Universidad de Cienfuegos "Carlos Rafael Rodríguez"
Facultad de Informática
Carrera de Ingeniería Informática

"Diseño de un Datamart y de los procesos ETL, para la toma de decisiones en las áreas de portadores energéticos y Transporte del MES"

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor:
Abdiel Alejandro Caballero Legrá

Tutores:

Ing. Milena Ramos Pérez

Msc. Antonio Toledo Dorrego

Consultante:

Ing. Miguel Pelayo García Fernández

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo (firmamos) la presente a los __días del mes de Junio del 2008.

_____<Firma de el (los) autor (es)>_____

Nombre(s) y apellidos de el (los) autor(es)

_____<Firma de el (los) tutor (es)>_____

Nombre(s) y apellidos de el (los) tutor(es)

AGRADECIMIENTOS

Doy infinitas gracias a mi Dios por ayudarme en todo y ser mi amigo fiel. Nunca me ha desamparado y ha estado siempre a mi lado. Dándome fuerzas, aliento, paz en la tormenta. Te amo mucho.

A mi esposa que siempre me ha apoyado con tanto amor y dedicación. Pasando noches de desvelo; que con una paciencia infinita ha esperado hasta este momento para ya nunca más si Dios quiere estar separados. Quiera Dios que pueda compensarte este tiempo de estar lejos. Nunca podría expresar lo mucho que te amo.

A papi por su carácter que me ayudó a no cejar nunca, ha ser un hombre decidido, un hombre de principios. Por enseñarme el amor a Dios. Te quiero mucho “pipucho”.

A mami por su ternura, por su dedicación en preparar mis cosas todos estos años de estudios, por sus consejo, por aguantarme muchas malacrianzas, por perdonarme tratos ásperos y duros. Por enseñarme el amor de Dios ¡Qué dicha tenerte madre! Te quiero mucho mima.

A los dos porque han sabido dirigir el hogar.

A mi hermana Dámaris y Yosbel por estar a mi lado y poder sentir su amor. Yo se que han orado por mí.

A mis suegros y mi cuñado por su amor para mí incondicional, por luchar conmigo como si fuera otro hijo, creo que gran parte de todo esto tan feliz se lo debo a ellos. Por darme a mi esposa. Los quiero mucho.

A Eliseo y Baby que no han tenido reserva conmigo para nada, como un dicho que Baby dice y yo lo aplico a mí: “lo único que no han hecho es parirme”. Gracias hermanos por su amor, me he sentido muy bendecido entre ustedes. Icha gracias también por todo tu amor y tu cariño de hermano. Oraré mucho por ustedes.

A ambas familias: La de mi esposa, que me ha recibido como un miembro más. A mi familia que también me han apoyado mucho. A todos los amo. No puedo nombrarlos a todos porque son muchos, gracias por todo lo que han hecho por mí.

A los hermanos de la Iglesia que han orado por mí.

A mis tutores por ayudarme en todo lo necesario.

A mi tutora que ha estado para todo lo que me ha hecho falta, gracias por haber batallado también conmigo todo este tiempo. Te estoy muy agradecido de verdad Milena.

DEDICATORIA

A mi Dios, todo se lo debo a Él.
A mi bella esposa, por tanto y tanto amor, y a nuestro hijo.

A toda mi familia por el apoyo que me han dado.
A la familia de mi esposa por todo su amor y apoyo.

RESUMEN

La correcta gestión en cualquier área de una empresa es uno de los pilares básicos del éxito de toda empresa. Las decisiones que se adopten tienen un gran peso en la consecución de la mejora competitiva de la empresa. Para conseguir adoptar la mejor decisión van a utilizar toda la información que le ayude en la tarea de tomarla. Para ello, el sistema de información de la empresa debe ser capaz de suministrar la información adecuada en el momento preciso a los decisores.

Es también de mucha importancia para el Ministerio de Educación Superior de Cuba (MES), un sistema para la ayuda de toma de decisiones, en áreas tan importantes como los portadores energéticos: combustible y energía eléctrica, teniendo en cuenta de ellos tanto la demanda que hace cada Centro de Educación Superior (CES), como la posterior asignación del MES a cada uno de los CES; abarca también como otra área de importancia, toda la información referente al transporte en cada CES; incluyendo indicadores y otros portadores.

El propósito de este trabajo es establecer una organización y modelo, que facilite el control de lo referente a los portadores energéticos y el transporte, facilitando así la manipulación y procesamiento de la información antes mencionadas que llegan de todos los Centros de Educación Superior (CES). Para ello se ha utilizado tecnologías basadas en Inteligencia de Negocio. Este trabajo permitirá el procesamiento analítico de información histórica, actual, agregada y nominalizada en correspondencia con las necesidades de información.

La propuesta de solución abarca los siguientes aspectos:

- El diseño y la implementación de un Datamart, con el empleo de la metodología de diseño basada en UML y en el Proceso Unificado: *Data Warehouse Engineering Process* (DWEP).
- El diseño de los procesos de extracción, transformación y carga necesarios para poblar y proporcionar consistencia a los datos almacenados en el Datamart.

ÍNDICE

INTRODUCCIÓN 1

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS 9

Introducción 9

1.1. La Inteligencia de Negocio como estrategia para la toma de decisiones 9

1.1.1. Data Warehouse 13

1.1.2. Datamarts 16

1.1.3. Procesos de Extracción, Transformación y Carga de Datos (ETL) 16

1.1.4. Modelo Multidimensional 18

1.1.5. Metodología para el Desarrollo de Data Warehouse 19

1.1.5.1. Modelo Conceptual 23

1.1.5.2. Modelo Lógico. 24

1.1.5.3. Modelo Físico 26

1.1.6. Procesamiento Transaccional en Línea y Procesamiento Analítico en Línea (OLTP y OLAP) 26

1.1.7. Camino hacia el Software Libre 29

1.1.7.1. Sistema Gestores de Bases de Datos Libres 30

1.1.7.2. Herramientas de Inteligencia de Negocio “Open Source” 32

1.1.7.3. Pentaho, una solución BI de “Open Source” 39

Conclusiones 41

CAPÍTULO 2. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL 43

Introducción 43

2.1. Modelo formalizado para el análisis de los portadores energéticos y transporte 43

2.2. Sistemas Existentes 44

El Transorg 44

El MACWIN 45

2.3. Tendencias y técnicas actuales 47

Conclusiones 48
CAPÍTULO 3. MODELO DE SOLUCIÓN PROPUESTO 49
Introducción 49
3.1. Sistema Propuesto 49
3.2 Diseño del Datamart 50
3.2.1 Modelo Conceptual 50
3.2.1.1 Nivel 1 (Definición Global del Modelo) 52
3.2.1.2 Nivel 2 (Definición de Estructura de Esquemas Estrella) 55
3.2.1.3 Nivel 3 (Definición de Estructura de Hechos y Dimensiones) 64
3.2.1.4 Mapeo de Datos 79
3.2.2 Modelo Lógico 80
3.2.2.1 Diseño de los Procesos ETL 90
3.2.3 Modelo Físico 108
3.2.3.1 Diagrama de Componentes 109
3.2.3.2 Diagrama de Despliegue 111
Conclusiones 114
CONCLUSIONES 115
RECOMENDACIONES 117
REFERENCIAS BIBLIOGRÁFICAS 118
GLOSARIO DE TÉRMINOS 120

Índice de Figuras

Figura 1. 1 Etapas por las que transitan los datos en un proceso de toma de decisiones. 11

Figura 1. 2 Sistema de Apoyo a la Toma de Decisiones. 12

Figura 1. 3 Ciclo de vida de DWEP. 20

Figura 1. 4 Arquitectura de la Suite Pentaho. 40

Figura 3. 1 Diagrama de Esquemas Estrellas correspondiente a la Demanda. 53

Figura 3. 2 Diagrama de Esquemas Estrellas correspondiente a Información General del Transporte. 54

Figura 3. 3 Diagrama de Esquemas Estrellas correspondientes al Plan operativo. 54

Figura 3. 4 Diagrama de Esquemas Estrellas correspondiente al Consumo. 55

Figura 3. 5 Estructura del Esquema Estrella: StarSchema_ Dem_Combustible 56

Figura 3. 6 Estructura del Esquema Estrella: StarSchema_ Dem_Electricidad 57

Figura 3. 7 Estructura del Esquema Estrella: StarSchema_ Dem_Proyecto 57

Figura 3. 8 Estructura del Esquema Estrella: StarSchema_ Dem_Portadores 58

Figura 3. 9 Estructura del Esquema Estrella: StarSchema_ Cant_Vehículo 58

Figura 3. 10 Estructura del Esquema Estrella: StarSchema_ Cant_Combustible 59

Figura 3. 11 Estructura del Esquema Estrella: StarSchema_ Cant_Actividad 59

Figura 3. 12 Estructura del Esquema Estrella: StarSchema_ Cant_Consumidor 60

Figura 3. 13 Estructura del Esquema Estrella: StarSchema_ Cant_Chapa 60

Figura 3. 14 Estructura del Esquema Estrella: StarSchema_Indicadores 61

Figura 3. 15 Estructura del Esquema Estrella: StarSchema_Plan_Combustible 61

Figura 3. 16 Estructura del Esquema Estrella: StarSchema_ Plan_Electricidad 62

Figura 3. 17 Estructura del Esquema Estrella: StarSchema_ Plan_Portadores 62

Figura 3. 18 Estructura del Esquema Estrella: StarSchema_ Cons_Combustible 63

Figura 3. 19 Estructura del Esquema Estrella: StarSchema_ Cons_Electricidad 63

Figura 3. 20 Estructura del Esquema Estrella: StarSchema_ Cons_Portadores 64

Figura 3. 21 Estructura de la Dimensión: Dim_Combustible. 64
Figura 3. 22 Estructura de la Dimensión: Dim_TipoColumnaC. 65
Figura 3. 23 Estructura de la Dimensión: Dim_ActividadG. 65
Figura 3. 24 Estructura de la Dimensión: Dim_Actividad. 65
Figura 3. 25 Estructura de la Dimensión: Dim_Centro. 66
Figura 3. 26 Estructura de la Dimensión: Dim_Servicio. 66
Figura 3. 27 Estructura de la Dimensión: Dim_TipoColumnaE. 67
Figura 3. 28 Estructura de la Dimensión: Dim_Proyecto. 67
Figura 3. 29 Estructura de la Dimensión: Dim_TipoIndicadores. 67
Figura 3. 30 Estructura de la Dimensión: Dim_TipoVehículo. 68
Figura 3. 31 Estructura de la Dimensión: Dim_TipoConsumidor. 68
Figura 3. 32 Estructura de la Dimensión: Dim_OtrosPortadores. 68
Figura 3. 33 Estructura de la Dimensión: Dim_TipoChapa. 69
Figura 3. 34 Estructura de la Dimensión: Dim_Moneda. 69
Figura 3. 35 Estructura de la Dimensión: Dim_DateCentro. 69
Figura 3. 36 Estructura de la Dimensión: Dim_Date. 70
Figura 3. 37 Estructura del Hecho: Fact_DemandaCombustible. 71
Figura 3. 38 Estructura del Hecho: Fact_DemandaElectricidad. 71
Figura 3. 39 Estructura del Hecho: Fact_DemandaProyecto. 72
Figura 3. 40 Estructura del Hecho: Fact _DemandaOtrosPortadores. 72
Figura 3. 41 Estructura del Hecho: Fact _CantidadTipoVehículo. 73
Figura 3. 42 Estructura del Hecho: Fact _CantidadCombustible. 73
Figura 3. 43 Estructura del Hecho: Fact _CantidadActividad. 74
Figura 3. 44 Estructura del Hecho: Fact _CantidadTipoConsumidor. 74
Figura 3. 45 Estructura del Hecho: Fact _CantidadTipoChapa. 75
Figura 3. 46 Estructura del Hecho: Fact _Indicadores. 75

Figura 3. 47 Estructura del Hecho: Fact _PlanOperativoCombustible. 76
Figura 3. 48 Estructura del Hecho: Fact _PlanOperativoElectricidad. 76
Figura 3. 49 Estructura del Hecho: Fact_PlanOperativoOtrosPortadores. 77
Figura 3. 50 Estructura del Hecho: Fact _ConsumoCombustible. 77
Figura 3. 51 Estructura del Hecho: Fact _ConsumoElectricidad. 78
Figura 3. 52 Estructura del Hecho: Fact _ConsumoOtrosPortadores..... 78
Figura 3. 53 Diagrama de Mapeo de los Datos. 79
Figura 3. 54 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Combustible..... 80
Figura 3. 55 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Electricidad. 81
Figura 3. 56 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Proyecto..... 81
Figura 3. 57 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Portadores. 82
Figura 3. 58 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Vehículo..... 82
Figura 3. 59 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Combustible..... 83
Figura 3. 60 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Actividad. 83
Figura 3. 61 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Consumidor. 84
Figura 3. 62 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Chapa. 84
Figura 3. 63 Modelo Lógico del Esquema Estrella: StarSchema_Indicadores. 85
Figura 3. 64 Modelo Lógico del Esquema Estrella: StarSchema_Plan_Combustible. 85
Figura 3. 65 Modelo Lógico del Esquema Estrella: StarSchema_ Plan_Electricidad 86
Figura 3. 66 Modelo Lógico del Esquema Estrella: StarSchema_ Plan_Portadores. 86
Figura 3. 67 Modelo Lógico del Esquema Estrella: StarSchema_ Cons_Combustible..... 87
Figura 3. 68 Modelo Lógico del Esquema Estrella: StarSchema_ Cons_Electricidad. 87
Figura 3. 69 Modelo Lógico del Esquema Estrella: StarSchema_ Cons_Portadores. 88
Figura 3. 70 Modelo Lógico de la Dimensión: Dim_Centro. 88
Figura 3. 71 Modelo Lógico de la Dimensión: Dim_Servicio. 89
Figura 3. 72 Modelo Lógico de la Dimensión: Dim_DateCentro. 89

Figura 3. 73 Modelo Lógico de la Dimensión: Dim_Date. 90
Figura 3. 74 Prototipo de la Estructura de las Tablas de Dimensiones. 91
Figura 3. 75 Proceso ETL correspondiente a la dimensión: Dim_Combustible 92
Figura 3. 76 Proceso ETL correspondiente a la dimensión: Dim_TipoColumnaC 92
Figura 3. 77 Proceso ETL correspondiente a la dimensión: Dim_ActividadG 93
Figura 3. 78 Proceso ETL correspondiente a la dimensión: Dim_Actividad 93
Figura 3. 79 Proceso ETL correspondiente a la dimensión: Dim_Centro 94
Figura 3. 80 Proceso ETL correspondiente a la dimensión: Dim_TipoCentro 94
Figura 3. 81 Proceso ETL correspondiente a la dimensión: Dim_Provincia 95
Figura 3. 82 Proceso ETL correspondiente a la dimensión: Dim_Servicio 95
Figura 3. 83 Proceso ETL correspondiente a la dimensión: Dim_Municipio 96
Figura 3. 84 Proceso ETL correspondiente a la dimensión: Dim_Empresa 96
Figura 3. 85 Proceso ETL correspondiente a la dimensión: Dim_TipoColumnaE 97
Figura 3. 86 Proceso ETL correspondiente a la dimensión: Dim_Proyecto 97
Figura 3. 87 Proceso ETL correspondiente a la dimensión: Dim_TipoIndicadores 98
Figura 3. 88 Proceso ETL correspondiente a la dimensión: Dim_TipoVehículo 98
Figura 3. 89 Proceso ETL correspondiente a la dimensión: Dim_TipoConsumidor 99
Figura 3. 90 Proceso ETL correspondiente a la dimensión: Dim_OtrosPortadores 99
Figura 3. 91 Proceso ETL correspondiente a la dimensión: Dim_TipoChapa 100
Figura 3. 92 Proceso ETL correspondiente a la dimensión: Dim_DateCentro 100
Figura 3. 93 Proceso ETL correspondiente al hecho: Fact_DemandaCombustible 101
Figura 3. 94 Proceso ETL correspondiente al hecho: Fact_DemandaElectricidad 102
Figura 3. 95 Proceso ETL correspondiente al hecho: Fact_DemandaProyecto 102
Figura 3. 96 Proceso ETL correspondiente al hecho: Fact _DemandaOtrosPortadores . 103
Figura 3. 97 Proceso ETL correspondiente al hecho: Fact _CantidadTipoVehículo 103
Figura 3. 98 Proceso ETL correspondiente al hecho: Fact _CantidadCombustible 104

Figura 3. 99 Proceso ETL correspondiente al hecho: Fact _CantidadActividad 104
Figura 3. 100 Proceso ETL correspondiente al hecho: Fact _CantidadTipoConsumidor 105
Figura 3. 101 Proceso ETL correspondiente al hecho: Fact _CantidadTipoChapa 105
Figura 3. 102 Proceso ETL correspondiente al hecho: Fact _Indicadores 106
Figura 3. 103 Proceso ETL correspondiente al hecho: Fact _ConsumoCombustible 106
Figura 3. 104 Proceso ETL correspondiente al hecho: Fact _ConsumoElectricidad 107
Figura 3. 105 Proceso ETL correspondiente al hecho: Fact _ConsumoOtrosPortadores 107
Figura 3. 106 Diagrama de Componentes (Dimensiones). 110
Figura 3. 107 Diagrama de Componentes (Hechos). 111
Figura 3. 108 Diagrama de Despliegue del Sistema. 112

Índice de Tablas

Tabla 1. 1 Diferencias entre las bases de datos transaccionales y los almacenes de datos. 15
Tabla 1. 2 Estereotipos empleados en el modelo conceptual del DW. 24
Tabla 1. 3 Estereotipos empleados en el modelo de los procesos ETL. 26
Tabla 1. 4 Comparación entre MySQL 5.x y PostgreSQL 8.0.x 31
Tabla 1. 5 Análisis comparativo entre herramientas ETL. 34
Tabla 1. 6 Comparación entre Pentaho y SpagoBI. 38

Tabla 3. 1 Esquemas Estrella correspondiente a la Demanda. 53
Tabla 3. 2 Esquemas Estrellas correspondiente a Información General del Transporte ... 53
Tabla 3. 3 Esquemas Estrellas correspondiente al Plan operativo. 54
Tabla 3. 4 Esquemas Estrellas correspondiente al Consumo. 54
Tabla 3. 5 Dimensiones propuestas en la solución. 55
Tabla 3. 6 Hechos propuestos en la solución. 56

INTRODUCCIÓN

Durante décadas, las tecnologías de la información han dedicado sus esfuerzos a la automatización de los procesos de carácter típicamente repetitivo o administrativo, haciendo uso de los sistemas de información operacionales o transaccionales en línea, ***OLTP*** (On-Line Transaction Processing). Los sistemas OLTP están orientados a la captura rápida y segura de grandes volúmenes de información detallada. En ellos, los conceptos más importantes son la actualización y el tiempo de respuesta y se ha invertido mucho esfuerzo en la construcción de sistemas capaces de registrar adecuadamente movimientos de almacenes o transacciones contables. Estos sistemas están basados en un modelo de tipo Entidad-Relación (ER), en que como el mismo nombre lo indica, está orientado a representar las relaciones entre las entidades de datos, persiguiendo evitar la redundancia de los mismos por razones de integridad y de eficiencia al almacenarlos.

Son utilizados para ingreso en línea, esto es, el registro masivo de información, en que se debe esperar que la información se almacene antes de pasar al siguiente documento. Estos sistemas cumplen con su objetivo que es registrar información y permitir la impresión de facturas, documentos y algunos informes predefinidos que satisfacen las "necesidades operacionales" de la empresa.

Durante la década de los noventa el requerimiento de información comenzó a variar. Se necesitaba poder contar con la información que se encontraba en los sistemas transaccionales, tanto a nivel de máximo detalle o en forma totalizada. Era fundamental también la visión histórica de las variables analizadas y el análisis de los datos del entorno. Estas nuevas necesidades, llamadas "necesidades informacionales", a primera instancia no eran difíciles de resolver dado que la información estaba efectivamente en los sistemas operacionales, ya que cualquier actividad que realizaba la empresa estaba reflejada de forma minuciosa en sus bases de datos. Sin embargo, la situación se hizo más difícil para los departamentos de informática los no daban a basto a la demanda de construcción de informes "especiales" requeridos siempre en forma urgente. Surgió entonces la idea de crear bases de datos estadísticas o Centros de Información que contuviesen toda la información necesaria en forma resumida y detallada. En dicho proceso, hubo que enfrentarse a varios problemas, los cuales están presentes actualmente en la mayoría de las empresas.

El primero fue identificar donde se encontraba la información que se requería para poblar este sistema centralizado. Parte de ella correspondía a la de los sistemas operacionales en uso y aunque muchas veces existía un desconocimiento de la estructura interna de ellos, era posible sin embargo recuperar la información. La dificultad mayor radicaba en aquella que se hallaba dispersa y duplicada entre distintas hojas de cálculos, bases de datos, ficheros de texto, etc., cada una almacenada en distintos formatos y versiones mantenidas por los usuarios. Una vez localizadas todas las fuentes de datos era necesario migrar esta información a una estructura común. Muchas veces esto llegó, y lo hace hoy en día a sobrepasar los recursos de los departamentos que debían construir y mantener los programas creados para este fin.

Pero tener un repositorio con la información necesaria, detallada al máximo y mínimo nivel no bastaba. Se debía poder acceder a ella en forma ágil y eficiente. La primera solución fue que se realizaran una serie de consultas pre-programadas por el departamento de informática y definidas de antemano por los expertos en las distintas áreas. Pero los usuarios siempre querían una nueva consulta o un campo calculado inusual. Surgió entonces una nueva idea. Los mismos usuarios podrían utilizar la herramienta de consulta para que pudiesen construir la consulta que quisieran. Pero no poseían los conocimientos necesarios del lenguaje de SQL y por lo tanto no era la mejor alternativa. También, al realizar consultas sobre toda la información almacenada, se veía perjudicado el nivel de servicio del resto de sistemas, dado que las consultas, solían ser muy costosas en recursos. Además, las necesidades se veían insatisfechas por la limitada flexibilidad a la hora de navegar por la información.

Como solución se planteó la generación de un entorno gemelo del operativo, separado del entorno operacional el cual sería refrescado con menor periodicidad y los requerimientos en el nivel de servicio al usuario serían más flexibles. Sin embargo esto no fue suficiente. La información mantenía la misma estructura que en las aplicaciones operacionales por lo que al realizar una consulta se debía acceder a muchos lugares para obtener el conjunto de datos deseado. El tiempo de respuesta a las solicitudes de información era excesivamente elevado. También, al proceder la información de distintos sistemas, con distintas visiones y distintos objetivos, en muchas ocasiones no era posible obtener la información deseada de una forma fácil y además carecía de la necesaria fiabilidad.

Con todos estas dificultades que presentaban los Centros de Información, aun así ellos respondían a las “necesidades informacionales” de los usuarios. Era necesario hacer algunas modificaciones todavía. Después de casi una década, ellos se transformaron en lo que actualmente se conoce como **Data Warehouse**.

Luego de una serie de análisis se han tomado algunas consideraciones para la construcción de un Data Warehouse: realizar un análisis profundo para identificar las fuentes orígenes, crear estructuras de datos que sean comunes e integren la información, trabajar en un ambiente independiente de los sistemas transaccionales (operacionales). La carga de información se puede realizar en forma manual, a través de creación de programas, o bien, cuando el volumen de información y dispersión de fuentes lo justifican, utilizando herramientas diseñadas para este fin.

Para poder acceder a la información en forma rápida y flexible se ha creado el modelo dimensional (DM), conocido también como **Diagrama en Estrella**, el cual permite trabajar con la información en diferentes niveles de detalle, aunque también se puede optar por un almacenamiento multidimensional.

Con esta tecnología llamada también, **Tecnología de Inteligencia del Negocio**, y las diferentes herramientas de acceso a la información ha permitido al usuario independizarse de los departamentos de informática, pudiendo crear ellos sus propias consultas, informes y realizar los análisis necesarios para la toma de decisiones a los diferentes niveles de la empresa.

Durante estos años, la idea inicial de mantener información centralizada a la que pudiese acceder un grupo específico de personas ha debido ir ampliándose para dar solución a los requerimientos de los diferentes departamentos y tipos de usuario. Es así como se han ido creando bases de datos especializadas y locales, orientadas a dar solución cada problemática, a las cuales se les ha llamado **Datamart**. Los cuales comparten en esencia los mismos principios que un Data Warehouse**.**

El MES no está exento de estas deficiencias informacionales, ni tampoco de las nuevas tecnologías. Teniendo en cuenta esto se estableció:

Trabajar con herramientas de software libre y de código abierto, respondiendo a los intereses de la dirección del gobierno cubano, estimulado por la política que promueve el Ministerio de la Informática y las Comunicaciones de Cuba, así como la Oficina Nacional de Informática, que abogan por una política informática independiente tanto en instituciones, como en entidades gubernamentales.

Diseñar una propuesta de arquitectura, basada en un modelo de datos informacional (de solo lectura), donde se publique la información íntegra, limpia y consistente para la toma de decisiones, a partir de los datos que son enviados desde los distintos CES. La información se almacenará mediante cargas periódicas de datos. Se registra el instante de tiempo del que data la información, se cuenta con información histórica, actual, lo que permite el estudio y comparación estadística, además de análisis de comportamientos y tendencias.

Situación Problémica

En el MES toda la información concerniente a los Portadores Energéticos; que abarca la demanda y la asignación a cada CES tanto de combustible como de electricidad, se tramita y almacena a través de libros de Excel, al igual que toda la información concerniente al Transporte. Lo que hace tedioso el llenado de los mismos, ya que luego de recibir cada archivo de los diferentes CES, se llena otro modelo general, consumiendo mucho tiempo. También se acumulan innumerables libros de Excel que hay que guardar, que dificulta a su vez la búsqueda de información importante en algún momento. Todo esto puede dar como resultado pérdida de información relevante para el MES, así como datos inconsistentes. El MES cuenta de modelos pre-establecidos para cada informe de Demanda (Combustible y Electricidad), Transporte, Plan Operativo (asignación), y otros informes, limitándose así a un reporte de información rígida, que si bien satisface muchas necesidades de información no cubre otras, que pudieran ser en determinado momento, dada alguna necesidad de información específica, una información imprescindible para el CES.

Problema

El MES no cuenta con una estructura para el apoyo en la toma de decisiones en la esfera del transporte y portadores energéticos responda en tiempo a los intereses de los directivos.

Esto está dado por la ausencia de una estructura robusta, que permita almacenar grandes volúmenes de información de forma eficiente, accediendo a ella lo suficientemente rápido y obteniendo disímiles informes; para así facilitar la toma de decisiones de los directivos y la ausencia además de una herramienta que capture la información de manera uniforme que proviene de los CES.

El Objetivo General de esta tesis es:

Desarrollar una propuesta de arquitectura para la recolección, preparación, visualización y procesamiento de la información de cada CES, que facilite la toma de decisiones en la esfera del transporte y portadores energéticos del MES, que abarcará desde el diseño del Datamart, de los procesos ETL y la propuesta de las tecnologías a emplear hasta el diseño e implementación del OLAP.

Objeto de estudio y campo de acción

El objeto de estudio lo constituye:

- Conocer cómo funciona el proceso de la Demanda y Asignación de los portadores energéticos en el Ministerio de Educación Superior, así como la información correspondiente al transporte y sus particularidades.
- Las técnicas y herramientas para la construcción y despliegue de una arquitectura Data Warehouse.
- Diseño y estructura de la base de datos de forma tal que responda a la eficiencia de un Data Warehouse.
- Qué información es la que debe poblar el **Datamart**, para que responda a los intereses informacionales futuros en el MES.
- Los procesos ETL (Extracción, Transformación, Carga) desde los sistemas fuentes al Datamart.

- El estudio del OLAP (Procesamiento Analítico en Línea) como alternativa a la manipulación eficiente de la información.

Campo de acción:

- La utilización de ***UML*** para etapas específicas en un proyecto Data Warehouse.
- El Modelo Multidimensional como disciplina de diseño de las bases de datos para el procesamiento analítico.
- Estudio y comparación de herramientas libres para el desarrollo de un Data Warehouse.
- Análisis de los modelos y tablas para el apoyo en el proceso de toma de decisiones en la esfera del transporte y portadores energéticos del MES.
- Estudio del uso eficiente de los procesos ETL.

Para dar cumplimiento al Objetivo General enunciado se plantean como Objetivos Específicos:

- Definir los procesos de extracción, transformación y carga de los datos de las diferentes fuentes hacia el almacén de datos.
- Diseñar la arquitectura para la toma de decisiones en la esfera del transporte y portadores energéticos del MES.
- Aplicar técnicas para la construcción y despliegue de una arquitectura Data Warehouse, con el empleo del método de diseño basado en UML y en Proceso Unificado: *Data Warehouse Engineering Process* (DWEP).
- Diseñar los modelos de datos necesarios para la toma de decisiones.
- Definir un sistema de apoyo a la toma de decisiones basado en tecnologías y herramientas de la comunidad "*Open Source*".

Para lograr los objetivos se llevaron a cabo las siguientes Tareas:

- Realizar un estudio del estado de arte de las tecnologías para el desarrollo de sistemas para el soporte a la toma de decisiones.

- Determinar los gestores de bases de datos libres adecuados, de acuerdo a las características del proyecto.
- Estudiar las propuestas de ***UML*** para etapas específicas en un proyecto Data Warehouse.
- Identificar los requerimientos de información del transporte y los portadores energéticos del MES.
- Determinar las características de software (sistema operativo) y hardware óptimas para el despliegue de la arquitectura.

Idea a defender:

El desarrollo de una propuesta de arquitectura para la toma de decisiones en la esfera del transporte y portadores energéticos del MES, permitirá el procesamiento analítico de información histórica, actual, agregada y nominalizada en correspondencia con las necesidades de información de todos los niveles de dirección.

Valor Práctico

Este Proyecto se empezará aplicar en el MES siendo de mucho valor práctico:

- Responde a los intereses de una estructura capaz de recopilar y almacenar información de gran volumen en el MES, que brinda de una manera rápida y confiable las estadísticas y las informaciones, para luego ser presentada a los directivos para la toma de decisiones.
- Es un trabajo que responde a las tecnologías de punta para el almacenamiento de datos, integrando las técnicas de base de datos y análisis de datos.
- Es un trabajo realizado con herramientas libres y de código abierto. Lo que disminuye altamente su costo y futuro mantenimiento.

La Novedad Científica se expresa en los siguientes aportes:

- Se desarrolla la arquitectura para el soporte a la toma de decisiones en la esfera del transporte y los portadores energéticos, en el MES.
- Se establecen las fuentes de datos para la realización de estudios futuros, que permitan el descubrimiento de conocimiento oculto que servirá de apoyo a la toma de decisiones con niveles de objetividad muy superiores a los actuales.

- Se desarrolla un proyecto completo de Data Warehouse con herramientas libres y de código abierto.

El presente trabajo se encuentra estructurado en tres capítulos. En el primer capítulo se hace un estudio del estado del arte acerca de los campos de investigación, donde se explican los conceptos fundamentales vinculados a los sistemas para el soporte a la toma de decisiones de la empresa. En el segundo capítulo se describe los sistemas existentes y técnicas actuales. En el tercer capítulo, se desarrolla la propuesta de arquitectura y se documentan los componentes de software desarrollados y el despliegue de la arquitectura.

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS

Introducción

Este capítulo presenta un estudio del estado actual de la tecnología Data Warehouse como arquitectura base para los Sistemas de Soporte para la Toma de decisiones. Se describen las características del modelo dimensional como disciplina de diseño de bases de datos y sus diferencias con el modelo Entidad-Relación. Se establece una comparación entre gestores de bases de datos libre. Se expresan las ventajas de la utilización del OLAP. Se estudian las distintas herramientas ETL de código libre.

1.1. La Inteligencia de Negocio como estrategia para la toma de decisiones

En la actualidad, en cualquier organización se hace necesario la toma decisiones, en ocasiones muy estratégicas para lograr un desarrollo satisfactorio. Generalmente estas decisiones, están basadas en enormes volúmenes de información registrada en bases de datos operacionales o de otros tipos de fuentes de datos. La recopilación y análisis de esta información, dado su carácter heterogéneo y su volumen se convierten usualmente en un problema para las organizaciones y es aquí donde interviene la Inteligencia de Negocio, mediante los Sistemas de Apoyo a la Toma de Decisiones.

Se define por **Inteligencia de Negocio o *Business Intelligence* (BI)** el conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de los datos existentes en una organización o empresa. Este conjunto de herramientas y estrategias tienen en común las siguientes características: **[BI, 2006]**

- *Accesibilidad a la información*. Los datos son la fuente principal. Lo primero que debe garantizar este tipo de herramientas o técnicas será el acceso de los usuarios a los datos con independencia de su procedencia.
- *Apoyo en la toma de decisiones*. Se busca ir más allá en la presentación de la información, de manera que los usuarios tengan acceso a herramientas de análisis que les permitan seleccionar y manipular sólo aquellos datos que les interesen.
- *Orientación al usuario final*. Se busca independencia entre los conocimientos técnicos de los usuarios y su capacidad para utilizar estas herramientas.

Para que una organización sea competitiva, las personas que toman las decisiones necesitan acceder de forma rápida y fácil, a la información, y esto se realiza por medio de los Sistemas de Apoyo a la Toma de Decisiones, o en inglés: *Decision Support Systems* (DSS).
Dichos sistemas se basan en crear modelos informáticos del negocio de modo que este pueda funcionar más eficientemente; y permitiéndole a la dirección de la organización tomar decisiones semiestructuradas o no estructuradas, mediante la combinación de datos y modelos de análisis. Su propósito es ayudar a la dirección para que "marque tendencias, señale problemas y tome decisiones inteligentes". **[Cedeño, 2005]**

Como sistema informático, los DSS consisten habitualmente en varios componentes: bases de datos fuentes, sistemas de Extracción-Transformación-Carga de datos, Data Warehouse (DW) o Almacenes de Datos, herramientas de Procesamiento Analítico en Línea (OLAP), bases de datos multidimensionales y otras herramientas de análisis de información. **[Ruggia, 2002]**

Tomar decisiones implica aceptar un riesgo, lo que es indudable es que el objetivo es minimizarlo. Aquí es donde intervienen las herramientas de Inteligencia de Negocio. Estas son las encargadas de transformar los datos del sistema en información útil para la toma de decisiones empresariales. La organización que carezca de estos sistemas tendría que realizar la transformación y análisis de la información manualmente y solo por el factor tiempo la solución no sería factible, sin contar con que el análisis de los datos sería superficial y por tanto, se perdería información valiosa.

Los datos en un proceso de toma de decisiones, evoluciona por varias etapas. Se comienza con la extracción, limpieza e integración de los datos provenientes de diversas fuentes y se transita por diversos procesos de análisis y búsqueda de conocimiento
(Figura 1.1)

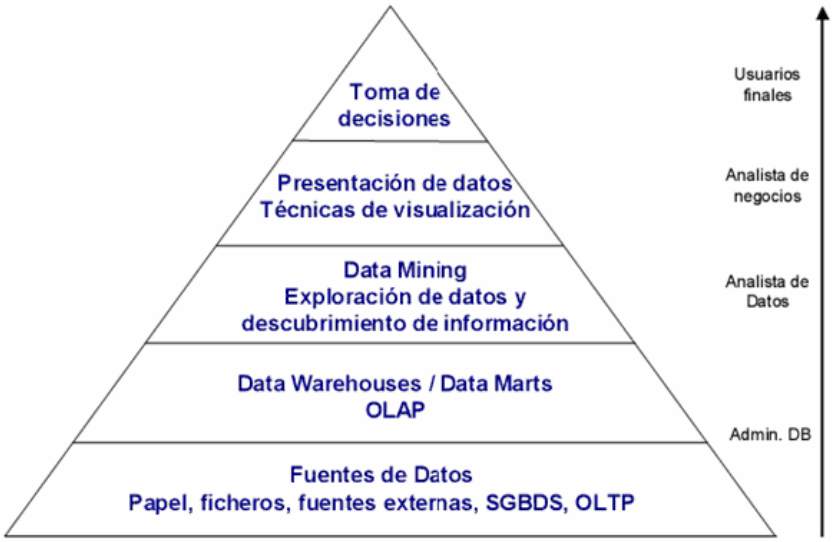


Figura 1. 1 Etapas por las que transitan los datos en un proceso de toma de decisiones.[1]

Tipos de Sistemas de Información: **[BI, 2006]**

- *Sistema Táctico*: Están diseñados para soportar las tareas de coordinación de actividades y manejo de documentación, definidos para facilitar consultas sobre la información almacenada en el sistema, proporcionar informes; en resumen, facilitar la gestión independiente de la información en los diferentes niveles de la organización.
- *Sistema Técnico-Operativo*: Son los que cubren la parte fundamental de las operaciones tradicionales de captura masiva de datos (data entry) y servicios básicos de tratamiento de datos, con tareas predefinidas. Son los llamados Sistemas Operacionales.
- *Sistema Global*: Este nivel de sistemas de información está surgiendo recientemente, y es consecuencia del desarrollo organizacional orientado a un mercado de carácter global, el cual obliga a pensar e implementar sistemas de comunicación más estrechos entre la organización y el mercado. Se convierte en el vehículo de comunicación entre la organización y el mercado con alcance nacional y global.

[1] Tomado de **[BI, 2006].**

- *Sistema Estratégico*: Estos sistemas facilitan la labor de la dirección, proporcionándole un soporte básico, mostrando la información de forma apropiada para la toma de decisiones. El objetivo fundamental es generar información que pueda ser trabajada y analizada de forma intuitiva en tiempo real, y con la posibilidad de integrar diferentes fuentes de datos para ofrecer una visión global que pueda ser compartida y distribuida.

Los sistemas de Inteligencia de Negocio tienen características muy peculiares que los diferencian de los sistemas operacionales. Estos están optimizados para consultar y divulgar la información recogida. Esto significa que los datos están desnormalizados para apoyar preguntas de alto rendimiento, soportados sobre nuevas estructuras de análisis (básicamente multidimencionales). Mientras que, en los sistemas operacionales los datos se normalizan completamente y se basan por lo general en modelos relacionales. Los procesos de ETL (Extracción, Tratamiento y Limpieza de los datos) son los encargados de traducir de un tipo de sistema a otro.

El apoyo para la toma de decisiones, no es parte de la tecnología de base datos por si misma, sino que resulta de la combinación de varias aplicaciones de esta técnica (Figura 1.2). Las aplicaciones reciben los nombres de Data Warehouse, Datamart o Mercado de datos, OLAP (Procesamiento Análitico en Línea), Minería de Datos, entre otros. **[Cedeño, 2005]**

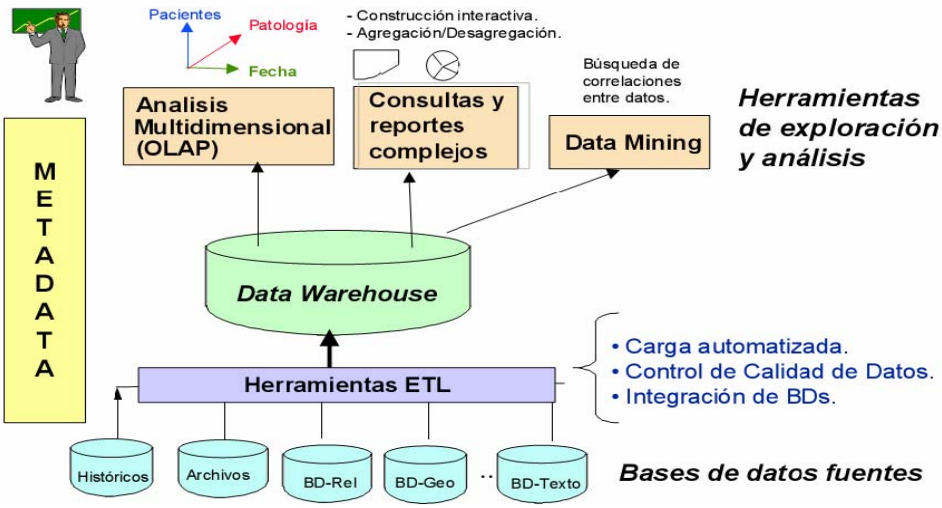


Figura 1. 2 Sistema de Apoyo a la Toma de Decisiones.[1]

[1] Tomado de **[BI, 2006].**

1.1.1. Data Warehouse

La primera condición para poder analizar y extraer información valiosa de un conjunto determinado de datos es disponer de ellos. En algunos casos esto puede parecer trivial pero en ocasiones la recopilación de la información y su tratamiento en el tiempo pueden convertirse en procesos complejos, donde se debe decidir, entre otras cosas, de qué fuentes internas o externas se obtendrán los datos y cómo deben organizarse.

Generalmente, las organizaciones tienen almacenada la información diaria en sistemas operacionales, los que están diseñados para satisfacer determinados requerimientos en el funcionamiento cotidiano de la organización, con tareas muy bien definidas y una carga de procesamiento determinada. Sin embargo, resulta ineficiente utilizar estos sistemas para la toma de decisiones, debido principalmente a su diseño físico y a la imposibilidad de dar respuesta a las complejas consultas que dicho proceso conlleva.

Estas divergencias hacen que sea inadecuado solapar en un mismo sistema el procesamiento operacional con la toma de decisiones de la empresa, fundamentalmente por restricciones de rendimiento, planeación del almacenamiento y administración de recursos; dando paso al surgimiento de los Almacenes de Datos o *Data Warehouse* (DW). **[Cedeño, 2005]**

Los "almacenes de datos" son una tecnología relativamente reciente, encaminada a proporcionar metodologías para recopilar e integrar los datos históricos de una organización, cuyo fin es el análisis, la obtención de resúmenes e informes complejos y la extracción de conocimiento. Esta tecnología está diseñada especialmente para organizar grandes volúmenes de datos de procedencia generalmente estructurada (por ejemplo bases de datos relacionales).

Existen dos paradigmas en el campo del *Data Warehousing*, el paradigma de Bill Inmon y el paradigma de Ralph Kimball, ambos conocidos como los padres del Data Warehouse.

El paradigma de Inmon: La tecnología Data Warehouse, forma parte de los sistemas de inteligencia de negocio. Una empresa debe tener un Data Warehouse y varios Datamarts

que se nutran de la información del Data Warehouse. En un Data Warehouse la información puede estar almacenada en 3ra Forma Normal. **[Kimball, 2007]**

El paradigma de Kimball: Un Data Warehouse es la unión de todos los Datamarts de las diferentes áreas de una empresa. La información se almacena siguiendo un modelo dimensional. **[Kimball, 2007]**

Ambos paradigmas son válidos pero se considera al de Ralph Kimball como el más ajustado a la evolución de esta tecnología dado que la mayoría de las organizaciones por diversos motivos, casi siempre tiempo y costo de producción, comienzan por la implementación de varios Datamarts que posteriormente se integran en un Data Warehouse, y el modelo dimensional se ha convertido en un patrón de diseño muy difundido en esta tecnología.

La siguiente tabla describe las diferencias que se deben tener en cuenta para estructurar y diseñar almacenes de datos en comparación con las bases de datos transaccionales. **[Hernández, 2004]**

Parámetros	Base de Datos Transaccional	Almacén de Datos
Propósito	Operaciones diarias. Soporte a las aplicaciones.	Recuperación de información, informes, análisis y minería de datos.
Tipo de datos	Datos de funcionamiento de la organización.	Datos útiles para el análisis, la sumarización, etc.
Características de los datos	Datos de funcionamiento, cambiantes, internos, incompletos.	Datos históricos, datos internos y externos, datos descriptivos.
Modelo de datos	Datos normalizados.	Datos en estrella, en copo de nieve, parcialmente desnormalizados, multidimensionales.

Número y tipo de usuarios	Cientos/miles: aplicaciones, operarios, administrador de la base de datos.	Decenas: directores, ejecutivos, analistas.
Acceso	SQL. Lectura y escritura.	SQL y herramientas propias (slice & dice, drill, roll, pivot). Lectura.

Tabla 1. 1 Diferencias entre las bases de datos transaccionales y los almacenes de datos**.**

Los Data Warehouse poseen una estructura de la información con un almacenamiento homogéneo y fiable, basado en la consulta y en el tratamiento jerárquico de los datos.
Esta tecnología está caracterizada por 4 aspectos fundamentales: **[Hernández, 2004]**

- *La organización de la información es temática*: los datos están agrupados por temas para facilitar su acceso y entendimiento al usuario final.
- *Integración consistente de los datos*: todas las inconsistencias (nomenclaturas, formatos y tipos de datos, etc.) deben de ser eliminadas.

- *Variante en el tiempo:* la información es exacta para un momento del tiempo, por tal motivo, no deben realizar actualizaciones salvo en raras ocasiones. Mantienen un registro histórico de la información.
- *La información es no volátil*: la información existe para ser leída y no modificada, por lo que, cuando se habla de actualización se está haciendo referencia a una carga incremental sin realizar ningún tipo de acción sobre la información que ya existía.

La organización y el mantenimiento de la información en un Almacén de Datos requiere tomar en cuenta cuestiones técnicas referentes a su ciclo de vida como: selección de esquemas de diseño, definición de los procesos de carga, ciclos de mantenimiento y definición de reglas para preservar la consistencia de los datos.
Un DW pasa a ser un integrador o recopilador de información de diferentes fuentes.

1.1.2. Datamarts

La tecnología Data Warehouse se caracteriza por poseer un enorme poder de extracción de información, y está preferentemente orientada al perfil de toda una organización. En ocasiones se hace necesario aplicar esta técnica a negocios más pequeños que no cuentan con tanta información y la implementación de un DW sería costosa e innecesaria. Por otra parte en algunas organizaciones que ya cuentan con DW surge la necesidad de hacer extracciones de datos de forma departamental y sin mucho gasto de recursos, que solo involucran un subconjunto de la información del DW, por lo que surge la idea de construir un almacén más limitado que facilite el acceso y esté sincronizado con el DW. Esta problemática trajo consigo que surgiera la idea de los Datamarts.

Podemos entender un Datamart como un subconjunto de los datos del Data Warehouse con el objetivo de responder a un determinado análisis, función o necesidad y con una población de usuarios específica. Al igual que en un Data Warehouse, los datos están estructurados en modelos de estrella o copo de nieve y un Datamart puede ser dependiente o independiente de un Data Warehouse.

¿Qué diferencia existe entonces entre un Datamart y un Data Warehouse?

Su alcance. El Datamart está pensado para cubrir las necesidades de un grupo de trabajo o de un determinado departamento dentro de la organización. Es el almacén natural para los datos departamentales. En cambio, el ámbito del Data Warehouse es la organización en su conjunto. Es el almacén natural para los datos corporativos comunes.

Los Datamarts al poseer un propósito más específico y orientado a un sector determinado gozan de innumerables ventajas entre las que se encuentran, su bajo costo de realización y operación, el corto tiempo de diseño y su rapidez para satisfacer los pedidos de información ya que manejan datos resumidos.

1.1.3. Procesos de Extracción, Transformación y Carga de Datos (ETL)

La recopilación de los datos que poblarán los almacenes de datos debe ir acompañada de una limpieza, transformación e integración de los mismos, para que estos estén en condiciones de someterse al análisis y aporten resultados válidos. La validez de los resultados en el proceso de extracción de conocimiento depende en gran medida de la

calidad de los datos analizados, debido a que estos constituyen la materia prima. Para poseer buena calidad en los datos estos deben estar completos, íntegros y consistentes.

Los procesos ETL (*Extraction, Transformation, Load*) son los encargados de ejecutar periódicamente tareas que extraen los datos, los transforman, los integran y los carga al Data Warehouse. Las fuentes de datos que pueden estar vinculadas suelen ser muy diversas, entre estas se pueden encontrar: bases de datos relacionales, ficheros textos, ficheros binarios, ficheros XML (*Extensible Markup Language*), Datamarts, hojas de cálculo, entre otros.

Los procesos ETL se encargan de realizar las siguientes tareas: **[Hernández, 2004]**

- Lectura de las fuentes de datos, ya sean base de datos operacionales o fuentes externas (extracción): se trata, generalmente, de obtener los datos mediante consultas SQL sobre base de datos operacionales. Esto se realiza preferiblemente en horario de poca carga transaccional. En caso de otro tipo de fuente se utilizan herramientas que generan XML o tablas de bases de datos que puedan integrarse en el almacén de datos.
- Creación de claves: se recomienda crear nuevas claves primarias.
- Integración de los datos: al obtener datos de fuentes heterogéneas es necesario definir las restricciones para conectarlos.
- Limpieza y transformación: se trata de evitar datos redundantes e inconsistentes, estandarizar medidas, formatos, fechas, tratar valores nulos, etc.
- Creación y mantenimiento de metadatos: en todo proceso de ETL es necesario guardar la información del propio proceso, así como las descripciones de las tareas a desarrollar, cuáles se deben realizar y en qué momento.
- Identificación de cambios: se puede realizar por carga total, comparación de instancias, análisis de archivos log, marcas de tiempo, empleo de disparadores o técnicas mixtas.
- Planificación de carga y mantenimiento.
- Prueba de calidad: se realiza mediante la aplicación de métricas de calidad a los datos almacenados.
- Carga de datos: es la tarea final del proceso, donde se insertan los datos en el almacén. En este momento se recomienda que el almacén deje de funcionar, por lo que, podría usarse un almacén espejo.

1.1.4. Modelo Multidimensional

El modelo de datos más extendido para la implementación de almacenes de datos es el multidimencional donde los datos se organizan en torno a los hechos (datos históricos) y tienen atributos o medidas que son detalladas en las dimensiones. **[Hernández, 2004]**
Las relaciones entre las tablas de hechos y las dimensiones están representadas en la llave primaria de la tabla de hechos, la que está compuesta por las llaves de todas las tablas de dimensiones con las que tiene relación (llaves foráneas).

Dadas las características de los almacenes de datos es ideal la utilización en su diseño de un Modelo Multidimensional (MMD). Este tipo de diseño tiene como ventajas para la toma de decisiones, sobre el Modelo Entidad-Relación (MER) que es muy flexible, está desnormalizado y orientado a los intereses de un usuario final, aunque esto no significa que existan inconsistencias en los datos. Mediante la utilización de un MMD se disminuyen la cantidad de tablas y relaciones entre ellas, lo que agiliza el acceso a los datos. **[BI, 2006]**

Tablas de Hechos: Representan la ocurrencia de un determinado proceso dentro de la organización y no tienen relación entre sí. Generalmente, almacenan medidas numéricas, las que representan valores de las dimensiones, aunque en ocasiones estas no están presentes y se les denominan “tablas de hechos sin hechos”.

Tablas de Dimensiones: Contienen, generalmente, una llave simple y atributos que la describen. En dependencia del esquema de diseño que se asuma pueden contener llaves foráneas de otras tablas de dimensión. Existe una dimensión fundamental en todo DW, *la dimensión tiempo*. Esto ocurre porque todo registro que se incluya constituye la ocurrencia de un fenómeno en un instante de tiempo definido. Dicha dimensión es la que establece uno de los objetivos fundamentales de la construcción de un DW, la conservación de un “histórico”.

Para la implementación de un MMD existen varios esquemas o patrones de diseño, pero los más utilizados son los siguientes: **[Cedeño, 2005]**

Esquema de Estrella: La tabla de hechos está en el centro de la estrella y están relacionadas con ella de forma radial todas las tablas de dimensiones, las cuales no se relacionan entre sí. No existen caminos alternativos en las dimensiones.

Esquema de Copo de Nieve: Es parecido al de estrella pero existen jerarquías en las dimensiones. Las tablas de dimensiones pueden estar relacionadas, o sea, existen caminos alternativos en ellas.

1.1.5. Metodología para el Desarrollo de Data Warehouse

En los últimos años, el uso de almacenes de datos ha alcanzado gran popularidad en las organizaciones, que cada vez están sometidas a un mercado más competitivo y dinámico. En este sentido, las organizaciones necesitan mejorar la gestión empresarial, hacerla más moderna y competitiva. Para ello, resulta imprescindible acudir al empleo de herramientas que les permitan manipular gran cantidad de información en un tiempo reducido y de forma eficiente. Debido a esta tendencia, cada vez más creciente, muchas personas se han dedicado a proponer modelos para acometer el diseño conceptual y lógico de los almacenes de datos, con el propósito de representar sus principales propiedades estructurales.

A pesar de los esfuerzos realizados no se han podido establecer métodos globales, ni estándares para el diseño del DW. Muchos autores proponen notaciones pero obvian aspectos importantes que no se deben pasar por alto, o sea, no abarcan íntegramente el proceso de diseño e implementación del DW. Entre los diferentes modelos y notaciones existentes se tienen: **[Luján, 2005]**

- *Dimensional-Fact Model*: propone un marco metodológico para definir un esquema conceptual tomando como punto de partida los esquemas de Entidad-Relación de las fuentes operacionales. Esta notación no abarca aspectos tan relevantes como el diseño de los procesos ETL.
- *Multidimensional Model*: modelo lógico que plantea que el diseño de los DW debe estar basado únicamente en las fuentes de información operacional y no se tiene en cuenta que estos sistemas están diseñados para realizar consultas, o sea, no se considera que su diseño debe, tener en cuenta las necesidades de los usuarios finales.

El método global orientado a objetos llamado *Data Warehouse Engineering Process (DWEP)*, basado en el Proceso Unificado de Desarrollo de Software (*Unified Process*) y en el Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML), constituye una propuesta que abarca todas las etapas de la construcción y despliegue de un DW. **[Luján, 2005]** El Proceso Unificado define un proceso de ingeniería de software genérico, establece que el ciclo de vida de un proyecto se divide en 4 etapas (Inserción, Elaboración, Construcción y Transición) y cada una de ellas transita por 5 flujos de trabajo (Requerimientos, Análisis, Diseño, Implementación y Prueba). Adicionalmente en el método propuesto se han agregado dos flujos de trabajo a continuación de los anteriores: Mantenimiento y Revisión Post-Desarrollo (Figura 4). **[Luján, 2005]**

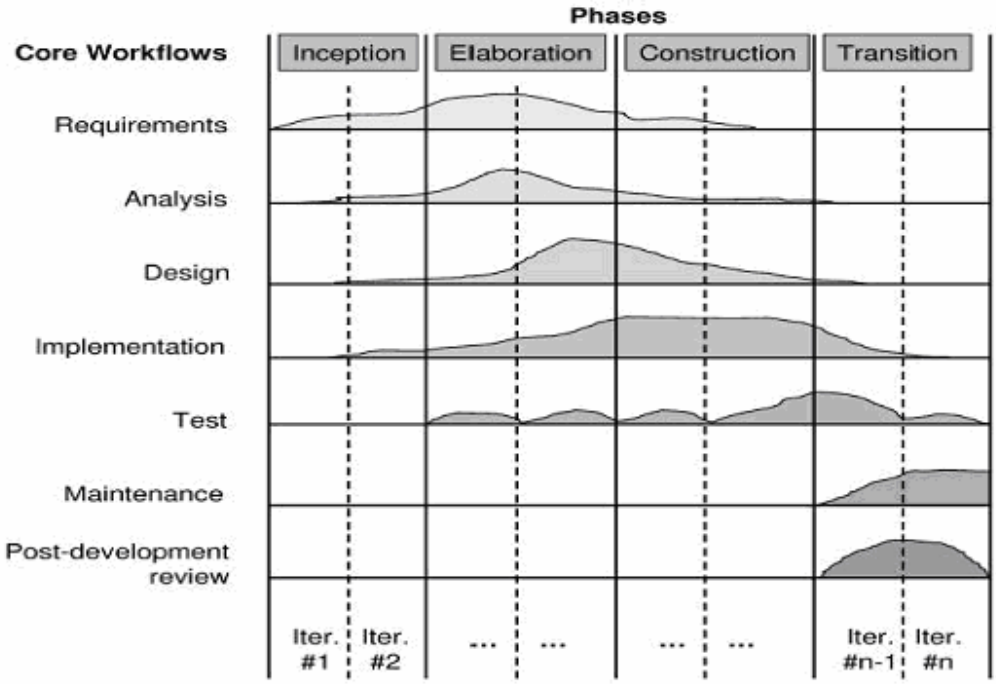


Figura 1. 3 Ciclo de vida de DWEP.[1]

DWEP se plantea los siguientes objetivos:

- Utilizar un lenguaje estándar de modelado (UML).
- Lograr establecer una metodología clara y sin limitaciones de desarrollo de DW.
- Posibilitar el modelado de todas las etapas del proceso, cumpliendo con todos los requerimientos.
- La metodología debe poseer varios niveles de detalle.

