figura 53.

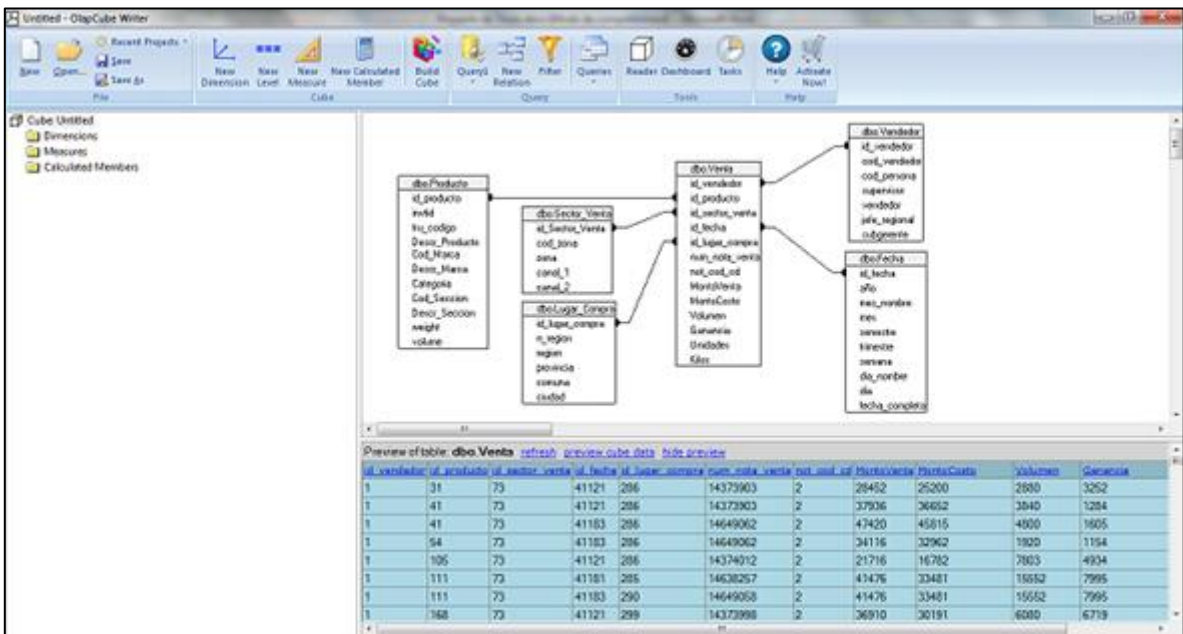


Figura 53: Pantalla de OLAP conectado con el Data Warehouse

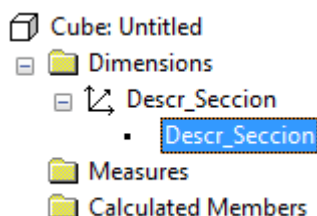
6.3.1.1.3 Paso 3: Conformar el cubo


La herramienta OlapCube permite arrastrar y soltar, además de la utilización de botones. Por lo que para generar las dimensiones, se arrastran los atributos hacia las dimensiones, respetando las jerarquías.

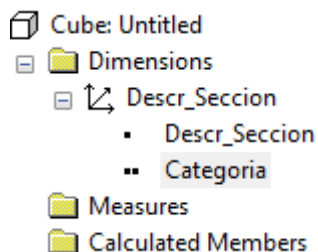
Para conformar el cubo, se utilizará como modelo el Cubo de Ventas 3, del Capítulo 5 punto 5.5.3, basándose estrictamente en los indicadores y dimensiones ahí mencionadas.

Para el ejemplo se realiza el siguiente procedimiento:

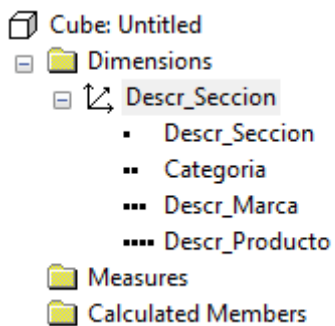
Primero se arrastra el atributo “descr_seccion” de la dimensión PRODUCTO a la carpeta Dimensions del costado izquierdo. Se eligió este atributo, debido a que es el primer atributo de la jerarquía. Lo que creará la primera dimensión de nombre “descr_seccion”.



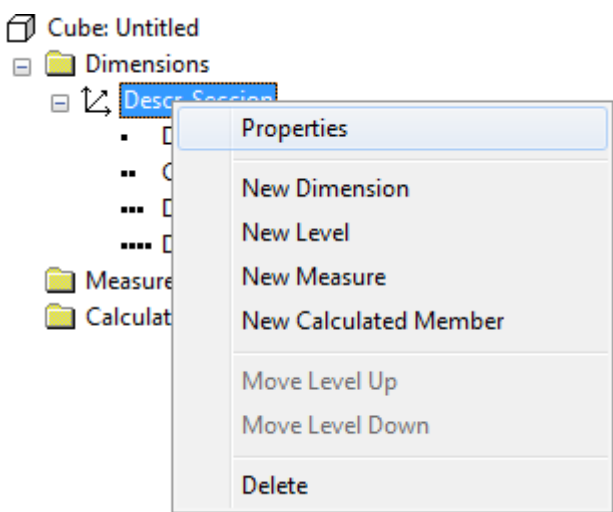
Luego arrastrar sobre la dimensión creada  Descr_Seccion el siguiente atributo, respetando la jerarquía, en este caso, corresponde al atributo “descr_categoria”.



Y así sucesivamente hasta obtener la dimensión con sus respectivos atributos, como se muestra a continuación:



Posterior a esto, para cambiar el nombre de la dimensión, se debe hacer clic derecho sobre la dimensión y seleccionar Properties.



