UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Carga inicial de datos legados del Sistema de Gestión de Seguros al Sistema Integral de Seguros Nacionales

Autor:

Jesús Yanser Vega Labarcena

Tutores:

Ing. Fernando Nápoles Gámez

Ing. Yordany Aguilera Martínez

La Habana, junio de 2017

"Año 59 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____	_____	_____
Jesús Yanser Vega Labarcena	Ing. Fernando Nápoles Gámez	Ing. Yordany Aguilera Martínez
Autor	Tutor	Tutor

DATOS DE CONTACTO

Síntesis de los tutores

Fernando Nápoles Gámez
Ingeniero en Ciencias Informáticas. Graduación 2010
Título de Oro
Cargo actual: Especialista A.
Proyectos en los que ha participado:
Sistema de Gestión Integral de Aduanas (GINA) 2010-2015
Sistema de Planificación de Actividades (SIPAC) 2015
Sistema Integral de Seguros Nacionales (SISEN) 2016-2017
Rol actual:
Arquitecto y administrador de base de datos
Desarrollador

Yordany Aguilera Martínez
Ingeniero en Ciencias Informáticas. Graduación 2016
Cargo actual: Recién Graduado en Adiestramiento.
Proyectos en los que ha participado:
Sistema Integral de Seguros Nacionales (SISEN) 2016-2017
Rol actual: Desarrollador

DEDICATORIA

Dedico el presente trabajo de diploma a mis padres, a mi familia y a mi novia, por todo su apoyo, amor y dedicación.

AGRADECIMIENTOS

Agradezco a mis padres por darme la vida, apoyo y por estar hoy donde estoy.

En general doy gracias a toda mi familia, por preocuparse por mí y apoyarme siempre que lo necesité.

A mi novia por soportarme todo este tiempo, ¡créanme, no es fácil! Y aún hoy, en contra de su voluntad, tener que aguantar que le diga "mi negra".

A mis tutores por ser la guía para lograr este objetivo, en especial a Fernando que tuvo que luchar conmigo en cada momento y nunca me dijo no. A Adalennys que fue como una tutora más y siempre me ofreció su ayuda. A Yisel por ayudarme y escucharme cada vez que llegaba con mis locuras a su casa, ¡claro, por darme café también! A Jorge, el muchacho de la ESEN que siempre me ayudó en todo lo que pudo. A Yoiler por brindarme su ayuda cuando la necesité.

Quiero darle las gracias a todos mis amigos de la UCI que han recorrido este largo camino conmigo. En especial a Yeider y Rafa, pues con ellos he compartido buenos momentos de mi vida. Cómo olvidar ese primer año de la carrera en el que no podíamos ni abrir los ojos para mirar las clases. Para mí fue un gusto formar parte del grupo del Morado (Yeider), el chico de los tenis Converse (Rafa) y el que bailó en calzoncillo (Yo).

Quiero darle las gracias a la UCI pues en ella me formé y conocí muchas personas importantes.

Muchas gracias a todos.

RESUMEN

El presente trabajo implementa una estrategia de carga inicial para el Sistema Integral de Seguros Nacionales, sistema desarrollado por la Universidad de las Ciencias Informáticas para la Empresa de Seguros Nacionales, con el objetivo de sustituir el Sistema de Gestión de Seguros. Para ello se realizó el diseño de la estrategia a través de la Metodología de desarrollo para proyectos de almacenes de datos y como herramienta para la realización de procesos de extracción, transformación y carga se utilizó Pentaho Data Integration, que es una de las herramientas libres más usadas actualmente. Su uso permitió desarrollar exitosamente todas las transformaciones y limpiezas necesarias para lograr la migración de datos, sin perder la integridad de los mismos, hacia una base de datos destino que presenta una estructura diferente y en otro gestor de bases de datos.

ABSTRACT

The present work implements an initial loading strategy for the Comprehensive System of National Insurance, a system developed by the University of Computer Science for the National Insurance Company, with the aim of replacing the Insurance Management System. For this, the strategy was designed through the Development Methodology for data warehouse projects and Pentaho Data Integration was used as a tool for carrying out extraction, transformation and loading processes, which is one of the most free tools Currently used. Its use allowed the successful development of all transformations and cleanings necessary to achieve the migration of data, without losing their referential integrity, to a destination database that presents a different structure and another database manager.

ÍNDICE

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicación (TIC's[1]) son un conjunto de técnicas, desarrollos y dispositivos avanzados que integran funcionalidades de almacenamiento, procesamiento y transmisión de datos (2). La tecnología digital, unida a la aparición de ordenadores cada vez más potentes, ha permitido a la humanidad progresar rápidamente en la ciencia y la técnica, desplegando sus armas más poderosas: la información y el conocimiento (3).

Las TIC's son esenciales para mejorar la productividad de las empresas, la calidad, el control y facilitar la comunicación entre otros beneficios, aunque su aplicación debe llevarse a cabo de forma inteligente. Con la evolución de las TIC's en el mundo, aparece una era llena de avances en la cual la información puede ser consultada desde cualquier lugar y en cualquier momento por más de un usuario o sistema informático, brindando beneficios a sectores tan importantes como la salud, la educación y la economía. En este último, se pueden encontrar cúmulos de información valiosa que necesita ser almacenada de manera segura, sin correr el riesgo de perder datos que puedan ser cruciales para la institución. Con estos nuevos avances Cuba ha comenzado un proceso de informatización del país, que abarca varias esferas. Una de las empresas que se encuentra inmersa es la Empresa de Seguros Nacionales (ESEN), la cual ofrece seguros a personas jurídicas, naturales y para el sector agropecuario. Como resultado de las relaciones existentes entre la ESEN y la Universidad de las Ciencias Informáticas, a petición de la ESEN, se desarrolla el Sistema Integral de Seguros Nacionales (SISEN), encargado de informatizar y centralizar la gestión de los procesos que se desarrollan en la ESEN. Surge con el objetivo de sustituir el Sistema de Gestión de Seguros (SIGES), por la necesidad de contar con un sistema actualizado con las últimas reglamentaciones de la entidad, así como el uso de las nuevas tecnologías. Dado que el SIGES fue desarrollado por un equipo pequeño y no especializado, solo se usaba en algunas provincias y en ocasiones, incluso, se utilizaba con ciertas diferencias o pequeños cambios, que no permitían uniformidad en el trabajo y en los procesos.

Por lo antes expuesto la ESEN necesita que el SISEN tenga toda la información registrada hasta el momento, que constituye el punto de partida para gestionar el resto de la información. Todos estos datos están contenidos en su antecesor SIGES, pero se encuentran en diferentes sistemas de gestión de base de datos (SGBD[2]) y con una estructura de datos distinta, lo que dificulta la carga de la información al nuevo sistema.

[1] Las TIC son el conjunto de tecnologías que permiten el acceso, producción, tratamiento y comunicación de información presentada en diferentes códigos (texto, imagen, sonido, etc.) (1)

[2] (Sistema de gestión de base de datos) o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos. (4)

Se realiza un acuerdo entre la ESEN y la UCI, que es una de las instituciones que se encarga de informatizar el país, de buscar una solución que resuelva las necesidades de la entidad, haciendo posible la carga inicial de los datos.

Por lo anteriormente descrito se plantea como **problema de la investigación**: ¿Cómo suministrar los datos necesarios para el funcionamiento del Sistema de Gestión Integral de Seguros Nacionales (SISEN), legados por Sistema de Gestión de Seguros (SIGES)?

La presente investigación tiene como **objeto de estudio** los procesos de extracción, transformación y carga (ETL), enmarcados en el **campo de acción** Transformaciones de datos.

Para solucionar el problema de la investigación planteado, se identifica como **objetivo general** de la investigación: desarrollar la carga inicial de los datos procedentes del SIGES hacia el SISEN, garantizando la integración de los datos.

El objetivo general se ha desglosado en los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos que permitan darle solución al problema planteado.
- ✓ Definir una estrategia para la carga inicial de los datos procedentes del SIGES hacia el SISEN.
- ✓ Realizar la carga inicial del SIGES hacia el SISEN a partir de la estrategia definida.
- ✓ Validar la integración de los datos cargados.