[1] Tomado de **[Luján, 2005]**.

DWEP abarca todas las etapas de diseño de un almacén de datos: análisis de las fuentes de datos operacionales, implementación, definición de los procesos ETL y análisis de los requisitos del usuario final. Además, combina un conjunto de modelos basados en UML, evitando que los desarrolladores tengan la necesidad de aprender nuevas notaciones de diseño.

Diagramas del Data Warehouse

El diseño de un DW generalmente se estructura en niveles o capas por las que transitan los datos, el método propuesto en **[Luján, 2005]** se compone de cinco etapas con tres niveles cada una.

Etapas del proceso: **[Luján, 2005]**

- Origen: se definen las fuentes operacionales de datos de las que se va a nutrir el DW. Ejemplo: sistemas OLTP, ficheros, Servicios Web, etc.
- Integración: se definen los procesos de mapeo de datos entre las fuentes de origen y el almacén de datos (Procesos ETL).
- Almacén de Datos: se realiza el diseño de la estructura del almacén de datos.
- Adaptación: se definen los procesos de mapeo entre el almacén de datos y las estructuras utilizadas por el cliente.
- Cliente: se definen de forma concreta las estructuras que usará el cliente para acceder a los datos. Ejemplo: estructuras OLAP.

En cada una de las etapas descritas anteriormente se analizan tres niveles: **[Luján, 2005]**

- Conceptual: se definen las estructuras correspondientes a la etapa desde el punto de vista conceptual, o sea, con un mayor nivel de abstracción, conteniendo solo los elementos más significativos.

- Lógico: incluye los aspectos lógicos del diseño de las estructuras correspondientes a la etapa. Ejemplo: definición de atributos, llaves primarias, procesos ETL, etc.
- Físico: abarca todos los aspectos referentes a la implementación física de las estructuras analizadas y diseñadas en los niveles anteriores.

Cada nivel correspondiente a una etapa incluye el diseño de diferentes tipos de diagramas, los que se basan en una extensión de UML que se adapta al dominio de diseño de este tipo de estructuras. Esta característica constituye la principal ventaja de la metodología.

Proceso de Desarrollo del Data Warehouse

En DWEP el uso de UML y la distribución del desarrollo de artefactos en etapas y niveles aportan una notación gráfica de modelado única. En este punto se hace necesario dotar a los desarrolladores de un método de empleo. Para ello, se propone la utilización de una extensión del ciclo de vida del proyecto del Proceso Unificado.

Descripción de los flujos de trabajo propuestos: **[Luján, 2005]**

- Requerimientos: en este flujo los expertos deben realizar las siguientes tareas: recoger las especificaciones y las reglas del negocio de interés para los usuarios finales del sistema, determinar las medidas y las agregaciones, analizar las posibles dimensiones existentes, determinar el tiempo y la frecuencia de carga de los datos hacia el DW, etc.
- Análisis: el principal objetivo de esta etapa es refinar y estructurar los requisitos y reglas incorporadas en el flujo anterior. También deben identificarse las fuentes de datos operacionales y externas que nutrirán al DW.
- Diseño: en este flujo se define la estructura del DW, dado que, la principal salida la constituye el esquema conceptual. Esta tarea tiene dos formas de ejecutarse:
    1. T*op-Down*, que plantea definir el esquema a partir de los requisitos de los usuarios finales.
    2. *Bottom-Up*, que pretende definirlo a partir de los requisitos de los datos provenientes de las diversas fuentes de datos.

  Además, en este flujo debe de definirse conceptualmente los procesos de extracción, transformación y carga de datos (mapeo de los datos) aplicando las reglas del negocio y las consultas iniciales.
- Implementación: en este flujo se construye de forma física las estructuras que componen el DW, se define el tipo de almacenamiento, el esquema lógico, los procesos ETL y se implementan los informes. O sea, se crean físicamente los componentes diseñados en el flujo anterior.
- Prueba: el objetivo de este flujo es verificar a partir de pruebas del sistema el cumplimiento de las tareas diseñadas e implementadas en flujos anteriores.
- Mantenimiento: este flujo comienza cuando el DW es construido y puesto en uso por el usuario final. No tiene fecha de fin, está vigente durante todo el ciclo de vida del DW.

En este flujo el usuario final puede solicitar la incorporación al sistema de nuevos requisitos y reglas del negocio, dando inicio a otra iteración del proceso completo.

- Revisión Post-Desarrollo: en este flujo los desarrolladores deben volver atrás para revisar las tareas realizadas anteriormente, la documentación, etc. Deben analizar las oportunidades existentes de introducir mejoras al sistema con el fin de lograr un mejor desempeño.

  Un aspecto importante sería evaluar el costo en tiempo y esfuerzo empleado en las diferentes etapas, con el objetivo de poder estimar estos parámetros en futuros proyectos.

1.1.5.1. Modelo Conceptual

Generalmente, los modelos conceptuales de las bases de datos se describen con el uso de Diagramas Entidad-Relación, los que proponen una vista del universo de datos a partir de su representación mediante entidades y relaciones **[Luján, 2005]**. El uso de este artefacto facilita la comprensión del modelo sin exigir mucho adiestramiento, pero presenta algunas desventajas: no representa un estándar debido a que han surgido diferentes extensiones para la notación, en diseños de modelos complejos resultan de difícil comprensión y el acercamiento al modelado de diseños orientados a objetos es débil, Ejemplo: representación de relaciones de herencia y composición.

Como variante de solución surge la representación de los diagramas conceptuales a partir de UML, específicamente mediante diagramas de clases **[Luján, 2005]**. El uso de este tipo de artefacto permite a los diseñadores de bases de datos realizar modelos más flexibles con la inclusión de: paquetes, entidades, atributos y relaciones de diferentes tipos; además de crear un estándar de modelado con el uso de una notación ampliamente conocida por los desarrolladores de sistemas y permitir un acercamiento al diseño de modelos orientados a objetos.

El proceso de modelado se divide en tres niveles: **[Luján, 2005]**

- **Nivel 1**: se define el modelo conceptual de forma global. Se realiza a partir de la representación de esquemas de estrella como paquetes y relaciones. Las relaciones indican que los esquemas implicados comparten al menos una dimensión. Debe seleccionarse adecuadamente el paquete principal con el objetivo de minimizar las dependencias y evitar ciclos.

- **Nivel 2**: se define la estructura de cada una de las estrellas. Se representan mediante paquetes y relaciones correspondientes a los hechos y dimensiones existentes. Las relaciones entre dimensiones implican que las dimensiones comparten, al menos, un nivel de sus jerarquías (esquema copo de nieve).
- **Nivel 3**: se define la estructura de todos los hechos y dimensiones. Se representan empleando un conjunto de clases referentes a cada nivel jerárquico en las dimensiones y en el caso de los hechos mediante el esquema de estrella completo.

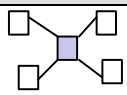
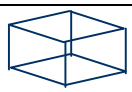
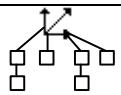
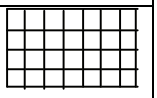
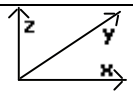
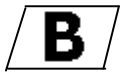
Estereotipo	Descripción	Icono
StarPackage	Paquete que representa el esquema de estrella, empleado en el nivel 1 para diseñar el modelo conceptual global del DW.	
FactPackage	Paquete que representa los hechos, empleado en el nivel 2 para diseñar la estructura de cada uno de los esquemas de estrella.	
DimensionPackage	Paquete que representa las dimensiones, empleado en el nivel 2 para diseñar la estructura de cada uno de los esquemas de estrella.	
Fact	Clase que representa los hechos, empleado en el nivel 3 para diseñar la estructura de cada uno de los paquetes de hechos.	
Dimension	Clase que representa las dimensiones, empleado en el nivel 3 para diseñar la estructura de cada uno de los paquetes de dimensiones.	
Base	Clase base que representa los niveles jerárquicos en cada dimensión, empleado en el nivel 3.	

Tabla 1. 2 Estereotipos empleados en el modelo conceptual del DW.

1.1.5.2. Modelo Lógico.

El modelo lógico debe basarse en los resultados de los diagramas realizados en el Modelo Conceptual.

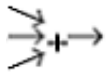
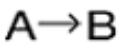
En esta tarea, se realiza un mapeo entre los artefactos desarrollados en el modelo anterior (Modelo Conceptual) y los correspondientes a este modelo, o sea, a partir de las clases y relaciones se obtienen tablas, columnas, tipos de datos, llaves, índices, dependencias, relaciones y vistas. El proceso de mapeo no se realiza a todos los elementos representados de forma conceptual, solo a los que lógicamente pueden ser representados en la base de datos. Ejemplo: los atributos calculables no se incluyen en la base de datos. En este modelo también se incluye la definición de los procesos ETL.

Formalmente, un proceso ETL puede definirse como: un flujo de trabajo que comienza con la extracción de los datos de un sistema fuente, los que son sometidos a un conjunto de transformaciones y finalmente son cargados en un DW. Los procesos ETL son los encargados de la extracción de los datos de las diferentes fuentes de información. Estos procesos no consisten en un simple mapeo de datos de una estructura a otra, sino que en su ejecución se realizan tareas de transformación, normalización y limpieza con el fin de proporcionarle consistencia a los datos y aplicar reglas o requerimientos del negocio en cuestión. El paso final de este proceso es la carga de los datos hacia el DW.

El diseño de un proceso ETL generalmente se divide en 6 etapas: **[Luján, 2005]**

1. Selección de las fuentes de datos de extracción.
2. Realización de transformaciones: se analiza la calidad de los datos provenientes de las fuentes. A continuación se transforman los datos para ajustarlos a las necesidades y reglas establecidas para el sistema. Entre las principales transformaciones están: filtrado, conversión de tipos de datos, calcular valores derivados, generación de secuencias, etc.
3. Unir las fuentes de datos que están relacionadas a través de uno o más atributos. Lograr unificar los datos que posteriormente serán cargados.
4. Seleccionar el destino de los datos. Pueden ser varios.
5. Realizar el mapeo de los atributos de los datos fuentes hacia los atributos de la estructura destino.
6. Cargar los datos.

La metodología de diseño de almacenes de datos DWEP propone una extensión de UML que permite el modelado de los procesos ETL. Para ello se ha definido el siguiente conjunto de estereotipos:

Estereotipo	Descripción	Icono
Aggregation	Agrega los datos en base a algún criterio. (SUM, AVG, MAX/MIN, COUNT, etc.)	
Conversion	Genera atributos derivados a partir de datos existentes: cambio de formato o tipo de datos, datos calculables, etc.	

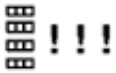
Filter	Filtra los datos a partir de restricciones.	
Incorrect	Redirige los registros incorrectos a un destino separado para someterlos a transformaciones.	
Join	Une dos fuentes de datos que están relacionadas por sus atributos.	
Loader	Carga los datos en las estructuras del DW.	
Merge	Integra los datos de varias fuentes con atributos compatibles.	
Surrogate	Genera una clave sustituta.	
Log	Registra las operaciones realizadas en el proceso ETL. Se utiliza fundamentalmente para monitorear las transformaciones.	

Tabla 1. 3 Estereotipos empleados en el modelo de los procesos ETL.

1.1.5.3. Modelo Físico

La metodología de diseño de almacenes de datos DWEP propone para la realización del modelo físico del DW mediante el empleo de diagramas de componentes y diagramas de despliegue de UML. El uso de estos artefactos permite tomar decisiones en tiempo de diseño del DW con el objetivo de minimizar el tiempo total de desarrollo del proyecto. Las decisiones pueden estar orientadas a los siguientes aspectos: almacenamiento de las tablas de hechos en varios discos duros (debido al gran volumen de datos que almacenan), realizar copias de tablas, utilizar particionamiento vertical u horizontal de la base de datos, etc. **[Luján, 2005]**

Estas consideraciones son propias de este modelo, ya que los temas referentes al despliegue físico del DW no se abordan en ninguna de las etapas anteriores.

1.1.6. Procesamiento Transaccional en Línea y Procesamiento Analítico en Línea (OLTP y OLAP)

El desarrollo de los sistemas de información sustentados sobre bases de datos, ha traído como consecuencia la proliferación de herramientas de consultas cada vez más

complejas. Por tanto, es necesario distinguir los diferentes tipos de procesamiento existentes: el procesamiento transaccional y el procesamiento analítico.

OLTP (*On-Line Transactional Processing* o Procesamiento Transaccional en Línea), trabajo primario en un sistema de información. Consiste en realizar transacciones, es decir, actualizaciones y consultas a la base de datos con un objetivo operacional: hacer funcionar las aplicaciones de la organización, proporcionar información sobre el estado del sistema y permitir actualizarlas.

OLAP (*On-Line Analytical Processsing* o Procesamiento Analítico en Línea), reúne un gran número de operaciones (solamente de consulta), en las se cruzan gran cantidad de información con el objetivo final de crear informes y resúmenes que sean útiles en la toma de decisiones. Los algoritmos que utiliza están implementados para optimizar los tiempos de respuesta a las consultas, logrando eficiencia y almacenando los datos en estructuras especializadas.
OLAP fue creado bajo las siguientes ideas:

- *Lograr rapidez de respuesta*: entregar la información a los usuarios finales en el menor tiempo posible, de 0 a 5 segundos.
- *Posibilitar el análisis*: Ofrecer análisis numérico y estadístico de los datos, con valores agregados. Esto permite analizar tendencias, causas, detectar variables de interés y descender hasta los niveles más bajos de la información, lo que se complementa con la ayuda de los motores de reportes y gráficos que se incluyen. También incluye vistas personalizadas.
- *Compartir Datos*: Incluye los mecanismos de seguridad necesarios para compartir la información entre los usuarios que se definan.
- *Basado en un Estructura Multidimensional*: Haciendo sencilla la selección y navegación de los datos.
- *Recuperación de Información*: Acceso a los datos y recuperación de información valiosa (solo lectura) para las diferentes aplicaciones clientes.

Existen tres tipos de OLAP: **[OLAP, 2007] [Hernández, 2004]**

ROLAP (*Relational OLAP*): se construye el almacén de datos directamente sobre un gestor de base de datos relacional, todas las tablas (hechos y dimensiones) son almacenadas en tablas relacionales.

MOLAP (*Multidimensional OLAP*): es la forma clásica de OLAP. Se construye el almacén de datos directamente sobre estructuras matriciales multidimensionales. Se almacenan las agregaciones y una copia de los datos bases. Una vez realizada la carga, el motor de MOLAP se encarga de brindar la información detallada y agregada. Solo se requiere la participación del servidor de bases de datos cuando se vuelvan a procesar los datos.

HOLAP (*Hybrid OLAP*): usa tablas relacionales para almacenar la información base y estructuras multidimensionales para las agregaciones, o sea, es una combinación de los dos anteriores.

Cada uno de los tipos de OLAP tienen beneficios en dependencia del problema en que se aplique. MOLAP requiere de menor espacio de almacenamiento y es más rápido calculando las agregaciones y devolviendo las respuestas, aunque se recomienda emplear para pequeños volúmenes de datos. ROLAP es considerado el más escalable, pero es más lento en el preprocesamiento y rendimiento de las consultas. HOLAP es rápido en el preprocesamiento y rendimiento de las consultas, aunque más lento que MOLAP y es escalable. HOLAP es ideal para grandes fuentes de datos.

OLTP y OLAP son dos tipos de procesamiento bien distintos, aunque ambos se realizan en tiempo real. Hasta hace poco tiempo en las organizaciones ambos se realizaban sobre las mismas bases de datos transaccionales, lo que traía consigo grandes problemas, debido a que las complejas consultas OLAP que involucraban muchas tablas y agregaciones perturbaban el trabajo de estos sistemas, causando demoras innecesarias y consumiendo muchos recursos. Además, los diseños que se asumen en las bases de datos transaccionales no son los más adecuados para realizar consultas OLAP. Actualmente, esto dejó de ser un problema debido al surgimiento de los DW.

1.1.7. Camino hacia el Software Libre

Hace ya algún tiempo en el mundo ha ido creciendo vertiginosamente un movimiento de personas que desarrollan, utilizan y promueven soluciones basadas en software libre.

Entiéndase por *software libre:* el software que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Guía Activa del Software Libre, se resumen cuatro libertades fundamentales: **[BI, 2006]**

- *Libertad 0*: libertad para ejecutar los programas sean cuales sean los propósitos.
- *Libertad 1*: libertad del usuario para estudiar el programa y adaptarlo a sus necesidades. El acceso al código fuente es condición indispensable.
- *Libertad 2*: posibilidad de distribuir cuantas copias se deseen.
- *Libertad 3*: posibilidad de mejorar el programa, así como, publicarlo para que toda la comunidad de usuarios pueda acceder a él.

El software libre suele estar disponible de varias formas, ya sea gratuitamente en Internet, o a precio del costo de distribución a través de otros medios. Este tipo de software aunque conserve su carácter de libre, puede ser vendido comercialmente. **[BI, 2006]**

Por otra parte, el llamado *software gratis* o *Freeware* evidencia un grado de libertad diferente al software libre ya que incluye algunas concesiones con respecto al código fuente (que puede estar incluido o no), a menos que se garanticen las licencias pertinentes para su modificación y la distribución de nuevas versiones modificadas. **[BI, 2006]**

Estos términos inicialmente se utilizaban solo dentro de la comunidad del software libre, pero en la actualidad se extienden cada vez más, y surgen muchas inquietudes entorno a la ambigüedad de sus definiciones. Por tal razón, es que aparece el término *open source software* (software de código abierto), donde su principal diferencia respecto al software libre radica en que éste tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "*Open Source*" se basa únicamente en los aspectos técnicos (compartir el código fuente como objetivo primario).

La principal ventaja del código abierto radica en que, permite que varios programadores cooperen por libre asociación formando equipos internacionales que pueden leer, modificar y redistribuir el código fuente de un programa, y de esta forma lograr una evolución acelerada del producto. La realidad es que este movimiento ha ganado mucha fuerza y se ha convertido en un fuerte competidor del software propietario, dada las soluciones que proponen cada vez más robustas y escalables.

El gobierno cubano se ha pronunciado al respecto, manifestando la necesidad de fomentar el desarrollo de soluciones basadas en software libre, lo que se evidenció en el 1er Taller Latinoamericano para la promoción del uso de Software Abierto en Educación, Ciencia, Cultura y Áreas Sociales celebrado en la Habana en junio del 2001, donde se planteó: *"es de vital importancia incluir los aspectos de software libre en las políticas que se tracen para favorecer la construcción de un mundo sin exclusión social y con igualdad de acceso a los programas tecnológicos y a la información"*.
Entre las tareas que se piensan realizar, está la migración de las aplicaciones hacia tecnología de software libre y utilizar como base el Sistema Operativo Linux.

1.1.7.1. Sistema Gestores de Bases de Datos Libres

Entiéndase por Sistema Gestor de Base de Datos (SGBD) un sistema que permite el procesamiento de la información almacenada en bases de datos, así como el acceso a la misma por varios usuarios en forma concurrente mediante la realización de peticiones. **[Cedeño, 2005]**
La política de diseño a utilizar debe estar basada en herramientas pertenecientes a la comunidad del software libre por lo que, el SGBD debe ser libre. Debido a la gran cantidad de información que se pretende guardar en el almacén de datos y a la necesidad de minimizar los tiempos de respuestas a las peticiones se debe seleccionar un SGBD robusto que pueda satisfacer estas necesidades.

En la siguiente tabla se muestra una comparación entre dos de los servidores de bases de datos líderes dentro de la comunidad del software libre: MySQL y PostgreSQL.

Parámetros	MySQL 5.x	PostgreSQL 8.0.x
Sistema Operativo	Linux, Windows, FreeBSD, MacOS X, Solaris, HP UX, AIX, entre otros.	Windows, más de 2 docenas de la familia Unix (Linux, todos los BSDs, HP-UX, AIX, OS X, Unixware, Netware...)
Soporta Subconsultas	Sí	Sí
Soporta Transacciones	Sí	Sí
Replicación	Sí	Sí
Soporta Llaves Foráneas	Sí	Sí
Vistas	Sí	Sí
Procedimientos Almacenados	Sí	Sí
Disparadores (Triggers)	Sí	Sí
Full joins	No	Sí
Restricciones (Constraints)	No	Sí
Cursores (Cursors)	Parcial (Solo lectura)	Sí
Dominios definidos por el usuarios	No	Sí
Lenguajes procedurales	ANSI SQL 2003	PL/pgSQL, PL/Tcl, PL/Perl, PL/Python PL/PHP, PL/Java o definido por el usuario
ODBC	Sí	Sí
JDBC	Sí	Sí
Velocidad en las Consultas (grandes volúmenes de datos)	Lento	Rápido

Tabla 1. 4 Comparación entre MySQL 5.x y PostgreSQL 8.0.x

A partir de la tabla comparativa se puede concluir que ambos gestores son estables, robustos y escalables, poseen además amplio respaldo de la comunidad de desarrolladores, por lo que su mantenimiento y desarrollo están garantizados. Aunque existe diferencias que permiten considerar a PostgreSQL el más adecuado para el propósito de este trabajo:

- PostgreSQL es más eficiente a la hora satisfacer peticiones que involucren grandes volúmenes de datos, lo cual constituye un requisito indispensables para un sistema de toma de decisiones.

- Ofrece más funcionalidades y tiene mayor experiencia en módulos que fueron incluidos en MySQL recientemente. **[MySQL, 2007].**
- Manipula mayor cantidad de usuarios concurrentes.
- Agregación en memoria usando tablas de hashing para hacer que las consultas de data-warehousing y OLAP sean más rápidas. **[PostgreSQL, 2006].**
- Implementa módulos adicionales para el trabajo con cubos en el entorno multidimensional.
- Mayor soporte de los estándares SQL 92 y SQL 99 que MySQL.

1.1.7.2. Herramientas de Inteligencia de Negocio "Open Source"

La comunidad "*Open Source*" hoy en día ha incursionado prácticamente en todas las áreas de la informática y existen algunas donde su supremacía es indudable. En el área de la Inteligencia de Negocio también se ha producido un despegue en el desarrollo de soluciones.

Una solución basada en Inteligencia de Negocio, de acuerdo a sus características debe de estar soportada por un conjunto de herramientas donde se establece una cooperación entre ellas para transitar por las diferentes etapas del proceso de análisis de los datos, desde la adquisición hasta la visualización de los resultados.
Entre las principales herramientas disponibles están:

- Herramientas ETL: Kettle, Clover, Enhydra Octopus, etc.
- Desarrollo OLAP: Mondrian y JPivot.
- Minería de Datos: WEKA, YALE y otras herramientas con versiones libres limitadas como: Tiberius, WizWhy, CART y See5 / C5.0.
- Motores de Reportes y Gráficos: JFreeReport, BIRT, JasperReport y JFreeChart.
- Entorno de desarrollo para Cuadros de Mando (*Dashboards*): JetSpeed y JBoss Portal.
- Gestores de Bases de Datos: MySQL y PostgreSQL.
- Soluciones completas: Pentaho y SpagoBI.

En este trabajo para el diseño de la solución se necesita de un SGBD para el almacenamiento y manipulación de los datos, de una herramienta ETL, de un servidor

OLAP y de un motor generador de reportes y gráficos. Para la selección de las herramientas adecuadas se realizó un análisis comparativo.

Herramienta ETL

En la siguiente tabla se muestra el análisis comparativo realizado entre las herramientas más significativas y populares disponibles para realizar procesos de ETL. **[Enhydra, 2007] [Talend, 2007] [ETL, 2007]**

Herramientas	Descripción	Fuentes de Datos	Multiplataforma
Enhydra Octopus	Permite crear bases de datos, tablas, índices y relaciones. Permite insertar y actualizar datos de acuerdo a varios criterios, así como generar llaves artificiales. Las transformaciones son descritas en ficheros XML.	Conexión mediante JDBC con cualquier base de datos que posea dicho driver. Ficheros de Texto: CSV (CSV-JDBC), MS-SQL (FreeTDS), XML (XML-JDBC) y archivos de propiedades (i18n-JDBC).	Sí (basado en Java)
Talend	Permite hacer transformaciones y mapeos complejos. Posee un entorno gráfico aunque las transformaciones deben ser programadas íntegramente (Perl), no es muy intuitivo.	Conexión mediante JDBC con cualquier base de datos que posea dicho driver. Gran integración con MySQL.	Sí (basado en Java e integrable con Eclipse)
CloverETL	Las transformaciones son descritas en ficheros XML y ejecutas de forma independiente (un hilo de ejecución para cada una). Permite una amplia variedad de transformaciones.	Conexión mediante JDBC con cualquier base de datos que posea dicho driver como: Oracle, MySQL, MS Excel, etc. Acceso a ficheros como: dBase, FoxPro y FlashFile. Posee componentes para la conexión con LDAP.	Sí (basado en Java)

Kettle	Permite realizar una amplia variedad de transformaciones y posee soporte para el trabajo con almacenes de datos. Posee un entorno gráfico muy intuitivo. Permite el monitoreo de las tareas ejecutadas. Las tareas son ejecutadas en hilos diferentes y su prioridad puede ser administrada.	Puede establecer conexión mediante: JDBC, ODBC y JNDI, con alrededor de 25 tipos de gestores de bases de datos. Permite acceder a ficheros: CSV, XML, de texto personalizado, Excel, etc. Permite conexión con Servicios Web.	Sí (basado en Java)
KETL	Las transformaciones son descritas en ficheros XML. Las tareas son ejecutadas en hilos diferentes permitiendo el paralelismo en su realización.	Conexión mediante JDBC con cualquier base de datos que posea dicho driver.	Sí (basado en Java)
OpenAdaptor	La plataforma es altamente integrable y sus componentes pueden utilizarse de forma independiente. Las transformaciones son descritas en ficheros XML.	Conexión mediante JDBC con cualquier base de datos que posea dicho driver. Acceso a ficheros XML.	Sí (basado en Java)
Scriptella	Las transformaciones son descritas en ficheros XML. El rendimiento y el bajo uso de memoria son sus objetivos principales.	Conexión mediante JDBC con cualquier base de datos que posea dicho driver. Acceso a ficheros: Texto, CSV y XML. Posee componentes para conexión con LDAP.	Sí (basado en Java)

Tabla 1. 5 Análisis comparativo entre herramientas ETL.

Es válido señalar que aunque estas herramientas poseen muchos puntos de divergencia también presentan aspectos semejantes entre los que se puede señalar: la independencia de la plataforma debido a que todas se basan en Java, la posibilidad de conexión mediante JDBC con las diferentes fuentes de datos y la descripción de las transformaciones en ficheros XML.

La herramienta ETL seleccionada es Kettle debido a la diversidad de fuentes de datos que soporta (mediante JDBC, ODBC y JNDI), incluyendo la comunicación con Servicios Web. Debe señalarse también aspectos de gran importancia como: la amplia variedad de

transformaciones incluyendo el trabajo con almacenes de datos (implementación de *"Slowly Changing Dimensions"*), la ejecución en paralelo de las tareas permitiendo la asignación de prioridades, el monitoreo de las tareas en ejecución y la posibilidad de diseñar todo el proceso de forma visual haciendo uso de una interfaz de usuario muy amigable e intuitiva.

Servidor OLAP

En la actualidad no existen muchos proyectos pertenecientes a la Comunidad *"Open Source"* encaminados al desarrollo de servidores OLAP. Existen varias herramientas orientadas a propósitos muy bien definidos y específicos, como es el caso de PALO (OLAP al revés), que trabaja fundamentalmente con hojas de cálculo tipo Excel y posee una alta eficiencia. Es válido destacar que se han dado grandes pasos en el desarrollo de herramientas para la visualización y manipulación de los resultados de consultas OLAP, como es el caso de: OpenI, JPivot, etc. **[BI, 2006]**

En la comunidad sobresale el proyecto del Servidor OLAP Mondrian registrado en el año 2001, que constituye el paso más serio dado al respecto por sus prestaciones, estabilidad y escalabilidad. **[Mondrian, 2007]**

Mondrian, es un servidor OLAP libre y multiplataforma pues se basa en Java. Permite el acceso a bases de datos relacionales y emplea como lenguaje de consulta MDX. Cumple con la arquitectura ROLAP.

Mondrian está estructurado en 4 capas con tareas bien definidas y que cooperan entre sí, ellas son: capa de presentación, capa dimensional, capa principal y capa de almacenamiento. **[Mondrian, 2007]**

La capa de presentación determina qué información será mostrada al usuario y de qué manera (tablas interactivas, gráficos de pastel, barras o líneas, etc.) para que pueda interactuar con ella y generar nuevos resultados. La capa dimensional es la encargada de analizar, validar y ejecutar las consultas MDX, lo que se realiza en varios pasos. La capa principal es la encargada de administrar la memoria caché de agregaciones. Su tarea consiste en recibir las peticiones de la capa dimensional y verificar si está cargada en caché; en caso contrario, enviar la petición a la capa de almacenamiento. La capa de almacenamiento es un gestor de bases de datos responsable de cargar los datos almacenados y agregados (hechos y dimensiones) de un gestor relacional. **[Mondrian, 2007]**

Los componentes de estas capas pueden estar todos distribuidos o desplegados en la misma máquina. Las capas dimensional y principal sí deben estar desplegadas juntas, pero la capa de almacenamiento podría estar en otra máquina y acceder a ella por vía remota JDBC. En sistemas multi-usuarios la capa de presentación podría estar desplegada en cada estación.
Mondrian sigue la siguiente estrategia de agregación: **[Mondrian, 2007]**

- Los hechos son almacenados en un gestor de bases de datos relacional. Los desarrolladores plantean que no debe desarrollarse un manipulador de datos almacenados si el gestor de base de datos puede realizar esta tarea.
- La lectura de los datos agregados almacenados en la memoria caché se realiza mediante consultas que utilizan GROUP BY. De igual forma plantean que no es necesario desarrollar un manipulador de agregaciones si el gestor de bases de datos puede realizar esta tarea.
- Si el gestor de base de datos soporta la ejecución de vistas y el administrador escoge crear vistas para agregaciones particulares, Mondrian las usará de forma implícita.

La idea general es delegar a la base de datos todas las tareas que por naturaleza le corresponden, de esta manera, se carga en cierta medida el servidor pero los clientes se benefician, tornándose ligeros.

Mondrian proporciona una API (*Application Programming Interface*) para aplicaciones clientes destinada a ejecutar consultas. La API no goza de una amplia aceptación debido a que utiliza un lenguaje distinto para especificar las consultas llamado MDX (*Multi-Dimensional Expressions*). MDX es un lenguaje de consultas para bases de datos con un diseño multidimensional, en la misma manera que SQL es usado para consultar bases de datos relacionales.

Para la visualización y manipulación de los resultados de las consultas OLAP se cuenta con JPivot, basada en la tecnología Java Server Page (JSP). JPivot es una biblioteca personalizada de etiquetas JSP que permite visualizar tablas OLAP y realizar sobre ellas operaciones típicas como: slice, drill down, roll up, etc. **[Mondrian, 2007]**

Soluciones completas de Inteligencia de Negocio

Una solución completa de Inteligencia de Negocio es un proyecto integrador de herramientas que cooperan entre sí, logrando una alta cohesión entre las partes y permitiendo modelar íntegramente cualquier proceso de búsqueda de conocimiento en una organización determinada. A este tipo de herramienta dentro de la comunidad de desarrolladores se le conoce como: "paraguas" o formalmente como: BPM (*Business Performance Management*).

Elegir un BPM para diseñar una solución de BI (*Business Intelligent*) es de vital importancia, desde el punto de vista del ahorro del tiempo que debe emplearse para adquirir las herramientas independientes y tratar de integrarlas de forma coherente.

Como ejemplos de soluciones completas están: Pentaho y SpagoBI. **[OpenSource, 2007]**

Pentaho: es una de las más completas de todas las soluciones, integra toda una serie de componentes Open Source muy utilizados y fiables: Servidor OLAP Mondrian, JPivot, Kettle ETL, Enhydra Shark, Quartz, Weka, JFreeReport, JFreeChart, JBoss Portal, entre otros.

SpagoBI: es una solución completa de Inteligencia de Negocio que incluye desde la extracción de los datos hasta la minería, pasando por análisis y la generación de reportes. Utiliza componentes muy similares a los de Pentaho.

Herramientas	Herramientas Incluidas	Posibilidad de Extensión
Pentaho	Servidor de Aplicaciones: JBOSS y Tomcat. Motor de Reportes: JFreeReport y BIRT. OLAP: Mondrian y JPivot. Minería de Datos: WEKA. ETL: Kettle. Planificador: Quartz.	Sí

SpagoBI	Servidor de Aplicaciones: JBOSS y Tomcat. Motor de Reportes: JasperReport y BIRT. OLAP: Mondrian y JPivot. Minería de Datos: WEKA. ETL: Talend. Planificador: Quartz.	Sí

Tabla 1. 6 Comparación entre Pentaho y SpagoBI.

Para elegir un BPM es necesario tomar en cuenta algunos aspectos: **[BI, 2006]**

- Debe permitir hacer **análisis predictivo**, con el objetivo fundamental de hacer proactivo el proceso de toma de decisiones.
- Debe permitir ejecutar **tareas de control** mediante la implementación de Reglas de Negocio.
- Debe posibilitar la **unificación e integración** tanto de la información manipulada como de las diferentes etapas o procesos de la solución diseñada. **Facilidad de uso**, debido a la diversidad de usuarios que potencialmente harán uso del sistema. Por esta razón, debe encaminar esfuerzos a elevar el nivel de usabilidad en la ejecución de tareas como: el diseño e implementación de la solución, adquisición de los datos y generación de resultados.
- Debe incorporar grandes **posibilidades de workflow,** con el objetivo de controlar y programar las diferentes tareas a ejecutar, así como, establecer la prioridad de las mismas.
- Poseer **autonomía en cada nivel**, debe permitir la gestión de cada etapa del proceso de forma independiente, aunque deben comunicarse con facilidad. Cumplir con el principio de alta cohesión y bajo acoplamiento entre las diferentes herramientas y componentes.
- **Establecer un entorno de colaboración**, es decir, permitir compartir información entre los usuarios del sistema de forma ágil y segura.
- **La velocidad de consulta**, esto debe ir en correspondencia con los requerimientos de la organización, el volumen de información manejada y la cantidad de usuarios que utilizarán el sistema.

- **Servicios de soporte y ayuda**, se debe contar con el respaldo de la comunidad de desarrolladores ante la ocurrencia de fallos en el sistema.

Pentaho y SpagoBI constituyen herramientas BPM, pues integran la mayoría de los componentes necesarios para modelar un proceso de toma de decisiones. Ambas cumplen con los aspectos anteriormente expuestos, pues utilizan componentes muy similares, con la diferencia de la herramienta ETL: Pentaho incluye Kettle y SpagoBI incluye Talend. Esta diferencia hace a Pentaho superior debido a que dentro de la solución, un paso de vital importancia es la adquisición, integración y carga de datos de fuentes heterogéneas y en este sentido Kettle posee una mayor gama de operaciones de transformación, posibilidad de conexión con fuentes diversas y una interfaz de usuario mucho más amigable que Talend. Por estas razones, es Pentaho la herramienta seleccionada para el diseño de la solución de BI.

1.1.7.3. Pentaho, una solución BI de "Open Source"

La Suite de BI Pentaho es diferente a las tradicionales propuestas en esta área. Se define como *"orientada a solución"* y *"centrada en procesos"*. Su base y arquitectura están dirigidas a lograr la integración de herramientas de análisis y reportes con un motor de flujo de trabajo (*workflow*) de procesos de negocio. **[Pentaho, 2006]**

La plataforma incluye los principales componentes requeridos para implantar soluciones basadas en la ejecución de las reglas de negocio, que pueden traducirse en procesos o actividades, además de brindar la información necesaria en el momento adecuado. Pentaho presenta una organización consecuente con las herramientas y tecnologías integradas: Reporting (Reportes), Análisis, Dashboards (Cuadro de Mando) y Administración de Procesos (Figura 5).

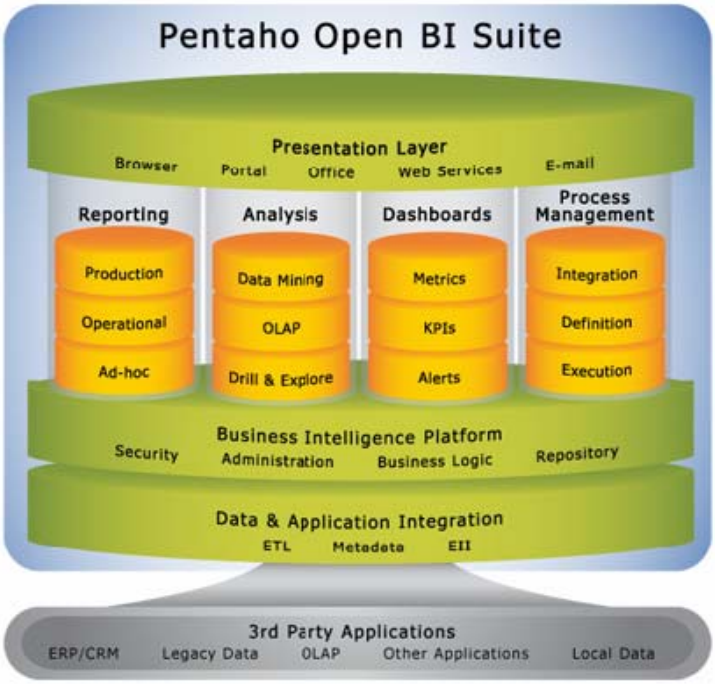


Figura 1. 4 Arquitectura de la Suite Pentaho.[1]

Tecnologías y herramientas integradas

Pentaho generalmente está desplegado sobre un servidor de aplicaciones J2EE JBoss, el que viene incluido en la instalación pre-configurada de la suite. Aunque también puede utilizarse Apache Tomcat, WebSphere, WebLogic y Oracle AS. El empleo de esta tecnología posibilita que la herramienta mantenga toda la información accesible en la intranet de la organización donde se implante. **[Pentaho, 2006]**

Pentaho integra las siguientes herramientas:

1. ***JFreeReport, BIRT o JasperReport*:** presentación de informes en los formatos habituales (HTML, PDF, RTF, MS Excel, etc.).
2. ***Report Design Wizard:*** herramienta de diseño de reportes que facilita el trabajo
3. con JFreeReport.
4. ***Apache FOP*:** generación de PDFs.
5. ***JFreeChart:*** librería para generar los gráficos más comunes (2D, 3D, barras, etc.)

[1] Tomado de **[Pentaho, 2006].**

6. ***Apache Axis:*** motor de Servicios Web. Las acciones de baja complejidad y que forman parte de la solución propuesta pueden ser publicadas en Servicios Web.
7. ***Quartz:*** planificador Open Source integrable con aplicaciones J2EE.
8. ***Enhydra Shark:*** motor de workflow (flujo de trabajo).
9. ***Librería JPivot*:** permite ver tablas OLAP a través de un navegador Web y realizar las tareas típicas de análisis OLAP (drill, down, slice, etc.).
10. ***Mondrian:*** servidor OLAP, sigue una arquitectura ROLAP. Junto a JPivot permite realizar consultas y presentar los resultados mediante un navegador Web.
11. ***WEKA:*** herramienta que implementa algoritmos de minería de datos. Dispone de técnicas como: clasificación, clustering, asociación y visualización.
12. ***Kettle:*** herramienta para realizar procesos ETL.

Conclusiones

En el proceso de captación de la información referente a las áreas de los portadores energéticos y el transporte se obtienen numerosos datos referentes a cada CES. Estos datos constituyen información valiosa para los directivos del MES y es necesario darles el tratamiento adecuado para extraer información útil, que permitan sentar bases para la toma de decisiones estratégicas.

Se define un modelo de análisis como proceso complejo donde deben conjugarse un gran número de métodos y tecnologías. En este sentido el uso de herramientas de Inteligencia de Negocio se consideró como una solución factible para el diseño de la solución, ya que sus principales esfuerzos se encaminan a permitir la explosión de la información contenida en grandes bases de datos para apoyar la toma de decisiones.

Se consideró necesario el diseño y desarrollo de un Datamart para almacenar el histórico de los resultados de la etapa de Diagnóstico del proceso, así como los procesos ETL para adquirir, integrar y cargar los datos provenientes de diferentes fuentes. Posteriormente debe procederse con el análisis y visualización de los resultados.

Para la implementación de la solución se decidió utilizar herramientas pertenecientes a la Comunidad *"Open Source"*, por el carácter de su distribución y su aceptación en el mercado debido a sus prestaciones. Específicamente se seleccionó como SGBD: PosgreSQL y la Suite Pentaho como solución integradora de las herramientas necesarias para implementar la solución.

CAPÍTULO 2. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

Introducción

El Modelo propuesto para el control y análisis de los portadores energéticos y el transporte, constituye una variante de solución que favorece el proceso de toma de decisiones en estas áreas en el MES. El estudio de este proceso, permitirá conocer que información se necesita almacenar, integrar y analizar para apoyar la toma de decisiones estratégicas.

En el presente capítulo se describe el modelo formalizado para el análisis de los portadores energéticos y transporte, y se caracterizan los sistemas existentes. Para finalizar se abordan aspectos referentes a las tendencias y técnicas actuales para la implementación de soluciones basadas en Inteligencia de Negocio.

2.1. Modelo formalizado para el análisis de los portadores energéticos y transporte

El modelo actualmente que se utiliza para la recogida de la información de los portadores energéticos y el transporte, está diseñado en libros de Excel (ver Anexos 2), lo que hace que sea muy difícil la obtención de análisis de dicha información.

La **demanda** recoge la información de esta forma:

- Cada CES de acuerdo a la asignación dada anteriormente por el MES y a sus necesidades, demanda una cantidad de un tipo de combustible para un mes del año, siempre con dos meses de antelación. Teniendo en cuenta la cantidad de combustible por cada actividad.
- Cada CES de acuerdo a los proyectos que tengan, demanda una cantidad de un tipo de combustible para un proyecto (con dos meses de antelación), para un mes del año.
- Cada CES después de haber hecho una evaluación de la asignación dada anteriormente dada por el MES y su actual necesidad, prepara un plan de electricidad (dos meses de antelación).
- Cada CES demanda una cantidad de tipos de portadores, teniendo en cuenta la necesidad que tenga, para un mes del año (con dos mes de antelación)

La **información general del transporte** recoge la información de esta forma:

Cada CES envía al MES el parque de vehículos con que cuenta, así como todos sus indicadores, cada mes.

El **consumo** recoge la información de esta forma:

- En el consumo de combustible, cada CES manda el consumo real de un tipo de combustible, de cada mes del año, por cada actividad.
- Cada CES envía el consumo de Electricidad real de cada mes.
- Cada CES envía el consumo de otros portadores cada mes.

El **plan operativo** recoge la información de esta forma:

Se le asigna a cada CES mensualmente una cantidad total de combustible, teniendo en cuenta la demanda de los centros, para cada moneda.

Se le asigna a cada CES mensualmente un plan de electricidad, teniendo en cuenta la demanda previamente hecha.

Se le asigna a cada CES mensualmente una cantidad de otros portadores, para cada tipo de moneda, tendiendo en cuenta su demanda.

2.2. Sistemas Existentes

Existen algunas herramientas informáticas con el propósito de gestionar eficientemente la información del transporte de los CES. Se han hecho pruebas de las mismas sobre todo en la CUJAE. Cuya finalidad es que todos los CES de país tengan una herramienta informática común, con el objetivo de un mejor control del transporte en el MES.

El Transorg

Actualmente la CUJAE utiliza una herramienta que se usa en la Base de Transporte, el Transorg. Este sistema aportó muchas facilidades al proceso de gestión del transporte pero estuvo sujeto a deficiencias desde del punto de vista de concepción del sistema y sus funciones, como en su implementación.

Partiendo del principio de que estaba destinado a funcionar en las Bases de Transporte de los centros del MES, la población de consumidores se reduce a una pequeña muestra de la totalidad de los activos que hacen uso de los combustibles y sus derivados, por tal

motivo ya Transorg no es funcional para la política de los CES de controlar el consumo total en sus instalaciones.

Por otra parte, se hace necesario destacar las características de las prestaciones que ofrecía dicho sistema. Implementado en Microsoft Acces, Transorg no contaba con niveles de acceso a la información para poder ser manipulada por más de un usuario. Ofrecía un conjunto de funcionalidades cuyo funcionamiento no eran lo más óptimo, era frecuente encontrarse con informes de calidad carentes de detalles y confiabilidad, el usuario estaba a expensas de perderse ante un gran cúmulo de informe a los cuales les faltaban datos imprescindibles para un posterior análisis. Por motivos como estos el sistema fue considerado incompleto e insuficiente.

El MACWIN

MACWIN (Mantenimiento Asistido por Computadora sobre Windows) es una herramienta enfocada a los sistemas GMAC (Gestión del Mantenimiento Asistido por Computadora sobre Windows) que ha contado con gran aceptación a nivel internacional, la cual ha sido implementada en la Base de Transporte de la CUJAE. Está concebida para apoyar la resolución de problemas técnicos y de gestión de mantenimiento.

Como filosofía se introducen los siguientes principios:

- Los activos, equipos e instrumentos no generan costos, sino los trabajos, actividades y proyectos de mantenimiento y reparaciones, siendo los costos la expresión cuantificada de los recursos consumidos.
- Gestionar lo que se hace y no lo que se gasta, que equivale a controlar en primer lugar, los trabajos, actividades y proyectos antes que los recursos.
- El propósito fundamental es ofrecer al operario, técnico y directivo de mantenimiento todo el poder de la información para que pueda actuar con eficacia en al aumento de la disponibilidad y disminución de los costos.
- Parte de la integración y explotación de las tres acciones básicas del mantenimiento (correctivo, preventivo y predictivo) además de integrar la gestión de mantenimiento dentro de la gestión empresarial.

La estructura del Macwin está conformada por el siguiente conjunto de módulos:

Patrimonio: Módulo encargado de la organización inicial del sistema (activos, equipos, instrumentos, su prioridad y sus características técnicas).

Personal: Organiza la información del personal de mantenimiento.

Mantenimiento: Módulo encargado del seguimiento técnico económico de las ordenes de trabajo correctivas, preventivas y predictivas, además de incluir herramientas para el control histórico.

Economía: Dedicado a cumplir con el control de los costos y el presupuesto del mantenimiento. Incluye opciones para la aprobación de la calidad del trabajo, además de realizar cierres contables quincenales y mensuales.

Producción: Brinda la posibilidad de realizar lecturas diarias del tiempo de trabajo de los activos, además garantizar la obtención de información que posibilite el cálculo de indicadores técnicos económicos propios de la gestión de mantenimiento.

Administración y Seguridad: A través de él se gestionan los usuarios autorizados a usar el sistema y los privilegios pertinentes para el acceso a los datos, además de incluir todo la información referente a la licencia del sistema y el archivo de la misma.

Implantaciones más representativas en el territorio nacional:

Empresa Niquelífera “Pedro Soto Alba” Moa Nickel, S.A

Combinado Mecánico “Gustavo Machin”

Refinería de Petróleo “Ñico López”

Hotel Nacional de Cuba

Empresa Niquelífera “Ernesto Che Guevara”

Unidad Básica Portuaria de Moa

Empresa de Cemento “Curazao de Mariel”

Empresa de Perforación y Extracción de Petróleo del Centro

Empresa Aeroportuaria “Cuba catering”

Centro de Ingeniería Genética y Biotecnología

Es un sistema que permite elevar la competitividad del Mantenimiento Integrado en el sector empresarial que cuenta con un número de funcionalidades que no son

explícitamente para una base de transporte, por lo cual carece de muchas (funcionalidades) muy útiles en dicho sector.

Está en fase de terminación una herramienta informática **TransAuto (Auto**matización del **Tran**sporte)", cuyo objetivo es que cada centro, perteneciente al ministerio (MES), tenga el control del uso de sus consumidores energéticos para así evitar gastos innecesarios y optimizar el uso de combustibles; además como parte importante, el control de toda la información general del transporte. Este sistema tributará numerosos datos valiosos al MES con la información que realmente se necesita. O sea este sistema, a diferencia de los otros, se ha hecho en total acuerdo con los directivos del MES.

Será el encargado de enviar al MES en archivos xml la información que se desea de los portadores energéticos y el transporte, de cada CES.

2.3. Tendencias y técnicas actuales

El la actualidad las organizaciones obtienen, crean y almacenan más datos que nunca debido a que a medida que se desarrollan los dispositivos de almacenamiento esta tarea resulta muy poco costosa. Los datos por lo general, provienen de una gran variedad de fuentes: sistemas legados u otras. Manejarlos y convertirlos en información útil para la toma de decisiones se está convirtiendo en una tarea cada vez más complicada.

En este sentido, las técnicas y herramientas de Inteligencia de Negocio (*Business Intelligent, BI*) juegan un papel trascendental, asistiendo a los usuarios en la toma de decisiones. Dentro de BI se incluye: diseño de Data Warehouse y Datamart, generación de reportes, consulta, análisis (OLAP y Minería de Datos), cuadros de mando, etc.

El diseño de Data Warehouse y Datamart incluye la integración de datos en un modelo de diseño apropiado para optimizar la realización de técnicas de análisis que sirvan de soporte para la toma de decisiones. La utilización de diseños multidimensionales y de herramientas de transformación de datos conlleva a que los procesos se desarrollen de forma rápida, eficiente y que los resultados se obtengan con mayor calidad y orientados a las necesidades del usuario final.

Por otra parte, de acuerdo a la característica de centralización de este tipo de arquitectura, se logran mayores niveles de seguridad de la información, donde deben

definirse varios niveles de acceso para los usuarios y un usuario ”administrador” encargado de realizar las tareas de actualización y demás tareas administrativas.

Conclusiones

Después de realizar un estudio de la situación actual en que se ejecuta el proceso de control de los portadores energéticos y el transporte, se identificaron problemas vinculados específicamente con el almacenamiento en el tiempo de las evaluaciones realizadas (mantenimiento de la historia) y su análisis, debido a la forma que actualmente se manipulan los datos.

En estos momentos los directivos del MES no cuentan con una herramienta que automatice adecuadamente el análisis y visualización de los resultados, por lo que se requiere el empleo de técnicas y arquitecturas (de Inteligencia de Negocio) que permitan el modelado y automatización del flujo de trabajo correspondiente.

Técnicamente se hace necesario el diseño de un Data Warehouse o un Datamart, con los respectivos procesos ETL requeridos, la utilización de técnicas de análisis de datos y la posterior visualización de los resultados.

CAPÍTULO 3. MODELO DE SOLUCIÓN PROPUESTO

Introducción

En este capítulo se exponen los pasos necesarios para diseñar una solución, basada en técnicas y arquitecturas de Inteligencia de Negocio, que se ajuste específicamente al modelo propuesto.

Se aplican y documentan las técnicas para la construcción y despliegue de una arquitectura Data Warehouse, con el empleo del método de diseño basado en UML: *Data Warehouse Engineering Process* (DWEP). Esta tarea abarca aspectos concernientes al diseño físico del repositorio de datos, así como sus correspondientes procesos de extracción, transformación y carga. Se describen los pasos a seguir para el análisis y la generación de reportes, además de detallar la implementación del modelo multidimencional diseñado y del esquema de diseño seleccionado.

3.1. Sistema Propuesto

El sistema que se propone se centra en la implementación de una herramienta de apoyo a la toma de decisiones para los directivos involucrados en el análisis de los portadores energéticos y del transporte del MES.

Para el manejo de la información se decidió implementar un Datamart debido a que un Data Warehouse se orienta a la recopilación de la información de toda una organización y su costo de desarrollo en recursos y tiempo es mucho mayor, en cambio, el Datamart es más reducido y menos costoso, debido a que tiene un carácter departamental, o sea, se centra en un área específica y mantiene las mismas características. Como patrón de diseño se decidió utilizar un modelo multidimencional basado en el esquema de diseño copo de nieve.

La población del Datamart se realiza a partir de la información proveniente de dos fuentes: un FTP y correo del MES. Se utiliza el FTP y el correo para almacenar archivos XML provenientes de cada CES, los archivos contienen toda la información importante de los portadores energéticos y el transporte de cada centro. A partir de estas fuentes se realiza la extracción, transformación y carga de los datos mediante los procesos de ETL.

El sistema que se propone posibilita realizar tareas de análisis sobre los datos almacenados. Se propone que se apliquen en una primera etapa del proyecto, técnicas OLAP, y los resultados se

muestren en reportes de varios tipos: tablas, gráficos, etc. Como recomendación, en un futuro aplicar técnicas de minería de datos y de cuadros de mando.

Se propone, utilizar una arquitectura basada en un diseño Web, donde los directivos puedan acceder a la información.

Usuarios y seguridad del proceso

En cuanto a los tipos de usuarios que utilizarán el sistema están:

- Directivos del MES que atienden las áreas del transporte y los portadores energéticos.
- Administrador: programar y configurar nuevas tareas para el sistema (análisis, ETL, reportes, planificación, etc.) y dar mantenimiento.

En cuanto a la seguridad de todo el proceso ETL, para tener una confiabilidad de que los datos que pueblan al Datamart son los que deseamos y no datos "extraños" (que no provengan realmente de algún CES). El sistema de gestión que estará implementado en los CES, generará archivos xml compactados con contraseña, que se la pondrá deliberadamente el director de Transporte de cada CES. Al llegar al MES será descompactado, sabiendo previamente la contraseña de cada CES, luego de aquellos que se logren descompactar se poblará el Datamart. Si no se puede descompactar, nunca esa información poblará el Datamart.

3.2 Diseño del Datamart

El diseño de un DW se estructura en niveles o capas por las que transitan los datos, para ello, es necesario crear tres modelos: conceptual, lógico y físico. A continuación, se describen la creación cada uno de los modelos.

3.2.1 Modelo Conceptual

Para seleccionar la estrategia de diseño del DW se tomaron en consideración aspectos como:

- El diseño debe basarse en el almacenamiento de los XML obtenidos del FTP o correo, incluyendo la mayor cantidad de información posible.
- El conjunto de datos debe poseer un nivel de granularidad que permita extraer información referente a los portadores energéticos y el transporte en sus diferentes niveles, permitiendo el monitoreo de cada nivel (a menor granularidad mayor nivel de detalle en la información).
- Se debe optimizar la realización de consultas sobre los aspectos medidos en el proceso, por separado o en conjunto, evitando enlaces innecesarios.
- El diseño debe evitar la jerarquía de dimensiones.

- Debe ser flexible y debe permitir posibles ampliaciones sin traer grandes complicaciones en el modelo.

Después de realizar el análisis del situación existente y de los requerimientos planteados para confeccionar la nueva propuesta de solución, se decidió seguir la estrategia de diseño: abajo-arriba (bottom-up), según la cual se debe definir el DW en base a los datos disponibles en las fuentes operacionales. Para implementar dicha estrategia se decidió emplear el esquema “copo de nieve” por existir la necesidad de crear jerarquías en las dimensiones y evitar el uso de consultas complejas con uniones innecesarias.

El proceso se divide en cuatro etapas: Demanda, Información General del Transporte, Plan operativo, Consumo. Cada una de estas etapas está compuesta por subniveles que aportan información y constituyen los niveles más bajos de granularidad de la información manipulada. Con el objetivo de mantener estos niveles y permitir el acceso rápido e independiente a cada uno, se diseñó un esquema de copo de nieve para nivel de información.