Aparecerá una ventana que permite cambiar el nombre de la dimensión, además de elegir el tipo de dimensión que es, en este caso es ESTÁNDAR y en el caso de la dimensión Fecha es tipo TIME. Hacer lo mismo para los atributos, como se ve en la Figura 54: Ventana de Características de la Dimensión

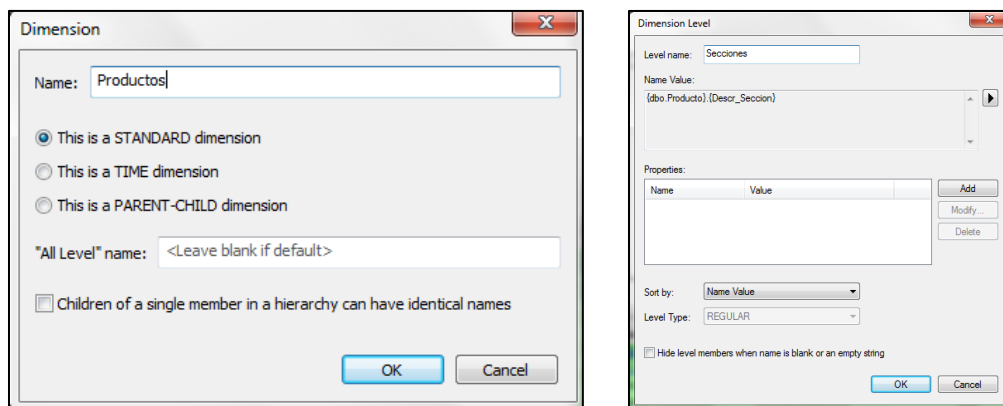
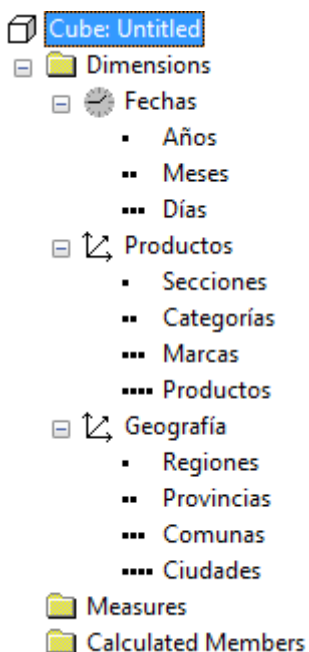
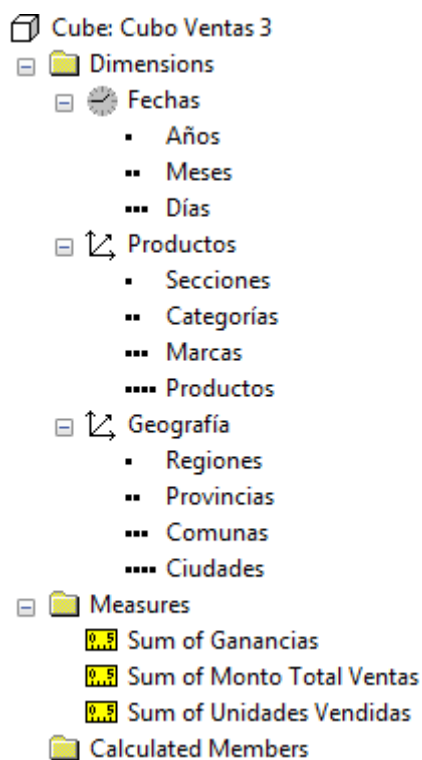


Figura 54: Ventana de Características de la Dimensión y Atributos

Se debe realizar el mismo procedimiento para las otras dos dimensiones, obteniendo el resultado que se muestra a continuación:



Luego agregar los indicadores, para ello se arrastran hasta la carpeta Measures, y se editan sus nombres de la misma forma que en el punto anterior, obteniendo finalmente esto:



Finalmente, se procede a construir el cubo, haciendo clic en Build Cube, en el círculo rojo como lo indica la Figura 55: Pantalla de Creación del Cubo. Cuando el cubo esté listo aparecerá un aviso indicando lo anterior más un recordatorio sobre la necesidad de comprar la licencia para crear cubos. Se debe hacer clic en NO para continuar con la carga del cubo.

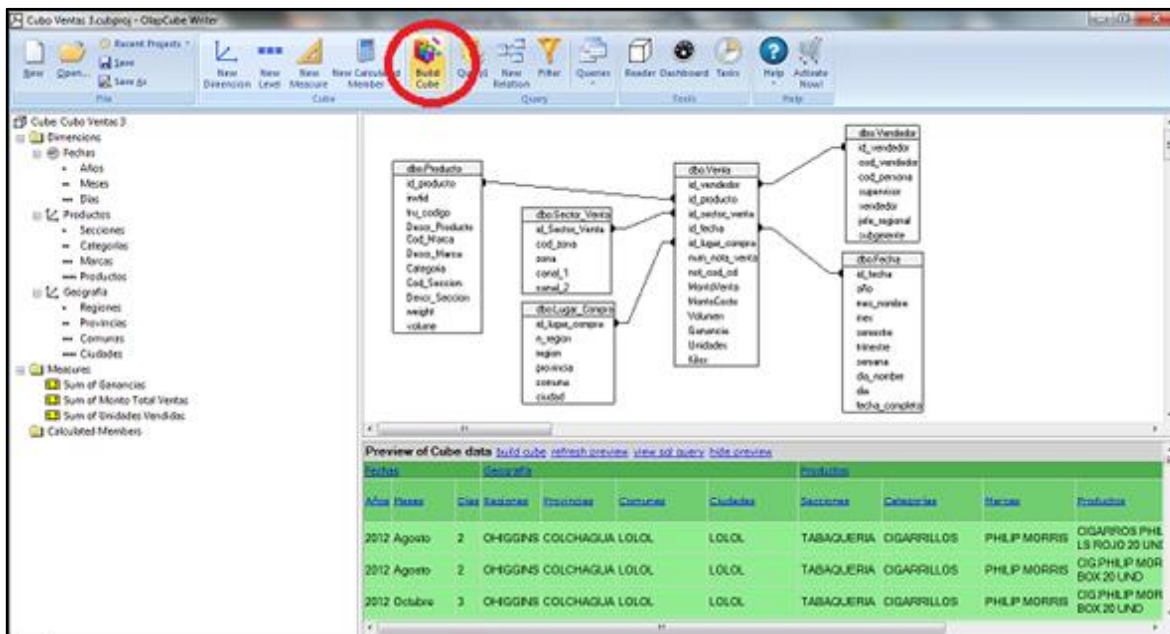


Figura 55: Pantalla de Creación del Cubo

Cabe señalar que el cubo generado en la prueba estaba compuesto por 300 mil registros, tomando un tiempo de generación de aproximadamente 7 minutos, prueba realizada en un equipo de uso domestico con las siguientes características:

Procesador: Intel Pentium IV 2.6 Ghz

Memoria: 1Gb Ram

Disco Duro: 160 Gb

6.3.1.1.4 Paso 4: Análisis y exploración del Cubo con OlapCube Reader

Para la manipulación del cubo se abre automáticamente luego del paso anterior, la herramienta OlapCube Reader, la que permite navegar de forma expedita por las dimensiones del cubo, apreciando a la vez los indicadores seleccionados, a continuación se presentan algunas capturas de las distintas funcionalidades.