Se propone la realización de las siguientes **tareas de la investigación** para dar cumplimiento a los objetivos planteados:

- ✓ Estudio sobre las herramientas, tecnologías, metodologías y tendencias actuales propuestas para la realización de migraciones de datos, con el objetivo de identificar y seleccionar las posibles a utilizar en la solución del problema planteado.
- ✓ Análisis de la información a migrar hacia el SISEN.
- ✓ Estudio de la estructura de datos de los sistemas en cuestión.
- ✓ Diseño e implementación de una estrategia de carga inicial desde el SIGES hacia el SISEN.
- ✓ Transformación, mapeo y limpieza de los datos a migrar.
- ✓ Carga de los datos procedentes del SIGES hacia el SISEN.
- ✓ Estudio de técnicas y métodos de evaluación para la validación de la propuesta de solución.
- ✓ Análisis de la solución propuesta a partir de técnicas y métodos aplicables a la investigación.

Métodos Teóricos:

Histórico – Lógico: se utiliza en la elaboración de la fundamentación teórica de la investigación, porque se analizan los almacenes de datos y las estrategias que se utilizan en el proceso de integración de datos, realizándose un estudio histórico acerca de los mismos.
Analítico-Sintético: se evidencia en el análisis de un grupo de estrategias generales que se utilizarán durante el proceso de integración de datos. Las estrategias se analizaron de manera independiente y se sintetizaron para una mejor comprensión.

Métodos Empíricos:

Observación: se utilizó para analizar qué estrategia de integración de datos utilizar para la migración, y por consiguiente se obtuvieron conocimientos que fueron usados en el proceso de integración de los datos.

El presente trabajo se encuentra estructurado en tres capítulos, los cuales se describen a continuación:

Capítulo 1: Fundamentación teórica

En este capítulo se tratan los temas fundamentales de los procesos ETL, así como los tipos de carga que se pueden realizar a través de este proceso. Además, se realiza un estudio de los principales temas de la migración de datos, así como sus procesos y etapas fundamentales. Se abordan las diferentes técnicas de integración de datos. Se fundamenta la selección de la metodología, las herramientas y tecnologías que serán utilizadas.

Capítulo 2: Análisis y diseño

En este capítulo se realiza un estudio del negocio con el objetivo de definir las reglas del mismo. Se analizan los sistemas involucrados en la migración de datos, así como las herramientas utilizadas y los artefactos generados por la metodología. Todo esto detallado en la propuesta de una estrategia de carga inicial, que ajusta la metodología a las particularidades del problema.

Capítulo 3: Implementación y validación

En este capítulo se hace referencia a la implementación de la propuesta de solución, que es una estrategia de carga inicial. Además, se realizan las pruebas de integración de los datos que incluyen la validación de los resultados de la migración.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se tratan los temas fundamentales de los procesos ETL, así como los tipos de carga que se pueden realizar a través de este proceso. Además, se realiza un estudio de los principales temas de la migración de datos, así como sus procesos y etapas fundamentales. Se abordan las diferentes técnicas de integración de datos. Se fundamenta la selección de la metodología, las herramientas y tecnologías que serán utilizadas.

1.2 ¿Qué es ETL?

Es un término inglés de las siglas Extract-Transform-Load que significan Extraer, Transformar y Cargar y se refiere a los datos en una empresa. ETL es el proceso que organiza el flujo de los datos entre diferentes sistemas en una organización y aporta los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos. ETL forma parte de la Inteligencia Empresarial (Business Intelligence), también llamado "Gestión de los Datos" (Data Management) (5).

En los procesos ETL existen dos tipos de carga definidas (6). Aunque se realice una carga u otra van a estar presentes las tres fases de los procesos ETL (7):

Extraer: es la primera parte del proceso ETL y es la encargada de extraer los datos desde los sistemas de origen. Cada sistema separado puede usar una organización diferente de los datos o formatos distintos. Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes.

Transformación: en la fase de transformación se aplicarán las modificaciones a los datos que así lo requieran, es una serie de **reglas de negocio** o funciones sobre los datos extraídos para convertirlos en datos que serán cargados en el sistema destino.

Carga: es la última fase, momento en el que los datos, después de ser transformados, son cargados en el sistema de destino. Es la fase que interactúa directamente con la base de datos de destino. Al realizar esta operación se aplicarán todas las restricciones y triggers[3] (disparadores, por su significado en español) que se hayan definido en ella. Estas restricciones y triggers (si están bien definidos) contribuyen a que se garantice la calidad de los datos en el proceso ETL y deben ser tenidos en cuenta.

[3] Un **trigger** (o disparador) en una base de datos es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE) (8) (9) (10).

1.2.1 Tipos de cargas ETL

Cuando se utiliza una herramienta ETL para cargar datos, esta nos permite hacerlo utilizando cualquiera de los dos tipos de carga que menciona Ralph Kimball y Joe Caserta en el libro “The Data Warehouse ETL Toolkit”:

Carga inicial: durante la carga inicial, la captura de cambios en los datos de origen no es importante porque es muy probable que se extraiga toda la fuente de datos o una porción de la misma desde un punto predeterminado en el tiempo (11) (12). Este tipo de carga solo se realiza una vez y no se actualiza más la base de datos.

Carga incremental: su propósito es capturar solo los datos que cambiaron en las fuentes desde la última extracción, lo que la hace más eficaz que realizar cargas completas, para luego eliminar y volver a cargar los datos que ya existían más los nuevos (11) (12). Este tipo de carga se realiza de forma periódica con el objetivo de capturar los cambios en los datos y poder mantener actualizado el destino.

Esta migración debe realizarse mediante el proceso ETL porque la estructura de ambas bases de datos es diferente y necesitan ser aplicados cambios sobre los datos. Después de realizar el estudio de los tipos de carga, se selecciona para el desarrollo del trabajo la Carga inicial, debido a que solo se necesita cargar una vez la base de datos del SISEN para que este quede operacional. Luego de cargarse los datos en el SISEN, el SIGES dejaría de ser utilizado por la ESEN, por lo que no sería necesario realizar cargas incrementales para actualizar los datos.

1.3 Herramientas ETL

El proceso ETL brinda facilidades para la migración de datos, además de aportar los métodos y herramientas necesarias para llevar a cabo las transformaciones que puedan ser requeridas. En ocasiones cuando se implanta un sistema se dispone de otro, por lo que se hace imprescindible las fases del proceso ETL: extracción, transformación y carga de los datos.

Entre las herramientas más conocidas en el mercado, podemos encontrar algunas como:

- ✓ Talend.
- ✓ Oracle Warehouse Builder (Enterprise ETL Option).
- ✓ Pentaho Data Integration Kettle.
- ✓ IBM Cognos Data Manager.

Todo proceso ETL cuenta de tres fases por las que pasa el flujo de datos: extracción, transformación y carga. Es importante que durante todo el proceso se mantenga controlado el flujo de datos. Un mal desempeño durante cualquiera de las fases puede llevar a la pérdida de información vital.

1.3.1 Talend

Ventajas:

- ✓ Es compatible con Microsoft SQL Server Integration Services (SSIS), funciona en Windows, Unix y Linux.
- ✓ Se puede conectar con: Oracle, DB2, MySQL, Sybase y PostgreSQL.
- ✓ El código fuente Java/Eclipse está disponible para su descarga y personalización.
- ✓ La interfaz gráfica de usuario (GUI) para importar metadatos, configuraciones, uniones de componentes y generación de código proporciona ganancias en productividad para desarrolladores y acaba siendo más rápida que la programación estándar.

Desventajas:

- ✓ Necesita de Java Data Base Connectivity (API que permite la realización de operaciones sobre bases de datos) para acceder a las fuentes.
- ✓ No tiene proceso automático de separación y redistribución de datos, lo cual puede generar cuellos de botella.
- ✓ Es una herramienta poco intuitiva y con una curva de aprendizaje compleja (13) (14).

1.3.2 Enterprise ETL Option

Ventajas:

- ✓ La opción empresarial ETL (Enterprise ETL Option) para Warehouse Builder es una opción que puede ser adquirida con Oracle Warehouse Builder como parte de la edición empresarial del motor de base de datos.
- ✓ Permite ejecutar cargas de datos usando métodos rápidos y eficientes tales como el Oracle Data Pump[4] y Transportable Tablespaces[5].
- ✓ Permite prever el efecto que puedan tener los cambios que se hagan en cualquier lugar de los metadatos del sistema ETL.
- ✓ Es posible generar un modelo para configurar los ambientes de desarrollo, pruebas y producción a niveles separados.

Desventajas:

[4] Oracle Data Pump es una alternativa nueva, rápida y flexible a las utilidades exportar e importar usadas en las versiones anteriores de Oracle (15) (16).

[5] Transportable Tablespaces (espacios de tabla transportables) se introdujeron en Oracle 8i para permitir copiar espacios de tabla enteros entre bases de datos en el tiempo que tarda en copiar los archivos de datos (17).