A continuación, se exponen los niveles de granularidad de los datos almacenados en el Datamart:

- Valoración de la Demanda de Combustible correspondiente a un tipo de combustible, en un centro, para una actividad, en un mes y año determinado.
- Valoración de la Demanda de Electricidad correspondiente a un servicio, en un centro, en un mes y año determinado.
- Valoración de la Demanda de Proyecto correspondiente a un proyecto, en un centro, de un tipo de combustible, en un mes y año determinado.
- Valoración de la Demanda de Otros Portadores correspondiente a un portador, en un centro, en un mes y año determinado.
- Valoración de la cantidad de un tipo de vehículo correspondiente al tipo de vehículo, en un centro.
- Valoración de la cantidad de combustible correspondiente a un tipo de combustible, en un centro.
- Valoración de la cantidad de actividad correspondiente a una actividad, en un centro.
- Valoración de la cantidad de un tipo de consumidor correspondiente a un tipo de consumidor, en un centro.

- Valoración de la cantidad de tipo de chapa correspondiente a un tipo de chapa, en un centro.
- Valoración de la cantidad de indicadores correspondiente a un tipo de indicador, en un centro.
- Valoración del Plan Operativo de Combustible correspondiente a un tipo de combustible, para un centro, un tipo de chapa, en un tipo de moneda, para un mes y año determinado.
- Valoración del Plan Operativo de Electricidad correspondiente a un servicio, para un centro, para un mes y año determinado.
- Valoración del Plan Operativo de Otros Portadores correspondiente a un portador, para un centro, en un tipo de moneda, para un mes y año determinado.
- Valoración del Consumo de Combustible correspondiente a un tipo de combustible, en un centro, en una actividad, en un mes y año determinado.
- Valoración del Consumo de Electricidad correspondiente a un servicio, en un centro, en un mes y año determinado.
- Valoración de Consumo de Otros Portadores correspondiente a un portador, en un centro, en un mes y año determinado.

3.2.1.1 Nivel 1 (Definición Global del Modelo)

En consecuencia con los niveles de granularidad definidos, y las cuatro etapas definidas, se ha establecido la definición global del modelo conceptual. En cada parte se agrupan esquemas copo de nieve de acuerdo a los siguientes criterios:

- Demanda*:* agrupa los esquemas estrellas correspondientes a los subniveles de la demanda (Demanda de combustible, Demanda de electricidad, Demanda de proyecto, Demanda de otros portadores).
- Información General del Transporte: agrupa los esquemas estrellas a los subniveles de la información general del Transporte (Cantidad por tipo de vehículo, Cantidad por combustible, Cantidad por actividad, Cantidad por tipo de chapa, Cantidad por tipo de consumidor y los resultados de los indicadores emitidos por los CES).
- Plan operativo*:* agrupa los esquemas estrellas correspondientes a los subniveles del plan operativo (Plan operativo de combustible, Plan operativo de Electricidad, Plan operativo de otros portadores).

- Consumo: agrupa los esquemas estrellas correspondientes a los subniveles del consumo (Consumo combustible, Consumo Electricidad, Consumo Otros Portadores).

A continuación, se muestra la definición de los esquemas estrellas correspondiente a los subniveles de la Demanda (Tabla 3.1 y Figura 3.1):

Subnivel	Esquema Estrella
Demanda de combustible	StarSchema_ Dem_Combustible
Demanda de electricidad	StarSchema_ Dem_Electricidad
Demanda de proyecto	StarSchema_ Dem_Proyecto
Demanda de otros portadores	StarSchema_ Dem_Portadores

Tabla 3. 1 Esquemas Estrella correspondiente a la Demanda.

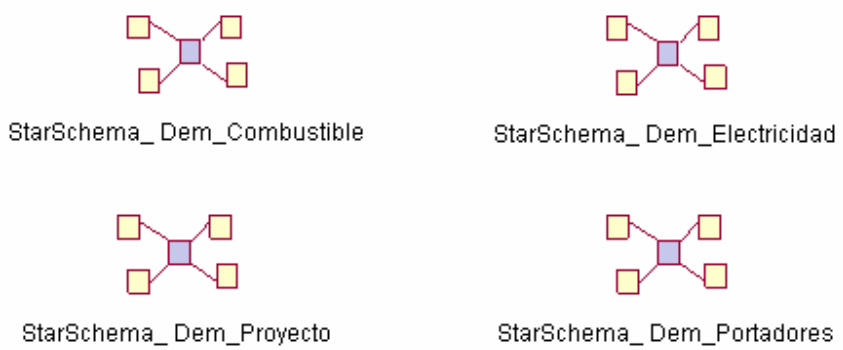


Figura 3. 1 Diagrama de Esquemas Estrellas correspondiente a la Demanda.

A continuación, se muestra la definición de los esquemas estrellas correspondientes a los subniveles de Cantidad por un criterio (Tabla 3.2 y Figura 3.2):

Subnivel	Esquema Estrella
Cantidad por tipo de vehículo	StarSchema_ Cant_Vehículo
Cantidad por combustible	StarSchema_ Cant_Combustible
Cantidad por actividad	StarSchema_ Cant_Actividad
Cantidad por tipo de chapa	StarSchema_ Cant_Chapa
Cantidad por tipo de consumidor	StarSchema_ Cant_Consumidor
Indicadores	StarSchema_Indicadores

Tabla 3. 2 Esquemas Estrellas correspondiente a Información General del Transporte

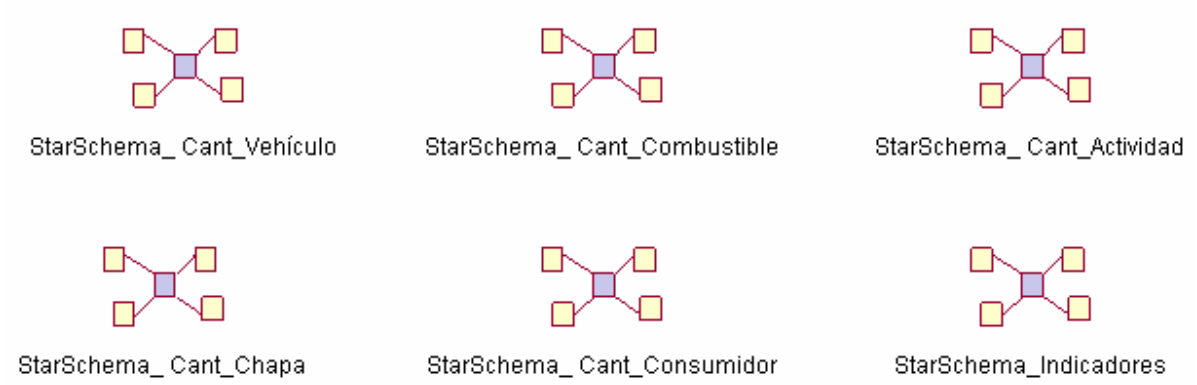


Figura 3. 2 Diagrama de Esquemas Estrellas correspondiente a Información General del Transporte.

A continuación, se muestra la definición de los esquemas estrellas correspondientes a los subniveles del Plan operativo (Tabla 3.3 y Figura 3.3):

Subnivel	Esquema Estrella
Plan operativo de combustible	StarSchema_ Plan_Combustible
Plan operativo de electricidad	StarSchema_ Plan_Electricidad
Plan operativo de otros portadores	StarSchema_ Plan_Portadores

Tabla 3. 3 Esquemas Estrellas correspondiente al Plan operativo.

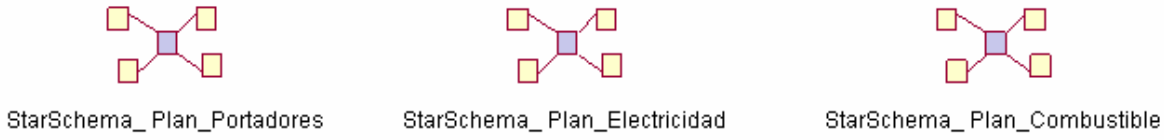


Figura 3. 3 Diagrama de Esquemas Estrellas correspondientes al Plan operativo.

A continuación, se muestra la definición de los esquemas estrellas correspondientes a los subniveles del Consumo (Tabla 3.4 y Figura 3.4):

Subnivel	Esquema Estrella
Consumo combustible	StarSchema_ Cons_Combustible
Consumo electricidad	StarSchema_ Cons_Electricidad
Consumo otros portadores	StarSchema_ Cons_Portadores

Tabla 3. 4 Esquemas Estrellas correspondiente al Consumo.

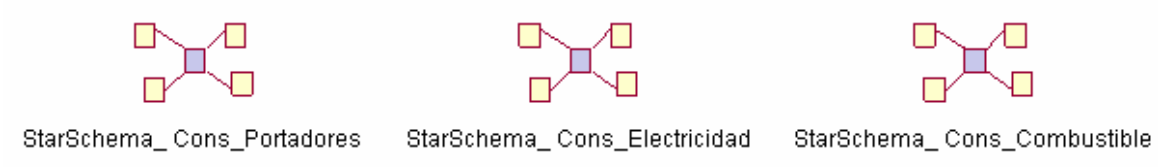


Figura 3. 4 Diagrama de Esquemas Estrellas correspondiente al Consumo.

3.2.1.2 Nivel 2 (Definición de Estructura de Esquemas Estrella)

En este nivel deben definirse las estructuras de los esquemas de estrellas (dimensiones y hechos).

En la siguiente tabla se listan las dimensiones propuestas en la solución (Tabla 3.5):

Nombre	Paquete del Modelo
Combustible	Dim_Combustible
Tipo de Columna para Combustible	Dim_TipoColumnaC
Actividad General	Dim_ActividadG
Actividad	Dim_Actividad
Centro	Dim_Centro
Provincia	Dim_Provincia
Tipo de Centro	Dim_TipoCentro
Servicio	Dim_Servicio
Municipio	Dim_Municipio
Empresa	Dim_Empresa
Tipo de Columna para Electricidad	Dim_TipoColumnaE
Proyecto	Dim_Proyecto
Tipo de Indicadores	Dim_TipoIndicadores
Tipo de Vehículo	Dim_TipoVehículo
Tipo de Consumidor	Dim_TipoConsumidor
Otros Portadores	Dim_OtrosPortadores
Tipo de Chapa	Dim_TipoChapa
Moneda	Dim_Moneda
Fecha del Centro	Dim_DateCentro
Tiempo	Dim_Date

Tabla 3. 5 Dimensiones propuestas en la solución.

La dimensión tiempo es fundamental en todo diseño de un DW debido a que da cumplimiento a uno de sus principales objetivos, la conservación del histórico de la información. Para el diseño de esta propuesta se creó una jerarquía en esta dimensión para permitir resumir datos por: Días, Meses y Años.

En la siguiente tabla se listan los hechos propuestos en la solución (Tabla 3.6):

Nombre	Paquete del Modelo
Demanda de Combustible	Fact_DemandaCombustible
Demanda de Electricidad	Fact_DemandaElectricidad
Demanda de Proyecto	Fact_DemandaProyecto
Demanda de Otros Portadores	Fact _DemandaOtrosPortadores
Cantidad por Tipo de Vehículo	Fact _CantidadTipoVehículo
Cantidad por Combustible	Fact _CantidadCombustible
Cantidad por Actividad	Fact _CantidadActividad
Cantidad por Tipo Consumidor	Fact _CantidadTipoConsumidor
Cantidad por Tipo Chapa	Fact _CantidadTipoChapa
Indicadores	Fact _Indicadores
Plan Operativo de Combustible	Fact _PlanOperativoCombustible
Plan Operativo de Electricidad	Fact _PlanOperativoElectricidad
Plan Operativo de Otros Portadores	Fact _PlanOperativoOtrosPortadores
Consumo de Combustible	Fact _ConsumoCombustible
Consumo de Electricidad	Fact _ConsumoElectricidad
Consumo Otros Portadores	Fact _ConsumoOtrosPortadores

Tabla 3. 6 Hechos propuestos en la solución.

A continuación, se muestra la definición de las estructuras de los esquemas estrella (Figuras 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20):

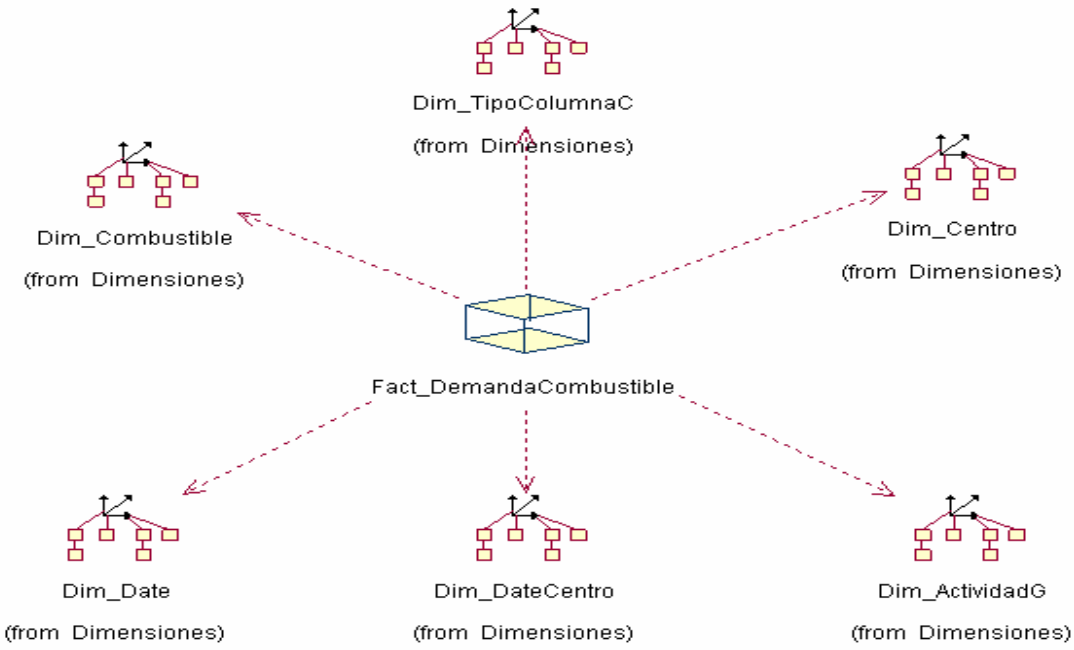


Figura 3. 5 Estructura del Esquema Estrella: StarSchema_ Dem_Combustible

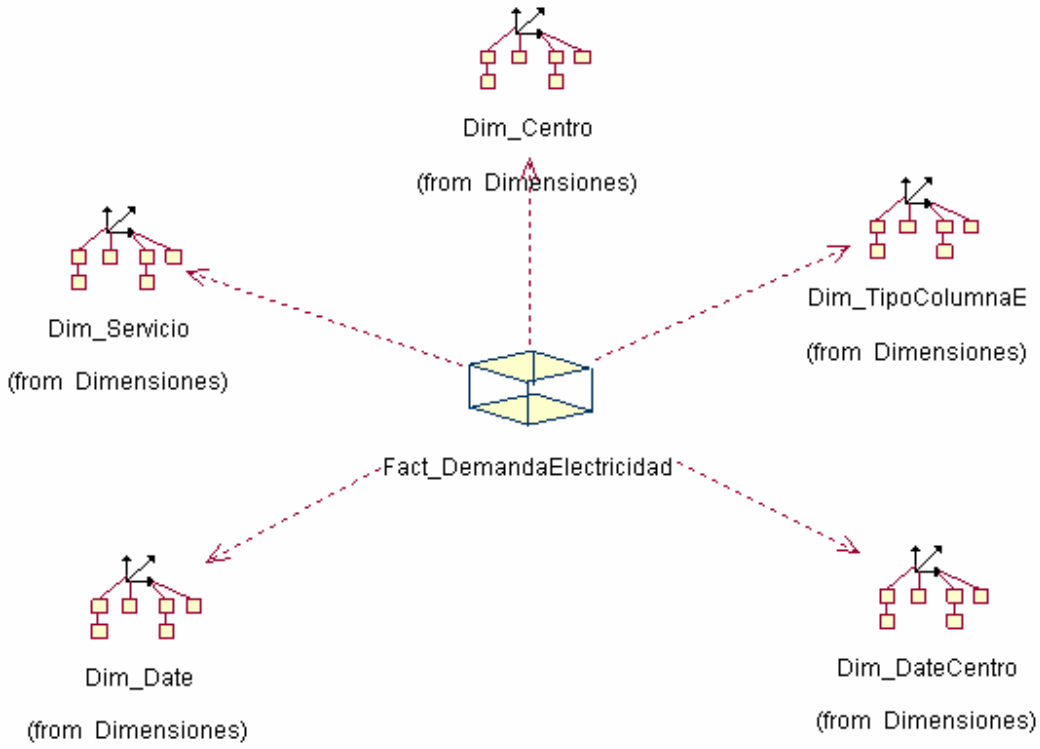


Figura 3. 6 Estructura del Esquema Estrella: StarSchema_ Dem_Electricidad

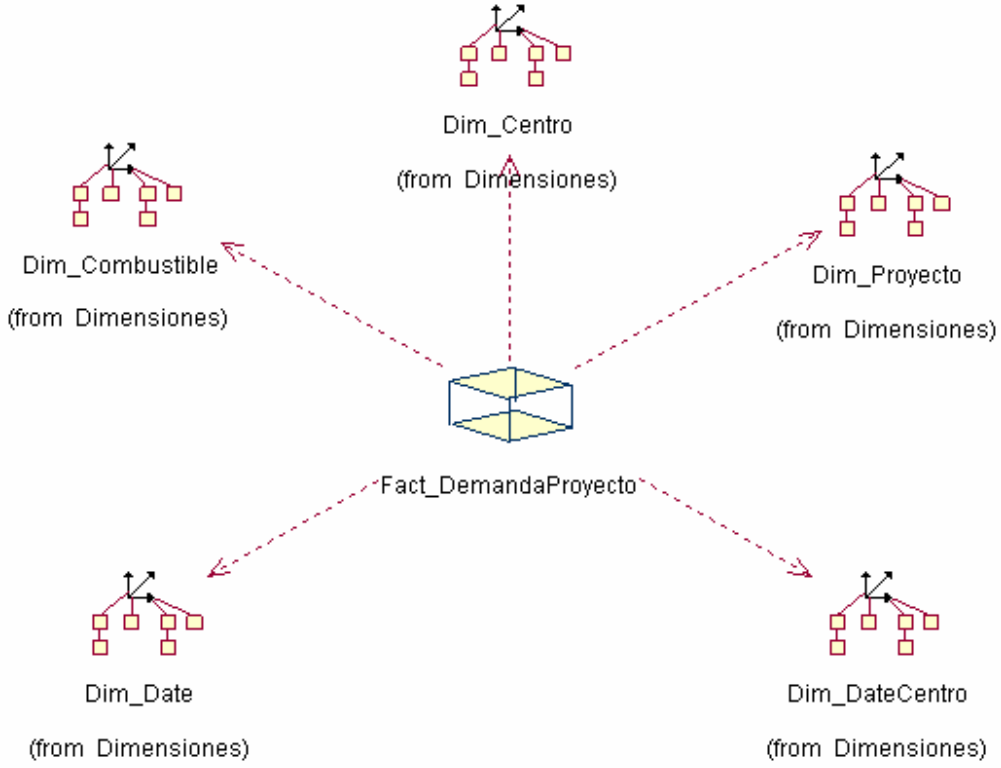


Figura 3. 7 Estructura del Esquema Estrella: StarSchema_ Dem_Proyecto

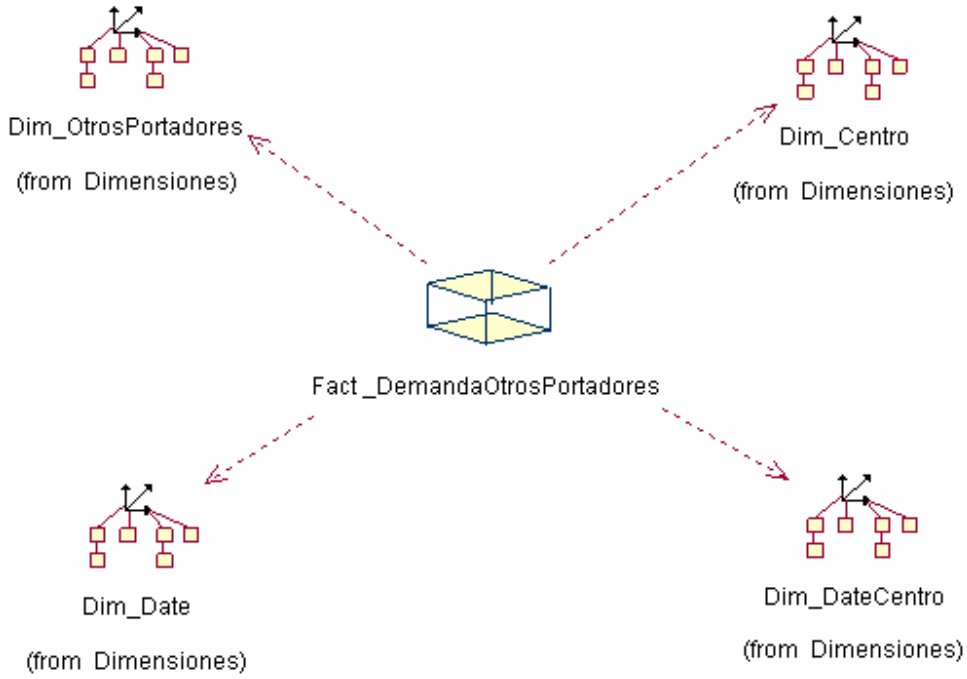


Figura 3. 8 Estructura del Esquema Estrella: StarSchema_ Dem_Portadores

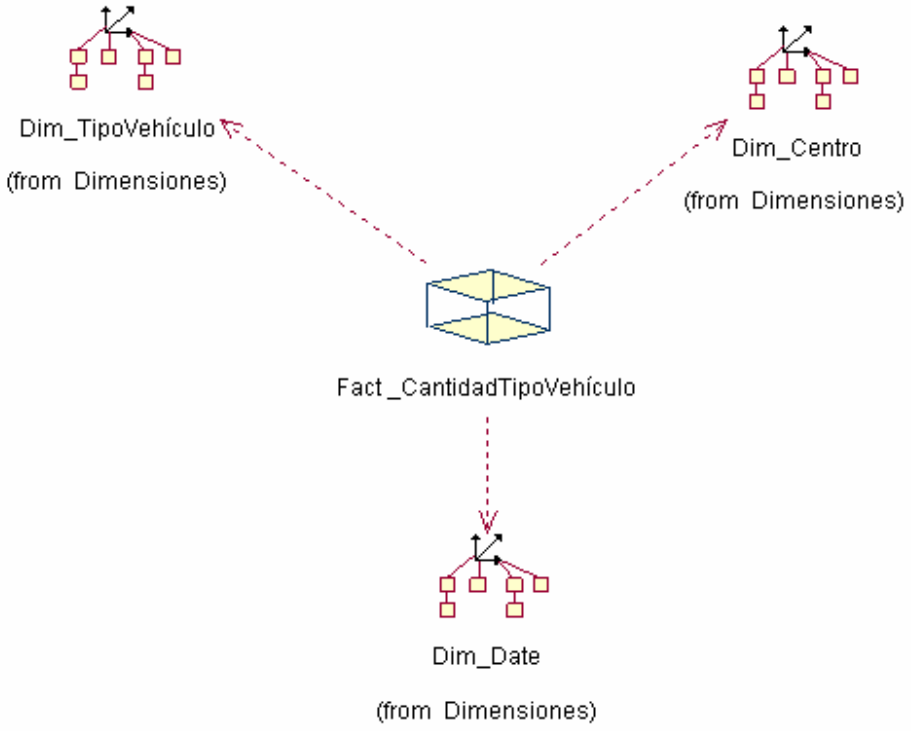


Figura 3. 9 Estructura del Esquema Estrella: StarSchema_ Cant_Vehículo

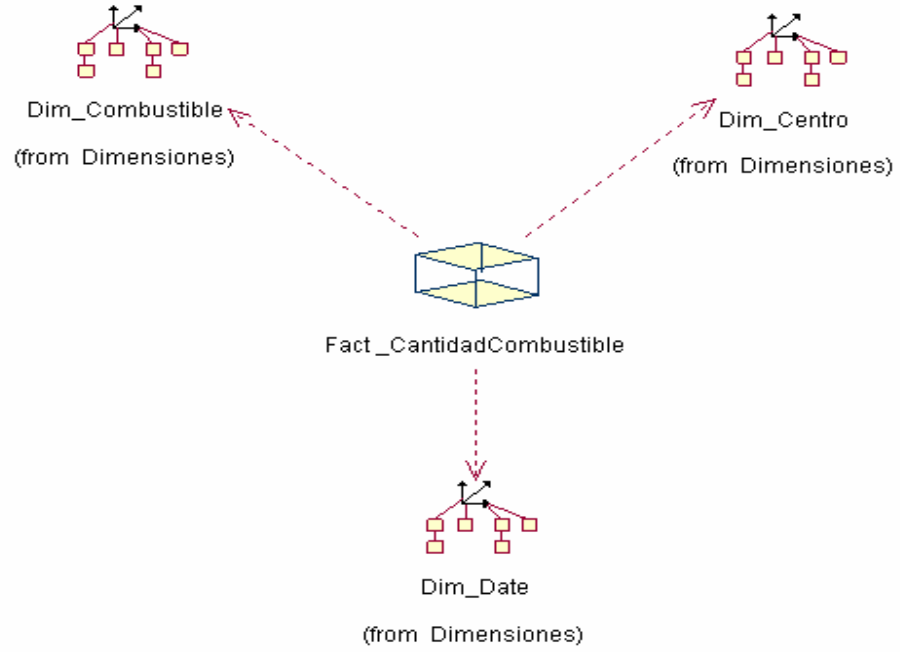


Figura 3. 10 Estructura del Esquema Estrella: StarSchema_ Cant_Combustible

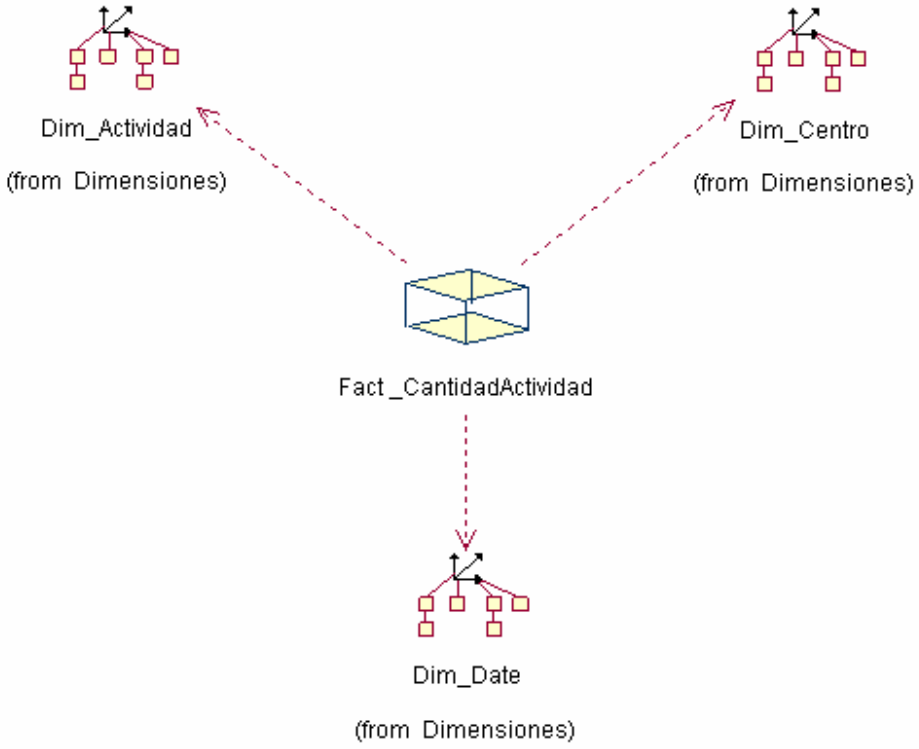


Figura 3. 11 Estructura del Esquema Estrella: StarSchema_ Cant_Actividad

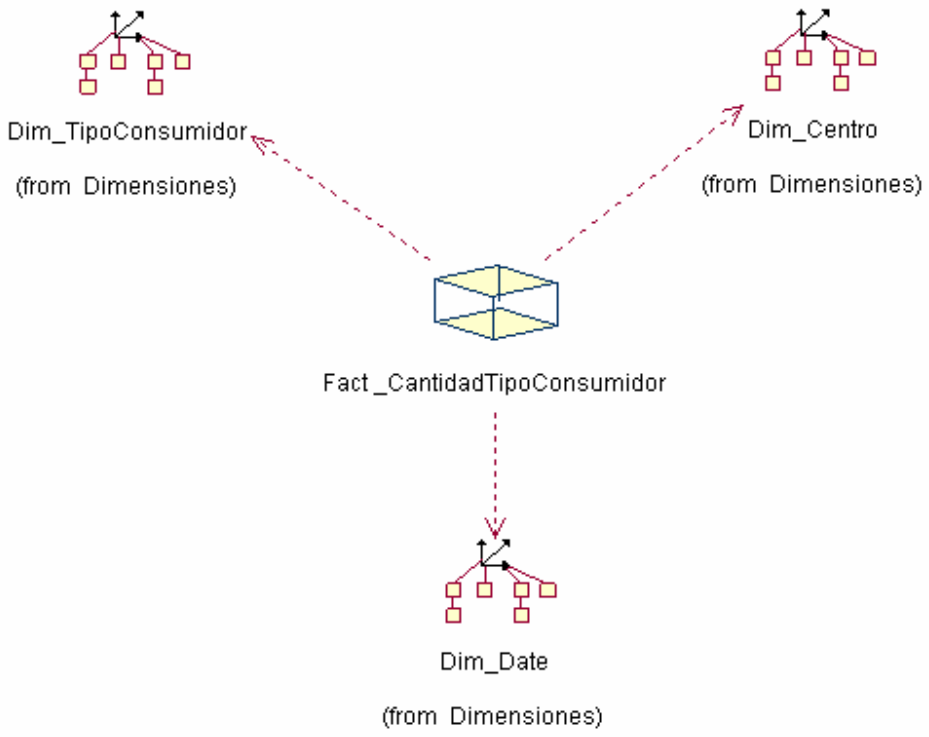


Figura 3. 12 Estructura del Esquema Estrella: StarSchema_ Cant_Consumidor

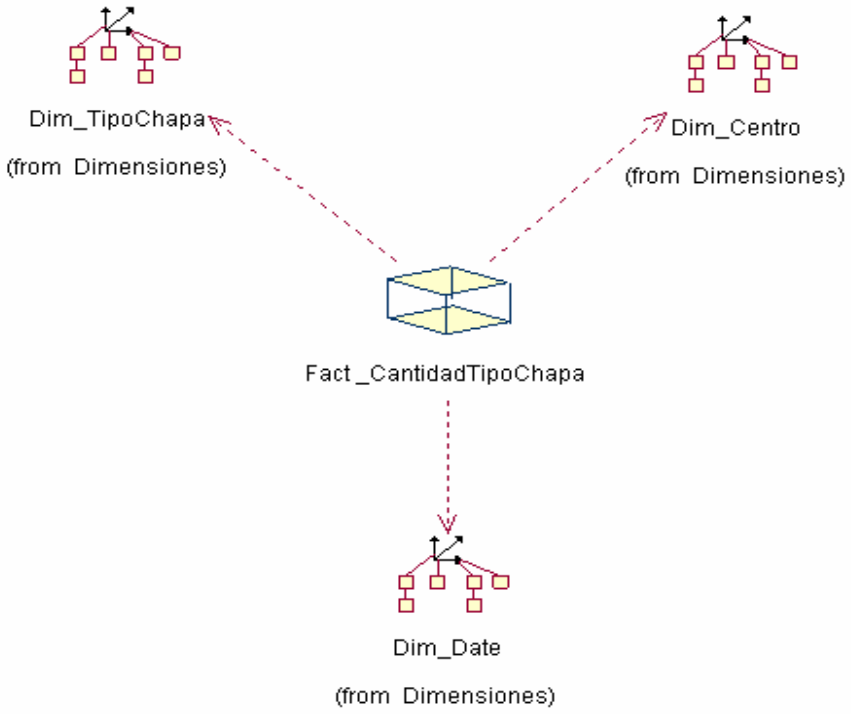


Figura 3. 13 Estructura del Esquema Estrella: StarSchema_ Cant_Chapa

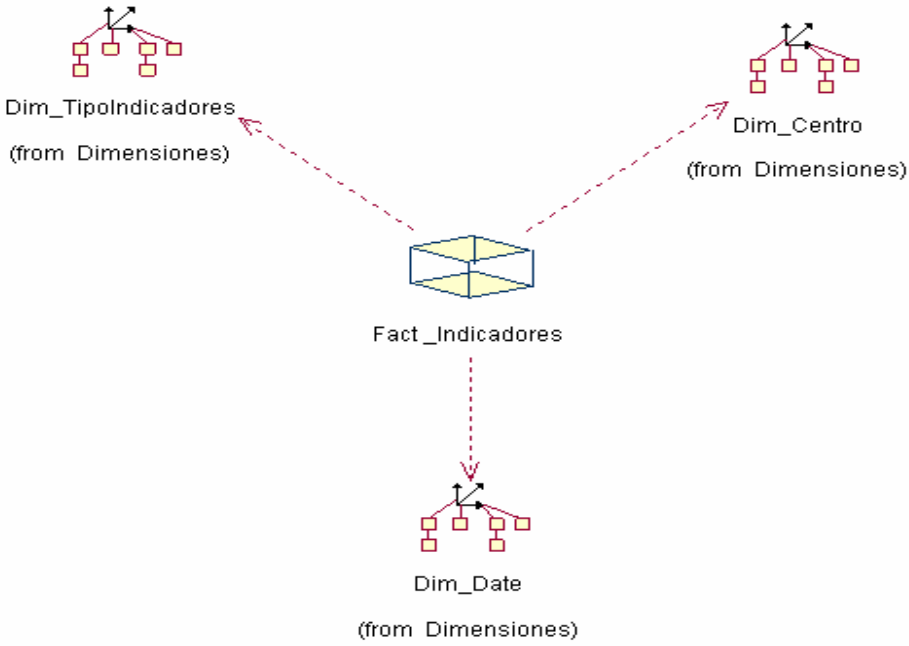


Figura 3. 14 Estructura del Esquema Estrella: StarSchema_Indicadores

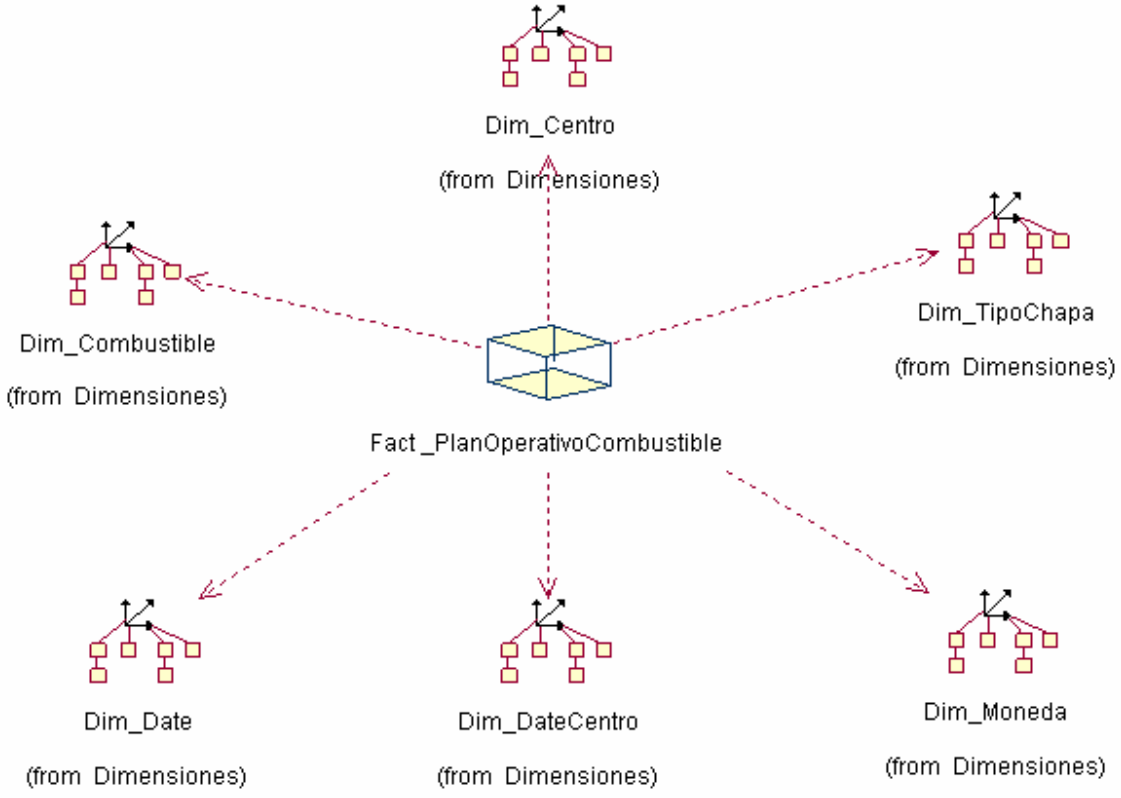


Figura 3. 15 Estructura del Esquema Estrella: StarSchema_Plan_Combustible

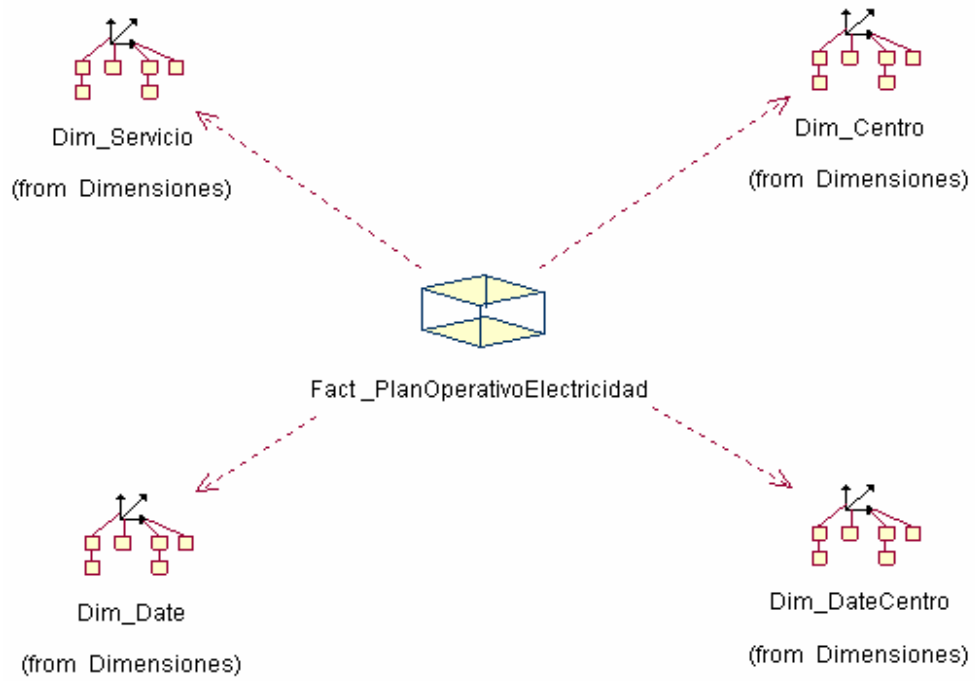


Figura 3. 16 Estructura del Esquema Estrella: StarSchema_ Plan_Electricidad

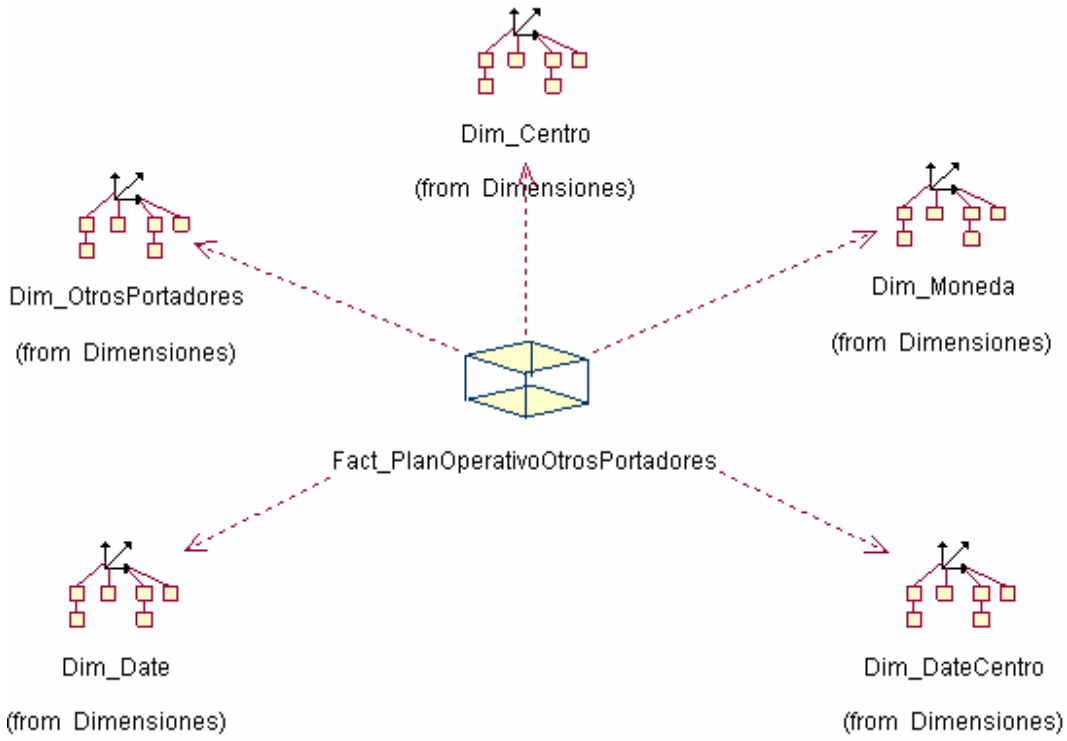


Figura 3. 17 Estructura del Esquema Estrella: StarSchema_ Plan_Portadores

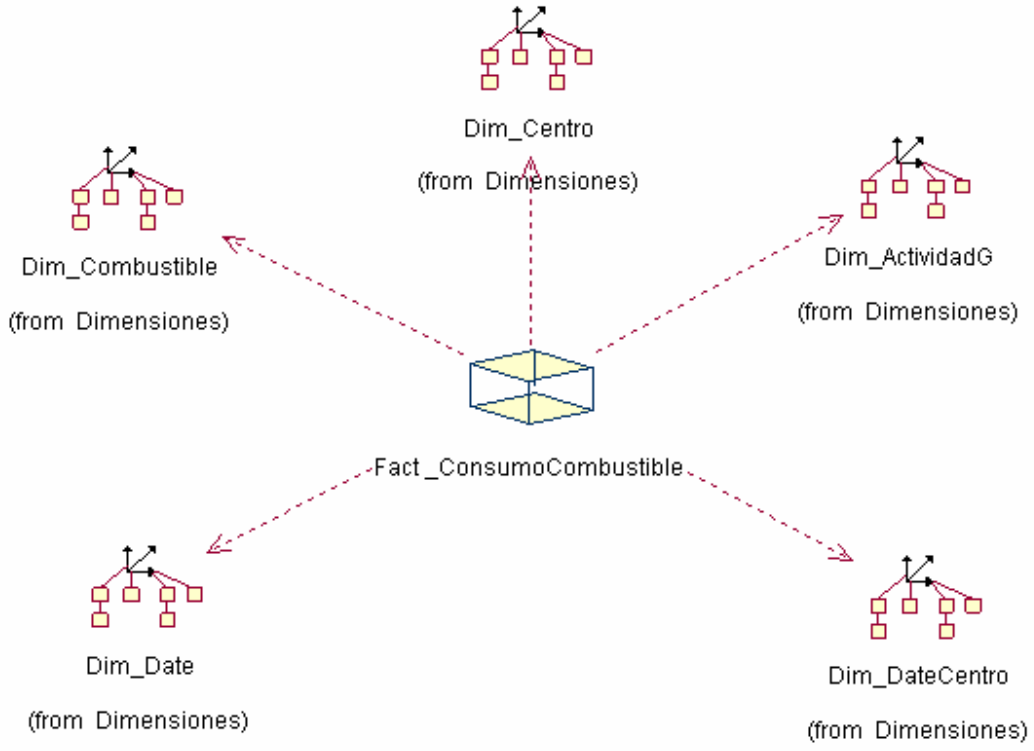


Figura 3. 18 Estructura del Esquema Estrella: StarSchema_ Cons_Combustible

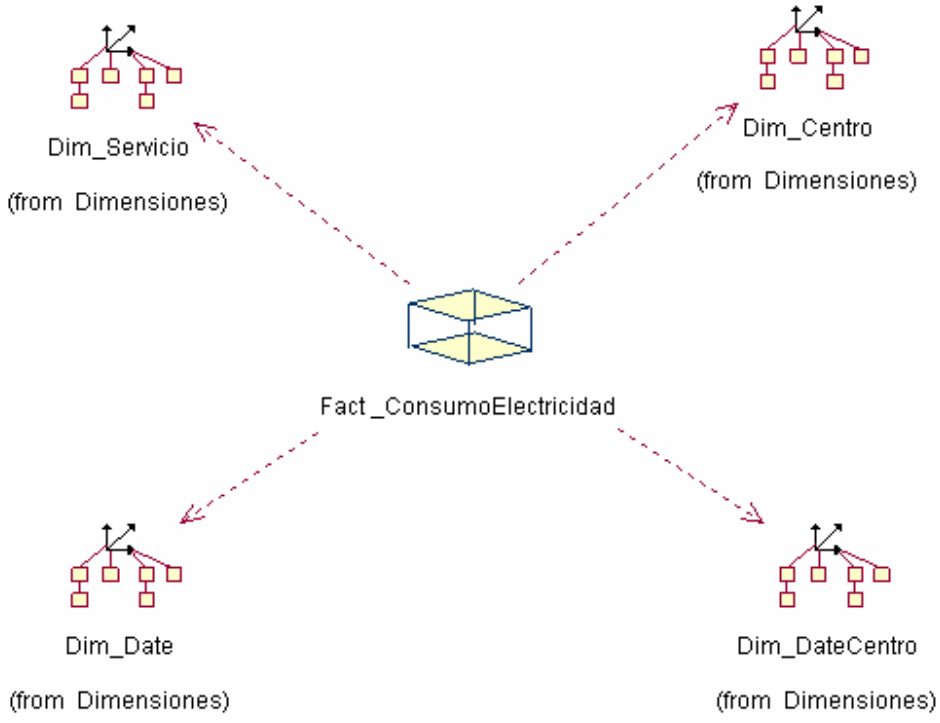


Figura 3. 19 Estructura del Esquema Estrella: StarSchema_ Cons_Electricidad

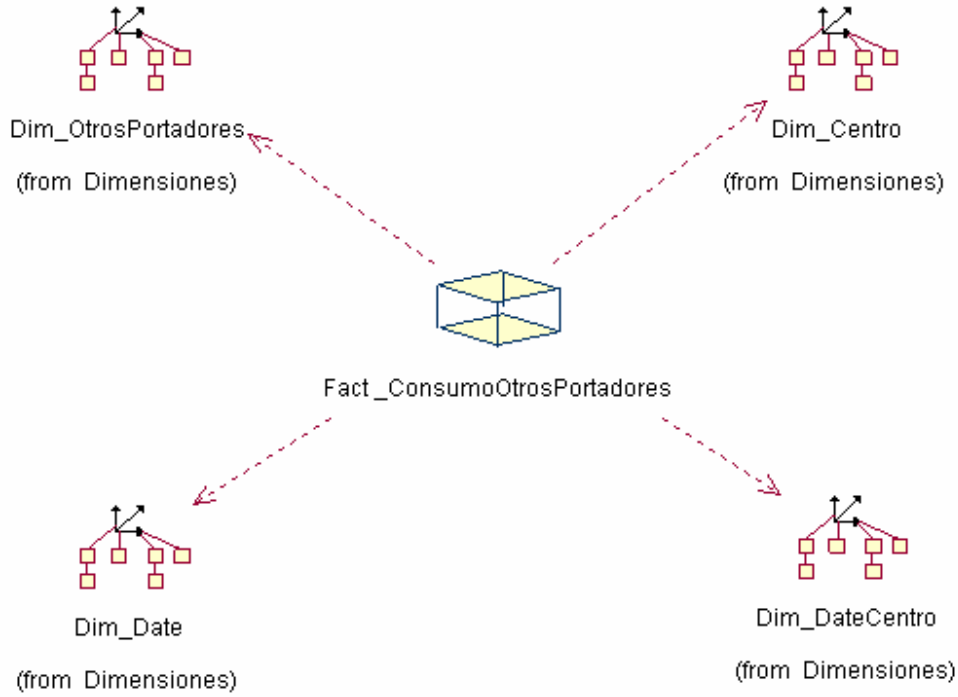


Figura 3. 20 Estructura del Esquema Estrella: StarSchema_ Cons_Portadores

3.2.1.3 Nivel 3 (Definición de Estructura de Hechos y Dimensiones)

En este nivel deben definirse las propiedades estructurales de las dimensiones y hechos, especificando los atributos y relaciones.

A continuación, se muestra la definición de las estructuras de las dimensiones (Figuras 3.21, 3.22, 3.23, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.31, 3.32, 3.33, 3.34, 3.35, 3.36):

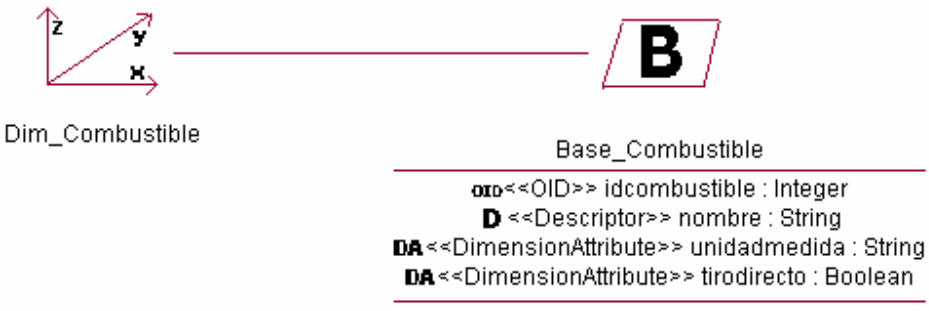


Figura 3. 21 Estructura de la Dimensión: Dim_Combustible.

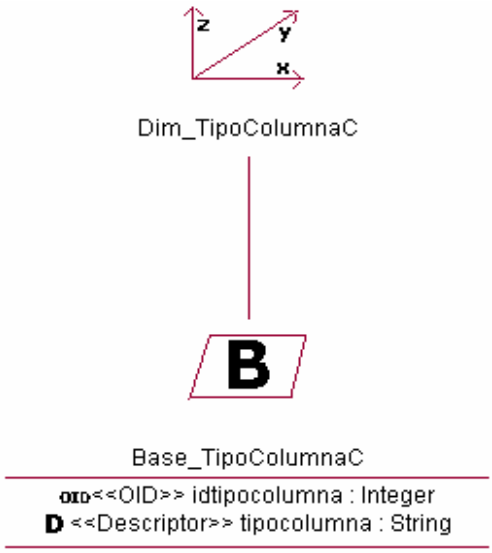


Figura 3. 22 Estructura de la Dimensión: Dim_TipoColumnaC.

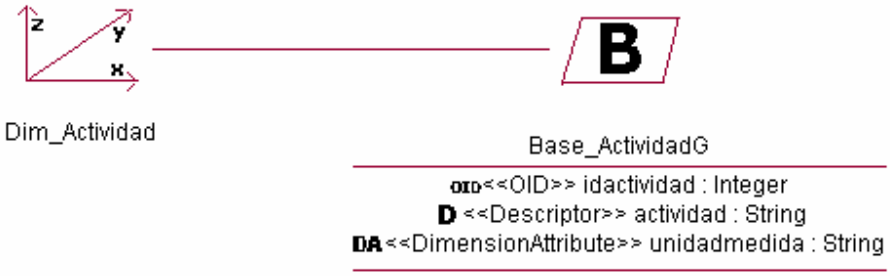


Figura 3. 23 Estructura de la Dimensión: Dim_ActividadG.

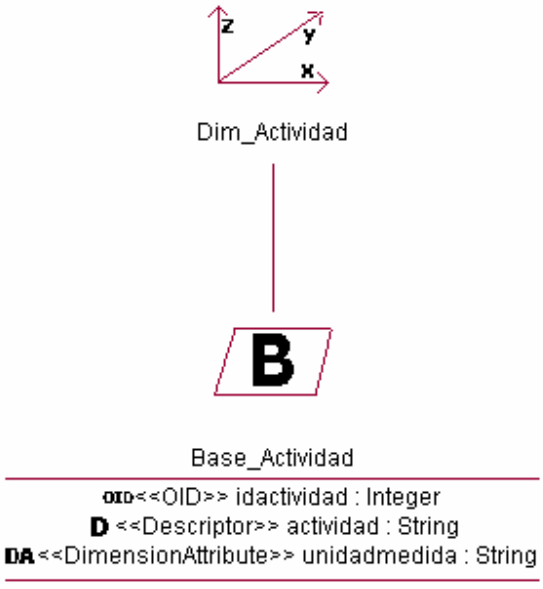


Figura 3. 24 Estructura de la Dimensión: Dim_Actividad.

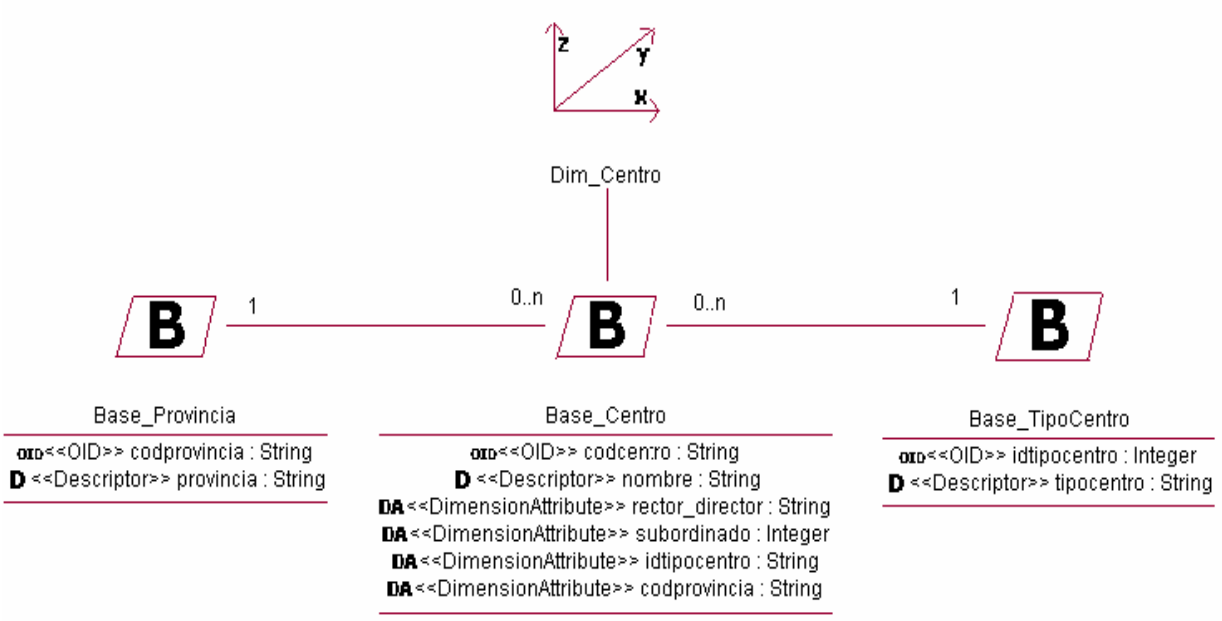


Figura 3. 25 Estructura de la Dimensión: Dim_Centro.