En la Figura 56 se muestra la dimensión FECHA, con el indicador Ganancias.

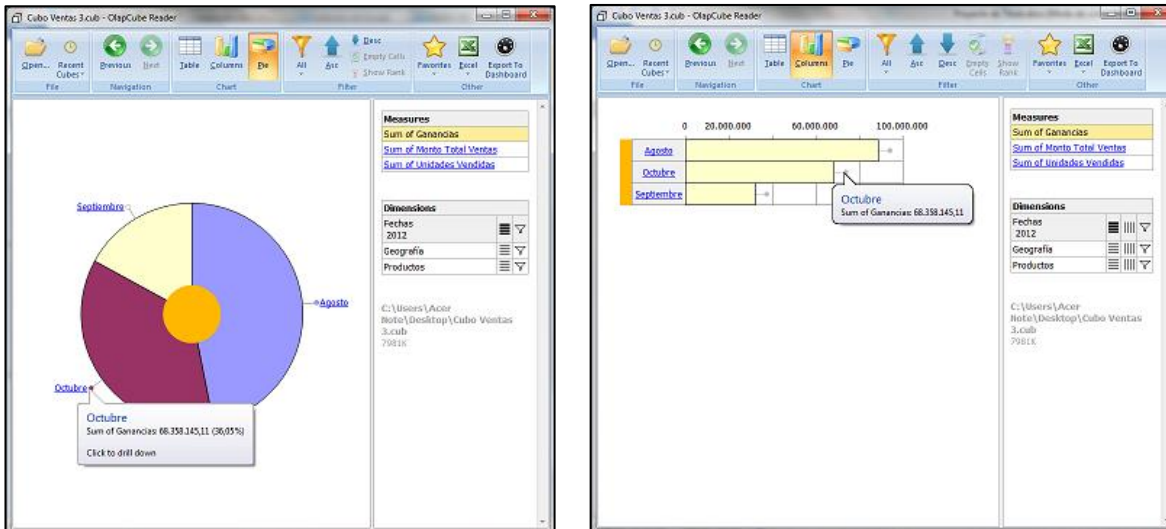


Figura 56: Pantalla de Análisis Dimensión FECHA

En la Figura 57 se muestra la dimensión LUGAR_COMPRA o Geografía, con el indicador Monto Total Ventas.

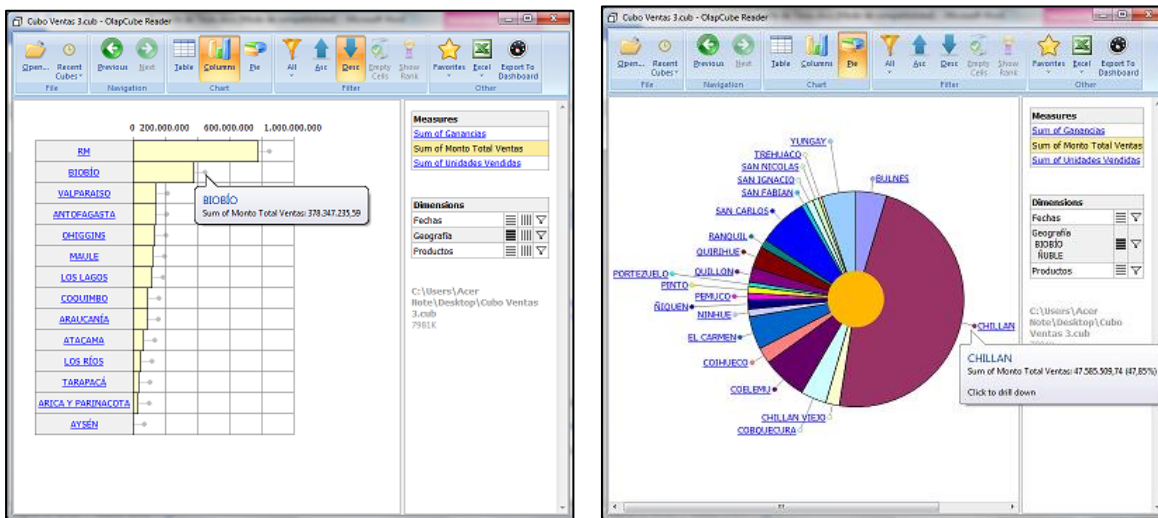


Figura 57: Pantalla de Análisis Dimensión LUGAR_COMPRA

Y en la Figura 58 se muestra la dimensión PRODUCTO, con el indicador Unidades Vendidas.

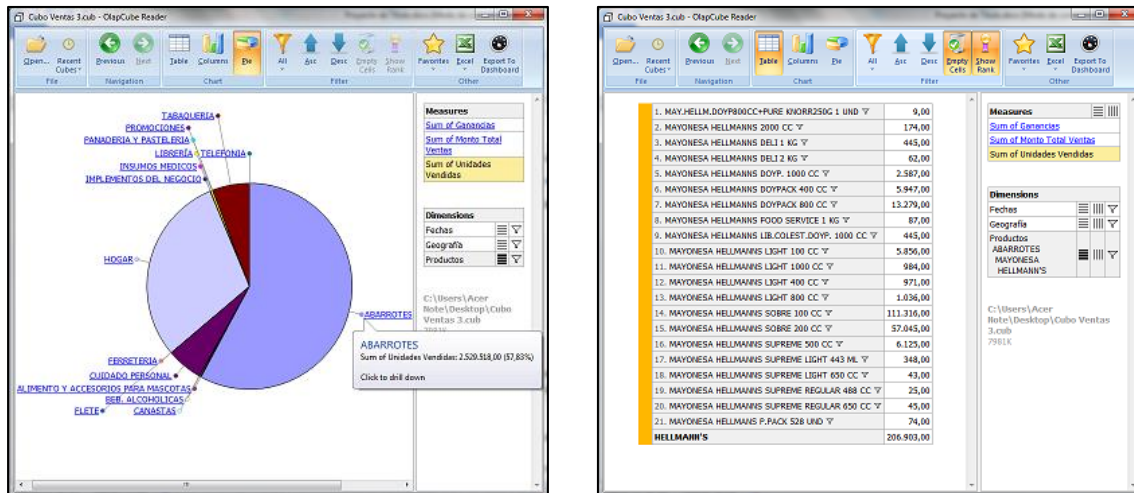


Figura 58: Pantalla de Análisis Dimensión PRODUCTOS

6.3.1.1.5 Paso 5: Análisis y Gráficos del cubo con Excel

Para trabajar con el cubo OLAP en Excel, se debe hacer clic en el icono Excel luego en “Open cube in new Workbook”, como se muestra en la Figura 59.