- ✓ Esta herramienta es fácil de usar cuando se trata de información almacenada en bases de datos Oracle, debido a las herramientas Data Pump y Transportable Tablespaces.
- ✓ No ofrece mucha compatibilidad a otras Bases de Datos (BD).
- ✓ Puede presentar alguna incompatibilidad o dificultad al tener que acceder a bases de datos como SqlServer o Postgres (13) (14).

1.3.3 IBM Cognos Data Manager

Ventajas:

- ✓ Proporciona funciones dimensionales de ETL para conseguir una inteligencia empresarial de alto rendimiento.
- ✓ Se puede integrar con la GUI de International Business Machines (IBM) Data Manager Designer para diseñar y crear prototipos.
- ✓ Se pueden ejecutar compilaciones y secuencias de trabajos en sistemas remotos desde un sistema de entorno de diseño de Data Manager.

Desventajas:

- ✓ Data Manager Engine se tiene que instalar en un sistema UNIX o Linux.
- ✓ Poca documentación, lo que afecta la línea de aprendizaje y pudiera traer como consecuencia que la migración no fuese realizada en el tiempo esperado (13) (14).

1.3.4 Pentaho Data Integration (Kettle)

Pentaho Data Integration (PDI, también llamado Kettle) es el componente de Pentaho responsable de los procesos de ETL. Aunque las herramientas ETL se utilizan con mayor frecuencia en entornos de almacenes de datos, PDI también puede utilizarse para otros fines:

- ✓ Migración de datos entre aplicaciones o bases de datos.
- ✓ Exportación de datos de bases de datos a archivos planos.
- ✓ Carga masiva de datos en bases de datos.
- ✓ Limpieza de datos.
- ✓ Integración de aplicaciones.

Las soluciones de Pentaho están escritas en Java y tienen un ambiente de implementación también basado en Java. Además de ser multiplataforma, incluye herramientas para realizar consultas, generación de informes y reportes, análisis interactivo, tableros de mando, ETL/integración de datos, data mining (minería de datos) y un servidor para la plataforma de BI (inteligencia de negocio) que lo ha convertido en la suite BI de software libre más popular del mundo. Productos de Pentaho se utilizan en el Sistema de Mando Aéreo de US Army, Lifetime Networks, Terra Industries y Sun Microsystems.

Características de Kettle:

- ✓ Plataforma: Windows, Unix y Linux.
- ✓ GUI: interfaz con indicadores visuales de transformación. Informes disponibles de la capa de metadatos.
- ✓ Código: aplicación 100% Java con transformaciones avanzadas en JavaScript mediante una interfaz empotrada. Diseño orientado a metadatos.
- ✓ Licencia: Mozilla Public License.
- ✓ Código fuente: el código fuente está disponible.
- ✓ Soporte: existe un foro, un buscador de problemas y la comunidad Pentaho, con varios artículos técnicos que son mejores que algunos de los vendedores oficiales de productos para ETL.
- ✓ Conectividad: soporta Oracle, DB2, SQL Server, PostgreSql, etc.
- ✓ Es una de las herramientas ETL libres más antiguas que tiene una gran cantidad de usuarios y una nueva dirección por parte del soporte técnico de Pentaho (13) (13).

1.4 Migración de datos

La migración de datos es el proceso mediante el cual se realiza una transferencia de datos de unos sistemas de almacenamiento a otros, de unos formatos de datos a otros o entre diferentes sistemas informáticos (18).

1.4.1 ¿Por qué la Migración de Datos?

Los datos son el bien más importante de una empresa, es por ello la necesidad de contar con un sistema más actualizado y que garantice aspectos vitales para el funcionamiento de la entidad. Hoy día los sistemas tienen que manejar grandes cantidades de información, lo que trae a su vez que estos sistemas estén periódicamente cambiando a versiones superiores y, con ellos, sus gestores de bases de datos. Por razones como las mencionadas anteriormente, empresas como la ESEN toman la decisión de migrar su información a sistemas más potentes que garanticen:

- ✓ Mayor seguridad
- ✓ Adaptabilidad a exigencias del mercado
- ✓ Integración de los datos

En el caso de la ESEN, el motivo que los lleva a tomar esta decisión es la necesidad de contar con un sistema único que sea capaz de manejar el negocio. Debido a que actualmente cuenta con diferentes

sistemas que manejan la información de manera modular y no centralizada, se pueden encontrar diferentes datos refiriéndose a lo mismo. Esto ha traído como consecuencia la falta de homologación entre los datos que se almacenan.

1.5 Etapas de la migración de datos

1- Descubrimiento y análisis de datos: consiste en evaluar y comprender los datos existentes, en particular, los que son requeridos por el nuevo sistema. Para determinar la calidad de los datos orígenes se realiza el perfilado de los datos, analizando cada uno en particular, así como sus interrelaciones.

2- Extracción: extraer los datos relevantes y depositarlos en un área intermedia donde la estructura sea similar a la estructura de origen.

3- Transformación: aplicar las reglas de transformación necesarias para garantizar la calidad de los datos a cargar y las modificaciones necesarias para el uso de los datos en el destino.

- ✓ **Limpieza de datos:** se corrigen los datos que se consideran incorrectos o inconsistentes. La entrada de este proceso es el perfilado realizado en la etapa de análisis y las reglas del negocio definidas.
- ✓ **Homologación:** implica la unificación de criterios, porque un dato en un origen puede ser correcto, pero no estar manejado de la misma forma en el destino. Requiere de la unificación de códigos, descripciones, etc.
- ✓ **Enriquecimiento:** se lleva a cabo al complementar y perfeccionar los datos maestros, si les faltase completitud. Un ejemplo práctico sería agregar los códigos postales a las direcciones en la base de datos de clientes, si se notificase dicha carencia.

4- Carga: la fase de mapeo y carga es la culminación de los procesos de integración de datos.

5- Validación del proceso de migración de datos: es en este momento cuando los resultados se hacen evidentes y se descubre si se alcanzó el nivel de calidad esperado, si se consiguió la ausencia de errores, si se cumplieron los plazos previstos para llevar a cabo el proyecto, si el nuevo sistema funciona a pleno rendimiento. (19)

1.6 ¿Cómo se lleva a cabo el análisis de datos en la migración?

El análisis de datos comienza con un reconocimiento de las reglas de negocio, no sólo por la entidad, sino también en lo relativo a su correlación, campo a campo, con el nuevo sistema. Cuando esta etapa concluye, ya se puede comenzar a perfilar los datos.

Como resultado del análisis se obtendrán las definiciones precisas para la limpieza de datos y el mapeo de campos que tendrán lugar en etapas posteriores.

1.7 Papel del perfilado de datos

Como Jack Olson explica en su libro “Data Quality: The Accuracy Dimension”: “el perfilado de datos emplea métodos analíticos para examinar los datos con el fin de desarrollar una comprensión cabal del contenido, la estructura y la calidad de los datos. Un buen sistema de perfilado de datos puede procesar grandes cantidades de datos y con las habilidades del analista, descubrir todo tipo de cuestiones que deben abordarse” (20).

Una vez que se cuenta con un perfil de la fuente, que debe incluir tipos de datos, variables, clasificación de las variables, inconsistencias en los datos, existencia de valores perdidos, es posible aplicar limpieza de datos.

Realizar el perfilado de datos permite conocer en qué estado se encuentran, pero aporta poca información de la estructura de la base de datos. Es necesario realizar un estudio de la base de datos, para conocer cuáles son los tipos de datos que tiene, si acepta valores nulos, la existencia de llaves, las relaciones con que cuenta y un gran cúmulo de información que puede ser obtenido.

1.8 Integración de datos

La integración de datos es el problema de combinar los datos que residen en diferentes fuentes, y proporcionar al usuario una visión unificada de estos datos. El problema de diseñar sistemas de integración de datos es importante en las aplicaciones actuales del mundo real y se caracteriza por una serie de cuestiones que son interesantes desde un punto de vista teórico (21).
Por lo general, un proyecto de integración de datos conlleva los siguientes pasos:

- ✓ **Acceso a los datos** de todas las fuentes y ubicaciones.
- ✓ **Integración de datos**, para que los registros de una fuente de datos se asignen a los registros de otra (por ejemplo, incluso en el caso de que un conjunto de datos utilice "lastname, firstname" y otro utilice "fname, lname", el conjunto integrado se asegurará de que ambos terminen en el lugar correcto). Este tipo de preparación de datos es esencial para que las aplicaciones analíticas y de cualquier otro tipo puedan utilizar los datos de manera satisfactoria.
- ✓ **Suministro de datos integrados** al negocio exactamente en el momento en que los necesita, en tiempo real o prácticamente en tiempo real.

