Universidad Central "Marta Abreu" de Las Villas

Facultad de Ingeniería Industrial y Turismo

Departamento de Ingeniería Industrial

Trabajo de diploma

Título: Extracción y documentación de patrones en los procesos ETL

Autor: Reinaldo de Jesús Morales Cruz

Tutor(a): Dr. Inty Saez Mosquera

-2012-

Agradecimientos

A mis dos madres Nélida y Elizabeth y mi primo Rafa por haber padecido todas mis experiencias en esta universidad.

A los "primos" Carlos, Dariel, Henry, Yariel por el increíble viaje de estos cinco años.

A las "primas" del 405 y 501 por los buenos tiempos.

A mis profesores todos por su esfuerzo y dedicación, especialmente a mi tutor, no imagino alguien mejor para trabajar en mi trabajo de diploma.

A Eliza por ser novia, amiga, mi mitad y enseñarme que el destino existe.

Dedicatoria

A mi hermanita Claudia para que se sienta orgullosa de mi.

Lo que uno encuentra en la vida es el destino. La manera en que lo encuentra es el esfuerzo personal.

Sai Baba

Índice

Introducción 1
Capítulo 1. El proceso de almacenamiento masivo de datos (Data Warehousing) 5
1.1. Características del proceso de almacenamiento de datos 5
1.1.1. Orientación al negocio 5
1.1.2. Integrada 6
1.1.3. Variante en el tiempo 6
1.1.4. No volátil 7
1.2. Cualidades 7
1.2.1. Ventajas 8
1.2.2. Desventajas 9
1.2.3. Redundancia 10
1.3. Procesos ETL: Kettel 10
1.3.1. Historia de Kettle 10
1.3.2. Programas que forman la herramienta 11
1.3.3. Trabajando con ficheros XML o repositorio 11
1.3.4. Concepto de transformación 12
1.3.5. Concepto de trabajo o Job 14
1.3.6. Interfaz de usuario 14
1.3.7. Pasos (steps) disponibles para las transformaciones 14
1.3.8. Implementación de un proceso ETL (Extract Transform Loading) 16
1.4. Subsistemas para los procesos ETL 17
1.5. Conclusiones parciales 21
2. Capítulo 2. Identificación de patrones basados en los subsistemas para procesos ETL 22
2.1. Introducción 22
2.2. identificación de los patrones en actividades 22
2.2.1. Patrones de Decodificar y Renombrar 22
2.2.2. Patrones de Etiquetado De datos 23
2.2.3. Patrones de agregación 24
2.2.4. Patrones de Limpieza de Datos 27
2.2.5. Patrones de validación de datos. 28
2.2.6 Patrones de generación de llaves y su gestión 30
2.2.7 Patrones de carga y mantenimiento de dimensiones 30
2.3. Conclusiones parciales 31
3. Capítulo 3: Modelación de transformaciones a partir de patrones de actividades de procesos ETL. 32
3.1 Introducción. 32
3.2 Patrones usados en la modelación de las transformaciones. 32
3.3 Procesos de transformación de los maestros de energía de la provincia Villa Clara (mayores). 32
3.4 Procesos de transformación de los maestros de energía de la provincia Villa Clara (mayores). 39
Conclusiones 44

Recomendaciones 45
Bibliografía..... 46

INTRODUCCIÓN

La planificación energética de la provincia es un proceso complejo que busca asignar recursos energéticos a los consumidores finales, que incluyen tanto al sector productivo como al residencial. Este proceso esta gobernado por una doble incertidumbre. De una parte, se desconoce –en el momento de la planificación, los valores de energía que podrán ser generados por el SEN[1] y por la otra, las demandas de los consumidores se presumen dentro de ciertos valores, pero no puede asegurarse que se mantendrán. Con una base insuficiente de información actualmente para la realización de pronósticos, las estimaciones de los planes de energía utilizan mayormente medidas de tendencia central (media, mediana, desviación típica), mientras que la información del comportamiento histórico de los consumos reales (medidos por los metros contadores de la empresa eléctrica) están en extensos ficheros de Excel® que en la mayoría de los casos prácticos, resulta poco viable su procesamiento para las actividades de planificación.

El comportamiento de la oferta de generación de energía del Sistema Energético Nacional, representa una variable a controlar más que un parámetro de funcionamiento de la economía nacional. Dado que la energía eléctrica no puede almacenarse para consumirse posteriormente (al menos no a escalas industriales), internacionalmente se utiliza una estrategia de seguimiento. La oferta sigue a la demanda siempre que sea posible. Cuando la demanda supera la capacidad de generación, se suceden los desagradables cortes energéticos. Una primera medida de eficacia de un sistema de planificación energética –desde el punto de vista no solo social, sino además económico, sería la capacidad de evitar cortes. Estos no solo generan una situación desagradable desde el punto de vista de bienestar en la población (sector residencial), sino que además tienen una importante arista económica al detener la producción industrial (sector estatal) y reducir con esto el PIG[2] y a nivel macro, el PIB nacional. Desde el punto de vista económico, la planificación energética busca satisfacer las necesidades de ambos sectores (residencial y estatal) sin incurrir en incremento sustanciales en la generación.

El Ministerio de Economía y Planificación (en lo adelante MEP) es el encargado de la fiscalización de la energía a nivel nacional y a través de sus dependencias provinciales, fiscaliza los consumos regionales. Sin embargo, la doble condición de incertidumbre hace oneroso la articulación completa del ciclo administrativo (Organización, Planificación, Ejecución y Control).

La primera etapa del ciclo es realizada por las propias empresas productivas[3], enfocado en sus planes y objetivos económicos, de acuerdo a los planes de la economía nacional. La etapa de planificación debe comenzar con la asignación de los planes de consumo energético para cada entidad. Es en este momento en que la ausencia de información histórica ordenada y estructurada para facilitar el proceso decisional de la asignación de los planes de energía, se convierte en una importante limitación.

A pesar de la disponibilidad de la información histórica (desde el año 2006 a la fecha) de los consumos de energía por cada metro contador en cada empresa, su ordenamiento y estructura hace impráctica su

[1] Sistema Energético Nacional (SEN).

[2] Producto Interno Geográfico (PIG). Este indicador es equivalente al PIB, solo que contiene información relacionada con las provincias.

[3] Esta investigación se centró en las empresas productivas, dejando el sector residencial para posteriores estudios.

utilización en la etapa de planificación. Si se tiene en cuenta el volumen de información que un solo mes implica (en términos de lecturas de metros contadores en las empresas productivas) se entiende perfectamente la necesidad de ordenar y estructurar esta información para poder disponer de servicios de datos que faciliten el proceso de consulta para la elaboración de los planes.

Es importante aclarara que la estimación de las necesidades de energía (primer paso de la planificación) es realizada por las propias empresas. En función de sus volúmenes de producción planificados y la estructura de su matriz energéticas, las empresas hace una estimación de la demanda de energía para el próximo periodo (mes siguiente). Sin embargo, esta estimación ha de ser contrastada con la información histórica, por la posibilidad práctica de producirse sobre-estimaciones o sub-estimaciones, que en cualquier caso, traen consigo dos situaciones no deseables. En el primer caso (sobre-estimación), implican un gasto de generación potencial y en el segundo caso, traen consigo el corte, debido a las actuales políticas de administración energética vigentes en el país.

Adicionalmente, la etapa de ejecución es realizada de manera independiente y asincrónica por cada una de las empresas involucradas en el proceso de planificación. En efecto, una vez asignada la cifra de energía, cada empresa comienza la ejecución del plan en función de sus propias demandas y relaciones de consumo, sin que tenga o exista relación entre sus consumos, incluso en aquellas que pertenecen a un mismo organismo de la administración central del estado y existan relaciones de producción entre ellas[4].

Estas condiciones introducen una importante limitación en la etapa de control. Asumiendo la perspectiva de control como la capacidad de comparar la realización de la actividad con respecto a su plan y la habilidad (de todos los instrumentos del sistema de control) de detectar a tiempo desviaciones, puede concluirse que la etapa de control es en la mayoría de los casos prácticos incompleta. La ausencia de un repositorio de datos históricos, ordenados y estructurado en función de mejorar la capacidad de análisis y respuesta a las preguntas de control, aparece nuevamente como una importante limitante en la gestión de las oficinas provinciales del MEP.

Sin embargo, los datos históricos de los consumos de energía (medido por sus metros contadores) de cada empresa, existen y están disponibles. De manera que esta limitante cambia de naturaleza (aun cuando conserva su contenido). Tal cambio se produce toda vez que los datos históricos existen, restando solo el problema de lidiar con su ordenamiento y estructuración consecuente con el proceso de planificación y control.

Ha de tenerse en cuenta que solo en la provincia de Villa Clara en un mes cualquiera, se toman alrededor de 2181 lectura de metros contadores. Tal cantidad de datos no puede ser procesada en su fuente original (ficheros de MS© Excel®). Para asegurar el proceso de ordenamiento y estructuración de datos en este tipo de escenario, se utilizan las herramientas de integración disponibles en las Suites de Business Intelligence (Bouman & Dongen, 2009; Dario, 2009; Seruca, Cordeiro, Hammoudi, & Filipe, 2006; Utley, 2008; Ventana Research, 2008).

[4] Las causas de este asincronismo son múltiples y entre ellas destaca por su peso la asincronía en la gestión de materias primas y materiales para la realización de la producción, que está sujeta en la mayoría de los casos a la capacidad financiera y los mecanismos establecidos en el país para las importaciones.

Los procesos de integración de datos representan la antesala de los denominados procesos de Data Warehousing (procesos de almacenamiento de datos). Estos procesos son los encargados de Extraer (limpiar, comprobar, validar, eliminar), Transformar (estructurar) y Cargar (mover físicamente los datos desde las fuentes originales al almacén) los datos necesarios para los procesos decisionales (del cual es un caso particular los procesos de planificación y control) (Chen, Filipe, Seruca, & Cordeiro, 2006; Hilton, 2007; Seipel & Turull-Torres, 2004; Wrembel & Koncilia, 2006). A estos procesos se les denomina ETL (Extract, Transform & Load, por sus siglas en idioma inglés) y generalmente son realizados por herramientas conocidas bajo la denominación genérica de herramientas ETL.

A pesar de la diversidad de contextos, relaciones, procesos, y significados de los datos en los diferentes escenarios de procesos de integración de datos, los procesos ETL comparten generalidades que pueden ser utilizadas. La primera generalidad que es posible advertir es que siempre (con independencia de la naturaleza o contenido del caso de integración) este proceso puede ser reducido a tres etapas: extraer, transformar y cargar.

Ralph Kimball (jcurtod, 2012) identificó 38 subsistemas presentes en la mayoría de los procesos ETL. Teniendo en cuenta esta primera reducción en la diversidad de estos procesos, esta investigación se planteó la siguiente pregunta:

1. A partir de los 38 subsistemas identificados por Kimball, ¿será posible reducir más esta diversidad mediante la identificación y clasificación de patrones en las propias transformaciones?

Consecuentemente, la posible respuesta afirmativa a esta interrogante introduce una importante ventaja. Si es posible identificar patrones en las transformaciones –aun comenzando por un grupo reducida de estas, que puede ir incrementándose progresivamente con la importante ventaja adicional de aumentar la disponibilidad de patrones, qué beneficios –desde el punto de vista práctico, traería consigo estos patrones. Concretamente, la segunda pregunta de investigación planteada se resume a:

2. A partir de los patrones identificados en las transformaciones analizadas, ¿será posible construir nuevas transformaciones válidas en procesos de integración en diferentes contextos y con diferentes propósitos?

En correspondencia con las preguntas de investigación planteadas, el **OBJETIVO GENERAL** de la investigación se centró en el estudio y análisis de las transformaciones disponibles en el repositorio de SIISVAE© (2005-2011) a fin de identificar patrones en los procesos ETL que puedan ser reutilizados en nuevos escenarios de integración, concretamente, en el caso del ordenamiento y estructuración de la información histórica de los consumos de energía eléctrica de la provincia de Villa Clara.

Para dar cumplimiento al objetivo general, este se estructuró en los **OBJETIVOS ESPECÍFICOS** siguiente:

1. Identificar las clasificaciones generales de los procesos de extracción, transformación y carga (ETL)
2. Definir patrones de procesos ETL reutilizables para propósitos y finalidades comunes
3. Utilizar la herramienta Pentaho Data Integration (PDI) para facilitar el uso de estos patrones

El informe de investigación se organizó en tres capítulos. En primer capítulo se hace una revisión de los conceptos y características de los almacenes de datos y los procesos de Data Warehousing, así como de los procesos de ETL, que fundamenta –a través de las conclusiones parciales de este capítulo, la

pertinencia, actualidad, necesidad y conveniencia de la investigación relacionada directamente con la respuesta a la primera pregunta de investigación. En este sentido, las conclusiones parciales de este capítulo dan respuesta a nivel teórico a la primera pregunta de investigación, así como a los objetivos específicos 1 y 2 total y parcialmente, respectivamente.

En el segundo capítulo se presentan un conjunto de patrones identificados, a partir del análisis y evaluación de las transformaciones disponibles en el repositorio de SIISVAE© (2005-2011). Adicionalmente, cada patrón presentado se comprobó en fuentes especializadas (ejemplo: el rincón del BI, Blog especializado en temas de Inteligencia Empresarial) su manifestación en otros procesos, a fin de comprobar su generalidad, validez y consistencia como patrón de integración. Este capítulo a través de sus conclusiones parciales, completa el objetivo específico 2 de la investigación.

Finalmente pero no menos importante, en el tercer capítulo se realiza una transformación –vía la reutilización de los patrones identificados, con el propósito de dar respuesta a la segunda pregunta de investigación y al tercer objetivo específico.

CAPÍTULO 1. EL PROCESO DE ALMACENAMIENTO MASIVO DE DATOS (DATA WAREHOUSING)

1.1. CARACTERÍSTICAS DEL PROCESO DE ALMACENAMIENTO DE DATOS

1.1.1. ORIENTACIÓN AL NEGOCIO

La primera característica del DW, es que la información se clasifica en base a los aspectos que son de interés para la organización. Esta clasificación afecta el diseño y la implementación de los datos encontrados en el almacén de datos, debido a que la estructura del mismo difiere considerablemente a la de los clásicos procesos operacionales orientados a las aplicaciones.

A continuación, y con el fin de obtener una mejor comprensión de las diferencias existentes entre estos dos tipos de orientación, se realizará un análisis comparativo:

- Con respecto al nivel de detalle de los datos, el DW excluye la información que no será utilizada exclusivamente en el proceso de toma de decisiones; mientras que en los procesos orientados a las aplicaciones, se incluyen todos aquellos datos que son necesarios para satisfacer de manera inmediata los requerimientos funcionales de la actividad que soporten. Por ejemplo, los datos comunes referidos al cliente, como su dirección de correo electrónico, fax, teléfono, D.N.I., código postal, entre otros, que son tan importantes de almacenar en cualquier sistema operacional, no son tenidos en cuenta en el depósito de datos por carecer de valor para la toma de decisiones, pero sí lo serán aquellos que indiquen el tipo de cliente, su clasificación, ubicación geográfica, sexo, edad, entre otros (Langer, 2007; Rainardi, 2008).
- En lo que concierne a la interacción de la información, los datos operacionales mantienen una relación continua entre dos o más tablas, basadas en alguna regla comercial vigente; en cambio las relaciones encontradas en los datos residentes del DW son muchas, debido a que por lo general cada tabla del mismo estará conformada por la integración de varias tablas u otras fuentes del ambiente operacional, cada una con sus propias reglas de negocio inherentes (Langer, 2007; Wrembel & Koncilia, 2006).

El origen de este contraste es totalmente lógico, ya que el ambiente operacional se diseña alrededor de las aplicaciones u programas que necesite la organización para llevar a cabo sus actividades diarias y funciones específica. Por ejemplo, una aplicación de una empresa minorista manejará: stock, lista de precios, cuentas corrientes, pagos diferidos, impuestos, retenciones, ventas, notas de crédito, compras, etc. De esta manera, la base de datos combinará estos elementos en una estructura que se adapte a sus necesidades.

En contraposición, por ejemplo, para un fabricante el ambiente DW se organizará alrededor de entidades de alto nivel tales como: clientes, productos, rubros, proveedores, vendedores, zonas, etc. Que son precisamente aquellos sujetos mediante los cuales se desea analizar la información. Esto se debe a que el depósito de datos se diseña para realizar consultas e investigaciones sobre las actividades de la organización y no para soportar los procesos que se realizan en ella (Rainardi, 2008).

En síntesis, la ventaja de contar con procesos orientados a la aplicación, esta fundamentada en la alta accesibilidad de los datos, lo que implica un elevado desempeño y velocidad en la ejecución de consultas, ya que las mismas están predeterminadas; mientras que en el DW para satisfacer esta ventaja se requiere que la información este desnormalizada, es decir, con redundancia1 y que la misma esté dimensionada, para evitar tener que recorrer toda la base de datos cuando se necesite realizar algún

análisis determinado, sino que simplemente la consulta sea enfocada por variables de análisis que permitan localizar los datos de manera rápida y eficaz, para poder de esta manera satisfacer una alta demanda de complejos exámenes en un mínimo tiempo de respuesta (Alshawi, Saez-Pujol, & Irani, 2003; Dario, 2009; Samtani, Mohania, Kumar, & Kambayashi, 1999; Wang, 2008).

1.1.2. INTEGRADA

La integración implica que todos los datos de diversas fuentes que son producidos por distintos departamentos, secciones y aplicaciones, tanto internas como externas, deben ser consolidados en una instancia antes de ser agregados al DW. A este proceso se lo conoce como Extracción, Transformación y Carga de Datos (Extraction, Transformation and Load - ETL) [5].

La integración de datos, resuelve diferentes tipos de problemas relacionados con las convenciones de nombres, unidades de medidas, codificaciones, fuentes múltiples, etc., cada uno de los cuales será correctamente detallado y ejemplificado más adelante (Abramowicz, 2007; Seruca, et al., 2006; Yang & SpringerLink (Online service), 2009).

La causa de dichos problemas, se debe principalmente a que a través de los años los diseñadores y programadores no se han basado en ningún estándar concreto para definir nombres de variables, tipos de datos, etc., ya sea por carecer de ellos o por no creer que sean necesarios. Por lo cual, cada uno por su parte ha dejado en cada aplicación, módulo, tabla, etc., su propio estilo personalizado, confluyendo de esta manera en la creación de modelos muy inconsistentes e incompatibles entre sí (Chen, et al., 2006; Manolopoulos, Filipe, Constantopoulos, & Cordeiro, 2006; Seruca, et al., 2006).