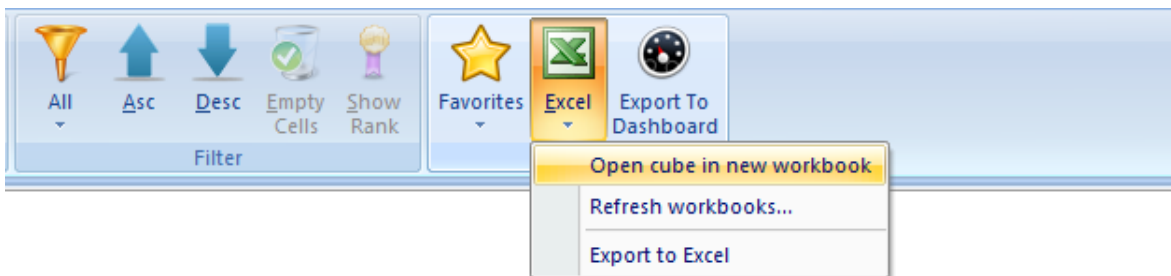
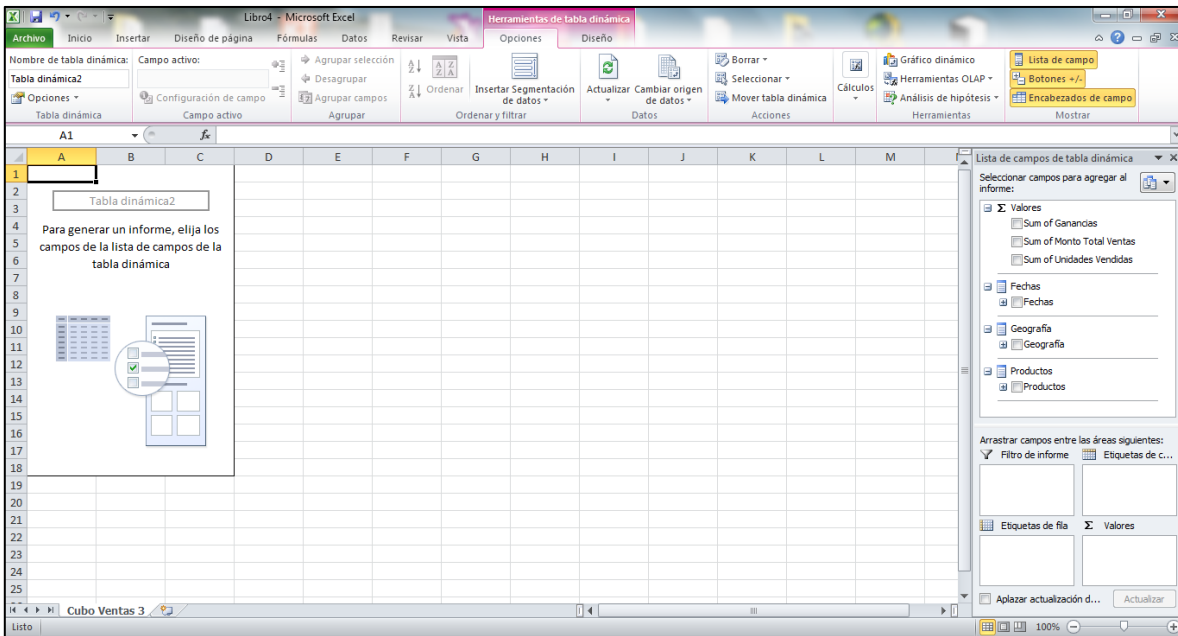


Figura 59: Pantalla de exportación del cubo a Excel

Luego se abrirá Excel con el cubo cargado, con un formato al estilo tabla dinámica, en la cual se podrá manipular los datos y generar reportes, entre otros.



A continuación se presentan tres ejemplos de gráficos generados que satisfacen los requerimientos de los clientes.

- Unidades vendidas de las marcas de margarinas por región, entre los meses agosto y octubre del año 2012 [Gráfico 1].

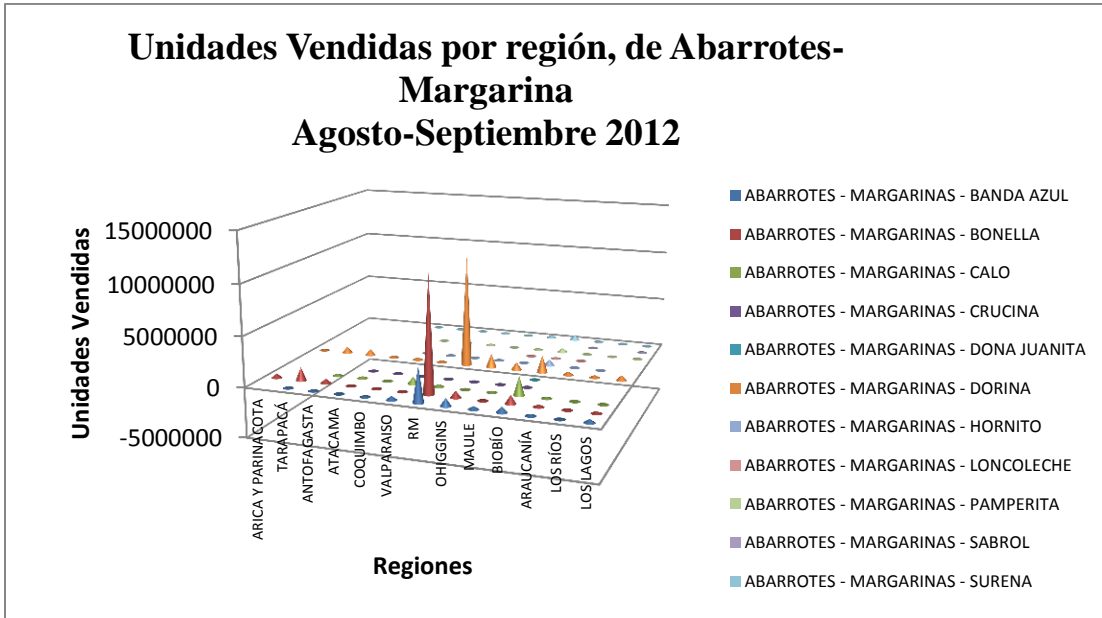


Gráfico 1: Unidades Vendidas Abarrotes-Margarinas Agosto-Octubre 2012

- Monto total de ventas en la categoría Tabaquería por región en el mes de septiembre 2012 [Gráfico 2].

