Universidad Central "Marta Abreu" de Las Villas

Facultad de Matemática Física y Computación

Ingeniería Informática

TRABAJO DE DIPLOMA

Estudio del Pentaho Data Integration en los procesos de integración de datos (ETL)

Autores: *Danay López Burgos*

Yaumara Galindo Artiles

Tutor: *MSc. Lisandra Díaz De la Paz*

Santa Clara

2013

"Año 55 de la Revolución"

DICTAMEN

Hago constar que el presente trabajo fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de los estudios de la especialidad de Ingeniería Informática, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor
Yaumara Galindo Artiles

Firma del autor
Danay López Burgos

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Laboratorio

El camino del éxito es angosto, difícil pero gratificante.

Dedicatoria

A tí:

A la personita que más quiero en esta vida, mi motivo de inspiración, ese pequeño gigante que con un beso hace de mí la persona más feliz del universo y también la más fuerte, esa persona tan especial es mi hijo Brayan.

A tí te la dedico nené.

***A mi madre Julia**:*

Por ser la mejor madre del mundo, mi mejor amiga y confidente, por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor, por su sacrificio y por cuidarme a mi pequeño, sin tu ayuda no lo habría logrado.

Gracias mami por ser tú.

A mi padre Silvio:

Que te puedo decir mi viejito lindo, por los ejemplos de perseverancia y constancia que te caracterizan y que me haz infundado siempre, por el valor mostrado para salir adelante, por tu amor incondicional, por cuidar de mi tesoro, por confiar en mí, gracias.

Te quiero mucho papi.

***A mi hermano Yoel**:*

Por ser el mejor ejemplo de hermano mayor del mundo, por guiar siempre a tu hermanita, por tus consejos y tu ayuda en todo momento, por tu cariño y confianza, por tus críticas y tu paciencia, gracias mi hermano por todo.

Por tu ayuda para que este sueño se convirtiera en realidad, gracias.

A mi querido esposo Branlys:

Por su paciencia y dedicación, por el amor que me ha dado siempre, por su apoyo, su confianza, por sus palabras alentadoras, por estar presente en los momentos que más lo he necesitado, por ser un buen amigo por encima de todo y por luchar junto a mí.

Por todos estos motivos, gracias amor.

Danay

A mi mamá y a mi papá

Porque son la luz de mi camino,

Mi apoyo y mis guías.

Su niña Yumi.

Agradecimientos

A Dios que todo lo puede.

A mi tutora MSc. Lisandra Díaz De la Paz, por su ayuda, guía y comprensión.

A mis padres por dedicarme su vida entera, por su amor incondicional, por sus horas de desvelo y por todas las preocupaciones que he puesto en ellos.

A mi hermano por amarme a su manera, tan única y especial. A mi abuelito que me espera impaciente con los brazos abiertos.

A toda mi familia por su preocupación y amor, en especial a mi tío José porque sin su ayuda no estaría aquí.

A Reni por todo el camino que hemos transitado juntos, por el tiempo y el amor que nos hemos dedicado.

A Cusi por todo el amor que me ha dado, sus consejos y su preocupación.

A mi abuela que no pudo ver su sueño hecho realidad, pero ha sido mi inspiración toda la vida.

A mis amigos de antes y de ahora que todos han contribuido a que sea hoy la persona que soy, especialmente Danay y Ketty (mis hermanitas) de quien he aprendido mucho, también a esas que estuvieron conmigo madrugadas de insomnio y días festivos Betsy, Gretter, Yalina, Lili y quien se fue incrementando con el paso del tiempo Yeni.

A Ernesto por brindarnos su tiempo, su hogar, su amistad y ayuda.

A mis amigos que me han dado apoyo en los últimos tiempos, que me escuchan y me quieren Henry, Enier, Jorge, Humberto, Carlo, Migue y Leo.

A mis más recientes pero no por eso menos importantes amigos los ciber Raúl, Make, David y todos los demás.

A todos los profesores que contribuyeron a mi formación profesional y como persona.

A mis compañeros que a través del transcurso de estos cinco años llegamos a ser una gran familia.

A todos y cada uno de ustedes gracias con todo mi corazón

Yaumara

Agradecimientos

Agradezco a toda mi familia que de una forma u otra me apoyaron y me ayudaron para que terminara mi tesis.

A mi turora MSc. Lisandra Díaz De la Paz por su gran apoyo, paciencia y motivación para la culminación de nuestros estudios profesionales y para la elaboración de esta tesis.

Yau, tú eres una persona muy especial, gracias por la amistad, el cariño y la confianza que me haz dado, de tí aprendí muchas cosas lindas, creo que la universidad sin Galindo pierde su encanto, gracias por luchar junto a mí en la tesis y por ser la mejor amiga.

A mis chicas preferidas Yalina , Betsy, Gretter, Liliaesther, aunque en este año nos hemos distanciado , quiero que sepan que las quiero mucho.

A Yenissey, por dedicar una minutos a conversar con Yaumara y conmigo en el cuarto.

Carlos, Ernesto, Henry gracias a ustedes por ser mis amigos, en todo momento, por tenerme paciencia, por sus detalles con Yaumara y conmigo, por sus aportes en la elaboración de esta tesis que me fue de mucha ayuda, la universidad sin ustedes no hubiera sido tan especial, los quiero mucho.

Enier, Jorge, Leandro, Humberto, Carlos Rodríguez, gracias por sus consejos , por brindarme su apoyo y por ser como son.

Mis compañeros de aula, a todos, sin excepciones, por darme los cinco años más lindos de mi vida, espero nos volvamos a reunir y disfrutar de ese grupo de Informática, un abrazo para ustedes de la más grande del aula.

Raúl, Leandro(Make), Sergio, David, Oscar, Daril, Eduardo(Piti) chicos gracias por soportarme todos los días en el laboratorio, por haberme hecho reír cuando más lo necesité, de verdad gracias por la amistad de todos ustedes , al final los cibernéticos con los informáticos si podemos ser amigos .

A los profesores, aquellos que marcaron cada etapa de nuestro camino universitario, y que me ayudaron en asesorías y dudas presentadas en la elaboración de la tesis.

Y a todos aquellos que participaron directa o indirectamente en la elaboración de esta tesis.

¡Gracias a ustedes! ***Danay***

RESUMEN

En la actualidad los volúmenes de datos crecen aceleradamente en todos los procesos de una organización. Con frecuencia los directivos necesitan analizar la información de diferentes departamentos de manera centralizada, lo que conlleva a la necesidad de consultar datos desde fuentes heterogéneas. Teniendo presente la diversidad de formatos, tipos y fuentes de datos en los cuales se puede encontrar dicha información, se hace imprescindible contar con procesos de extracción, transformación, limpieza, carga (ETL) y métodos de integración de datos que permitan una vista unificada con la mayor calidad posible. Para lograr lo anteriormente expuesto se utiliza la herramienta Spoon perteneciente al Pentaho Data Integration (PDI) en la implementación de transformaciones y trabajos desde su interfaz gráfica.

Como resultado se logra describir y valorar críticamente la herramienta Spoon apoyados en el análisis de los patrones que agrupan los subsistemas de Kimball y de las anomalías detectadas en un caso de estudio.

ABSTRACT

Currently data volumes are growing rapidly in all process of an organization. Often managers need to analyze information from different departments in a centralized manner, which leads to the need to query data from heterogeneous sources. Bearing in mind the diversity of formats, types and data sources on which you can find this information, it is essential to have processes for extracting, transforming, cleaning, loading (ETL) and data integration methods that enable a unified view with the highest possible quality. To achieve the above the tool Spoon belonging to Pentaho Data Integration (PDI) is used in the implementation of transformations and jobs since its graphical interface.

As a result achieves to describe and critically evaluate Spoon supported tool in the analysis of the patterns grouped Kimball subsystems and the anomalies detected in a case study.

Tabla de contenido

INTRODUCCIÓN

Las bases de datos constituyen una herramienta fundamental de control y manejo de las operaciones comerciales, mantienen la información necesaria para la actividad diaria de la organización y suministran datos a los sistemas de información corporativos (Sanz, 2010).

En la actualidad existe en las empresas un número considerable de información almacenado en diferentes fuentes de datos. Dicha información se puede encontrar de disímiles maneras debido 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, entre otros, ya sea por carecer de ellos o por no creer que sean necesarios. Cada uno por su parte ha dejado en cada aplicación, módulo o tabla, su estilo personalizado, influyendo de esta manera en la creación de modelos inconsistentes e incompatibles entre sí, de toda esta situación surge un importante problema para los directivos: obtener información centralizada de múltiples fuentes heterogéneas entre sí, alcanzando la calidad requerida en los datos.

Esta necesidad de obtener información es la principal razón de negocios que conduce al concepto de Almacén de Datos (acrónimo del inglés Data Warehouse (DW)), el DW convierte los datos operacionales de una organización en una herramienta competitiva, por hacerlos disponibles a los empleados que lo necesiten para el análisis y toma de decisiones.

Contar con una buena distribución de la información y con datos adecuadamente organizados es vital para la toma de decisiones. Los sistemas de integración de datos son los que se encargan de solucionar este tipo de problemas (Hungria, 2009). 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, para poder explotar la información existente en los diferentes sistemas operacionales, se debe extraer, transformar y cargar hacia el DW. Para realizar estos procesos, se necesita una herramienta ETL, estas siglas significan Extract, Transform & Load (extraer, transformar y cargar).

El proceso de construcción de un sistema ETL puede ser extraordinariamente exigente y complejo, estando además limitado por muchos aspectos tales como: los requerimientos, los formatos y deficiencias de los datos de origen, las habilidades del personal disponible, las necesidades de los usuarios finales, el presupuesto del proyecto, las ventanas de tiempo para los procesos de actualización y demás. Teniendo en cuenta esto, no se debe despreciar el tiempo y los recursos que se han de utilizar para su construcción (Matt Casters, 2010).

Las herramientas ETL surgen para facilitar la implementación de los procesos de integración de datos. La herramienta que se aborda en el presente trabajo es una de las más utilizadas en la actualidad, Pentaho Data Integration (PDI), la cual maneja dos tipos diferentes de objetos, las transformaciones y los trabajos, dentro de sus características se encuentra que es de código abierto con licencia LGPL, de fácil lectura tanto para los XML que recogen transformaciones y trabajos como para el repositorio relacional de objetos ETL, API fácil de instalar, aplicable a todos los tipos de bases de datos, GUI fácil de usar, habilidad para la importación y exportación de datos de un formato a otro cualquiera, permite la extensión de código mediante plugins, entre otras. Esas características hacen posible que personas no muy experimentadas en Inteligencia de Negocio (BI, por sus siglas en inglés) puedan realizar contribuciones valiosas a los proyectos. Esta herramienta es de alta usabilidad, razón por la cual surge el interés de analizar cuán ventajoso resulta al personal que la utiliza y explota emplearla en el proceso de integración de datos.

En el presente trabajo se realiza un estudio de la herramienta Spoon del PDI en la integración de datos identificando las fortalezas y posibles debilidades que presenta dicha herramienta. Además se analizan los patrones de ETL basados en los subsistemas de Kimball, los cuales son tratados mediante la implementación de un caso de estudio.

Objetivo general

Caracterizar la herramienta Spoon del PDI en el proceso de integración de datos a partir del análisis de los subsistemas de Kimball y la exposición de un caso de estudio.

Objetivos específicos

1. Argumentar los procesos ETL de un Almacén de Datos.
2. Analizar algoritmos y métodos que se aplican en la integración de datos.
3. Realizar un estudio de las herramientas que existen para implementar los procesos de ETL.
4. Valorar críticamente la herramienta Spoon.
5. Aplicar los patrones de ETL que agrupan los subsistemas de Kimball a un caso de estudio mediante la herramienta Spoon.

Preguntas de investigación

1. ¿En qué consisten los procesos de ETL y qué herramientas se utilizan para su ejecución?
2. ¿Qué métodos y algoritmos se utilizan en la integración de datos?
3. ¿Cuáles son los patrones de ETL y su vinculación con los subsistemas de Kimball?
4. ¿Cuáles son las ventajas y posibles desventajas que presenta el Spoon en la integración de datos?

Hipótesis de investigación

Aplicando los patrones que agrupan los subsistemas de Kimball a una solución de ETL disponible y empleando la herramienta de integración de datos Spoon en un caso de estudio se puede caracterizar dicha herramienta.

Justificación de la investigación

En la mayoría de las organizaciones el volumen de datos crece constantemente y se encuentra disperso en fuentes de datos heterogéneas. La carencia de una visión global de la información que permita realizar análisis y consultas para mejorar la toma de decisiones por parte de los gerentes da paso a la necesidad de integrar los datos. Por tanto, se considera trascendente estudiar algoritmos y métodos que se utilizan en la integración de datos y las herramientas que se emplean para ejecutar los procesos de ETL como es el caso del Spoon perteneciente al PDI. El estudio planteado ayuda entre otros aspectos, a conocer cuán factible resulta utilizar dicha herramienta en los procesos de integración de datos y ETL.

Viabilidad

Para implementar las transformaciones y trabajos se utiliza la herramienta gráfica Spoon del PDI en la solución de las anomalías detectadas en las fuentes de datos de un caso de estudio.

En el grupo de Base de Datos del Centro de Estudios Informáticos (CEI) se cuenta con los medios técnicos, software y con el personal necesario para desarrollar la presente investigación.

El presente trabajo se estructura en tres capítulos:

Capítulo 1: En este capítulo se muestran las etapas de los procesos ETL de integración de datos. Se exponen varios conceptos de integración de datos desde el punto de vista de varios autores. Además se analizan los principales métodos y algoritmos utilizados para la integración de datos y se presentan aspectos fundamentales sobre las herramientas de integración de datos más utilizadas divididas en software propietario y de código abierto, haciendo énfasis en las herramientas del PDI.

Capítulo 2: En este capítulo se analiza la arquitectura de la herramienta PDI haciendo hincapié en la caracterización de su interfaz gráfica Spoon y ejemplificando sus componentes (pasos, saltos, transformaciones, trabajos y repositorios). Además se analizan los subsistemas de Kimball y se organizan en patrones que se reconocen como actividades típicas en los procesos ETL.

Capítulo 3: En este capítulo se implementan transformaciones y trabajos en el caso de estudio "Recursos Humanos" a partir de las anomalías detectadas en el proceso de integración de datos y de los patrones expuestos en el capítulo anterior. Además se identifican las ventajas y posibles debilidades de la herramienta Spoon en base a la experiencia que se tiene del uso de la misma en su versión libre 4.2.1.

CAPÍTULO I: CONSIDERACIONES GENERALES SOBRE LOS PROCESOS DE INTEGRACIÓN DE DATOS Y ETL DE UN ALMACÉN DE DATOS

En el presente capítulo se abordan aspectos generales sobre la integración de datos dentro de los cuales se presentan conceptos desde la visión de diversos autores y se muestran métodos y algoritmos utilizados en los procesos de integración. Además se exponen las principales características de los procesos de ETL en Almacenes de Datos. También se analizan herramientas de integración de código abierto y propietario enfocando el análisis en los principales aspectos del objeto de estudio PDI como herramienta de integración de datos y de implementación de procesos de ETL.