Los puntos de integración afectan casi todos los aspectos de diseño, y cualquiera sea su forma, el resultado es el mismo, ya que la información será almacenada en el DW en un modelo globalmente aceptable y singular, aun cuando los sistemas operacionales y demás fuentes almacenen los datos de maneras disímiles, para que de esta manera el usuario final este enfocado en la utilización de los datos del depósito y no deba cuestionarse sobre la confiabilidad o solidez de los mismos.

1.1.3. VARIANTE EN EL TIEMPO

Debido al gran volumen de información que se manejará en el DW, cuando se le realiza una consulta, los resultados deseados demorarán en originarse. Este espacio de tiempo que se produce desde la búsqueda de datos hasta su consecución es del todo normal en este ambiente y es, precisamente por ello, que la información que se encuentra dentro del depósito de datos se denomina de tiempo variable.

Esta característica básica, es muy diferente de la información encontrada en el ambiente operacional, en el cual, los datos se requieren en el momento de acceder, es decir, que se espera que los valores procurados se obtengan a partir del momento mismo de acceso (Grabot, Mayère, & Bazet, 2008; Gunasekaran, 2008; Minoli, 2008).

Esto contribuye a una de las principales ventajas del almacén de datos: los datos son almacenados junto a sus respectivos históricos. Esta cualidad que no se encuentra en fuentes de datos operacionales,

[5] En lo adelante se utilizará el acrónimo ETL para referirse a los procesos de Extracción, Transformación y Carga de datos desde diferentes orígenes hacia el almacén de datos. En todos los casos en que exista un acrónimo reconocido en la lengua española, su uso será preferente para evitar crear nuevos acrónimos que puedan crear confusiones.

garantiza poder desarrollar análisis de la dinámica de la información, pues ella es procesada como una serie de instantáneas, cada una representando un periodo de tiempo. Es decir, que gracias al sello de tiempo se podrá tener acceso a diferentes versiones de la misma información (Torra & Narukawa, 2007; Yang & SpringerLink (Online service), 2009).

Es importante tener en cuenta la granularidad3 de los datos, así como también la intensidad de cambio natural del comportamiento de los fenómenos de la actividad que se desarrolle, para evitar crecimientos incontrolables y desbordamientos de la base de datos. El intervalo de tiempo y periodicidad de los datos debe definirse de acuerdo a la necesidad y requisitos de los usuarios. Es elemental aclarar, que el almacenamiento de datos históricos, es lo que permite al DW desarrollar pronósticos y análisis de tendencias y patrones, a partir de una base estadística de información.

1.1.4. NO VOLÁTIL

La información es útil para el análisis y la toma de decisiones solo cuando es estable. Los datos operacionales varían momento a momento, en cambio, los datos una vez que entran en el DW no salen (ver figura 1.1).

La actualización, o sea, insertar, eliminar y modificar, se hace de forma muy habitual en el ambiente operacional sobre una base, registro por registro, en cambio en el depósito de datos la manipulación básica de los datos es mucho más simple, debido a que solo existen dos tipos de operaciones: la carga de datos y el acceso a los mismos. Por esta razón es que en el DW no se requieren mecanismos de control de la concurrencia y recuperación (Bouman & Dongen, 2009; Dario, 2009; Wang, 2008).

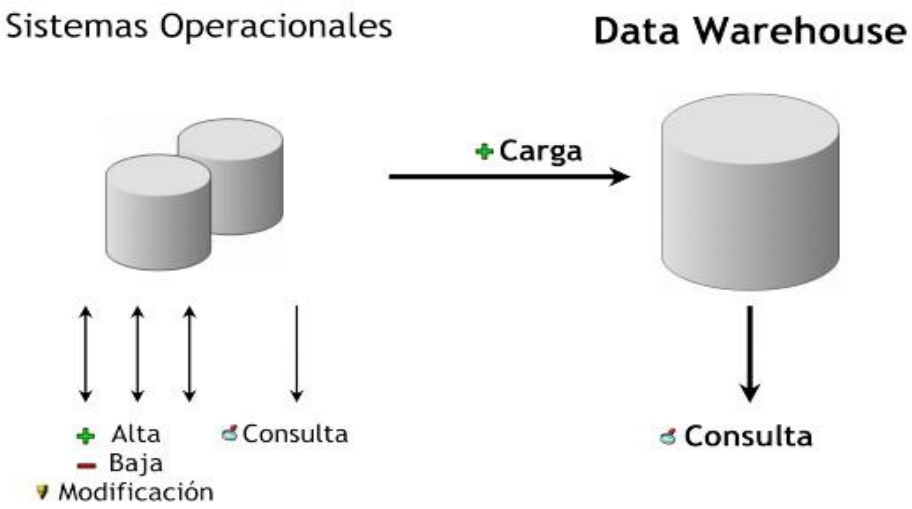

Figura 1.1. Almacén de datos no volátil. Fuente: (Dario, 2009)

1.2. CUALIDADES

Una de las primeras cualidades que se puede mencionar del DW, es que maneja un gran volumen de datos, debido a que consolida en su estructura la información recolectada durante años, proveniente de diversas fuentes y áreas, en un solo lugar centralizado. Es por esta razón que el depósito puede ser soportado y mantenido sobre diversos medios de almacenamiento. Además, como ya se ha mencionado,

el almacén de datos presenta la información sumarizada y agregada desde múltiples versiones, y maneja información histórica.

Organiza y almacena los datos que se necesitan para realizar consultas y procesos analíticos, con el propósito de responder a preguntas complejas y brindarles a los usuarios finales la posibilidad de que mediante una interface amigable, intuitiva y fácil de utilizar, puedan tomar decisiones sobre los datos sin tener que poseer demasiados conocimientos informáticos. El DW permite un acceso más directo, es decir, la información gira en torno al negocio, y es por ello que también los usuarios pueden sentirse cómodos al explorar los datos y encontrar sus complejas relaciones (Wrembel & Koncilia, 2006).

Cabe aclarar que el Data Warehousing no se compone solo de datos, ni tampoco solo se trata de un depósito de datos aislado. El Data Warehousing hace referencia a un conjunto de herramientas para consultar, analizar y presentar información, que permiten obtener o realizar análisis, reporting, extracción y explotación de los datos, con alta performance, para transformar dichos datos en información valiosa para la organización (Rainardi, 2008; Raisinghani, 2004).

Con respecto a las tecnologías que son empleadas, se pueden encontrar las siguientes:

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

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

1.2.1. VENTAJAS

A continuación se enumerarán algunas de las ventajas más sobresalientes que trae aparejada la implementación de un Data Warehousing y que ejemplifican de mejor modo sus características y cualidades (Ballard, International Business Machines Corporation. International Technical Support Organization., & Books24x7 Inc., 2006; Dario, 2009; Wang, 2008; Wrembel & Koncilia, 2006):

- Transforma datos orientados a las aplicaciones en información orientada a la toma de decisiones.
- Integra y consolida diferentes fuentes de datos (internas y/o externas) y departamentos empresariales, que anteriormente formaban islas, en una única plataforma sólida y centralizada.
- Provee la capacidad de analizar y explotar las diferentes áreas de trabajo y de realizar un análisis inmediato de las mismas.
- Permite reaccionar rápidamente a los cambios del mercado.
- Aumenta la competitividad en el mercado.
- Elimina la producción y el procesamiento de datos que no son utilizados ni necesarios, producto de aplicaciones mal diseñadas o ya no utilizadas.
- Mejora la entrega de información, es decir, información completa, correcta, consistente, oportuna y accesible. Información que los usuarios necesitan, en el momento adecuado y en el formato apropiado.

- Logra un impacto positivo sobre los procesos de toma de decisiones. Cuando los usuarios tienen acceso a una mejor calidad de información, la empresa puede lograr por sí misma: aprovechar el enorme valor potencial de sus recursos de información y transformarlo en valor verdadero; eliminar los retardos de los procesos que resultan de información incorrecta, inconsistente y/o inexistente; integrar y optimizar procesos a través del uso compartido e integrado de las fuentes de información; permitir al usuario adquirir mayor confianza acerca de sus propias decisiones y de las del resto, y lograr así, un mayor entendimiento de los impactos ocasionados.
- Aumento de la eficiencia de los encargados de tomar decisiones.
- Los usuarios pueden acceder directamente a la información en línea, lo que contribuye a su capacidad para operar con mayor efectividad en las tareas rutinarias o no. Además, pueden tener a su disposición una gran cantidad de valiosa información multidimensional, presentada coherentemente como fuente única, confiable y disponible en sus estaciones de trabajo. Así mismo, los usuarios tienen la facilidad de contar con herramientas que les son familiares para manipular y evaluar la información obtenida en el DW, tales como: hojas de cálculo, procesadores de texto, software de análisis de datos, software de análisis estadístico, reportes, tableros, etc.
- Permite la toma de decisiones estratégicas y tácticas.

1.2.2. DESVENTAJAS

A continuación se enumerarán algunas de las desventajas más comunes que se pueden presentar en la implementación de un Data Warehousing (Bouman & Dongen, 2009; Dario, 2009; Rainardi, 2008; Wrembel & Koncilia, 2006):

- Requiere una gran inversión, debido a que su correcta construcción no es tarea sencilla y consume muchos recursos, además, su misma implementación implica desde la adquisición de herramientas de consulta y análisis, hasta la capacitación de los usuarios.
- Existe resistencia al cambio por parte de los usuarios.
- Los beneficios del almacén de datos son apreciados en el mediano y largo plazo.
- Este punto deriva del anterior, y básicamente se refiere a que no todos los usuarios confiarán en el DW en una primera instancia, pero sí lo harán una vez que comprueben su efectividad y ventajas. Además, su correcta utilización surge de la propia experiencia.
- Si se incluyen datos propios y confidenciales de clientes, proveedores, entre otros, el depósito de datos atentará contra la privacidad de los mismos, ya que cualquier usuario podrá tener acceso a ellos.
- Infravaloración de los recursos necesarios para la captura, carga y almacenamiento de los datos.
- Infravaloración del esfuerzo necesario para su diseño y creación.
- Incremento continuo de los requerimientos del usuario.
- Subestimación de las capacidades que puede brindar la correcta utilización del DW y de las herramientas de BI en general.

1.2.3. REDUNDANCIA

Debido a que el DW recibe información histórica de diferentes fuentes, sencillamente se podría suponer que existe una repetición de datos masiva entre el ambiente DW y el operacional. Por supuesto, este razonamiento es superficial y erróneo, de hecho, hay una mínima redundancia de datos entre ambos ambientes.

Para entender claramente lo antes expuesto, se debe considerar lo siguiente:

- Los datos del ambiente operacional se filtran antes de pertenecer al DW. Existen muchos datos que nunca ingresarán, ya que no conforman información necesaria o suficientemente relevante para la toma de decisiones.
- El horizonte de tiempo es muy diferente entre los dos ambientes.
- El almacén de datos contiene un resumen de la información que no se encuentra en el ambiente operacional.
- Los datos experimentan una considerable transformación, antes de ser cargados al DW. La mayor parte de los datos se alteran significativamente al ser seleccionados, consolidados y movidos al depósito.

En vista de estos factores, se puede afirmar que, la redundancia encontrada al cotejar los datos de ambos ambientes es mínima, ya que generalmente resulta en un porcentaje menor del 1%. La arquitectura general de un almacén de datos se resume en la figura 1.2.

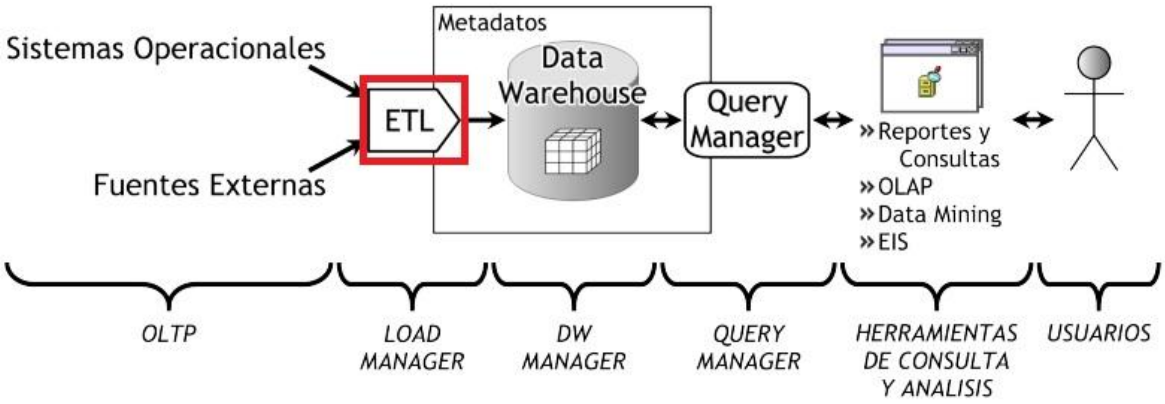

Figura 1.2. Arquitectura general de un almacén de datos. Fuente: (Dario, 2009)

En la figura se ha destacado el componente encargado de los procesos de Extracción, Transformación y Carga de datos a un almacén de datos. Este proceso constituye en centro de la esta investigación y es sobre él que se han formulados las preguntas de investigación.

1.3. PROCESOS ETL: KETTEL

1.3.1. HISTORIA DE KETTLE

En el año 2001, el belga Matt Casters empezó el desarrollo de una herramienta para uso personal, consciente de las dificultades que había tenido durante su experiencia laboral como constructor de Data Warehouse para la integración de sistemas. Durante los siguientes años, fue desarrollando la

herramienta, primero utilizando Java y su librería gráfica AWT, para finalmente pasar a SWT. La herramienta fue añadiendo funcionalidades, acceso a bases de datos, tratamiento de ficheros y componentes hasta llegar a 2004 con la versión 1.2. El proyecto fue subido a Javaforge, donde la gente podía descargárselo y utilizarlo. En la versión 2.0 se incluyo un sistema de plugins para permitir el desarrollo de conectores de Kettle con otros sistemas (como SAP) y en 2005 fue liberado el código y puesto a disposición de todos en Javaforge. El proyecto creció con rapidez y la comunidad se involucro en su desarrollo con mucha actividad, hasta entrar dentro de la orbita de Pentaho (al ser vendido por el autor), que lo incluyo como herramienta ETL (Extract, Transform & Load) en su suite de productos. Matt Caster ha estado desde entonces trabajando en Pentaho y desarrollando su arquitectura como parte del equipo de Pentaho, interveniendo en las diferentes versiones hasta llegar a la 3.2 (y el desarrollo de la nueva versión 4.0, que saldrá durante 2010).

El nombre de Kettle viene de KDE Extraction, Transportation, Transformation and Loading Environment, pues originariamente la herramienta iba a ser escrita para KDE, el famoso escritorio de Linux. El producto ha sido renombrado como Pentaho Data Integration y a partir de ahora nos referiremos a el como PDI (Bouman & Dongen, 2009).

1.3.2. PROGRAMAS QUE FORMAN LA HERRAMIENTA

PDI esta formado por un conjunto de herramientas, cada una con un propósito específico.

- **Spoon**: es la herramienta gráfica que nos permite el diseño de las transformaciones y trabajos. Incluye opciones para pre-visualizar y testear los elementos desarrollados. Es la principal herramienta de trabajo de PDI y con la que construiremos y validaremos nuestros procesos ETL.
- **Pan**: es la herramienta que nos permite la ejecución de las transformaciones diseñadas en Spoon (bien desde un fichero o desde el repositorio). Nos permite desde la línea de comandos preparar la ejecución mediante scripts.
- **Kitchen**: similar a Pan, pero para ejecutar los trabajos o jobs.
- **Carte**: es un pequeño servidor web que permite la ejecución remota de transformaciones y jobs.

1.3.3. TRABAJANDO CON FICHEROS XML O REPOSITORIO

Cuando trabajamos con Spoon, tenemos dos formas de guardar los elementos que vamos diseñando (ver figura 1.3):

- **Repositorio**: disponemos de una base de datos, con una estructura especial, donde son guardados las transformaciones y trabajos construidos. Puede ser útil para el trabajo en equipo y para disponer de un lugar centralizado donde se va registrando todo lo realizado.
- **Ficheros**: las transformaciones y trabajos son guardados a nivel del sistema de ficheros, en archivos XML (con extensión .ktr para las transformaciones y .kjb para los jobs). Cada transformación y trabajo tiene un fichero asociado, que incluye en formato XML el metadata que define su comportamiento.

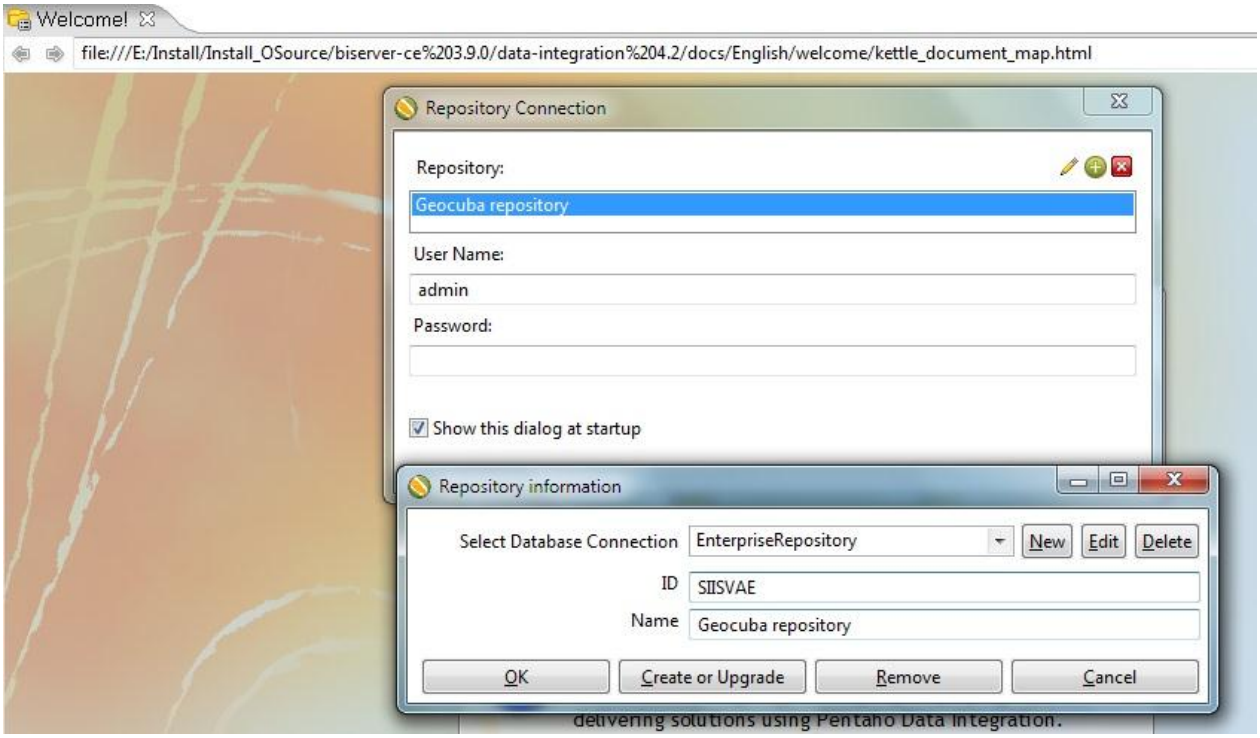

Figura 1.3. Trabajo con el repositorio de Pentaho Data Integration (PDI). Fuente: repositorio SIISVAE©2005-2011

Aunque seleccionemos uno u otro tipo de repositorio, siempre tendremos la opción de convertir de uno a otro modo utilizando componentes de PDI. Veremos un ejemplo de conversión del repositorio cuando terminemos el diseño de los procesos ETL. No se puede trabajar simultáneamente con los dos métodos, por lo que siempre habrá que elegir uno en concreto (Bouman & Dongen, 2009).