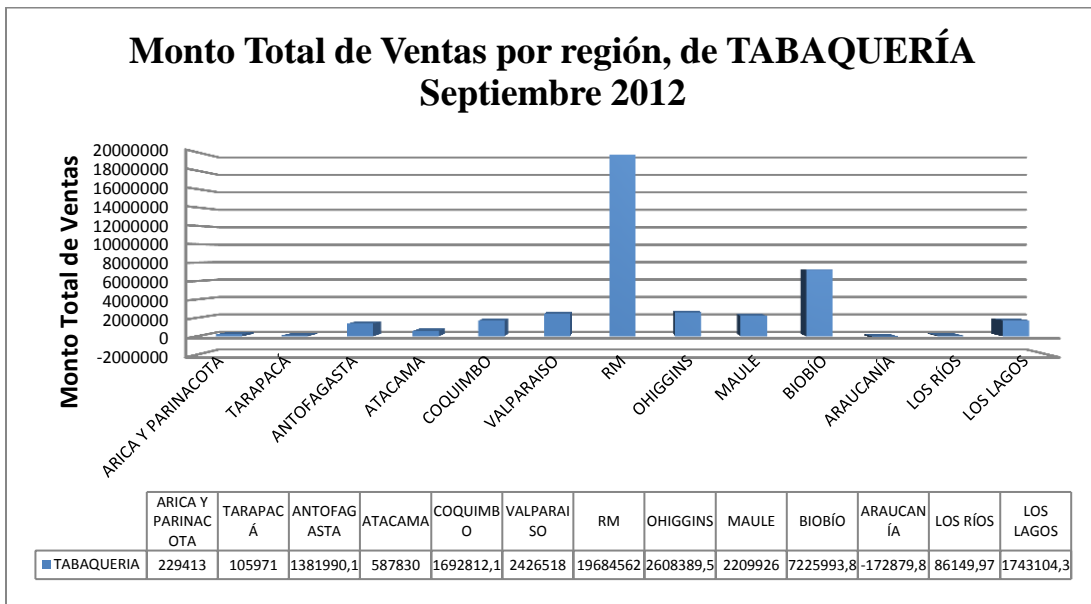


Gráfico 2: Monto Total de Ventas Tabaquería Septiembre 2012

- Ganancia de los gerentes generada en el primer día de los meses de agosto hasta octubre del año 2012, por el Canal 2 [Gráfico 3].

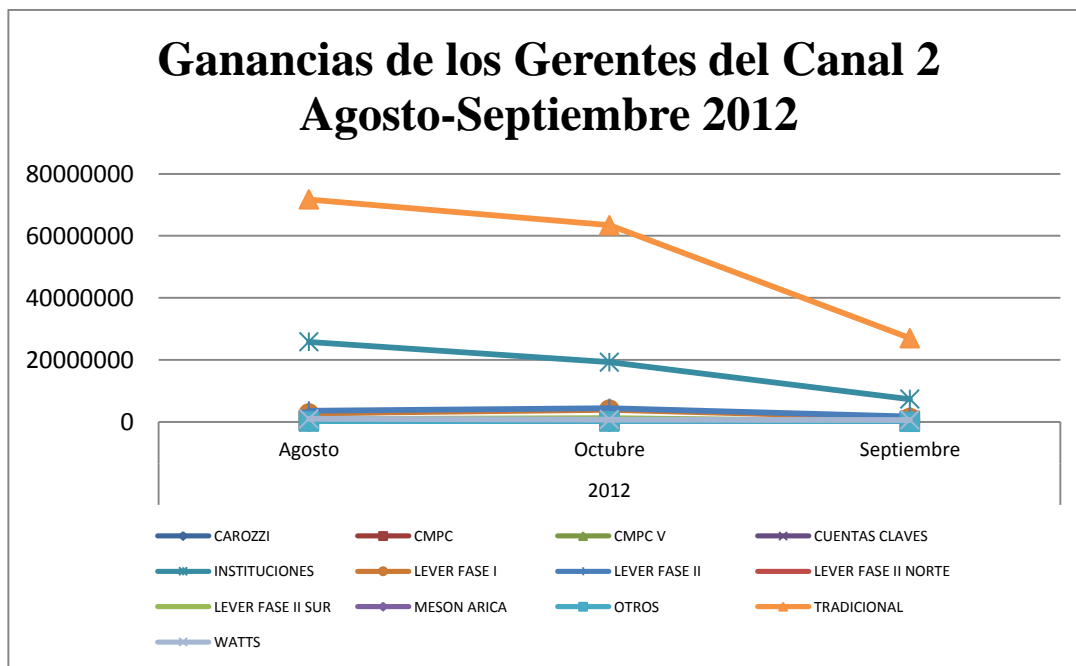


Gráfico 3: Ganancia de los Gerentes generada el primer día de cada mes por Canal 2

6.3.2 Mondrian

Mondrian también conocido como Pentaho Analysis Services Community Edition, es una herramienta OLAP que permite a los usuarios de negocio analizar grandes cantidades de datos en tiempo real. Además permite construir cubos de información para análisis multidimensional.

Dichos cubos se componen de archivos XML y en ellos se definen las Dimensiones y las conexiones de los datos. Los archivos XML por lo general son complejos de realizar manualmente por lo que es común utilizar herramientas gráficas para realizar la edición de estos.

Para trabajar con los cubos generados se utiliza la herramienta adicional que ofrece Open Source Pentaho [22], Workbench para editar cubos.

En la arquitectura de Mondrian se ejecuta sobre un servidor web que permite la comunicación entre aplicaciones OLAP con bases de datos. El núcleo del servidor

Mondrian es similar a JDBC pero exclusivo para OLAP. Proporciona la conexión a la base de datos y ejecuta las sentencias SQL.

Información adicional:

- Página Oficial del Producto: <http://www.pentaho.com>
- Idioma de la aplicación: inglés.
- Software gratuito

6.3.2.1 Mondrian Schema Workbench

El Mondrian Schema Workbenches es un diseñador de interfaz que permite crear y probar gráficamente esquemas de cubo OLAP Mondrian. Mondrian procesa las solicitudes de MDX con los ROLAP (OLAP Relacional). Estos archivos XML de metadatos son modelos que se crean en una estructura específica utilizada por Mondrian. Estos modelos XML pueden ser considerados como estructuras con forma de cubo de datos, que utilizan HECHOS existentes y tablas de dimensiones que se encuentran en la base de datos. No requiere construir un cubo real ya que solo construye el modelo relacional de los metadatos.