1.9 Metodología

La metodología de desarrollo de software se refiere a un marco que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. Su propósito es establecer un contrato social entre todos los participantes en un proyecto para conseguir la solución más eficaz con los

recursos disponibles. Es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a producir nuevo software (22) (23).

Para el desarrollo del presente trabajo fue definida como metodología a utilizar la Metodología de desarrollo para proyectos de almacenes de datos o MDPAD, que toma como base la metodología de Kimball para definir los aspectos específicos del desarrollo de Almacenes de Datos (DW). Para incorporar los principios básicos que permiten una adecuada gestión del proyecto, utiliza la Guía para los Fundamentos de la Dirección de Proyectos. Los temas asociados a Integración de Modelos de Madurez de Capacidades (CMMI) se incorporan a partir del Programa de Mejora, por lo tanto hereda algunos de sus enfoques, artefactos y actividades (24). La misma cuenta de siete fases que se muestran a continuación:

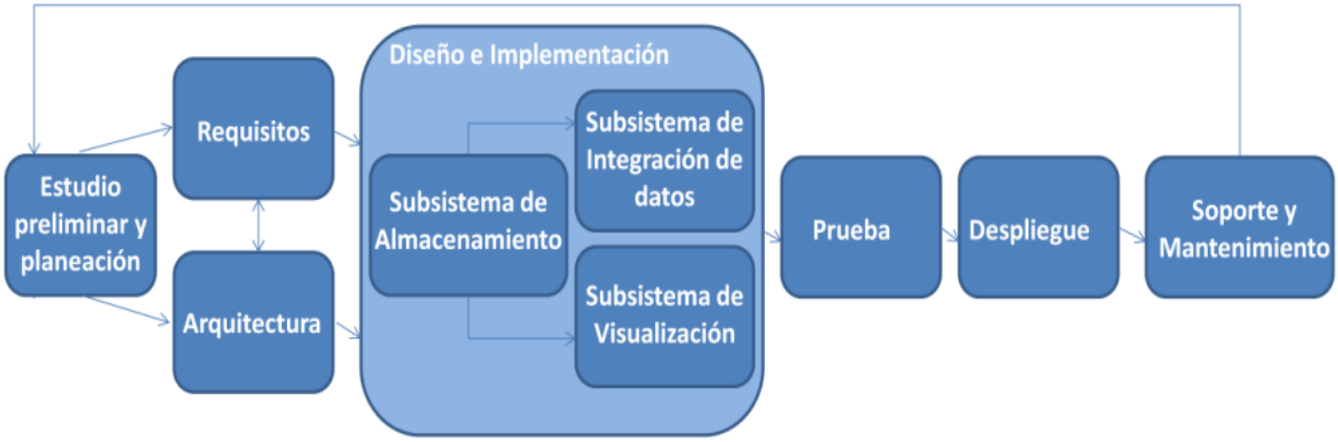


Figura 1 Fases de la MDPAD

Al ser una metodología desarrollada para almacenes de datos, solo se utilizará de ella las cinco primeras fases, que es la parte referida a procesos ETL. A pesar de no ser específica para este tipo de procesos, consta de los artefactos necesarios para desarrollar de manera exitosa la migración de datos.

1.10 Herramientas a utilizar

1.10.1 ER/Studio Embarcadero

Es una herramienta líder en el modelado de datos, ayuda a descubrir, documentar y reutilizar los activos en datos. Con soporte de ida y vuelta para bases de datos, los arquitectos tienen la potencia de hacer ingeniería inversa fácilmente y optimizar las bases de datos existentes. Las fuertes capacidades en colaboración de ER/Studio pueden conseguir mayor productividad y forzar el cumplimiento de los estándares de la organización (25).

Funcionalidades y beneficios:

- ✓ **Soporte completo al ciclo de vida de Bases de Datos.**
  Ingeniería inversa y directa.
  Generación automatizada de código de bases de datos.
- ✓ **Almacén de Datos y Soporte a la Integración.**
  Documentación Visual del Linaje de Datos.
  Modelado Dimensional de los modelos lógico y físico.
- ✓ **Diseños de calidad de Bases de Datos.**
  Validación de diseño y creación de la integridad referencial.
  Capacidad de planificación y modelado de seguridad.

1.10.2 PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia Berkeley Software Distribution (BSD) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (26).

1.10.3 PgAdmin III

PgAdmin III es una aplicación gráfica con licencia de código abierto para administrar bases de datos PostgreSQL. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir del PostgreSQL 7.3 ejecutándose en varias plataformas, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres (27).

La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad (27).

1.10.4 SQLServer

Microsoft SQL Server es una plataforma de base de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de

comercio electrónico; es también una plataforma de Business Intelligence para soluciones de integración, análisis y creación de informes de datos (28).

Principales características:

- ✓ Facilidad de instalación, distribución y utilización, porque incluye un conjunto de herramientas administrativas y de desarrollo que mejora el proceso de instalación, distribución, administración y uso de SQL Server en varios sitios.
- ✓ Características de bases de datos corporativas, debido a que protege la integridad de los datos a la vez que minimiza la carga de trabajo que supone la administración de miles de usuarios modificando la base de datos simultáneamente.
- ✓ Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor, y los terminales o clientes de la red solo acceden a la información.

1.10.5 Data Cleaner

El perfilado es una actividad esencial de cualquier programa de calidad de datos, gestión de datos maestros o gobernanza de datos. Una de las principales herramientas para el análisis de calidad de datos y perfilado es Data Cleaner, aplicación de código abierto para el perfilado, la validación y comparación de datos.

Según su creador Kasper Sorense, el sistema requiere Java Runtime Enviroment 5.0 o una versión superior y drivers de JDBC[6]. La misma permite la evaluación del nivel de calidad de los datos contenidos en el sistema de información. Es una aplicación fácil de usar que genera sofisticados informes y gráficos que permiten a los usuarios determinar el nivel de calidad de los datos. Es utilizada, además, para identificar y analizar la estructura del origen de datos y combinar resultados y gráficos, creando vistas fáciles de interpretar para evaluar la calidad de los mismos (30).

1.11 Propuesta de estrategia

Actualmente la mayoría de las migraciones fracasan por la falta de una estrategia de integración que guíe el proceso. Siguiendo la metodología MDPAD y ajustándola al problema, se proponer como parte

[6] Java Database Connectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice (29).

de la propuesta de solución, la implementación de una estrategia de carga inicial cuyos pasos se muestran a continuación:

1. Estudio de la base de datos.
2. Diseño de la arquitectura funcional.
3. Mapeo de los datos.
4. Diseño de la estructura del expediente de proyecto.
5. Estandarización de los nombres.
6. Diseño e implementación de las transformaciones.
7. Diseño e implementación de los trabajos.
8. Diseño e implementación del trabajo principal.
9. Ejecución del trabajo principal.
10. Pruebas de integración.

Una estrategia es la secuencia de pasos lógicos a seguir para lograr un objetivo específico (31–33). Posteriormente se explica que se hace durante cada paso de la estrategia:

Estudio de la base de datos: en este paso se realiza un análisis de la estructura de la base de datos fuente para conocer sus principales características.

Diseño de la arquitectura funcional: durante este paso se realiza el diseño de la arquitectura, es donde se define el modo en que van a estar relacionados los sistemas fuente, el destino y el subsistema de integración.

Mapeo de datos: en este paso se define el destino que va a tener cada atributo de las columnas que conforman las tablas de la base de dato.

Diseño de la estructura del expediente de proyecto: se crea la estructura del expediente de proyecto, la cual contiene las carpetas con las consultas SQL, transformaciones, trabajos, base dato fuente y destino, etc. Posibilita constar con una estructura organizada.

Estandarización de los nombres: Se crea un estándar de codificación, para lograr un mayor entendimiento del trabajo, lo que puede ser de gran ayuda a la hora de brindar soporte en caso de ser necesario.

Diseño e implementación de las transformaciones: En este paso se realiza el diseño e implementación de cada una de las transformaciones con que va a contar el trabajo.

Diseño e implementación de los trabajos: Durante este paso se diseñan e implementan los trabajos necesarios para guiar la ejecución de los trabajos y transformaciones.

Diseño e implementación del trabajo principal: En este paso se diseña e implementa el trabajo principal que será el encargado de guiar la ejecución de las transformaciones y trabajos en el orden correcto.

Ejecución del trabajo principal: Se ejecuta el trabajo principal, encargado de ejecutar cada una de las transformaciones y trabajos presentes en su diseño.

Pruebas de integración: Se realizan las pruebas necesarias para comprobar que se hallan cargado correctamente los datos en la base de dato destino, comprobando a su vez la integración de los datos.