1.3.4. CONCEPTO DE TRANSFORMACIÓN

La transformación es el elemento básico de diseño de los procesos ETL en PDI. Una transformación se compone de pasos o steps, que están enlazados entre si a través de los saltos o hops. Los pasos son el elemento más pequeño dentro de las transformaciones. Los saltos constituyen el elemento a través del cual fluye la información entre los diferentes pasos (siempre es la salida de un paso y la entrada de otro). En el ejemplo de la imagen, en el primer paso estamos recuperando registros de una tabla de la base de datos, y los registros recuperados van siendo transmitidos a los siguientes pasos a través del salto, y se van realizando operaciones sobre los datos con los diferentes pasos incluidos (ver figura 1.4).

Figura 1.4. Ejemplo de una transformación en PDI. Fuente: repositorio SIISVAE©2005-2011

Se cuenta con un amplio repertorio disponible de pasos que nos permiten abordar casi cualquier necesidad en el diseño de nuestros procesos de integración de datos. Los pasos están agrupados por categorías y cada uno de ellos esta diseñado para cumplir una función determinada. Cada paso tiene una ventana de configuración específica, donde se determina los elementos a tratar y su forma de comportamiento. Una transformación no es ningún programa ni un ejecutable, simplemente es un conjunto de metadatos en XML que le indican al motor de PDI las acciones a realizar. La figura 1.5 resume las herramientas y componentes del PDI (Fabbri, Gaál, McCammon, & North Atlantic Treaty Organization. Scientific Affairs Division., 2002; Rainardi, 2008).

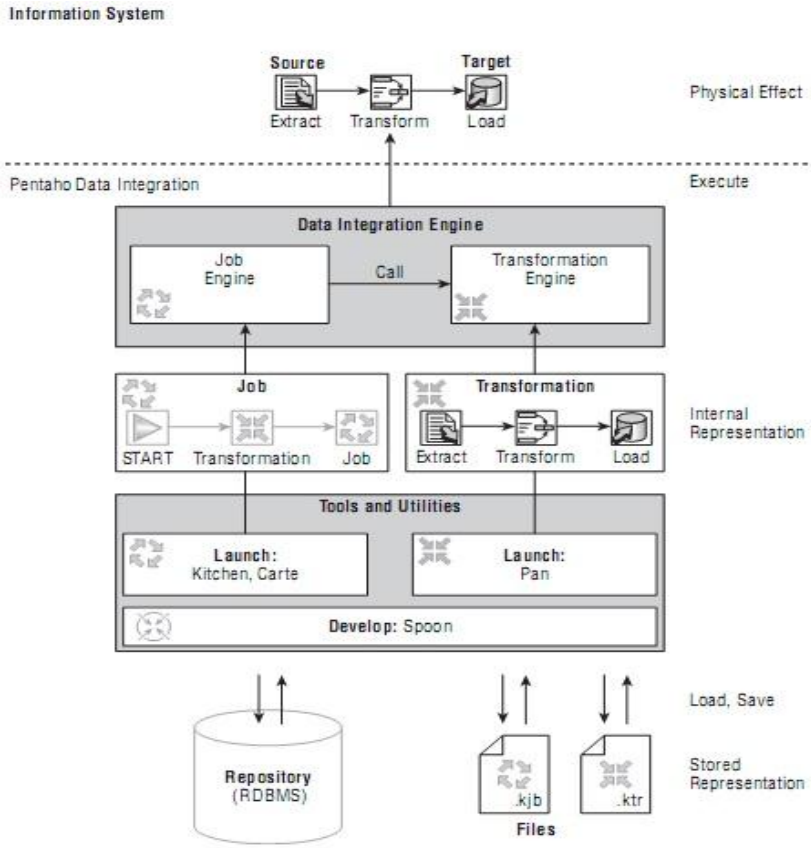

Figura 1.5. Herramientas y componentes del PDI. Fuente: (Bouman & Dongen, 2009)

1.3.5. CONCEPTO DE TRABAJO O JOB

Un trabajo o job es similar al concepto de proceso. Un proceso es un conjunto sencillo o complejo de tareas con el objetivo de realizar una acción determinada. En los trabajos podemos utilizar pasos específicos (que son diferentes a los disponibles en las transformaciones) como para recibir un fichero vía ftp, mandar un email, ejecutar un comando, entre otros. Además, podemos ejecutar una o varias transformaciones de las que hayamos diseñado y orquestar una secuencia de ejecución de ellas. Los trabajos estarían en un nivel superior a las transformaciones.

Los saltos o hops entre los componentes de un job indican el orden de ejecución de cada uno de ellos (no empezando la ejecución del elemento siguiente hasta que el anterior no ha concluido). El paso de un componente del job a otro también puede ser condicional, según si el resultado de ejecución ha sido correcto o no (tal y como vemos en el ejemplo de la imagen). Al igual que las transformaciones, un job no es ningún programa, es también un conjunto de metadatos en XML, que le describen al motor de PDI la forma de realizar las diferentes acciones (Bouman & Dongen, 2009).

1.3.6. INTERFAZ DE USUARIO

La interfaz de usuario es muy sencilla, disponiendo de dos perspectivas. Una de visualización (View), donde vemos los componentes que forman el job o la transformación, y otra de diseño (Design), donde vemos los pasos disponibles. Según estemos trabajando con transformaciones o con trabajos, los steps disponibles irán cambiando. En la imagen, se puede ver la perspectiva Diseño (ver figura 1.6). A la izquierda tenemos los diferentes pasos que iremos arrastrando a la sección de la derecha (grid de diseño). Los pasos tanto de transformaciones como de trabajos los iremos enlazando con los correspondientes saltos.

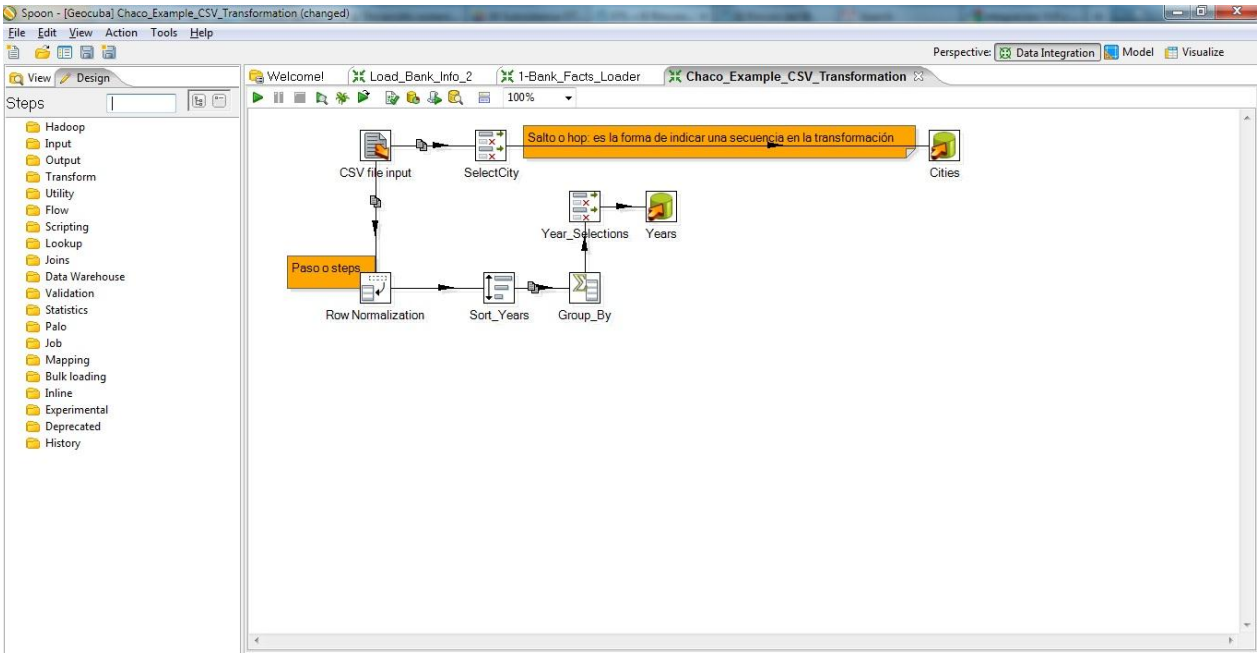

Figura 1.6. Interfaz de usuario PDI perspectiva de diseño. Fuente: repositorio SIISVAE©2005-2011

1.3.7. PASOS (STEPS) DISPONIBLES PARA LAS TRANSFORMACIONES

Se dispone de un amplio conjunto de pasos para las transformaciones (ver figura 1.7).

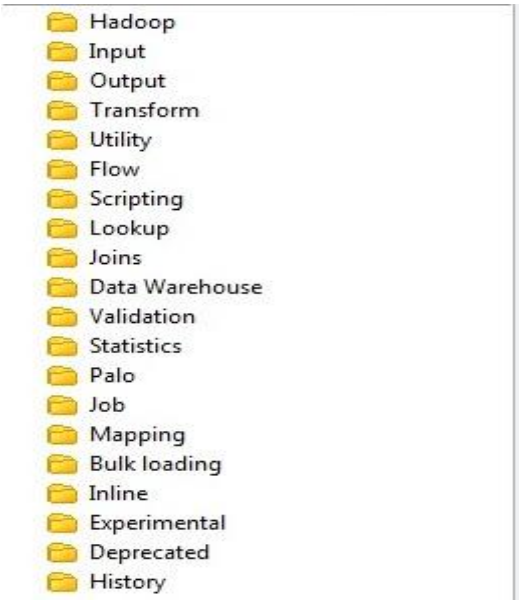

Figura 1.7. Pasos (steps) disponibles en PDI.

En cada una de las categorías, se dispone de varios pasos. Cada una de las categorías, representan acciones generales, en tanto las especificidades quedan implementadas en cada uno de los pasos que contiene.

Así la categoría de Input agrupa todos los conectores del PDI para realizar la extracción de datos desde diferentes fuentes (Access, CVS files, Excel, XML, LDAP, Mondrian, Text files, google analytics, entre otros varios incluyendo conectores para SAP®). En todos los casos, cada paso tiene detalles específicos para modelar la interacción con el origen de datos (en términos de los procesos de integración, cada paso implementa un Wrapper[6] para cada fuente).

Para una descripción detallada de cada paso, ver el Anexo 1 (la descripción de cada paso, está por el momento disponible solo en idioma inglés).

La última versión del PDI incorpora otras dos perspectivas: la de modelación de cubos OLAP (Modeler), la vista para diseñar el reporte (Reporting), y la vista preliminar (Visualize). Las figuras 1.8, 1.9 y 1.10 muestran estás vistas en el mismo orden en que fueron nombradas[7].

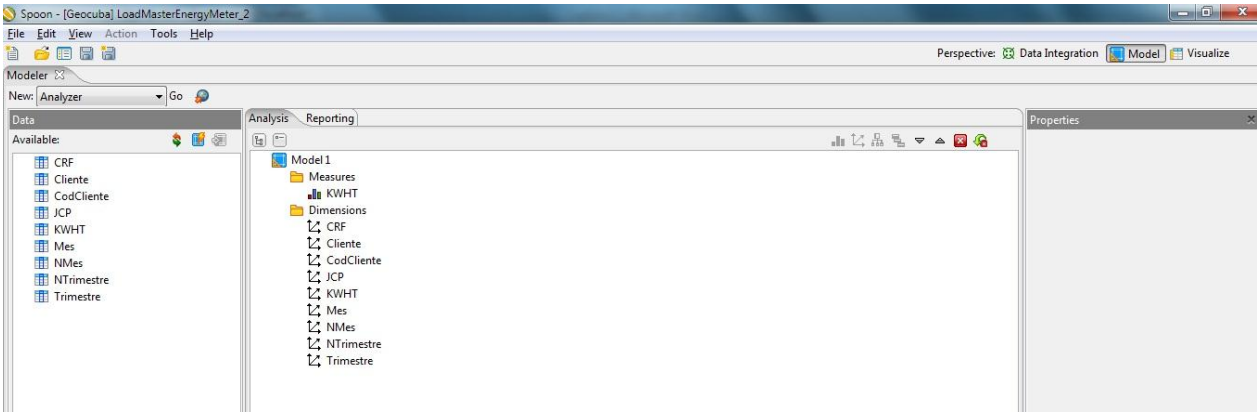

Figura 1.8. Perspectiva del modelador (Modeler). Fuente: repositorio SIISVAE©2011

[6] Un Wrapper (envoltura) aísla los detalles de manipulación de los datos en las fuentes, implementando una interfaz específica en cada una de ellas y una común para todos los bróker de datos (PDI en este caso).

[7] En todos los casos, el ejemplo está construido sobre el caso de estudio real de la investigación.

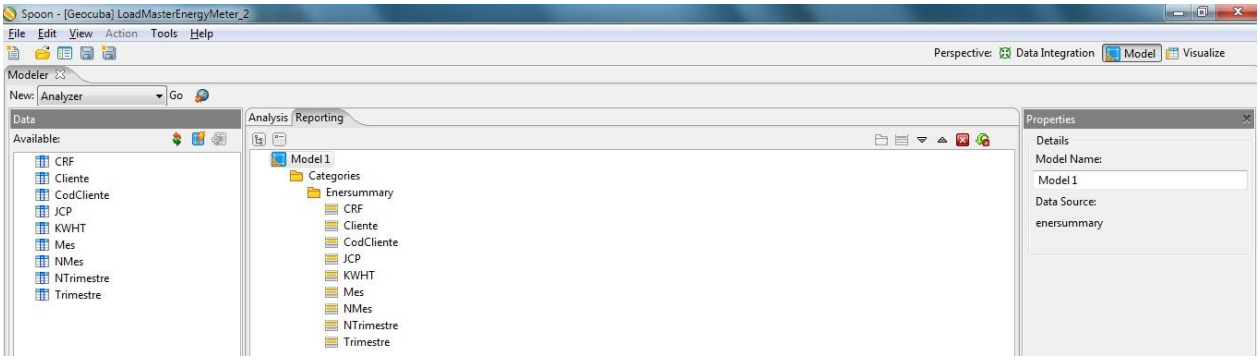

Figura 1.9. Perspectiva para el diseño de reportes (Reporting). Fuente: repositorio SIISVAE© 2011

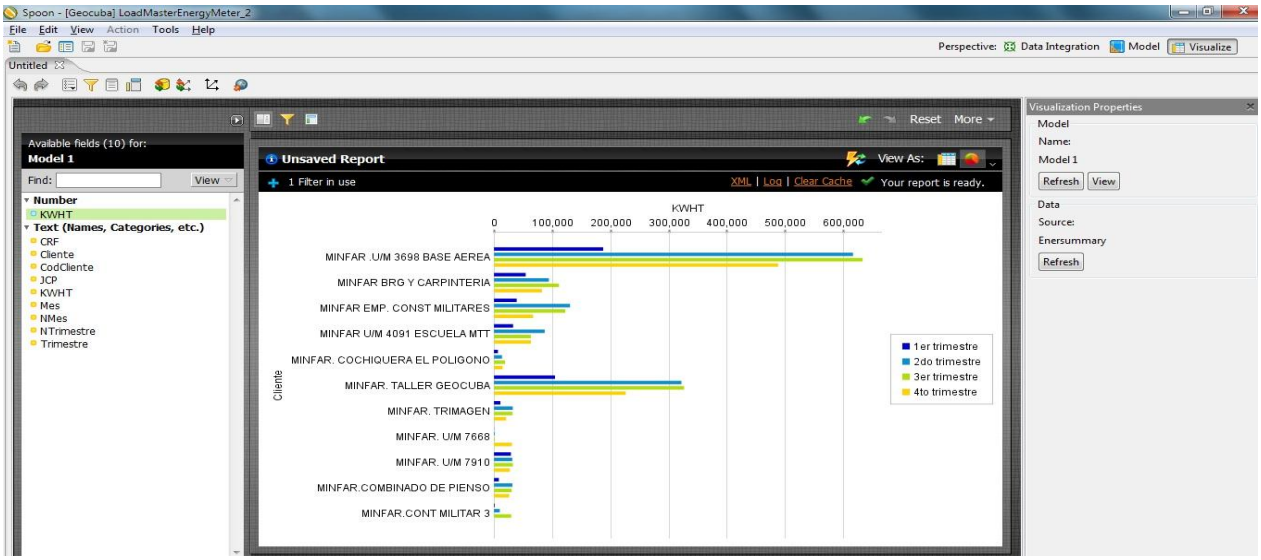

Figura 1.10. Perspectiva de visualización (Visualize). Fuente: repositorio SIISVAE© 2011

Las posibilidades que incorpora esta nueva característica del PDI (la integración de estas tres perspectivas) redundan en ahorros sustanciales de tiempo para la construcción de una solución, su puesta a punto y s despliegue. Las versiones anteriores de la Suite de Pentaho (versión Community) contenían las mismas herramientas pero por separado, adicionando largos tiempos y hacían considerablemente "más escarpada la curva de aprendizaje" con la herramientas disponibles la construir soluciones de Inteligencia de Negocio (en lo adelante BI, por sus siglas en idioma inglés).