1.1 Procesos de integración de datos en un Almacén de Datos

Existen tres variantes para realizar los procesos de integración de datos:

- ETL, cuando la transformación se realiza en la herramienta de integración.
- ELT, cuando la etapa de transformación se hace en el SGBD.
- ETLT, cuando el proceso se realiza en ambos primero en la herramienta de integración y después de la carga en el SGBD.

ETL (Extract, Transform & Load): Extrae los datos desde diversas fuente, los transforma y limpia antes de cargarlos en una base de datos, data mart o DW (Sybven, 2013).

ELT (Extract, Load & Transform): Extrae los datos, los carga en una base de datos, a continuación, realiza una serie de transformaciones a los datos dentro de la base de datos (Sybven, 2013), esta variante es impulsado principalmente por Oracle RDBMS. Esta variante dispone de software que realizan transformaciones que migra datos en un motor de base de datos, a menudo generando sentencias SQL y procedimientos entre diversas tablas de datos.

ETLT (Extract, Transform, Load & Transform): es una combinación de ETL y ELT en el cual la transformación se realiza mediante un servidor ETL, en función de los tipos de transformaciones (Sybven, 2013).

Los procesos ETL cubren el transcurso de cómo los datos se cargan desde el sistema de origen hasta las fuentes de salidas, actualmente comprende una etapa de limpieza como una etapa separada:

- Extracción: obtención de información de las distintas fuentes tanto internas como externas.
- Transformación: filtrado, limpieza, depurado, homogenización y agrupación de la información.
- Carga: organización y actualización de los datos y los metadatos en la base de datos.
- Explotación: extracción y análisis de la información en los distintos niveles de agrupación.

1.1.1 Extracción

Extracción de datos es la selección sistemática de datos operacionales usados para poblar el componente de almacenamiento físico DW (Nader, 2003). El paso comprende la extracción de datos desde el sistema de origen y la hace accesible para su posterior procesamiento. El objetivo principal del paso de extracción es recuperar todos los datos necesarios desde el sistema de origen como sea posible, debe ser diseñado de manera que no afecte negativamente el sistema de origen en términos de rendimiento.

Existen diversas variantes para realizar el proceso (Javlin, 2011):

- Notificación de actualizaciones: Si el sistema de origen es capaz de proporcionar una notificación de que un registro ha cambiado y describir el cambio, esta es la forma más sencilla de obtener los datos.
- Extracto incremental: algunos sistemas pueden no ser capaces de proporcionar una notificación de que una actualización se ha producido, pero son capaces de

identificar los registros que han sido modificados y proporcionar un extracto de dichos registros.

- Extracto completo: algunos sistemas no son capaces de identificar qué datos se han cambiado en absoluto, por lo que un extracto completo es la única forma de poder obtener los datos del sistema. El extracto completo requiere mantener una copia del último extracto en el mismo formato con el fin de ser capaz de identificar los cambios. El extracto completo maneja supresiones también.

Al utilizar extractos incrementales o completos, la frecuencia de retorno es extremadamente importante. Particularmente para los extractos completos, los volúmenes de datos pueden ser de decenas de gigabytes.

1.1.2 Transformación

Transformación de datos es el proceso para realizar otros cambios en los datos operacionales para reunir los objetivos de orientación a temas e integración principalmente (Nader, 2003). En la etapa de transformación se aplican un conjunto de reglas para transformar los datos desde el origen al destino. Esto incluye convertir los datos a una manera compatible y estándar usando las mismas unidades para que más tarde puedan ser acoplados. La etapa de transformación también requiere la combinación de datos de varias fuentes, generar agregados, la generación de claves sustitutas, clasificación, derivar nuevos valores calculados y la aplicación de reglas avanzadas de validación. Además en esta etapa se debe tener en cuenta la limpieza, calidad y conformación de los datos para lograr transformaciones eficaces.

1.1.2.1 Limpieza

La limpieza es la corrección en los datos de posibles errores, por ejemplo: datos incompletos, duplicados, formatos inconsistentes en cuanto a descripción, abreviaturas y unidades de medidas, falta de datos de entrada o datos que violen las restricciones de integridad del sistema. La etapa de limpieza es una de las más importantes, ya que garantiza la calidad de los datos en el almacén de datos.

En esta etapa se deben corregir las anomalías que se detecten en el proceso de la unificación de datos, a continuación se exponen algunas de las anomalías más frecuentes:

- No estandarización de valores
    - Esta anomalía consiste en la existencia de uno o varios campos que poseen datos escritos con formato diferente pero que significan lo mismo, o sea datos que no siguen un estándar o norma predeterminada.
    - Por ejemplo el caso típico de: (categorías de sexo Masculino / Femenino / Desconocido, M / F / null, hombre / mujer / no disponible, se convierten a la norma Masculino / Femenino / Desconocido).
- Existencia de valores nulos.
    - Como su nombre lo indica en las fuentes se encuentran valores nulos o vacíos, lo cual atenta contra un adecuado análisis de la información. La solución que se brinda es remplazarlos por un valor normalizado o sea una constante.
- Esquemas no integrados.
    - Esta anomalía se pone de manifiesto cuando se trata de integrar varias fuentes de datos y en una de ellas aparece el nombre de un campo escrito de una forma y en la otra fuente aparece escrito con otro nombre, pero los valores que tienen ambas fuentes son equivalentes. Por ejemplo: una columna denominada raza y otra llamada color de piel y en ambos casos los valores son: Blanca, negra y mulata. La solución que se brinda es integrarlas a ambas en una sola columna bajo un mismo nombre.
- Existencia de duplicados.
    - Como su nombre lo indica es cuando se está en presencia de dos o más filas donde coinciden todos o casi todos los valores de sus campos. La solución que se ofrece es eliminar los valores duplicados o repetidos.

1.2.1.2 Calidad de los datos

La Calidad de los Datos es un término que abarca tanto el estado de los datos, así como el conjunto de procesos para lograr dicho estado. El objetivo es disponer de datos libre de errores, duplicados, omisiones, variaciones y datos innecesarios. Los datos deben ser correctos, inequívocos, coherentes y completos (Paz, 2012):

- Datos correctos: Los valores y las descripciones de los datos deben describir su verdadera definición.
- Datos inequívocos: Los valores y las descripciones de los datos sólo pueden tener un único significado.
- Datos coherentes: Los valores y las descripciones de datos deben usar una notación constante para transmitir su verdadero significado. *Ejemplo:* para mantener la coherencia de los datos se debe utilizar solo una nomenclatura.
- Datos completos: Se debe garantizar que los valores individuales y las descripciones de los datos, se definan para cada caso, permitiendo identificar que valores posibles puede tomar cada dato y se debe asegurar que el número total de registros completados después que se realice el proceso de integración debe ser del 100% completo asegurando que no se pierde información en alguna parte del flujo de datos.

1.2.1.3 Conformación

Cuando la información se encuentra limpia y con una calidad adecuada, esta es unificada, conformada y normalizada. Los indicadores son calculados de una forma racional, lo mismo que los atributos de las dimensiones, para que estén unificados y en todos los sitios donde aparezcan tengan la misma estructura y el mismo significado (Javlin, 2011).

1.1.3 Carga

El proceso de carga es la inserción sistemática de datos en el componente de almacenamiento físico DW (Nader, 2003), o en cualquier sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad

de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos, los Almacenes de Datos por su parte mantienen un historial de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo. La integridad referencial es necesaria para asegurar la consistencia mantenida por la herramienta de ETL.

Algunas de las Técnicas para resolver estos problemas que se presentan con gran frecuencia son:

- Utilitario de carga en lote: ordenar artículos de entrada sobre llave de clúster y usar E/S secuencial; construir índices y tablas derivadas
- Usar paralelismo y técnicas incrementales

1.2 Integración de datos

La integración de datos proporciona un mecanismo para unir datos de diferentes fuentes en un esquema único. La integración se lleva a cabo en dos etapas (Préstamo, 2004):

- Homogenización: transformación de la información del formato original de las fuentes naturales al formato y modelo de datos del DW.
- Integración: la información recuperada es agregada y organizada al esquema del DW.

Principales conceptos

A continuación se muestran varias definiciones de integración de datos expresadas desde el punto vista de diferentes autores:

- Según Behrendt (1993) define el proceso de integración como la especificación de reglas de transformación entre cada esquema remoto y el esquema global.
- La integración de datos (Lenzerini, 2002) es el problema de combinar los datos que residen en diferentes fuentes y que proporciona al usuario una vista unificada de estos datos. El problema de diseñar sistemas de integración de datos es importante en el mundo real actual y se caracteriza por una serie de cuestiones que son interesantes desde un punto de vista teórico.

- La integración de datos es un proceso por el cual varias bases de datos asociadas, con esquemas locales se integran para formar una sola base de datos, con un esquema global asociado (Peter Mc.Brien 2003).
- La integración de datos es un desafío generalizado enfrentado en las aplicaciones que necesitan para realizar consultas a través de múltiples fuentes de datos heterogéneas y autónomas (Alon Halevy, 2006).
- Integración de datos y sistemas de intercambio ofrecen una interfaz uniforme a una multitud de fuentes de datos y la capacidad de compartir datos a través de múltiples sistemas (Xin Dong, 2007).
- El enfoque de los sistemas de integración de datos es producir un esquema global integral con éxito, la integración de datos procede de fuentes de datos heterogéneas en formato y en estructura (Po, 2008).
- Según Hungria (2009), se conoce como integración de datos a la necesidad de obtener información a partir de la extracción de datos distribuidos en múltiples fuentes heterogéneas entre sí, que hay que combinar para facilitar al usuario una vista unificada de esos datos.
- La integración de datos de expresión como lo define Matt Casters en el (2010) se refiere al proceso de combinación de datos desde fuentes diferentes para proporcionar una única vista comprensible sobre todo de los datos combinados.

Teniendo presente las definiciones y comentarios de estos autores se puede concluir que la integración de datos es el proceso de combinar los datos de diferentes fuentes heterogéneas entre sí con el fin de ofrecer al usuario una vista unificada de los datos limpia, libre de anomalías y con la calidad requerida.

1.2.1 Algoritmos de integración de datos

Según explica en su libro Kieran Hogg (2009) en su libro 'El análisis de la integración de datos' Algoritmos existen varios algoritmos para la integración de datos, dentro de los que se destacan los siguientes: Clio, BDK, Merge, Dumas y Hummer. Para tener en cuenta el

funcionamiento de estos algoritmos existen 4 criterios:

1. Entrada: tipo de datos que debe tomar el algoritmo.

2. Etapas: etapas de integración que se implementan

   - Pre-integración
   - Comparación del esquema
   - Conformación del esquema
   - La fusión y restructuración
     - Complementar y corregir
     - Minimalidad
     - Comprensibilidad

3. Información auxiliar: herramientas externas que utiliza.

4. Salida: forma de salida del resultado de la integración.

1.2.1.1 Clío

El algoritmo Clío aborda el problema de la transformación de conceptos en el esquema, en tres pasos principales:

1. Extracción de concepto: este proceso extrae las entidades del mundo real en el esquema, junto con sus dependientes y relaciones.
2. Principio de correlación y fusión: el concepto de mapeo y la fusión de proceso se consigue mediante la comparación de los conceptos extraídos y la búsqueda de los que corresponden. La herramienta de mapeo de generación de Clío se utiliza para obtener una especificación de una mejor relación.
3. Refinamiento del esquema integrado: paso final de Clío, refina el esquema con la ayuda del usuario. La restricción de coincidencia es un concepto aceptado y permitido, se presenta al usuario con algunas versiones diferentes para elegir de un esquema final. A partir de los conceptos, Clío coincide con los comunes a ambas

fuentes de datos. La fusión inicial tiene lugar en este punto. La entrada del usuario se solicita para refinar los resultados de integración.

Clío acepta como su entrada esquemas XML o base de datos relacional y su salida es un esquema integrado.

1.2.1.2 BDK

El nombre BDK tiene origen en las iniciales de sus autores, es una "técnica general" para la integración de datos, tiene un enfoque formal: se define como una generalización de todos los esquemas, está ideado de tal manera que un operador binario fusione a la vez conmutativa y asociativamente. El modelo de transformación está realizado para preservar las relaciones, restricciones y así permitir la traducción de cualquier modelo de otra fuente y viceversa.

Existen restricciones de activar la habilidad para describir los modelos alternativos, como el esquema relacional (ER) y funcional. Los modelos BDK son esquemas que se pueden representar como un conjunto de grafos dirigidos donde los atributos son nodos y las relaciones son aristas. Hay dos tipos de relaciones que son permitidas, "atributo" y "especialización". En la etapa de fusión se destruyen objetos del mismo nombre de los dos modelos en el objeto con el mismo destino. Con el fin de hacer frente a las clases implícitas en el esquema, el modelo se debilita antes de la integración, se utilizan después para eliminar un conjunto de reglas y volver a derivar las clases del nuevo esquema.

BDK no ofrece restricciones a su entrada ya que es un modelo genérico para la integración, su salida es un modelo genérico de alto nivel también.

1.2.1.3 Merge

El algoritmo Merge (combinar) está basado en el BDK. Este algoritmo integra dos modelos en base a sus correspondencias comunes. No está atado a una aplicación particular, ya sea de base de datos, XML u ontología

Consta de 5 etapas:

1. Inicialización: el resultado de la fusión, G, se inicializa a un modelo vacío θ.
2. Elementos: una relación de equivalencia es creada por el cotejo de los elementos de A, B y MapAB. Inicialmente, cada elemento está contenido en su propio grupo, sin embargo, si existe una relación entre un elemento en ambos A y B y una asignación está contenida en MapAB, entonces estos se agrupan.
3. Propiedades del elemento: la propiedad de cada elemento se calcula por un conjunto de reglas definidas.
4. Relaciones: por cada par de elementos de G, ahora se crean relaciones entre los grupos en la que cada elemento está en un grupo diferente a la otra y no existe ninguna relación.
5. Resolución de conflictos: tras los pasos anteriores, G es ahora una unión de A, B y MapAB sin duplicados. Sin embargo, durante este proceso pueden producirse conflictos dentro del modelo. Para cada conflicto, verifica si hay una regla definida para la resolución de conflictos, si no se aplica una regla predeterminada.

Este algoritmo no hace restricciones a su entrada o salida, ya que recibe y ofrece un modelo genérico.

1.2.1.4 DUMAS

El algoritmo DUMAS acepta únicamente bases de datos relacionales como entrada. No implementa las etapas de pre-integración, conformación o restructuración. Tiene variables que se pueden modificar para variar el rendimiento del algoritmo.

- **K**- duplicados a utilizar para la creación de la matriz de similitud promedio.
- **TokenThreshold**-valor en el que dos símbolos se definen como duplicados.
- **K Threhold**-el valor en el que dos tuplas se definen como duplicados.

El algoritmo está compuesto de tres pasos que describen su funcionalidad y como se implementa.

El **paso 1** consiste en tomar cada fila de la primera tabla, tratando toda la fila como una cadena. A continuación se compara esta cadena con cada fila de la otra tabla, también como una cadena. Para la comparación de cadenas, se usa SoftTFIDF, una forma modificada de un TFIDF (Términos de frecuencia, frecuencia inversa de documento, esto es que si el término es frecuente en la coincidencia y no es frecuente en el documento, se obtiene una puntuación superior a uno, común en el documento) de la suite SecondString. El autor lo describe como un "TFIDF basado en la distancia métrica, ampliado para usar " soft " como señal de coincidencia. En concreto, las fichas se consideran una coincidencia parcial si consiguen un buen resultado utilizando un comparador de cadena interior".