1.12 Conclusiones parciales

En este capítulo se trataron los principales conceptos para lograr la comprensión del trabajo, se presentaron las principales herramientas ETL existentes, posibilitando conocer sus principales ventajas y desventajas para su selección en el capítulo posterior. Para guiar el proceso de desarrollo se definió como metodología a utilizar la Metodología de desarrollo para proyectos de almacenes de datos, desarrollada en el centro DATEC. Además, se elaboró la propuesta de estrategia a utilizar como guía de integración de datos.

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

2.1 Introducción

En este capítulo se realiza un estudio del negocio con el objetivo de definir las reglas del mismo. Se analizan los sistemas involucrados en la migración de datos, así como las herramientas utilizadas y los artefactos generados por la metodología. Todo esto se encuentra detallado en la propuesta de una estrategia de carga inicial, que ajusta la metodología a las particularidades del problema. Además, se selecciona como parte especial de este capítulo la herramienta ETL a utilizar.

2.2 Descripción de la solución propuesta

Se desea realizar la carga inicial de datos legados del sistema SIGES hacia el SISEN a través de la implementación de una estrategia de carga inicial, que permitirá cargar los datos. Garantizando su integración, realizando las limpiezas y transformaciones de datos necesarias, dado que las bases de datos fuente y destino poseen estructuras diferentes. Para el diseño se utiliza la MDPAD y a lo largo del capítulo se abordan los pasos del 1 al 5 de la estrategia propuesta, los pasos del 6 al 10 se realizan en el capítulo posterior.

2.3 Aplicación de la Metodología de desarrollo para almacenes de datos

La MDPAD, está basada en la metodología creada por Kimball, fue desarrollada en la Universidad de las Ciencias Informáticas (UCI) por el Centro de Tecnologías de Gestión de Datos (DATEC). Es una metodología para llevar a cabo el diseño de todas las fases de los almacenes de datos, incluyendo las fuentes de datos operacionales, los procesos ETL y el propio esquema del almacén de datos. Es una metodología muy útil para el diseño de las transformaciones de datos que se desea realizar, porque describe de forma bien específica todo el proceso necesario para alcanzar el objetivo propuesto. Propone la realización del perfilado de datos, que describe el estado de los datos almacenados en los sistemas fuente, además de los mapeos de datos. En este trabajo solo se utiliza de la metodología las cinco primeras fases de la misma, debido a que fue desarrollada para almacenes de datos, tomando de ella todos los artefactos necesarios para realizar la carga inicial, teniendo como elementos principales los referentes a ETL. Esta metodología se aplica a través de la estrategia de carga inicial que fue propuesta como solución en el capítulo anterior.

2.4 Análisis de las bases de datos

Con el objetivo de conocer a grandes rasgos los aspectos principales, puntos en común y principales diferencias, de las bases de datos que intervienen en la migración, se realiza el paso número uno de la estrategia de carga inicial, que consiste en un estudio sobre las estructuras de las dos bases de datos. Luego de realizado el estudio se obtuvo la siguiente información:

Base	Tablas	Atributos	Relaciones	Sistema
sis_hol	248	2541	0	SIGES
prod	408	2255	679	SISEN

Tabla 1 Comparación entre las bases de datos.

En el caso de la base de datos del SISEN “prod” tiene un mayor número de tablas, debido a que esta se encuentra normalizada y cuenta con mayor nivel de detalle que la base de datos del SIGES “sis_hol”. En cuanto a la cantidad de atributos es bastante similar porque ambas manejan la misma información pero de formas diferentes.

Entre los principales tipos de datos encontrados en ambas BD se puede encontrar:

- varchar (sis_hol y prod)
- integer (sis_hol y prod)
- char (sis_hol y prod)
- datatime (sis_hol y prod)
- decimal (sis_hol y prod)
- boolean (sis_hol y prod)
- nvarchar (sis_hol)
- nchar (sis_hol)

Aunque en ambas se manejan tipos de datos similares, no siempre coinciden los atributos de una base de datos con la otra en cuanto a los tipos de datos y longitud de los mismos.

Se realiza un muestreo sobre las tablas **poliza, agente** y **cliente,** debido a que tienen la mayor cantidad de atributos en la BD **sis_hol** perteneciente al SIGES**.** Se obtuvieron los valores máximos y mínimos que permiten insertar la BD.

Tipo de dato	Valor mínimo	Valor máximo
varchar	12 caracteres	12 caracteres
char	2 caracteres	12 caracteres
integer	-10^9	10^9
decimal	(4,2)	(13,2)
nchar	10 caracteres	10 caracteres

nvarchar	1 caracteres	165 caracteres

Tabla 2 Longitud de los principales tipos de datos

Las siguientes figuras muestran los principales tipos de datos, así como su longitud y si permiten valores nulos o no.

poliza

idpoliza	CHAR(12)	NOT NULL
cliente	CHAR(11)	NOT NULL
agente	INTEGER	NULL
finicio	CHAR(10)	NULL
fdesde	CHAR(10)	NULL
fhasta	CHAR(10)	NULL
tipopoliza	INTEGER	NULL
formapago	INTEGER	NULL
femicion	SMALLDATETIME	NULL
moneda	INTEGER	NULL
poliza	CHAR(12)	NOT NULL
modalida	INTEGER	NULL
cant_tot	DECIMAL(9,4)	NULL
cant_cap	DECIMAL(8,0)	NULL
valoraseg	DECIMAL(13,2)	NULL
primas	DECIMAL(9,2)	NULL
pago	CHAR(2)	NULL
usuario	CHAR(40)	NULL
tipo_doc	INTEGER	NULL
endoso	CHAR(10)	NULL
fecha_end	CHAR(10)	NULL
prima1	DECIMAL(9,2)	NULL
p_bon_rec	DECIMAL(4,2)	NULL
p_desc	DECIMAL(4,2)	NULL
producto	INTEGER	NULL
f_creado	CHAR(10)	NULL
id_tabla	INTEGER	IDENTITY

Figura 2 Diseño de la tabla “poliza” de la BD sis_hol

Agentes

id_tabla	INTEGER	IDENTITY
NOLICENCIA	INTEGER	NULL
SITUACION	INTEGER	NULL
CARNETID	NVARCHAR(15)	NULL
NOMBRES	NVARCHAR(150)	NULL
DIRPART	NVARCHAR(165)	NULL
DPA	NVARCHAR(4)	NULL
PERSONA	NVARCHAR(1)	NULL
idterrit	INTEGER	NULL
idcargo	INTEGER	NULL
telefono	CHAR(10)	NULL
PERSONAS	INTEGER	NULL
RAMOS	CHAR(69)	NULL
F_creado	CHAR(10)	NULL
F_update	CHAR(10)	NULL
F_VENC	CHAR(10)	NULL
usuario	NCHAR(10)	NULL
ccsf	BIT	NULL

Figura 3 Diseño de la tabla "Agentes" de la BD sis_hol

cliente

id_tabla	INTEGER	IDENTITY
idcliente	CHAR(11)	NOT NULL
direccion	CHAR(130)	NULL
dpa	CHAR(4)	NULL
sector	CHAR(2)	NULL
orga	CHAR(3)	NULL
cod	CHAR(5)	NULL
nombrecomp	CHAR(60)	NULL
n_cliente	CHAR(11)	NOT NULL
fax	CHAR(25)	NULL
email	CHAR(30)	NULL
telefono	CHAR(25)	NULL
cae	CHAR(6)	NULL
nae	CHAR(10)	NULL
subord	CHAR(10)	NULL
cult	CHAR(10)	NULL
usuario	CHAR(10)	NULL
f_creado	CHAR(10)	NULL
comercial	CHAR(3)	NULL
cuent_banc	CHAR(30)	NULL

Figura 4 Diseño de la tabla "cliente" de la BD sis_hol

2.5 Reglas del negocio

Una regla de negocio es una condición que se debe satisfacer cuando se realiza una actividad de negocio. Una regla puede imponer una política de negocio, tomar una decisión o inferir nuevos datos de datos existentes (34).

Las reglas del negocio se clasifican en varias categorías, éstas son:

- ✓ Reglas de variables: son las reglas que de definen las variables calculables que son objeto de análisis.
- ✓ Reglas de almacenamiento: son las reglas que definen características específicas del almacenamiento de alguna variable.
- ✓ Reglas de transformación: son las reglas que implican la transformación de alguna variable durante los procesos de integración de datos.