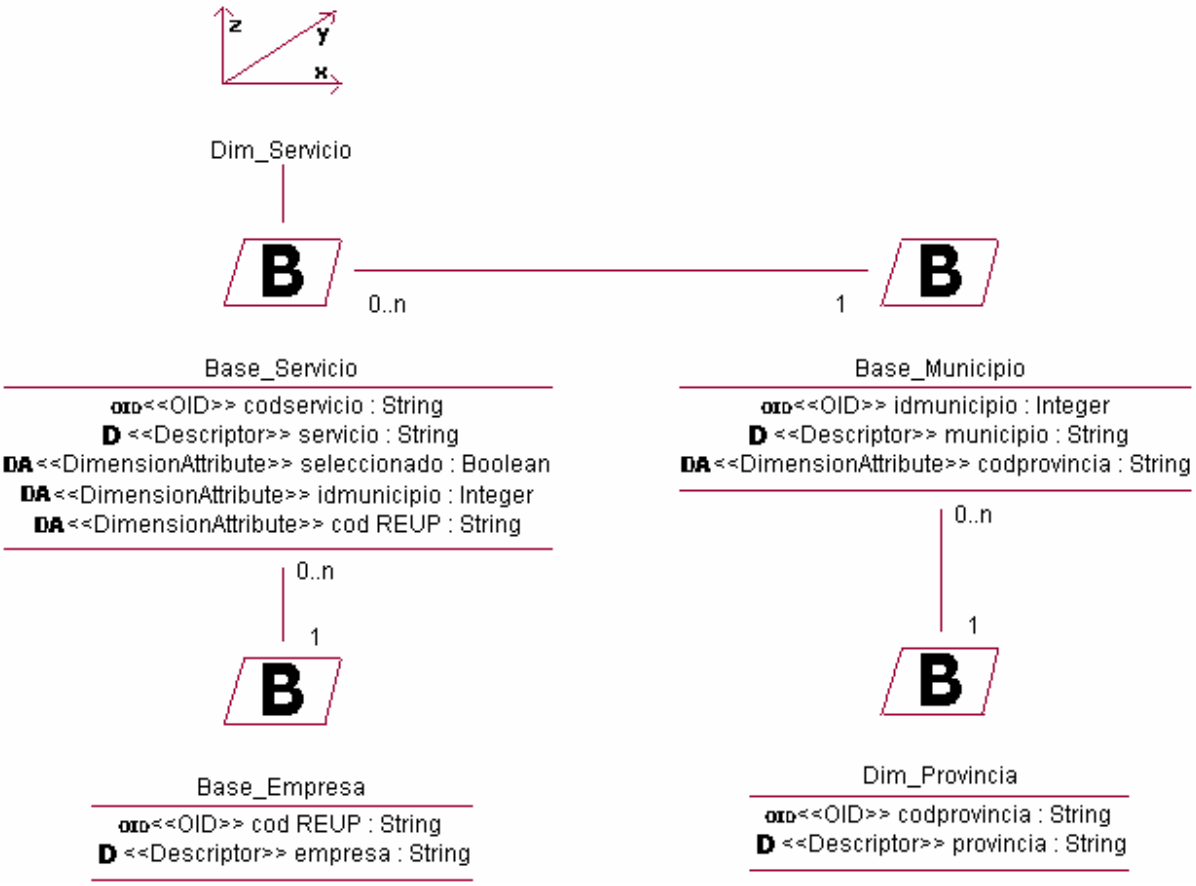


Figura 3. 26 Estructura de la Dimensión: Dim_Servicio.

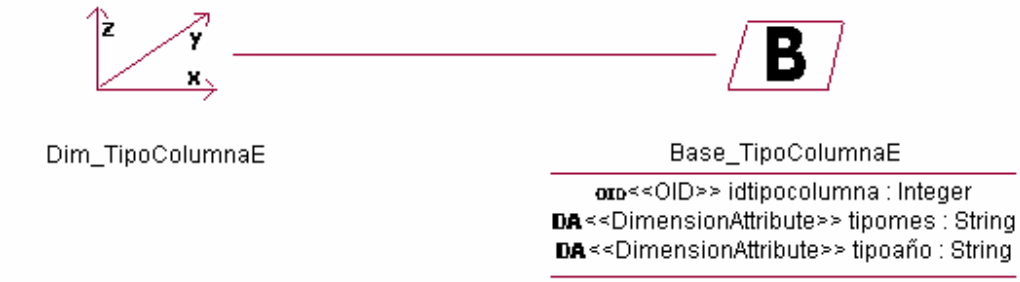


Figura 3. 27 Estructura de la Dimensión: Dim_TipoColumnaE.

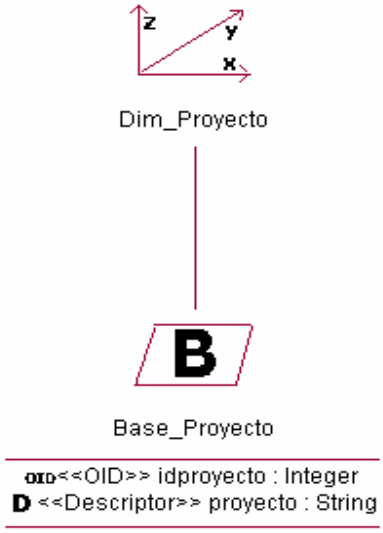


Figura 3. 28 Estructura de la Dimensión: Dim_Proyecto.

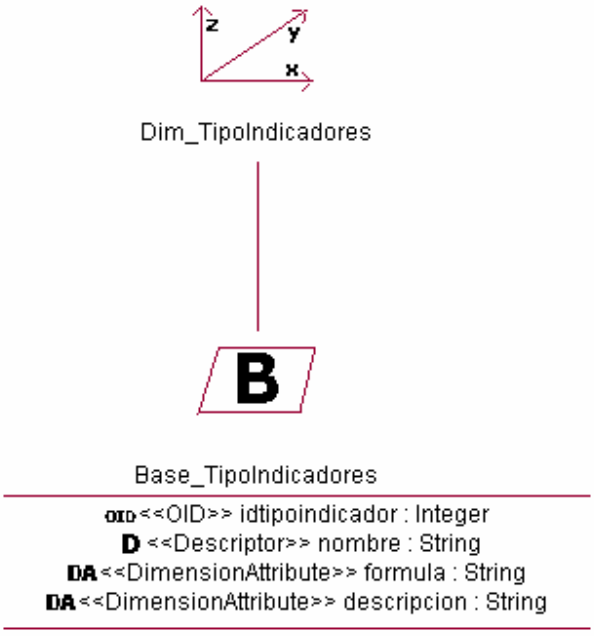


Figura 3. 29 Estructura de la Dimensión: Dim_TipoIndicadores.

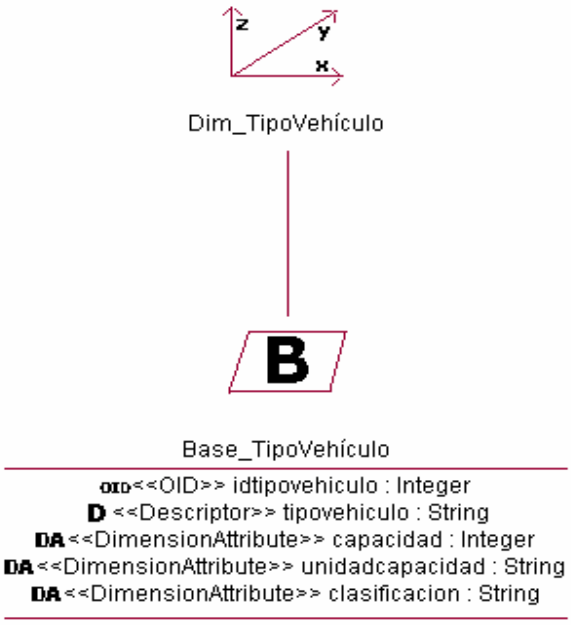


Figura 3. 30 Estructura de la Dimensión: Dim_TipoVehículo.

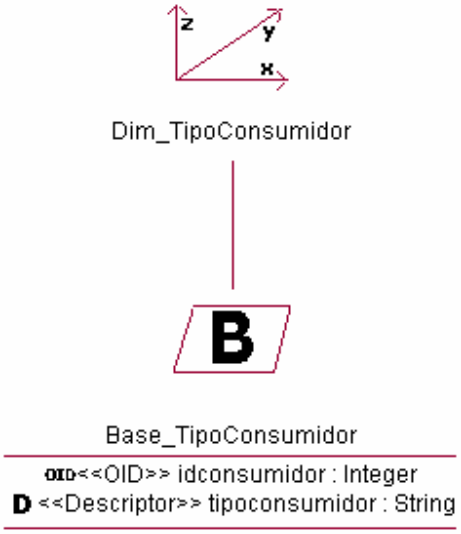


Figura 3. 31 Estructura de la Dimensión: Dim_TipoConsumidor.

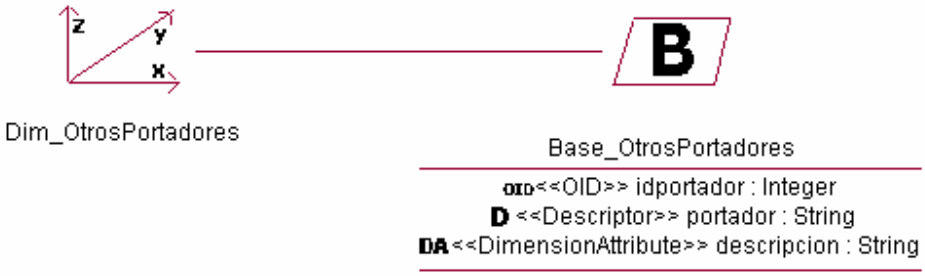


Figura 3. 32 Estructura de la Dimensión: Dim_OtrosPortadores.

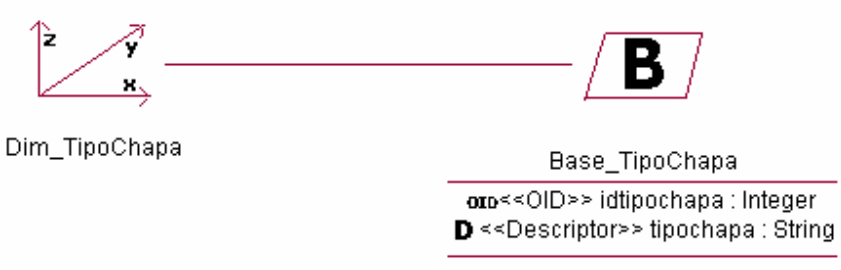


Figura 3. 33 Estructura de la Dimensión: Dim_TipoChapa.

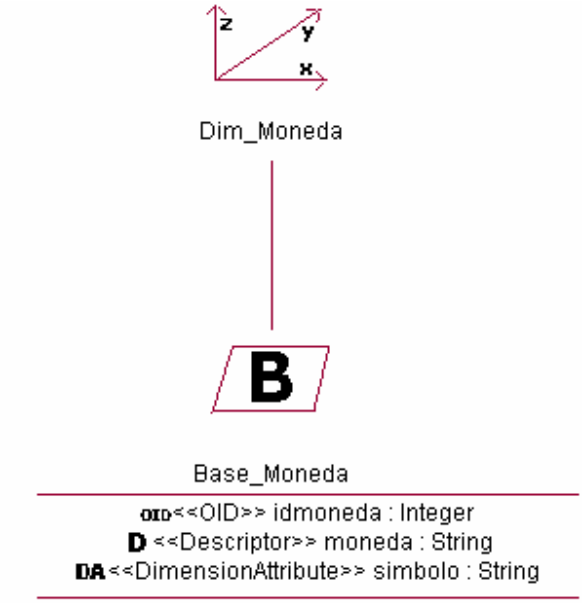


Figura 3. 34 Estructura de la Dimensión: Dim_Moneda.

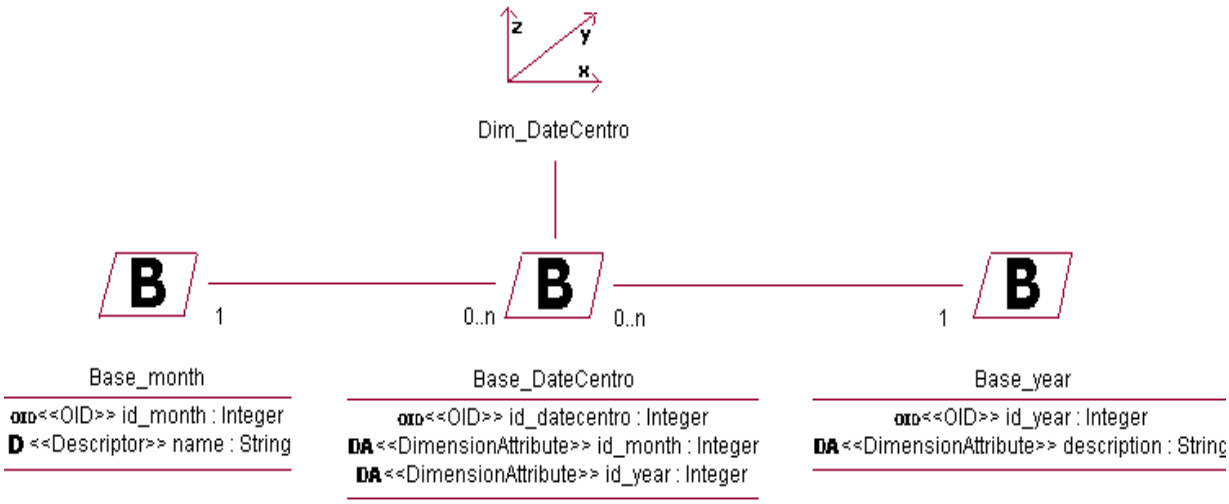


Figura 3. 35 Estructura de la Dimensión: Dim_DateCentro.

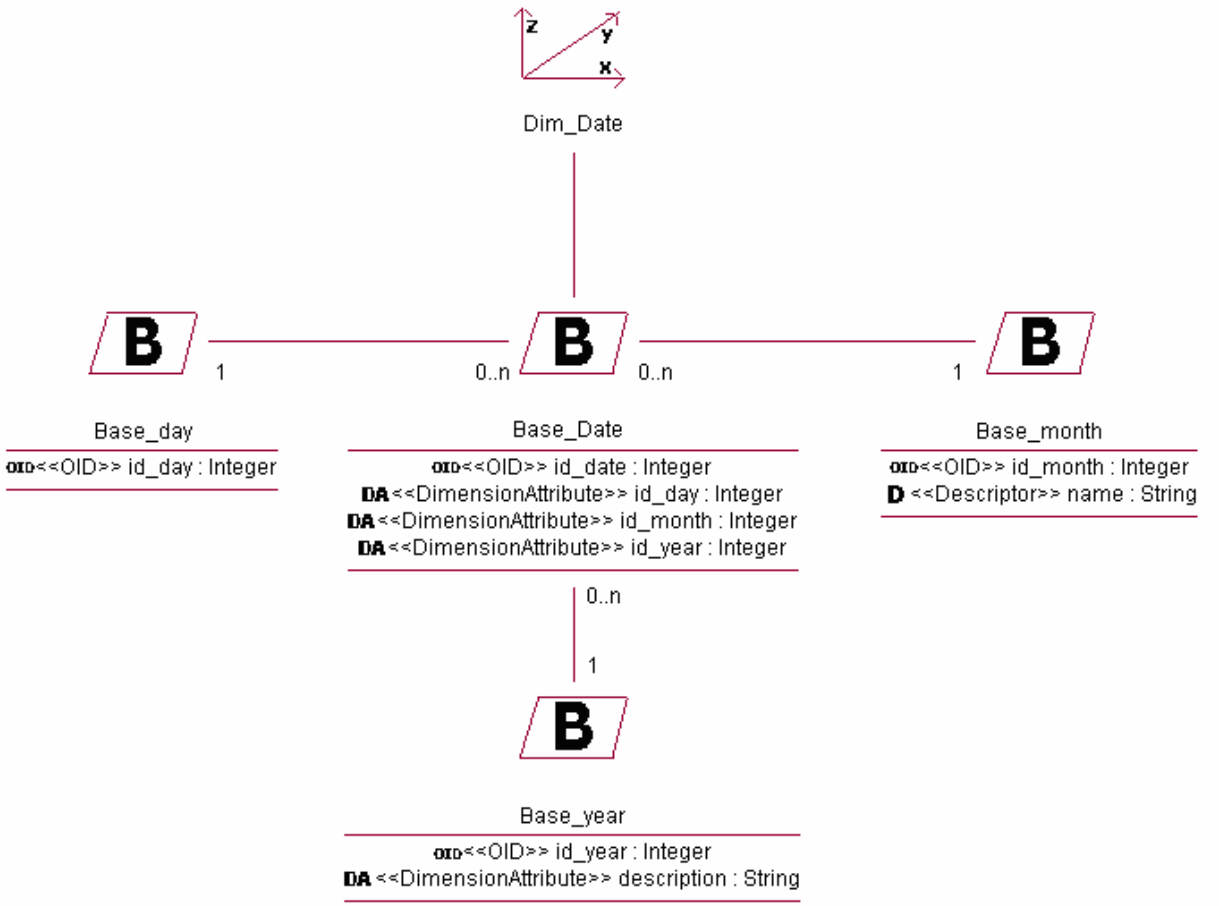


Figura 3. 36 Estructura de la Dimensión: Dim_Date.

A continuación, se muestra la definición de las estructuras de los hechos (Figuras 3.37, 3.38, 3.39, 3.40, 3.41, 3.42, 3.43, 3.44, 3.45, 3.46, 3.47, 3.48, 3.49, 3.50, 3.51, 3.52):

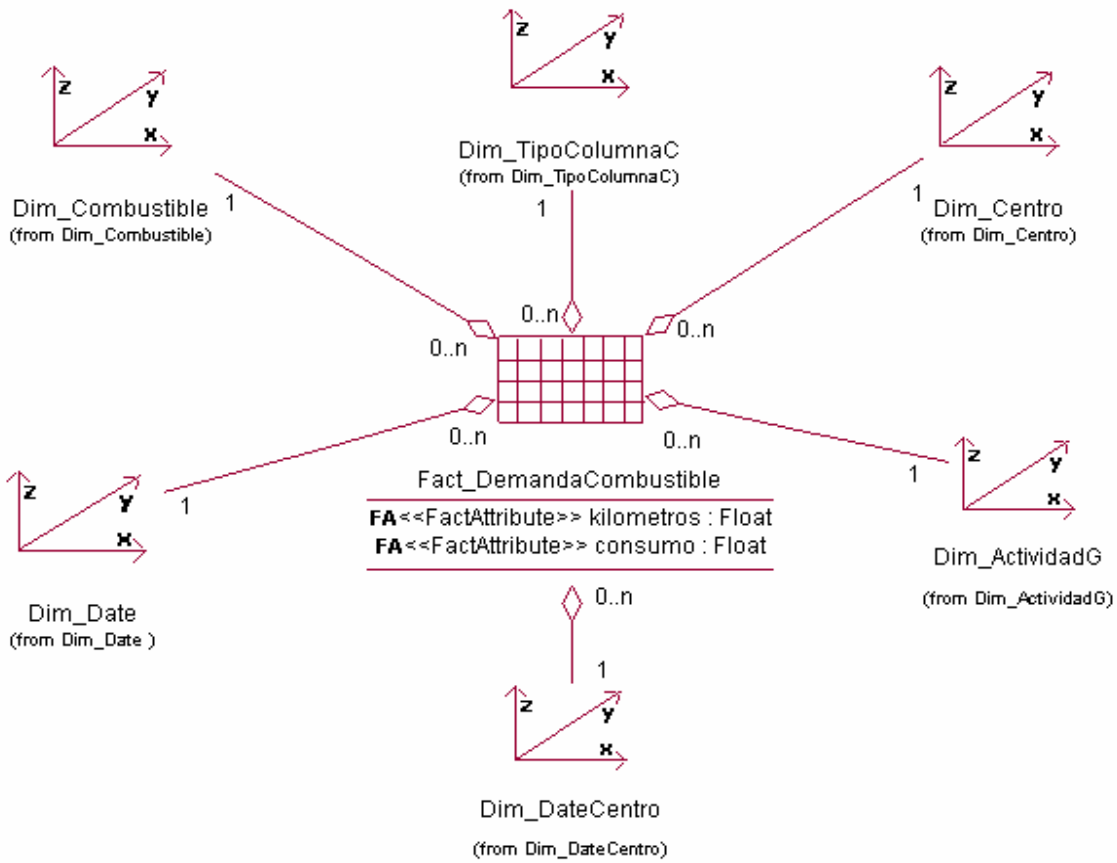


Figura 3. 37 Estructura del Hecho: Fact_DemandaCombustible.

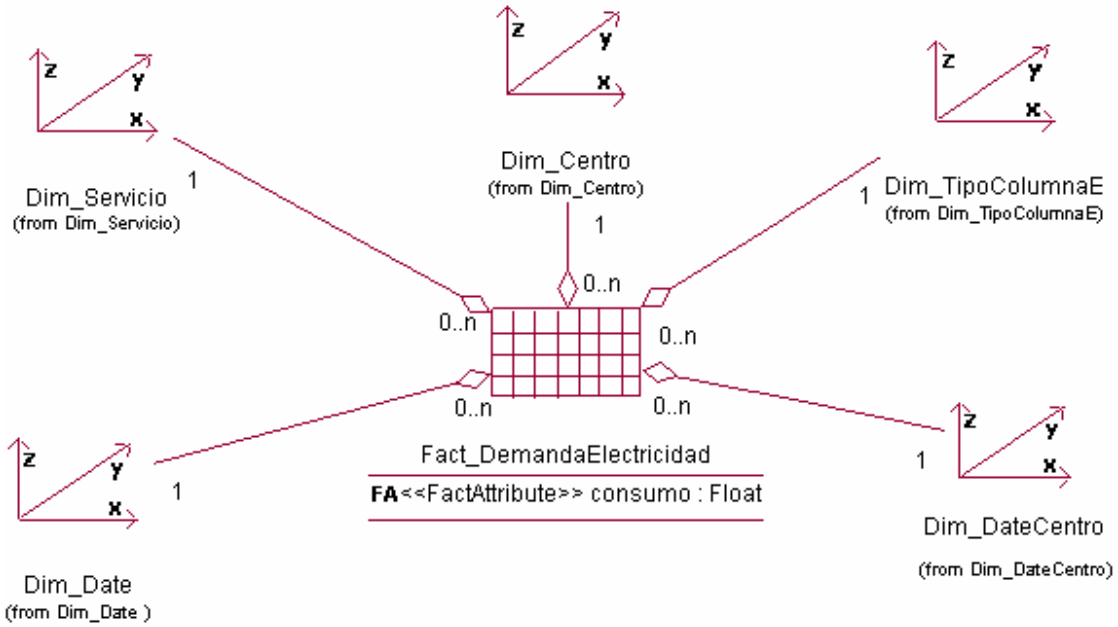


Figura 3. 38 Estructura del Hecho: Fact_DemandaElectricidad.

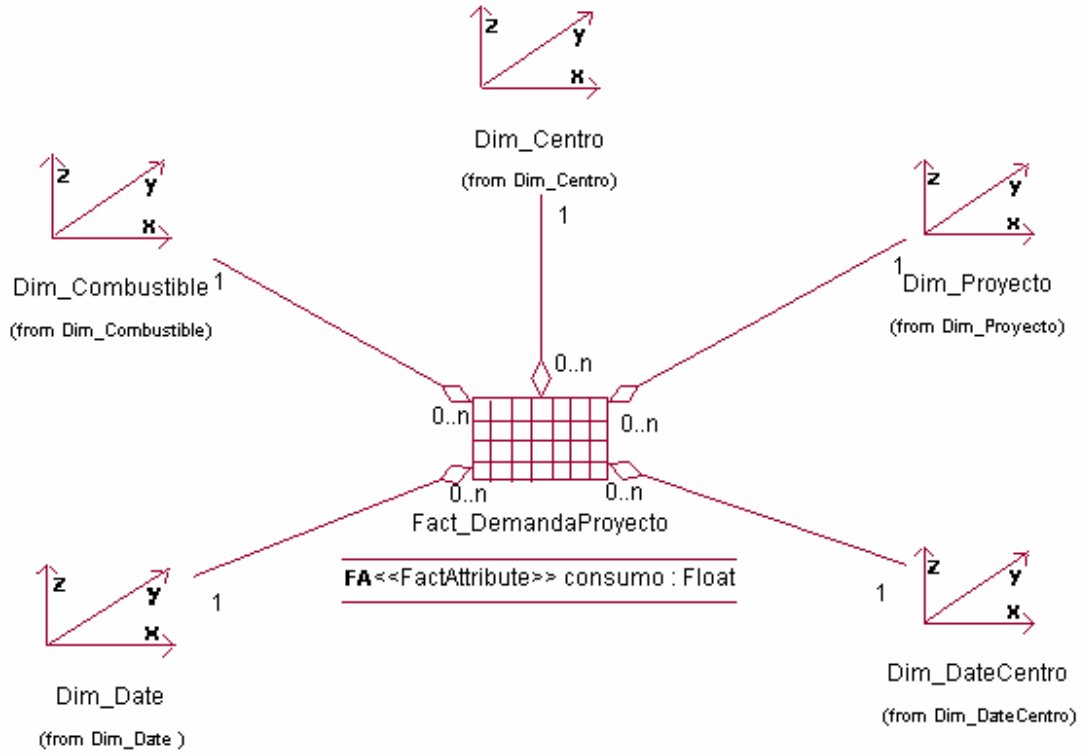


Figura 3. 39 Estructura del Hecho: Fact_DemandaProyecto.

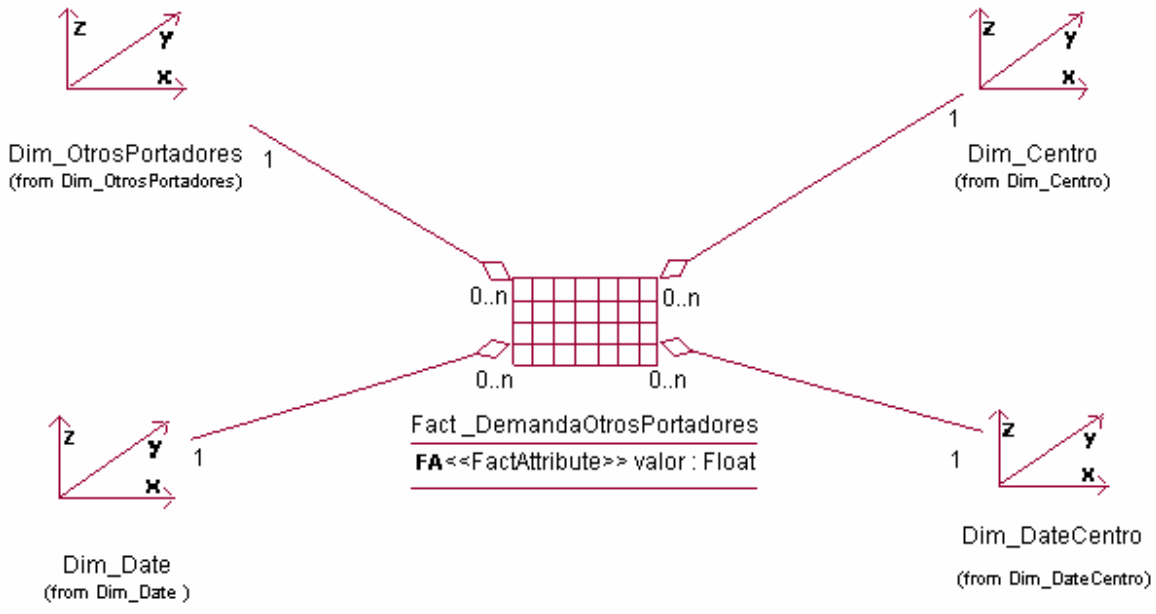


Figura 3. 40 Estructura del Hecho: Fact _DemandaOtrosPortadores.

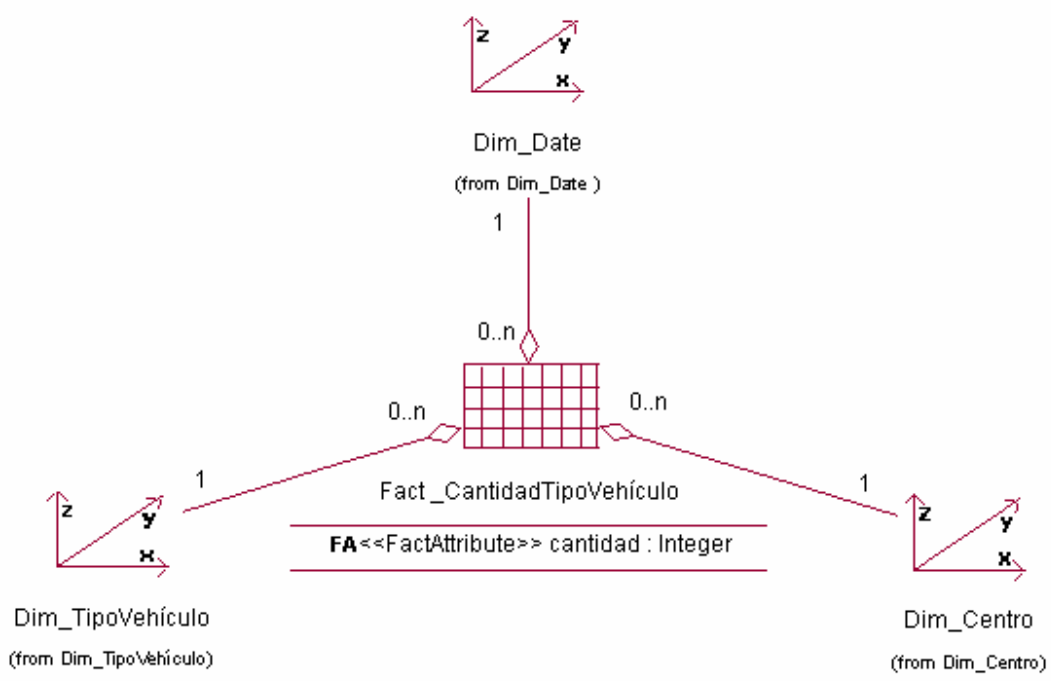


Figura 3. 41 Estructura del Hecho: Fact _CantidadTipoVehículo.

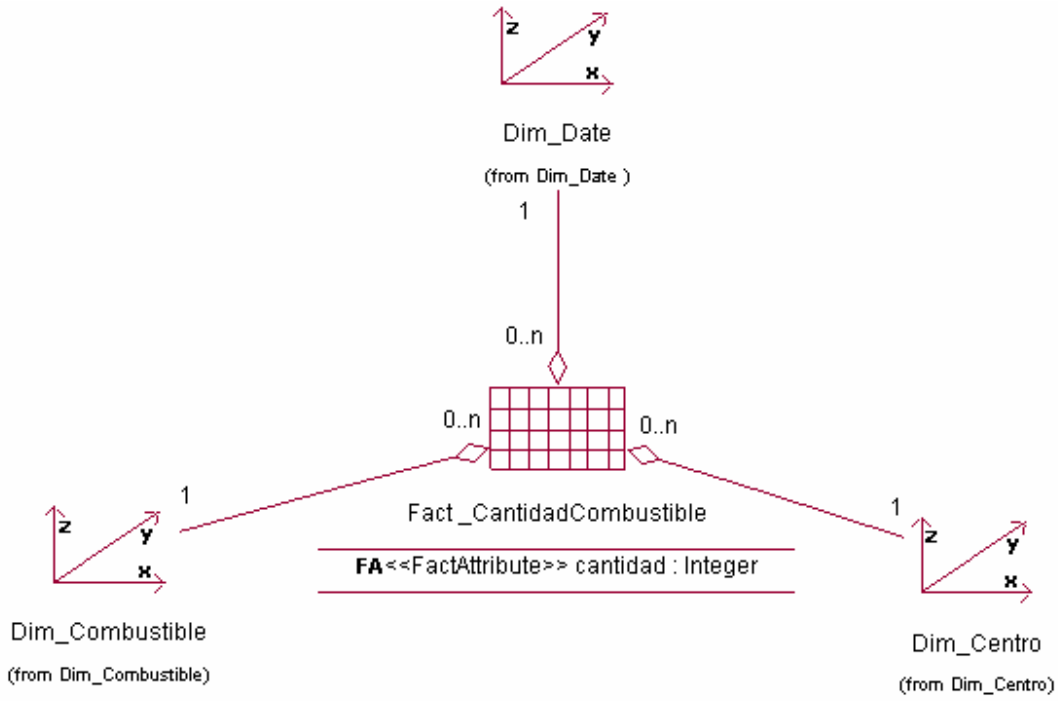


Figura 3. 42 Estructura del Hecho: Fact _CantidadCombustible.

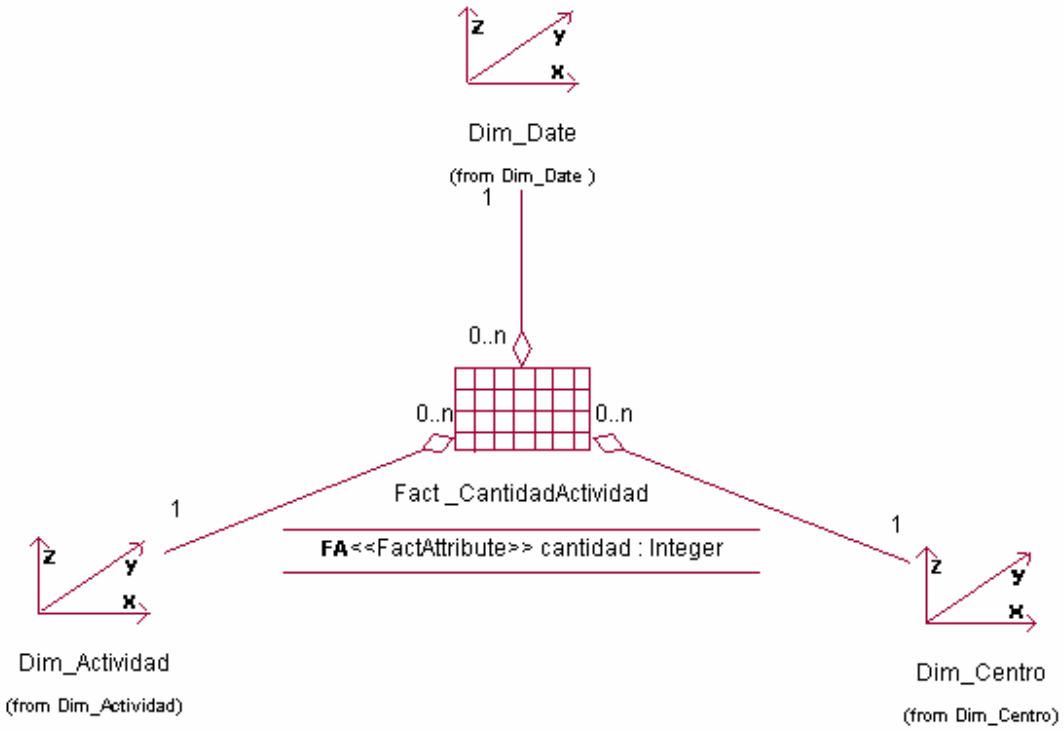


Figura 3. 43 Estructura del Hecho: Fact _CantidadActividad.

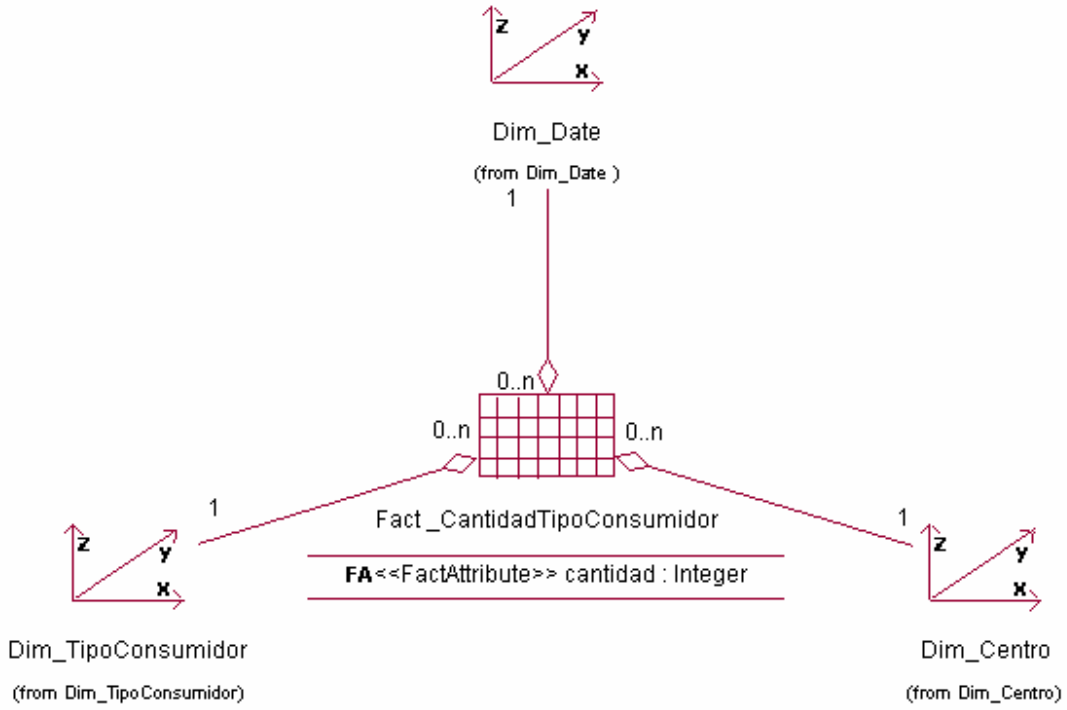


Figura 3. 44 Estructura del Hecho: Fact _CantidadTipoConsumidor.

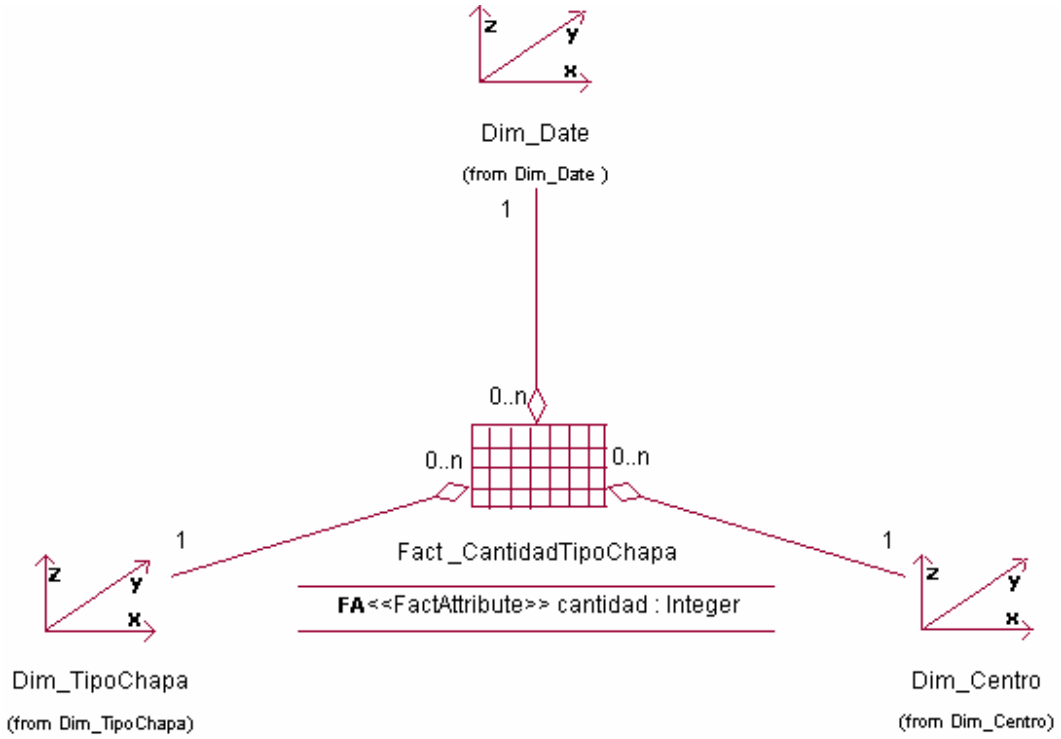


Figura 3. 45 Estructura del Hecho: Fact _CantidadTipoChapa.

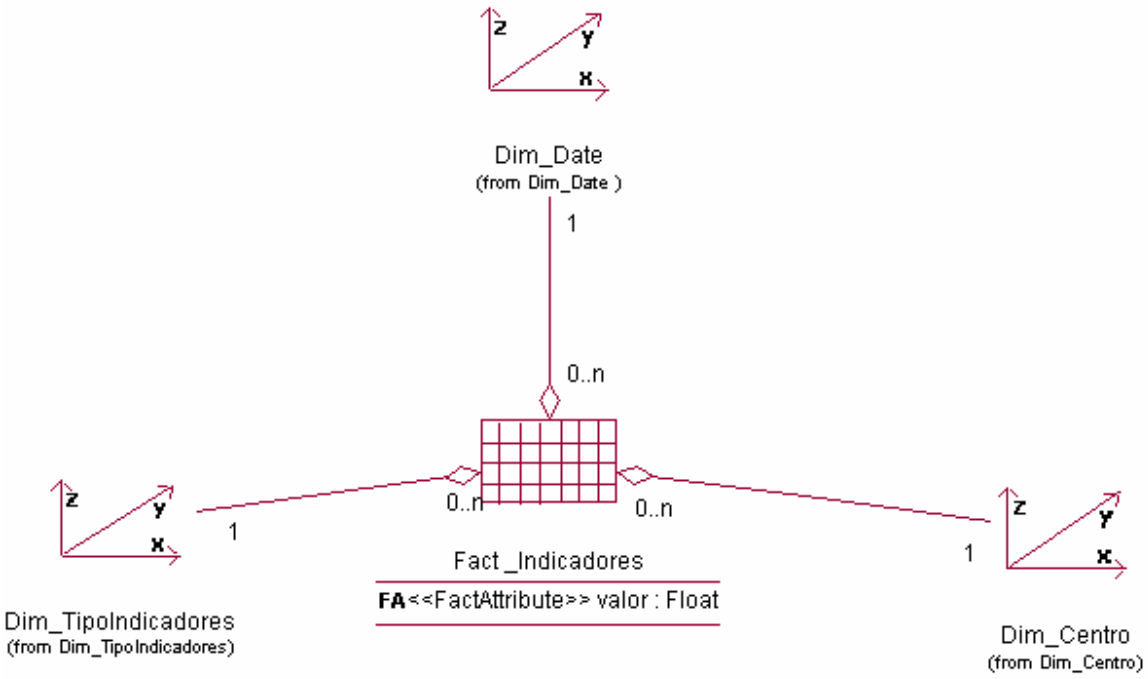


Figura 3. 46 Estructura del Hecho: Fact _Indicadores.

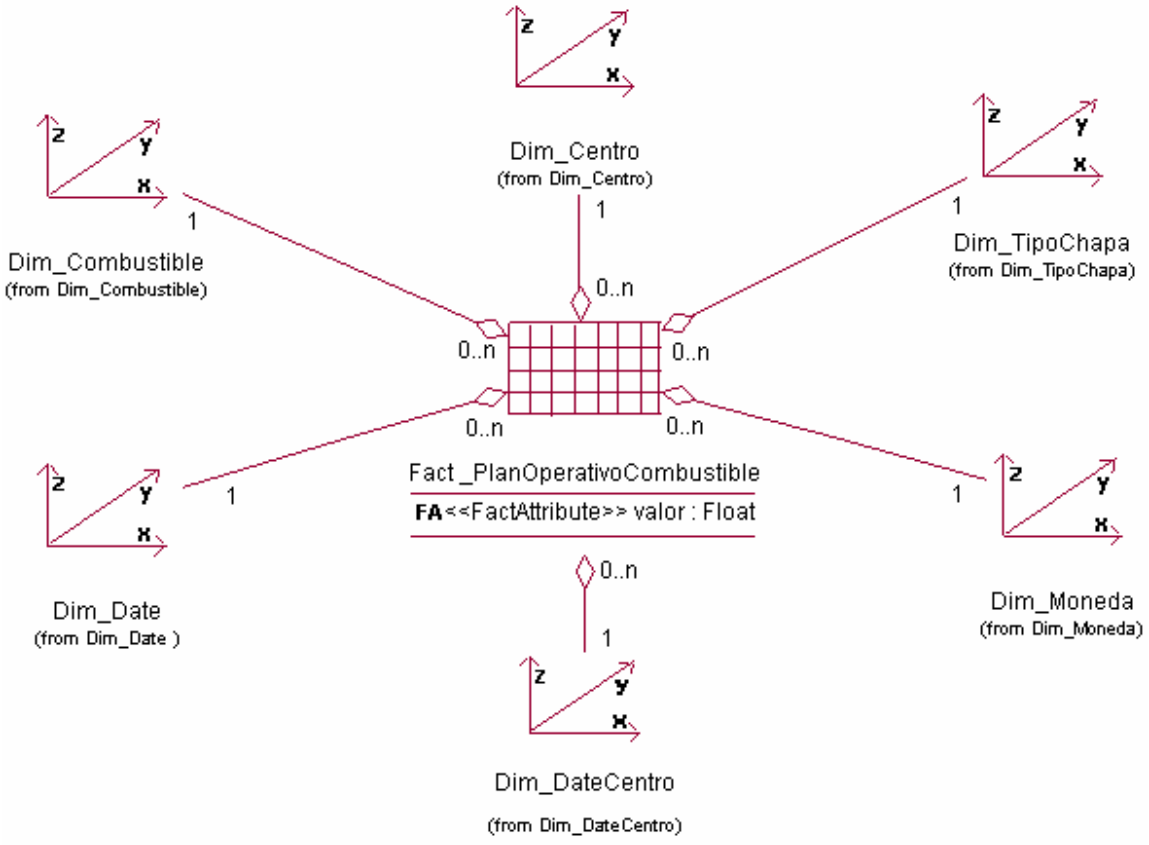


Figura 3. 47 Estructura del Hecho: Fact _PlanOperativoCombustible.

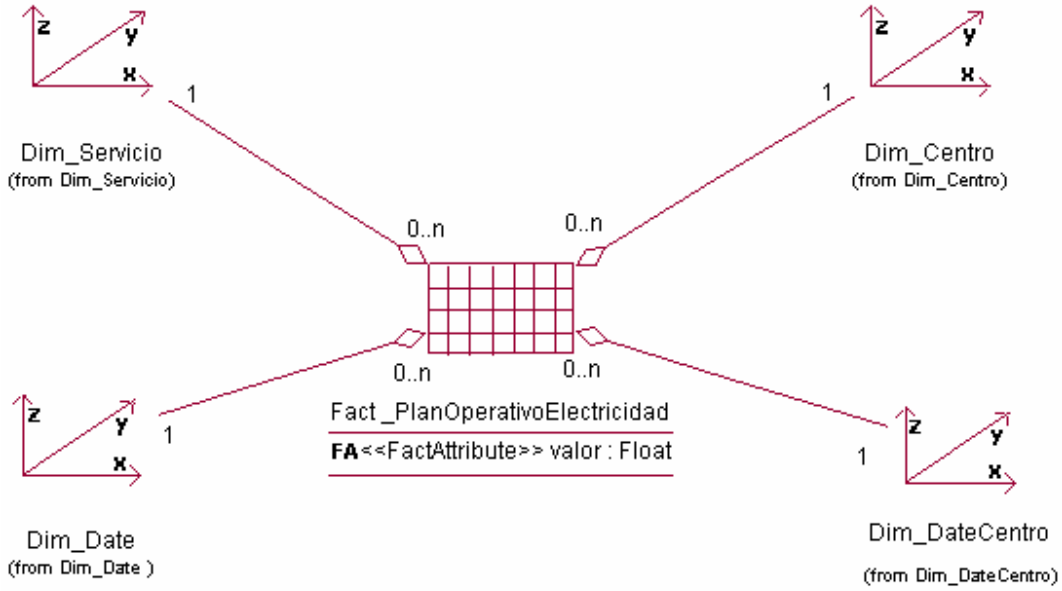


Figura 3. 48 Estructura del Hecho: Fact _PlanOperativoElectricidad.

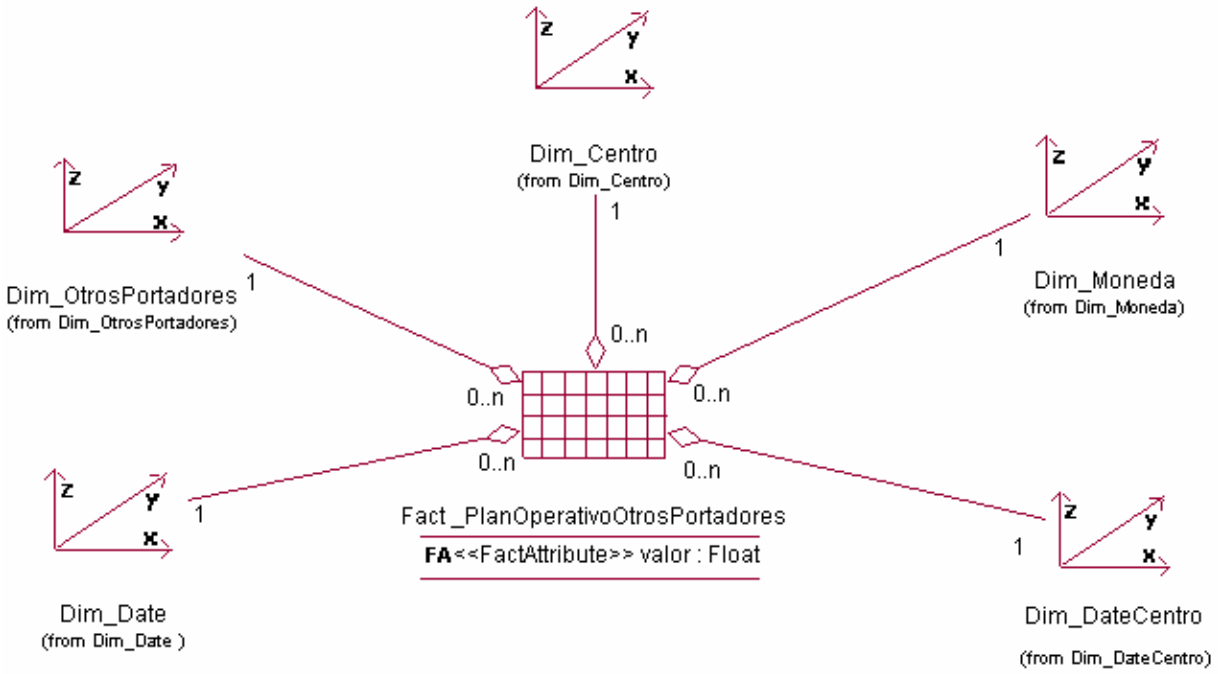


Figura 3. 49 Estructura del Hecho: Fact_PlanOperativoOtrosPortadores.

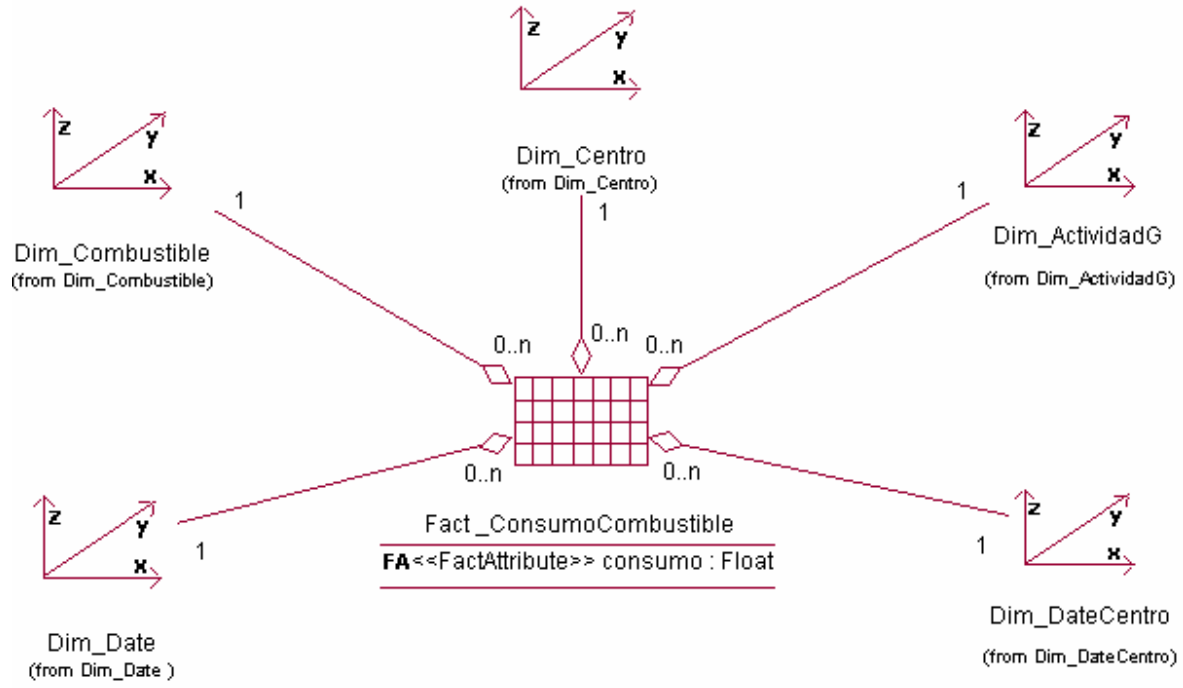


Figura 3. 50 Estructura del Hecho: Fact _ConsumoCombustible.

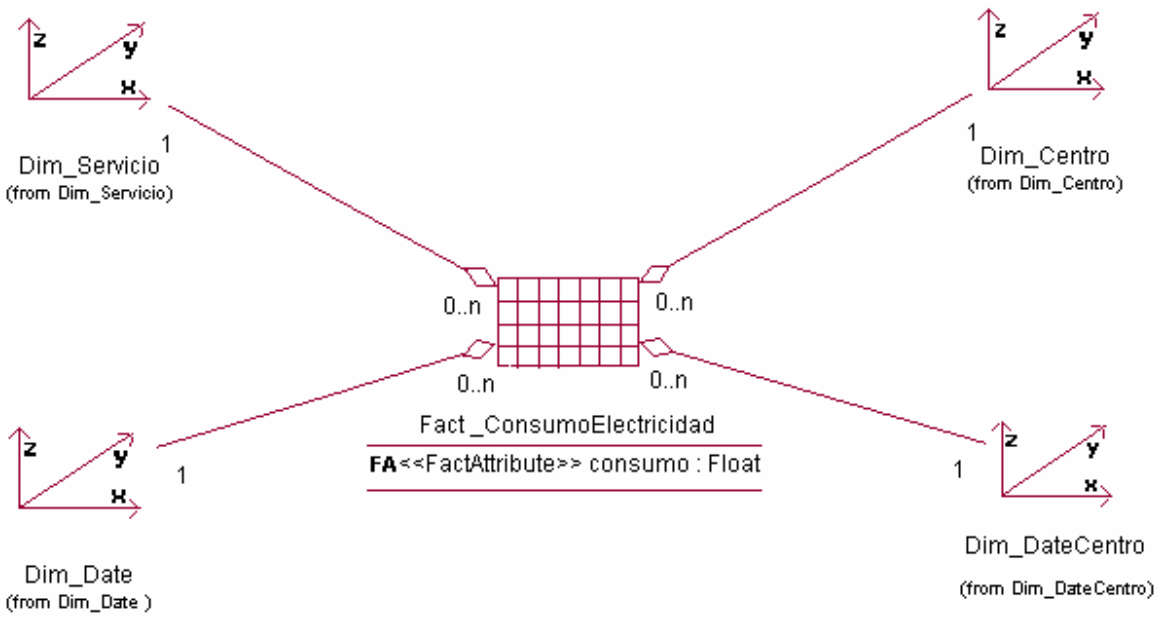


Figura 3. 51 Estructura del Hecho: Fact _ConsumoElectricidad.