1.3.8. IMPLEMENTACIÓN DE UN PROCESO ETL (EXTRACT TRANSFORM LOADING)

El proceso ETL los datos desde sus orígenes en los sistemas operacionales, representa un sub-proceso dentro del proceso de construcción de una solución de BI usando almacenes de datos (en lo adelante, DW por sus siglas en idioma inglés). En la mayoría de los casos prácticos, es a través de estos procesos que se consiguen las metas de integración de datos en los proyectos de BI.

Los retos y desafíos de los procesos de integración, escapan de los objetivos y alcances de esta investigación. En su lugar, se presentarán las herramientas disponibles en PDI para su realización, haciéndolos coincidir –en lo posible, con los subsistemas identificados por Kimball (2004) para este tipo de procesos. Tal y como quedó establecido en el objetivo específico 2, en base a la experiencia acumulada por el grupo SIISVAE©2011, se identificarán en base a estos subsistemas los patrones de

transformaciones que constituyen buenas prácticas en el diseño, implementación y despliegue de estos procesos.

1.4. SUBSISTEMAS PARA LOS PROCESOS ETL

Los procesos ETL se estiman consuman alrededor del 70% del tiempo y los esfuerzos en los proyectos de BI, desarrollados sobre DW. La explicación que la mayoría de los diseñadores y miembros de equipos de proyectos BI esgrimen ante estas estadísticas, se resume en que estos procesos dependen generalmente de: de las fuentes de datos originales, la idiosincrasia de los datos (sus significados intrínsecos), los lenguajes de scripting, las herramientas para construir procesos ETL, las habilidades de los equipos internos, y finalmente de las consultas y herramientas de reporte que los usuarios finales tienen disponibles (jcurtod, 2012).

Después de 18 meses de análisis e investigaciones, Kimball (2004) logró identificar 38 subsistemas que son necesarios en todos los procesos de ETL. La ventaja de estos subsistemas es que una vez identificados, es posible incluirlos en los proyectos de integración de datos a la manera de "Buenas Prácticas", reduciendo el tiempo de identificación y desarrollo de este tipo de soluciones.

Los 38 subsistemas de Kimball (2004)

1. **Sistemas de extracción**: adaptadores de fuentes de datos (programación de trabajos para push/pull/dribble). Usualmente incluyen actividades de filtrado y ordenamiento de datos en los orígenes, conversión de tipos de datos propietarios, etiquetado de datos para transferirlos a los sistemas de ETL.
2. **Cambios en los sistemas de captura de datos**: lectores de trazas de los sistemas operacionales, fechas y sus secuencias, números de filtrado, comparación en los sistemas ETL en base a CRC (Cyclical Redundancy Control).
3. **Sistema de perfilado de datos**: análisis de las propiedades de las columnas de datos, descubriendo o haciendo inferencias sobre los dominios de datos, así como análisis estructural identificando posibles relaciones entre llaves candidatas y primarias, análisis de reglas de datos, y análisis de las reglas.
4. **Sistema de limpieza de datos**: usualmente parte de un diccionario para comprobar los nombres de los individuos u organizaciones, incluye también catálogos de productos o nombre de locaciones. Eliminación de datos duplicados (muchas veces se utilizan reglas fuzzy). Mantenimiento de referencias de respaldo (por ejemplo llaves naturales) para todas las fuentes originales que se utilizan. Los análisis de supervivencia (surviving) utiliza lógicas especializadas para fusiones de datos que preservan campos específicos para las versiones finales.
5. **Conformación de datos**: identificación y hacer cumplir para las dimensiones de conformada por atributos especiales y llenado de las medidas de la tabla de hechos (aspectos básicos de la integración de datos sobre múltiples fuentes de datos).
6. **Ensamblaje la auditoría de las dimensiones**: ensamblar los contextos de los metadatos alrededor de la carga de la tabla de hechos de tal forma que el contexto de los metadatos puedan ser adjuntado a la tabla de hechos como una dimensión normal.
7. **Manipular la ventana de calidad**: los procesos ETL aplican sistemáticamente pruebas en línea para comprobar los flujos de datos en busca de problemas de calidad. Estas pruebas alimentan los subsistemas de manejo de errores (ver el subsistema 8).

8. **Manejo de errores**: es el subsistema que se encarga de reportar y responder a todos los errores del proceso ETL. Usualmente incluye lógicas para manejar varias clases de errores y los mecanismos para manejar los errores del proceso en tiempo real, además del monitoreo de la calidad del proceso.
9. **Sistema de creación de claves subrogadas**: mecanismo robusto una corriente de llaves subrogadas, independiente en cada dimensión. Son independientes también de las instancias de los sistemas operacionales, disponibles para servir a clientes distribuidos.
10. **Procesador de dimensiones que cambian lentamente**: lógica de transformación para manejar tres tipos de cambios que se pueden presentar en los atributos de una dimensión. Tipo I: se sobreescribe el nuevo valor; Tipo II: se crea un nuevo registro y el Tipo III: se crea un nuevo campo.
11. **Manipulador de inserción/actualización tardía a las dimensiones**: lógicas de inserción y actualización para los cambios en una dimensión que son retrasadas cuando arriban al almacén de datos.
12. **Constructor de dimensiones fijas jerárquicas**: sistema de validación, mantenimiento y comprobación de todas los tipos de relaciones one-to-many en las dimensiones con jerarquías (esquemas copos de nieve).
13. **Constructor de dimensiones variables jerárquicas**: sistema de validación, mantenimiento y comprobación de todas las formas dinámicas de relaciones jerárquicas en las dimensiones (sin importar sus niveles). Ejemplos típicos resultan los organigramas empresariales y la típica explosión de necesidades de los sistemas MRP I, II y III.
14. **Constructor tablas puentes para dimensiones multievaluadas**: creación y mantenimiento de tablas asociativas (tablas puentes) usadas para describir relaciones many-to-many entre dimensiones. Pueden incluir sistemas de ponderación para describir roles en determinadas situaciones.
15. **Constructor de dimensiones tipo salto**: creación y mantenimiento de dimensiones consistentes en bajas cardinalidades (usualmente para mantener información complementaria), etiquetas e indicadores en la mayoría de sistemas de producción.
16. **Cargador para la granularidad de la tabla de hechos**: sistema para actualizar las transacciones granuladas en la tabla de hechos, incluye la manipulación de los índices y las particiones. Normalmente se trata de la adición de nuevos registros para los datos recientes. Utiliza tuberías de llaves subrogadas (ver el subsistema 19).
17. **Cargador de instantáneas granulares para la tabla de hechos**: sistema para actualizar periódicamente instantáneas de la granularidad de la tabla de hechos (incluye la manipulación de índices y particiones). Incluye frecuentemente estrategias de sobre-escrituras para actualizaciones incrementales de los periodos más actuales de la tabla de hechos. Utiliza tubería de llaves subrogadas (ver subsistema 19).
18. **Cargador acumulativo de instantáneas granulares de la tabla de hechos**: sistema para la actualización acumulativa de instantáneas granulares de la tabla de hechos (incluye la manipulación de índices y particiones) y actualiza las llaves extranjeras de ambas dimensiones, además de acumular las medidas de los hechos. Utiliza tuberías para las llaves subrogadas (ver subsistema 19).
19. **Tuberías de llaves subrogadas**: tubería, proceso concurrente (muti-hilo) para remplazar las llaves de los datos de entrada de los sistemas operacionales por nuevas llaves que solo son válidas en el DW. Este subsistema está implementado en la totalidad de las herramientas ETL disponibles en el

mercado (implementar su control y ejecución manualmente, es un proceso complejo que puede conducir a errores y hacer ilegible el DW).

20. **Manipulador de inserción/actualización tardía en la tabla de hechos**: lógica para la inserción y actualización de registros que han sido retrasados en la carga al DW. Al igual que en subsistema 11, los retrasos pueden deberse a necesidades específicas, como puede ser el caso de agrupamientos o fusiones para calcular medidas agregadas en la tabla de hechos.
21. **Constructor de agregados**: creación y mantenimiento de la estructura física del DW (conocido como agregados). Los agregados son usados conjuntamente con las facilidades de consultas, así como para mejorar el desempeño de las consultas. Incluye los agregados independientes y las vistas (la Suite de Pentaho incluye una herramienta para el diseño de los agregados).
22. **Constructor de cubos multidimensionales**: creación y mantenimiento del esquema de estrella para la carga los cubos multidimensionales (OLAP), incluyen preparación especial para las dimensiones jerárquicas, dependiendo de los requerimientos la tecnología específica en la que será implementado el cubo (ejemplo: Mondrian).
23. **Constructor de particiones en tiempo real**: lógica especial para cada tipo de tabla de hechos (ver subsistemas 16, 17, y 18) que mantiene una partición "caliente" (actualizada) en memoria que contiene únicamente los registros más frescos del almacén (los relacionados con la última actualización).
24. **Sistema de gestión de dimensiones**: sistema de administración para la gestión de las dimensiones, crean réplicas de las dimensiones desde una localización central hacia todos los proveedores de tablas de hechos. Similar al subsistema 25.
25. **Sistema de aprovisionamiento para tablas de hechos**: sistema de administración para los proveedores de tablas de hechos, que reciben las dimensiones de sus sistemas proveedores (subsistema 24). Incluye sustitución de llaves locales, control de versión de las dimensiones, y gestión de cambios en los agregados.
26. **Programador de trabajos**: sistema para programar y lanzar procesos ETL. Pueden esperar por una amplia variedad de condiciones, incluyendo dependencias entre trabajos y tienen la capacidad de enviar alertas.
27. **Monitor de flujo de trabajo**: tableros de control y sistema de reporte para todos los trabajos ejecutados por el programador de trabajos. Incluye estadísticas como número de registros procesados, resumen de errores, y acciones emprendidas.
28. **Sistemas de recuperación y reinicio**: sistemas encargados de resumir los trabajos que son abortados por el programador de trabajos. Además pueden restaurar (o hacer un back-up) de un trabajo completo (ETL) y reiniciarlo. Este subsistema depende significativamente de los sistemas de respaldo (ver subsistema 36).
29. **Sistema de paralelización**: sistemas que sacan provecho de la existencia de múltiples procesadores en los ordenadores y servidores. Múltiples procesadores pueden sacar ventaja al manejar flujos de datos. Algunas condiciones específicas de las ETL invocan procesamiento paralelo de forma automática, como por ejemplo cuando no es necesario escribir en disco, o esperar porque se verifique una condición en medio de una transformación.
30. **Sistema para el escalado de problemas**: sistema adicional manual para identificar condiciones de error y ofrecer un adecuado nivel de solución e interacción. Incluye desde mecanismos simples de

registro de eventos de error, operadores de notificación, notificación a supervisores, hasta desarrollo de subsistemas específicos de notificación.

31. **Sistema de control de versiones**: capacidad para crear instantáneas consistentes para archivar y recuperar todos los metadatos en una tubería de ETL. Control de entrada-salida en todos los módulos de la ETL y trabajos. Comparación de la capacidad de las fuentes para revelar diferencias entre las versiones.
32. **Sistema de migración de versiones**: desarrollo de pruebas de producción. Mover una tubería (flujo completo) de una ETL hacia una prueba, y posteriormente de regreso al área de desarrollo. La interfaz para los sistemas de control de versión para restaurar una migración (deshacerla). Una interfaz simple para configurar la información de la conexión para una versión completa. Es independiente de la localización de la base de datos para la generación de las llaves subrogadas.
33. **Analizador de dependencias y líneas temporales**: muestra las fuentes físicas de la última transformación, y todas sus acciones de cualquier elemento seleccionado (es posible seleccionar un elemento intermedio de la ETL o uno al final).
34. **Sistema de reporte de cumplimiento**: cumple con los estatutos regulatorios presentando la línea temporal y los principales reportes de los resultados. Facilita la comprobación de que todas las transformaciones y los datos no han sufrido cambios en relación al diseño. Revela quién ha tenido acceso o cambiado algún dato.
35. **Sistema de seguridad**: seguridad administrada por un sistema de roles. Revela quién ha accedido o cambiado los datos, transformaciones o sus metadatos.
36. **Sistema de respaldo**: respalda los datos y metadatos para recuperar, restaurar, razones de seguridad y el cumplimiento de requerimientos específicos.
37. **Gestión del repositorio de metadatos**: sistema para capturar y mantener los metadatos de las ETL incluye todas lógicas de transformación. Incluye el procesamiento de metadatos, metadatos técnicos y metadatos de negocios.
38. **Sistema de gestión de proyectos**: sistema para el mantenimiento y comprobación de todo el desarrollo de las ETL.

Según lo declarado por el propio Kimball (2004) está lista es muy extensa e incluso, pudieran aparecer más subsistemas en los próximos tiempos. A pesar de esto, se reconocen algunas actividades típicas en los procesos ETL. Estas actividades comprenden:

1. **Captura de datos cambiantes**: en muchos casos, la extracción de datos se refiere solo a captar aquellos datos que han cambiado de un periodo a otro (contenido en los subsistemas 2, 10, 11, 12, y 13).
2. **Etiquetado de datos**: algunas veces no es posible o eficiente realizar las transformaciones inmediatamente después de la extracción. Cuando esto sucede, los datos extraídos son movidos a almacenamientos intermedios que son llamados comúnmente área de etiquetado (contenido en el subsistema 1).
3. **Validación de datos**: es el proceso de verificar si los datos captados son correctos, además de reportar los errores encontrados (subsistemas 12 y 13).
4. **Limpieza de datos**: es el proceso de corregir los datos que son captados con errores (subsistemas 4, 12 y 13).

5. **Decodificar y renombrar**: en muchos casos, los datos provenientes de los sistemas operacionales no son útiles para propósitos de reportes, debido que están etiquetados con nombres en código, sobrenombres o acrónimos que no son entendibles por todos los posibles usuarios. Una parte importante de muchas transformaciones se encargan de lidiar con este problema (1).
6. **Agregación**: tradicionalmente las soluciones de BI presenta datos agregados a los usuarios. Muchas veces los agregados son calculados con anterioridad por las mismas transformaciones (en muchas situaciones prácticas, cuando hay cambios importantes de granularidad en la tabla de hechos, se calculan agregados que cambian el nivel de detalle de los hechos) (subsistemas 16, 17 y 18).
7. **Generación de claves y su gestión**: en las dimensiones, los nuevos registros tienen claves únicas que pertenecen al dominio del DW y no al operacional(es) de donde son captados. Estás llaves únicas son conocidas como llaves subrogadas y son generadas y mantenidas en la propia transformación (subsistema 19).
8. **Carga de la tabla de hechos**: la tabla de hechos es llenada tras la captura de nuevos hechos (16, 17 y 18).
9. **Carga y mantenimiento de las dimensiones**: en muchos casos, nuevos hechos implican nuevos registros en las dimensiones (subsistemas 10, 11, 12, 13, 14, 15 y 16).

1.5. CONCLUSIONES PARCIALES

1. Los procesos ETL comparten la suficiente generalidad (con independencia de los contextos y escenarios de procesos de integración) como para ser clasificados y organizados en categorías que facilitan su reutilización.

2. La herramienta ETL disponible en la versión Community de Pentaho, Pentaho Data Integration (PDI) facilita a la vez que generaliza el diseño de procesos ETL. A través de los pasos de disponibles (organizados y clasificados por categorías de tareas) representan una herramienta que es posible integrar al diseño por patrones.

3. Las facilidades de diseño y ejecución de modelos de cubos OLAP disponibles y accesibles desde la propia herramienta PDI, facilitan extraordinariamente el proceso de comprobación de la calidad y efectividad de la transformación diseñada, aportando elementos sólidos para comprobar la eficiencia del diseño guiado por patrones.

4. La generalización y clasificación de los pasos para las transformaciones, unido a la capacidad de diseño y visualización de modelos OLAP disponibles en la herramienta PDI, le convierten en la elección para la comprobación de las respuestas dadas a las preguntas de investigación planteadas.

2. CAPÍTULO 2. IDENTIFICACIÓN DE PATRONES BASADOS EN LOS SUBSISTEMAS PARA PROCESOS ETL

2.1. INTRODUCCIÓN

En el capítulo anterior quedaron expuestos los diferentes subsistemas propuestos por Ralph Kimball para mejorar la implementación en los procesos ETL. Estos 38 subsistemas fueron agrupados por actividades de una manera tipificada, a partir de las relaciones que en transformaciones específicas adquieren. A partir de estos subsistemas se pueden definir una serie de patrones que en un caso concreto, pueden ser reutilizados en la(s) nueva(s) transformación(es). Estos patrones han sido identificados para la herramienta de integración de datos Pentaho Data Integration.

2.2. IDENTIFICACIÓN DE LOS PATRONES EN ACTIVIDADES

Con los 38 subsistemas de Kimball están representados los fraccionamientos que componen los procesos ETL, pero la contemplación separada de cada uno de ellos dificulta su comprensión como parte de un proceso integrador de datos. Las actividades típicas de procesos ETL permiten una identificación más acertada de los patrones que existan en cualquier transformación. Estos patrones encontrados específicamente para transformaciones de datos en Pentaho Data Integration, son planteados como la descripción de un algoritmo de pasos o cajas que, siguiendo el orden expuesto, pueden resultar reutilizables para la ejecución de un proceso ETL con un objetivo común o parecido, siempre y cuando existan intereses parciales compatibles con los entradas/resultados de los patrones.

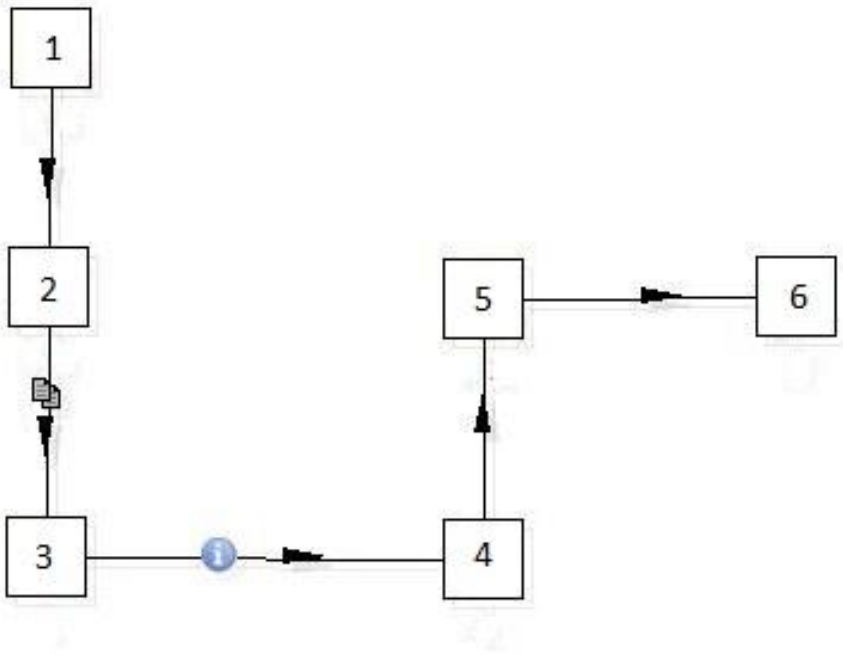

Figura 2.1. Ejemplo de concatenación de pasos en Pentaho Data Integration

2.2.1. PATRONES DE DECODIFICAR Y RENOMBRAR

La decodificación ocurre cuando los valores de un campo en el sistema de la fuente se trazan a otros valores en el sistema objetivo. El cambio de nominación ocurre cuando el nombre de un campo particular en la fuente se da un nuevo nombre en el sistema de blanco. Estas actividades no incorporan

información extra en el sentido formal, sin embargo, estas actividades pueden ayudar a hacer datos más accesibles al usuario final (ver figura 2.2).