La fase de transformación de un proceso ETL aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas fuentes de datos requieren alguna pequeña manipulación de los datos. En el siguiente trabajo fueron detectadas diferentes reglas de negocio en varias tablas, las cuales propician tomar decisiones en esta fase. Fueron encontradas 0 reglas de variable, 10 reglas de almacenamiento y 34 de transformación, constituyendo estas reglas el punto de partida para realizar las transformaciones sobre los datos.

2.6 Mapa lógico

Representa el origen y destino físico de cada uno de los atributos que conforman las tablas según la estructura física de las fuentes de datos y el modelo físico de la base de datos.

Las siguientes tablas son ejemplos del mapeo realizado sobre las columnas de las tablas, en el caso de las filas que aparecen en blanco es porque ese atributo no está presente en la BD fuente pero se necesita en la destino. La mayoría de estos datos fueron suministrados por el personal del proyecto en correspondencia con las características del negocio. Se muestran las principales reglas del negocio con las transformaciones que son realizadas sobre los datos:

Destino				**Fuente**			
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
base_estructura	id_estructura	varchar	nomencladores	dbo	provincia	idprov	varchar
base_estructura	nombre	varchar	nomencladores	dbo	provincia	nombre	char
base_estructura	abreviatura	varchar					
base_estructura	nivel_jerarquico	integer					
base_estructura	correo	varchar					
base_estructura	telefono	varchar					
base_estructura	id_provincia	integer	nomencladores	dbo	provincia	idprov	char

Tabla 3 Mapeo de la tabla base_estructua

La tabla **provincia** se transforma en la tabla **base_estructura**. Al realizar las transformaciones, los atributos quedan de la siguiente forma:

- ✓ **id_estructura:** se forma agregándole un ‘0’ al comienzo de idprov utilizando la funcionalidad de Kettle “Valor Java Script Modificado”.

- ✓ **nombre:** se eliminan los espacios que contiene al final de la fila usando la función “String Operations”.
- ✓ **abreviatura**: es generada a través de un mapeo de datos, dado el identificador de cada provincia idprov se genera una abreviatura empleando la funcionalidad “Mapeo de Valores”.
- ✓ **nivel_jerarquico**: se agrega ‘0’ como un valor constante utilizando “Valor Java Script Modificado”.
- ✓ **id_provincia**: se renombra el atributo idprov con la funcionalidad “Seleccionar Renombrar Valores”.

Los atributos **correo** y **telefono** no pueden ser cargados en la tabla base_estructura, porque no están presentes en la tabla **provincia**, además de ser atributos opcionales que no influyen en el correcto funcionamiento del sistema**.**

Destino			**Fuente**				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_natural	id_persona	varchar	prod	mod_base	pers_persona	id_persona	varchar
pers_natural	ni_pasaporte	varchar	sis_hol	dbo	Agentes, i_receptor	CARNETID, id_recep	varchar
pers_natural	id_sexo	integer					
pers_natural	edad	integer					
pers_natural	primer_apellido	varchar	sis_hol	dbo	Agentes, i_receptor	NOMBRES, nombre	varchar
pers_natural	segundo_apellido	varchar	sis_hol	dbo	Agentes, i_receptor	NOMBRES, nombre	varchar

Tabla 4 Mapeo de la tabla pers_natural

Las tablas **Agentes** y **i_receptor** se transforman en la tabla **pers_natural**. Al realizar las transformaciones, los atributos quedan de la siguiente forma:

- ✓ **id_persona:** se genera a partir del código de la estructura más un número que es obtenido de la tabla base_variable hasta completar diez dígitos después del código de la estructura y el sufijo de la persona. Por ejemplo 032 es el código de la estructura, el valor obtenido de la tabla

base_variable es 20 y el sufijo es AG, quedaría de la siguiente forma **'0320000000020AG'**. Los valores restantes que faltan para completar los diez dígitos se completan con ceros a la izquierda. El identificador es creado utilizando las funcionalidades "Añadir Secuencia" y "Valor Java Script Modificado".

- ✓ **ni_pasaporte:** se renombran los atributos CARNETID e id_recep empleando la funcionalidad "Seleccionar Renombrar Valores".
- ✓ **id_sexo:** se asigna el identificador 3, porque no es posible saber cuál es el sexo de cada persona con los datos que se tienen. Se utilizó la funcionalidad "Valor Java Script Modificado".
- ✓ **edad:** se calcula a partir del CARNETID utilizando la funcionalidad "Valor Java Script Modificado".
- ✓ **primer_apellido:** se obtiene de los atributos NOMBRES y nombre utilizando la funcionalidad "Valor Java Script Modificado".
- ✓ **segundo_apellido:** se obtiene de los atributos NOMBRES y nombre utilizando la funcionalidad "Valor Java Script Modificado".

Destino			**Fuente**				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_representante	id_representante	varchar	sis_hol	dbo	i_receptor	id_recep	varchar
pers_representante	id_persona	varchar	prod	mod_base	pers_persona	id_persona	varchar
pers_representante	activo	boolean	sis_hol	dbo	i_receptor	estado	boolean
pers_representante	f_inicio	date					
pers_representante	f_fin	date					
pers_representante	cod_historico	bigint					
pers_representante	id_territorio	integer	sis_hol	mod_nomencladores	nom_provincia	id_provincia	integer

Tabla 5 Mapeo de la tabla pers_representante

La tabla **i_receptor** se transforma en la tabla **pers_representante**. Al realizar las transformaciones, los atributos quedan de la siguiente forma:

- ✓ **id_representante:** se obtiene del atributo id_recep. Se eliminan los espacios en blanco de ambos lados utilizando la funcionalidad "Operaciones String" y se renombra a id_representante usando "Seleccionar Renombrar Valores".
- ✓ **id_persona:** se genera a partir del código de la estructura más un número que es obtenido de la tabla base_variable hasta completar diez dígitos después del código de la estructura y el sufijo de la persona. Por ejemplo 032 es el código de la estructura, el valor obtenido de la tabla base_variable es 20 y el sufijo es AG, quedaría de la siguiente forma **'0320000000020RTV'**. Los valores restantes que faltan para completar los diez dígitos se completan con ceros a la izquierda. El identificador es creado utilizando las funcionalidades "Añadir Secuencia" y "Valor Java Script Modificado".
- ✓ **activo:** se obtiene del atributo estado, el cual es renombrado a activo usando la funcionalidad "Valor Java Script Modificado".
- ✓ **cod_historico:** es generado a partir de una secuencia utilizando la funcionalidad "Añadir Secuencia".
- ✓ **id_territorio:** es obtenido mediante una búsqueda en base de datos a partir del atributo id_provincia usando la funcionalidad "Búsqueda en Base de Datos".

Destino			**Fuente**				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_comercial	id_persona	varchar	prod	mod_base	pers_persona	id_persona	varchar
pers_comercial	f_inicio	date	sis_hol	dbo	Agentes	F_creado	date
pers_comercial	f_fin	date	sis_hol	dbo	Agentes	F_VENC	date
pers_comercial	id_comercial	varchar	sis_hol	dbo	Agentes	NOLICENCIA	integer
pers_comercial	cod_historico	bigint					

Tabla 6 Mapeo de la tabla pers_comercial

La tabla **Agentes** se transforma en la tabla **pers_comercial**. Al realizar las transformaciones, los atributos quedan de la siguiente forma:

- ✓ **id_persona:** se genera a partir del código de la estructura más un número que es obtenido de la tabla base_variable hasta completar diez dígitos después del código de la estructura y el sufijo de la persona. Por ejemplo 032 es el código de la estructura, el valor obtenido de la tabla base_variable es 22 y el sufijo es AG, quedaría de la siguiente forma **'0320000000022AG'**. Los valores restantes que faltan para completar los diez dígitos se completan con ceros a la izquierda. El identificador es creado utilizando las funcionalidades "Añadir Secuencia" y "Valor Java Script Modificado".
- ✓ **f_inicio:** se obtiene del atributo F_creado después de renombrar el campo usando la funcionalidad "Seleccionar Renombrar Valores".
- ✓ **f_fin:** se obtiene del atributo F_VENC después de renombrar el campo, utilizando la funcionalidad "Seleccionar Renombrar Valores".
- ✓ **id_comercial:** es obtenido del atributo NOLICENCIA, se le cambia el tipo de dato integer a varchar y se renombra a id_comercial, todas las operaciones se realizan con la funcionalidad "Seleccionar Renombrar Valores".
- ✓ **cod_historico:** es generado a partir de una secuencia que empieza en uno con incremento uno, usando la función "Añadir Secuencia".