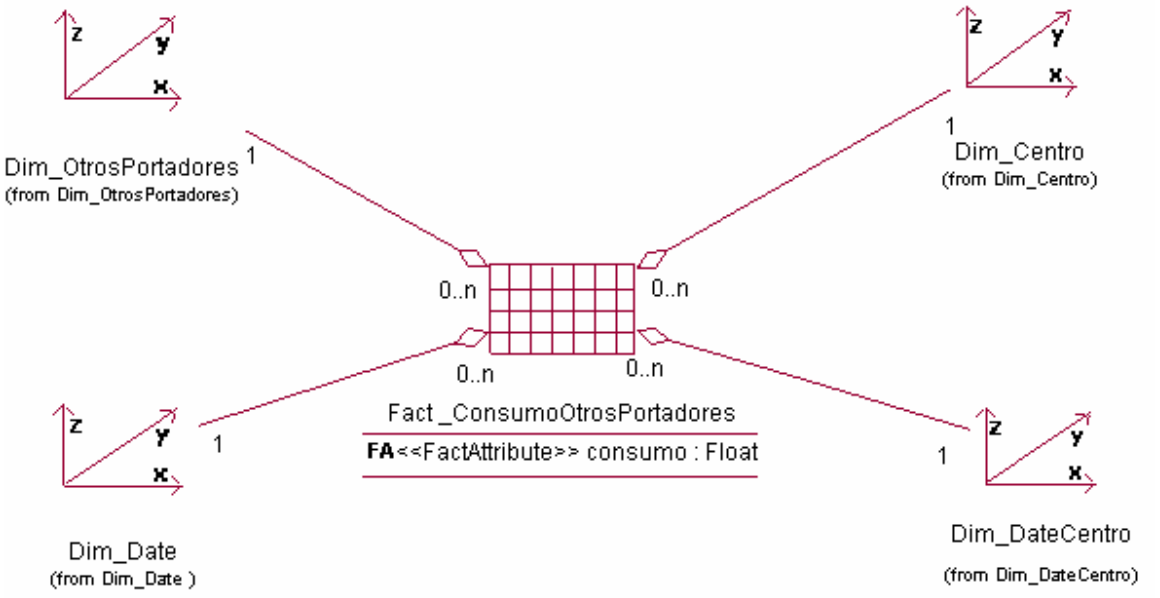


Figura 3. 52 Estructura del Hecho: Fact _ConsumoOtrosPortadores.

3.2.1.4 Mapeo de Datos

En toda solución de Inteligencia de Negocio que incluya la construcción de un DW o un Datamart es de vital importancia la ejecución de procesos de extracción, limpieza y carga de los datos (ETL) provenientes de las diferentes fuentes operacionales. Para ello, se emplean diagramas de mapeo que representan el flujo de los datos.

En el modelo intervienen tres elementos fundamentales:

- Proveedor: encargado de generar los datos (esquema, tabla o atributo).
- Consumidor: encargado de recibir los datos.
- Emparejamiento: forma y tareas (transformaciones) realizadas durante el mapeo de datos.

En esta propuesta de solución los datos que nutren el Datamart provienen de dos fuentes: un FTP o correo del MES.

El diagrama de mapeo de los datos quedaría de la siguiente forma (Figura 3.53):

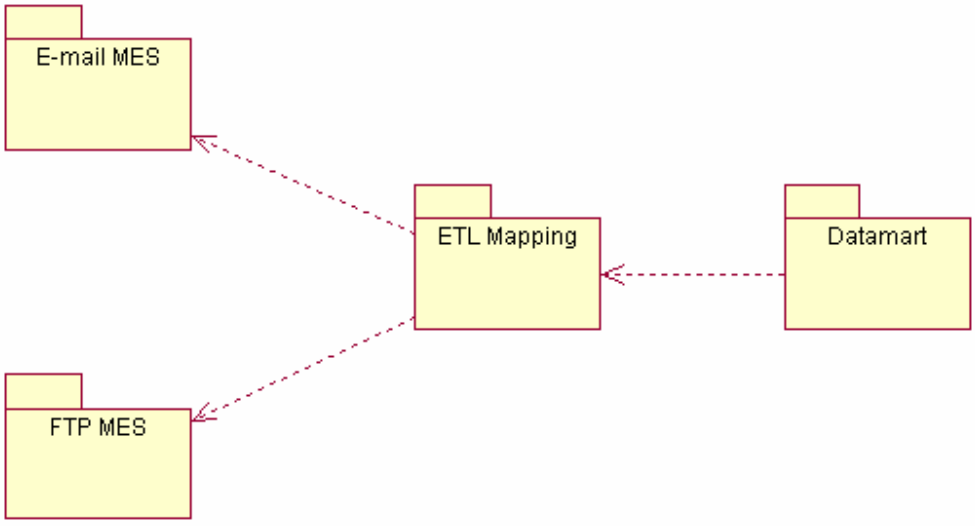


Figura 3. 53 Diagrama de Mapeo de los Datos.

El modelo lógico debe basarse en los resultados de los diagramas realizados en el Modelo Conceptual.

3.2.2 Modelo Lógico

Basados en los artefactos elaborados en el modelo anterior, se realizó un mapeo a los elementos que lógicamente podrían ser representados en la base de datos. O sea, a partir de las clases y relaciones modeladas se derivaron: tablas, columnas, tipos de datos, llaves, índices, dependencias y relaciones.

A continuación, se muestran los diagramas correspondientes a los esquemas estrella modelados, con sus respectivas tablas de hechos y dimensiones. Esta es una representación lógica de la base de datos (Figuras 3.54, 3.55, 3.56, 3.57, 3.58, 3.59, 3.60, 3.61, 3.62, 3.63, 3.64, 3. 65, 3.66, 3.67, 3.68, 3.69, 3.70, 3.71, 3.72, 3.73):

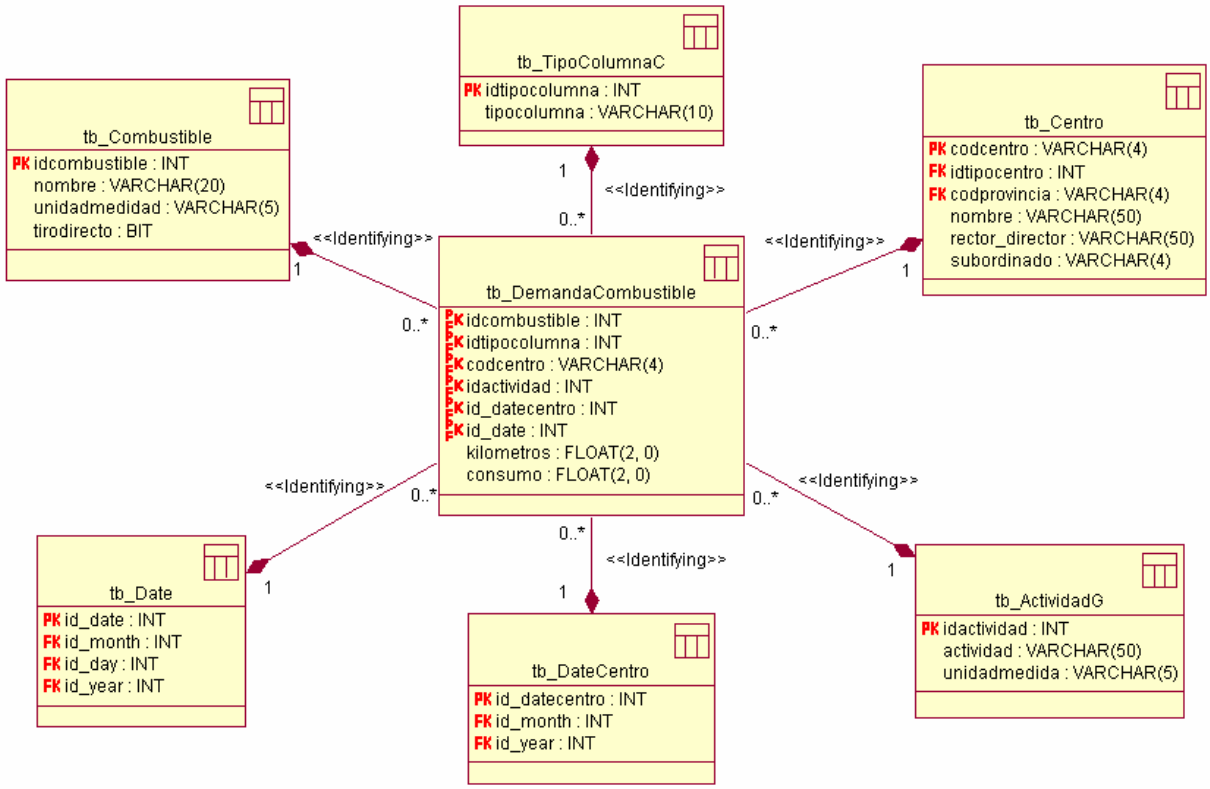


Figura 3. 54 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Combustible.

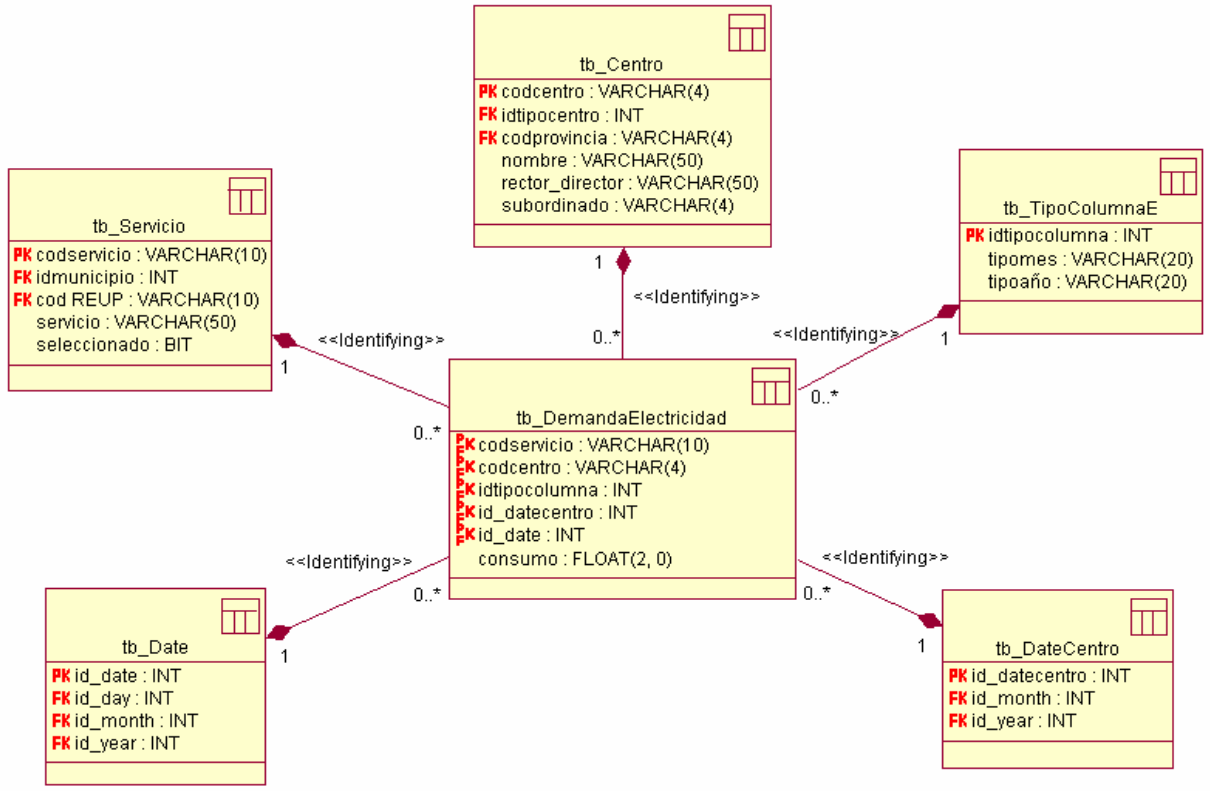


Figura 3. 55 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Electricidad.

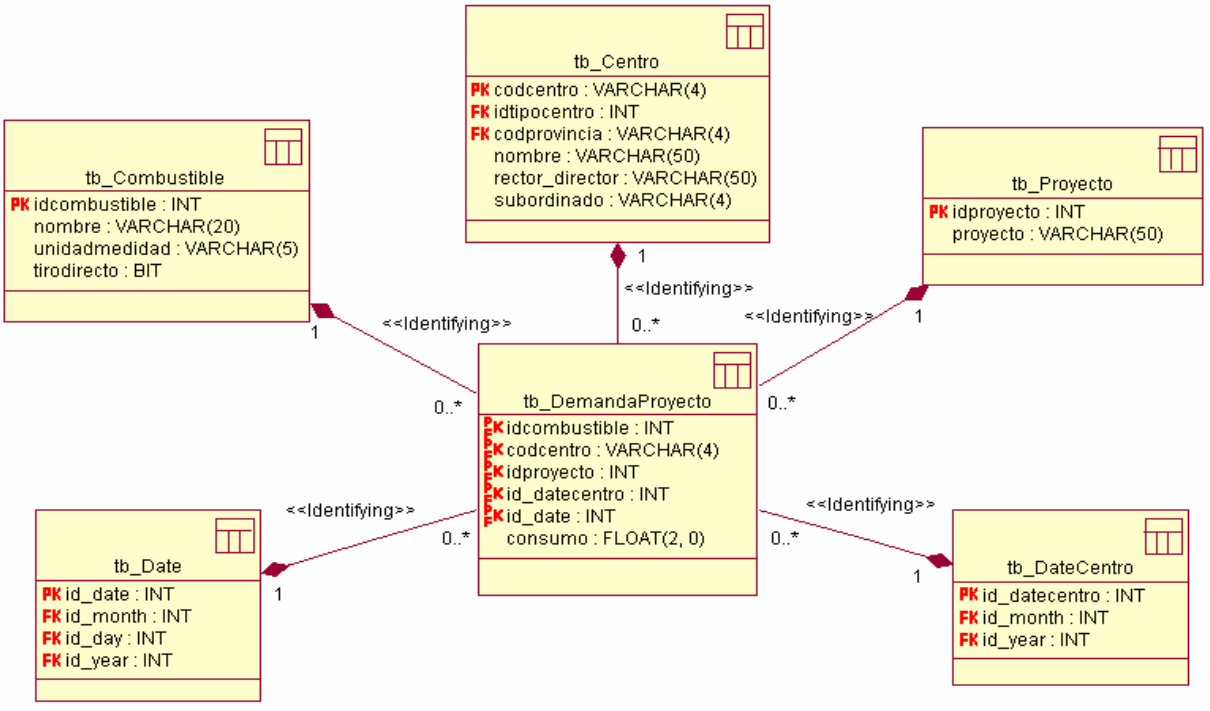


Figura 3. 56 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Proyecto.

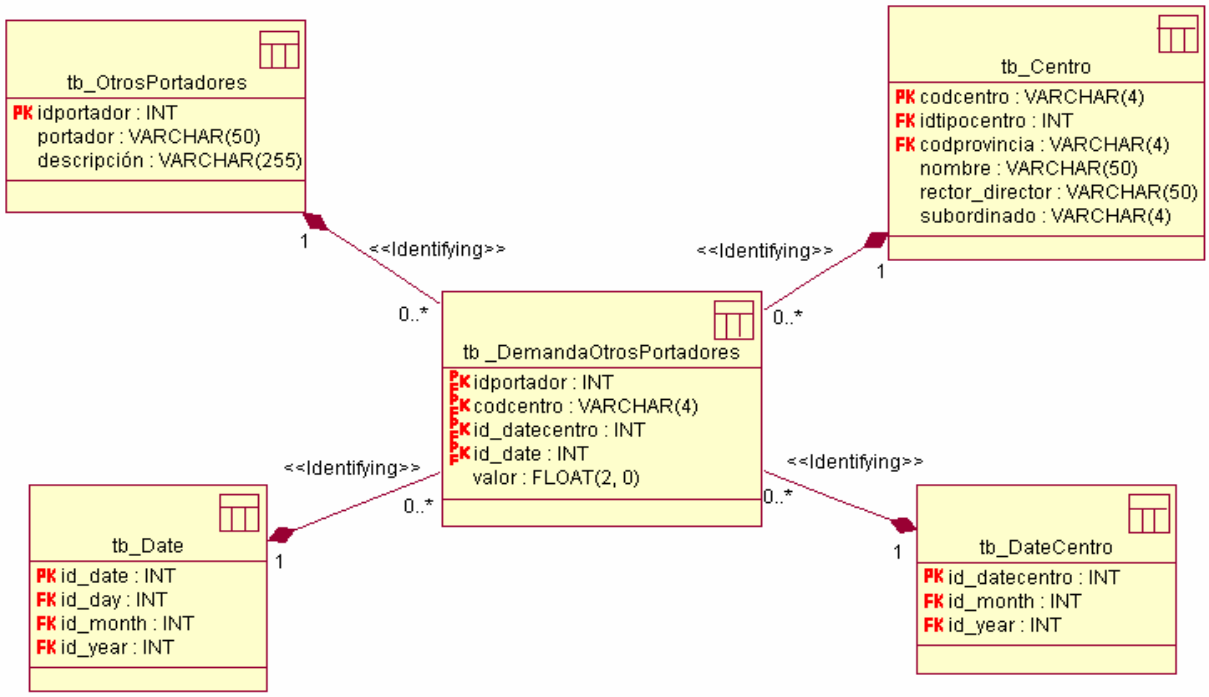


Figura 3. 57 Modelo Lógico del Esquema Estrella: StarSchema_ Dem_Portadores.

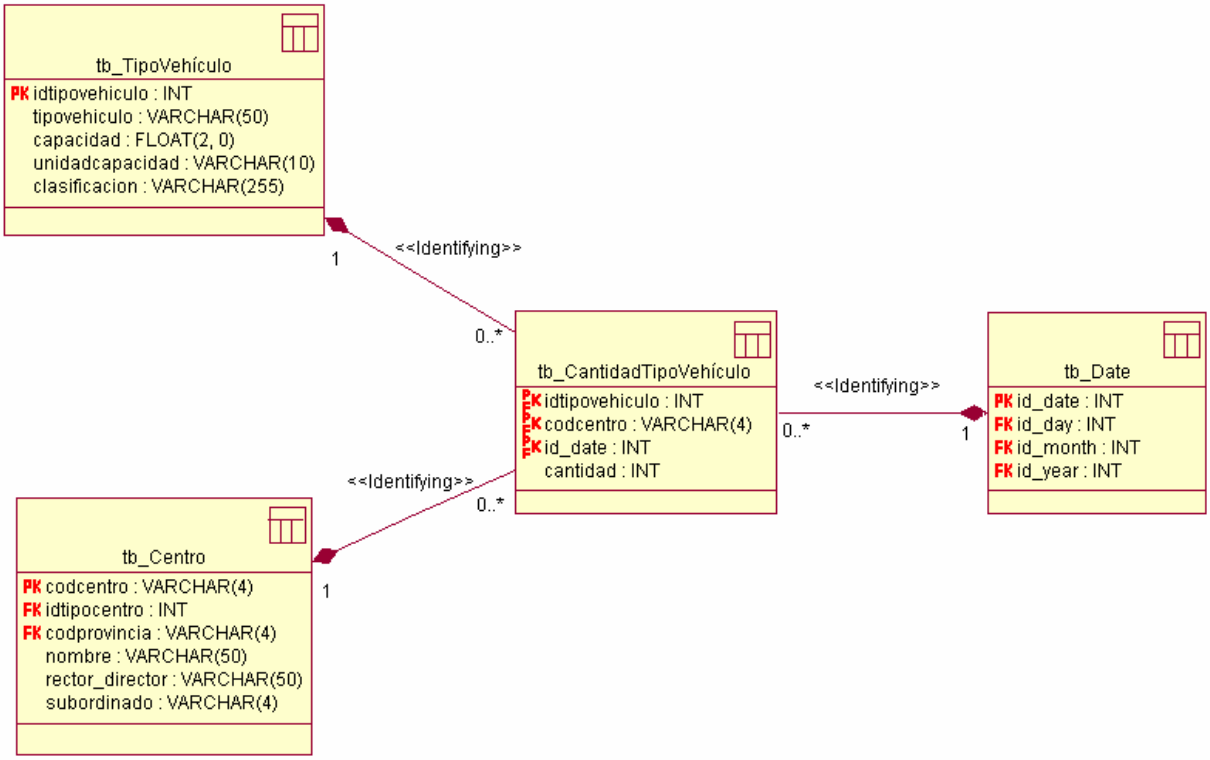


Figura 3. 58 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Vehículo.

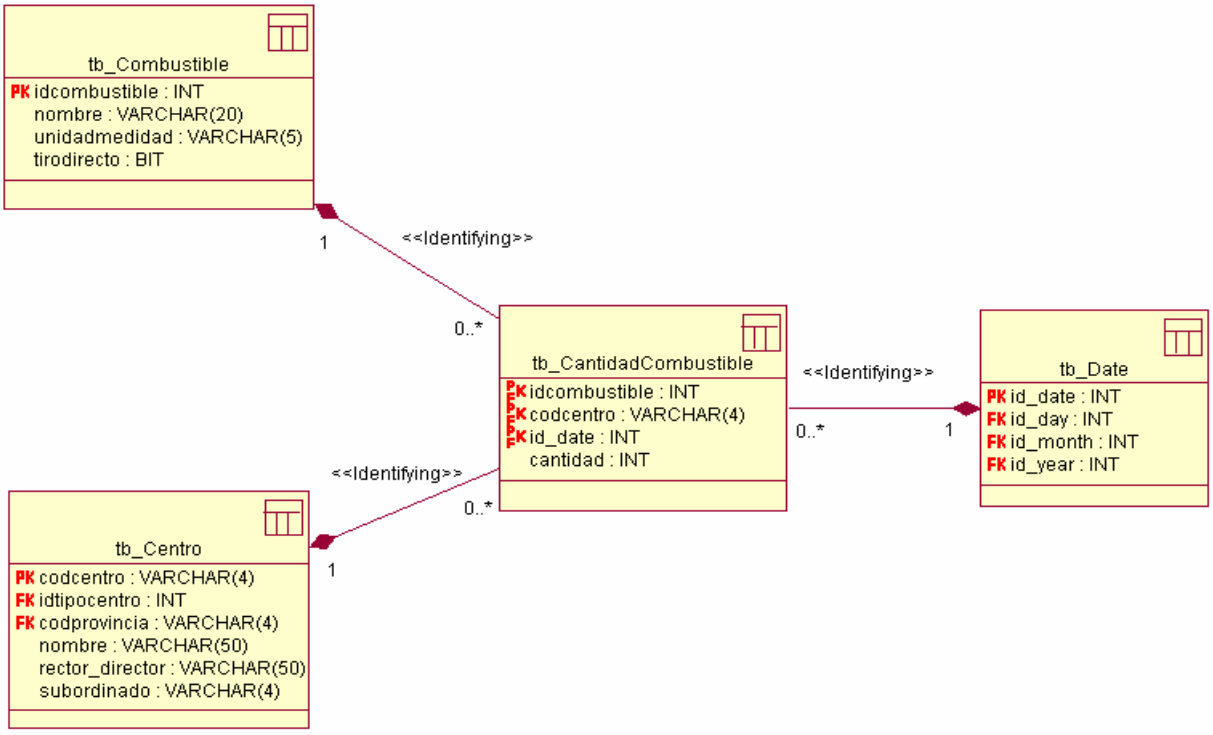


Figura 3. 59 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Combustible.

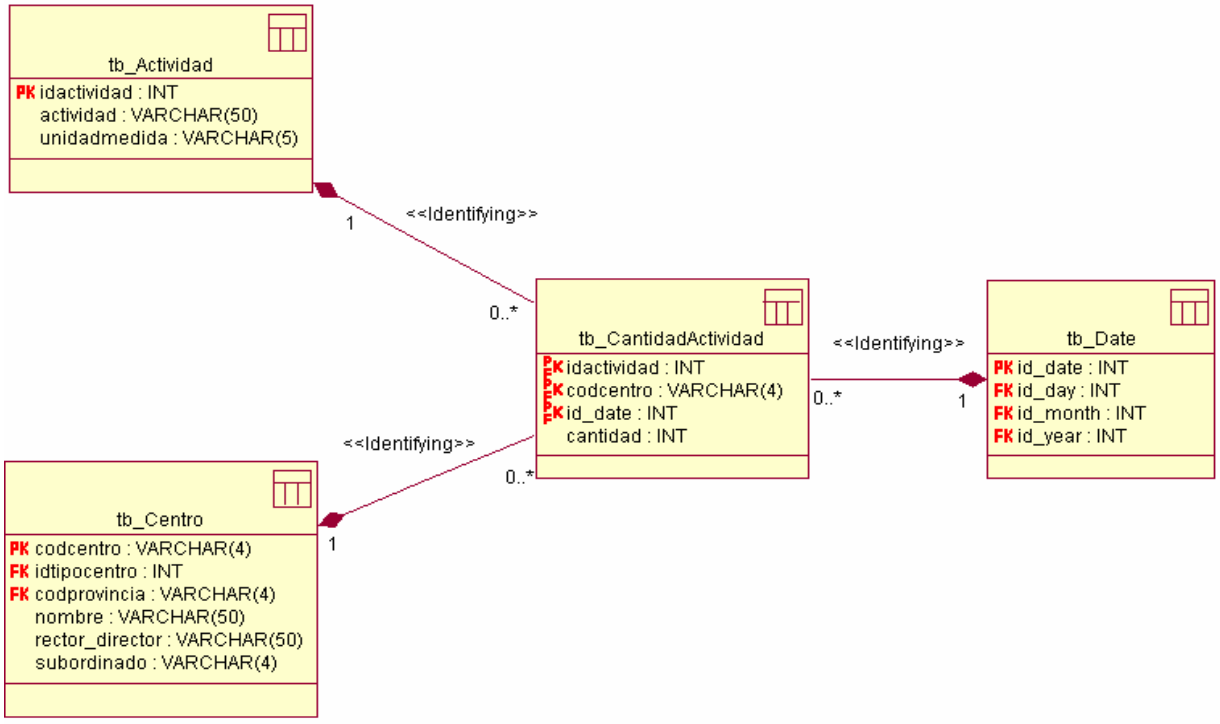


Figura 3. 60 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Actividad.

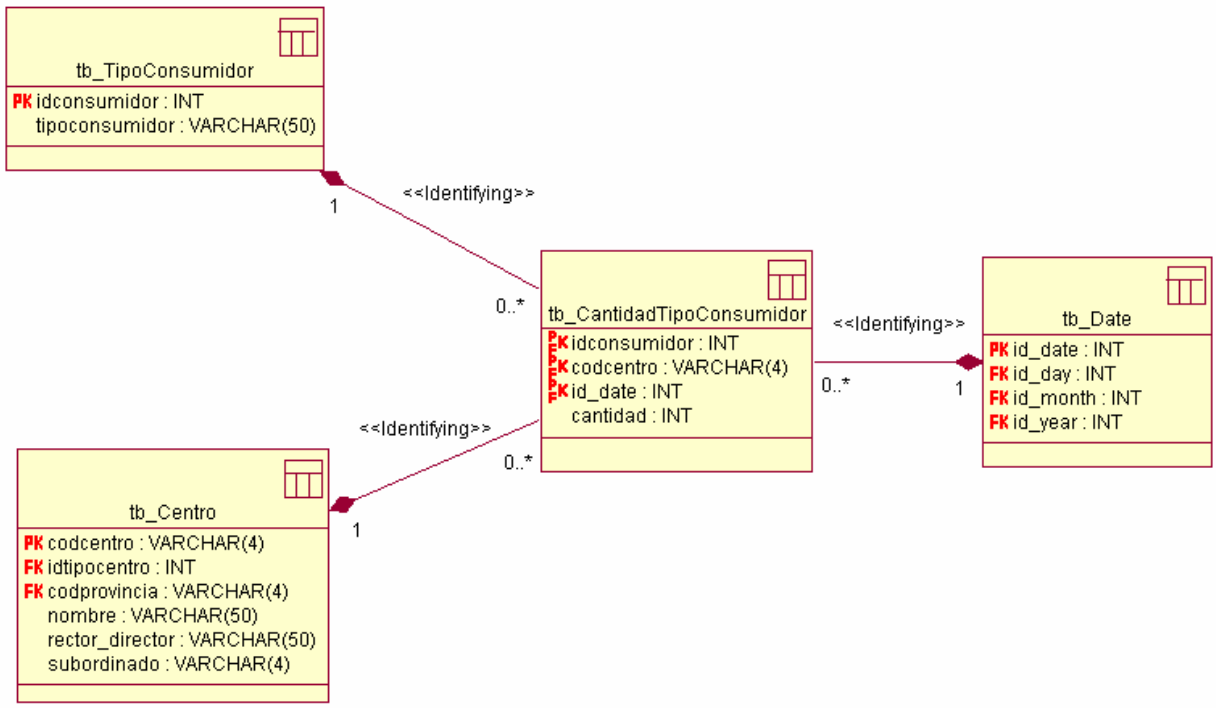


Figura 3. 61 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Consumidor.

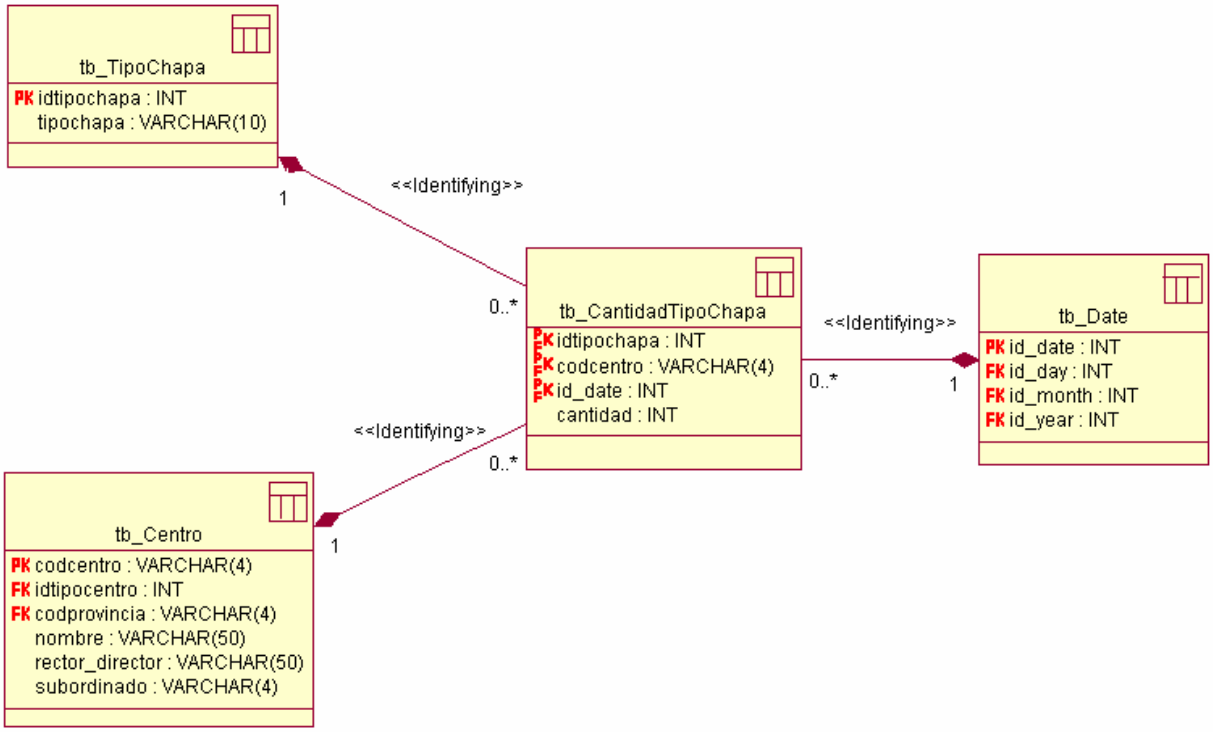


Figura 3. 62 Modelo Lógico del Esquema Estrella: StarSchema_ Cant_Chapa.

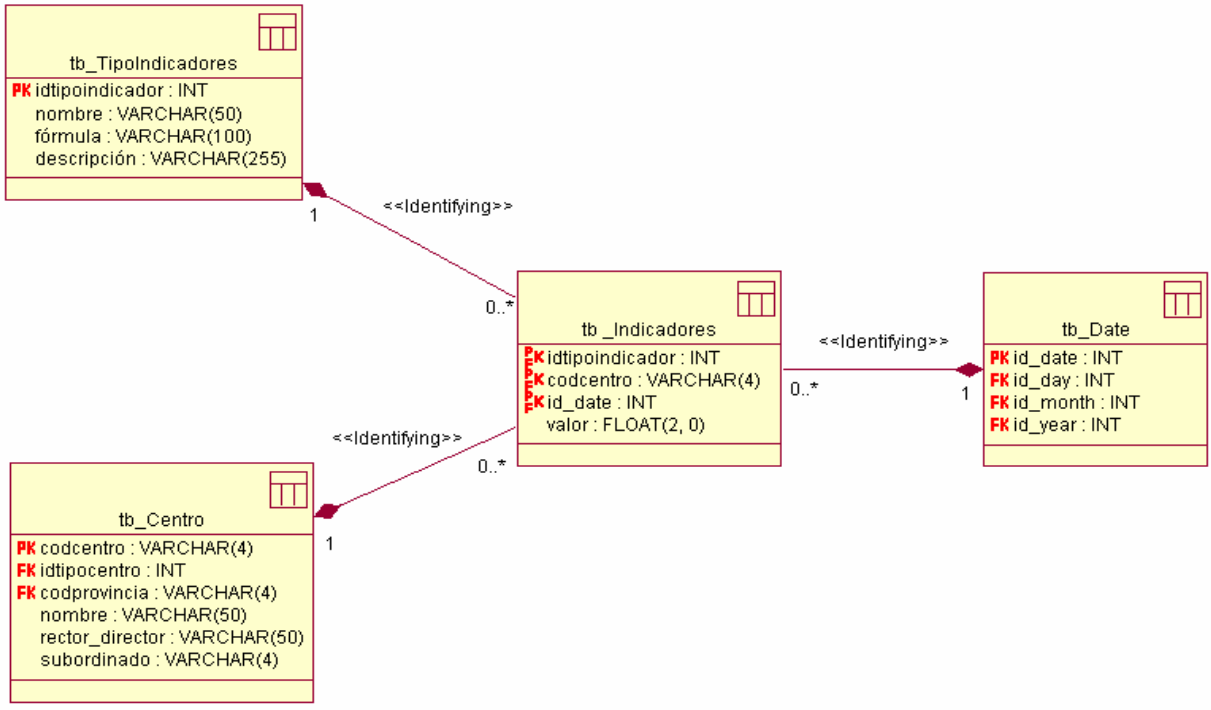


Figura 3. 63 Modelo Lógico del Esquema Estrella: StarSchema_Indicadores.

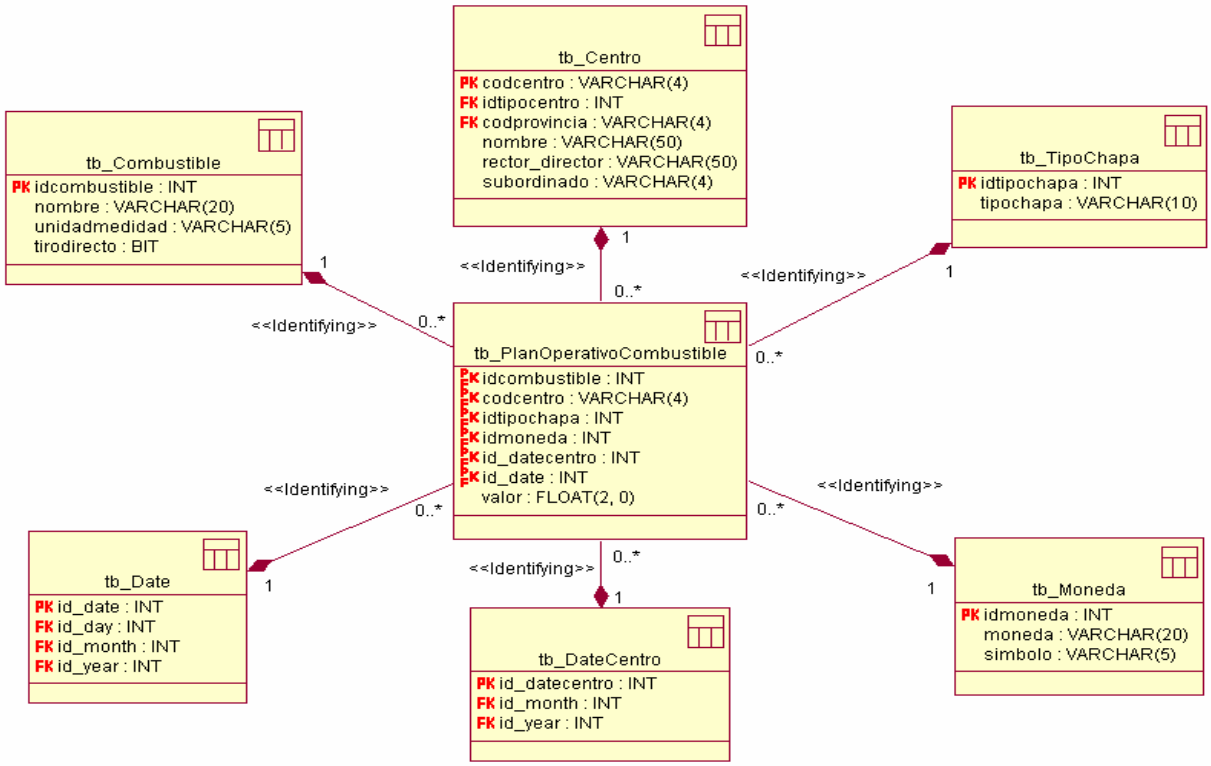


Figura 3. 64 Modelo Lógico del Esquema Estrella: StarSchema_Plan_Combustible.

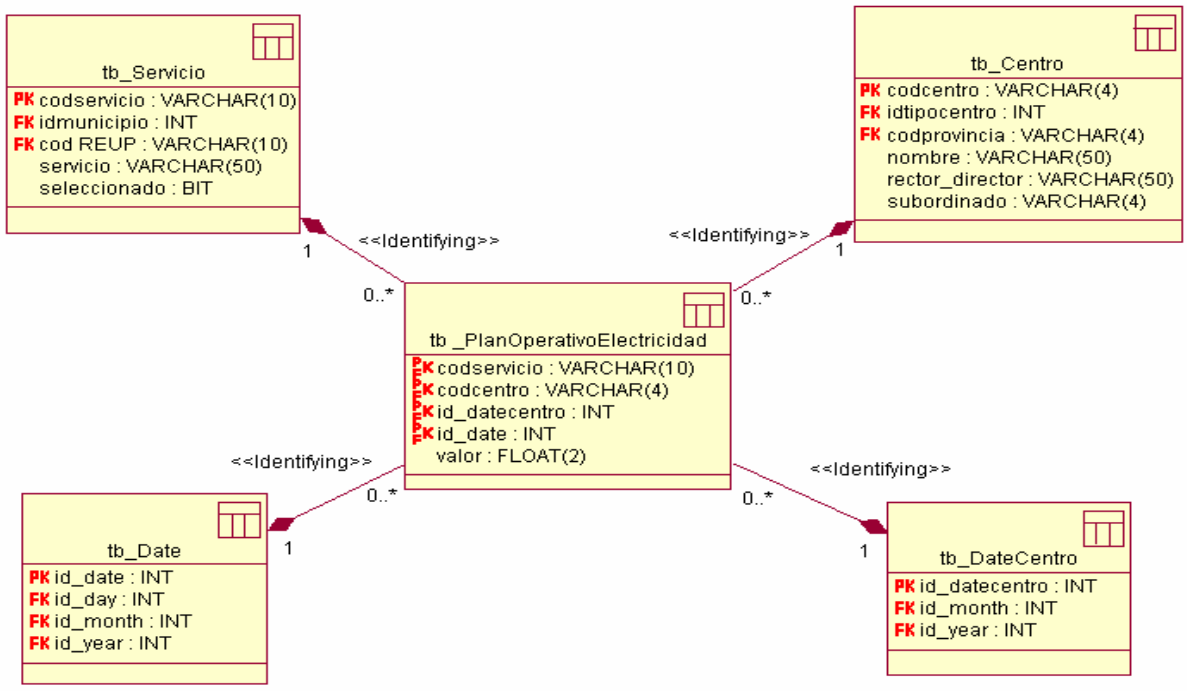


Figura 3. 65 Modelo Lógico del Esquema Estrella: StarSchema_ Plan_Electricidad

.

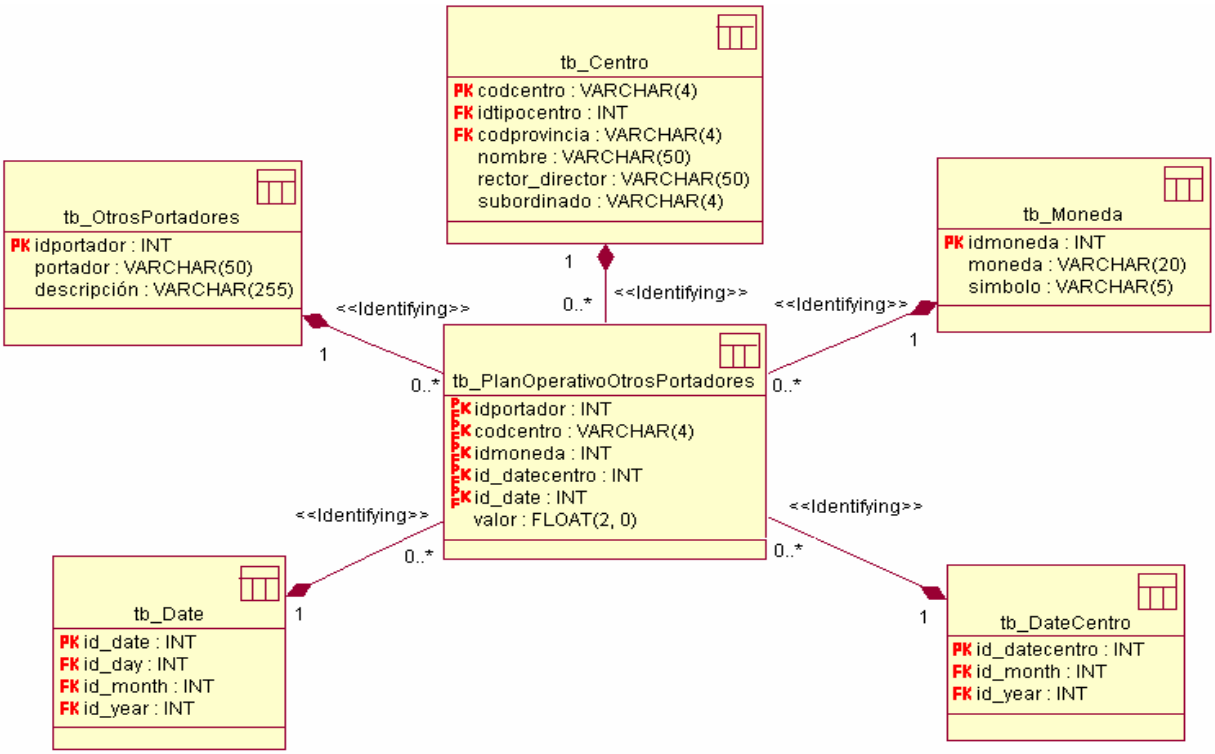


Figura 3. 66 Modelo Lógico del Esquema Estrella: StarSchema_ Plan_Portadores.

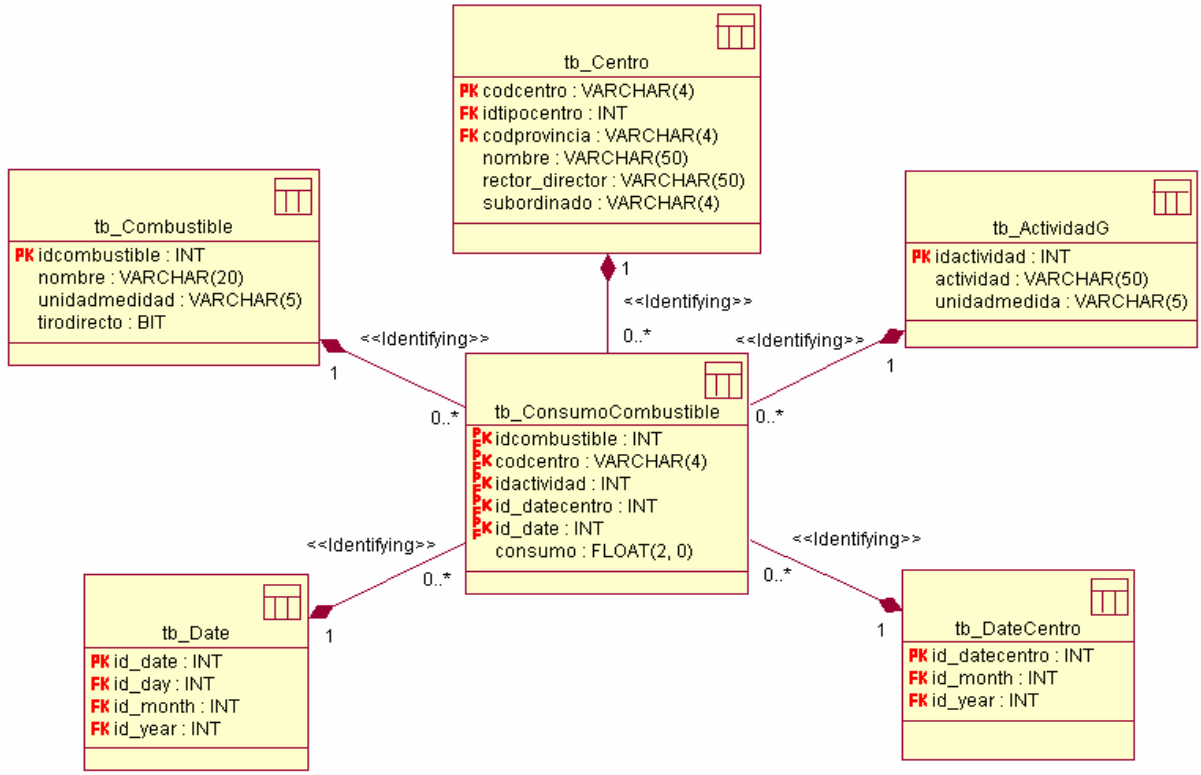


Figura 3. 67 Modelo Lógico del Esquema Estrella: StarSchema_ Cons_Combustible.

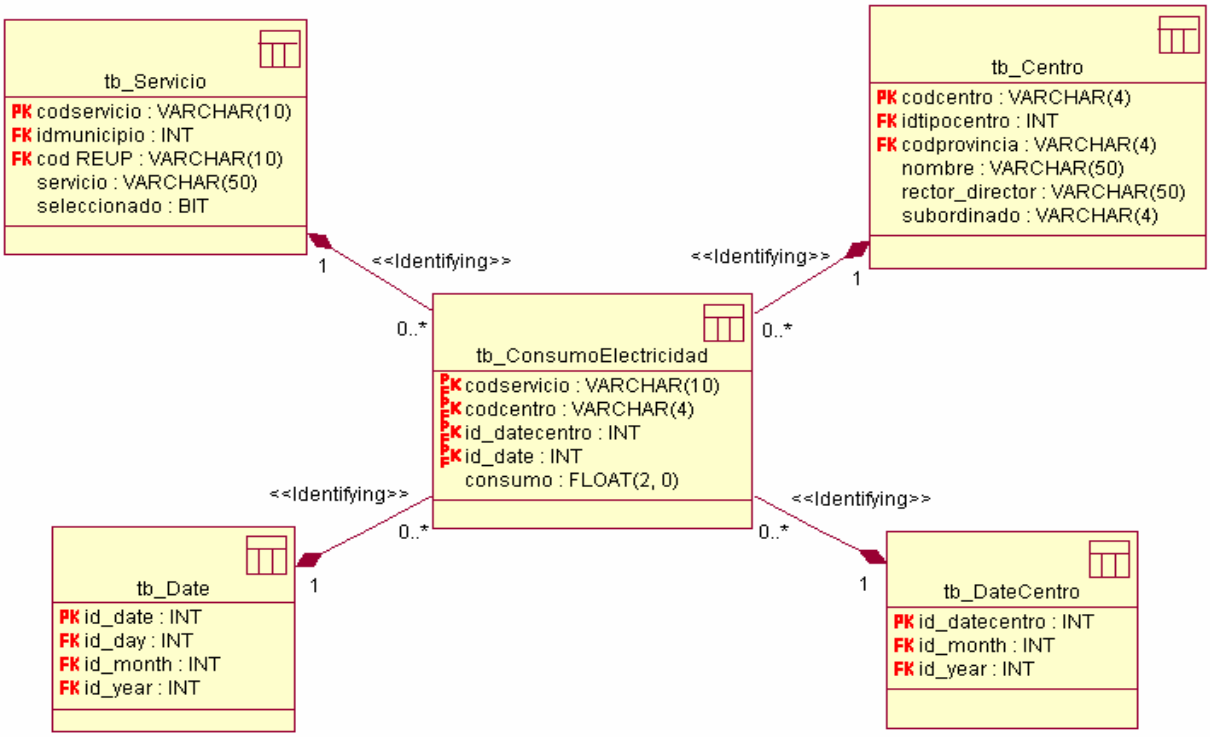


Figura 3. 68 Modelo Lógico del Esquema Estrella: StarSchema_ Cons_Electricidad.

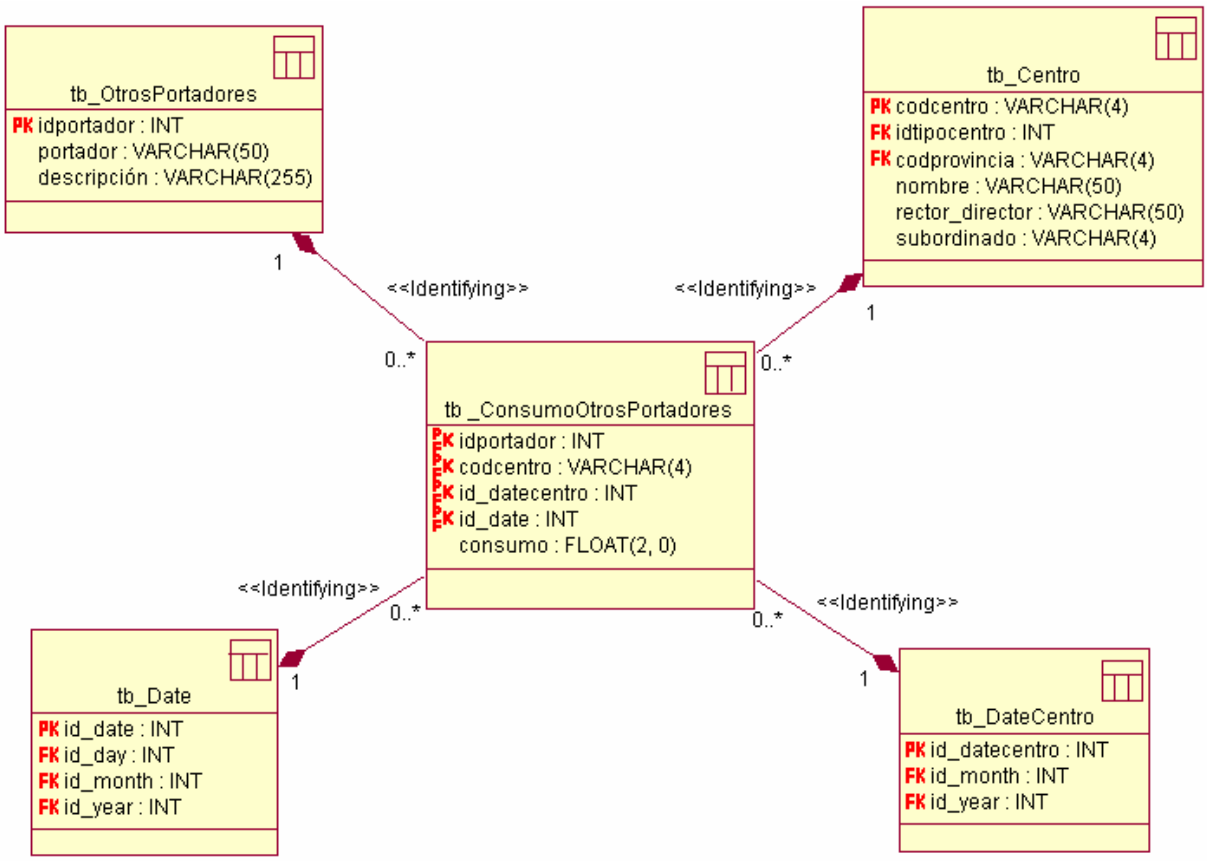


Figura 3. 69 Modelo Lógico del Esquema Estrella: StarSchema_ Cons_Portadores.

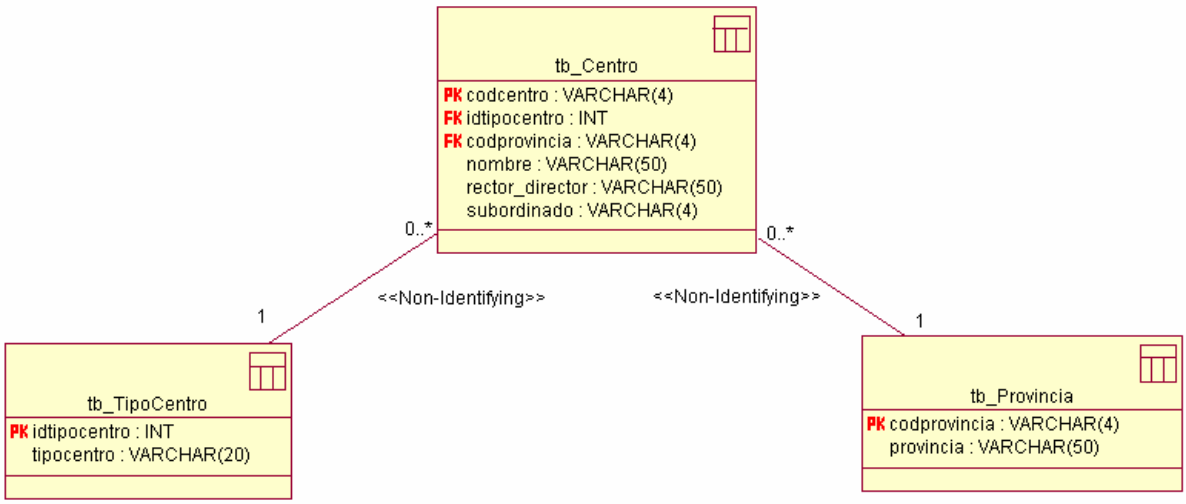


Figura 3. 70 Modelo Lógico de la Dimensión: Dim_Centro.