Proporciona las siguientes funcionalidades:

- Editor de esquemas integrados con origen de datos subyacente para su validación.
- Prueba consultas MDX y base de datos.
- Permite examinar la estructura de bases de datos subyacente.
- Contiene una interfaz gráfica bastante completa.

6.4 Data Mining

La minería de datos puede definirse inicialmente como un proceso de descubrimiento de nuevas y significativas relaciones, patrones y tendencias al examinar grandes cantidades de datos.

La disponibilidad de grandes volúmenes de información y el uso generalizado de herramientas informáticas ha transformado el análisis de datos orientándolo hacia

determinadas técnicas especializadas englobadas bajo el nombre de minería de datos o data mining.

Las técnicas de minería de datos pretenden lograr el descubrimiento automático del conocimiento en la información almacenada de modo ordenado en grandes bases de datos. Para lograr esto se espera descubrir patrones, perfiles y tendencias a través del análisis de los datos utilizando tecnologías de reconocimiento de patrones, redes neuronales, lógica difusa, algoritmos genéticos y otras técnicas avanzadas de análisis de datos que las empresas no conocen en su totalidad [23].

A diferencia de la definición de las herramientas de consulta y análisis, que básicamente se basan en sistemas relacionales y el resultado se presenta de forma tabular. O las herramientas OLAP que son más genéricas y funcionan sobre sistemas de información transaccional, permitiendo realizar agregaciones y combinaciones de datos de maneras mucho más complejas y ambiciosas, con el objetivo de análisis más estratégicos. Las herramientas de minería de datos permiten extraer patrones, tendencias y regularidades para describir y comprender mejor los datos, además de predecir comportamientos futuros. Por lo que en conclusión la minería de datos analiza los datos y el resto de las herramientas mencionadas anteriormente, facilitan el acceso a la información para que el análisis sea más efectivo, es decir, son instrumentos de apoyo a la minería de datos.

Finalmente se presentan cinco productos de software de minería de datos con código abierto.

6.4.1 Orange

Orange es una herramienta de minería de datos, que permite una visualización y análisis de datos, para principiantes y expertos. La minería de datos a través de la programación visual o de Python scripting, permite el aprendizaje automático. Además posee complementos para la bioinformática y la minería de texto, equipado con funciones de análisis de datos.

Principales características de la herramienta [24]:

- Programación Visual, que permite diseñar un proceso de análisis de datos. Orange posee un recordatorio de opciones, lo que permite sugerir combinaciones más utilizadas de forma inteligente.
- Visualización, desde diagramas de dispersión, gráficos de barras, árboles, a los dendrogramas, redes y mapas de calor.
- Interacción y Análisis de Datos, a través de esquemas de análisis de datos.
- Posee más de 100 widgets, cubriendo así la mayoría de las tareas estándar de análisis de datos. También ofrece especializados complementos, como Bio Orange para la bioinformática.
- Con interfaz de Python scripting, la programación y el desarrollo de nuevos algoritmos complejos o procedimientos de análisis de datos resulta de manera más fácil e inductiva, permitiendo la utilización y reutilización de código.
- Tiene la capacidad para extender la interfaz de scripting o incluso permite crear complementos autónomos.
- Proporciona información detallada de todos los widgets disponibles, permitiendo así guiar en todo el desarrollo de scripting.
- Una de sus principales características es que es Open Source o código abierto, libre para navegar y acceder al código fuente, extender y reutilizar los componentes.

Información adicional:

- Página Oficial del Producto: <http://orange.biolab.si/>
- Idioma de la aplicación: inglés.
- Fecha de última versión: Febrero, 2013
- Producto gratuito
- Plataforma: Windows, Mac OS X, y los sistemas operativos Linux.

6.4.2 RapidMiner

RapidMiner, antes llamado YALE (Yet Another Learning Environment), es un ambiente para la minería de datos y el aprendizaje automático utilizados en la investigación. Siendo sin duda el líder mundial de código abierto del sistema para la minería de datos. Está disponible como una aplicación independiente para el análisis de datos y como motor de minería de datos para la integración en los productos propios. Existen miles de aplicaciones de RapidMiner en más de 40 países que dan a sus usuarios una ventaja competitiva en el mercado laboral.

Algunas de las principales características de la herramienta son [25]:

- Posee un diseño que permite a los usuarios realizar acciones de forma más intuitiva.
- Varias capas de la vista de los datos, garantizan un manejo más eficiente de los datos.
- El modo servidor, o el acceso a través de la API de Java, tienen una interfaz gráfica de usuario.
- Dispone de más de 500 operadores para la integración y transformación de los datos, minería de datos, evaluación y visualización de los datos.
- Genera esquemas de metadatos automáticos de forma optimizada.
- Definición de los bloques de construcción reutilizables.
- Utiliza XML con formato estandarizado para el intercambio de los procesos.
- Diseño de proceso gráfico para tareas estándar, lenguaje de scripts para operaciones arbitrarias.
- Machine Learning Library WEKA totalmente integrado.
- Tiene acceso a fuentes de datos, como Excel, Access, Oracle, IBM DB2, Microsoft SQL, Sybase, Ingres, MySQL, Postgres, SPSS, dBase, archivos de texto y más.
- Entrega una completa solución de minería con respecto a la integración, transformación y métodos de modelado de los datos.

Información adicional:

- Página Oficial del Producto: <http://rapidminer.com/>
- Idioma de la aplicación: inglés.
- Producto gratuito

- Plataforma: Windows, Mac OS X, y los sistemas operativos Linux.

6.4.3 Weka

Escrito en Java, Weka (Waikato Environment for Knowledge Analysis) es una conocida suite de software para el aprendizaje, que soporta varias tareas típicas de minería de datos, especialmente los datos del proceso previo, el agrupamiento, clasificación, regresión, visualización y selección de características. Sus técnicas se basan en la hipótesis de que los datos están disponibles en un único archivo plano o una relación, donde se etiqueta cada punto de datos por un número fijo de atributos. WEKA proporciona acceso a bases de datos SQL utilizando Java Database Connectivity y puede procesar el resultado devuelto por una consulta de base de datos. Su interfaz de usuario principal es el Explorer, pero es la misma funcionalidad que se entrega desde la línea de comandos o a través de la interfaz basada en componentes de flujo de conocimientos.