Destino			**Fuente**				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_ agente	nro_inscripcion	varchar	sis_hol	dbo	Agentes	NOLICENCIA	varchar
pers_ agente	activo	boolean	sis_hol	dbo	Agentes	SITUACION	boolean
pers_ agente	id_estructura	varchar	sis_hol	mob_base	base_ estructura	id_estructura	integer
pers_ agente	id_territorio	integer	sis_hol	mod_ configuracion	conf_territorio	id_territorio	integer
pers_ agente	id_agente	varchar	sis_hol	dbo	Agentes	CARNETID	varchar
pers_ agente	id_persona	varchar	sis_hol	mod_base	pers_persona	id_persona	varchar

pers_ agente	comercial	boolean	sis_hol	mod_base	pers_comercial	id_comercial	string

Tabla 7 Mapeo de la tabla pers_agente

La tabla **Agentes** se transforma en la tabla **pers_agente**. Al realizar las transformaciones, los atributos quedan de la siguiente forma:

- ✓ **nro_inscripcion:** se obtiene del campo NOLICENCIA después de ser renombrado a nro_inscripcion usando la funcionalidad “Seleccionar Renombrar Valores”.
- ✓ **activo:** se obtiene del campo SITUACION después de ser renombrado en activo, mediante la funcionalidad “Seleccionar Renombrar Valores”.
- ✓ **id_estructura:** es obtenido de una búsqueda en base de datos usando la funcionalidad “Búsqueda en Base de Datos” en la tabla base_estructura del esquema mod_base, se le cambia el tipo de dato de integer a varchar utilizando “Seleccionar Renombrar Valores”.
- ✓ **id_territorio:** este identificador se obtiene de una búsqueda en base de datos utilizando usando la funcionalidad “Búsqueda en Base de Datos” en la tabla conf_territorio del esquema mod_configuracion. Manteniendo el mismo tipo de dato y nombre.
- ✓ **id_agente:** es obtenido a partir del campo CARNETID después de ser renombrado a id_agente usando la funcionalidad “Seleccionar Renombrar Valores”.
- ✓ **id_persona:** se obtiene de una búsqueda en base de datos con la funcionalidad “Búsqueda en Base de Datos” en la tabla pers_persona del esquema mod_base.
- ✓ **comercial:** se realiza una búsqueda en la base de datos usando la funcionalidad “Búsqueda en Base de Datos” en el esquema mod_base en la tabla pers_comercial para obtener los identificadores de cada persona comercial que exista en la tabla y en caso de que no exista se devuelve **null** como valor por defecto. Después de tener los identificadores y los valores **null** que son obtenidos con tipo de dato string y convertidos a booleanos utilizando la funcionalidad, cambiando los identificadores que aparecieron por **true** y los valores **null** por false utilizando la funcionalidad “Valor Java Script Modificado”.

2.7 Selección de la herramienta para procesos ETL

Para realizar la selección de la herramienta fue necesario apoyarse en la información obtenida sobre cada una de ellas en el capítulo anterior y en la siguiente tabla de comparación:

Parámetros	Talend	Kettle	Oracle warehouse	Data Manager
Conectividad				
Costo				
Facilidad				

Tabla 8 Comparación de herramientas ETL

 Buena

 Regular

 Mala (14)

Las herramientas Oracle Warehouse y Data Manager quedan descartadas por ser privativas. En la mayoría de los casos, las herramientas ETL transfieren datos de sistemas heredados, por lo que su conectividad es muy importante (14).

Talend puede conectarse a todas las bases de datos actuales, archivos planos, xml, excel y servicios web, pero depende de los controladores Java para conectarse a esos datos fuentes. A diferencia de Kettle que se puede conectar a una gran variedad de bases de datos, archivos planos, xml, excel y servicios web sin necesidad de los controladores de Java, debido a que está desarrollado 100% en Java.

Después de realizar un análisis de las principales herramientas más utilizadas actualmente en el mercado, se comprobó que existen diversas herramientas de gran calidad que pueden ser utilizadas. En el caso del presente trabajo se escogió la herramienta Pentaho Data Integration (Kettle ETL)[7], porque permite dar solución a los problemas encontrados durante el mapeo de datos de manera eficiente. Además de ser estable, multiplataforma, cuenta con suficiente documentación, está desarrollada en java lo que la hace ser una herramienta potente y que cuenta con gran variedad de funcionalidades, permite la conexión a cualquier base de datos. Además, posee una interfaz amigable, su curva de aprendizaje no es compleja y permite desarrollar todas las transformaciones necesarias para dar solución al problema planteado.

[7] **Pentaho Data Integration** (PDI, también llamado Kettle) es el componente de Pentaho responsable de los procesos de extracción, transformación y carga (ETL).

2.8 Arquitectura

Al hablar de arquitectura en el ámbito informático se hace alusión a la especificación de la estructura de un sistema computacional, entendida como la organización de los componentes y relaciones entre ellos. Se deben tener en cuenta, además, los requisitos del sistema y las restricciones a las que está sujeto, así como las propiedades no funcionales del mismo y su impacto sobre la calidad. Las reglas y decisiones de diseño que gobiernan esta estructura y los argumentos que justifican las decisiones tomadas constituyen otros aspectos a tener en cuenta.(35)

Al abordar la arquitectura de una base de datos se debe tener en cuenta la forma de representar el origen y estructura global de los datos, la comunicación, los procesos y la presentación al usuario final. La arquitectura lógica que se definió para dar solución al paso número dos de la propuesta de estrategia de carga inicial consta de tres niveles:

1. Fuentes de datos: se refiere al origen de los datos.
2. Subsistema de integración: incluye los procesos que permiten que los datos sean extraídos de las fuentes, transformados e integrados en la fuente destino.
3. Subsistema de almacenamiento: base de datos que contiene las tablas cargadas a través de los procesos de ETL.

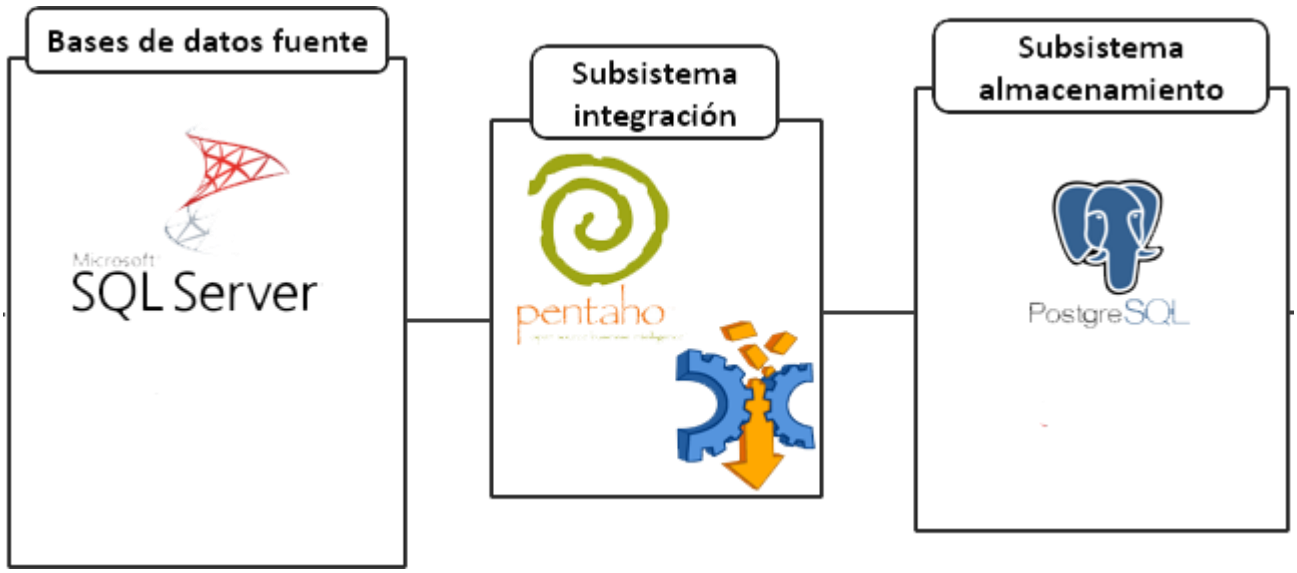


Figura 5 Diseño de la Arquitectura

2.9 Estructura del expediente de proyecto

Se define la siguiente estructura de Expediente de Proyectos de Migración de Bases de Datos basado en la metodología descrita anteriormente.