Se recibe una puntuación de la instancia comparador y se compara con el K Threshold. A continuación se agrega como "marcador" de tamaño K. Una vez que este proceso se ha repetido, se obtiene una lista de los duplicados K superior.

En el **paso 2** las tuplas duplicadas K superiores, se comparan cada campo de la primera tabla con cada uno en la segunda. Esto muestra como resultado una matriz de similitud de campo, un conjunto de puntuaciones de SoftTFIDF para cada comparación por campo. Para estas dos tuplas implica que cualquier valor por encima de tokenThreshold corresponde a una coincidencia en las dos columnas. La variación de tokenThreshold afecta a los resultados en este punto, un umbral más bajo puede identificar incorrectamente dos columnas duplicadas y a la inversa, también puede pasar por alto coincidencias correctas. El problema está parcialmente mejorado en la siguiente etapa.

En el **paso 3**, al final de la segunda etapa, se obtuvieron matrices de similitud K. Para tener una visión general de las tablas, se promedian las matrices para crear una matriz de similitud acumulada. Para obtener el resultado final, los valores de esta matriz se comparan con tokenThreshold, los valores anteriores son ahora las columnas finales coincidentes.

DUMAS genera una lista de pares de columnas duplicadas.

1.2.1.5 Hummer

Este algoritmo de integración se basa en el algoritmo DUMAS expuesto anteriormente. Hummer, combina varios proyectos en su sistema. El algoritmo consta de tres pasos principales, lleva como título: "Fusión de datos en tres pasos"**.**

1. Esquema de combinación y transformación de datos: el primer paso en el proceso es la alineación de esquema. Para ello se utiliza el algoritmo DUMAS. El esquema se transforma mediante la designación de una fuente elegida y cambia el nombre de todos los nombres semánticamente similares para que coincida con la fuente elegida. La transformación se completa tomando la unión externa completa de las tablas.
2. Detección de duplicados: Una vez que una alineación del esquema propuesto es producido, se presenta al usuario para agregar o quitar manualmente emparejamientos erróneos. Hummer entonces detecta los duplicados, no sólo teniendo en cuenta sus nodos de texto, sino también los de sus hijos, al identificar duplicados.
3. Resolución de conflictos: el paso final consiste en resolver la representación de los objetos del mundo real. Esta declaración es consecuente con los pasos de integración anteriores, devuelve los resultados apropiados una vez que se toman en consideración. También es posible realizar operaciones tales como la especificación de un precedente de una correspondencia duplicada.

Hummer acepta esquemas XML o base de datos relacional como su entrada y ofrece como salida una sola tabla con una representación para cada objeto del mundo real.

1.2.2 Métodos de integración de datos

Uno de los aspectos más importantes en el diseño de un sistema de integración de datos según expone Lenzerini (2002) es la especificación de las correspondencias entre los datos de las fuentes y estos en el esquema global. Estas correspondencias determinan como las consultas realizadas al sistema son respondidas. Desde el campo de la integración de la

información existen diferentes enfoques para especificar las correspondencias entre los datos de un sistema. Estos acercamientos son Vista Local (LAV) y Vista Global (GAV).

1.2.2.1 Principales características de los métodos

Los componentes principales de un sistema de integración de información son el esquema global, las fuentes y las correspondencias (*mappings*). Formalmente este tipo de sistema *I* es definido como una tupla <*G*, *S*, *M*> dónde:

- *G* es el esquema global en un lenguaje LG sobre un alfabeto AG.
- *S* el esquema de la fuente en un lenguaje LS sobre un alfabeto AS.
- *M* el mapeo entre el *G* y *S*. Un conjunto de aserciones del tipo:
  - $qS \rightsquigarrow qG$
  - $qG \rightsquigarrow qS$
  - Donde qG y qS son consultas de igual aridad, sobre los lenguajes LM, G y LM, S respectivamente.

El esquema de la fuente describe la estructura de las fuentes, donde están los datos. Mientras el esquema global proporciona una vista virtual, integrada y conciliada de las fuentes subyacentes. Mediante las correspondencias se establece la conexión entre los elementos del esquema global y aquellos que pertenecen al esquema de la fuente.

1.2.2.2 Comparación de los diferentes enfoques

En la tabla 1 se muestra una comparación de los métodos LAV y GAV basado en las siguientes de características: modelado, calidad, extensibilidad y procesamiento de consultas.

Características	LAV	GAV
Modelado	En este enfoque las correspondencias de *M* asocia cada elemento *s* de la fuente *S* a una consulta *qG*. Luego, las aserciones de *M* son del tipo: s → qG.	El enfoque GAV se basa en la idea de que el mapeo *M* asocia a cada elemento *g* en *G* una consulta *qs* sobre la fuente. Luego, las aserciones de *M* son del tipo: g→ qS.
Calidad	Depende de la caracterización de las fuentes.	Depende de cómo se hayan compilado las fuentes en el esquema global.
Extensibilidad	Basta con agregar las aserciones.	Implica rehacer el esquema global.
Procesamiento de consultas	Necesita más razonamiento pues se debe replantear la consulta en términos de las fuentes pero la información que entregan las aserciones son exactamente al revés (fuentes en términos de vista global).	Es más sencillo pues la información de las aserciones es exactamente la necesitada, se le dice al sistema como utilizar las fuentes para poder obtener los datos.

Tabla 11.2.2.2 Comparación de los diferentes* enfoques *LAV y GAV

1.3 Herramientas que se utilizan para realizar los procesos de ETL

Existen una amplia variedad de herramientas que se pueden utilizar en los procesos de ETL, las empresas productoras de software compiten entre sí para darle mayor popularidad a sus productos. Estos software se pueden clasificar en dos géneros, software libre y propietario, la primera categoría le brinda la posibilidad al usuario de acceder al código fuente de estos productos para concebir modificaciones ofreciéndole popularidad a las herramientas que se encuentran dentro de esta clase además el valor de estos productos son más bajos haciendo que mayor cantidad de usuarios accedan a ellos. Por otra parte el software propietario es más costoso y no permite acceder al código fuente, pero es más utilizado por empresas ya que ofrece una mayor seguridad del producto a sus usuarios.

1.3.1 Software propietario

El software propietario ofrece entre sus ventajas la propiedad y decisión de uso del software por parte de la empresa, soporte para todo tipo de hardware, mejor acabado de la

mayoría de aplicaciones, menor necesidad de técnicos especializados pero a su vez el alto costo de sus licencias hace que muchas empresas no utilicen sus productos. Entre las herramientas comerciales más conocidas para la implementación de los procesos ETL se encuentran:

1.3.1.1 SSIS SQLServer Integration Services

El gestor de base de datos SQL Server posee entre sus principales características facilidad de instalación, distribución y utilización. Entre sus herramientas posee un administrador corporativo y un analizador de consultas. Puede utilizarse el mismo motor de base de datos a través de plataformas que van desde equipos portátiles que ejecutan Microsoft Windows 95 ò 98 hasta grandes servidores con varios procesadores que ejecutan Microsoft Windows NT®, Enterprise Edition. Entre sus funcionalidades se encuentra el almacenamiento de datos, incluye herramientas para extraer y analizar datos resumidos para el procesamiento analítico en línea (acrónimo del inglés Online Analitycal Processing (OLAP)). También se utiliza para combinar datos de almacenes de datos heterogéneos, llenar almacenamientos de datos y puestos de datos, limpiar y normalizar datos, generar BI en un proceso de transformación de datos y automatizar las funciones administrativas y la carga de datos (Grecol, 2012).

1.3.1.2 Ab Initio

Herramienta ETL que se alimenta de diversas fuentes, procesando la información, aplicando reglas de negocio y alimentando otros sistemas o DW. La herramienta tiene un Front-End de desarrollo gráfico, implementado con módulos inter conectables que a su vez pueden ser programados con lógica específica (Corporation, 2010).

Estos procesos se generan en background shell scripts de UNIX por lo que se requiere tener cierto conocimiento de línea de comandos UNIX, Linux, o similar.

1.3.1.3 IBM InfoSphereDataStage

Esta herramienta brinda soporte a la recopilación, integración y transformaciones de grandes volúmenes de datos, con estructuras de datos sencillas y complejas. Gestiona los nuevos datos en cuestión de segundos, así como grandes cantidades de datos que bloquean el sistema, en intervalos diarios, semanales o mensuales. Ofrece tres funciones fundamentales necesarias para conseguir una correcta integración de datos empresariales: la conectividad más global para acceder con rapidez y facilidad a cualquier sistema de origen o destino; herramientas avanzadas de desarrollo y mantenimiento, que agilizan la implementación y simplifican la administración y una plataforma escalable que permite gestionar con facilidad los actuales volúmenes masivos de datos empresariales (Wikipedia, 2011).

1.3.1.4 Informatica

Es una suite de integración de datos comercial, fundada en el año 1993, es el líder en participación de mercado en la integración de datos. Tiene 2600 clientes entre ellos se encuentran las compañías de Fortune, las empresas en el Dow Jones y organizaciones gubernamentales. La empresa tiene como único objetivo la integración de datos. Tiene un paquete grande que le ofrece a las empresas para integrar sus sistemas, limpiar sus datos y posibilita conectarse a un gran número de sistemas actuales y anteriores. Es muy costoso, requiere algún tipo de entrenamiento de su personal para usarlo y probablemente requiera la contratación de consultores también. Es muy rápido y se puede escalar para sistemas grandes. Tiene "Optimización Push down", que utiliza un enfoque ELT, por tanto la etapa de transformación se hace en la base de datos fuente (Corporation, 2012).

1.3.1.5 Oracle Data Integrator

Oracle Data Integrator es la herramienta de integración de datos de Oracle (Gil, 2011). Es la apuesta de Oracle en cuestiones de integración de datos y sustituye a OWB (Oracle Warehouse Builder). Forma parte de la solución OFM (Oracle Fusion Middleware) y está totalmente integrada con otras soluciones Oracle relacionadas con la gestión de datos. Ejecuta procesos con altos volúmenes de datos, obteniendo excelentes tiempos de

respuesta. Actualiza los DW, data marts, cubos OLAP y sistemas analíticos en general. Gestiona de forma transparente las cargas totales o incrementales, considera dimensiones SCD (Slowly Changes Dimensions), asegura la integridad y consistencia de datos y facilita la trazabilidad del dato (origen del dato, detalle de transformaciones y destino del dato). Procesos de integración de datos basados en datos de entrada, procesos .batch eventos y ejecución de servicios.

Conectividad: ficheros planos, ficheros XML, directorios LDAP, conexiones vía ODBC, JDBC e integración con arquitecturas SOA.

Alta disponibilidad y escalabilidad: Gestión y administración centralizadas (consola ODI). ODI se integra con la plataforma Oracle Fusion Middleware, ofreciendo sus componentes como aplicaciones Java EE, optimizados para aprovechar al máximo las capacidades de su servidor de aplicaciones Oracle WebLogic. Los componentes ODI están provistos de funcionalidades que permiten su despliegue en un entorno de alta disponibilidad, escalabilidad y seguridad. Alta productividad en el diseño de procesos de integración de datos.

1.3.1.6 SAP Data Integrator

Herramienta ETL que permite acceder a los datos desde múltiples fuentes, integrarlos en *.batch* y tiempo real y distribuirlos corporativamente, facilitando una visión analítica integral de la evolución del negocio. Como resultado, se obtiene una mejora en la eficacia operacional y en la cadena de suministro, la optimización de la relación con el cliente y la creación de nuevas oportunidades de negocio (Community, 2009).

1.3.2 Software de código abierto

El software de código abierto se define por la licencia (GPL) que lo acompaña, que garantiza a cualquier persona el derecho de usar, modificar y redistribuir el código libremente, existen varias herramientas de software libre que realizan los procesos ETL como las que se muestran a continuación.

1.3.2.1 Apatar

Apatar es un proyecto de código abierto que fue fundado en el año 2005. La primera versión de la herramienta fue liberada bajo la licencia GPLv2 en febrero de 2007. Esta herramienta de integración de datos proporciona conectividad a una variedad de bases de datos, aplicaciones, protocolos, archivos y mucho más. Además permite a los desarrolladores, administradores de bases de datos y usuarios de negocios integrar la información entre una variedad de fuentes y formatos de datos y proporciona una interfaz de usuario intuitiva que no requiere codificación para configurar un trabajo de integración de datos. Apatar ofrece una serie de capacidades sin igual en un paquete de código abierto: la conectividad con Oracle, MS SQL, MySQL, Sybase, DB2, MS Access, PostgreSQL, XML, InstantDB, Paradox, BorlandJDataStore, CSV, MS Excel, QED, HSQL, ERP Compiere, SalesForce.Com, SugarCRM, Goldmine, de fuentes de datos JDBC. Ofrece una sola interfaz para gestionar todos los proyectos de integración, flexibles opciones de implementación, integración bidireccional, fácil personalización, código fuente de Java incluido. Independiente de la plataforma, se ejecuta en Windows, Linux, Mac, 100% basado en Java, sin codificación y con diseñador visual (Community, 2010).

1.3.2.2 CloverETL

Herramienta basada en Java. Marco de trabajo para la integración de datos y la creación de transformaciones de datos. El componente base sigue el concepto de gráficos de transformación que consisten en nodos individuales. Cualquier transformación puede ser definida como un conjunto de nodos interconectados a través del cual los datos fluyen. Se puede utilizar como una aplicación independiente o estar integrada en un proyecto mayor. CloverETL trabaja con todos los datos estructurados, permite la combinación, transformación y circulación de los datos de cualquier origen.

Las aplicaciones para CloverETL:

- Migración de datos
- Recopilación de datos
- ETL para Almacenes de Datos
- Integración de datos

- Licencias y políticas

CloverETL Engine es una herramienta de código abierto distribuido bajo licencia dual (comerciales y LGPL), que permite una total transparencia y control sobre la herramienta, la fuente de código completo para el motor que está disponible para todos los clientes y usuarios (Gutiérrez, 2010).

1.3.2.3 Talend Open Studio

Es una herramienta de código abierto de integración de datos (no es una suite de Inteligencia de Negocios (BI) completa). Utiliza un enfoque de generación de código, interfaz gráfica de usuario dentro de Eclipse RC. Comenzó alrededor de octubre 2006, cuenta con una comunidad más pequeña que la de Pentaho pero tiene 2 empresas financieras que lo apoyan.

Talend tiene una gran cantidad de componentes, su enfoque es tener un componente distinto según la acción a realizar y para el acceso a base de datos u otros sistemas hay componentes diferentes según el motor de base de datos que se utilice. Trabaja con el concepto de workspace, a nivel de sistema de ficheros. En ese lugar se almacenan todos los componentes de un proyecto (todos los trabajos, su definición de metadatos, código personalizado y contextos). El repositorio se actualiza con las dependencias de objetos al ser variados. Si se modifica el repositorio de una tabla, por ejemplo, se actualiza en todos los trabajos donde se utiliza (Levin, 2008).