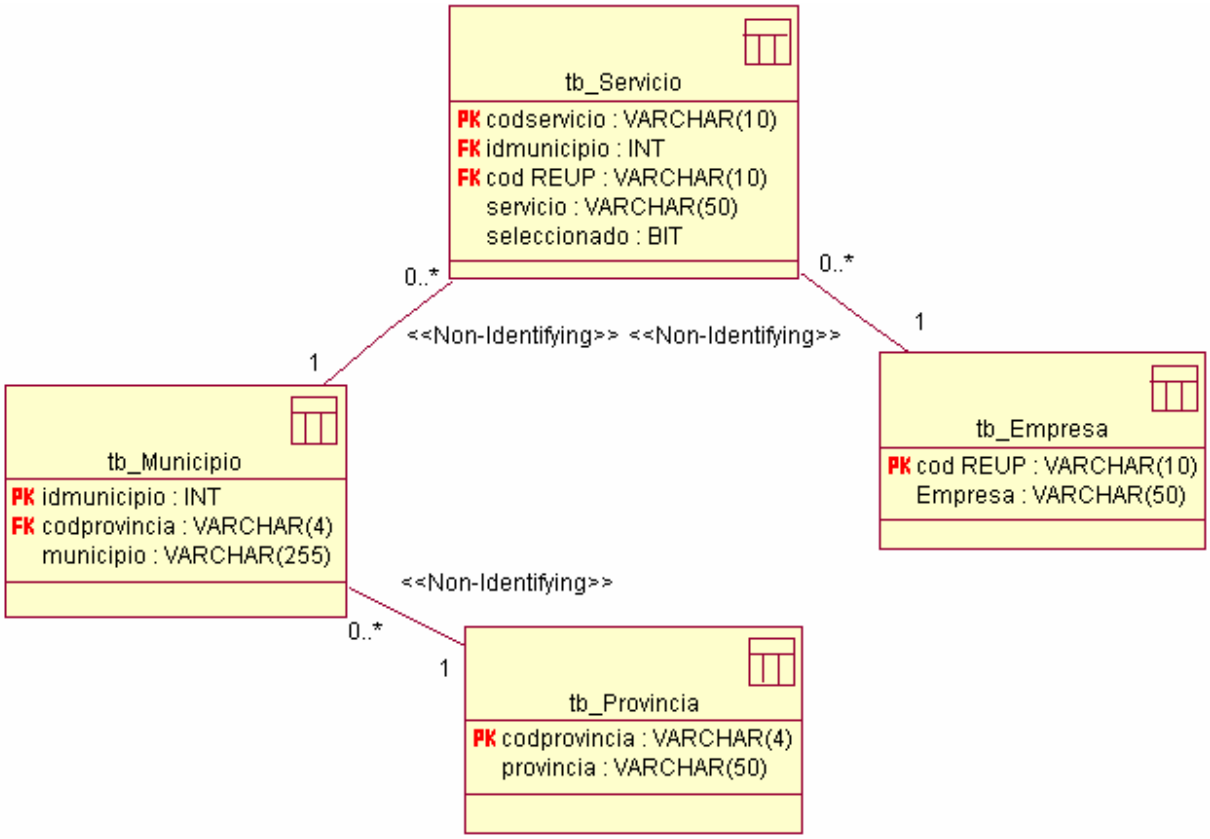


Figura 3. 71 Modelo Lógico de la Dimensión: Dim_Servicio.

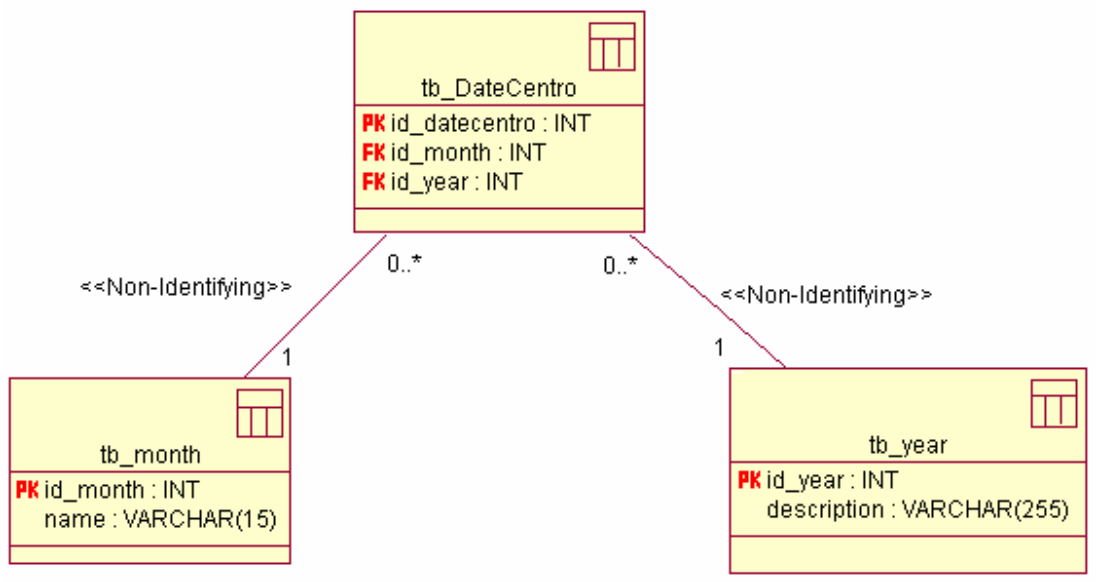


Figura 3. 72 Modelo Lógico de la Dimensión: Dim_DateCentro.

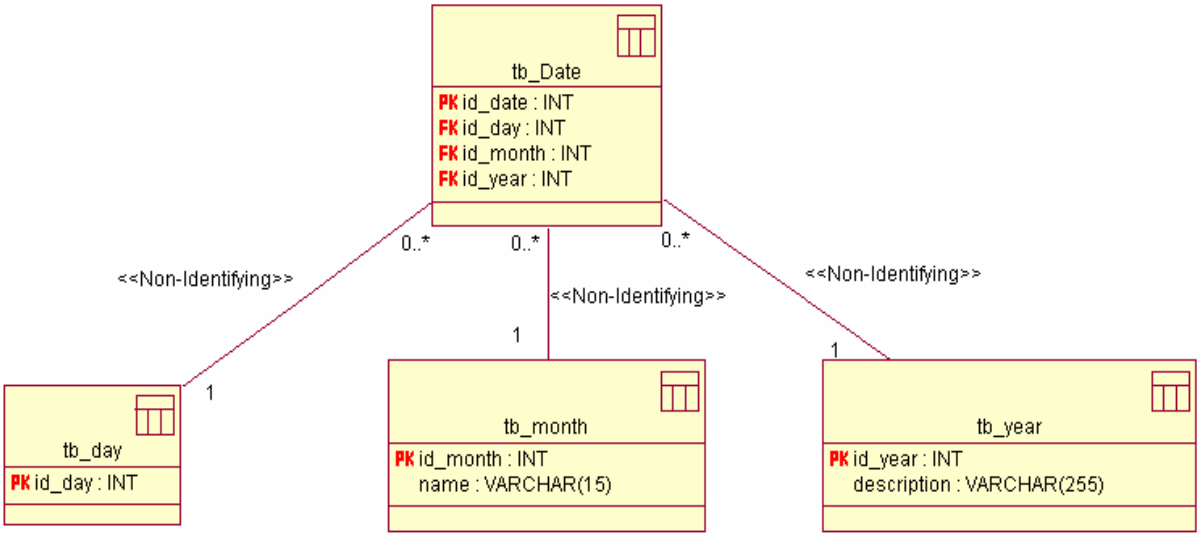


Figura 3. 73 Modelo Lógico de la Dimensión: Dim_Date.

3.2.2.1 Diseño de los Procesos ETL

Los datos que pueblan el Datamart provienen de dos fuentes opcionales: un FTP o correo del MES. Ambos sistemas deben proveer los datos necesarios para poblar el Datamart.

Para el diseño de los procesos ETL primeramente se debe analizar la calidad de los datos provenientes de las fuentes, para posteriormente definir las transformaciones necesarias con el propósito de lograr consistencia e integración de la información.

Dimensiones

Un DW tiene entre sus objetivos fundamentales el mantenimiento de la historia de la información, lo que constituye la principal diferencia de este tipo de arquitectura con los sistemas OLTP. En los sistemas OLTP la información es operacional y está sometida a constantes actualizaciones, acción que no debe realizarse en un DW. Por tal motivo, debe de seleccionarse que técnica desarrollar ante la ocurrencia de pequeños cambios, en el tiempo, en los datos que nutren las dimensiones, con el objetivo de mantener la consistencia e integración de la información. Este fenómeno se conoce como: *"Slowly Changing Dimensions".*

Para tratar este problema existen varias técnicas, la seleccionada para este trabajo es una implementación propuesta por uno de los padres del Data Warehouse, *Ralph Kimball*, y constituye una de las técnicas más comúnmente usada en este tipo de implementaciones. Una de sus ventajas fundamentales es que permite el rastreo de un número ilimitado de cambios. **[Simón, 2005]**

En este método se crea un nuevo registro para cada cambio ocurrido a un elemento de la dimensión tratada. La llave primaria debe de ser una llave auto generada (*Surrogate Key*) y que difiera de la que poseía el registro en el sistema operacional. Esta técnica simplifica la longitud del enlace entre las tablas de dimensiones y las tablas de hechos. **[Simón, 2005]**

Para permitir establecer la relación entre los registros correspondientes a un mismo elemento y su monitoreo en el tiempo, se emplean dos atributos: OLTP_ID y Version, el primero representa el identificador del elemento en el sistema operacional y el segundo representa la versión del cambio a que corresponde el registro en cuestión. Para finalizar solo se necesita conocer el rango de tiempo en que estuvo vigente cada representación de un elemento, para ello, se introducen dos atributos: Date_From y Date_To, los que representan la fecha de inicio y fecha de fin de dicho período de tiempo. **[Pentaho, 2007a]**

Las tablas de dimensiones deben tener la siguiente estructura:

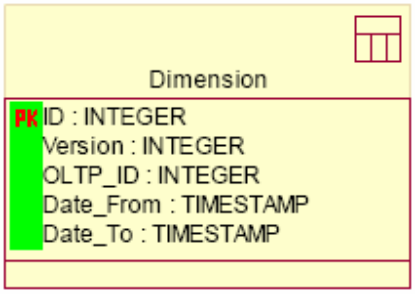


Figura 3. 74 Prototipo de la Estructura de las Tablas de Dimensiones.

En este trabajo, los procesos ETL asociados a las dimensiones se componen de los siguientes elementos:

1. Fuente de datos (FTP, correo).
2. *Filter*: se utiliza para evitar la carga de registros incompletos o indeseados.
3. *Incorrect*: se utiliza para redirigir los registros que no cumplieron con las restricciones establecidas en el filtro.
4. *Log*: se utiliza para registrar los errores: registros incompletos o indeseados.
5. *Surrogates Key*: se utiliza para generar la nueva llave primaria del registro cargado.
6. *Loader*: en este elemento se realizan las tareas relacionadas con el tratamiento del "*Slowly Changing Dimensions*" y finalmente se cargan los datos en el Datamart.
7. Destino de los datos (tablas del Datamart).

A continuación, se muestran los diagramas de los procesos ETL correspondientes a las tablas de dimensiones (Figuras 3.75, 3.76, 3.77, 3.78, 3.79, 3.80, 3.81, 3.82, 3.83, 3.84, 3.85, 3. 86, 3.87, 3.88, 3.89, 3.90, 3.91, 3.92):

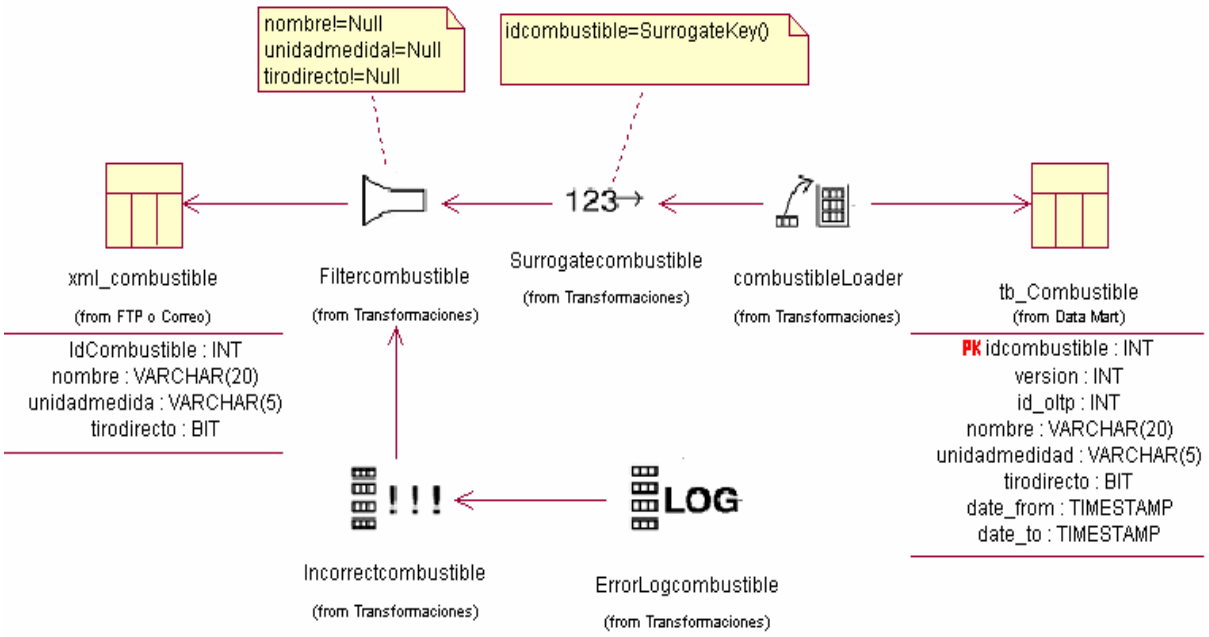


Figura 3. 75 Proceso ETL correspondiente a la dimensión: Dim_Combustible

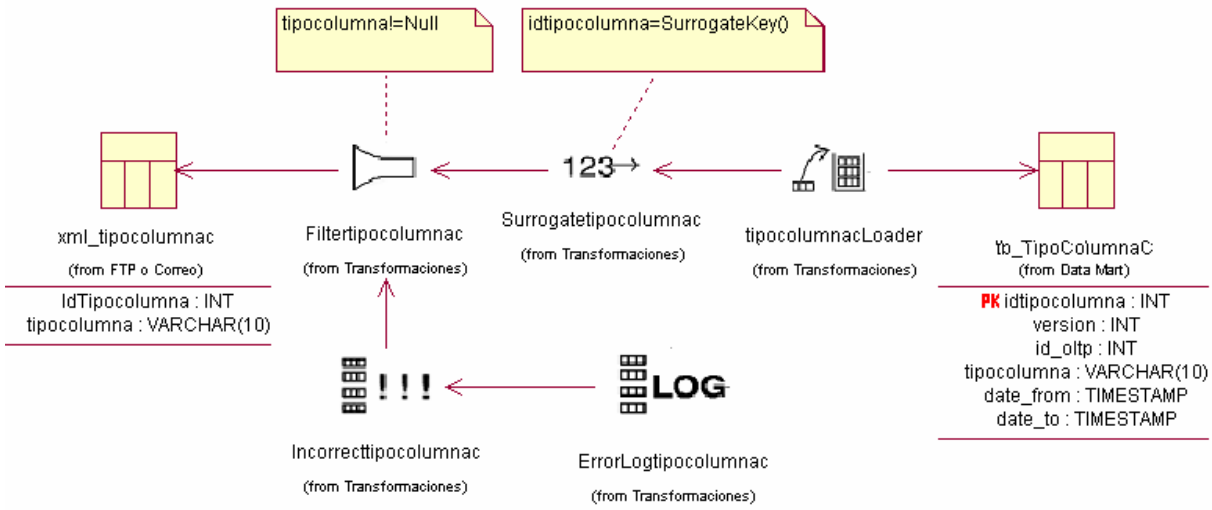


Figura 3. 76 Proceso ETL correspondiente a la dimensión: Dim_TipoColumnaC

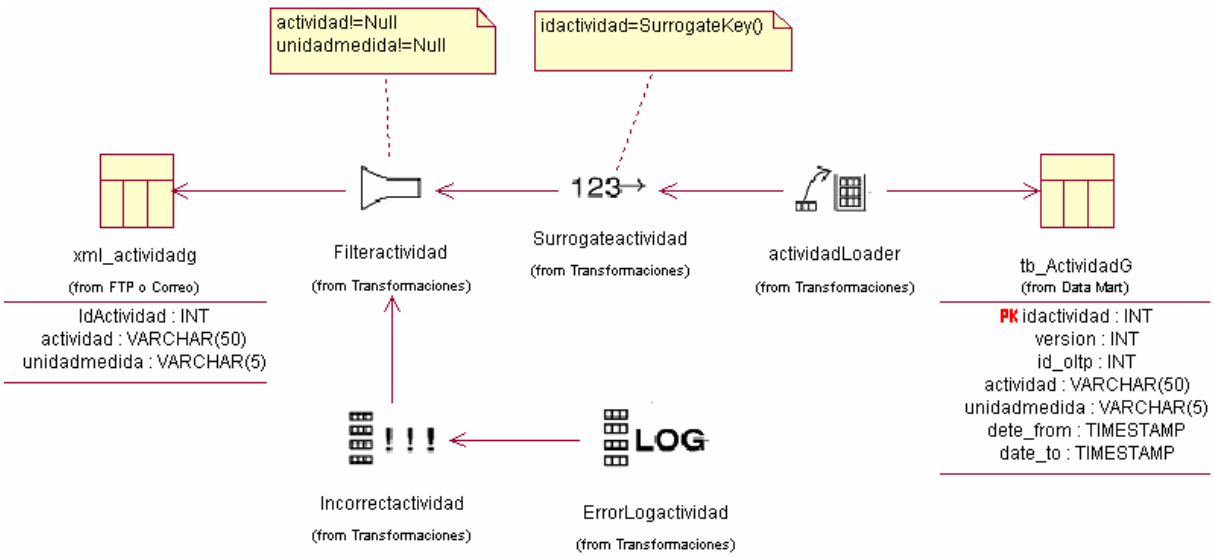


Figura 3. 77 Proceso ETL correspondiente a la dimensión: Dim_ActividadG

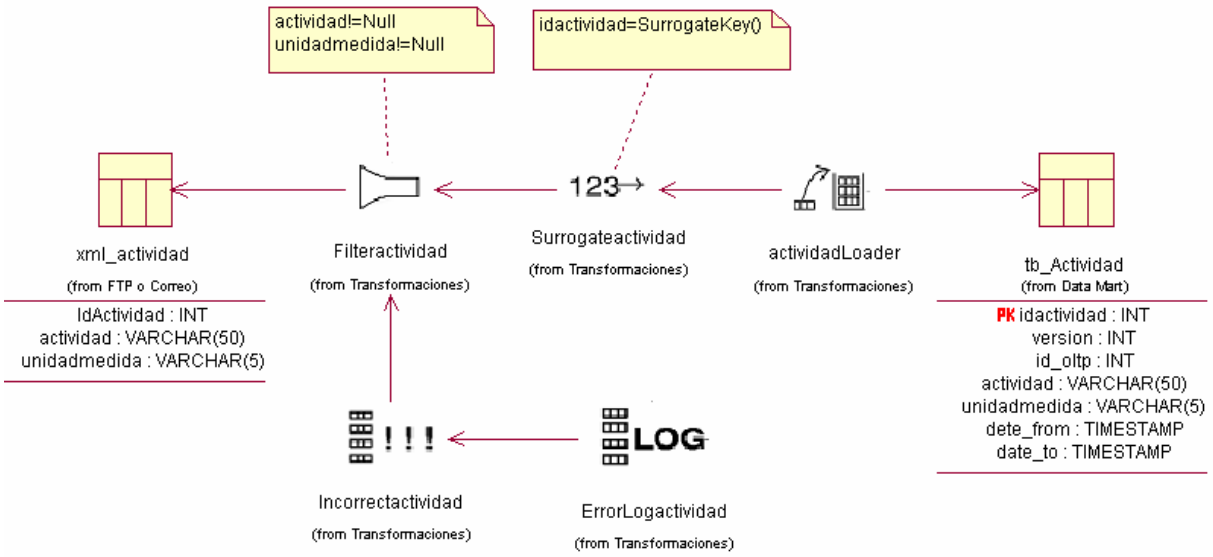


Figura 3. 78 Proceso ETL correspondiente a la dimensión: Dim_Actividad

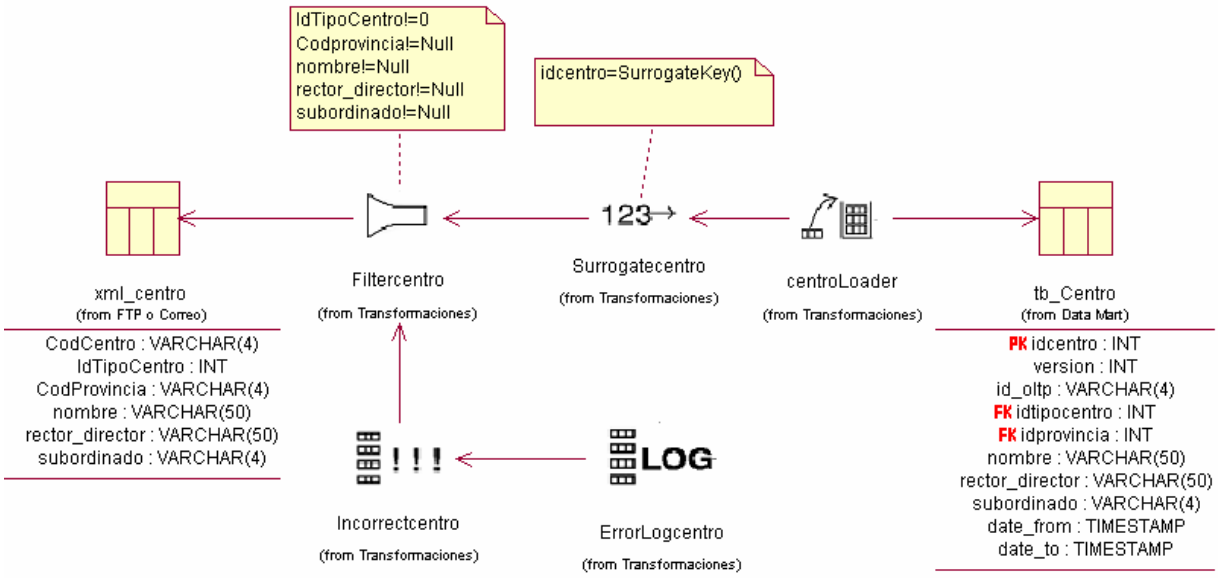


Figura 3. 79 Proceso ETL correspondiente a la dimensión: Dim_Centro

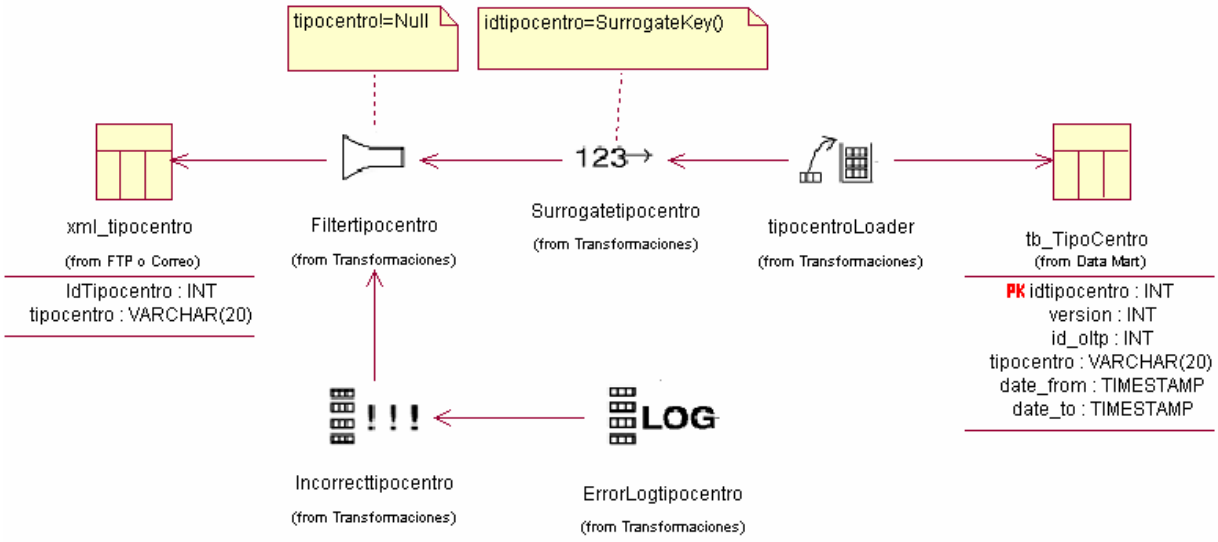


Figura 3. 80 Proceso ETL correspondiente a la dimensión: Dim_TipoCentro

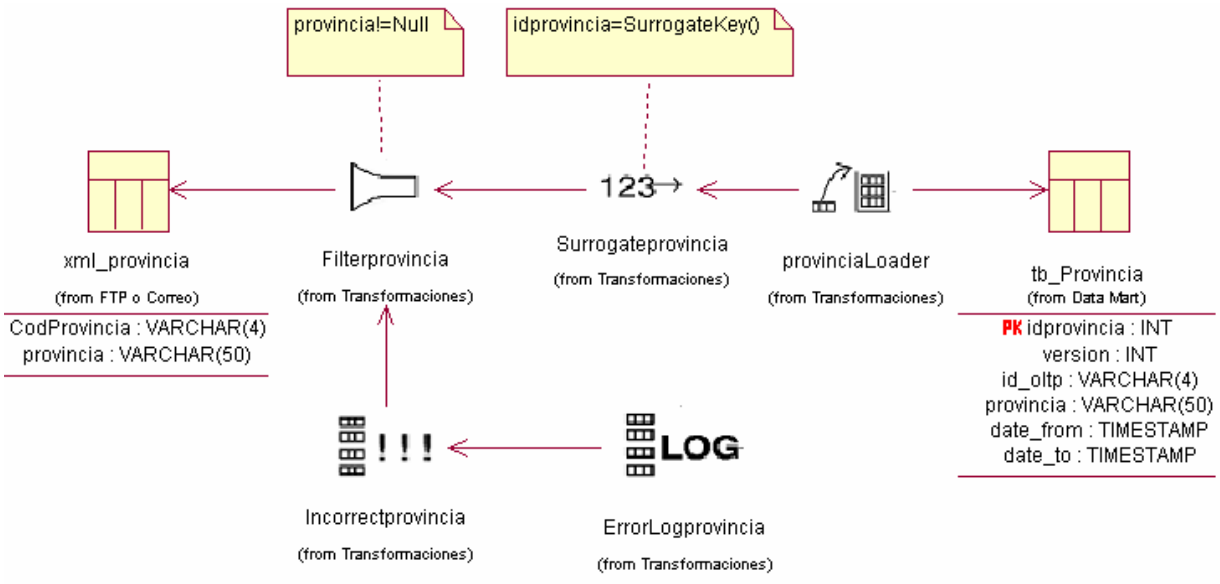


Figura 3. 81 Proceso ETL correspondiente a la dimensión: Dim_Provincia

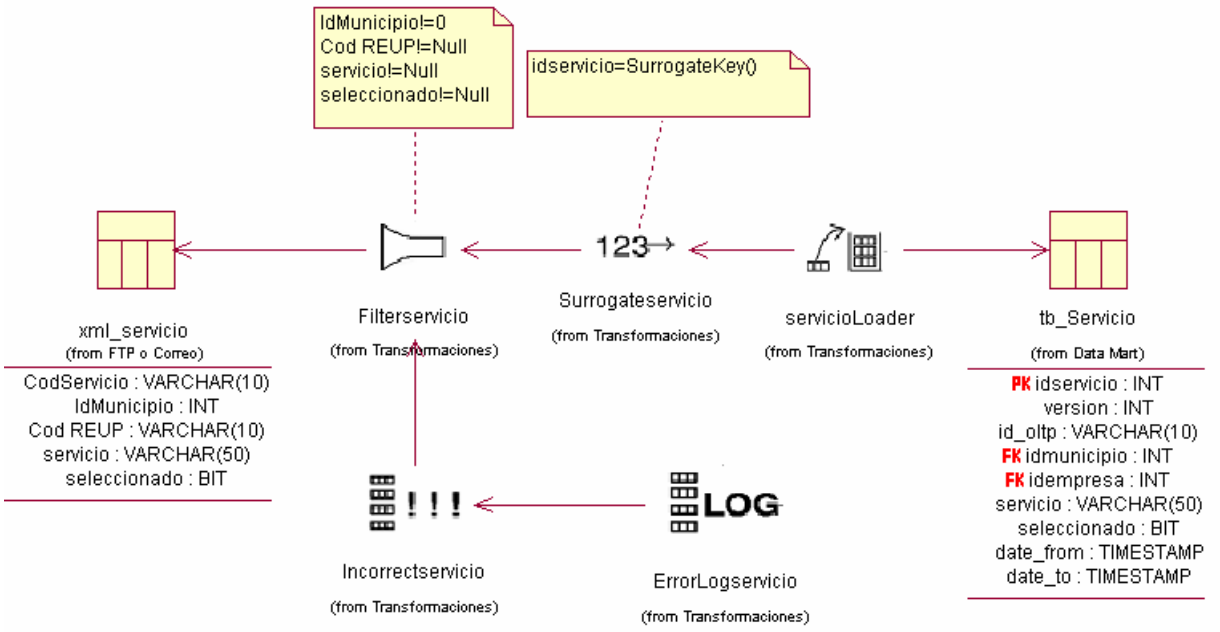


Figura 3. 82 Proceso ETL correspondiente a la dimensión: Dim_Servicio

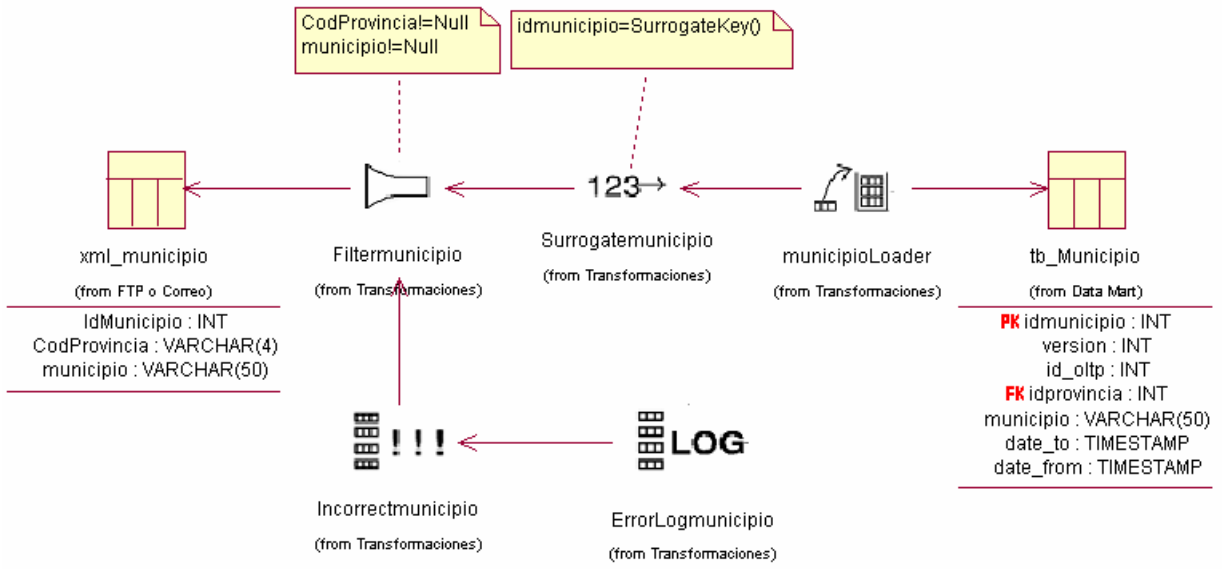


Figura 3. 83 Proceso ETL correspondiente a la dimensión: Dim_Municipio

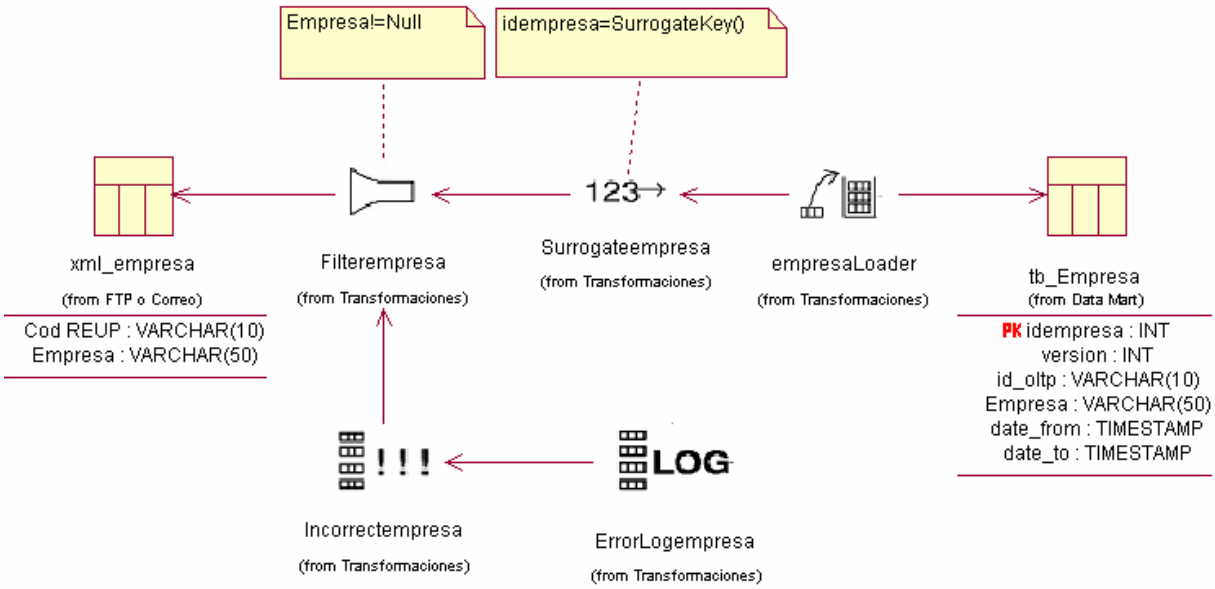


Figura 3. 84 Proceso ETL correspondiente a la dimensión: Dim_Empresa

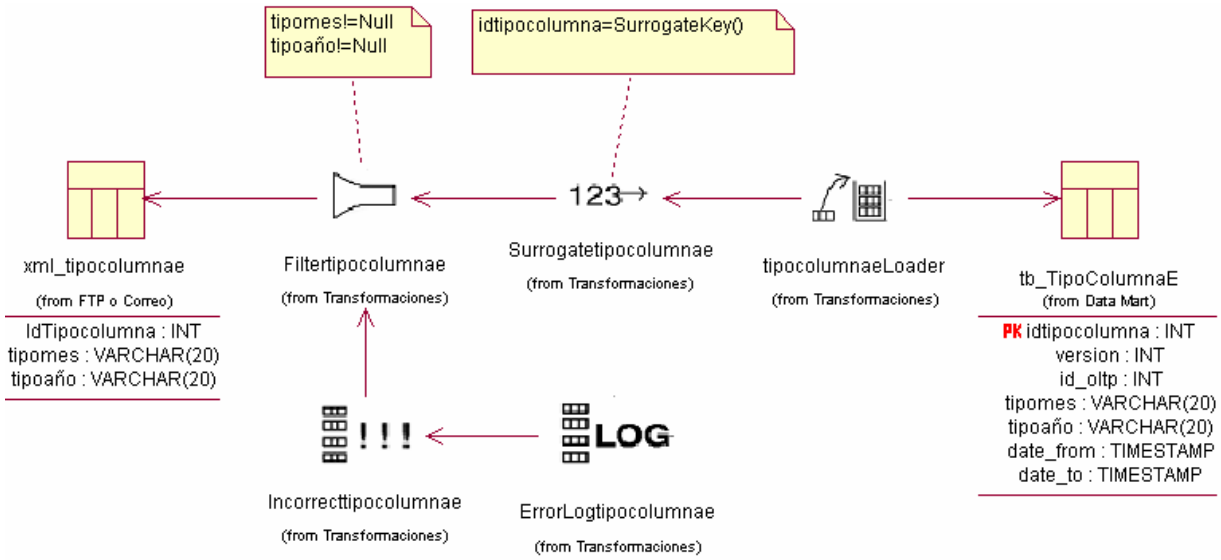


Figura 3. 85 Proceso ETL correspondiente a la dimensión: Dim_TipoColumnaE

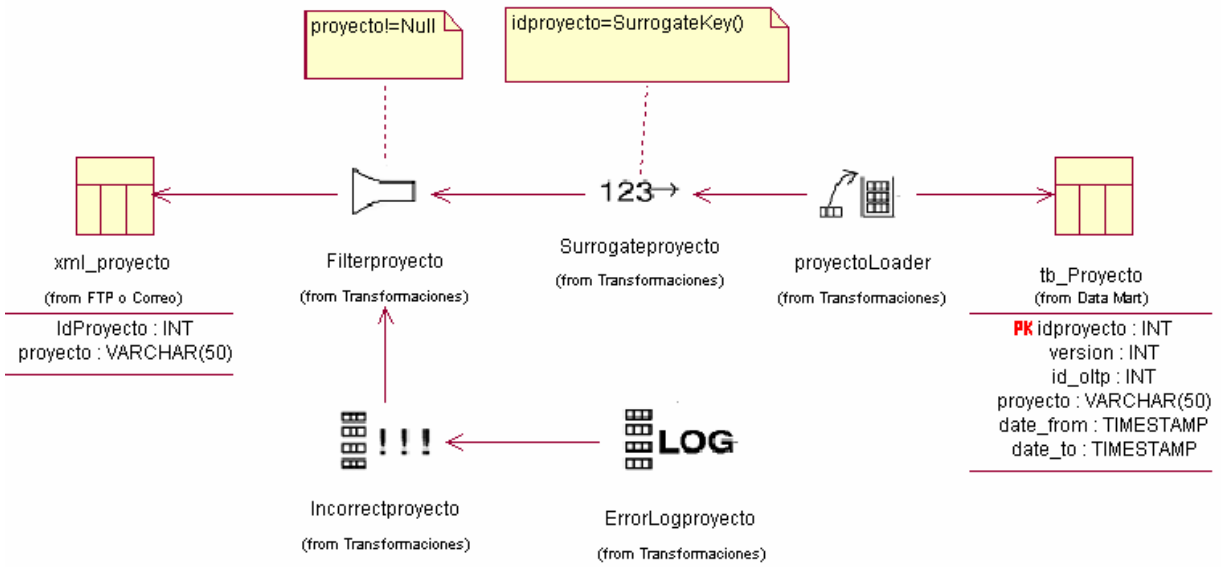


Figura 3. 86 Proceso ETL correspondiente a la dimensión: Dim_Proyecto

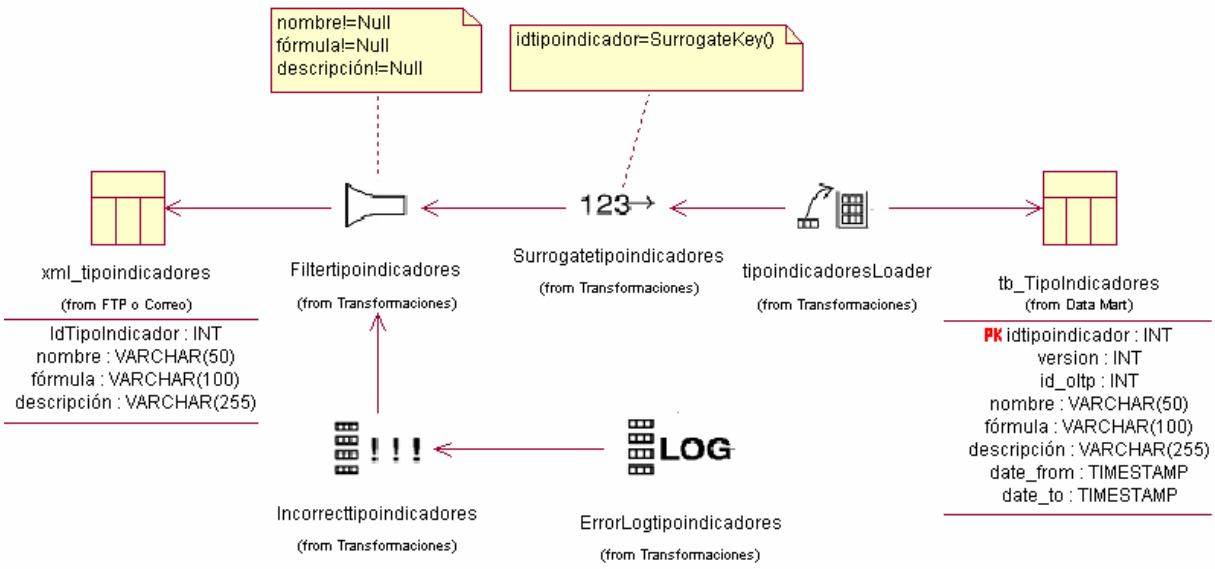


Figura 3. 87 Proceso ETL correspondiente a la dimensión: Dim_TipoIndicadores

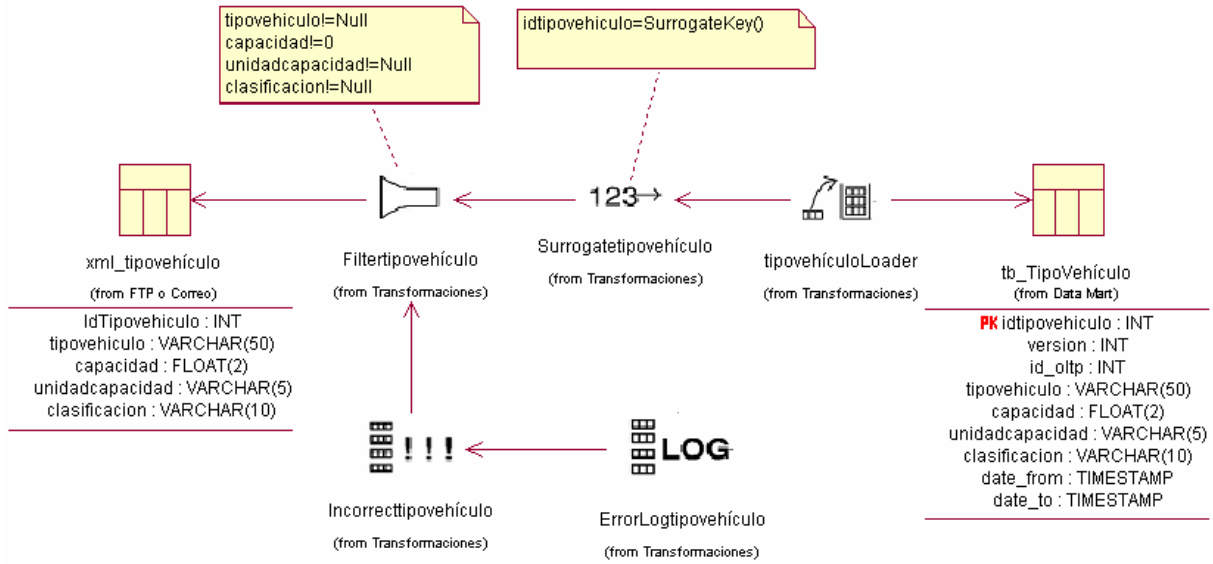


Figura 3. 88 Proceso ETL correspondiente a la dimensión: Dim_TipoVehículo

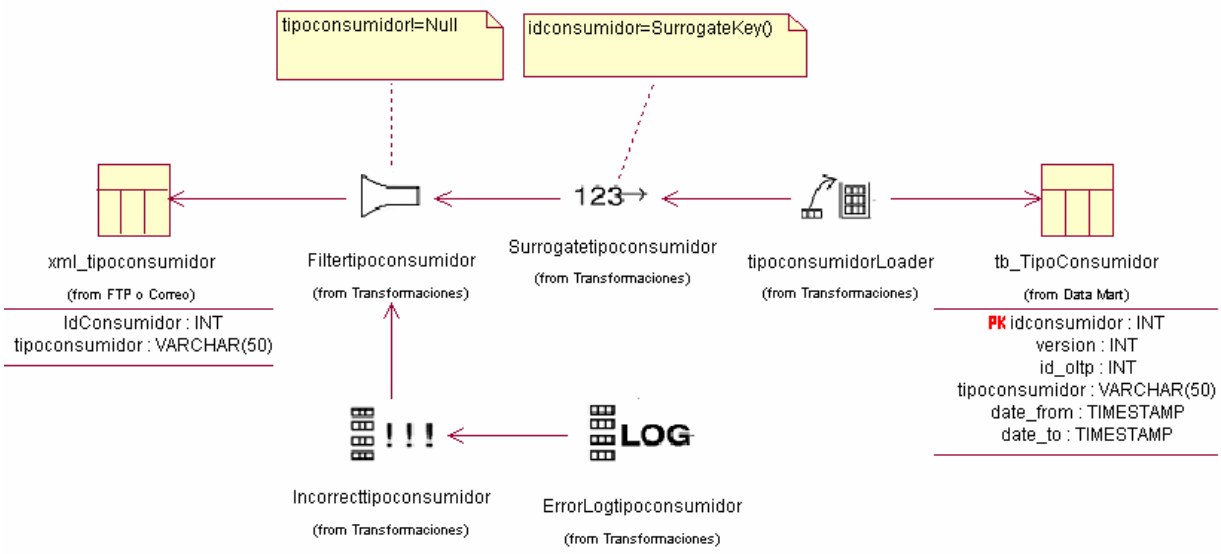


Figura 3. 89 Proceso ETL correspondiente a la dimensión: Dim_TipoConsumidor

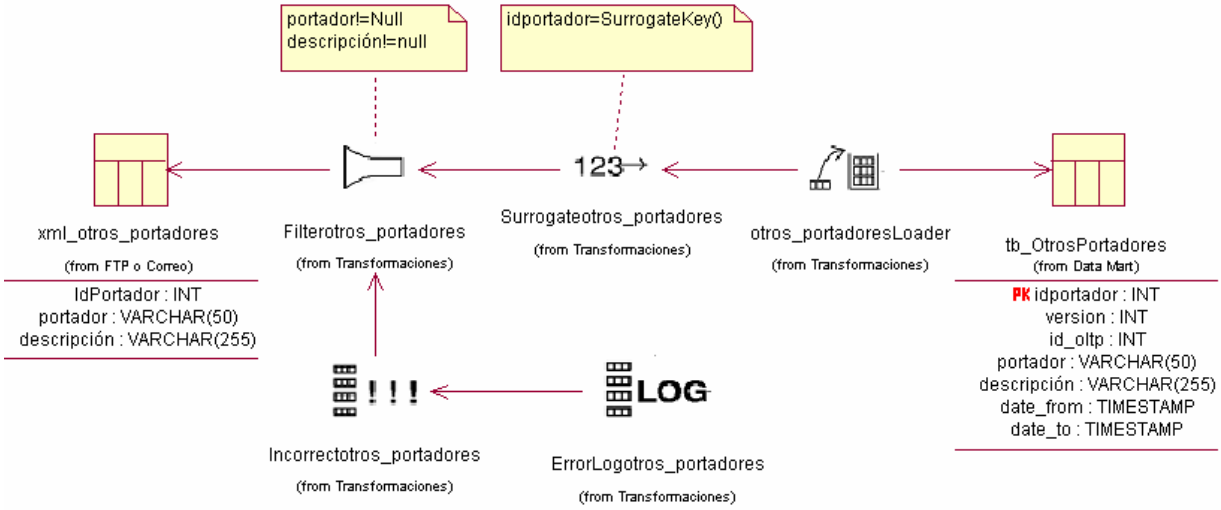


Figura 3. 90 Proceso ETL correspondiente a la dimensión: Dim_OtrosPortadores

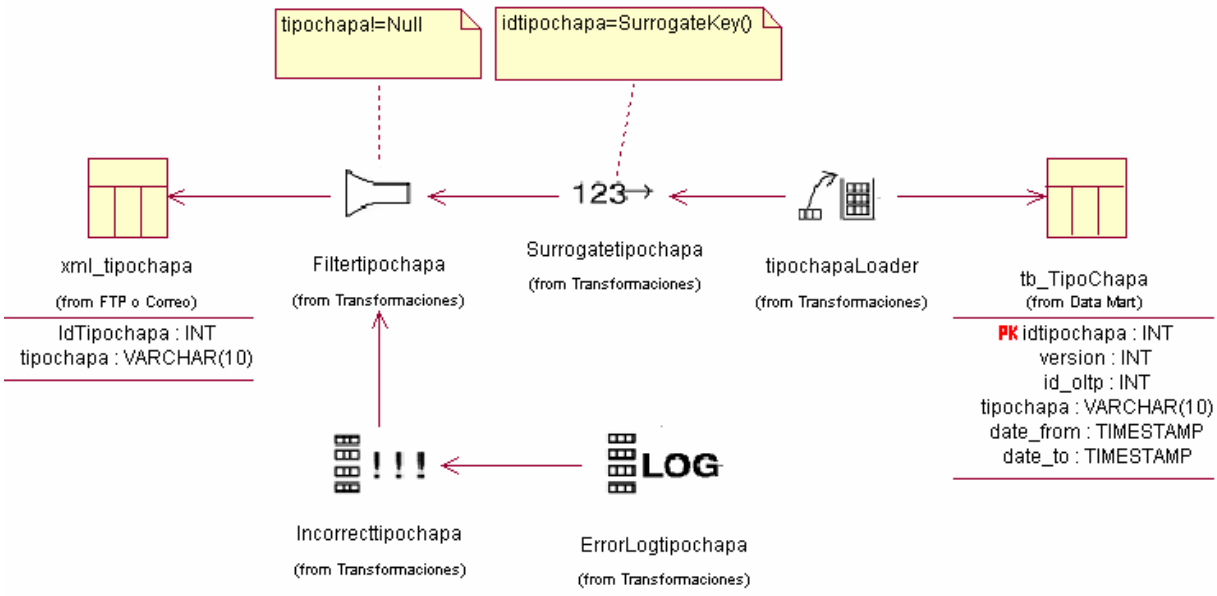


Figura 3. 91 Proceso ETL correspondiente a la dimensión: Dim_TipoChapa

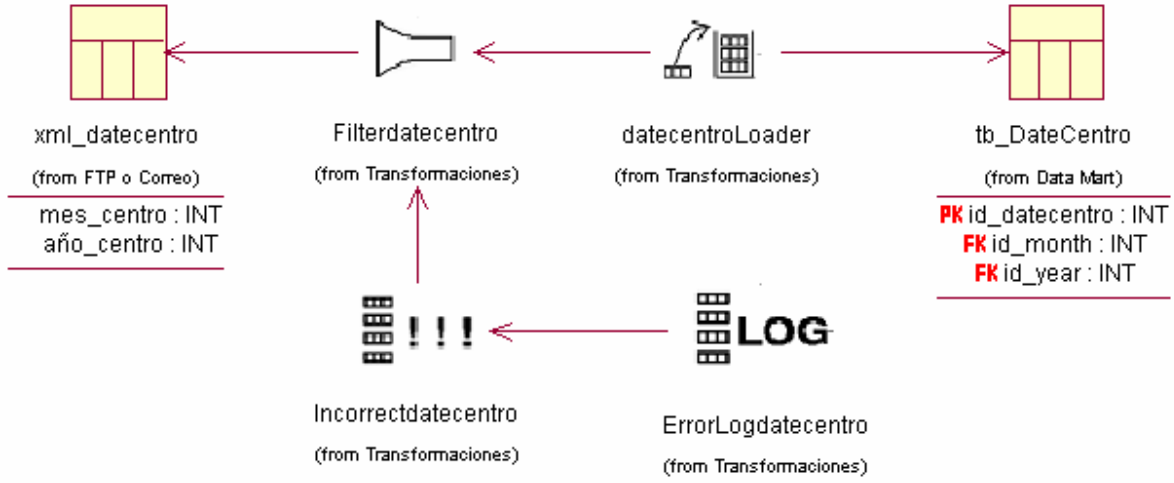


Figura 3. 92 Proceso ETL correspondiente a la dimensión: Dim_DateCentro

Hechos

Los procesos ETL asociados a la carga de los hechos, de forma general, se componen de los siguientes elementos:

1. Fuente de datos (FTP, correo).
2. *Filter*: se utiliza para evitar la carga de registros incompletos o indeseados.
3. *Incorrect*: se utiliza para redirigir los registros que no cumplieron con las restricciones establecidas en el filtro.
4. *Log*: se utiliza para registrar los errores: registros incompletos o indeseados.
5. *Loader*: en este elemento se realiza el mapeo de las llaves provenientes del sistema operacional con las nuevas llaves generadas (*Surrogate Key*) para las dimensiones y finalmente se cargan los datos en el Datamart.
6. Destino de los datos (tablas del Datamart).