El registro de las fechas es realizado de forma que cada una representa un día específico. Para diversos usos de la información, se hace necesario una separación o descomposición de la fecha en los distintos atributos que de ella se pueden llegar a necesitar. La desagregación es realizable a través de la herramienta **Calculator**, ingresando la fecha como campo a trabajar, una de sus opciones de cómputo es la identificación de los distintos elementos de tiempo que lo componen[8]. De este paso surgen tantos nuevos campos como descomposiciones que de la fecha se hagan. Estos nuevos campos serán renombrados con los valores numéricos que le son asignados en el paso **Calculator,** con los nombres habituales manejados por el usuario, a través del paso **Value Mapper** (que será tratado posteriormente)**.**

Figura 2.2. Proceso de descomposición de fecha. Fuente: repositorio SIISVAE©2005-2011

Se pueden usar tantas descomposiciones como sean necesarias de acuerdo al interés del modelador.

2.2.2. PATRONES DE ETIQUETADO DE DATOS

La extracción de datos puede tener considerable impacto en el funcionamiento y disponibilidad del sistema operacional de la fuente. Existe un requisito determinante de mantener la cantidad de tiempo de extracción a un mínimo en un intento por disminuir el impacto en operaciones normales tales como entrada de datos.

Además de la duración real del proceso de la extracción, la materia de la sincronización puede también entrar en el juego. En algunos casos, los datos de varios sistemas operacionales distintos necesitan ser combinado antes de que se completen el almacén de datos.

Como solución a estos problemas, los datos generalmente se almacenan temporalmente en una zona de espera supuesta inmediatamente después de la extracción. En la mayoría de los casos, la zona de etiquetado de datos es simplemente una base de datos relacional diseñada específicamente para servir como almacenador intermediario entre los sistemas de la fuente y almacén de datos. Los datos al ser almacenados en un sistema de base de datos distinto, los índices que pueden ayudar a mejorar el funcionamiento del subsecuente procesamiento de datos se puede agregar libremente sin alterar el sistema de la fuente. Una transformación más completa, que crea una tabla para almacenar toda la información relacionada con la fecha (ver figura 2.3).

[8] Se tratará como herramienta de transformación, paso o caja de transformación a las distintas opciones para transformar disponible en el Kettel.

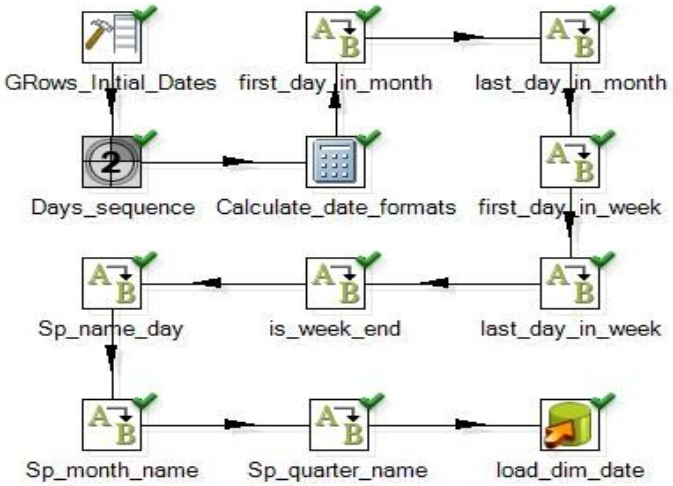

Figura 2.3. Descomposición de fecha a partir de una secuencia. Fuente: repositorio SIISVAE©2005-2011

Esta transformación representa un patrón que puede utilizarse siempre que se desee tener toda la información posible relacionada con la fechas, incluso para varios años, dependiendo del límite del generador de secuencia de días.

En esta transformación se utilizan los mismos pasos que en el patrón anterior. La diferencia radica en que se utiliza un paso adicional (en relación al patrón anterior) para crear una secuencia de días válidas dentro de un año. A diferencia también, en lugar de leer la fecha desde un sistema operacional (relacionado con un hecho a modelar en el almacén), este patrón utiliza un generador de secuencia de días, para completar un año (o pueden completarse más, dependiendo de los intereses del modelador). Este patrón tiene más utilidad para generar en el área de etiquetado, la dimensión temporal del almacén que su uso directo en una transformación sobre fuentes de datos directamente.

2.2.3. PATRONES DE AGREGACIÓN

Los hechos (expresados como datos en los sistemas operaciones) ocurrieron en un momento siempre anterior a la fecha en que se realizan los procesos ETL. Para propósitos de auditorías de los datos cargados al almacén, etiquetar los datos cargados con la fecha en que se realiza este proceso. Para ello luego de recoger los datos de la fuente (archivos en distintos formatos), se añade el paso **Get System Info**, que debe ser editado creando un nuevo campo que sea calculado, seleccionando la opción de **Today 00:00:00** para identificar el limite final de la fecha como el comienzo del día que se ejecuta el proceso ETL. Estas dos acciones se entrelazan con el lenguaje usado en el paso **Modified Java Script Value**, donde se crea una variable que identifica la diferencia entre las fechas, de la fuente de datos y la fecha actual del sistema. Resulta útil en este proceso el uso de la función ***diff = dateDiff(x,y)***, siendo ***diff*** la variable de la diferencia, **(x,y)** los nombres de los campos que contiene las fechas respectivas[9] (ver figura 2.4).

[9] Hay otra forma de hacer esto sin necesidad de recurrir a la programación de un paso en javascript. Para un ejemplo, consultar la documentación del repositorio de SIISVAE© figura 2.3

Figura 2.4. Proceso de identificación de intervalos de fecha. Fuente: repositorio SIISVAE©2005-2011

Las heterogeneidades que pueden existir entre las fechas de los dos campos, deben ser eliminadas antes de encontrar la separación entre ellas, para evitar errores de compatibilidad de los formatos de fecha.

Otras transformaciones más complejas generan patrones asociados a las tareas básicas que resuelven. Así por ejemplo, cuando en los sistemas operacionales las bases están muy normalizadas y se sigue el enfoque de diseño de Kimball (esquema de Estrella y desnormalización para la construcción del almacén), es necesario desnormalizar los datos hasta conseguir los niveles de redundancia necesarios para soportar la granularidad de la tabla de hechos. Este es el caso que se resume a continuación[10] (ver figura 2.5).

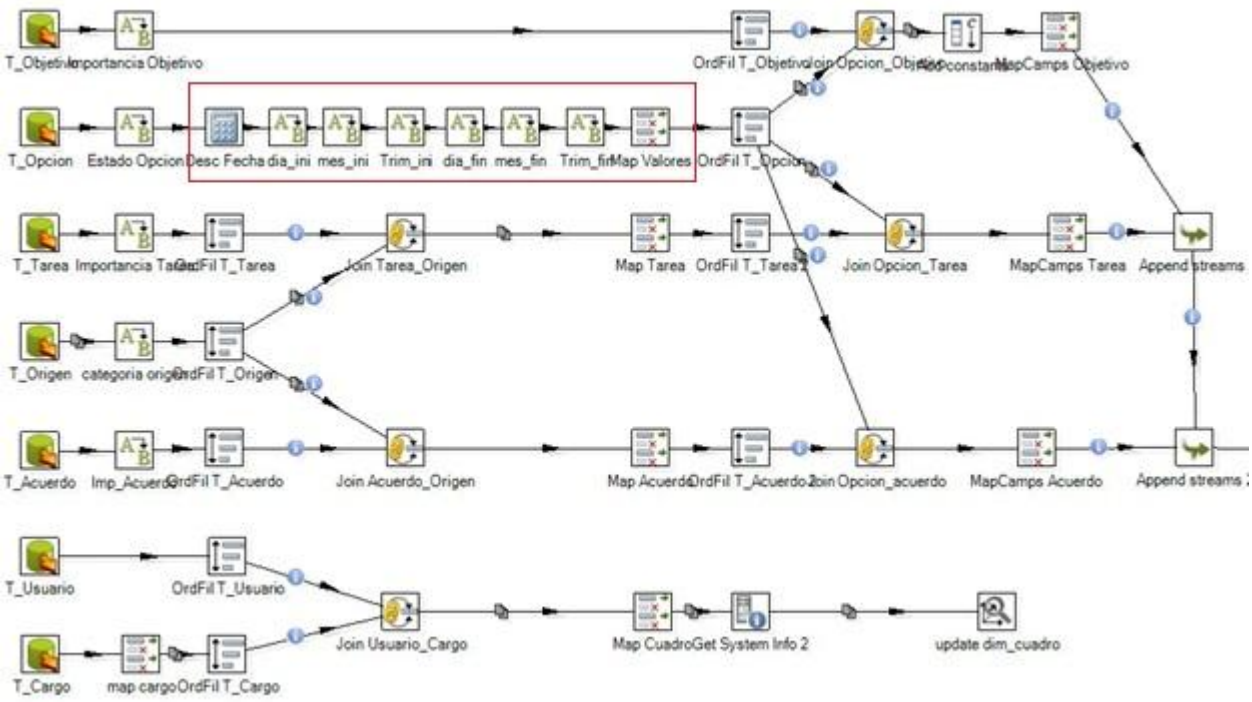

Figura 2.5. Proceso de desnormalización. Fuente: repositorio SIISVAE©2005-2011

En este caso, el sistema operacional estaba en 5ta forma normal. La transformación desnormaliza hasta conseguir un flujo redundante que soporte la granularidad de los hechos. Se trata de un sistema para el control del cumplimiento de las tareas, objetivos y acuerdos de los cuadros de un consejo de dirección[11].

[10] El lector atento identificará otros patrones tratados anteriormente en el ejemplo de la figura 2.5, como es el caso del tratamiento y descomposición del campo fecha (señalada con un recuadro rojo).

[11] Los detalles de la empresa a la que pertenece el sistema operacional son omitidos para proteger la integridad de la empresa y los acuerdos de confidencialidad de los datos y sus relaciones.

Cada una de las tablas de origen de dato representa una entidad del mundo real modelado (el resto de las tablas del sistema, representan clasificadores y nomencladores generales).

Con la excepción de las tablas T_Usuario y T_Cuadro que después de un proceso de unión y mapeo resultan organizadas para reflejar la acción completa: usuario del sistema *que coloca una acción (Tarea, Objetivo, Acuerdo) sobre un cuadro*, las demás tablas requieren de mayor manipulación. En todos los casos las manipulaciones están orientadas a conseguir la misma regla expresiva enunciada anteriormente (señalada en este mismo párrafo en cursivas).

El patrón resultante en este caso corresponde a los procesos de separación y unión que se manifiestan en todas las tablas. El proceso es sencillo en su estructura pero complejo en la semántica que resulta. La sencillez proviene de la necesidad de "desarmar" la estructura normalizada de la base de datos de origen y expresar la misma información de forma redundante. Así –por ejemplo, una opción puede ser o bien una Tarea o un Objetivo o un Acuerdo, es por esto que aparece un Merge Join inmediatamente después de la manipulación en las tablas T_Tarea, T_Opción, T_Objetivo y T_Acuerdo (incluyendo por supuesto el origen de cada una de ellas almacenado en T_Origen). En cada caso y de acuerdo a la información semántica que quiere adicionarse, hay pasos que se encargan de mapear los códigos numéricos provenientes de la normalización del operacional en etiquetas semánticas en el almacén (este proceso puede ser revertido posteriormente).

Una recomendación adicional proviene del proceso de limpieza de datos. En este sistema, los datos recogidos desde la interfaz de usuario, son almacenados en el operacional con toda la información del etiquetado de HTML utilizado en la interfaz. Esto genera una cantidad inmanejable de símbolos y códigos especiales que no tienen significado alguno en el almacén y lo que es peor: encierran información importante que si es requerida (sirva de ejemplo el siguiente fragmento de HTML: * Cuadro: Reinaldo de Jesús *).

En estos casos, la adición de un paso de limpieza es imprescindible. El cuadro 1 resume las expresiones regulares utilizadas para limpiar las cadenas de las etiquetas HTML.

Cuadro 1. Expresiones regulares para limpieza de etiquetas HTML. Fuente: repositorio SIISVAE©2010

```
//se elimina las cadenas identificadas como tipicas de informacion innecesaria

result = result.replace((/{[^{-}]+}/g),'');    ///// elimina good {bad} good
result = result.replace((/>\s*</g),'');                    ///// elimina > <
result = result.replace((/>(Normal|false|ES|X-NONE|MicrosoftInternetExplorer4| |\d+)</g),'');
///// elimina
result = result.replace((/ /g),'');
result = result.replace((/<br>/g),'');
result = result.replace((/>[^<->]+MsoNormalTable[^<->]+</g),'');
result = result.replace((/>[^a-zA-Z0-9_]+st[0-9]+[^a-zA-Z0-9_]+</g),'');   // elimina los > st1\:* < y
similares, ej > st25\:* <
// eliminar marcadores
result = result.replace((/<>/g),' '); // reemplaza los <> por " "
result = result.replace((/>/g),''); // reemplaza los > por ""
result = result.replace((/</g),' '); // reemplaza los < por ""
```

El procesamiento del flujo por condiciones, genera un patrón identificable que puede ser resumido y reutilizado también, dada la generalidad que presenta. La figura 2.6 muestra un ejemplo en el que se comprueban tres condiciones.

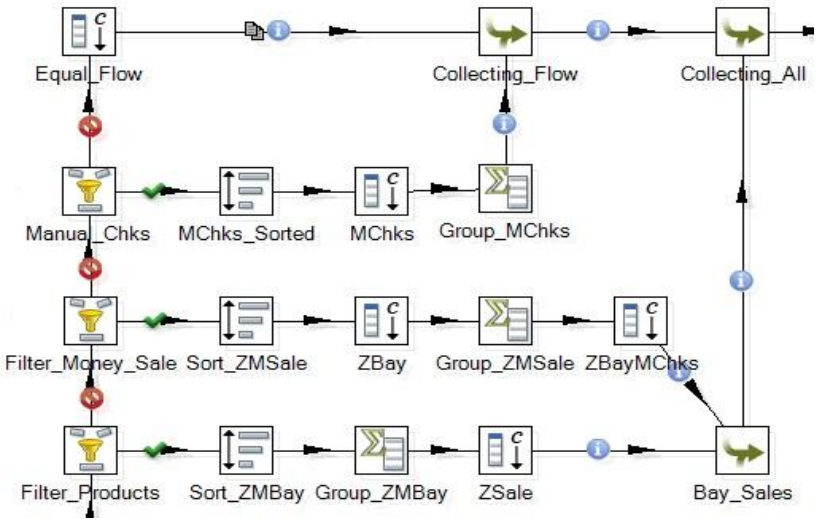

Figura 2.6. Patrón clasificación por comparación. Fuente: repositorio SIISVAE©2005-2011

En este caso el patrón es mucho más sencillo. Se trata de realizar un filtrado de cada una de las filas del flujo de datos. En cada uno de los posibles casos, se ordena el flujo de acuerdo al criterio de búsqueda en la comparación (así los coincidentes serán procesados primero). Posteriormente se realiza una acción de agrupación (en realidad este es el paso que se encarga de clasificar) y posteriormente se adiciona una constante, que tiene la finalidad de mantener constante la cantidad de campos en el flujo. Esto es obligatorio porque producto de la clasificación se agrega uno o más campos que contiene el valor de los cálculos agregados que se realizaron.

Destaca el hecho de que al calcularse solo un valor agregado en cada caso y ser tres los casos inspeccionados, solo se agrega una constante en cada ramificación. Este patrón constituye la representación visual y mediante las combinaciones de los pasos disponibles en Kettel de una condicional múltiple del tipo ***If Then Else***.

2.2.4. PATRONES DE LIMPIEZA DE DATOS

En muchos casos, las posibles situaciones con los datos de fuente son conocidas con antelación, y los procesos se encaminan a corregir los datos que serían inválidos para la herramienta integradora. Proceso conocido como limpieza de datos. Esta puede ser asegurada verificando si los datos cumplan ciertas reglas, desechando o corrigiendo los que no sigan el patrón previsto, fijando los valores prefijados para los datos que falta, eliminando la información que se duplica, normalizando datos para conformar mínimo y los valores máximos, tareas realizables en PDI por su amplia gama de herramientas.

1. Los registros de datos obtenidos con posterioridad a su ocurrencia, plantean el inconveniente de interpretación errónea de la fecha de recogida, como acontecimiento real del hecho (hechos que se contabilizan o registran con posterioridad a la fecha en que son registrados). Para tratar esta situación que generalmente requiere un tratamiento específico, es conveniente el uso de la programación (que

está disponible mediante el empleo del paso **Modified Java Script Value**, donde se especifican los pasos necesarios para cotejar el valor de la fecha de registro con su correspondiente realización. En general, cualquiera que sea el tratamiento a los datos (sean fechas o no) que no sigan una pauta incorporada en los pasos de las transformaciones de Kettel, este es el paso a utilizar, porque es posible extender las funcionalidades a través de la programación (ver figura 2.7).

Figura 2.7. Ajuste de valores de fecha registrados con posterioridad a su ocurrencia. Fuente: repositorio SIISVAE©2005-2011

En los casos en que el valor del campo de fecha que se está trabajando se registra en una fecha fuera del nivel que engloba al mismo campo (entiéndase hora dentro de día, día dentro de mes, mes dentro de año) se requiere una expresión que haga coincidir al campo como el ultimo valor de su nivel anterior.