Además, se presentan algunas características de interés [26]:

- Utiliza formato de datos XRFF como .xrff o .xrff.gz
- Permite la conexión de IDEs como Eclipse y NetBeans.
- Contiene instrucciones paso a paso sobre cómo configurar WEKA como un proyecto.
- Permite clasificar nuevos, filtros, etc., disponibles en WEKA (utilizando el GenericObjectEditor).
- Accede a bases de datos de Windows como MS Access y otras bases de datos en general (MySQL, PostgreSQL, MS SQL Server, Oracle, etc) mediante JDBC.
- Contiene documentación que describe el uso de XML en WEKA.
- Documentación sobre el apoyo Weka para la importación de PMML modelos.

Información adicional:

- Página Oficial del Producto: <http://www.cs.waikato.ac.nz/~ml/weka/>
- Idioma de la aplicación: inglés.
- Producto gratuito
- Plataforma: Windows, Mac OS X, y los sistemas operativos Linux.

6.4.4 jHepWork

Diseñado para los científicos, ingenieros y estudiantes, jHepWork de código abierto para el análisis de estructuras de datos, se crea como un intento de hacer un análisis del entorno de los datos, usando paquetes de código abierto con una interfaz de usuario comprensible, de esta forma creando una herramienta competitiva para programas comerciales. Esto se hace especialmente para las comunidades científicas interactivas en formatos 2D y 3D y contiene numerosas bibliotecas científicas implementadas en Java, para funciones matemáticas, números al azar, y otros algoritmos de minería de datos.

Específicamente, algunas de sus características son [27]:

- Se basa en un lenguaje de programación de alto nivel Jython, aunque Java también puede ser usado para llamar a bibliotecas jHepWork.
- Permite el análisis de grandes volúmenes de datos numéricos, minería de datos, análisis estadístico.
- Saca un máximo provecho de los procesadores multinúcleo.
- Se puede utilizar con varios lenguajes de scripting para la plataforma Java, como Jython (el lenguaje de programación Python) y BeanShell. Aportando más potencia y sencillez para la computación científica. La programación también se puede hacer en lenguaje Java.
- Finalmente, los cálculos simbólicos se puede hacer usando Matlab/Octave lenguaje interpretado de alto nivel.

Información adicional:

- Página Oficial del Producto: <http://jwork.org/jhepwork/>
- Idioma de la aplicación: inglés.
- Producto gratuito
- Plataforma: Windows, Mac OS X, sistemas operativos Linux y además de sistemas operativos Android.

6.4.5 KNIME

KNIME (Konstanz Information Miner) es de uso fácil y comprensible, con fuente abierta de integración de datos, procesamiento y análisis de estos. Se ofrece a los usuarios la capacidad de crear de forma visual los flujos de datos, ejecutar selectivamente algunos o todos los pasos de análisis, y luego estudiar los resultados, modelos y vistas interactivas.

KNIME está escrito en Java y está basado en Eclipse, haciendo uso de su método de extensión para apoyar plugins, proporcionando así una funcionalidad adicional. A través de plugins, los usuarios pueden añadir módulos de texto, imagen, y el procesamiento de series de tiempo, además de la integración de varios otros proyectos de código abierto, tales como el lenguaje de programación de R, WEKA y LIBSVM.

Entre sus características funcionales más importantes podemos encontrar las siguientes [28]:

- Proporciona más de 1000 rutinas analíticas para los datos, ya sea nativa o a través de R y WEKA, por temas tales como:
 - Las estadísticas uni y multivariado
 - Data Mining
 - Series de tiempo
 - Procesamiento de imágenes
 - Web Analytics
 - Text Mining
 - Análisis de Redes

- Análisis de Medios Sociales
- En su funcionalidad de diseñador de informes, permite utilizar los datos y los resultados como insumo para generar informes integrales, que pueden ser guardados como plantillas y de esta forma crear informes en formatos comunes como PDF, XLS, DOC, PPT y otros.

Información adicional:

- Página Oficial del Producto: <http://www.knime.org/>
- Idioma de la aplicación: inglés.
- Producto gratuito
- Plataforma: Windows, Mac OS X, sistemas operativos Linux.

Capítulo 7

Conclusiones

7.1 Introducción

Luego de realizar una investigación y análisis acerca de los sistemas Data Warehouse, se pueden sacar bastantes conclusiones, debido a lo amplio que es el tema, algunas de las cuales serán personales, reflejando así la experiencia vivida en el uso de estas tecnologías. En el presente informe de proyecto de título, se trató de abarcar los contenidos más relevantes a la hora de implementar un sistema Data Warehouse en el área de una empresa que contiene un gran volumen de datos como lo es Rabie S.A.

Todo lo anterior obliga a replantearse las estrategias actuales, provocando de esta forma, una transformación en el negocio. Siendo un factor clave de éxito, la capacidad de la organización para gestionar de forma eficiente sus datos y transformarlos en información útil y disponible para acertar en las decisiones.

7.2 Conclusiones por objetivos planteados

El desarrollo de este proyecto se centró en, proponer el diseño e implementación de un Data Warehouse, como también en demostrar a través de resultados el funcionamiento de éste, permitiendo apoyar y potenciar la entrega de información del área de Soporte de Información de la empresa Rabie S.A. a las áreas dependientes de ésta de manera eficiente y oportuna. Y si bien el diseño estuvo orientado a las necesidades actuales de la empresa, es completamente modificable y adaptable a nuevos requerimientos, ya que su diseño permite gran flexibilidad y escalabilidad.

De acuerdo a los objetivos planteados en este proyecto, definidos en el Capítulo 1, se obtuvieron las siguientes conclusiones:

- **“Definir el marco teórico de un Data Warehouse, con objeto de reconocer el valor de esta tecnología y las posibilidades que ofrece a las organizaciones”.**

La investigación realizada para reconocer los conceptos que se encuentran relacionados a un Data Warehouse, permitió dimensionar los diversos beneficios que trae consigo la implementación de un sistema de este tipo y los alcances que pueda tener en la organización (ver Capítulo 3).

- **“Comparar las metodologías más utilizadas en la implementación de un Data Warehouse, con el propósito de que la organización pueda identificar la que más se ajusta a sus necesidades”.**

Se definieron las distintas metodologías utilizadas para dicha implementación (ver Capítulo 3), identificando de esta forma la metodología a utilizar en el desarrollo de este proyecto (ver Capítulo 4), dicha metodología fue Hefesto, la cual se ajusta más a las condiciones actuales del área que evaluará la implementación.