A continuación, se muestran los diagramas de los procesos ETL correspondientes a las tablas de hechos (Figuras 3.93, 3.94, 3.95, 3.96, 3.97, 3.98, 3.99, 3.100, 3.101, 3.102, 3.103, 3.104 3.105):

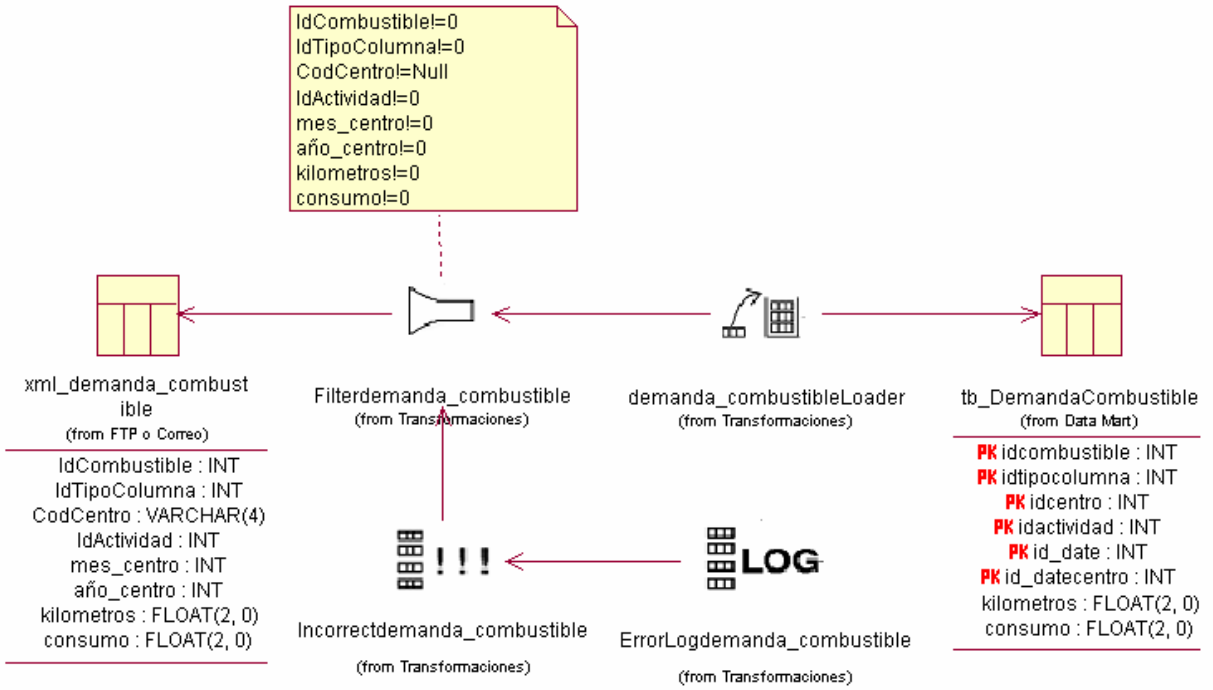


Figura 3. 93 Proceso ETL correspondiente al hecho: Fact_DemandaCombustible

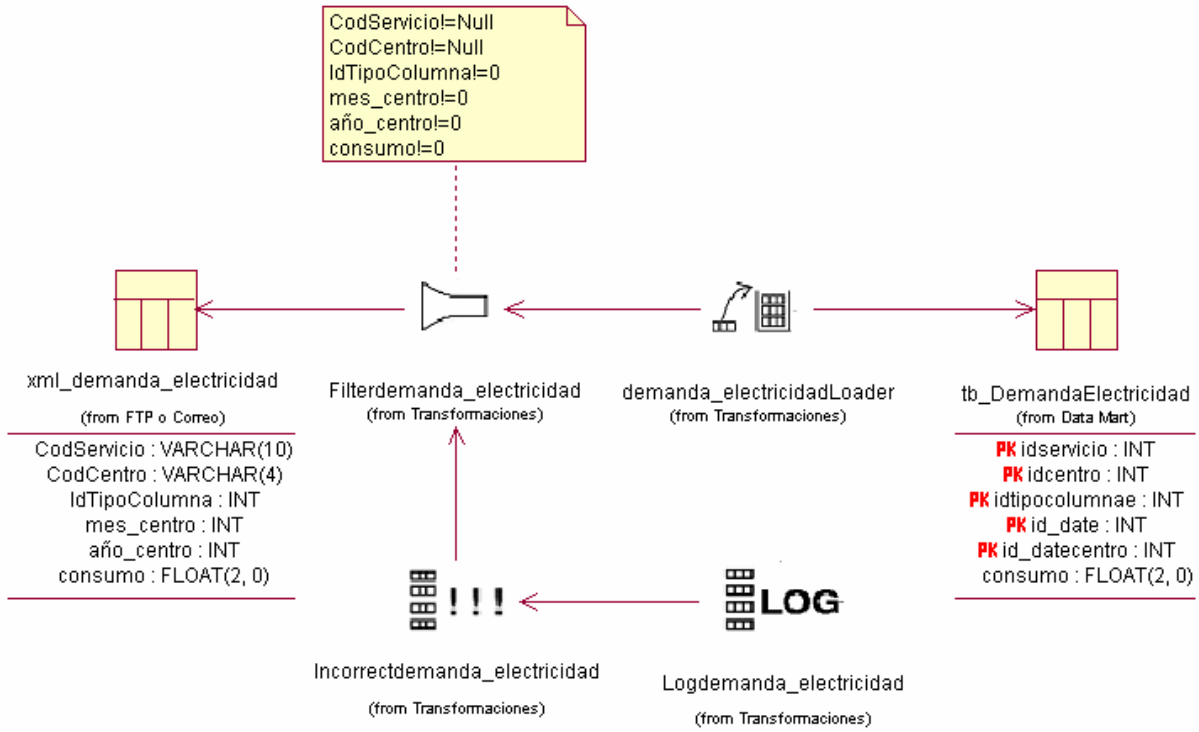


Figura 3. 94 Proceso ETL correspondiente al hecho: Fact_DemandaElectricidad

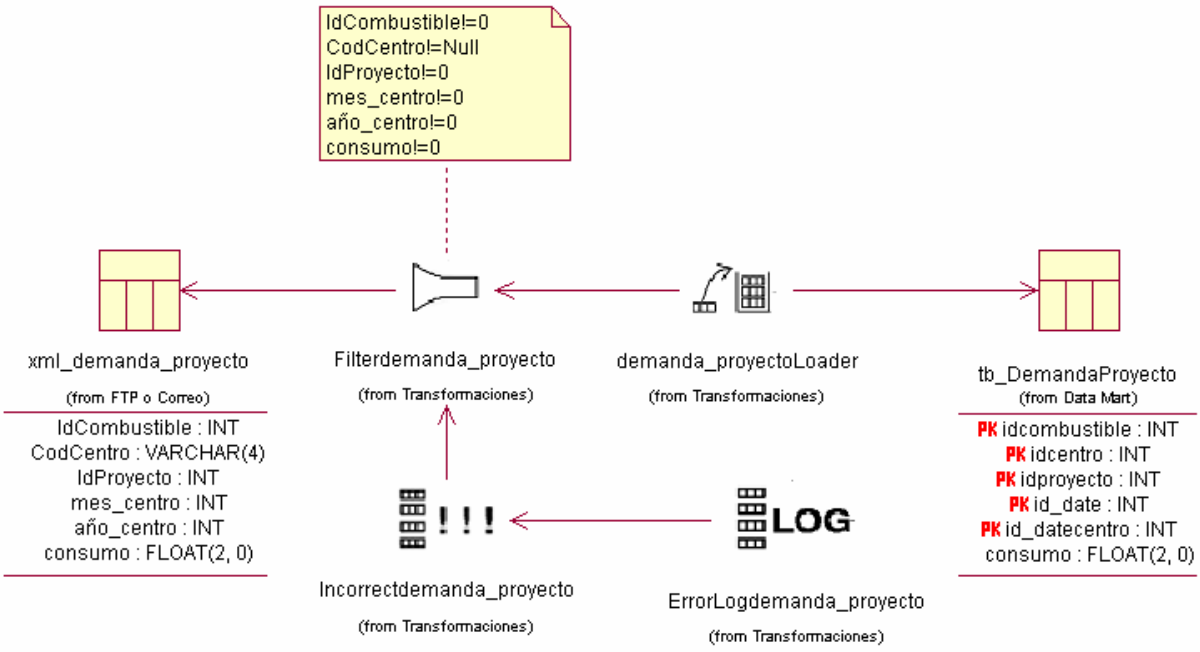


Figura 3. 95 Proceso ETL correspondiente al hecho: Fact_DemandaProyecto

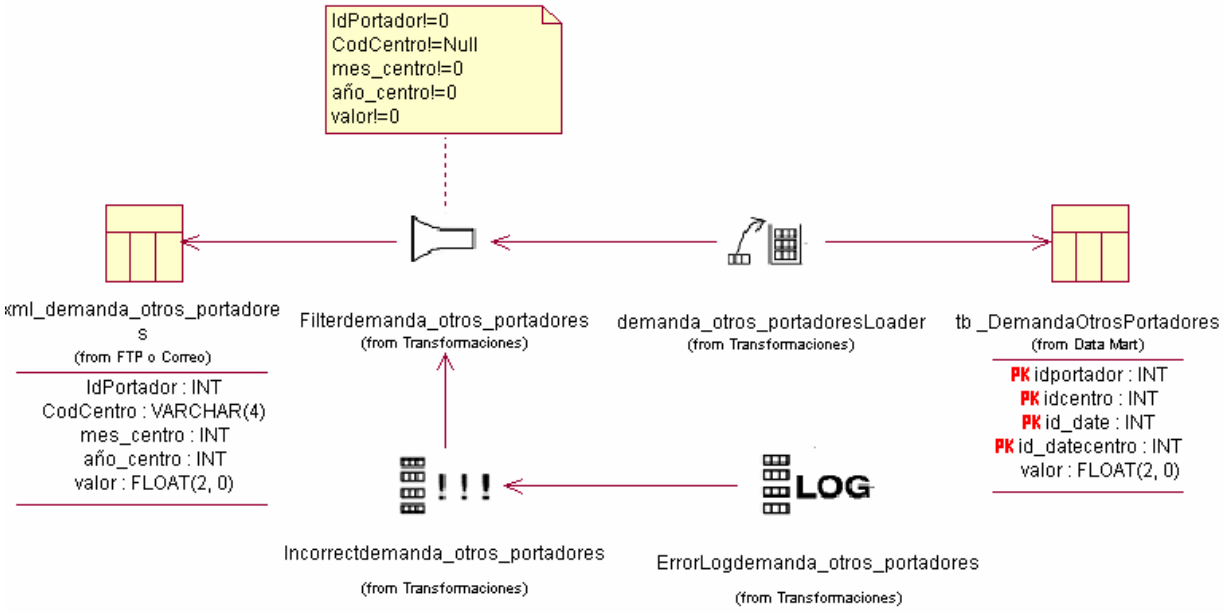


Figura 3. 96 Proceso ETL correspondiente al hecho: Fact _DemandaOtrosPortadores

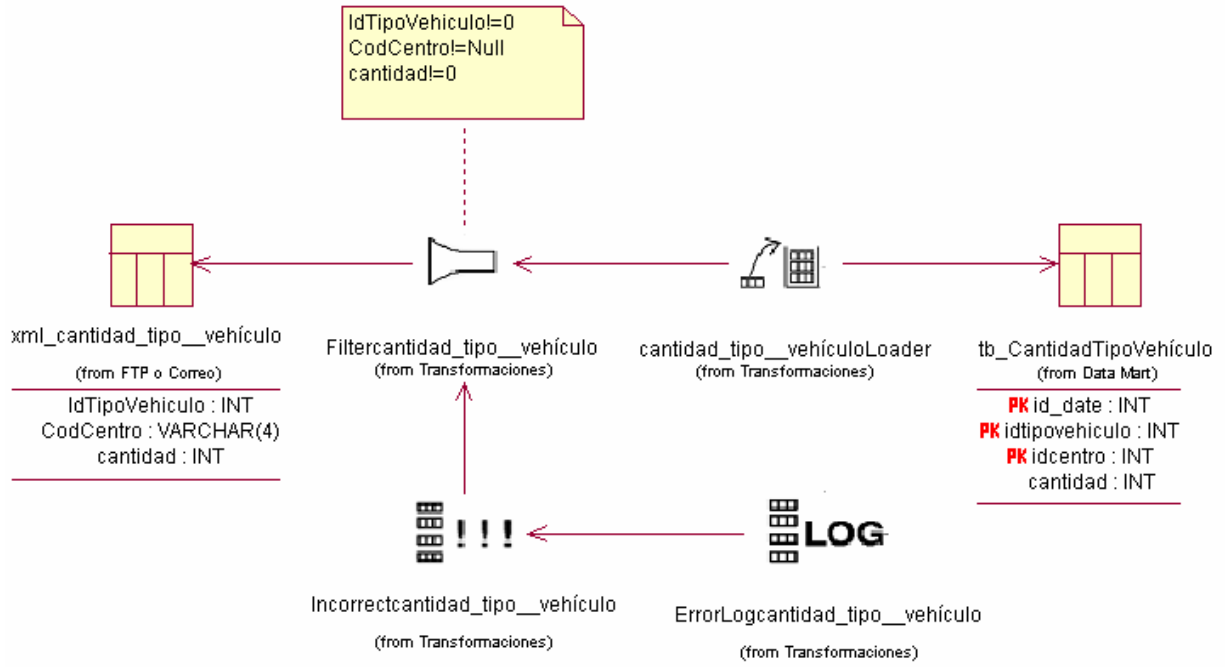


Figura 3. 97 Proceso ETL correspondiente al hecho: Fact _CantidadTipoVehículo

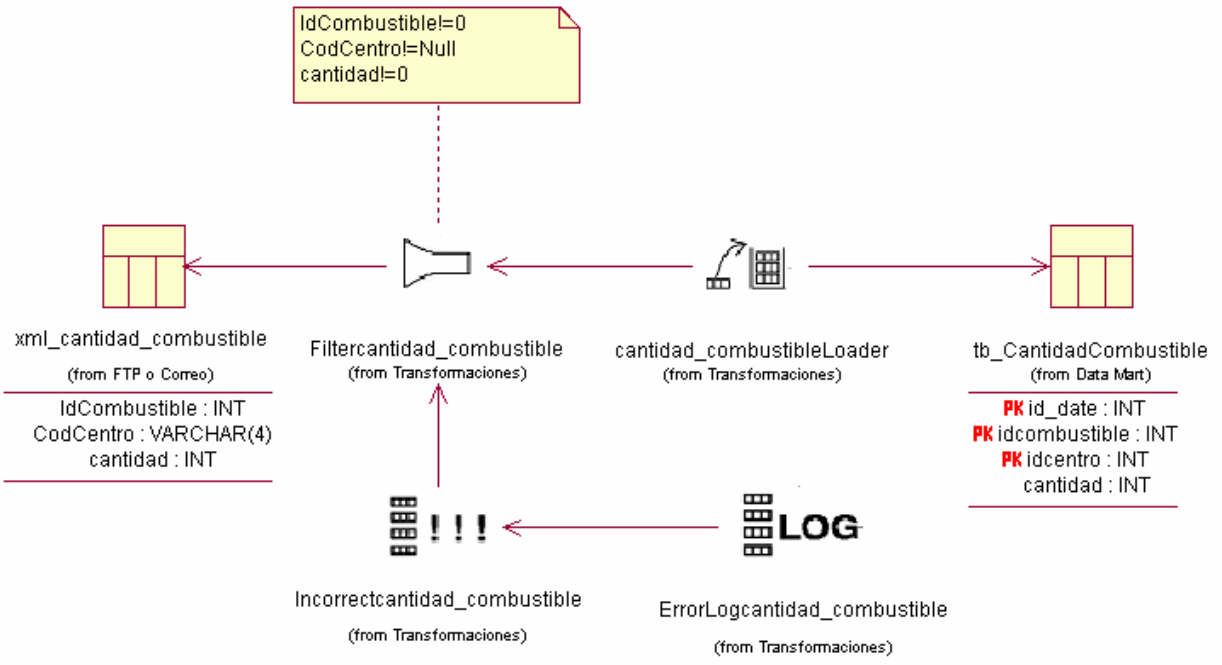


Figura 3. 98 Proceso ETL correspondiente al hecho: Fact _CantidadCombustible

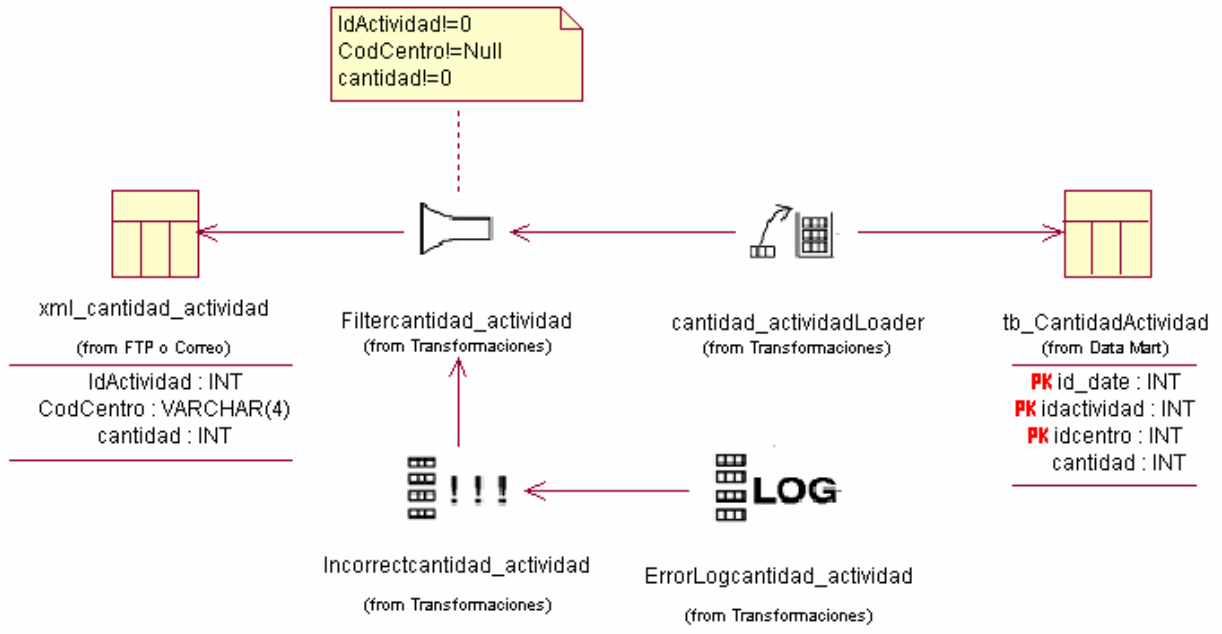


Figura 3. 99 Proceso ETL correspondiente al hecho: Fact _CantidadActividad

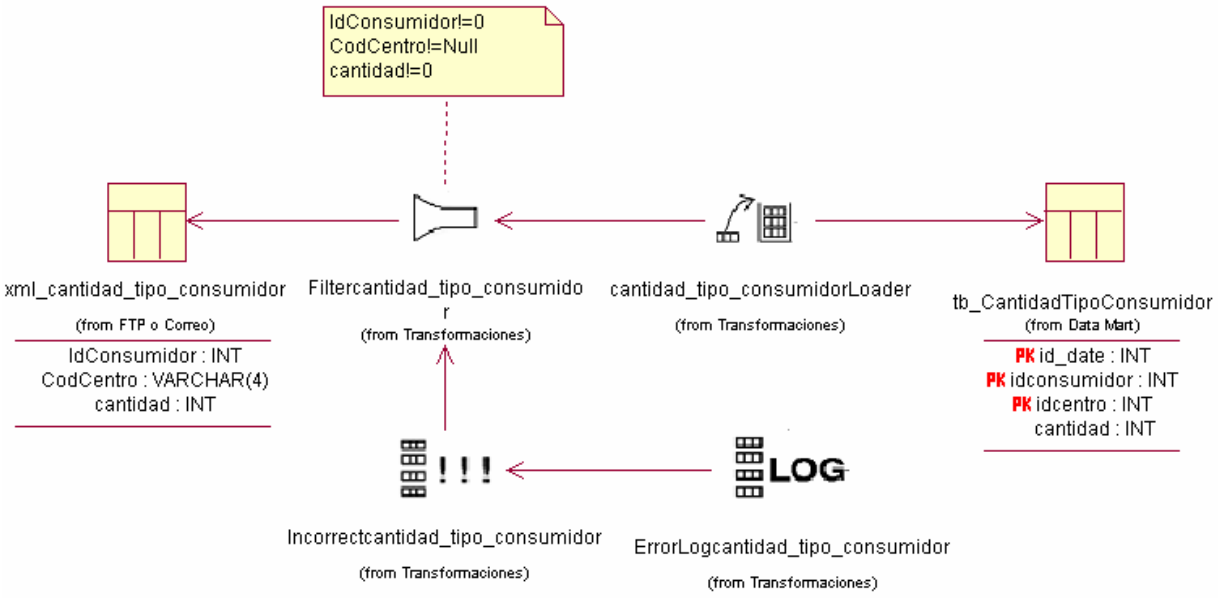


Figura 3. 100 Proceso ETL correspondiente al hecho: Fact _CantidadTipoConsumidor

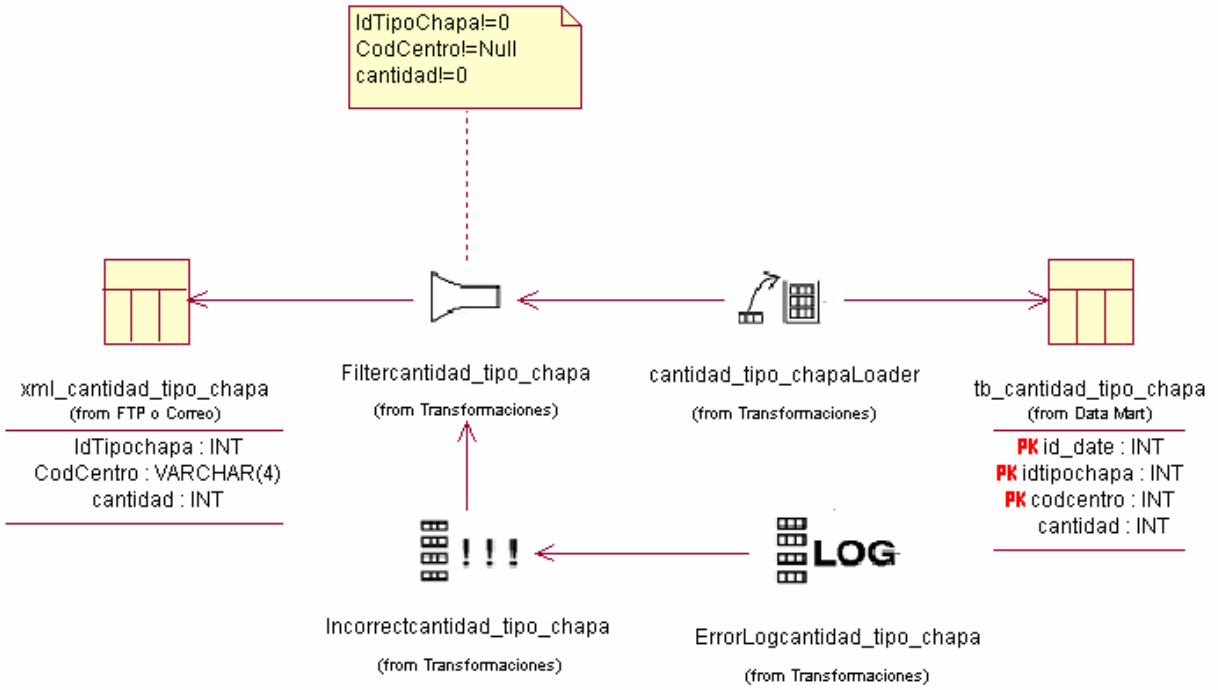


Figura 3. 101 Proceso ETL correspondiente al hecho: Fact _CantidadTipoChapa

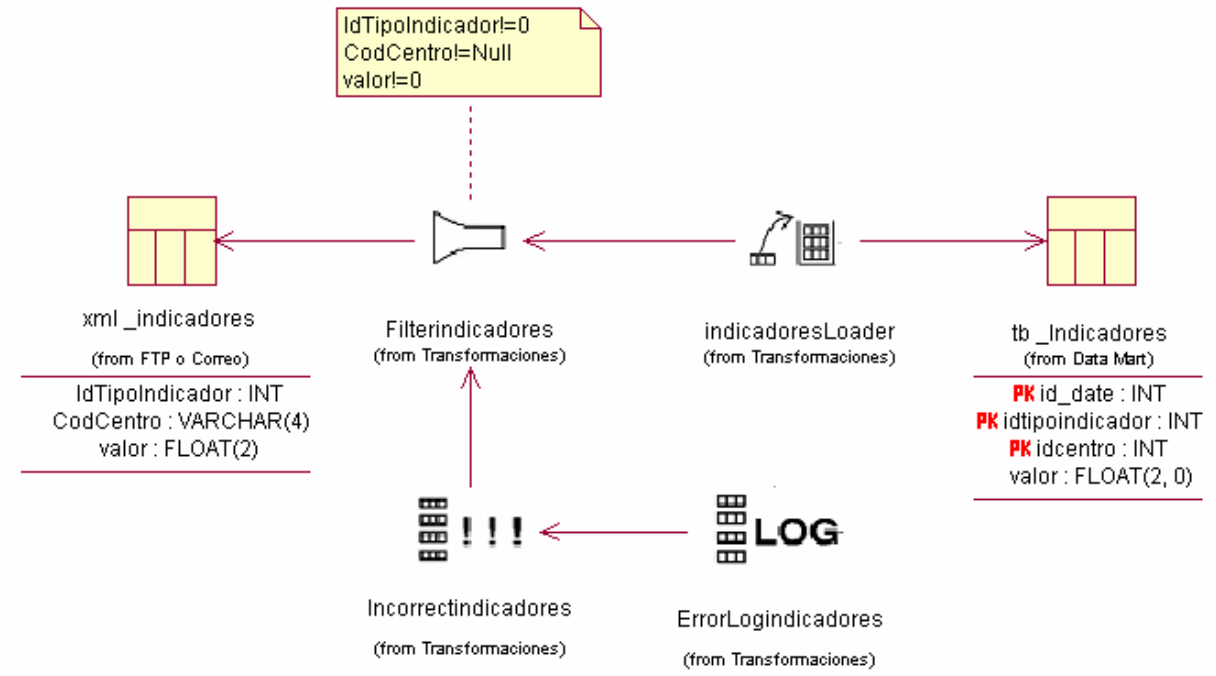


Figura 3. 102 Proceso ETL correspondiente al hecho: Fact _Indicadores

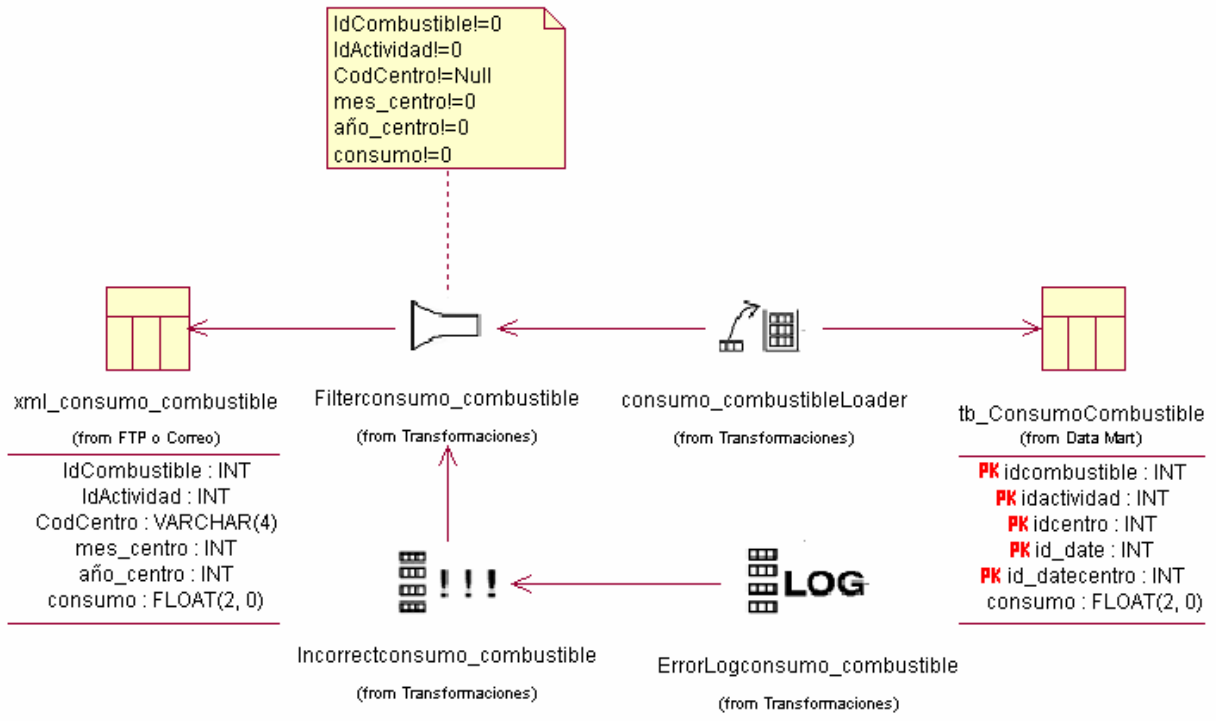


Figura 3. 103 Proceso ETL correspondiente al hecho: Fact _ConsumoCombustible

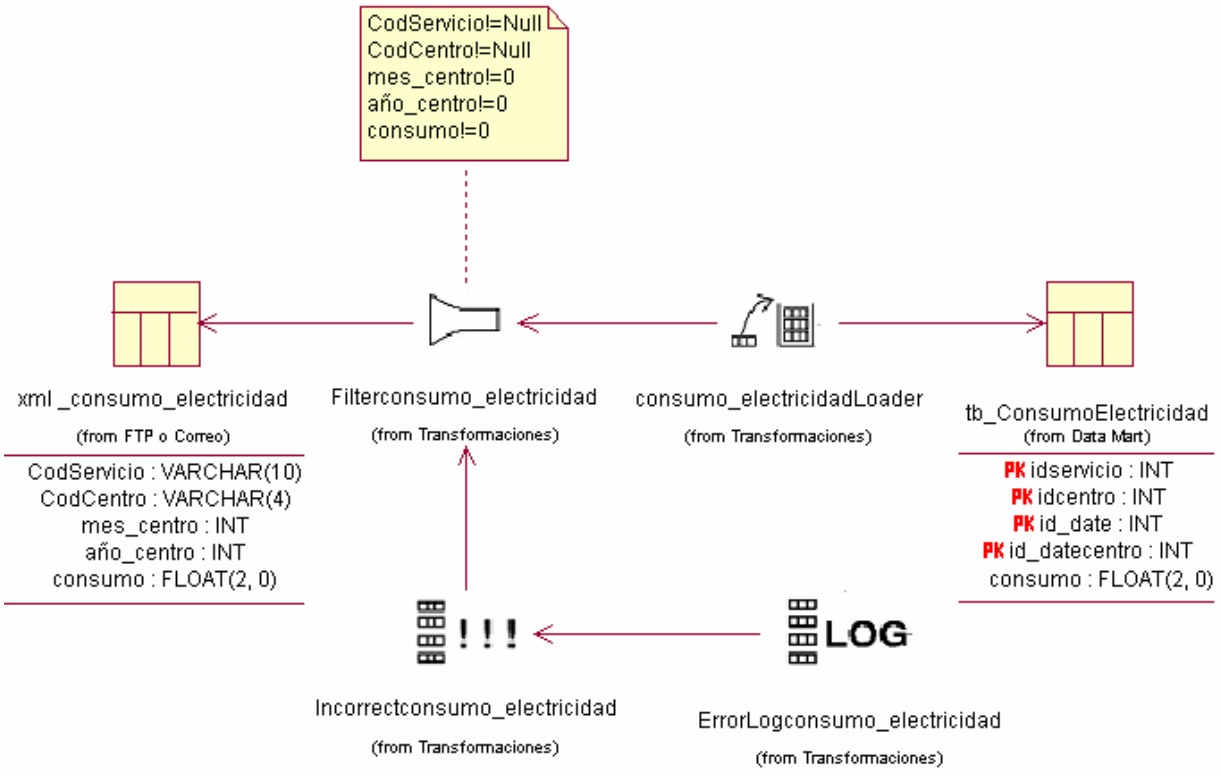


Figura 3. 104 Proceso ETL correspondiente al hecho: Fact _ConsumoElectricidad

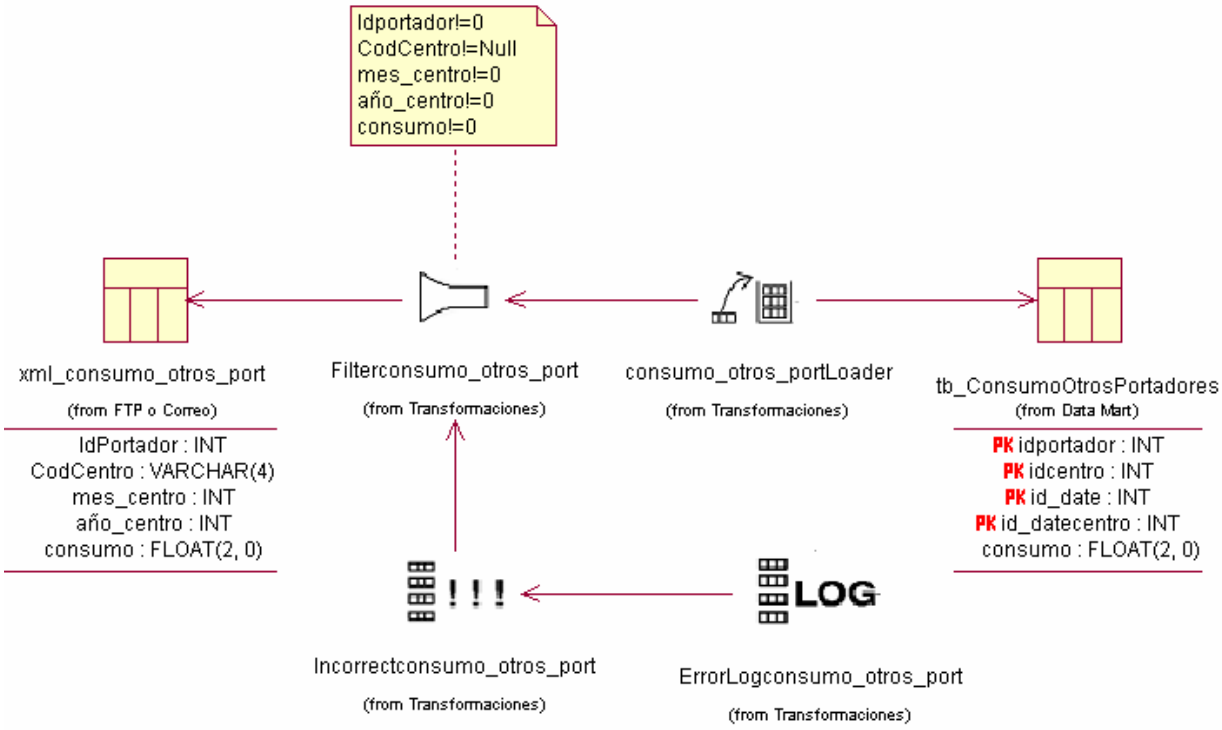


Figura 3. 105 Proceso ETL correspondiente al hecho: Fact _ConsumoOtrosPortadores

3.2.3 Modelo Físico

Para la descripción del modelo físico del Datamart primeramente debe determinarse la estructura del repositorio de datos y en consecuencia definir aspectos referentes al despliegue del sistema como: la distribución de los componentes y la identificación de los elementos de software y hardware. Finalmente, de acuerdo a estas consideraciones debe de construirse el diagrama de componentes y el diagrama de despliegue del sistema.

A continuación, se listan las tablas, con sus correspondientes campos (los campos subrayados representan la llave primaria), que conforman el Datamart:

Tablas de Dimensiones

1. tb_Combustible(idcombustible, version, id_oltp, date_from, date_to, nombre, unidadmedida, tirodirecto)
2. tb_TipoColumnaC(idtipocolumna, version, id_oltp, date_from, date_to, tipocolumna)
3. tb_ActividadG(idactividad, version, id_oltp, date_from, date_to, actividad, unidadmedida)
4. tb_Actividad(idactividad, version, id_oltp, date_from, date_to, actividad, unidadmedida)
5. tb_Centro(idcentro, version, id_oltp, date_from, date_to, nombre, rector_director, subordinado, idtipocentro, idprovincia)
6. tb_TipoCentro (idtipocentro, version, id_oltp, date_from, date_to, tipocentro)
7. tb_Provincia(idprovincia, version, id_oltp, date_from, date_to, provincia)
8. tb_Servicio(idservicio, version, id_oltp, date_from, date_to, servicio, seleccionado, idmunicipio, idempresa)
9. tb_Municipio (idmunicipio, version, id_oltp, date_from, date_to, municipio, idprovincia)
10. tb_Empresa(idempresa, version, id_oltp, date_from, date_to, Empresa)
11. tb_TipoColumnaE(idtipocolumna, version, id_oltp, date_from, date_to, tipomes, tipoaño)
12. tb_Proyecto(idproyecto, version, id_oltp, date_from, date_to, proyecto)
13. tb_TipoIndicadores(idtipoindicador, version, id_oltp, date_from, date_to, nombre, fórmula, descripción)
14. tb_TipoVehículo(idtipovehiculo, version, id_oltp, tipovehiculo, capacidad, unidadcapacidad, clasificacion, date_from, date_to)
15. tb_TipoConsumidor(idconsumidor, version, id_oltp, date_from, date_to, tipoconsumidor)
16. tb_OtrosPortadores(idportador, version, id_oltp, date_from, date_to portador, descripción)
17. tb_TipoChapa(idtipochapa, version, id_oltp, date_from, date_to, tipochapa)
18. tb_Moneda(idmoneda, , version, id_oltp, date_from, date_to, moneda, simbolo)
19. tb_DateCentro(id_datecentro, id_month, id_year)
20. tb_Date(id_date, id_day, id_month, id_year)
21. tb_day(id_day)
22. tb_month(id_month, name)
23. tb_year(id_year, description)

Tablas de Hechos

1. tb_DemandaCombustible(idcombustible, idtipocolumna, idcentro, idactividad, id_datecentro, id_date, kilometros, consumo)
2. tb_DemandaElectricidad(idservicio, idcentro, idtipocolumnae, id_datecentro, id_date, consumo)
3. tb_DemandaProyecto(idcombustible, idcentro, idproyecto, id_datecentro, id_datecentro, id_date, consumo)
4. tb_DemandaOtrosPortadores(idportador, idcentro, id_datecentro, id_date, valor)

5. tb_Indicadores (idtipoindicador, idcentro, id_date, valor)
6. tb_CantidadTipoVehículo(idtipovehiculo, idcentro , id_date, cantidad)
7. tb_CantidadCombustible (idcombustible, idcentro, id_date, cantidad)
8. tb_CantidadActividad (idactividad, idcentro, id_date, cantidad)
9. tb_CantidadTipoConsumidor(idconsumidor, idcentro, id_date, cantidad)
10. tb_CantidadTipoChapa(idtipochapa, idcentro, id_date, cantidad)
11. tb_PlanOperativoCombustible(idcombustible, idcentro, idtipochapa, idmoneda, id_datecentro, id_date, valor)
12. tb_PlanOperativoElectricidad(idservicio, idcentro, id_datecentro, id_date, valor)
13. tb_PlanOperativoOtrosPortadores(idportador, idcentro, idmoneda, id_datecentro, id_date, valor)
14. tb_ConsumoCombustible(idcombustible, idactividad, idcentro, id_datecentro, id_date, consumo)
15. tb_ConsumoElectricidad(idservicio, idcentro, id_datecentro, id_date, consumo)
16. tb_ConsumoOtrosPortadores(idportador, idcentro, id_datecentro, id_date, consumo)

3.2.3.1 Diagrama de Componentes

Un componente en UML es una parte de un sistema (ficheros de configuración, JavaBeans, Servlet, interfaces, bases de datos, tablas, etc.) que encapsula implementación, brinda una interfaz de comunicación y puede contener otros componentes. **[Luján, 2005]**

En un diagrama los componentes se encuentran conectados por medio de relaciones de dependencia. Los componentes más empleados en este tipo de soluciones posee como estereotipo: Database, Tablespace y Table. Las relaciones de dependencia que los relacionan poseen como estereotipo: reside (Figuras 3.106 y 3.107).

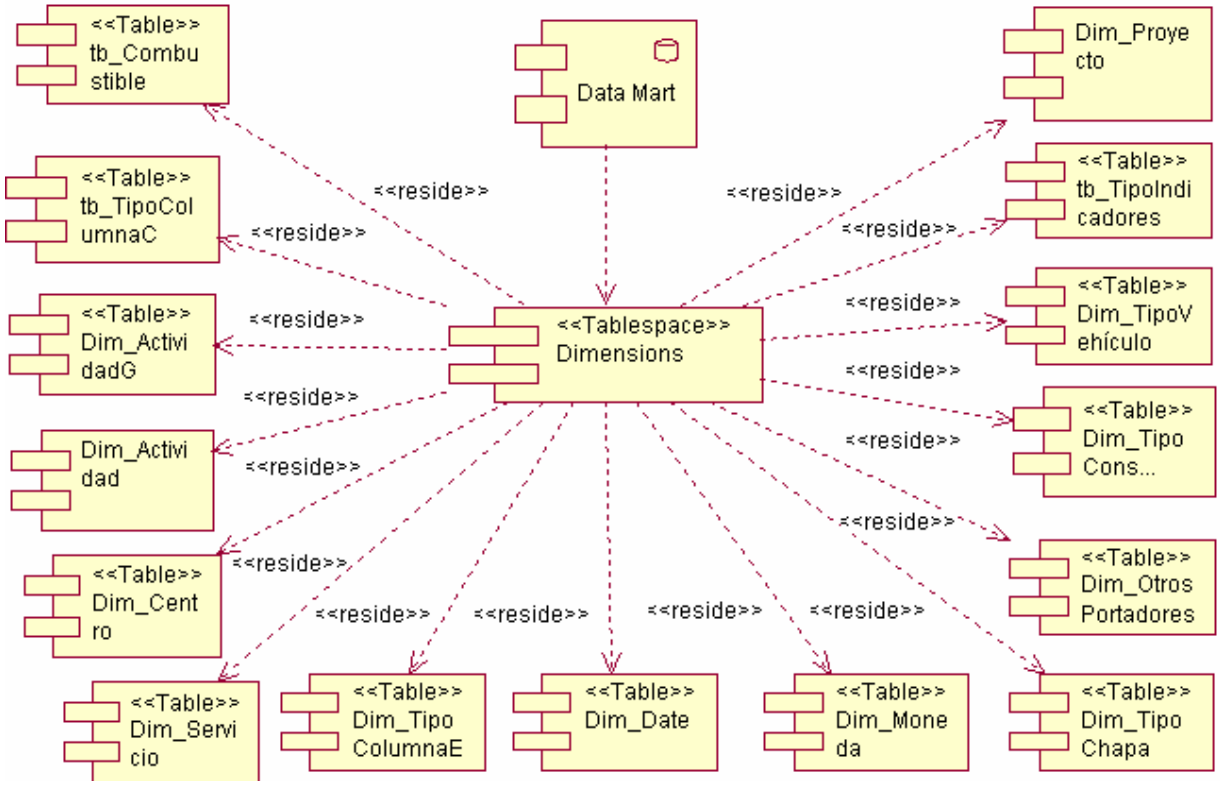


Figura 3. 106 Diagrama de Componentes (Dimensiones).

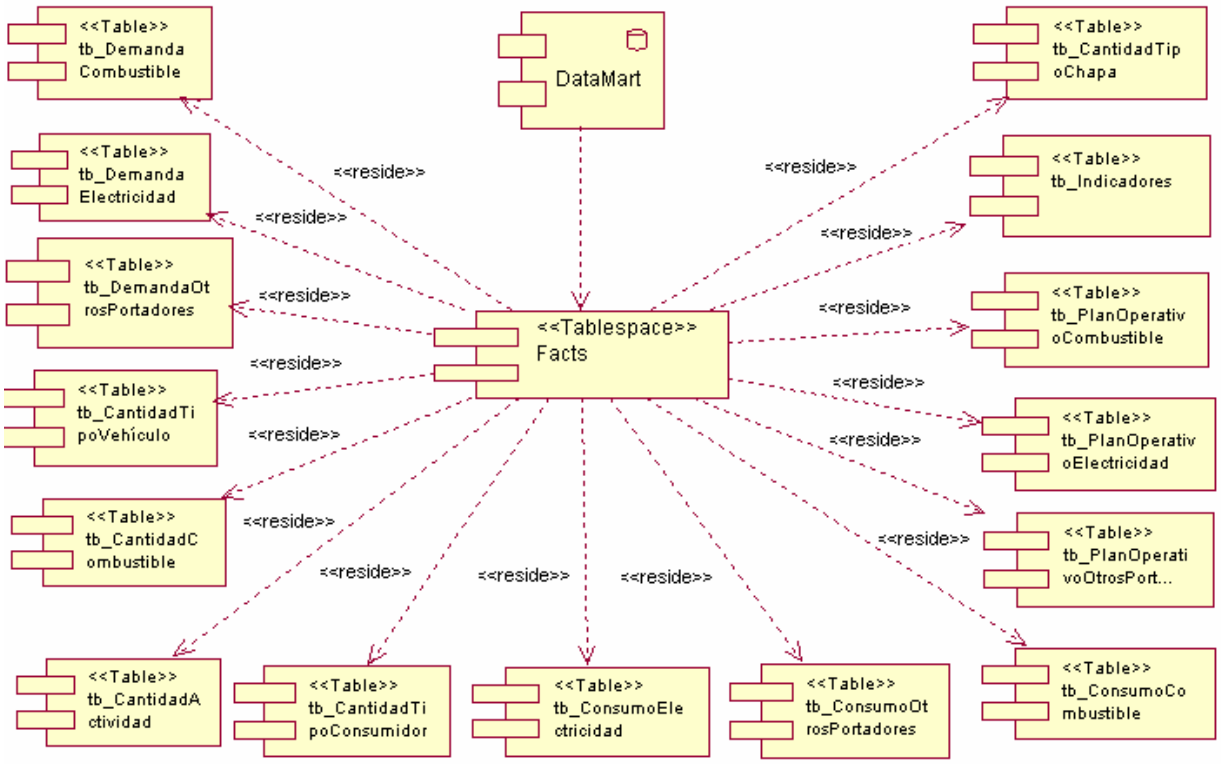


Figura 3. 107 Diagrama de Componentes (Hechos).

3.2.3.2 Diagrama de Despliegue

Un diagrama de despliegue en UML es un artefacto que muestra las características de distribución de los diferentes componentes de la aplicación y los sistemas legados asociados, en tiempo de ejecución. Los elementos involucrados, a los que se les denomina nodos, pueden representar elementos de hardware o de software (Web Server, Databases, etc.) y se relacionan en consecuencia con la comunicación que existe entre ellos. **[Luján, 2005]**

(Figura 3.108)

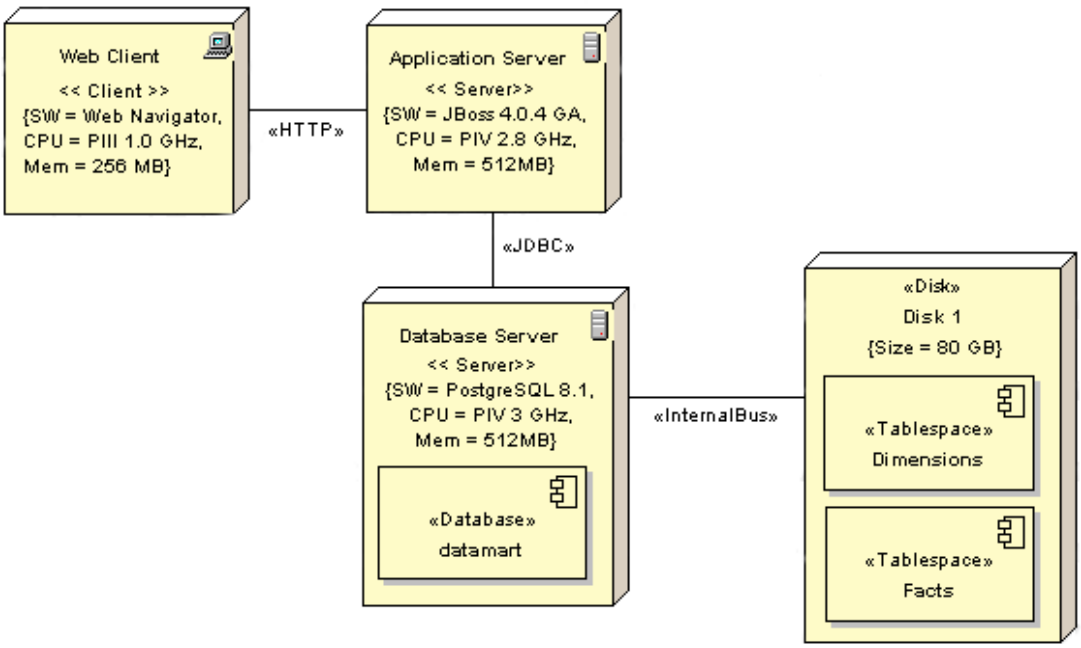


Figura 3. 108 Diagrama de Despliegue del Sistema.

La arquitectura del sistema propuesto deberá estar constituida por varios nodos clientes (Web Client) con independencia de la plataforma (Windows, Linux, etc.). Estos solo necesitan estar conectados a la red LAN y usar un navegador Web (Mozilla, Internet Explorer, etc.) para realizar las peticiones al nodo del servidor de aplicaciones (JBoss 4.0.4, Application Server). Este intercambio se realiza a través del protocolo HTTP.

El servidor de aplicaciones establece comunicación con el nodo servidor de datos (PostgreSQL 8.1, Database Server) para acceder a la información almacenada en el DW (principalmente consultas). La aplicación desplegada en este nodo estará escrita en Java por lo que es independiente de la plataforma.

3.2.3.2 Diagrama de Despliegue

3.3 Beneficios Tangibles e Intangibles

El desarrollo de todo sistema implica un costo, y la decisión de acometer el proyecto depende en gran medida de los beneficios, económicos y sociales o tangibles e intangibles, que se obtengan con los resultados del producto.

El sistema propuesto presenta un valor práctico bien definido, ya que está orientado a dotar al MES de una herramienta de apoyo a la toma de decisiones estratégicas. Los resultados permitirán el monitoreo de las áreas de los portadores energéticos y el transporte. Este monitoreo, ayudará a establecer un proceso de retroalimentación permitiendo conocer las principales fortalezas y debilidades en estos procesos y en qué sentido deben estar encaminados los esfuerzos. Esfuerzos que podrían verse reflejados en el perfeccionamiento continuo de dichos procesos.

En el caso particular de esta propuesta, los beneficios identificados son intangibles, debido a que no constituye un medio de ingreso económico para la entidad. Los beneficios reportados son difíciles de medir, debido a que se necesitaría analizar el valor de la información brindada (consistente y en corto tiempo) y determinar si su aprovechamiento provoca algún beneficio cuantificable.

A continuación, se listan los beneficios identificados:

- Provee al MES, de una herramienta para la recolección, preparación, y procesamiento de la información recogida en las áreas de los portadores energéticos y el transporte de todos los CES.
- Los resultados del análisis constituyen una base confiable para los encargados de dirigir los procesos vinculados a las áreas económicas y de aseguramiento de los CES, otorgándole mayor poder de decisión.
- En cuanto al almacenamiento de los datos, se gana en centralización, seguridad y se disminuyen los errores e inconsistencias.
- Minimiza la pérdida de información, debido que el proceso de análisis y visualización de los resultados se realiza de forma automatizada.
- Permite un mayor aprovechamiento de la información recopilada.
- Permite mantener un histórico de los resultados de los CES.

- Se minimiza el tiempo perdido, por la espera de la información. Se optimizan los procesos.
- Elimina los gastos concernientes al pago de licencias, debido a que fue totalmente desarrollado sobre herramientas de la comunidad "*Open Source*".

Analizado detenidamente los beneficios que otorga la implantación del sistema, se llega a la conclusión, que es factible la creación del proyecto.

Conclusiones

En este capítulo se realizó la descripción del sistema propuesto para dar solución a la problemática planteada. Para ello se desarrollaron las etapas de construcción y despliegue de una arquitectura Data Warehouse, con el empleo de la metodología de diseño basada en UML y en el Proceso Unificado: *Data Warehouse Engineering Process* (DWEP).

La aplicación de la metodología DWEP demostró que constituye una solución factible acorde a la necesidad de existencia de un estándar para el modelado de una solución basada en Data Warehouse que abarque todas las etapas del diseño, implementación y mantenimiento.

La descripción de la solución basada en DWEP transitó por la construcción de varios modelos con diferentes niveles de detalle: modelo conceptual, modelo lógico y modelo físico. Esta estructura permitió ir analizando de forma incremental los requisitos a tener en cuenta para la implementación final del repositorio de datos.

Para finalizar se analizaron los beneficios que trae consigo la realización de un proyecto de este tipo y se llegó a la conclusión que tomando en cuenta el ahorro de tiempo y la confiabilidad de la información, el proyecto resulta factible.

CONCLUSIONES

De acuerdo a la propuesta planteada, como respuesta a la necesidad de contar con una herramienta de apoyo a la toma de decisiones, para monitorear el proceso de control de los portadores energéticos y el transporte en el MES, se diseñó una solución basada en arquitecturas de Inteligencia de Negocio.

La propuesta abarca: el diseño y la implementación de un Datamart, la implementación de los procesos de extracción, transformación y carga de datos de las diferentes fuentes operacionales y el empleo de técnicas de análisis y generación de reportes. De esta manera, se cumple con el objetivo principal de este trabajo y se concluye que el empleo de este tipo de solución favorece en gran medida, al control de estas áreas en el MES.

En cuanto a los resultados obtenidos es posible concluir lo siguiente:

- Se diseñó un modelo para la toma de decisiones en la esfera de portadores energéticos y el transporte del MES, donde se definió los procesos de extracción, transformación y carga.

- Se establecen las bases para la realización de estudios que permitan el descubrimiento del conocimiento oculto que servirá de apoyo a la toma de decisiones con niveles de objetividad muy superiores a los actuales.

- Se aplicaron técnicas de construcción y despliegue de la arquitectura Data Warehouse, empleado la metodología de diseño: *Data Warehouse Engineering Process* (DWEP) y se comprobó que constituye una poderosa ayuda para los desarrolladores. Esto se debe a que dicha metodología abarca todas las etapas del ciclo de vida de un proyecto de Data Warehouse y para su asimilación no se necesita estudiar ninguna nueva notación, ni flujos de trabajo a seguir, pues se basa en el Proceso Unificado y utiliza como notación una extensión de UML.

- La utilización de PostgreSQL o MySQL como gestor de bases de datos para el almacenamiento de los datos de un Data Warehouse, Kettle como herramienta ETL para las cargas periódicas de información y el procesamiento analítico a través del servidor OLAP Mondrian, brindan un marco de trabajo para la

realización de un proyecto de Data Warehouse con herramientas libres y de código abierto.

La solución propuesta contribuye al éxito del proceso de controlar las áreas de portadores energéticos y el transporte en el MES, favoreciendo la toma de decisiones estratégicas.

RECOMENDACIONES

- Ampliar el estudios sobre la Metodología de Diseño de Data Warehouse: *Data Warehouse Engineering Process* (DWEP) e incorporarla a otros trabajos en el MES.
- Concluir la implementación del *add-in* para Rational Rose, que brinda soporte al modelado de los artefactos propuestos en DWEP.
- Desarrollar un generador de reportes *ad-hoc*, que permita personalizar las consultas al repositorio de datos y de acuerdo a ello, permitir explotar al máximo la información almacenada.
- Realizar la implementación del sistema en el MES a través del OLAP como alternativa a la toma de decisiones de los directivos.
- Aplicar técnicas de minería de datos, con el objetivo de extraer conocimiento (reglas, tendencias, agrupamientos, etc.) a partir de los datos almacenados.
- Aplicar otras técnicas de análisis de la información como: los cuadros de mando, con el objetivo de agrupar información en entornos visuales que resulten más intuitivos para los usuarios.
- Realizar un estudio sobre los métodos de análisis de factibilidad existentes, que se ajusten a proyectos de *Data Warehousing*.

REFERENCIAS BIBLIOGRÁFICAS

[BI, 2006] TODO BI, "Informe Business Intelligence". Recopilación de los mejores artículos de inteligencia de negocio del 2006. Abril 2006.

[Cedeño, 2005] Cedeño Trujillo, Alexis: "Data Warehouse para la Estadística Docente basado en Software Libre"; Tesis de Diplomado, C. Habana, CEIS, CUJAE, 2005.

[Ruggia, 2002] Ruggia Raúl: "Diseño de DSS y DW basado en reutilización de especificaciones: Análisis y propuesta de estrategias", Instituto de Computación, Facultad de Ingeniería, Universidad de la República. Uruguay, 2002.

[Kimball,2007] "Bill Inmon vs. Ralph Kimball": http://www.1keydata.com/datawarehousing/inmon-kimball.html. Fecha de consulta: febrero de 2007.

[Hernández, 2004] Hernández, J.O; Ramírez, M.J.Q; Ferri, C.R: "Introducción a la Minería de Datos", Editorial Pearson, 2004.

[Luján, 2005] Luján-Mora, Sergio. "Data Warehouse Design with UML", Tesis Doctoral, Universidad de Alicante, Junio de 2005.

[OLAP, 2007] "OLAP": http://en.wikipedia.org/wiki/OLAP. Fecha de consulta: enero del 2007.

[Enhydra, 2007] "Enhydra Octopus Application": http://www.enhydra.org/tech/octopus/index.html. Fecha de consulta: febrero de 2008.

[Talend, 2007] "Talend, un nuevo ETL Open Source": http://todobi.blogspot.com/2007/01/talend-un-nuevo-etl-open-source.html. Fecha de consulta: febrero del 2007.

[ETL, 2007] "Open Source ETL (Extraction, Transform, Load)": http://www.manageability.org. Fecha de consulta: febrero de 2008.

[Mondrian, 2007] "Architecture Layers of a Mondrian system": http://mondrian.pentaho.org. Fecha de consulta: febrero del 2007.

[OpenSource, 2007] "Open Source in e collaborative way": http://oscarmanzano.f2o.org. Fecha de consulta: enero de 2008.