La toma de información de diversas fuentes implica la posibilidad de recibir datos duplicados, donde la misma entidad se inscribe múltiples veces en un solo sistema, o la misma entidad existe en sistemas múltiples pero no se puede entrecruzar directamente por la carencia de llaves o de referencias, por lo que es necesario un cruzamiento de las tablas en la transformación para así eliminar las filas repetidas.

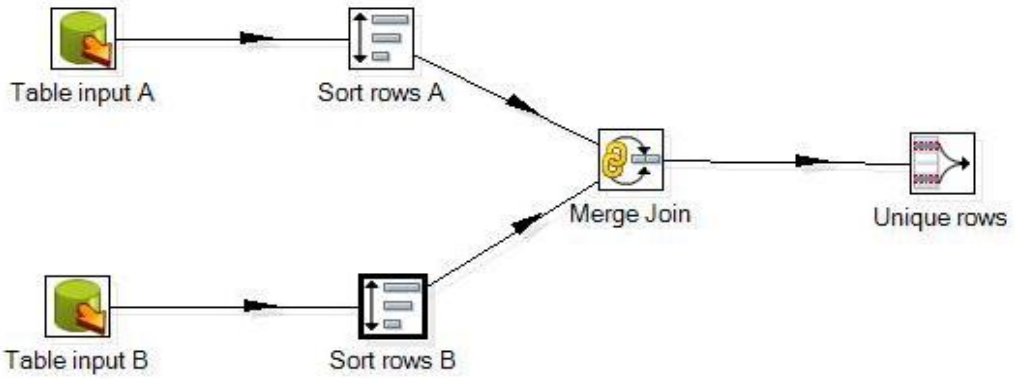

Figura 2.8. Proceso de identificación de intervalos de fecha. Fuente: repositorio SIISVAE©2005-2011

Una vía sencilla y práctica consiste en organizar ambos flujos por separado, basándose en un campo común para ambas entradas. Estos flujos deben ser unidos posteriormente por la opción **Merge Join** por la facilidad que ofrece esta opción para controlar las características de fusión, donde seleccionando la opción INNER se obtiene un flujo dentro del otro. La herramienta de unificar filas **Unique Rows** detecta las filas que tengan los mismos valores y las convierte en una fila única, permitiendo al sistema interpretar el registro como de una sola ocurrencia.

2.2.5. PATRONES DE VALIDACIÓN DE DATOS.

Una vez que se adquieren (y se almacenan posiblemente los datos en una zona de espera), existe generalmente un cierto proceso en el lugar para determinar la validez de los datos. Los datos inválidos se

deben tratar diferentemente que datos válidos, porque pueden corromper la confiabilidad del almacén de datos. La detección de datos inválidos es un requisito previo para tratarlo diferentemente.

En el contexto de ETL y de la validación de datos, los datos se consideran inválidos cuando contienen errores lógicos. Esto ocurre cuando se encuentran los registros que no habrían podido ser incorporados si todas las condiciones ejecutados por el sistema de origen de datos (y su sistema de base de datos subyacente) se hubieran sido hechos cumplir.

Por ejemplo, los datos para campos requeridos pueden faltar, o los valores en un campo pueden contradecir valores en otro campo, por ejemplo cuando una fecha de expedición cae antes de la fecha correspondiente de la orden. No hay manera de determinar validez de los datos sin realmente comprobarla. Si datos de fuente inválidos accidentalmente termina incluida en el almacén de datos, la información errónea puede llegar al usuario final, llevando esto a la desconfianza general del almacén de datos y de los procesos que mantienen la integración de datos.

La validación de datos como parte del proceso de integración de datos tiene una ventaja inmediata. La no detección de datos inválidos proporciona la seguridad que el sistema de la fuente puede ser confiado. Por el contrario su detección ofrece una oportunidad única de mejorar el sistema de la fuente (ver figura 2.9).

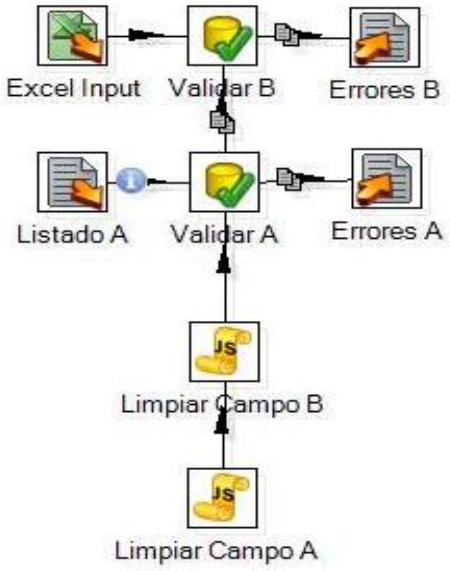

Figura 2.9. Proceso validación de datos a partir de registros ajenos al flujo Fuente: repositorio SIISVAE©2005-2011

La necesidad de validar los datos como parte de su limpieza o de la comprobación de elementos que ausentes o incorrectos dentro de un campo, es resuelto dentro de una serie de pasos sencillos. El campo que posea un listado de elementos únicos -ID de una persona, Código de un producto o cualquier llave sustituta creada- debe ser "normalizado" mediante Java para evitar incongruencias e indeterminaciones que afecten el paso de validación. **Data Validator** permite definir reglas de validación simples para características de los campos en el flujo y permite manejar también los errores, por esto para comprobar la existencia, la falta o incoherencias de elementos en un listado, se convierte en la herramienta idónea.

La validación es hecha de manera que se comprueba a partir de los registros aportados por una forma de entrada de cualquier tipo (referencia de comparación) haciéndolos coincidir con los campos del flujo del sistema, la condición de validación es sencilla: los hechos que coincidan en ambos, continúan en circulación en la transformación. Las no concordancias, pueden ser resumidas y presentadas de varias formas, como en un archivo de tipo texto tal cual aparece en la figura del ejemplo.

2.2.6 PATRONES DE GENERACIÓN DE LLAVES Y SU GESTIÓN

Todas las tablas de la dimensión tienen llaves primarias subrogadas, y las tablas de hechos se ensamblan a las tablas de la dimensión usando solamente referencias a estas llaves. Los valores para estas llaves no se deben derivar de los sistemas de la fuente que alimentan el almacén de datos (con la excepción posible de la tabla de la fecha dimensión). En lugar, deben ser generados como parte del proceso de integración de datos.

Una tabla de dimensión contiene descripciones sobre una entidad o una categoría particular de un negocio. Las dimensiones son uno de los bloques básicos de un almacén de datos. En ellas se pueden organizar los datos en las diferentes perspectivas que se estimen necesarias crear.

Una manera de llenar las dimensiones conservando los datos históricos, se realiza a través del paso **Dimension lookup/update**, que busca en la dimensión si el registro que se está procesando ya está en la dimensión. Si este es el caso, puede el modelador tratarlo de tres formas diferentes, coincidiendo con los tres tipos de **Slowly Changing Dimension** reconocidas por Kimball.

Figura 2.10. Actualización de dimensión conservando valores históricos. Fuente: repositorio SIISVAE©2005-2011

El ejemplo es una muestra de la inserción de valores actualizados en una dimensión sin prescindir de los históricos, donde se establece un rango de fechas que indica el periodo de validez de los datos.

La inserción de la caja **Get System Info** actúa como el indicador de la fecha en que se realiza la transformación y establece el flujo principal de la fecha que como condición debe existir dentro del rango de validez.

2.2.7 PATRONES DE CARGA Y MANTENIMIENTO DE DIMENSIONES

La mayoría de las tablas de la dimensión no son estáticas. Su contenido necesita adaptarse según las adiciones y los cambios que ocurren en los sistemas proveedores de datos. Existen ejemplos simples, tales como nuevos productos que necesiten ser agregados a la tabla de la dimensión del producto. Ejemplos más complejos incluyen la manipulación de varios tipos de dimensión de lento cambio. Almacenar los cambios en las tablas de la dimensión es una de las responsabilidades primeras del proceso de integración de datos.

Parte del procedimiento anterior es aplicable para los registros variables, luego de un periodo de tiempo establecido mediante la inserción de una fecha invariable usando el paso **Add Constants** que marca antes y después del hecho, para ello se establecen los límites de fecha de validez. Al ser un registro cambiante se incluye además un campo que muestra la versión de la inscripción siendo un valor automáticamente incrementado que mantiene un número de revisión para los registros. Además es generado un indicador para señalar cuál de ellos se corresponde con el actual.

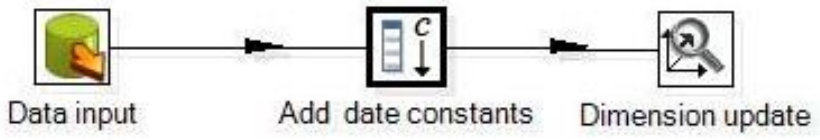

Figura 2.11. Actualización de dimensión cambiante a partir de fecha constante. Fuente: repositorio SIISVAE©2005-2011

2.3. CONCLUSIONES PARCIALES

1. A partir de los 38 subsistemas identificados por Kimball y de la clasificación realizada de estos subsistemas en el marco de esta investigación, fue posible la identificación de patrones de transformación como resultado del análisis y evaluación de las transformaciones disponibles en el repositorio de SIISVAE© 2005-2011.
2. La herramienta PDI a través de las generalizaciones realizadas en cada uno de los pasos disponibles para el diseño de transformaciones, facilita al tiempo que garantiza la suficiente generalidad (para absorber la diversidad de contextos y escenarios de integración), así como la especificidad y expresividad semántica para identificar patrones reutilizables, gracias a la generalidad y diversidad de contextos y aplicaciones posibles a implementar en los procesos ETL.
3. La reusabilidad de los patrones ha quedado comprobada toda vez que han sido identificados atendiendo fundamentalmente a su presencia en varios contextos y escenarios de integración, en los cuales fue convenientemente establecido el propósito de la transformación.
4. De esta generalidad (de los procesos ETL y los pasos disponibles en el PDI) es posible deducir una regla semántica empírica hasta el momento, pero que su consistencia puede ser comprobada vía el mismo proceder utilizado en su derivación (análisis e identificación de patrones). El análisis e identificación de patrones constituye un proceso natural que consigue identificar reglas semánticas que contribuyen a mejorar y potenciar el diseño de procesos ETL guiados por patrones.

3. Capítulo 3: Modelación de transformaciones a partir de patrones de actividades de procesos ETL.

3.1 Introducción.

En este capitulo se construye a partir de los patrones identificados el proceso ETL encargado del ordenamiento histórico y la estructuración de los datos de consumo del sector empresarial de la provincia (denominados mayores consumidores). Por la similitud entre los procesos de gestión (en las etapas de planificación y control dentro del ciclo de gestión de la energía) entre el sector estatal (mayores) y el residencial (menores) se decidió incluir el diseño de la transformación para este último caso, como una comprobación más de la capacidad del diseño guidado por patrones de producir procesos ETL consistentes. La estrategia utilizada se basó en incluir el propio proceso ETL para los mayores como un patrón más disponible para la transformación de los menores.

3.2 Patrones usados en la modelación de las transformaciones.

Para la modelación de las transformaciones fueron seleccionados los patrones que en su conjunto resolvieran el objetivo de la transformación y satisficieran los intereses de las partes implicadas. En la misma las situaciones que plantea con el "desfase" entre el registro y la ocurrencia del hecho, la necesidad de diferenciar campos específicos con valores nulos, además del interés en une presentación sencilla dela fecha y mantener dimensiones y tablas de hechos que coleccionen los reportes necesarios se hizo necesario el uso de los patrones de renombrar y decodificar, limpieza de datos y de carga y mantenimiento de dimensiones.

3.3 Procesos de transformación de los maestros de energía de la provincia Villa Clara (mayores)

La transformación presente realiza las cargas de los maestros de energía de la provincia, para todos los metros contadores de las empresas locales y los Organismos de la Administración Central del Estado (OACE), con el objetivo de determinar posibles irregularidades en el sistema de registro del consumo eléctrico en distintas instituciones de la provincia de Villa Clara. Entre las distintas formas de poder determinar estas alteraciones, la detección de registros de consumo de valor nulo (0) en los metro contadores de las entidades, es una de ellas, dada la imposibilidad de existencia de un centro de trabajo activo que no emplee energía eléctrica (ver figura 3.1).

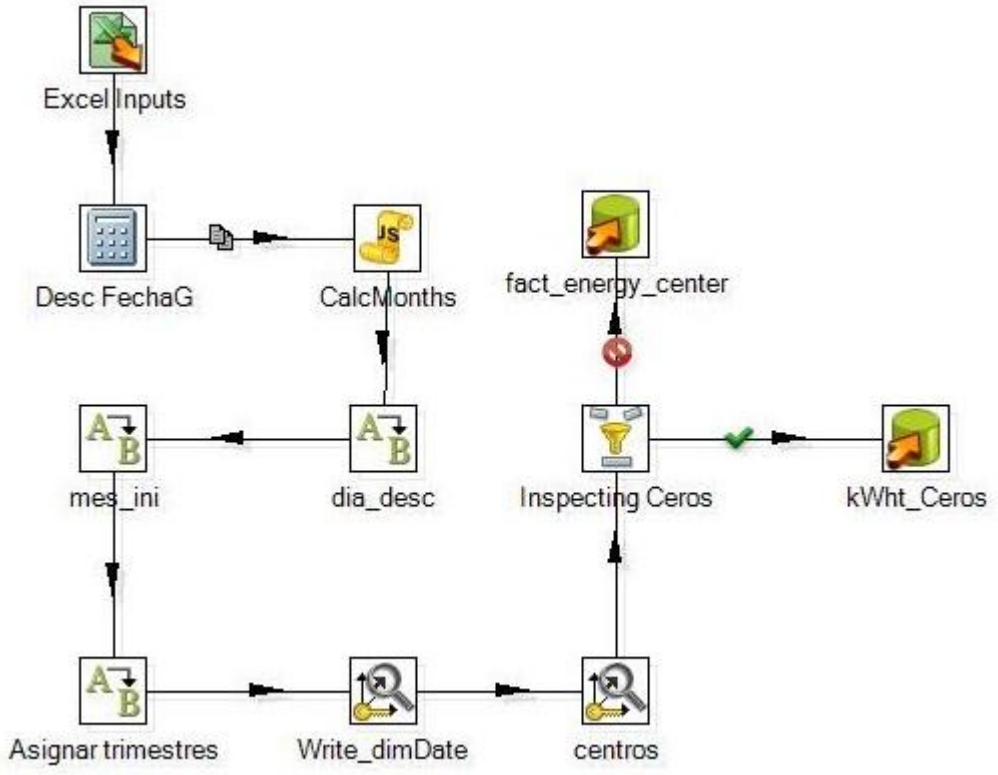

Figura 3.1. Transformación de carga de los maestros de energía de empresas e instituciones estatales. Fuente: SIISVAE©2005-2011

Los registros de las lecturas de cada mes se realizan en el mes posterior, los valores del mes de enero se corresponden a los obtenidos en febrero, esto se repite para el resto de los meses del año (consecuentemente con el procedimiento de lectura y cobro de la energía eléctrica). La extracción de los datos para la transformación se realiza a través de consulta de 12 archivos de Microsoft© Excel™ (cada uno de los cuales, contiene la lectura de todos los clientes en cada mes del año) que cumplen estas características y presentan un dominio extenso de campos relacionados al consumo eléctrico como son: el mes de lectura, el centro de trabajo y su dirección, el circuito y el metro contador, así como el importe en USD o moneda nacional.

Para la facilitación de la transformación de los datos y por la necesidad futura de manejar el campo de mes de forma independiente es aplicable el patrón de transformación de fecha expuesto en la Figura 3.2. Como parte de este patrón, se usa la herramienta Calculator que, empleando el campo que contiene los atributos de fecha en el caso el correspondiente al MES proveniente de las tablas recogidas en archivos de Excel, identifica como forma de cálculo los elementos que conforman la fecha y los descompone en día de la semana, semana, mes, trimestre, año.

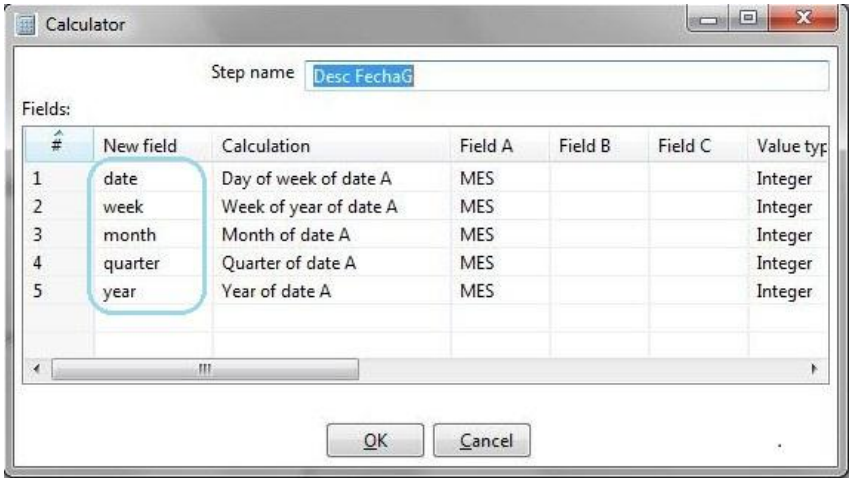

Figura 3.2. Descomposición del campo MES usando la herramienta Calculator. Fuente: SIISVAE©2005-2011

Estos nuevos campos (date, week, month, quarter, year) creados como parte de la descomposición son sumados al flujo principal proveniente de las tablas recogidas.

La lectura del registro eléctrico en el mes posterior de su ocurrencia real plantea la necesidad de indicar al integrador de datos el ajuste a hacer dentro del campo mes, incluso el posible error de interpretación del mes de diciembre cuya lectura correspondería al año posterior. Para ello el uso de ***Modified Java Script Value*** es la forma más efectiva de traducir, valiéndose de la descomposición realizada en el paso anterior de los campos de año y mes, indicando que el valor real del mes es la resta del número resultante menos uno; y que para un valor de año 2012 la fecha debe ser interpretada como el mes 12 del año 2011.

Cuadro 3.1. Código en javascript para actualizar la fecha de consumo

```
var aYear = 2011

if (year.getInteger()!= aYear) {
        month.setValue(12)
        year.setValue(year.getInteger() -1)}
else {
        month.setValue(month.getInteger()-1)}
```

El campo de la fecha cuando deja el paso Calculator se encuentra estructurado de manera que los datos que antes correspondían a la fecha, pertenecen ahora a nuevos campos creados que los configuran de forma separada. Como parte del patrón que estructura las fechas es necesario para su reorganización más limpia y sencilla a una función que etiquete los datos, según los valores que serán utilizados para los análisis mediante el paso ***Value Mapper*** se encarga de asignar los nuevos nombres con los que serán recogidos el día, el mes y el trimestre.