1.3.2.4 Pentaho Data Integration

Pentaho es una suite de herramientas de Inteligencia de Negocios de código abierto. Todas las herramientas que pertenecen a Pentaho están programadas en código 100% Java. Dentro de las que destaca un producto para la integración de datos llamado PDI. El cual utiliza un enfoque innovador y tiene una fuerte interfaz gráfica muy fácil de usar para el usuario. La compañía comenzó en torno a 2001 y no fue hasta 2002 cuando Kettle se integró en ella. Debido a la usabilidad e importancia de esta herramienta este trabajo se centra en su investigación.

1.4 Características de Pentaho Data Integration

Pentaho es una suite de herramientas de Inteligencia de Negocios que tiene dos versiones, la versión comercial y la versión de código abierto. Es líder mundial de sistemas de BI *código abierto*, ofrece una amplia gama de herramientas orientadas a la integración de información y al análisis inteligente de los datos de su organización. Cuenta con potentes capacidades para la gestión de procesos ETL, informes interactivos, análisis multidimensionales de información OLAP y minería de datos. Todos estos servicios están integrados en una plataforma Web, en la que el usuario puede consultar de una manera fácil e intuitiva.

Los módulos incluidos por Pentaho pueden utilizarse de manera conjunta o de forma separada según las necesidades de su organización. Transforma e integra datos entre sistemas de información existentes y los Data marts que compondrán el sistema BI.

Algunas de sus características más significativas:

- Entorno gráfico de desarrollo.
- Uso de tecnologías estándar: Java, XML, JavaScript.
- Fácil de instalar y configurar.
- Basado en dos tipos de objetos: transformaciones (colección de pasos en un proceso ETL) y trabajos (orquestación de transformaciones).
- Permite rápida y eficientemente extraer datos, transformarlos, limpiarlos, validarlos, cargarlos, etc. desde donde se encuentren.
- Librería de transformaciones completa con más de 100 objetos de mapeo.
- Código 100% Java, multiplataforma y soporte de una amplia cantidad de fuentes de datos, incluyendo aplicaciones integradas, sobre 30 plataformas propietarias y de código abierto, archivos planos, documentos Excel, y más.
- Soporte avanzado de Almacenes de Datos para cambios lentos y dimensiones basura.
- Herramienta gráfica de muy fácil uso (control lógico de flujo).

- Basado en repositorio facilita el uso de componentes de transformación, colaboración y administración de modelos, conexiones, logs, etc.
- Rendimiento y escalabilidad de clase empresarial con soporte a procesamiento masivo paralelo (MPP) a través de ejecución en clúster.
- Monitoreo y depurador integrado.
- Calendario programador de transformaciones y trabajos (Scheduler)
- Incluye cuatro herramientas:
    - SPOON: para diseñar transformaciones y trabajos usando un entorno gráfico.
    - PAN: para ejecutar transformaciones diseñadas con Spoon desde la línea de comandos.
    - KITCHEN: para ejecutar trabajos diseñados con Spoon desde la línea de comando.
    - CARTE: para ejecutar trabajos y transformaciones en un servidor Web de forma remota.

Spoon

Es el entorno de desarrollo integrado que ofrece una interfaz gráfica de usuario para la creación y edición de trabajos y transformaciones (Matt Casters, 2010).

Pan y Kitchen

Son muy similares en concepto y uso. La lista de comandos disponibles y la línea de opciones es prácticamente idéntica para ambas herramientas. La única diferencia entre estas herramientas es que Kitchen está diseñado para ejecutar los trabajos, mientras que Pan ejecuta las transformaciones.

Carte

Es un servidor Web sencillo que permite ejecutar transformaciones y trabajos remotamente. Lo hace aceptando archivos XML que contiene la transformación a ejecutar y la configuración de ejecución. También permite monitorear, iniciar y detener remotamente las transformaciones y trabajos que corren en el servidor Carte.

1.5 Conclusiones Parciales

Del estudio realizado sobre los algoritmos y métodos que se aplican en la integración de datos, se concluye que los algoritmos están estrechamente vinculados a la localización de anomalías en especial a la detección de duplicados, además los algoritmos BDK y Merge ofrecen soluciones sin restricciones de entrada ni de salida lo que los convierte en algoritmos genéricos más potentes que el resto de los enunciados. También en este capítulo se describieron las etapas que intervienen en los procesos ETL haciendo énfasis en la limpieza de datos, pues no basta solo con extraer datos desde fuentes heterogéneas, es necesario detectar las anomalías y corregirlas mediante transformaciones del proceso de limpieza que garanticen la calidad de los datos. Además se realizó un estudio de diversas herramientas de software propietario y libre que se utilizan en los procesos ETL, de las cuales se seleccionó el el PDI en su versión libre por su importancia e impacto a nivel mundial y ser una herramienta de código abierto que garantiza a cualquier persona el derecho de modificar y redistribuir el código libremente.

CAPÍTULO II: ANÁLISIS DE LA HERRAMIENTA SPOON Y LOS SUBSISTEMAS DE KIMBALL.

En este capítulo se expone la estructura interna del PDI, haciendo énfasis en la herramienta con interfaz gráfica Spoon. Se muestran los requisitos de hardware y software necesarios para su instalación y se describen los componentes de la herramienta: repositorios, transformaciones, pasos, saltos y trabajos, para lo cual se presentan ejemplos del uso de los mismos. Además, como parte esencial del capítulo se muestran los 34 subsistemas de Kimball y los patrones que surgen a partir de ellos para realizar una solución ETL.

2.1 Requisitos de hardware y software para la instalación del PDI

Para el uso de la suite Pentaho es necesario disponer de requisitos específicos de hardware y tener en cuenta la instalación previa de varios programas los cuales se mencionan a continuación:

Requisitos mínimos de hardware

- Procesador de arquitectura Pentium de 2.0 GHZ
- 768 MB de memoria RAM
- Disco Duro con al menos 2 GB libres

Requisitos de software

- Instalar el *jdk* en una versión posterior a la 5.
- Añadirle a la variable de entorno *Path* el camino del *jdk*.
- Crear la variable de entorno JAVA_HOME y ponerle el camino del *jdk*.
- Ejecutar el Spoon corriendo los ficheros *Spoon.bat* para Windows y *Spoon.sh* para Linux.

2.2 Estructura interna del PDI

En la figura 2.1 se muestra la estructura interna del PDI, en la cual se puede apreciar cómo se realizan los procesos dentro de la herramienta.

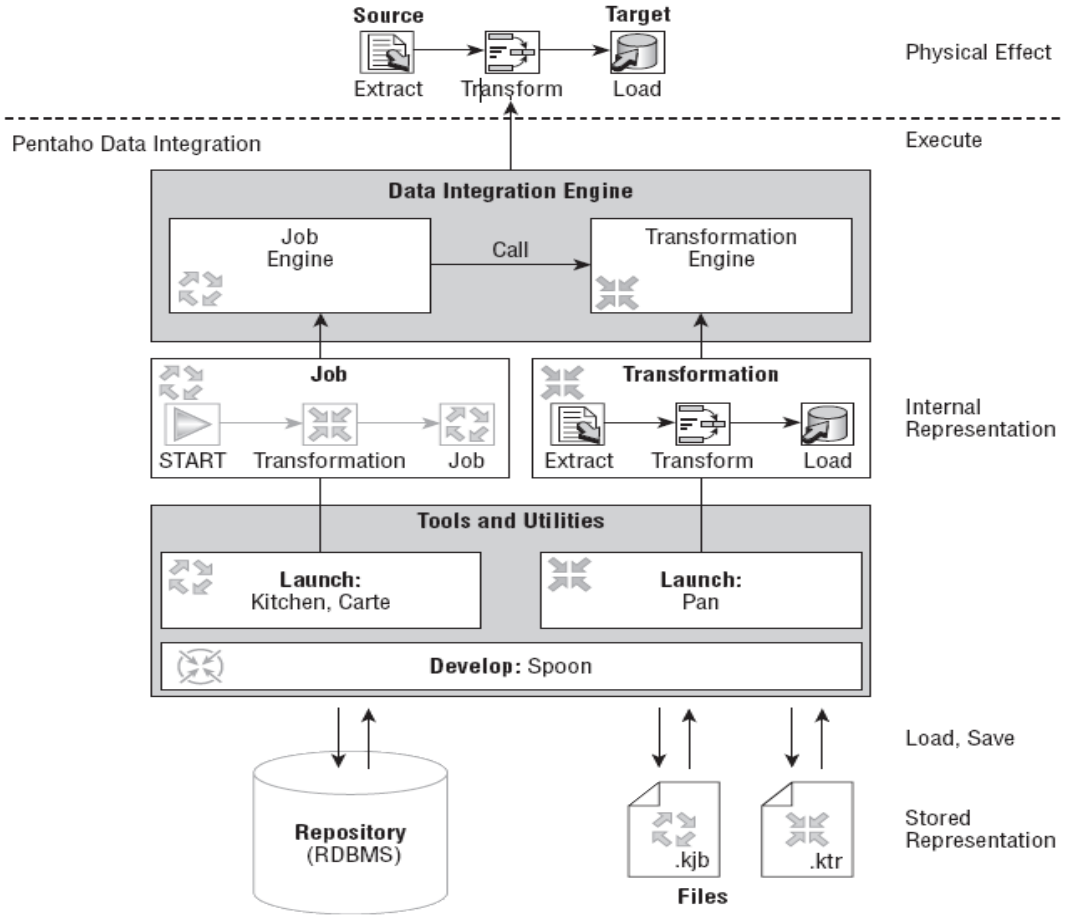

***Figura 2.1: 2.2 Estructura interna del* PDI**

A continuación se argumenta cómo funciona la estructura interna del PDI de abajo hacia arriba como se muestra en la figura 2.1:

- La representación almacenada de los repositorios desde los cuales se guardan y se cargan los archivos con extensión **.kjb** pertenecientes a los trabajos y los **.ktr** referentes a las transformaciones.

- Las herramientas y utilidades de desarrollo (Spoon) y de ejecución (Kitchen, Pan y Carte).

- La representación interna de cómo se realizan los trabajos (se hace un llamado para ejecutar una transformación) y en las transformaciones se extraen los datos desde las diversas fuentes, se procesan a través de diversos pasos y finalmente se exportan hacia la salida prevista que puede ser una tabla en una base de datos, un fichero de texto plano, una hoja de cálculo Excel, etc.

- Muestra en la parte de ejecución la ingeniería para la integración de datos exponiendo como para realizar un trabajo se realiza una llamada a la transformación pertinente. Ejemplifica a efectos físicos lo que se muestra en el Spoon como herramienta gráfica dentro del software.

2.3 Componentes de la herramienta Spoon

El Spoon está compuesto por disímiles componentes que permiten de forma gráfica realizar los procesos de ETL, los repositorios y ficheros donde se almacenan los metadatos, los pasos, los saltos por donde fluyen los datos, las transformaciones y los trabajos. A continuación se muestran y ejemplifican estas componentes.

2.3.1 Repositorios y ficheros

Los repositorios se utilizan para guardar información acerca de los datos y componentes que se utilizan en los procesos ETL, o sea sirven para almacenar y consultar los metadatos de la aplicación.

Los repositorios en el Spoon pueden ser creados de dos formas diferentes en dependencia del tipo de almacenamiento que se utilice para guardar las transformaciones, trabajos y los metadatos:

- Repositorio almacenado en una carpeta determinada donde se almacenan los ficheros.

- Repositorio que usa una base de datos relacional determinada para almacenar los metadatos de los procesos ETL.

En la figura 2.2 se muestran los dos tipos de repositorios que pueden ser creados.

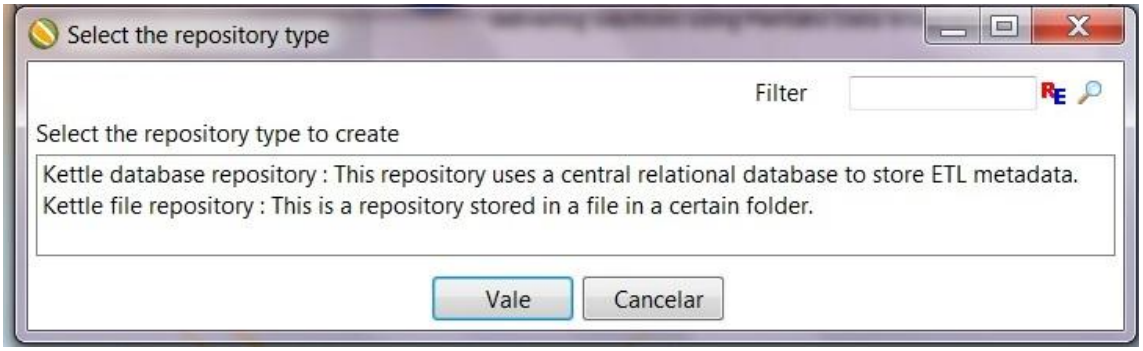

Figura 2.2: Tipos de Repositorios

Cada forma presenta determinadas características, como se muestra en la figura 2.3. Al utilizar una base de datos para el repositorio, se tiene guardada mayor cantidad de información acerca de lo que se ha implementado con la herramienta, pues se crean por defecto 42 tablas donde se almacenan los metadatos. Además esta vía permite aprovechar las ventajas que ofrece una base de datos como es que brinda la posibilidad de hacer copias de seguridad o resguardos para que no se pierda la información y para reutilizarlo en otros proyectos. El *repositorio* dispone de una base de datos, con una estructura especial, donde son guardados en forma de tablas que almacenan los metadatos de las transformaciones y trabajos construidos. Puede ser útil para el trabajo en equipo y para administrar un lugar centralizado donde se almacena y registra todo lo referente a los procesos realizados (Cruz, 2012)

Otra forma de guardar los elementos que van a ser diseñados es a través de *ficheros***,** en los cuales son guardadas las transformaciones y trabajos a nivel de sistemas de archivos que incluyen los metadatos, (con extensión .**ktr** para las transformaciones y .**kjb** para los trabajos). El repositorio es almacenado en una carpeta, lo cual ofrece gran visibilidad del proyecto, permite reutilizarlo y exportarlo con gran facilidad.

En ambos casos la herramienta permite exportar los repositorios al formato XML, muy utilizado actualmente por disímiles herramientas.

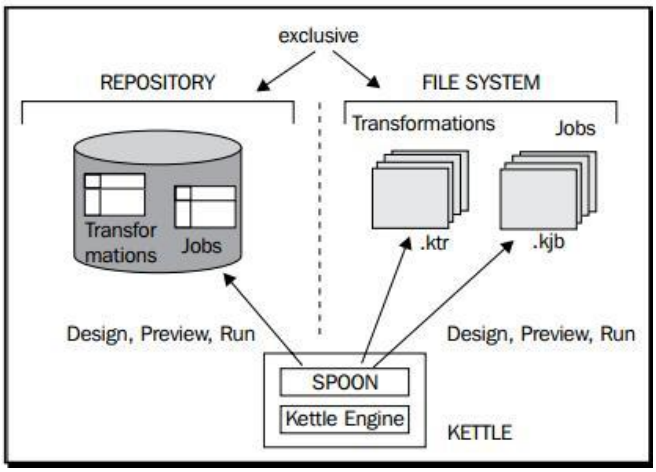

Figura 2.3 2.3.2

2.3.2 Pasos

Los *pasos* están agrupados por categorías y cada uno está diseñado para cumplir una función determinada. Cada paso tiene una ventana de configuración específica, donde se determinan los elementos a tratar y su forma de comportamiento.