[Pentaho, 2006] Pentaho Corporation: "Pentaho Open Source Business Intelligence Platform. Technical White Paper". 2006.

[Simón, 2005] Simón Cuevas, Alina. "Modelo Dimensional", DATACIMEX, Noviembre 2005.

[Pentaho, 2007a] Pentaho Corporation: "Pentaho Data Integration", Spoon 2.5.0 User Guide, Abril 2007.

[MySQL, 2008] "Sitio Oficial de MySQL" : http://www.mysql.com/. Fecha de consulta: junio de 2008.

[PostgreSQL, 2006] “Kit de Prensa de PostgreSQL 8.1”, http://www.postgresql.org/about/press/presskit81.html.es, Fecha de consulta: junio de 2008.

GLOSARIO DE TÉRMINOS

Almacén de datos: en inglés: *Data Warehouse*, es una colección de datos orientada a un dominio, integrada, no volátil y que varía en el tiempo.

Business Intelligence: Conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

Carga: Es una de las etapas de los procesos ETL. Es el momento en el cual los datos de la fase anterior son cargados en el destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de procesos diferentes (actualizaciones, mantenimiento de la historia de los datos, etc.).

Consultas ad-hoc: consultas no predefinidas por el sistema, son creadas en tiempo de ejecución por los usuarios según sus necesidades.

Dashboard (Cuadros de Mando): Recurso que agrupa gran cantidad de información en entornos visuales muy llamativos y prácticos. Pueden incluir mapas, gráficos y componentes visuales conformando un entorno intuitivo para el usuario, consiguiendo la representación de complejos modelos de datos, relaciones entre variables y fórmulas de manera sencilla.

Datamart: Es un almacén de datos departamental, o sea, orientado a un sector determinado de la organización. Es también conocido: mercado de datos.

Extracción: Primera etapa de un proceso ETL, consiste en extraer los datos desde las fuentes y convertirlos en un formato preparado para iniciar el proceso de transformación.

LDAP: en ingles: *Lightweight Directory Access Protocol*. Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Minería de Datos: Conjunto de técnicas para la inducción de conocimiento útil a partir de masas muy grandes de datos. Tiene, por lo tanto, un solapamiento importante con otrasdisciplinas como la estadística tradicional, el reconocimiento de patrones, la inteligencia artificial, etc.

ANEXOS

Los procesos de extracción, transformación y carga se implementaron en su totalidad utilizando la herramienta ETL Kettle v2.5.0. De acuerdo, a la concepción de diseño de Kettle los procesos se dividen en dos categorías: *Transformation y Job*. En la primera categoría, están comprendidas todas las tareas referentes a la realización de los procesos ETL: lectura de las fuentes heterogéneas, transformaciones de diversos tipos, manejo de dimensiones, mapeo de datos, etc. y en la segunda categoría, se agrupan las tareas encargadas de ejecutar (de forma planificada o no), las transformaciones anteriores.

Kettle permite el diseño e implementación de los procesos en forma flujos de trabajo. Estos se componen de etapas (*Steps*) por las que transitan los datos y se relacionan mediante saltos (*Hops*), que describen la dirección en que circulan. Esta tarea, se realiza en un amigable ambiente visual. Kettle propone un conjunto de *Steps* para cada categoría de procesos (*Transformation o Job*).

Steps empleados en diseño de las transformaciones (Transformation):

Anexo 1

Steps	Descripción	Icono
Filter Rows	Permite filtrar registros basados en condiciones y comparaciones. Después de su ejecución, permite direccionar datos de acuerdo a su clasificación (correctos o no) hacia destinos diferentes.	
Text File Output	Permite exportar los datos que recibe, hacia un fichero texto. En este trabajo, se utiliza para registrar los errores ocurridos. Se generan ficheros con los datos separados por puntos y comas (;).	
Mapping	Permite la ejecución de otra transformación por debajo de la actual, necesita de valores de entrada y de salida. En este trabajo, se utiliza para ejecutar las transformaciones que realizan el mapeo de las llaves OLTP en las dimensiones, o sea, dada la llave OLTP de un registro se obtiene la llave que le corresponde en la tabla de dimensión.	

Mapping Input Specification	Permite especificar las variables de entrada de una transformación de mapeo.	
Mapping Output Specification	Permite especificar las variables de salida de una transformación de mapeo.	
Dimension Lookup/Update	Se utiliza con dos propósitos: 1. Implementar la propuesta de solución de Ralph Kimball, al fenómeno conocido como: *"Slowly Changing Dimensions".* Permite generar la llave primaria del nuevo registro (de varias formas), realizar las comparaciones pertinentes para conocer si ocurrió algún cambio en los atributos de la dimensión, controlar el campo version (incrementarlo), asignar el rango de validez de cada versión del cambio y finalmente cargar los datos en la tabla dimensión. 2. Realizar el mapeo de registros existentes en las tablas correspondientes a las dimensiones a partir de la llave primaria que poseía en el sistema operacional.	
Get System Info	Permite la recuperación de información del entorno, en este caso la fecha en que son ejecutadas las transformaciones.	
Java Script Value	Permite la ejecución de script. En este trabajo, se utiliza para procesar (dividir) los parámetros de la fecha y convertirlos en variables independientes.	
Insert / Update	Permite realizar inserciones o actualizaciones en una tabla de la base de datos. En este trabajo, se utiliza para cargar los datos en las tablas de hechos.	

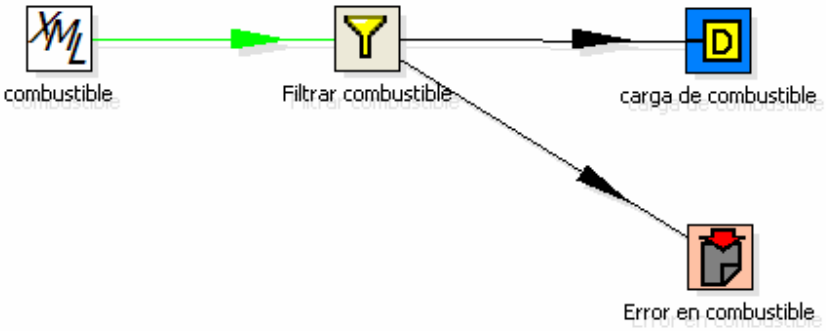
Database Lookup	Permite hacer búsquedas en una tabla de la base de datos. En este trabajo, se utiliza para obtener la llave primaria de un registro fecha a partir de sus componentes (día, mes y año).	
XML Input	Permite leer archivos xml. En este trabajo se utiliza para leer los xml que vienen de los centros.	

Steps empleados en diseño de los trabajos (Job):

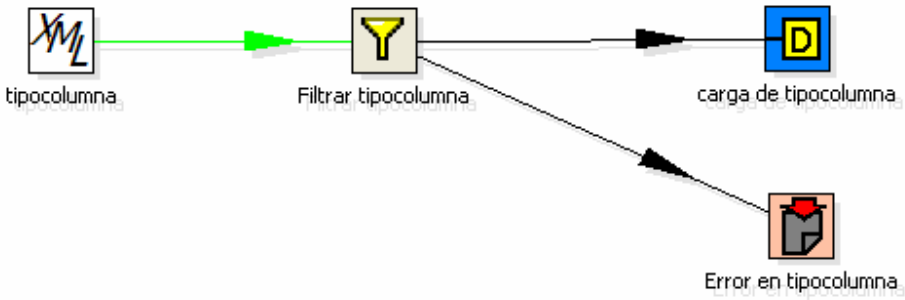
Steps	Descripción	Icono
Start	Punto de partida de la ejecución de un Job. Esta etapa permite realizar la planificación de ejecución de un Job (intervalo, diario, semanal o mensual).	
Transformation	Permite ejecutar desde el contexto de un Job una transformación previamente implementada.	
Job	Permite ejecutar desde el contexto de un Job otro Job previamente implementado.	

Transformaciones: Carga de dimensiones.

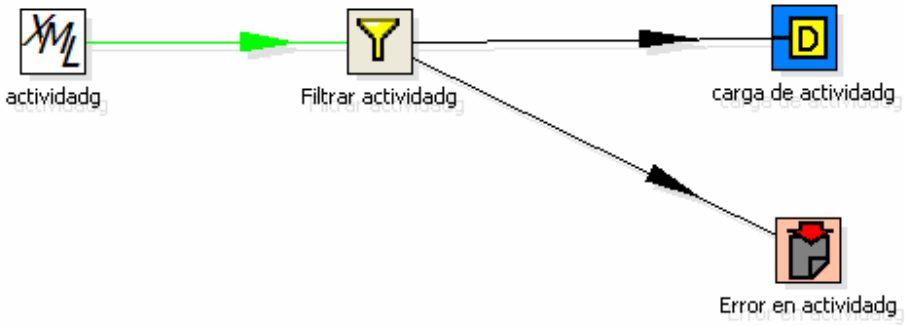
Dim_Combustible:



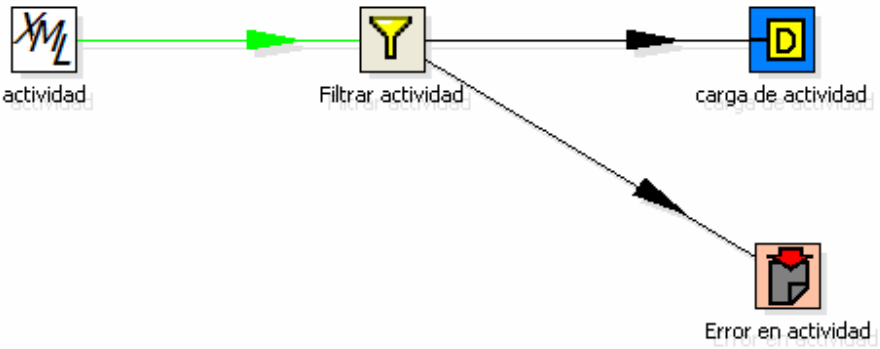
Dim_TipoColumnaC:



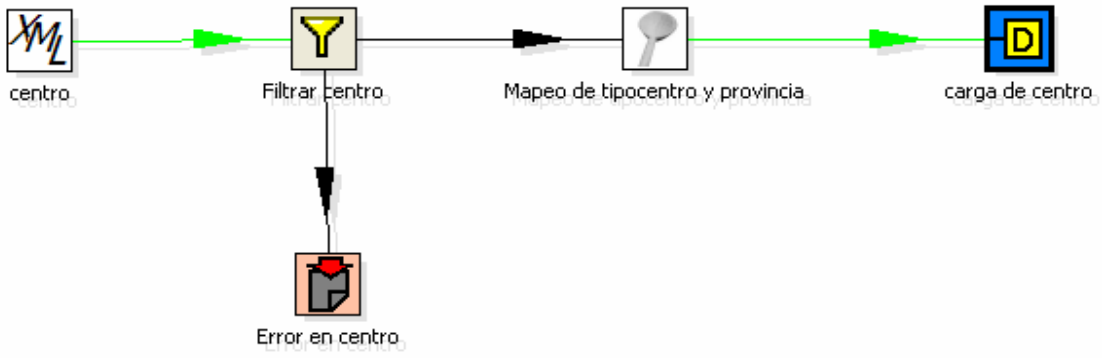
Dim_ActividadG:



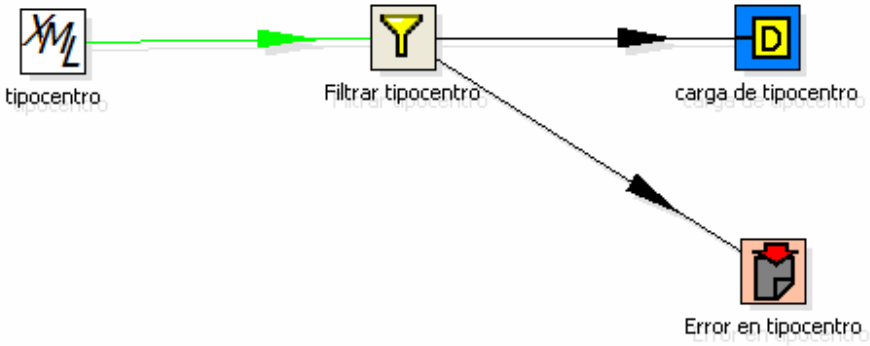
Dim_Actividad:



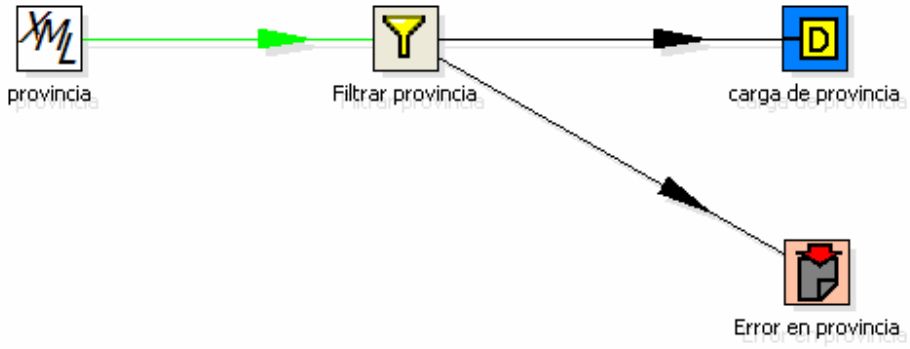
Dim_Centro:



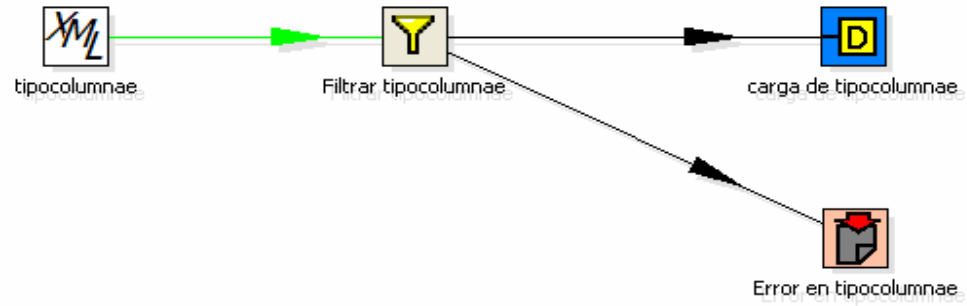
Dim_Tipocentro:



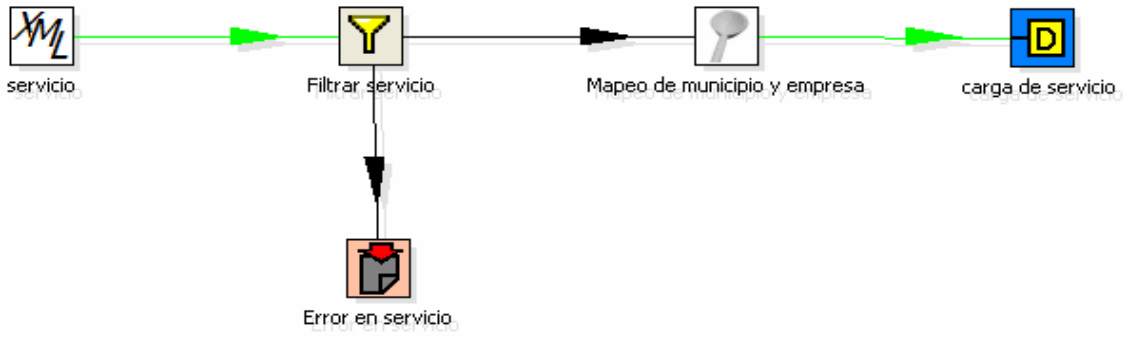
Dim_Provincia:



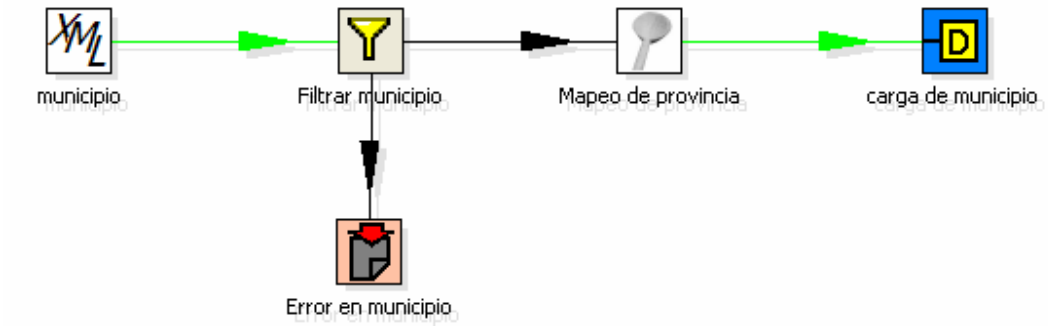
Dim_TipoColumnaE:



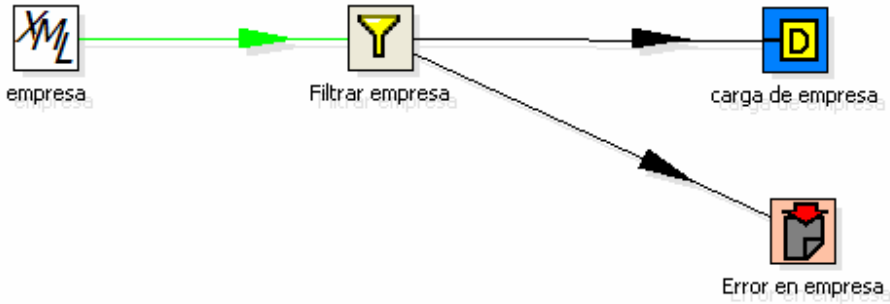
Dim_Servicio:



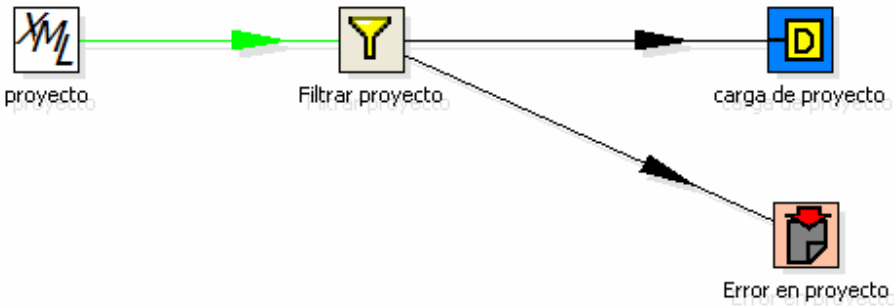
Dim_Municipio:



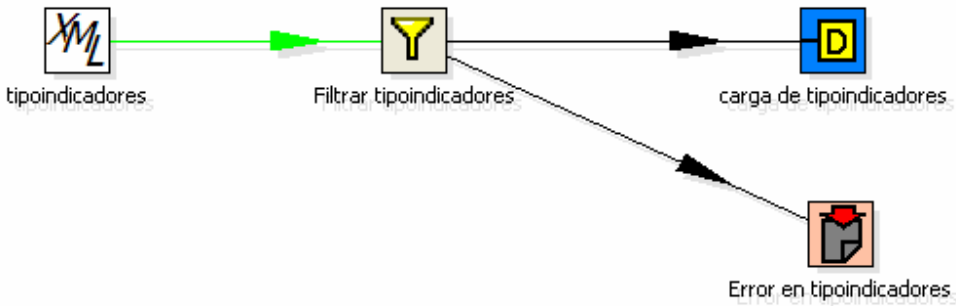
Dim_Empresa:



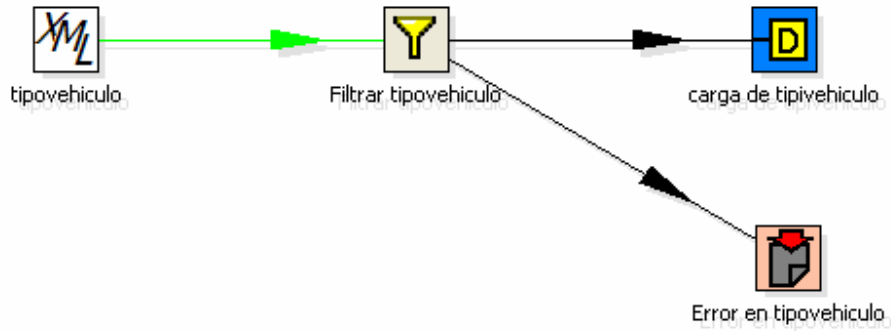
Dim_Proyecto:



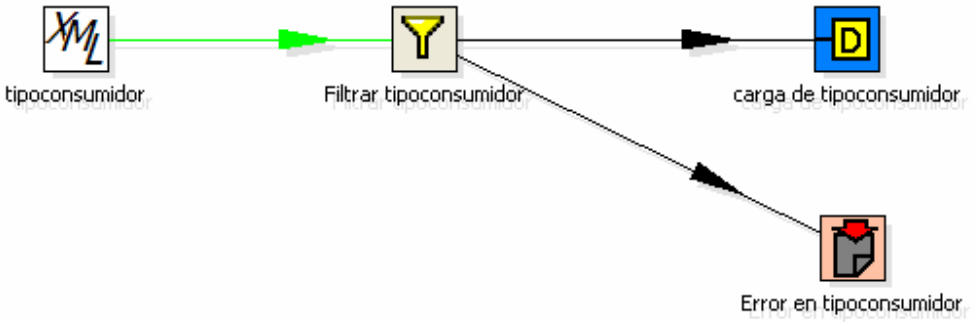
Dim_TipoIndicadores:



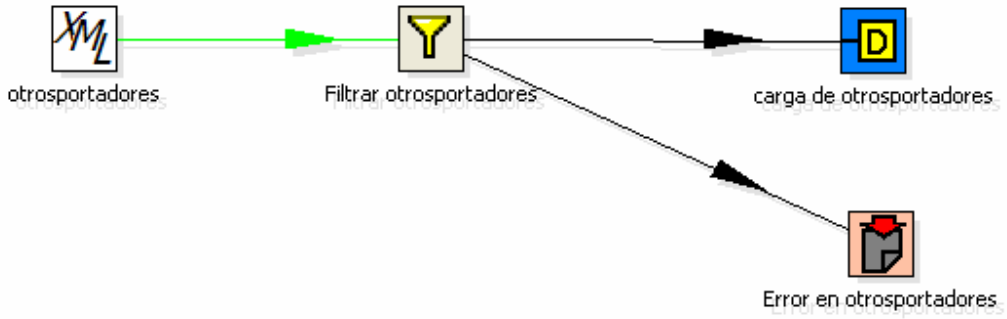
Dim_TipoVehículo:



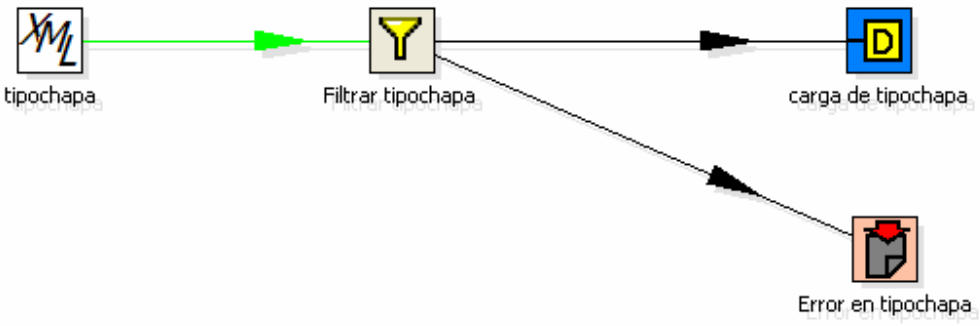
Dim_TipoConsumidor:



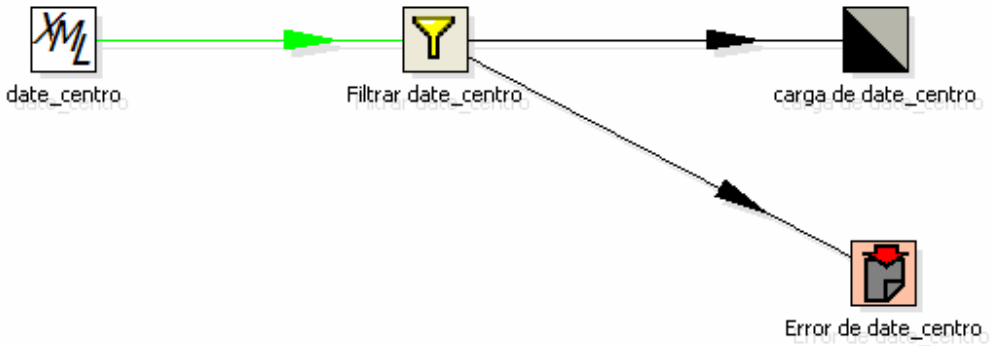
Dim_OtrosPortadores:



Dim_TipoChapa:

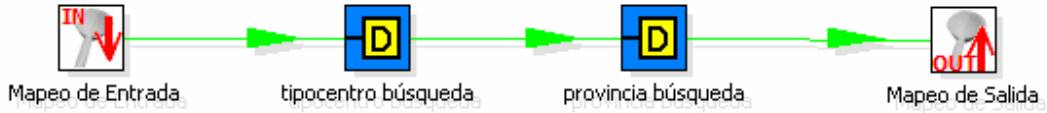


Dim_DateCentro:



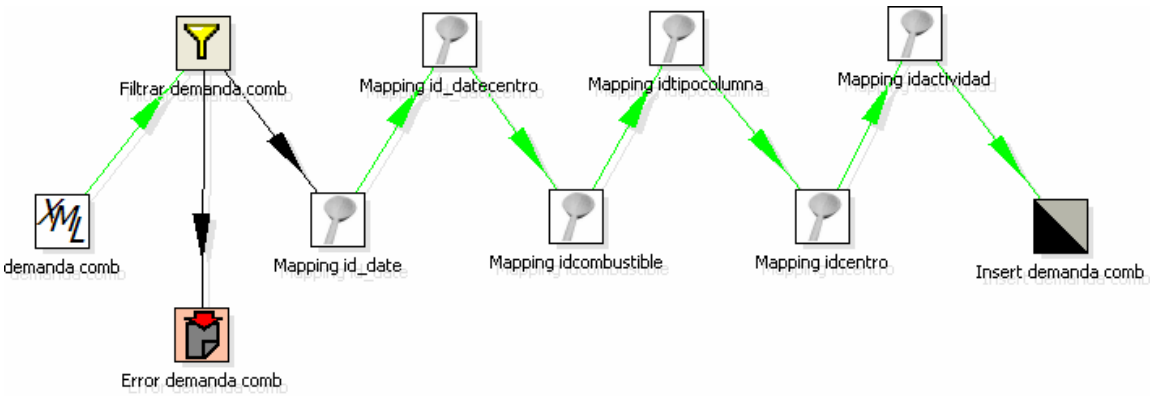
Dim_Date:

Ejemplos de mapeo de llaves OLTP en las tablas de dimensiones:
Mapeo de tipocentro y provincia:

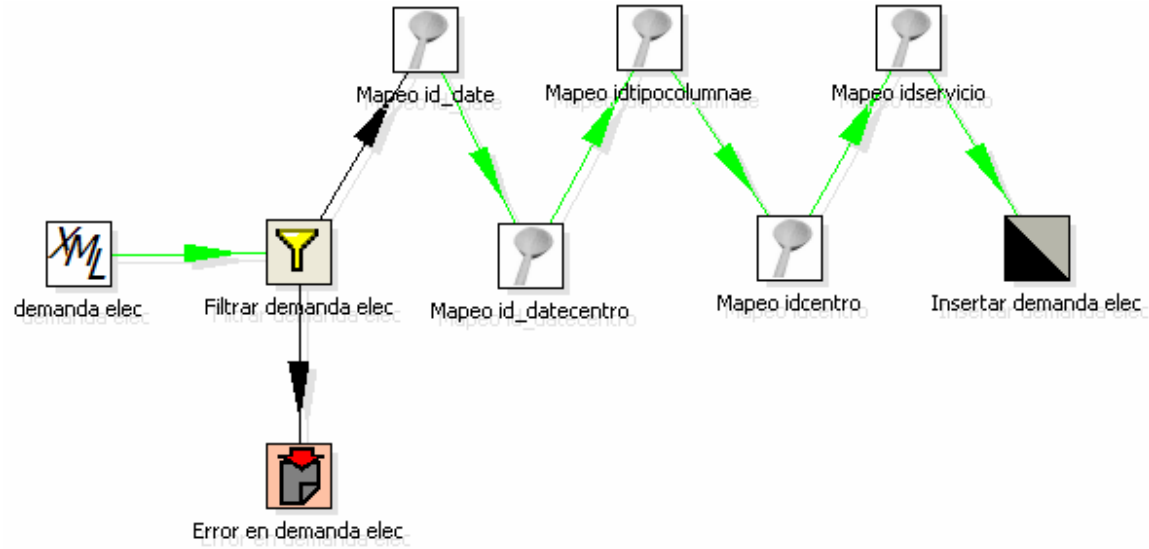


Mapeo de provincia:

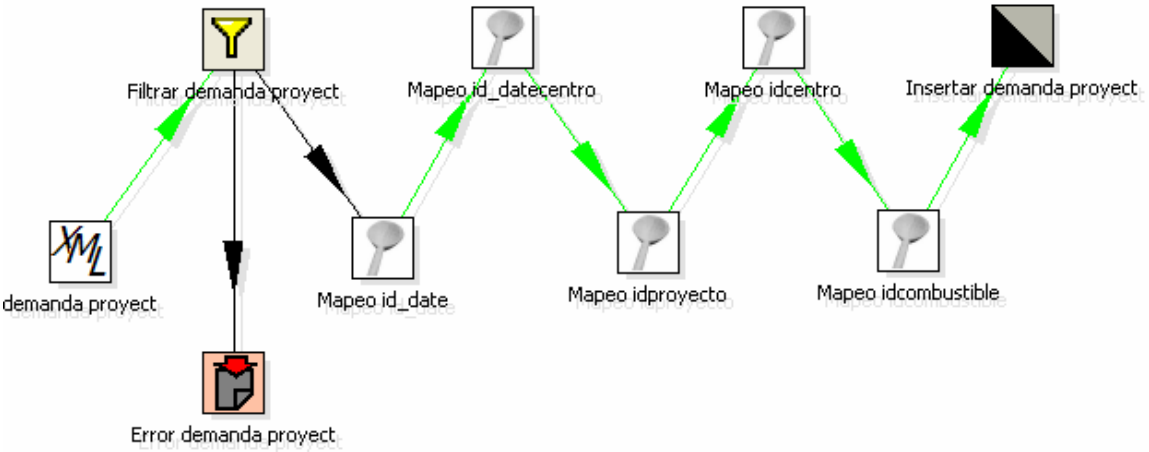
Transformaciones: Carga de hechos.
Fact_DemandaCombustible:



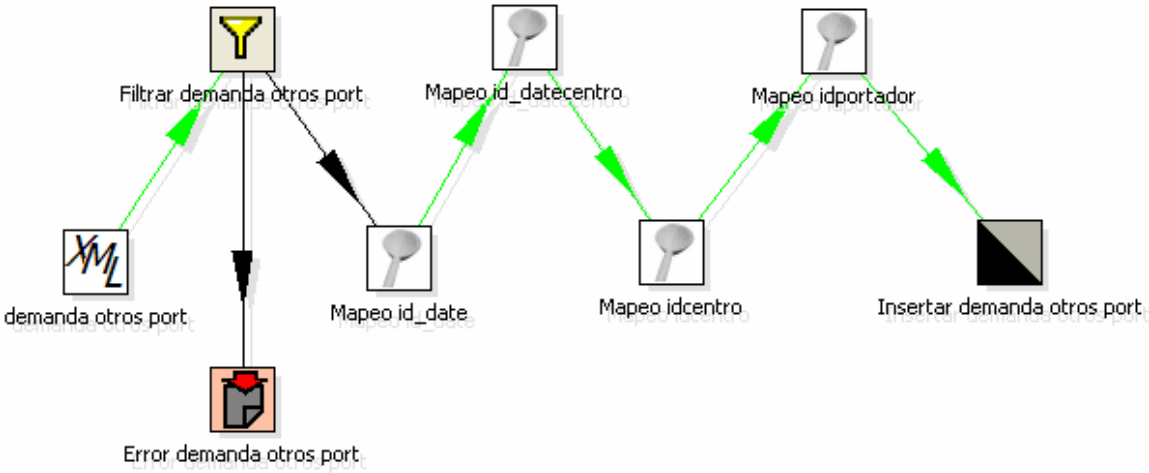
Fact_DemandaElectricidad:



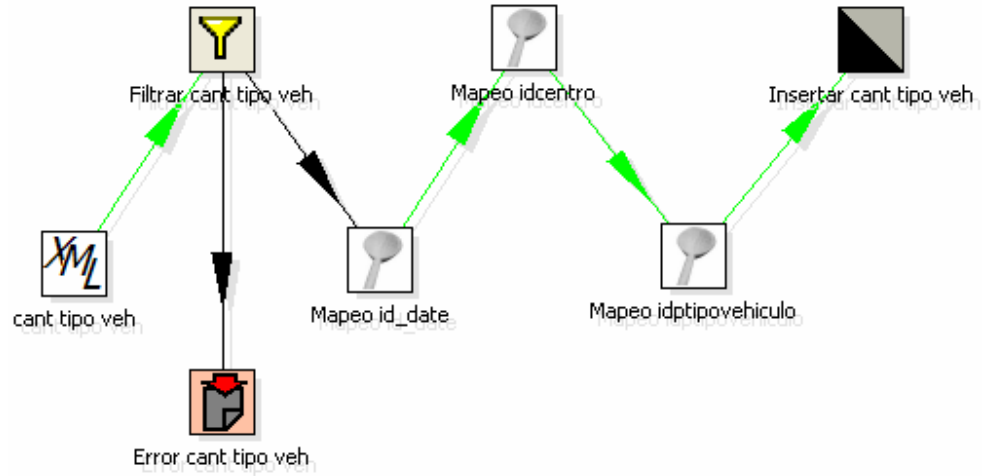
Fact_DemandaProyecto:



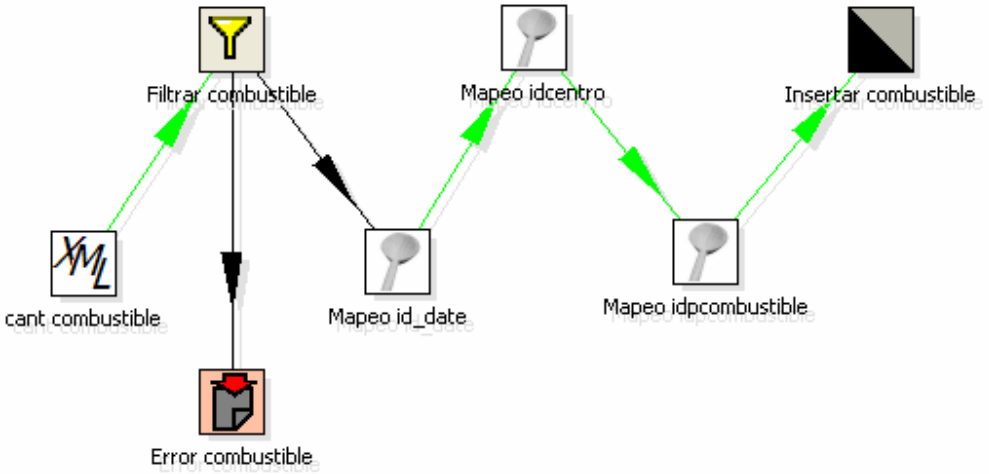
Fact _DemandaOtrosPortadores:



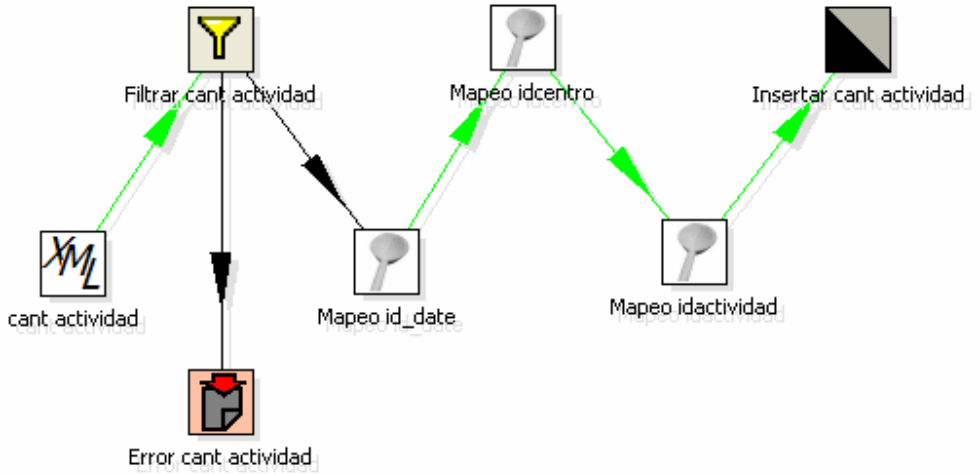
Fact _CantidadTipoVehículo:



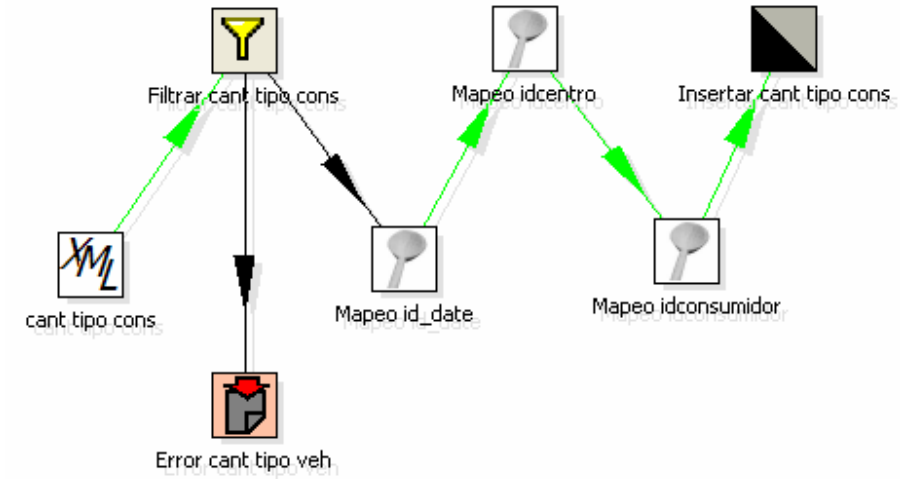
Fact _CantidadCombustible:



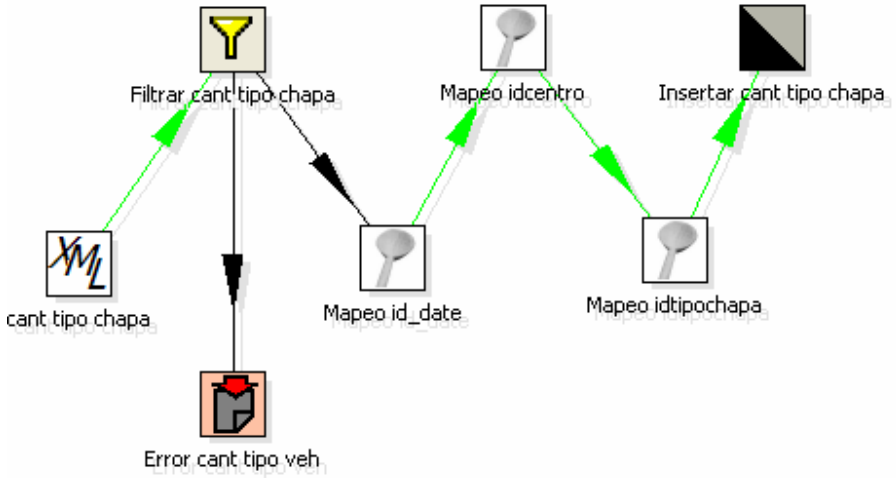
Fact _CantidadActividad:



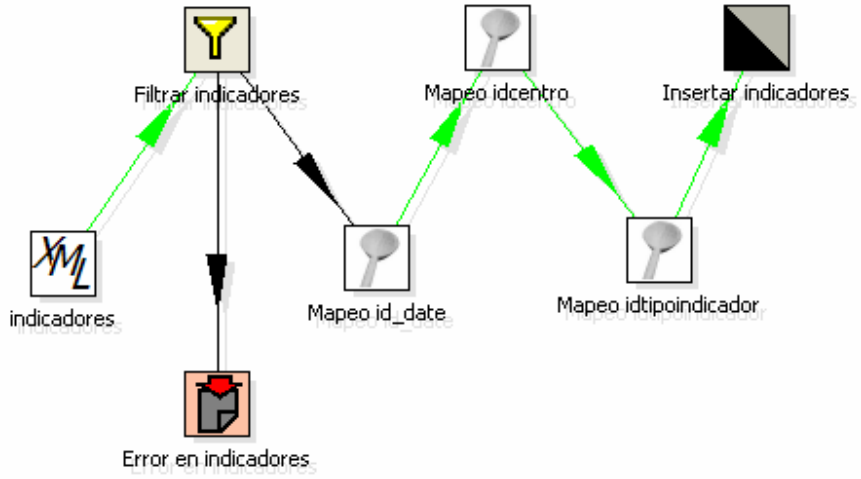
Fact _CantidadTipoConsumidor:



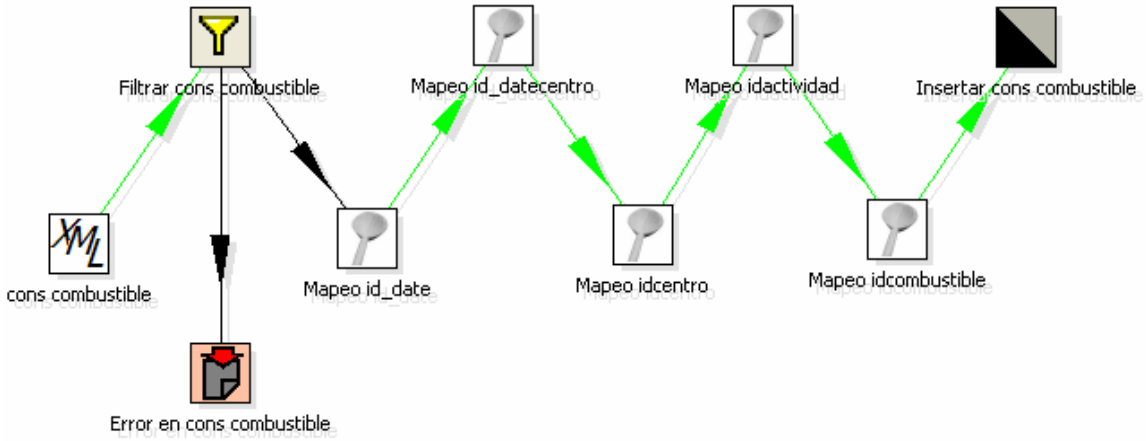
Fact _CantidadTipoChapa:



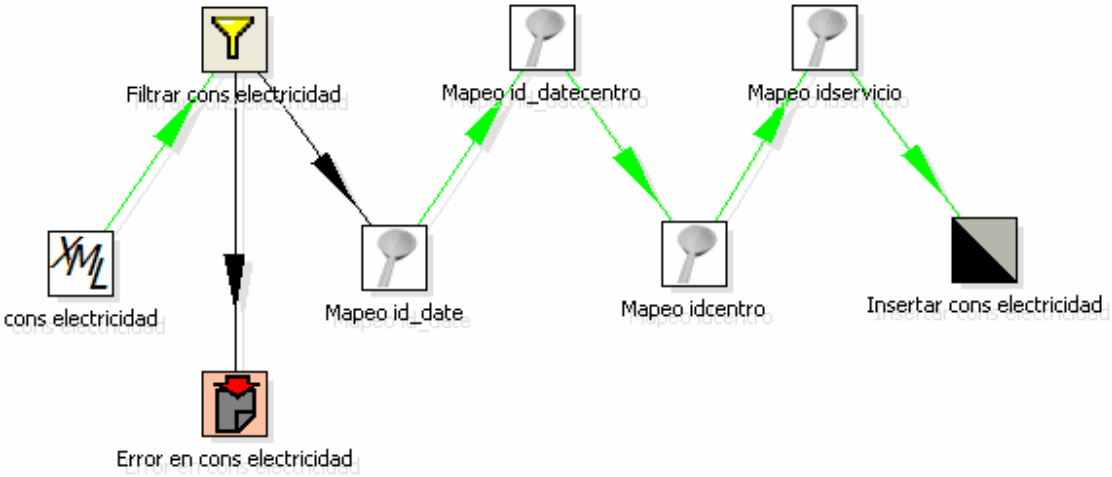
Fact _Indicadores:



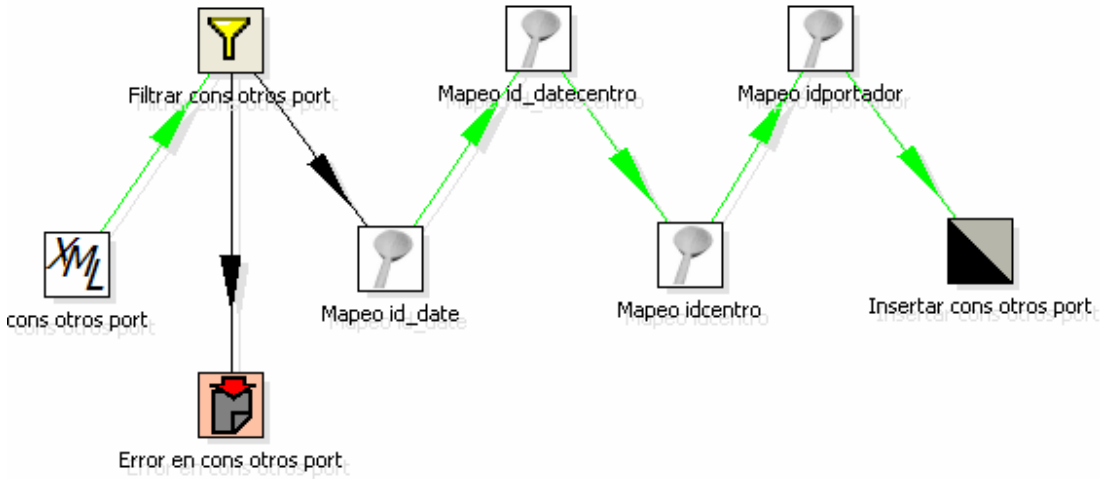
Fact _ConsumoCombustible:



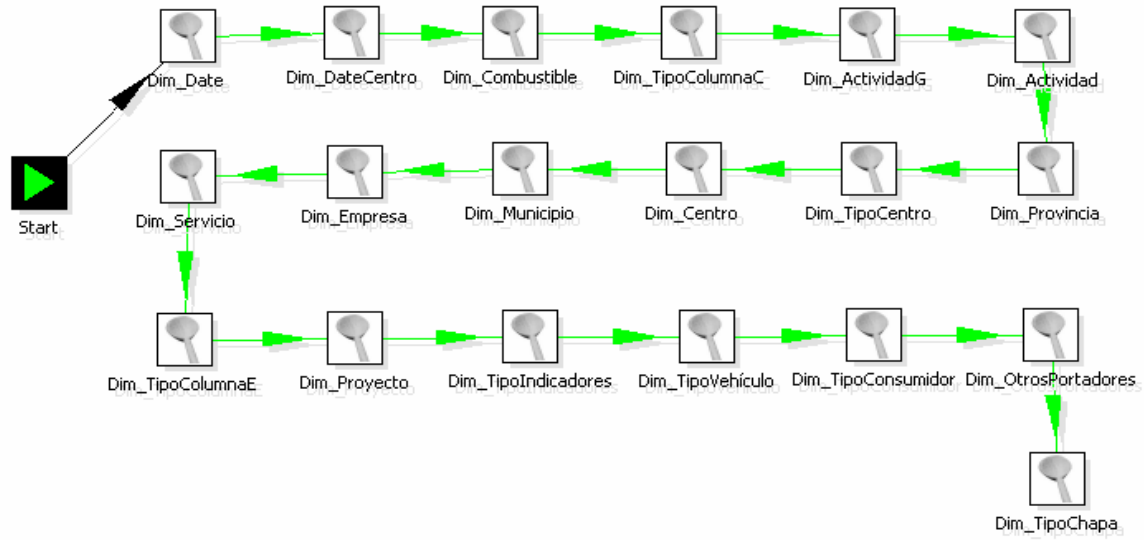
Fact _ConsumoElectricidad:



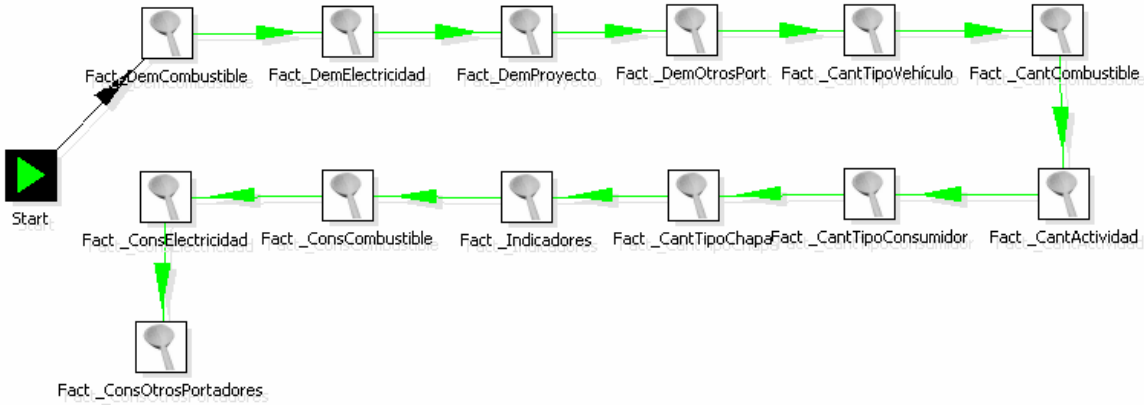
Fact _ConsumoOtrosPortadores:



Jobs: Ejecución de transformaciones.
Carga de dimensiones:



Carga de hechos:



Proceso ETL:

Anexo 2

Ejemplo de algunos modelos Excel, donde se recoge la información de los portadores energéticos.

	A	B	C	D	E	F	G	H	I	J
1	MODELO DE CONTROL DEL CONSUMO YCAPTACION DE DEMANDA DE COMBUSTIBLES Y LUI									
2										
3	(1) MES A PLANIFICAR:				(2) CENTRO :	CULT				
4										
5	U.M para el consumo: M Litros				(3) CODIGO :					(4) PRODUCT
6										
7			ACUMULADO HASTA:Abril			EST. MES ACTUAL Mayo			DEMANDA PROX. ME	
8	ACTIVIDAD	U.M N. ACT	NIVEL ACTIV	CONS. REAL	INDICE	NIVEL ACTIV	CONS EST.	INDICE	NIVEL ACTIV	DEM. CONS.
9	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
10	Cocción de Alimentos	Comensal	0,2783	9,991	**35,9001**	0	2,480	**50,00000**	0,123	2,494
11	Omnibus	MKm Recorrido	2,424	1,21202	**0,5000**	0,000	0,000	**#¡DIV/0!**	0	0
12	Microbus	MKm Recorrido	13,77	1,53	**0,1111**	4,642	0,490	**0,10556**	4,5	0,45
13	Autos	MKm Recorrido	0	0	**#¡DIV/0!**	0,000	0,000	**#¡DIV/0!**	0	0
14	Jeeps	MKm Recorrido	0		**0!**	0,000	0,000	**#¡DIV/0!**	0	0
15	Camiones	MKm Recorrido	12,75145	2	**000**	2,162	0,450	**0,20814**	4,95	0,99
16	Camionetas	MKm Recorrido	0,146		**1370**	0,000	0,000	**#¡DIV/0!**	1750	250
17	Paneles	MKm Recorrido	0	0	**#¡DIV/0!**	0,000	0,000	**#¡DIV/0!**	0	0
18	Tractores	MHoras	1,52	3,399	**2,2362**	0,384	1,055	**2,746094**	0,384	0,847
19	Bombas	MHoras			**#¡DIV/0!**			**#¡DIV/0!**	0	0
20	Fogones	MHoras			**#¡DIV/0!**			**#¡DIV/0!**	0	0
21	Crematorios	MHoras			**#¡DIV/0!**			**#¡DIV/0!**	0	0
22	Barco	Mmillas			**#¡DIV/0!**			**#¡DIV/0!**	0	0
23	Ordeñadoras	MHoras			**#¡DIV/0!**			**#¡DIV/0!**	0	0
24	Equipo para Soldar	MHoras			**#¡DIV/0!**			**#¡DIV/0!**	0	0
25	Grúa	MHoras			**#¡DIV/0!**			**#¡DIV/0!**	0	0
26	Labiratorios								0	0
27										
28	G. Electrógenos Emerg.	kW.h			**#¡DIV/0!**			**#¡DIV/0!**		
29	Compensados (19)	Vehículos			**#¡DIV/0!**			**#¡DIV/0!**	0	0
30	Privados (19)	Vehículos			**#¡DIV/0!**			**#¡DIV/0!**	0	0

EuroVal desactivar

MODELO EXCEL DE CONSUMO Y DEMDA COMBUSTIBLE

	A	B	C	D	E	F	G	H	I
12	CENTRO:	UPR							
13									
14									
15			CANTIDAD REAL A UTILIZAR						
16			Tiro Directo		Tarj. Mag.	Precio	Importe	1% INRE	TOTAL
17	Combustible en MN		Litros	Tons		Pesos/Litro	Pesos/CUC	Litros	Litros
18	Gasolina Reg. B-90 Comp.				450	0,55	247,50		Tot.Gasol.
19	Gasolina Reg. B-90 Priv.					0,80	0,00		5320
20	Gasolina Reg. B-90 Est.				3000	0,55	1650,00	30	Tot.Diesel
21	Diesel Regular Est.		0		4500	0,50	2250,00	45	5895
22	Diesel Regular Comp.					0,50	0,00		
23	Diesel Regular Priv.					0,75	0,00		
24	Diesel Regular G. Elect.					0,50	0,00		
25	Combustible en CUC								
26	Diesel Regular Est.				260	0,50	130,00		
27	Diesel Regular Comp.					0,50	545,00		
28	Gasolina Reg. B-83				1090	0,45	#¡REF!		
29	Gasolina Reg. B-90 Comp.					0,55	0,00		
30	Gasolina Reg. B-90 Est.				750	0,55	412,50		
31	Gasolina Esp. B-94 Comp.					0,65	0,00		
32	Gasolina Esp. B-94 Est.					0,65	0,00		
33	Otros portadores en MN		0						
34	Fuel Oil		7795						
35	Mezcla								

PLAN OPERATIVO COMBUSTIBLE

	A	B	C	D	E	F	G	H	I	J
1	CENTRO:	UHO				Mes:	Mayo			
2									CONSUMO REAL	
3									2006	2007
4									(1)	(2)
5	ANALISIS DE CONSUMOS DE LOS SERVICIOS ELECTRICOS SELECCIONADO									
6	PROV	CENTRO	Empresa	REEUP	Municipio	ORG	CODCLI	NTA		
7	111 UHOLM	Residencia Universitaria Universidad de Holguín "Oscar Lucero Moya"	UNIVERSIDAD DE HOLGUIN "OSCAR LUCERO MOYA"	223006821	Holguín	MES	442	M1B	25985,0	29039,0
8									0,1	0,1
9									0,1	0,1
10									0,1	0,1
11									0,1	0,1
12									0,1	0,1
13									0,1	0,1
14									0,1	0,1
15									0,1	0,1
16									0,1	0,1
17									0,1	0,1
18									0,1	0,1
19	ANALISIS CONSUMO TOTAL DE SERVICIS ELECTRICOS SELECCIONADOS:								25986,1	29040,1
20	ANALISIS DE CONSUMOS DE LOS SERVICIOS ELECTRICOS NO SELECCION									

CONSUMO DE ELECTRICIDAD