En una primera etapa, los valores del campo del día de la semana que se encuentran ordenados como valores numéricos de valores de 1 a 7 se hacen corresponder con la designación natural para esos días. En esta caja de Kettel se identifica el campo date al que pertenecen los días de la semana, y a partir de

este se crea un nuevo campo (date_name) donde son ubicados los nuevos nombres de los días de la semana (ver figura 3.3).

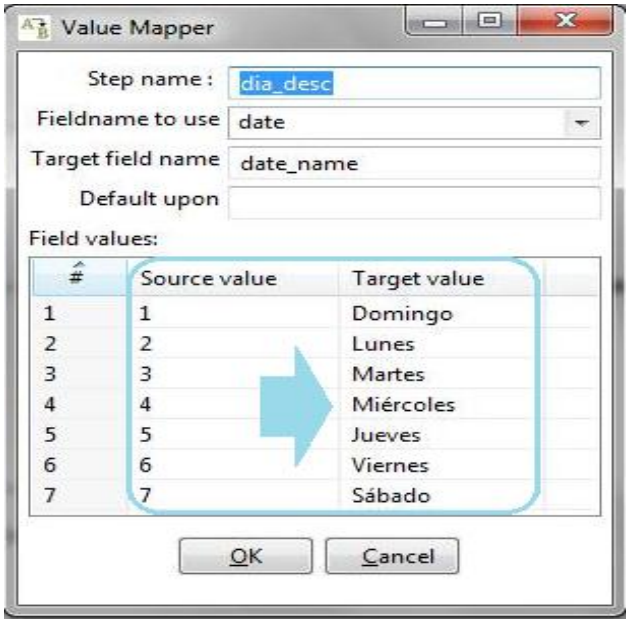

Figura 3.3. Renombramiento de los registros del campo date mediante Value Mapper. Fuente: SIISVAE©2005-2011

Como parte de las etapas de renombramiento el campo creado en **Calculator** esa designado como valores numéricos del 1 al 12 por lo que en la segunda etapa los meses se designan de manera consecutiva siendo enero el valor primero(1) y diciembre el último(12).

La segunda fase del cambio de denominación inserta un nuevo campo (month_name) que es utilizado en la tercera etapa como referencia para asignar a los meses el trimestre al que pertenezcan. Así los primeros tres meses Enero, Febrero y Marzo ocupan las casillas del primer trimestre (1er trimestre), ocurriendo sucesivamente para el resto hasta completar los cuatro trimestres que conforman el año (ver figura 3.4).

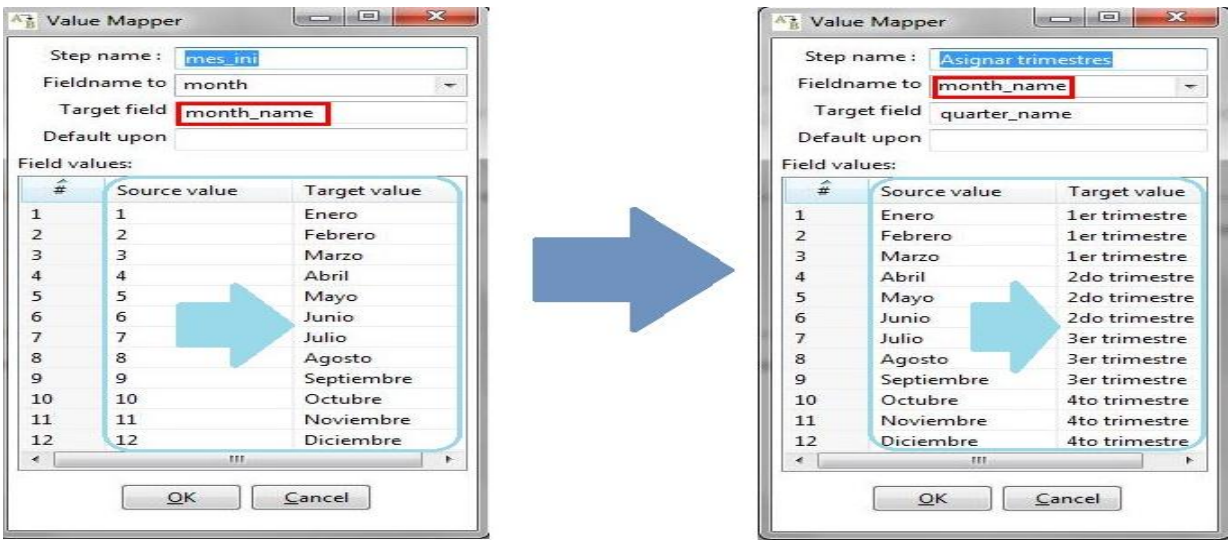

Figura 3.4. Renombrar el valor de mes y asignación de trimestres usando un campo común. Fuente: SIISVAE©2005-2011

La información de los registros también es guardada en una base de datos que conserva solamente los datos actualizados eliminando los viejos. La opción **Combination Lookup\Update** permite esta actualización de la base de datos mediante la creación de llaves subrogadas para cada dimensión a crear. **Combination Lookup\Update** busca en la dimensión las combinaciones de los campos llave que se establecen en su interfaz, y devuelve la llave substituta del registro correspondiente. En el caso de no existir, como ocurre en el ejemplo, el paso genera una nueva llave sustituta e inserta una fila con los campos dominantes y la llave sustituta generada. En todo caso, la llave sustituta se agrega al flujo de salida.

La primera dimensión que pasa este proceso es la dimensión que guarda la fecha, donde la llave sustituta tk_fecha es asignada a los campos que identifican los meses que son analizados, almacenando los valores de día, semana y trimestres a los que pertenecen, así como la fecha de actualización. El campo MES, incorporado por los archivos de entrada, es identificado como Fecha en la nueva dimensión creada en la base de datos ddsenergia.

De igual manera ocurre con la dimensión que almacena la información de los centros laborales que recoge a través de la llave tk_center el nombre del centro, el metraje, el código, su circuito y dirección.

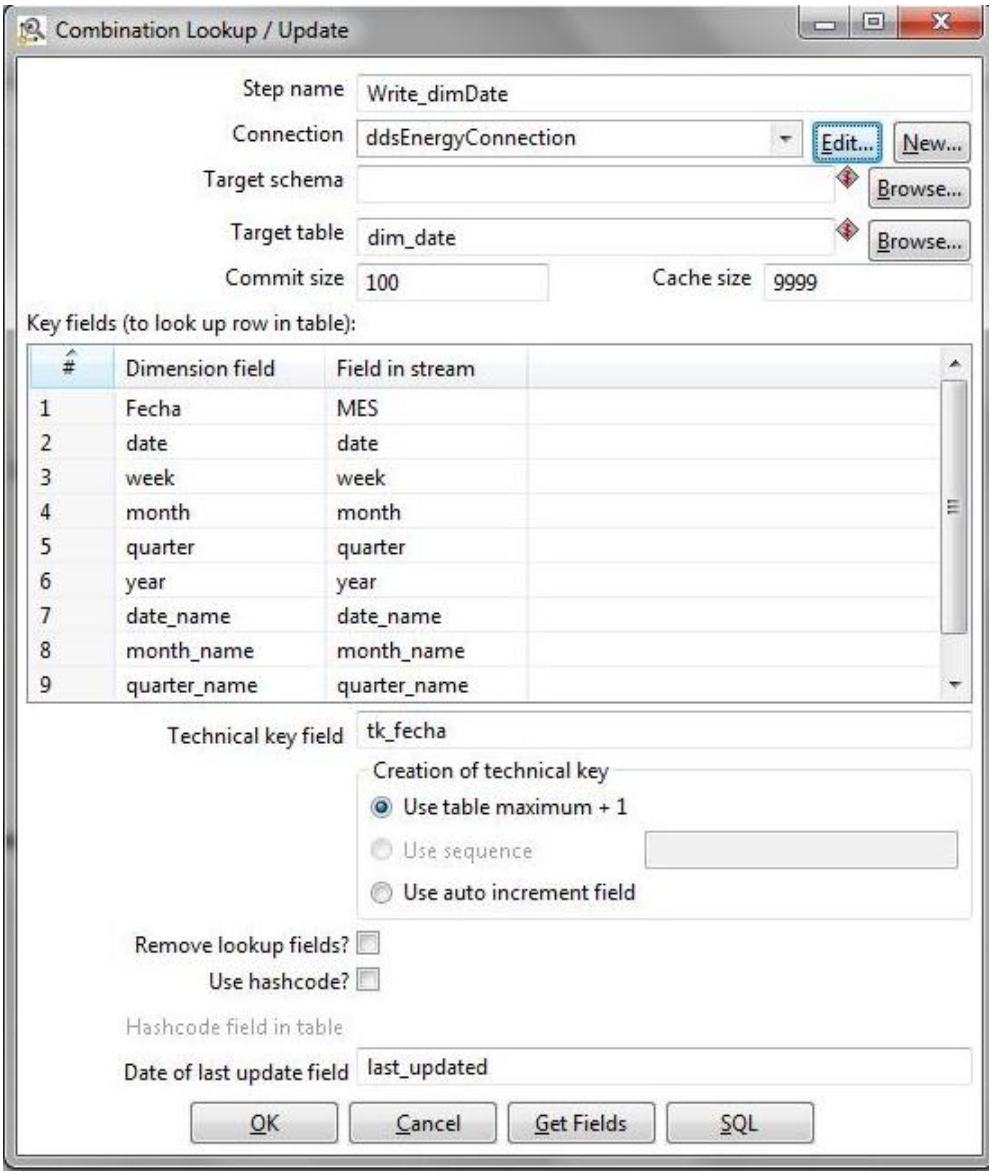

Figura 3.5. Actualización de la dimensión dim_date usando Combination \lookup\update. Fuente: SIISVAE©2005-2011

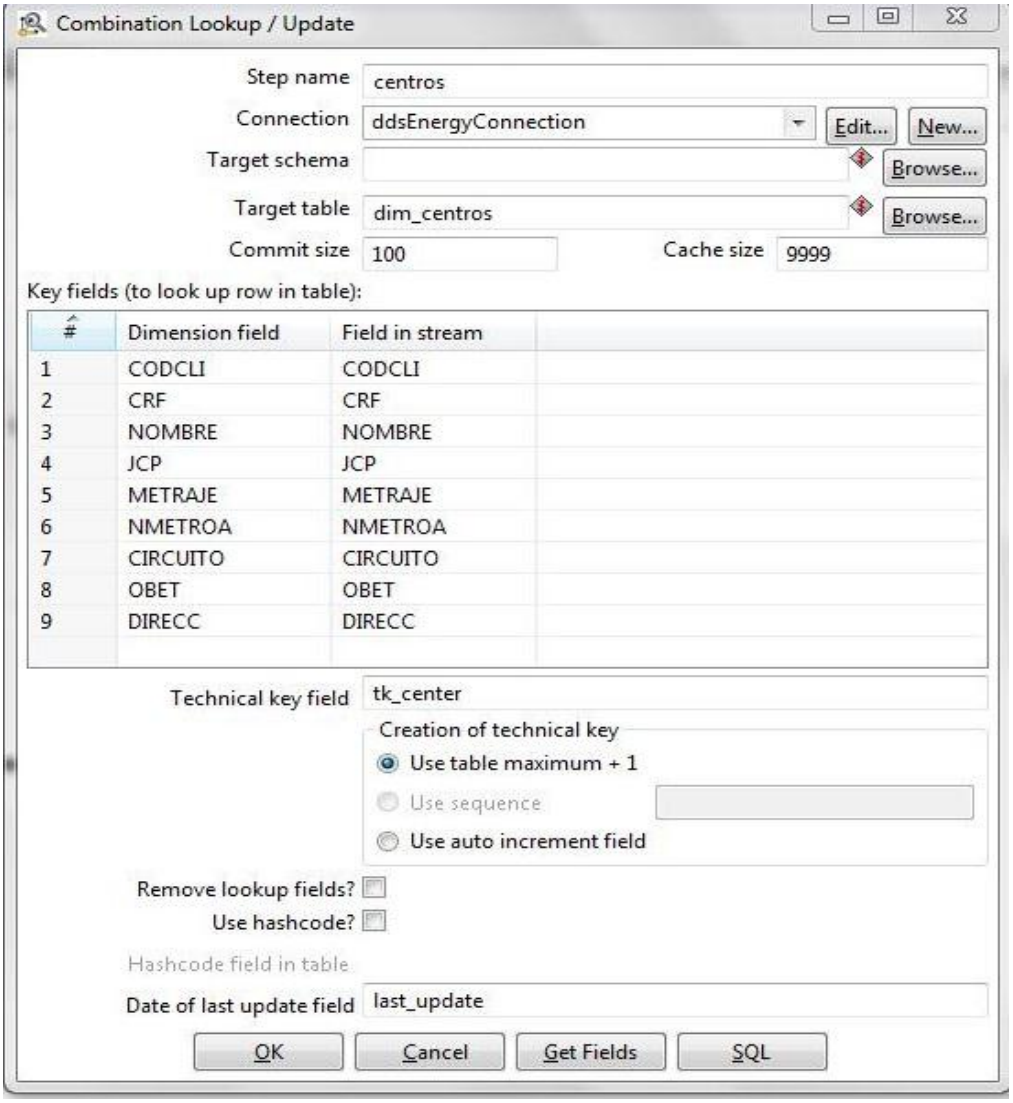

Figura 3.6. Actualización de la dimensión dim_centros mediante Combination lookup\update. Fuente: SIISVAE©2005-2011

La opción de **Combination Lookup\Update** es la herramienta idónea para el almacenamiento de estas dimensiones porque ambas conservan valores únicos de fecha que no se alteran con el tiempo como corresponde a una **Slowly Changing Dimension (SCD) Type I**.

Una vez que el flujo ha sido organizado a través de los pasos anteriores se necesita aun el depurado que cumpla con los objetivos de la transformación, la determinación de lecturas de valor nulo (0) kilowatts. La opción **Filter rows** permite dividir el flujo en dos corrientes según se cumpla o no esta condición. Se escoge el campo referido a las lecturas de kilowatts/hora totales (KWHT) y se establece un comparador de igualdad a cero. Las salidas terminan en tablas que recogen los hechos.

El incumplimiento de la restricción, significando los organismos estatales que tuvieron valores de lecturas expresadas en KWHT, envía los valores a la tabla de hechos **fact_energy** que vincula los valores de las llaves substitutas de cada centro laboral con la fecha de su registro (*tk_fecha* y *tk_center*), además el factor potencia y los kilowatts asociados a la entidad (ver figura 3.7).

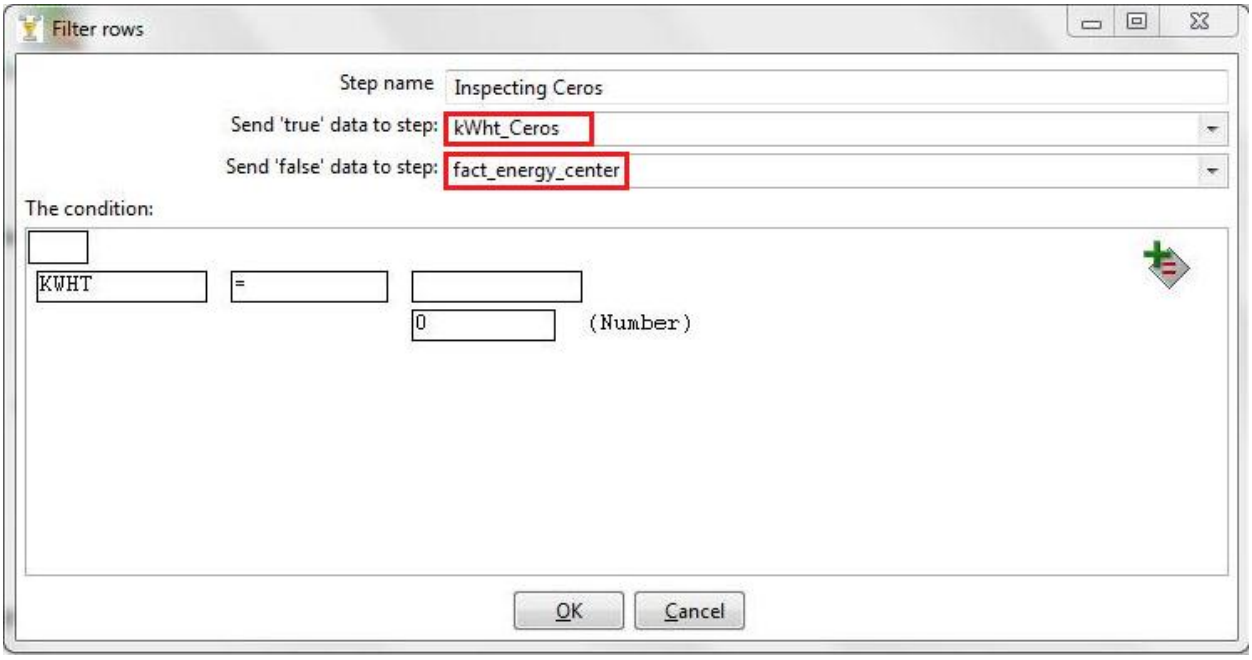

Figura 3.7. Separación del flujo de acuerdo a la condición de gasto nulo en el campo KWHT. Fuente: SIISVAE©2005-2011

El cumplimiento de la condición planteada en el filtrado, objetivo de la transformación para encontrar los consumos nulos, rellena una tabla de hechos **fact_ceros** donde se relacionan solamente los valores de los centros con registros nulos con su fecha específica como únicos datos de interés acerca de estos centros.

3.4 Procesos de transformación de los maestros de energía de la provincia Villa Clara (menores)

Los registros del sector residencial en la provincia Villa Clara son analizados mediante una transformación similar a la expuesta en el epígrafe anterior principalmente por el objetivo común que presentan.

La determinación de los valores nulos de los metros contadores en el sector residencial adquiere considerable importancia por el amplio número de usuarios pertenecientes a este lo que hace casi imprescindible el uso de una herramienta que procese estos datos de forma rápida, además este objetivo permite el uso de los patrones usados previamente (ver figura 3.8).