Características generales de los pasos (Matt Casters, 2010):

- Un paso es un elemento central en la transformación.
- Se representa gráficamente en forma de un ícono.
- Un paso tiene que tener un nombre único en una sola transformación.
- Virtualmente, cada paso es capaz de leer y escribir filas de datos (la única excepción es el paso Generar filas, que sólo escribe los datos).
- La mayoría de los pasos pueden tener múltiples saltos de salida. Un paso puede ser configurado para distribuir o copiar datos a sus saltos de salidas. Al distribuir los datos, el paso alterna entre todos los saltos salientes para cada fila de. Al copiar datos, cada fila se envía a todos los saltos salientes.
- Cuando se ejecuta una transformación, se inician una o más copias de cada paso, cada uno se ejecuta en su propio hilo. Durante la carrera, todas los pasos copias se

ejecutan de forma simultánea, con filas de datos que constantemente fluyen a través de sus saltos de conexión.

- Más allá de estas capacidades estándar, cada paso, obviamente, tiene una funcionalidad distinta que está representado por el tipo de paso.

Existen pasos de transformaciones y otros de trabajos, estos se clasifican en categorías para hacer más fácil su manejo dentro de la herramienta.

Pasos disponibles para las transformaciones

- Entrada: permite recuperar datos desde bases de datos (JDBC), Access, CSV, ficheros Excel, Tablas, LDAP, Mondrian, RSS u otras fuentes.
- Salida: permite cargar datos en bases de datos u otros formatos de salida.
- Transformación: permite realizar operaciones con datos como filtrar, ordenar, partir campos, añadir nuevos campos, mapear, etc.
- Utilidades: permite operar con filas o columnas y realizar otras operaciones como enviar un email, escribir en ficheros de log.
- Flujo: permite realizar operaciones con el flujo de datos como fusionar, detectar flujos vacíos, realizar operaciones diferentes en función de una condición.
- Scripting: permite crear scripts de JavaScript, SQL, expresiones regulares, fórmulas y expresiones Java.
- Búsqueda de datos: permite añadir información al flujo de datos mediante la búsqueda en bases de datos y otras fuentes.
- Uniones: permite unir filas en función de diferentes criterios.
- Almacén de Datos: permite trabajar con dimensiones SCD y realizar búsquedas y actualizaciones en combinación.
- Validación: permite validar tarjetas de crédito, datos, direcciones de correo o XSD.
- Estadística: permite realizar operaciones estadísticas sobre un flujo de datos.

- Trabajos: permite realizar operaciones propias de un trabajo.
- Mapeado: permite realizar el mapeo entre campos de entrada y salida.
- Embebido: permite realizar operaciones con sockets.
- Experimental: incluye los pasos en fase de validación.
- Obsoleto: incluye los pasos que desaparecerán en la siguiente versión del producto.
- Carga bulk: permite realizar cargas bulk a Infobright, lucidDB, MomentDB y Oracle.
- Historial: recopila los pasos frecuentemente usados por el desarrollador.

Pasos disponibles para trabajos

- Generales: permite iniciar un trabajo, ejecutar transformaciones o trabajos entre otras operaciones.
- Correo: permite enviar correos, recuperar cuentas o validarlas.
- Gestión de ficheros: permite realizar operaciones con ficheros como crear, borrar, comparar o comprimir.
- Condiciones: permite realizar comprobaciones necesarias para procesos ETL como la existencia de un fichero, una carpeta o una tabla.
- Scripting: permite crear scripts de JavaScript, SQL y Shell.
- Carga bulk: permite realizar cargas bulk a MySQL, MSSQL, Access y ficheros.
- XML: permite validar XML y XSD.
- Enviar ficheros: permite enviar o recoger ficheros desde FTP y SFTP.
- Repositorio: permite realizar operaciones con el repositorio de transformaciones y trabajos.

2.3.3 Saltos

Los saltos son los componentes conectores que indican el orden de ejecución de cada paso (no empezando la ejecución del elemento siguiente hasta que el anterior no ha concluido). A través de ellos viajan los metadatos en forma de registros y tablas, como se muestra en la figura 2.4.

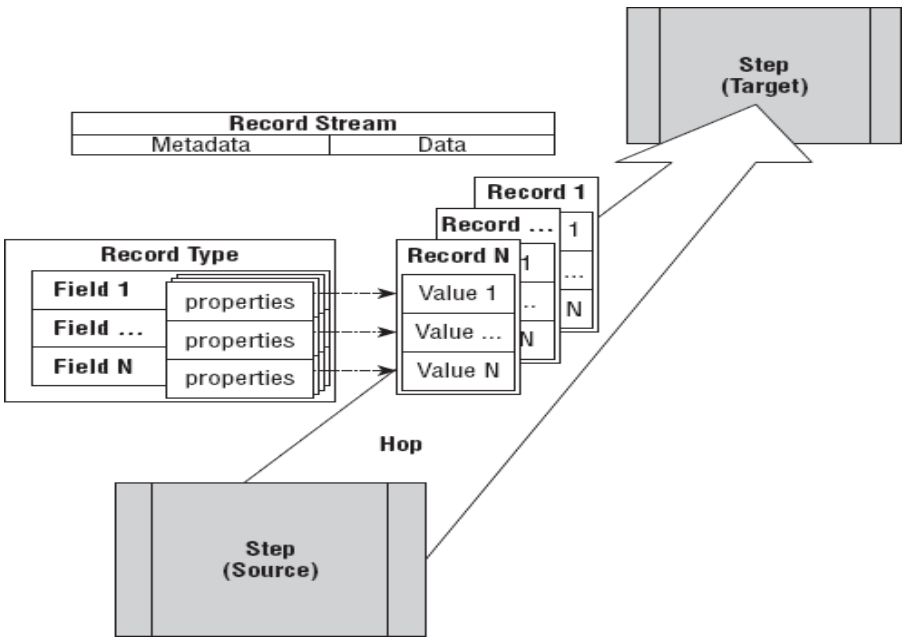

Figura 2.2: Estructura de los saltos

Saltos de transformación

Un salto, es representado por una flecha entre dos pasos, define la ruta de datos entre los pasos. El salto también representa un búfer fila denominado fila situada entre dos pasos. El tamaño de los conjuntos de filas se puede definir en la configuración de transformación. Cuando una fila está llena, el paso que escribe filas se detiene. Cuando una fila está vacía, la etapa de hacer la lectura va a esperar un poco hasta que las filas estén disponibles de nuevo (Matt Casters, 2010).

Durante la creación de nuevos saltos, es importante recordar que los bucles no están permitidos en las transformaciones. Esto se debe a que una transformación depende en gran

medida de los pasos anteriores para determinar los valores de los campos que se pasan de una etapa a otra.

Saltos de trabajo

Los saltos se utilizan en un trabajo para definir una ruta de ejecución entre las entradas del trabajo. Esto se hace en la forma de la relación entre dos entradas de trabajo, así como un tipo de evaluación de los resultados. Este tipo de evaluación puede ser cualquiera de los siguientes:

- Incondicional: significa que la próxima entrada de trabajo se ejecuta sin importar lo que sucedió en la anterior. Este tipo de evaluación se indica mediante un ícono de bloqueo sobre una flecha de salto de negro.
- Siga cuando el resultado es cierto: Este camino sigue cuando el resultado de la ejecución anterior de la entrada de trabajo es cierto. Este tipo de salto se indica con un ícono verde (éxito) dibujado sobre un salto una flecha verde.
- Siga cuando el resultado es falso: Este camino se sigue cuando el resultado de la ejecución de la entrada anterior de trabajo es falsa o incorrecta. Esto se indica mediante un ícono de stop de color rojo dibujando sobre un salto una flecha roja.

El tipo de evaluación se puede establecer mediante el menú del botón derecho del salto o a través de las opciones, haciendo clic en los iconos pequeños de los saltos.

2.3.4 Transformaciones

Una transformación se compone de pasos, que están enlazados entre sí a través de los saltos. Es el elemento básico de diseño de los procesos ETL en PDI. 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).

Como se muestra en la figura 2.5, la transformación es una entidad hecha de pasos unidos por saltos. Estos pasos y saltos construyen caminos por los que fluyen los datos. Los datos entran o se crean en un paso, el paso aplica algún tipo de transformación a ella y finalmente

deja paso a los datos. Por lo tanto, se dice que una transformación es orientada al flujo de datos.

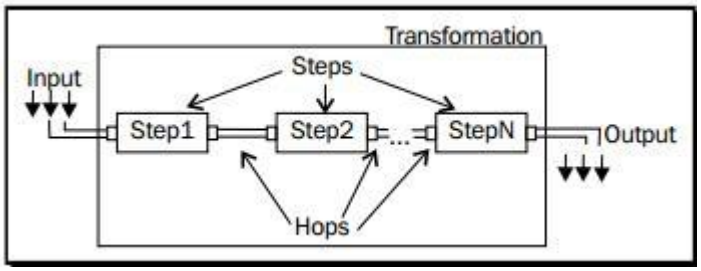

Figura 2.3: Estructura de una transformación

En la figura 2.6 se muestra un ejemplo de una transformación, en la cual se muestra como paso de entrada una base de datos Microsoft Access, seguidamente se ordenan las filas del archivo por el campo determinado, para en el siguiente paso eliminar los elementos duplicados y finalmente se guardan los datos en un fichero Excel.

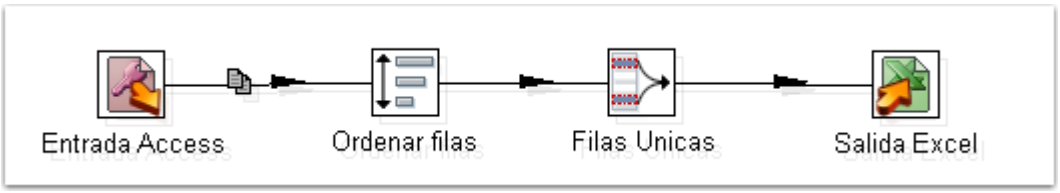

Figura 2.6: Ejemplo de una transformación utilizando el Spoon

2.3.5 Trabajos

Un trabajo (job) es un conjunto sencillo o complejo de tareas con el objetivo de realizar una acción determinada. En los trabajos se puede utilizar pasos específicos (que son diferentes a los disponibles en las transformaciones) como recibir un fichero vía ftp, mandar un email, ejecutar un comando, entre otros. Además, se pueden ejecutar una o varias transformaciones diseñadas previamente y orquestar una secuencia de ejecución de ellas.

En la figura 2.7 se muestra un ejemplo de un trabajo, el cual se inicia con el paso básico de *START* y en donde se orquestan varias transformaciones con el objetivo de llenar diferente dimensiones y la tabla de hecho, por motivos específicos el sistema muestra interés en saber si la dimensión libro se llena correctamente y en caso que no lo sea ser notificado

inmediatamente a través del email, en el caso de las demás transformaciones solo se asegura que de contener algún error se aborte el trabajo inmediatamente.

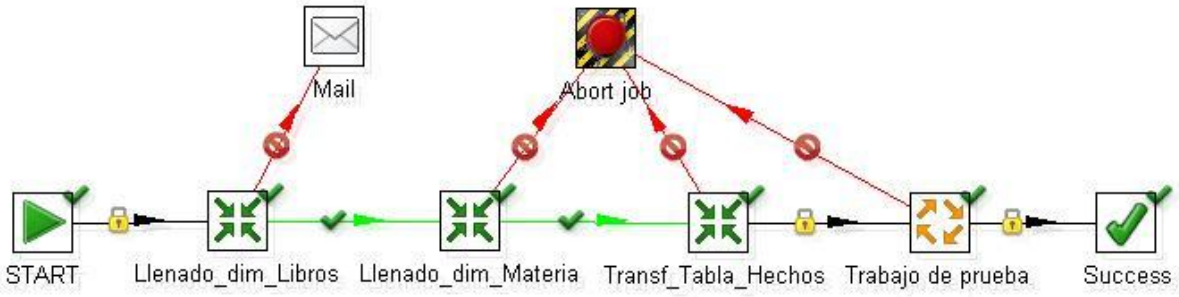

Figura 2.7: Ejemplo de trabajo

2.4 Subsistemas de Ralph Kimball

En este capítulo se muestran y explican los 34 subsistemas de ETL tal y como los define Ralph Kimball (Kimball and Caserta, 2004). La lista de los subsistemas puede verse como una arquitectura general para ETL, siendo requisitos magníficos para validar cualquier solución ETL disponible, no sólo para el PDI sino también para cualquier software de integración de datos.

En un inicio se crearon 38 subsistemas para añadir estructura a las diversas tareas que son parte de un proyecto ETL (Kimball and Caserta, 2004); más adelante en el 2008, Wiley publicó la segunda edición de uno de los libros más vendidos de inteligencia de negocio el Data Warehouse Lifecycle Toolkit, también por Ralph Kimball y sus colegas del Kimball Group (Kimball, 2008). En ese libro, los subsistemas se reestructuraron por segunda vez, lo que resultó en una lista ligeramente condensada que consta de 34 subsistemas de ETL, por otra parte el propio Kimball declaró que en los próximos tiempos pudieran seguir apareciendo otros subsistemas.

Las cuatro áreas principales que conforman los 34 subsistemas son:

- Extracción: obtención de los datos de sus fuentes de origen.

- Limpieza y conformación: consiste en acciones que permiten validar y aumentar la calidad de la información.
- Entrega: carga y actualización de los datos del almacén de datos.
- Gestión: Controla y supervisa la correcta tramitación de todos los componentes de la solución ETL.

A continuación se argumentan los 34 subsistemas agrupados en las cuatro áreas principales, de acuerdo con (Matt Casters, 2010).

Extracción

- Sistema de perfiles de datos (subsistema 1): consiste en la exploración de los datos para verificar su calidad y si cumple los estándares conforme los requerimientos.
- Cambio de Sistema de captura de datos (subsistema 2): detecta los cambios para refinar los procesos ETL y mejorar su rendimiento.
- Sistema de Extracción (subsistema 3): permite la extracción de datos desde la fuente de origen a la fuente destino.

Limpieza y conformación

- Limpieza de datos (subsistema 4): implementa los procesos de calidad de datos que permite detectar las incoherencias de calidad.
- Rastreo de eventos de errores (subsistema 5): captura todos los errores que proporcionan información valiosa sobre la calidad de datos y permiten la mejora de los mismos.
- Creación de dimensiones de auditoría (subsistema 6): permite crear metadatos asociados a cada tabla. Estos metadatos permiten validar la evolución de la calidad de los datos.
- Sistemas de Deduplicación (subsistema 7): eliminar información redundante de tablas importantes. Requiere cruzar múltiples tablas en múltiples sistemas de

información para detectar el patrón que permite identificar cuando una fila está duplicada.

- Conformación de datos (subsistema 8): permite identificar elementos equivalentes que permiten compartir información entre tablas relacionadas.