- **“Identificar los datos necesarios para conformar el Data Warehouse con sus respectivas fuentes (componente OLTP, On Line Transaction Processing)”.**

La identificación de los datos se realizó a través de la generación de los requerimientos, lo que permitió la interiorización en el área, detectando sus reales necesidades (ver Capítulo 5).

- **“Establecer los procesos aplicados a los datos para manipularlos, integrarlos y transformarlos, de modo que, posteriormente, sea posible cargar los resultados obtenidos en el Data Warehouse (componente Load Manager)”.**

La definición de los procesos ETL necesarios para la implementación del Data Warehouse, en el área de Soporte de Información, permitió dimensionar la dificultad de extraer los datos de tablas que contienen grandes cantidades de atributos además de un gran volumen de datos. Sin mencionar que se tuvo que

generar claves primarias para cada una de las dimensiones debido a que no existían atributos claves para las identidades extraídas.

- **“Proponer el modelo de la base de datos, al área de Soporte de Información, basándose en los tipos de modelado de un Data Warehouse (componente Data Warehouse Manager)”.**

Se diseñó un modelo que permite dar respuesta al 85% de las consultas generadas en el área de ventas. Para ello se generaron tres cubos los que permiten demostrar el funcionamiento del sistema Data Warehouse a través de la aplicación OlapCube, presentando de esta forma resultados con datos reales a través de gráficos y planillas.

Además, a través del Data Warehouse y los cubos OLAP se demostró que se puede obtener información que no existe en algún informe actual, logrando valiosas funcionalidades que no estaba previstas en un comienzo.

A través de reuniones con el personal del área de Soporte de Información, fue posible dar cuenta de las limitaciones de los sistemas transaccionales y de los beneficios que trae consigo la implementación de un Data Warehouse en dicha área.

8. Bibliografía

- [1] Bernabeu Ricardo Dario, "Data Warehousing: Investigación y Sistematización de Conceptos - HEFESTO: Metodología propia para la Construcción de un Data Warehouse," Córdoba, Argentina, Tesis doctoral 2009.
- [2] Rabie S.A. (2013, Enero) Distribuidora Rabie. [Online]. <http://www.rabie.cl/>
- [3] Raúl Horacio Saroka, *Sistemas de Información en la Era Digital*, Primera ed. San Martín, Argentina: Fundación OSDE, 2002.
- [4] Josep Lluís Cano, *Business Intelligence: Competir con Información*, Primera ed., Escuela Banespyme, Ed. Barcelona, España: Fundación Cultur, 2007.
- [5] Elizabeth Santana Beoto, Joan Manuel , and Granadillo Rodriguez, "Primeros pasos hacia la Inteligencia de Negocios: Almacenes de datos (DataWareHouse), Herramientas ETL(PDI), Procesamiento analítico en línea.," *MsC. Julio Cesar Camps*, vol. 10, no. 2, 2011.
- [6] Date C. J., *Introducción a los Sistemas de Bases de Datos*, Séptima ed.: Pearson, 2001.
- [7] A. Mendez, A. Mártire, P. Britos, and Martí Garcia, "Fundamentos de Data Warehouse," *Reportes Técnicos en Ingeniería del Software*, vol. 5, no. 1, pp. 19-26, 2003.
- [8] Juan Vicente Pérez Palanca, "Proceso de desarrollo de indicadores de gestión. Un caso de estudio en el contexto de un ERP.," Universidad Politécnica de València, València, Venezuela, Tesis de Magister 2010.
- [9] Griselda E. Bressán. (2012, Noviembre) Universidad Nacional del Nordeste, Facultad de las Ciencias Exactas y Naturales y Agrimensura. [Online]. <http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MineriaDatosBressan.htm>
- [10] Bill Inmon and Richard Hackathorn, *Building The Data Warehouse*, Third Edition ed.: Jhon Wiley and sons, 1992.
- [11] María de los Ángeles Gil Estallo and Fernando Giner de la Fuente, *Los Sistemas de Información en la Sociedad Del Conocimiento*, Primera Edición ed. España: ESIC, 2004.
- [12] William Inmon, *Building the Data Warehouse*, 4th ed.: John Wiley & Sons Inc, 2005.
- [13] Itziar Angoitia Espinosa, "Data warehouse para la gestión de lista de espera sanitaria," Universidad Politécnica de Madrid, Facultad de Informática, Madrid, Tesis 2008.

- [14] Instituto Nacional de Estadística e Informática INEI, *Manual para la Construcción de un Data Warehouse*, Profesor J. Elliott, Ed. Lima, Perú, 1997.
- [15] Ralph Kimball and Margy Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, Second Edition ed., Robert Elliott, Ed. Canadá: John Wiley & Sons Inc, 2002.
- [16] Arun Sen and Atish P. Sinha, "A Comparison of Data Warehousing Methodologies," *Communications of the ACM*, vol. 48, no. 3, pp. 79-84, 2005.
- [17] Carlos Fernández. Dataprix. [Online]. <http://www.dataprix.com/manual-para-la-adquisici-n-un-sistema-data-warehouse>
- [18] R. Kimball, M. Ross, W. Thornthwaite, and J. Mundy, *The data warehouse lifecycle toolkit*, 2nd ed.: Wiley Publishing, Inc., 2008.
- [19] Joel Mija Ferrer, Luis Otiniano Dávila, and César Velásquez Haro, "Estado del Arte de Data Warehouse".
- [20] SAP. (2013, Febrero) SAP The Best-Run Businesses Run SAP. [Online]. <http://www.sap.com>
- [21] Josep Curto Díaz, *Introducción al Business Intelligence*, Primera Edición en Lengua Castellana ed. Barcelona, España: UOC, 2010.
- [22] Pentaho. Pentaho. [Online]. <http://www.pentaho.com/>
- [23] César Pérez López and Daniel Santín Gonzalez, *Minería de datos: técnicas y herramientas*, Primera ed., Paraninfo S.A., Ed. Madrid, España: THOMSON Ediciones, 2007.
- [24] Orange. (2013) Orange. [Online]. <http://orange.biolab.si/>
- [25] Rapid-i. (2013) Rapid-i. [Online]. <http://rapidminer.com/>
- [26] The University of WAIKATO. (2013) The University of WAIKATO. [Online]. <http://www.cs.waikato.ac.nz/~ml/weka/>
- [27] jHepWord. (2013) jHepWord. [Online]. <http://jwork.org/jhepwork/>
- [28] Knime. (2013) Knime. [Online]. <http://www.knime.org/>