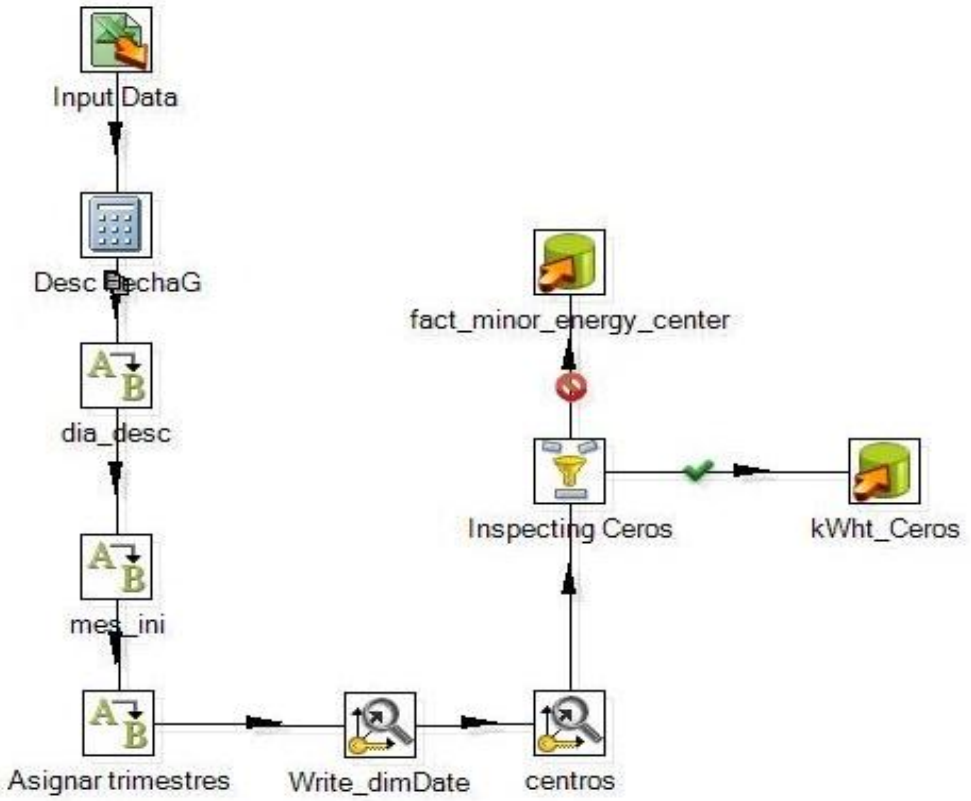

Figura 3.8. Transformación de carga de los maestros de energía de empresas e instituciones estatales

Los registros de los consumos en este sector son recogidos igualmente en 12 archivos de Microsoft© Excel™ correspondientes a cada mes del año que se está analizando .Estos archivos contienen una serie de campos que son de interés para la Oficina de Planificación y Economía entre los que se encuentran los campos pertenecientes al nombre y dirección particular del habitante, la ruta a la que pertenece, la localidad, la tarifa y el consumo para ese mes.

Como parte del patrón de descomposición de la fecha se inserta en la transformación la herramienta **Calculator** que igualmente descompone el campo de fecha correspondiente ***dfech_fact*** en campos de día, semana, mes, trimestre y año. De la misma forma estos nuevos campos son incorporados al flujo de la transformación (ver figura 3.9).

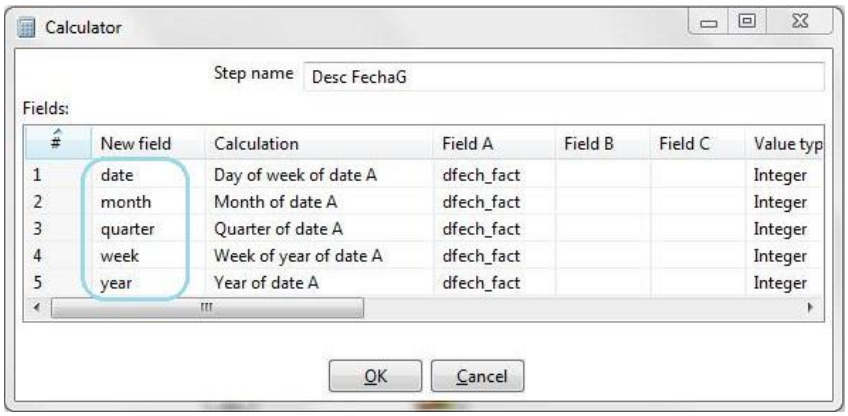

Figura 3.9. Descomposición del campo dfech_fact usando la herramienta Calculator

La diferencia principal entre ambas transformaciones está dada por prescindir del paso Modified Javascript Value, encargado de eliminar la asincronía entre la lectura del consumo y el momento en que se realiza. En el caso de los menores (sector residencial) las lecturas se hacen dentro del mismo mes en que ocurrió el consumo.

Prescindiendo de este paso se continúa análogamente a renombrar los campos creados a partir del paso **Calculator** usando la caja de **Kettel Value Mapper** acometiendo su objetivo en el mismo orden y empleando la misma asociación de campos que en la transformación anterior (ver figura 3.10).

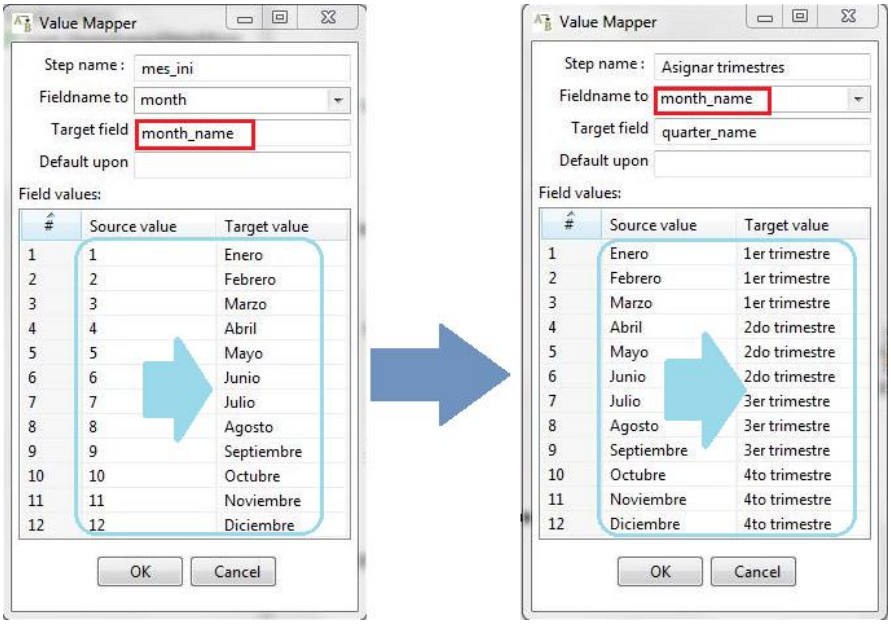

Figura 3.10. Renombrar el valor de mes y asignación de trimestres usando un campo común

Una vez realizados todos estos pasos se procede a la creación de las distintas dimensiones que recogerán los registros de las fechas y los centros. Dada la similitud de los registros entre ambas transformaciones es evidente el tratamiento de ambas dimensiones como SCD Type I.

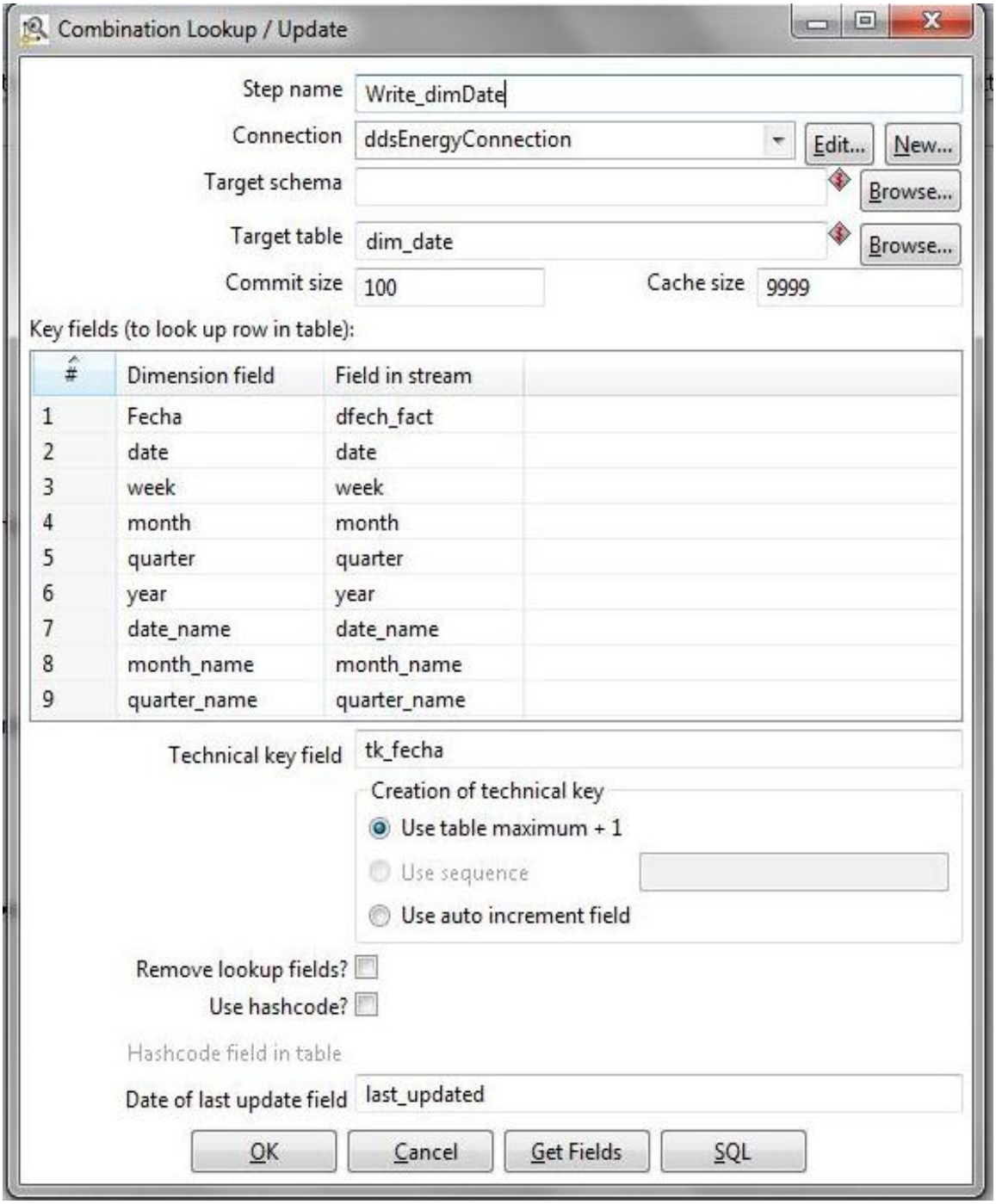

Figura 3.11. Actualización de la dimensión dim_date usando Combination \lookup\update

Se crean dimensiones con los valores asociados a las fechas en la dimensión **DIM_DATE** donde son asignadas de igual forma llaves subrogadas ***tk_date*** a los registros. La dimensión **DIM_MINOR_CENTROS** asociada a las residencias recoge bajo las llaves ***tk_centros*** los datos pertenecientes a cada una de ellas como su código, ruta, folio, nombre, dirección y tarifa.

La validación ocurre comparando los datos recogidos en el campo ***nconsumo*** con el valor nulo (0) dividiendo el flujo en dependencia al cumplimiento de la condición establecida en el paso **Filter Rows**. Igualmente los valores son recogidos en tablas de hecho de acuerdo a este criterio.

Los registros hallados de valor nulo pasan a llenar la tabla de hechos **FACT_MINOR_CEROS** donde se recogen las llaves subrogadas de cada centro con la fecha de ocurrencia de registro de este valor nulo. Las residencias con valores existentes de consumo son almacenadas junto a la fecha que le corresponde, junto a los registros de consumo y el importe que este significó.

CONCLUSIONES

1. La investigación dio respuesta a las dos preguntas planteadas. Los procesos ETL tiene la **suficiente** generalidad para extraer de ellos patrones reutilizables y la necesaria flexibilidad para combinar estos patrones en nuevos diseños, en contextos y escenarios de integración diferente.
2. Los principales resultados prácticos obtenidos por la investigación dan cumplimiento al objetivo general planteado. El análisis e identificación de patrones de transformaciones facilitó el diseño del proceso ETL para el ordenamiento histórico y la estructuración de los consumos de energía del sector empresarial en la provincia de Villa Clara.
3. Los patrones de procesos ETL identificados facilitaron el diseño de nuevos procesos. En particular, la ETL para los mayores consumidores de energía resultó un patrón reutilizable para el caso del sector residencial (menores consumidores).
4. Los resultados obtenidos permiten concluir que el proceso iterativo de analizar e identificar patrones en procesos ETL, acelera –vía la composición de proceso ETL a partir de los propios patrones, el diseño de soluciones de integración aún en contextos diferentes.
5. El principal resultado práctico obtenido como resultado del almacenamiento, clasificación y documentaciones de los patrones, es la mejora progresiva de la comprensión por parte de diseñadores y encargados de negocios, de los procesos de integración que conducen. Tal condición **TRANSFORMA** una solución de este tipo de un repositorio de transformaciones y procesos ETL a un repositorio de conocimientos.

RECOMENDACIONES

1. Divulgar los resultados de la investigación, buscando promover el intercambio de repositorios de transformaciones para mejorar la base de patrones disponibles.
2. Realizar estudios encaminados a agregar información semántica (significados) a los patrones identificados, con vista a facilitar el trabajo de herramientas CASES.
3. Estudiar otras herramientas para integración de datos, a fin de identificar patrones de intercambio entre ellas que faciliten el intercambio de conocimientos proveniente de los patrones de ETL.

BIBLIOGRAFÍA

1. Abramowicz, Witold. (2007). *Business information systems : 10th international conference, BIS 2007, Poznan, Poland, April 25-27, 2007 : proceedings*. New York: Springer.
2. Alshawi, Sarmad, Saez-Pujol, Isabel, & Irani, Zahir. (2003). Data warehousing in decision support for pharmaceutical R&D supply chain. *International Journal of Information Management, 23*(3), 259-268.
3. Ballard, Chuck, International Business Machines Corporation. International Technical Support Organization., & Books24x7 Inc. (2006). Dimensional modeling in a business intelligence environment 1st.
4. Bouman, Roland, & Dongen, Jos van. (2009). Pentaho® Solutions: Business Intelligence and DataWarehousing with Pentaho and MySQL®. In Inc. Wiley Publishing (Series Ed.) Robert Elliott & Sara Shlaer (Eds.), (pp. 651).
5. Chen, Chin-Sheng, Filipe, Joaquim, Seruca, Isabel, & Cordeiro, José (Eds.). (2006). *Enterprise Information Systems VII*: Springer.
6. Dario, Bernabeu R. (2009). HEFESTO: metodología para la construcción de un almacén de datos (pp. 146). Córdova.
7. Fabbri, Andrea G., Gaál, Gabor, McCammon, Richard B., & North Atlantic Treaty Organization. Scientific Affairs Division. (2002). *Deposit and geoenvironmental models for resource exploitation and environmental security*. Dordrecht ; Boston: Kluwer Academic Publishers.
8. Grabot, Bernard, Mayère, Anne, & Bazet, Isabelle. (2008). *ERP Systems and Organisational Change: A Socio-technical Insight*: Springer.
9. Gunasekaran, Angappa. (2008). *Techniques and Tools for the Design and Implementation of Enterprise Information Systems*. United States of America: IGI Global.
10. Hilton, Brian N. (2007). *Emerging spatial information systems and applications*. Hershey, PA: Idea Group Pub.
11. jcurtod. (2012). 34 subsistemas ETL de Kimball Retrieved 05.23.2012, 2012, from http://bi.social.uoc.edu/smc/blog/34-subsistemas-ETL-de-kimball
12. Langer, Arthur M. (2007). *Analysis and Design of Information Systems* (3 ed.): Springer.
13. Manolopoulos, Yannis, Filipe, Joaquim, Constantopoulos, Panos, & Cordeiro, José (Eds.). (2006). *Enterprise Information Systems* (1 ed.): Springer.
14. Minoli, Daniel. (2008). Official enterprise architectural standard. In Taylor & Francis Group (Ed.), *Enterprise Architecture: A to Z* (pp. 119-129). New York: Auerbach.
15. Rainardi, Vicent. (2008). *Building a Data Warehouse in SQL with examples* (1st ed.): Springer-Verlag New York.
16. Raisinghani, Mahesh S. (2004). *Business intelligence in the digital economy : opportunities, limitations and risks*. Hershey, PA: Idea Group Pub.
17. Samtani, Sunil, Mohania, Mukesh, Kumar, Vijay , & Kambayashi, Yahiko. (1999). Recent Advances and Research Problems in DataWarehousing Retrieved 24.04.2004, from www.springerlink.com/index/48acmd68h83nkcwl.pdf
18. Seipel, Dietmar, & Turull-Torres, José María (Eds.). (2004). *Foundations of Information and Knowledge Systems*: Springer.
19. Seruca, Isabel, Cordeiro, José, Hammoudi, Slimane, & Filipe, Joaquim (Eds.). (2006). *Enterprise Information Systems VI* (1 ed.): Springer.
20. Torra, Vicenç, & Narukawa, Yasuo. (2007). *Modeling Decisions: Information Fusion and Aggregation Operators (Cognitive Technologies)* (1 ed.): Springer.
21. Utley, Craig. (2008). Business Intelligence with Microsoft Office PerformancePoint Server 2007 Alyson Powell Erwin (Ed.) Retrieved from D:\Donwloads\Business Inteligence\Adrian\Business.Intelligence.with.Microsoft.Office.PerformancePoint.Server.2007.pdf doi:10.1036/0071493700
22. Ventana Research, Inc. (2008). Business Intelligence research agenda for 2008. In Inc Ventana Research (Ed.), *Research Agenda* (pp. 11).
23. Wang, John (Ed.). (2008). *Encyclopedia of Data Warehousing and Mining* (2 ed.): Information Science Reference.
24. Wrembel, Robert, & Koncilia, Christian. (2006). *Data Warehouses and Olap: Concepts, Architectures and Solutions*: IGI Global.

25. Yang, Jianhua, & SpringerLink (Online service). (2009). Information systems modeling, development, and integration ; proceedings, Third International United Information Systems Conference, UNISCON 2009, Sydney, Australia, April 21 - 24, 2009, from http://dx.doi.org/10.1007/978-3-642-01112-2 MIT Access Only