Entrega

- Dimensiones lentamente cambiantes (SCD) (subsistema 9): implementa la lógica para crear atributos de variabilidad lenta a lo largo del tiempo.
- Creación de sistemas de llaves subrogadas (subsistema 10): permite crear claves subrogadas independientes para cada tabla.
- Jerarquías (subsistema 11): permite hacer inserciones en estructuras jerárquicas de tablas.
- Dimensiones especiales (subsistema 12): permite crear dimensiones especiales como dimensiones basura, mini-dimensiones, dimensiones estáticas y dimensiones de encogido o enrollado.
- Tablas de hecho (subsistema 13): permite crear tablas de hecho.
- Pipeline de claves subrogadas (subsistema 14): permite remplazar las claves operacionales por las claves subrogadas.
- Constructor de tablas multievaluadas (subsistema 15): permite construir tablas puente para soportar las relaciones N: M.
- Gestión para información tardía (subsistema 16): permite aplicar modificaciones a los procesos en caso que los datos tarden en llegar.
- Gerente de dimensión (subsistema 17): autoridad central que permite crear y publicar dimensiones conformadas.
- Proveedor de tablas de hecho (subsistema 18): permite la gestión de las tablas de hecho.
- Creador de agregados (subsistemas 19): permite gestionar agregados.
- Creador de cubos OLAP (subsistema 20): permite alimentar de datos a esquemas OLAP desde esquema dimensionales relacionales.

- Propagador de datos (subsistema 21): permite preparar información conformada y ser entregada para cualquier propósito especial.

Gestión

- Programador de trabajos (subsistema 22): permite gestionar ETL de la categoría de trabajos.
- Sistema de apoyo (subsistema 23): realiza copias de respaldo de los procesos ETL.
- Reinicio y recuperación (subsistema 24): permite reiniciar un proceso ETL en el caso de error.
- Control de versiones (subsistema 25): permite hacer control de versiones de un proyecto ETL y de los metadatos asociados.
- Migración de versiones (subsistema 26): permite pasar proyectos en fase de prueba a producción mediante versionado.
- Monitorización de flujo de trabajo (subsistema 27): dado que un proceso de ETL es un flujo de trabajo, es necesario monitorizarlos para medir su rendimiento.
- Ordenación (subsistema 28): permite calibrar los procesos ETL para mejorar su rendimiento.
- Linealidad y dependencia (subsistema 29): identifica elementos dependientes. Permite identificar las transformaciones en las que participa o ha participado. Permite la trazabilidad del dato.
- Escalado de problemas (subsistemas 30): soporta la gestión de incidencias.
- Paralelismo / Clustering (subsistema 31): permite el uso de procesos en paralelo, grid computing y clustering para mejorar el rendimiento y reducir el tiempo del proceso.
- Seguridad (subsistemas 32): gestiona el acceso a ETL y metadatos.

- Gerente de complacencia (subsistema 33): permite soportar la legislación vigente respecto la custodia y responsabilidad de datos que debe aplicarse a la organización.
- Repositorio de metadatos (subsistema 34): captura los metadatos de los procesos ETL, de los datos de negocio y de los aspectos técnicos.

2.4.1 Patrones definidos a partir de los subsistemas

Según lo expuesto en (Cruz, 2012) los subsistemas se pueden organizar en patrones que se reconocen como actividades típicas en los procesos ETL.

1. Captura de datos cambiantes: en muchos casos, la extracción de datos se refiere solo a captar aquellos datos que han cambiado de un período 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).

2.5 Conclusiones Parciales

En este capítulo se explicó la estructura interna del PDI, los componentes del Spoon y se expusieron los requisitos de software y hardware que se necesitan para que se ejecute el PDI. Además, se argumentaron los 34 subsistemas definidos por Kimball y se presentó una manera de organizarlos en patrones que se reconocen como actividades típicas en los procesos ETL, demostrándose que los subsistemasson una lista de requisitos para validar cualquier solución ETL, por el rigor con que fueron definidos y las áreas que los conforman.

CAPÍTULO III: ANÁLISIS DE LA INTEGRACIÓN DE DATOS EN EL CASO DE ESTUDIO "RECURSOS HUMANOS"

En este capítulo se implementa el caso de estudio "Recursos Humanos" a partir de las anomalías detectadas en el proceso de integración de datos, se aplican los patrones descritos en el capítulo anterior. Además se muestran los puntos fuertes y posibles débiles de la herramienta PDI descubiertos en la confección del presente trabajo.

3.1 Descripción del caso de estudio "Recursos Humanos"

En el área de Recursos Humanos (RRHH) de la UCLV se manejan grandes cantidades de información referente a los trabajadores, esta información se encuentra dispersa en varias fuentes de datos sin mantener un control estricto sobre ellas, ya que para realizar cualquier modificación se tiene que sobrescribir, haciendo imposible el análisis histórico de la información. El caso de estudio se centra en la integración de los datos desde las diversas fuentes, detectando las anomalías y dándole solución a través de transformaciones implementadas en la herramienta Spoon del PDI.

Para la realización del caso de estudio se cuenta con una base de datos en Access del mes de abril de 2013, que contiene once tablas cada una con sus atributos, utilizando para el análisis del presente trabajo, la tabla Empleado_Gral , la cual contiene 119 campos y un total de 12716 personas. Además se extrajeron varios ficheros Excel de una base de datos en Visual Fox Pro con información actualizada hasta el año 2011 en la cual se almacena información de los trabajadores y de los cuadros de la UCLV.

El objetivo principal de la implementación de este caso de estudio es integrar estas fuentes de datos y realizar procesos de extracción, limpieza, conformación y carga para lograr que los datos finales tengan calidad.

Para ello fue necesario hacer una exploración manual de las fuentes de datos para detectar la existencia de anomalías, de lo cual resultó: la existencia de duplicados, valores ausentes, esquemas no integrados, valores sin estandarizar, etc. Razón por la cual se implementan transformaciones y trabajos para solucionarlas.

Para implementar las transformaciones del caso de estudio se crea la dimensión Persona con los siguientes atributos de tipo cadena (string): *Id_Empleado, No_CI, Nombre, Apellido_1, Id_Categoria, Sexo, Color_Piel, Apellido_2, Id_Grado_Cientifico*.

Se tuvo en cuenta, luego de un estudio de los patrones ,ya mencionados en el capítulo anterior, la utilización de algunos de ellos: patrones de renombrar y decodificar, patrones de limpieza de datos y patrones de validación de datos, es importante aclarar que en su contenido están presentes varios de los subsistemas mencionados por Kimball, ellos son los subsistemas(1, 4, 7, 10, 12, 13, 30) los cuales son implementados con la herramienta PDI para agilizar los procesos ETL.

3.1.1 Transformación "Integración de los ficheros Excel"

En la Figura 3. 1*: Transformación "Integración de los ficheros Excel"* se muestra la transformación "Integración de los ficheros Excel" implementada en el Spoon, para dar solución a una de las anomalías detectadas: existencias de valores duplicados. A continuación se explican los pasos fundamentales que se emplearon en la transformación.

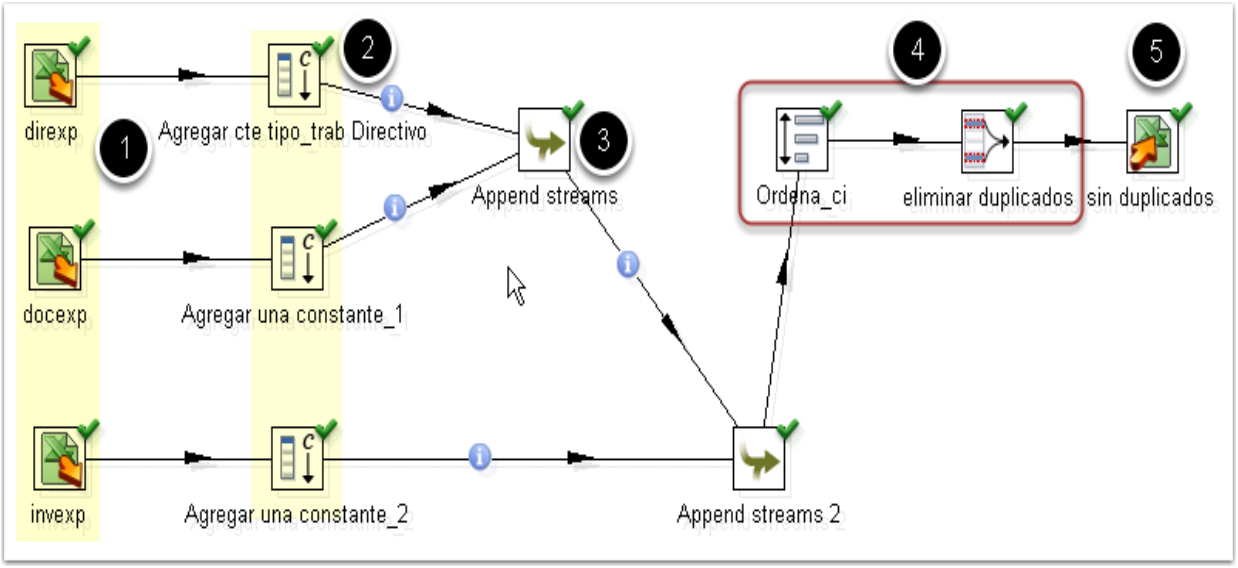

Figura 3. 1*: Transformación "Integración de los ficheros Excel"*

1) Entrada Excel (Categoría Entrada**):** Ofrece la posibilidad de leer los datos de uno o más archivos de Excel. En este paso se cargaron los archivos: direxp.xls, docexp.xls e invexp.xls.

2) Añadir constantes (Categoría Transformar**):** Se utiliza para agregar valores fijos, sólo tiene que especificar el nombre, tipo y valor en forma de una cadena. En este caso se agregó la constante *tipo_trab* para identificar en el fichero de salida el tipo de trabajador.

3) **Anexar corrientes** (Categoría Flujo): El paso "Anexar corrientes" une los datos de dos pasos, lee el procesamiento de la segunda corriente después de que el primero haya terminado. Con este paso se integran los ficheros Excel.

4) Ordenar_CI/Eliminar duplicados (Categoría Transformar**):** Con los pasos Ordenar Filas y Fila Única se puede ordenar por una clave los registros del flujo de datos y eliminar registros duplicados. En el ejemplo, se han utilizado ambos pasos para evitar que pasen al fichero de salida trabajadores duplicados, véase figura 3.1.

5) Salida Excel (Categoría Salida**):** Se almacenan los resultados de la transformación en un fichero de salida Excel.

En la figura 3.2 se muestra una tabla con los resultados de la transformación implementada en la Figura 3. 1*: Transformación "Integración de los ficheros Excel"*

Execution Results

Execution History | Logging | Step Metrics | Performance Graph

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	direxp	0	0	203	203	0	0	0	0	Finished	0.9s	224	-
2	Agregar cte tipo_trab Directivo	0	203	203	0	0	0	0	0	Finished	0.9s	224	-
3	docexp	0	0	1191	1191	0	0	0	0	Finished	1.2s	977	-
4	Agregar una constante_1	0	1191	1191	0	0	0	0	0	Finished	1.2s	977	-
5	invexp	0	0	38	38	0	0	0	0	Finished	0.7s	53	-
6	Agregar una constante_2	0	38	38	0	0	0	0	0	Finished	0.9s	42	-
7	Append streams	0	1394	1394	0	0	0	0	0	Finished	1.4s	1,014	-
8	Append streams 2	0	1432	1432	0	0	0	0	0	Finished	1.7s	856	-
9	Ordena_ci	0	1432	1432	0	0	0	0	0	Finished	1.7s	848	-
10	eliminar duplicados	0	1432	1239	0	0	0	0	0	Finished	1.7s	848	-
11	sin duplicados	0	1239	1239	0	1239	0	0	0	Finished	2.3s	543	-

Figura 3. 2*: Salida del fichero "Integración de los ficheros Excel"*

Del análisis realizado en el caso de estudio, se detectaron 193 filas duplicadas en los ficheros Excel y la solución que se brinda a esta anomalía es eliminar las filas que estén repetidas, lo cual se hace con los pasos **Ordenar_CI/Eliminar duplicados**. El valor de la entrada enmarcado con el color rojo muestra el total de datos producto de la integración de los ficheros Excel, o sea 1432. El valor de la salida enmarcado con el color azul, destaca los datos resultantes luego ejecutada la transformación para eliminar los elementos duplicados.

3.1.2 Transformación "Salida tabla Empleado_General"

En la Figura 3. 3*: Transformación "Salida tabla Empleado_General"*se muestra la transformación "Salida tabla Empleado_General". En este caso el paso utilizado es una entrada tabla de la base de datos de RRHH en Access, para dar solución a las anomalías detectadas en el proceso, se aplican los patrones: renombrar y decodificar, validar datos y limpieza de datos.

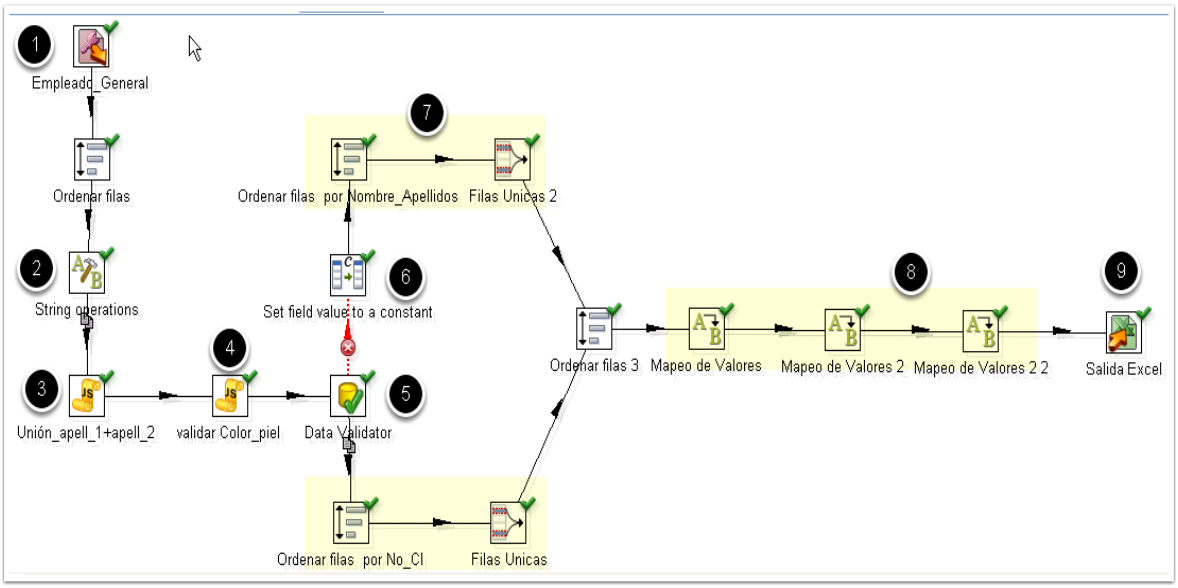

Figura 3. 3*: Transformación "Salida tabla Empleado_General"*

1) Entrada Access (Categoría Entrada**):** Se carga la base de datos que está en el SGBD Microsoft Access y se seleccionan los campos de interés, el resto no se extrae.

2) String Operations (Categoría Transformar**):** En este paso algunos campos fueron convertidos a mayúscula garantizando así la uniformidad de los datos, y se eliminaron los espacios en blanco a ambos lados de varios campos, tal es el caso del No_CI.

3) Valor Java Script Modificado (Categoría Scripting): Permite escribir código JavaScript para realizar operaciones sobre los datos que no se pueden efectuar con un paso específico. El objetivo de este script es unir el campo Nombre con Apellido_1 y Apellido_2.

4) Valor Java Script Modificado (Categoría Scripting): El objetivo de este script es validar el campo Nombre_Apellidos para que solo entren letras al campo y no números.

5) Data Validator: (Categoría Validar): Admite definir reglas de validación simple para características de los campos en el flujo y permite manejar también los errores, en este paso se valida que el dato que entre al campo No_CI tenga longitud once.

6) Set file value to a constant: Este paso reemplaza el campo seleccionado por un valor constante.

7) Ordenar Filas /Filas Únicas (Categoría Transformar**):** Con los pasos Ordenar Filas y Fila Única se y eliminan registros duplicados.

8) Mapeo de valores: Nuevos campos son renombrados con los valores que le son asignados, permitiendo así la estandarización de valores. Ejemplo:

9) Salida Excel (Categoría Salida**)**: Se cargan los datos que resultaron después del proceso de limpieza en un archivo Excel.

Con los pasos **Ordenar Filas /Filas Únicas** se eliminan los elementos duplicados y con el **Mapeo de Valores** se estandarizan los valores**,** como se ilustra en la Figura 3. 4*: Estandarización de valores en la Transformación "Salida tabla Empleado_General"*, observándose que el campo *Id_Grado_Científico* que aparece codificado se cambia por valores.

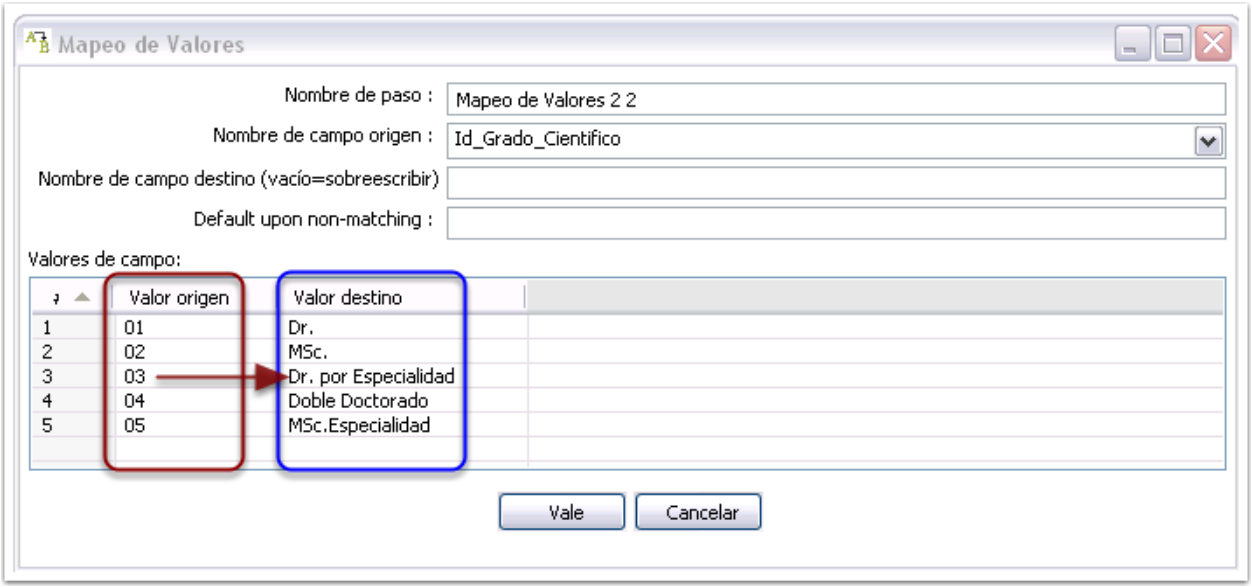

Figura 3. 4*: Estandarización de valores en la Transformación "Salida tabla Empleado_General"*

También se puede identificar en la transformación la utilización del patrón renombrar y decodificar con el paso **Mapeo de Valores** y con **Valor Java Script Modificado** y **Data Validator** el patrón de Validación de Datos.

Otra variante de la Transformación "Salida tabla Empleado_General_2"

La diferencia de esta transformación está en los pasos que se usan para la eliminación de las anomalías, por ejemplo como se muestra en la Figura 3. 5*: Transformación "Salida tabla Empleado_General_2"*, para la detección de valores nulos se emplea el paso **Filtrar Filas** y con un **Replace in String** se le asigna un valor constante a los elementos nulos, luego con el mismo paso **Replace in String** pero en uso diferente se estandarizan los valores.

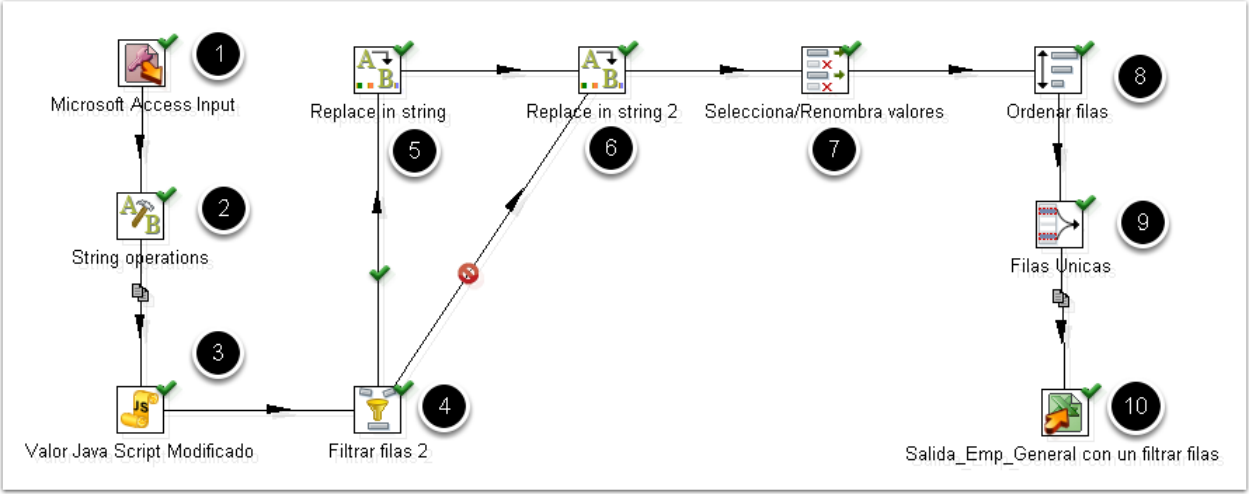

Figura 3. 5*: Transformación "Salida tabla Empleado_General_2"*

4 y 5) Filtrar Filas (Categoría Flow), este paso permite filtrar las filas en función de la condición que se exprese, por ejemplo en este caso se comprueba si el valor especificado del No_CI es nulo o no? , véase Figura 3. 6*: Buscando existencia de valores nulos*. Si es nulo se procede a remplazar esos valores con el paso **Replace in string** (categoría Transformar), el cual permite realizar la sustitución de valores, por los elementos deseados.

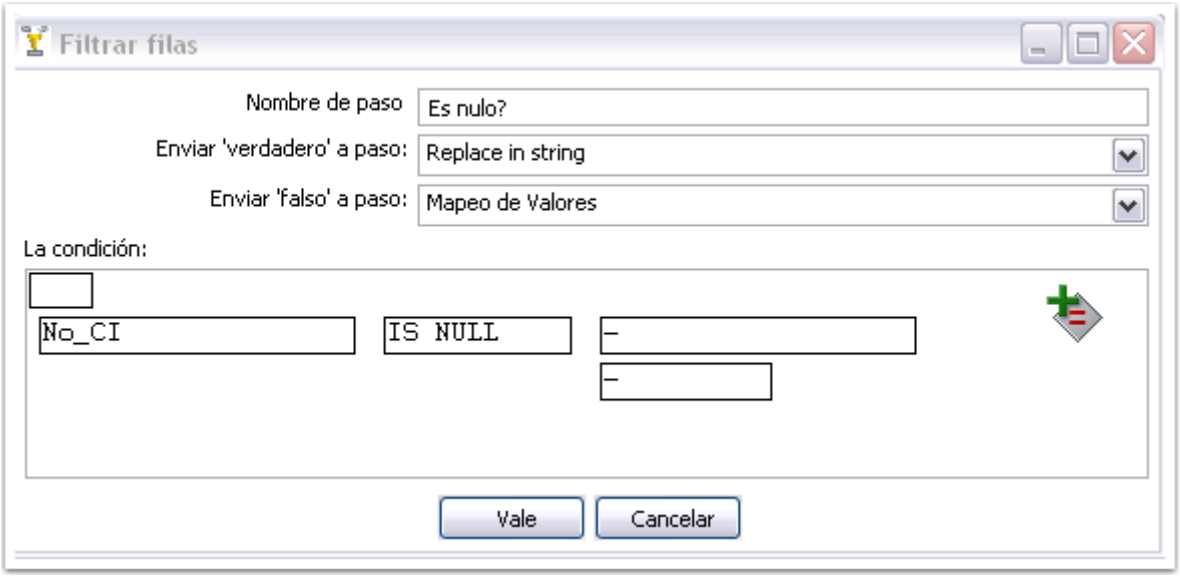

Figura 3. 6*: Buscando existencia de valores nulos*

6) Replace in string2: En este paso son estandarizados algunos valores, ver la configuración del paso en la Figura 3. 7*: Estandarizando valores.*

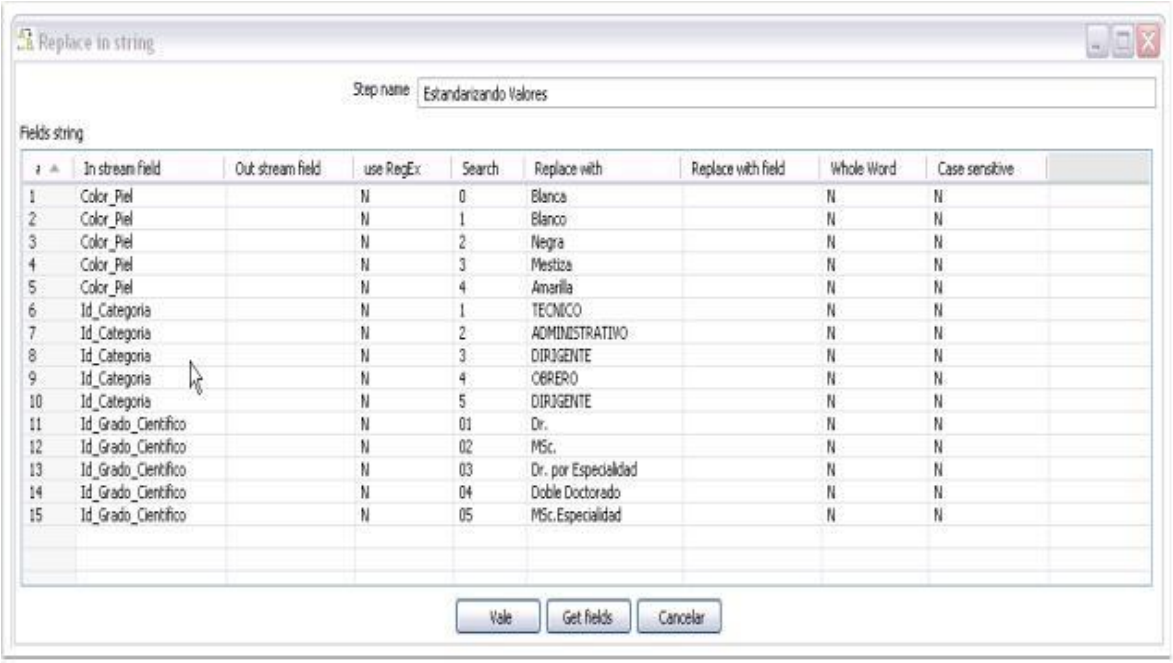

Figura 3. 7*: Estandarizando valores.*

3.1.3 Transformación "Resultado_Final"

En la Figura 3. 7*: Estandarizando valores.* se muestra la integración de los resultados obtenidos de la transformación "Integración de los ficheros Excel" y "Salida tabla Empleado_General".

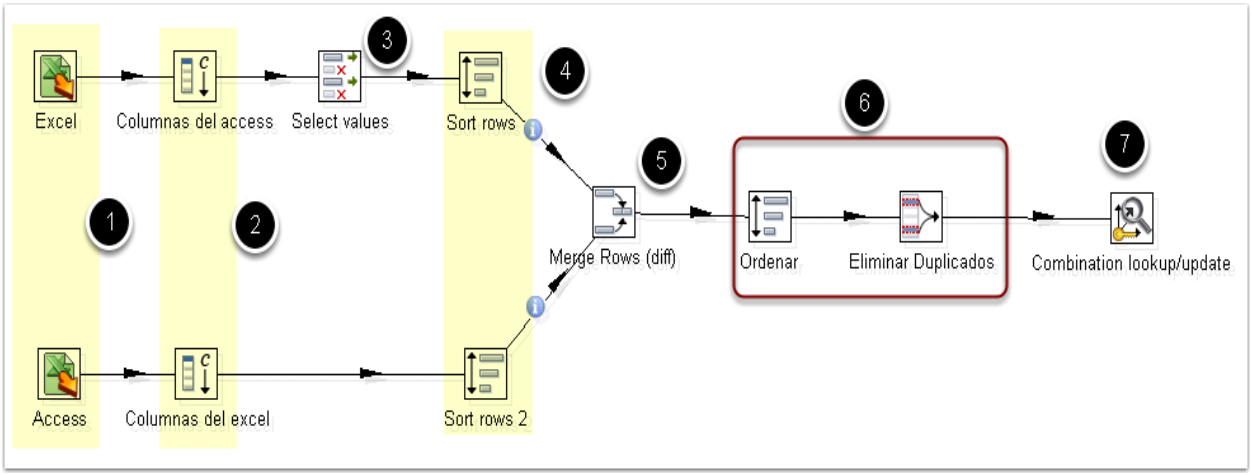

Figura 3. 8*: Transformación "Resultado Final"*

A continuación se explica detalladamente cada paso de la transformación que se observa en la figura 3.8.

1) Entrada Excel (Categoría Entrada**)**: Se cargan los archivos resultantes de las transformaciones "Integración de los ficheros Excel" y "Salida tabla Empleado_General".

2) Añadir constantes (Categoría Transformar**)**: Se agrega en ambos flujos valores para garantizar la similitud de los archivos, manteniendo la misma cantidad de campos.

3) Seleccionar / Renombrar valores (**Categoría Transformar**): En este paso se renombran algunos campos para hacer corresponder con el nombre con los campos del otro flujo.

4) Ordenar Filas (**Categoría Transformar**): Se considera importante añadir este paso previo a un mezclar filas para realizar la combinación correctamente.

5) Merge Rows (**Categoría Unión**): Esta paso combina dos flujos de filas, ordenadas en una determinada clave. Los dos flujos de filas, un flujo de referencia (los datos viejos) y un flujo de comparación (los nuevos datos), se combinan.

6) Ordenar Fila/Filas Únicas (**Categoría Transformar**)**:** Una vez ordenados los valores, se eliminan los duplicados.

7) Combination lookup/update (Categoría Almacén de Datos): Permite la búsqueda y actualización en combinación de la tabla *dim_persona* de la base de datos RRHH. En esencia este paso busca en la tabla correspondiente si existe la fila que se está comparando en ese momento, si existe no se inserta; en caso de no existir la fila el paso genera una nueva llave subrogada e inserta una fila con los campos y la llave subrogada generada. En todo caso, la llave subrogada se agrega al flujo de salida.

En esta transformación se implementa el patrón limpieza de datos, en el cual están contenidos los subsistemas 1, 7 y 10.

A partir del estudio del fichero Excel obtenido como resultado de la transformación "Integración de los ficheros Excel" y la tabla Empleado_General almacenada en Access se

detecta la anomalía *esquemas no integrados*, en un primer caso se localizan campos que están presentes en una fuente y no en la otra y en un segundo caso se observan campos que tienen nombre diferente pero igual contenido. Para solucionar esta anomalía se utilizan los pasos **Añadir constantes** en el primer caso y **Select / Rename Values** en el segundo**.** En la fFigura 3. 9*: Configuración del paso añadir constante* se muestra la configuración del paso *Añadir constantes.*

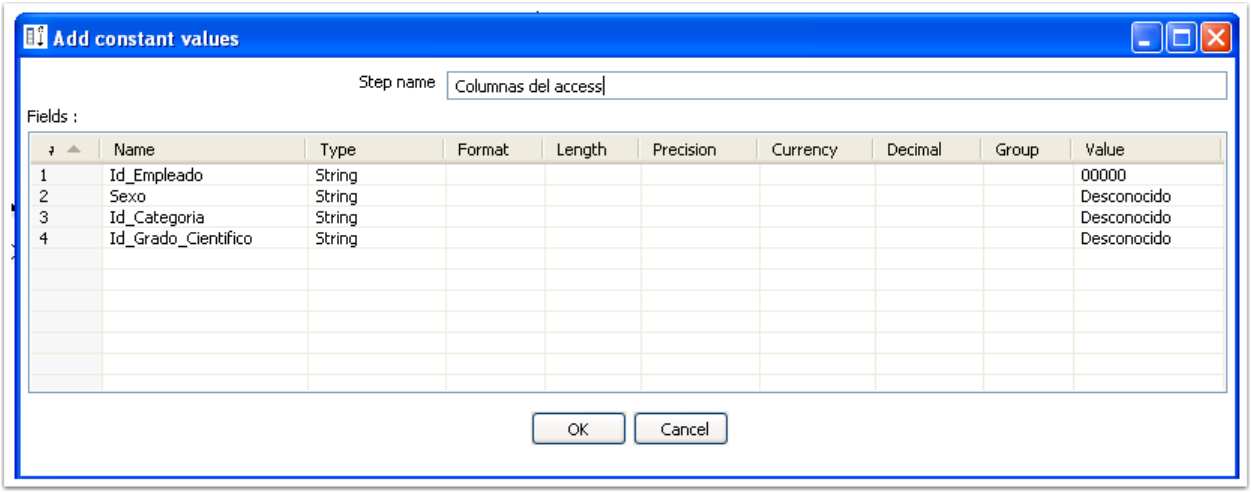

Figura 3. 9*: Configuración del paso añadir constante*

La configuración del paso **Select / Rename Values** se observa en la figura 3.10.

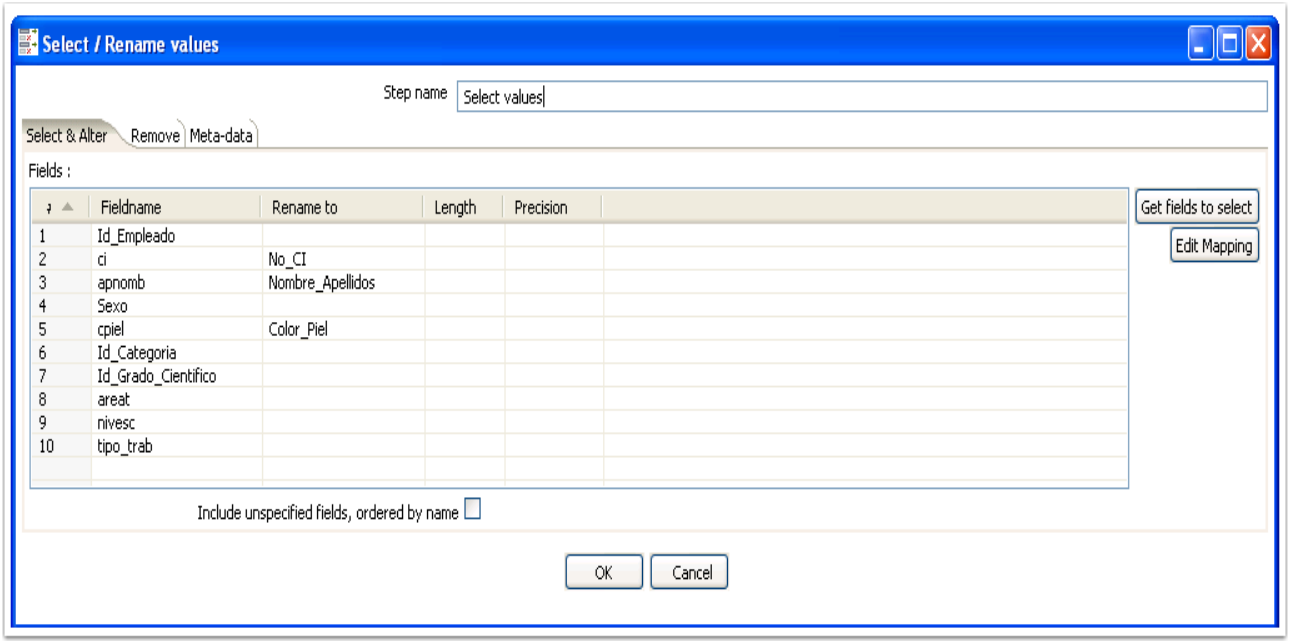

Figura 3. 10*: Configuración del paso Select / Rename Values*

3.1.4 Trabajo donde se orquestan las transformaciones anteriores

Para orquestar las transformaciones se implementa un trabajo, el cual comienza con el paso *START* y posteriormente se introducen las transformaciones en un orden lógico, como se muestra en la figura 3.12.

Figura 3. 11 Orquestación de trabajos

1) Se ejecuta la transformación "Integración de los ficheros Excel" implementada anteriormente.

2) Después de ejecutada dicha transformación comienza a ejecutarse de manera automática la transformación "Salida_Access2" y así sucesivamente se van ejecutando las transformaciones presentes en el flujo de datos.

3.2 Valoración crítica

A continuación se muestran algunas ventajas y sugerencias propuestas a la herramienta Spoon del PDI basados en las características que presenta, las pruebas que se han realizado al software a través del caso de estudio analizado anteriormente y entrevistas realizadas a algunos usuarios del software. Se hace referencia a la versión libre 4.2.1 del Spoon.

Principales ventajas

1. La herramienta contiene pasos que permite escribir código JavaScript para realizar operaciones sobre los datos que no se pueden efectuar con un paso específico.
2. La herramienta ha sido diseñada para cubrir las necesidades en la integración de datos, de manera tal que permite una fácil gestión de los mismos y con un estudio básico del software cualquier usuario puede trabajar directamente con ella.
3. El software soporta gran cantidad de volúmenes de datos.
4. A nivel gráfico, se puede incluir notas con comentarios en el dibujo de los procesos. Con el proyecto kettle-cookbook se puede generar documentación en HTML.
5. Con la gestión de errores en los pasos que lo permiten, se puede interactuar con esos errores y solucionarlos sin terminar el proceso (a pesar de no ser siempre posible).
6. Para su ejecución muestra variantes desde la herramienta (tiempos de respuesta bastante buenos) o a nivel de comandos con Pan (para las transformaciones) y Kitchen (para los trabajos). Son dos utilidades muy sencillas y funcionales que permiten ejecutar los diseños (desde ficheros o desde el repositorio). También se dispone de la herramienta Carte, que es un servidor Web sencillo, el cual permite la ejecución remota de trabajos y transformaciones.
7. Como característica interesante se tiene la encapsulación de transformaciones a través de los mappings, lo cual permite definir transformaciones para procesos repetitivos (similar a una función).

Principales dificultades

1. El software presenta una ayuda online disponible en todo momento con un grupo de especialistas a disposición de la comunidad para responder cualquier tipo de problemática que surja, pero no tiene implementado un manual de usuario dentro de la herramienta, la ayuda que trae es muy pobre y casi inexistente dificultando a los usuarios que no estén en línea por motivos ajenos a la comunidad Pentaho la utilización de algunos componentes de forma tal que la única manera de averiguar el funcionamiento del mismo es probándolo.
2. Posee una mezcla de idiomas (ej. Versión 4.2.1 estudiada), algunas descripciones se encuentran en inglés y otras en español dificultando la comprensión de la herramienta y su manejo.
3. En la configuración de los pasos de entrada a la hora de traer los campos no permite seleccionar de la fuente de entrada el o los campos que desea traer, sino que los carga todos sin importar la cantidad que sean y entonces el usuario debe seleccionar el o los que desea e ir eliminando el resto haciendo el trabajo tedioso. Porque la otra vía es escribir el nombre específico del campo, para lo cual necesita conocer de antemano o buscar en la fuente de entrada dicho nombre y escribirlo exactamente igual. Esta opción evidentemente consume tiempo y puede que no funcione debido a un error humano al teclear el nombre manualmente.
4. Los mensajes del compilador no son muy explícitos, en ocasiones solo informan de la ocurrencia del error y no se argumenta la causa.
5. Las funciones de trabajo con cadenas, números y fechas predefinidas en el paso JavaScript se pueden continuar extendiendo, agregándole otras funciones que se utilizan mucho en los lenguajes de alto nivel, para potenciar aún más los pasos de scripting.

3.3 Conclusiones parciales

En este capítulo se demostró a través del caso de estudio RRHH como erradicar las anomalías presentes en los datos una vez que se desea integrar datos provenientes de

fuentes de datos heterogéneas. Además, se ejemplificaron algunos de los patrones de los procesos ETL descritos en el capítulo anterior. De igual forma se mostraron algunas de las ventajas y dificultades presentadas en el estudio de la herramienta en base a la experiencia del equipo de trabajo y el tiempo de explotación del Spoon en su versión libre 4.2.1.

CONCLUSIONES

1. Se analizaron los algoritmos y métodos que se aplican en la integración de datos.
2. Se determinó que en los procesos de ETL se deben detectar las anomalías existentes en los datos y solucionarlas en la etapa de limpieza.
3. Se argumentaron los subsistemas de Kimball y se organizaron por patrones definidos en los procesos ETL.
4. Se utilizó la herramienta Spoon en la integración de datos del caso de estudio RRHH.
5. Se realizó una valoración crítica del uso de la versión 4.2.1 de la herramienta Spoon del PDI.

RECOMENDACIONES

1. Continuar el estudio de la herramienta, probando otros casos de estudio.
2. Documentar y archivar todos los proyectos que se realicen con la herramienta PDI para su uso posterior.
3. Continuar el estudio de los subsistemas de Kimball para demostrar nuevos patrones en el proceso de ETL.

REFERENCIAS BIBLIOGRÁFICAS

ALON HALEVY, A. R., JOANN ORDILLE 2006. DataIntegration:TheTeenageYears.

BEHRENDT, W., HUTCHINSON, E., JEFFERY, K.G., KALMUS, J., MACNEE, C.A., AND WILSON, M.D. 1993. Using anintelligent agent to mediate multibase information access.

COMMUNITY, A. 2010. *ApatarForge* [Online]. Available: http://www.apatarforge.org/Apatar Open Source Data Integration Community.htm [Accessed 6 de febrero 2013].

COMMUNITY, S. 2009. *SAP BusinessObjects Data Integrator* [Online]. Available: http://www.sap.com/index.epx.htm [Accessed 16 de enero 2013].

CORPORATION. 2010. *Ab initio* [Online]. Available: http://www.abinitio.com/ [Accessed].

CORPORATION, I. 2012. Available: Informatica Data Quality Solution la forma de abordar los problemas de calidad de datos.htm [Accessed 6 de febrero 2013].

CRUZ, R. D. J. M. 2012. *Extracción y documentación de patrones en los procesos ETL*. Ingenieria, Martha Abreu de lasVillas.

GIL, J. V. 2011. *Oracle Data Integrator* [Online]. Available: http://blog/oracle-data-integrator-11g.htm [Accessed 16 de enero 2013].

GRECOL, M. L. 2012. Microsoft SSIS and Pentaho Kettle: A Comparative Study for Three-Tier Data Warehouses.

GUTIÉRREZ, L. A. V. 2010. *CÓMO ABORDAR UN PROYECTO DE BUSINESS INTELLIGENCE EN UNA EMPRESA U ORGANIZACIÓN*. Proyecto de Grado para optar el título de Ingeniero de Sistemas, UNIVERSIDAD EAFIT

HOGG, K. 2009. El análisis de la integración de datos Algoritmos.

HUNGRIA, R. M. C. 2009. Aspectos semánticos de la Integracion de fuentes de Datos

JAVLIN. 2011. *Información de Integración de Datos*

Vista rápida en el mundo de los datos [Online]. Available: http://www.dataintegration.info/etl [Accessed 3/12/2012 2012].

KIMBALL, G. 2008. Data Warehouse Lifecycle Toolkit.

KIMBALL, R. & CASERTA, J. 2004. The Data Warehouse ETL Toolkit. *In:* PUBLISHING, W. (ed.).

LENZERINI, M. 2002. Integracion de Datos una perspectiva teorica.

LEVIN, J. 2008. Open Source ETL tools vs Commerical ETL tools Available from: http://www.jonathanlevin.co.uk/2008/03/open-source-etl-tools-vs-commerical-etl.html].

MATT CASTERS, R. B., JOS VAN DONGEN 2010. Pentaho Kettle Solutions.

NADER, I. J. 2003. *" Sistema de Apoyo Gerencial Universitario "* TESIS DE MAGISTER EN INGENIERÍA DEL SOFTWARE

PAZ, M. L. D. D. L. Year. Pentaho Data Integration: ETL. *In*, 2012.

PETER MC.BRIEN , A. P. 2003. Data Integration by Bi-Directional Schema Transformation Rules.

PO, L. 2008. Improving Data Integration through Disambiguation Techniques.

PRÉSTAMO, M. M. Y. 2004 *Construcción de un Data Warehouse de datos del medio ambiente para la toma de decisiones: aplicación a los datos hidrológicos* Licenciatura en Ingeniería en Sistemas Computacionales, Universidad de las Américas Puebla.

SANZ, M. R. 2010. *Análisis y diseño de un DATA MART para el seguimiento académico de alumnos en un entorno universitario.*

SYBVEN, P. C. 2013. Available: http://www.InfoPrimer.htm [Accessed].

WIKIPEDIA 2011. IBM InfoSphereDataStage.

XIN DONG, A. Y. H., CONG YU 2007. *Data Integration with Uncertainty*.