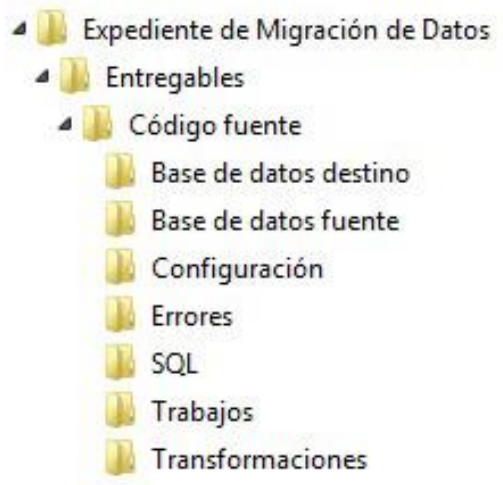


Figura 6 Diseño del expediente del proyecto

Se definió una estructura de repositorio para el código fuente de los procesos de integración de datos que incluye los siguientes elementos:

- **Bases de datos destino:** almacena las bases de datos destino en cada uno de los formatos, teniendo en cuenta el control de versiones.
- **Bases de datos fuente:** almacena las bases de datos fuentes en cada uno de los formatos, teniendo en cuenta el control de versiones.
- **Configuración:** en caso de que sea necesario, se almacenarán los *.xml* para la establecer la conexión a las bases de datos.
- **Errores:** almacena los metadatos de tratamiento de errores y los archivos que contienen los datos exportados para el posterior tratamiento de errores a partir de decisiones del equipo de desarrollo y los clientes.
- **SQL:** contiene los archivos auxiliares en lenguajes de consulta utilizados en los procesos de integración de datos.
- **Trabajos:** almacena los archivos que orquestan la ejecución de las transformaciones.
- **Transformaciones:** almacena los archivos de las transformaciones que implementan los procesos de extracción, transformación, limpieza y carga de los datos.

2.9.1 Estandarización de los nombres

La definición de la nomenclatura para el subsistema de integración de datos parte de la estructura de repositorio definida con anterioridad:

- **Bases de datos destino:** contiene los archivos de las bases de datos destino con la denominación: destino_nombre. En caso de utilizarse el control de versiones para los archivos la denominación utilizada será: destino_nombre_fecha.
- **Bases de datos fuente:** contiene los archivos de las bases de datos fuente con la denominación: fuente_nombre. En caso de utilizarse el control de versiones para los archivos la denominación utilizada será: fuente_nombre_fecha.
- **Configuración:** en caso de que sea necesario, se almacenarán los *.xml* para la establecer la conexión a las bases de datos.
- **Errores:** los metadatos de tratamiento de errores se denominarán error_concepto.
- **SQL:** para los archivos auxiliares en lenguajes de consulta utilizados se utilizará la denominación: sql_nombre.
- **Trabajos:** para los archivos de los trabajos se utilizará la nomenclatura: #_trab_nombre. En los casos en que se migre desde varias fuentes de datos se utilizará la nomenclatura: #_trab_fuente_nombre.
- **Transformaciones:** para los archivos de las transformaciones se utilizará la nomenclatura: #_trans_concepto. En los casos en que se migre desde varias fuentes de datos se utilizará la nomenclatura: #_trans_fuente_concepto.

Se enumeraron las transformaciones y los trabajos para ordenar la ejecución de los mismos en correspondencia con la integración de las tablas de las bases de datos destino.

2.9.2 Nomenclatura a utilizar en los nombres de los componentes

Componente	Nomenclatura
Entradas de datos	extraer_fuente_concepto
Selecciona/Renombra (Seleccionar campos)	seleccionar_campos
Selecciona/Renombra (Eliminar campos)	eliminar_campos
Selecciona/Renombra (Tipos de datos)	tipo_de_datos
Añadir secuencia	secuencia_nombre
Filtrar filas	filtrar_descripción
Búsqueda en base de datos	fuente_descripción

Salida de datos	cargar_destino_concepto

Tabla 9 Nomenclatura a utilizar

Se utilizó una nomenclatura similar para el resto de los componentes. Se tuvo en cuenta que los nombres fueran sugerentes en correspondencia con su uso.

Para lograr una nomenclatura estándar de las estructuras de almacenamiento del área temporal se definieron las reglas siguientes:

Nombre del área temporal de almacenamiento: temp_nombre.

Nombre de los esquemas: fuente_nombre. Para los esquemas de gestión de metadatos de utilizará la nomenclatura: metadatos_nombre.

Nombre de las tablas: concepto_nombre.

2.10 Conclusiones parciales

En el capítulo se diseñó una estrategia de carga inicial basada en la metodología propuesta, a partir de la cual se efectuó un estudio de las bases de datos donde se detectaron las principales diferencias existentes entre la fuente y el destino. Fueron determinadas las reglas del negocio y se realizó el mapeo de datos correspondiente a las tablas que serán migradas durante la ejecución del trabajo. Además, se detectaron las principales transformaciones que son necesarias para cargar los datos en el sistema destino. Se definió que la herramienta a utilizar sea Kettle porque permite resolver el problema actual, es una herramienta libre y desarrollada en java. Además, se diseñó la estructura del expediente de proyecto, brindando organización y comodidad a la hora de efectuar el trabajo. Se definió la nomenclatura a utilizar en los nombres de los componentes para lograr un mayor entendimiento del proceso realizado.

CAPÍTULO 3. Implementación y validación

3.1 Introducción

En este capítulo se hace referencia a la implementación de la estrategia de carga inicial, en él se implementaran los pasos del 6 al 10 de la solución propuesta. Se hará uso de la herramienta Kettle ETL para dar solución al problema planteado. Además, se realizan las pruebas de integración de los datos, que incluyen la validación de los resultados de la migración.

3.2 Funciones utilizadas de la herramienta

La herramienta Kettle ETL posee varias funcionalidades que permiten realizar transformaciones sobres los datos. A continuación se muestran algunas de las que fueron utilizadas en el desarrollo de la solución.

- ✓ **Función Entrada Tabla:**

Esta función se utiliza para extraer los datos de la base de datos origen.

- ✓ **Función Añadir Secuencia:**

Se utiliza para generar secuencias de números a partir de un valor dado con el incremento que se desee.

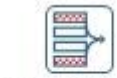
- ✓ **Función Búsqueda en Base de Datos:**

Es utilizada para mediante comparaciones entre atributos obtener campos de una tabla en específico.

- ✓ **Función Consulta base de datos:**

Permite realizar consultas SQL sobre tablas de una base de dato.

- ✓ **Función Filas Únicas:**



Se utiliza para eliminar los valores duplicados que pueda tener un campo determinado.

- ✓ **Función Filtrar Filas:**

Es utilizado para tomar diferentes acciones dependiendo de si se cumplen o no los parámetros de comparación.

- ✓ **Función Insertar Actualizar:**

Permite insertar los datos en una tabla destino y en caso de que los valores ya existan, se actualizan si existe alguna modificación.

- ✓ **Función Mapeo de Valores:**

Esta función permite modificar el valor de los datos que vienen en el flujo.

- ✓ **Función Ordenar Filas:**

Se utiliza para ordenar los datos de un campo en específico.

- ✓ **Función Salida Fichero de Texto:**

Es utilizado para generar ficheros de salida con los campos que son especificados.

- ✓ **Función Salida Tabla:**

Se utiliza para insertar los datos en la tabla destino.

- ✓ **Función Seleccionar Renombrar Valores:**

Permite renombrar el nombre de los campos que se encuentran en el flujo de datos, eliminarlos, además de poder especificar el tipo de dato y la longitud que pueden llegar a tener.

- ✓ **Función Operaciones sobre String:**

Esta función permite realizar varias operaciones como eliminar espacios en blanco, caracteres especiales, especificar que el campo solo puede contener dígitos, etc.

- ✓ **Función Valor Java Script Modificado:**

Se utiliza para crear o trabajar con funciones en código JavaScript.

3.3 Desarrollo

Se realiza la carga inicial a través de una migración para dar solución al objetivo propuesto, siguiendo los pasos propuestos en la estrategia:

6 **Implementación de las transformaciones:** se realizaron cada una de las siguientes transformaciones en correspondencia con las especificaciones del negocio y las particularidades de la BD destino.

7 **Implementación de los Trabajos:** Se construyó un esquema que contiene todas las transformaciones de la fase anterior, en el que se define la secuencia de ejecución de cada una de

ellas. También se pueden encontrar uno o varios trabajos dentro del otro, como se muestra en la Figura 15 y 16.

A continuación se muestran los procedimientos correspondientes para realizar la migración de los datos contenidos en la tabla base_estructura descrita en el capítulo anterior:

- ✓ Estas transformaciones comienzan creando la conexión a las bases de datos fuente y destino. Ambas conexiones fueron realizadas de manera similar, cambiando los campos específicos de cada una. A continuación se puede observar en la Figura 7 cómo se creó la conexión a la fuente.

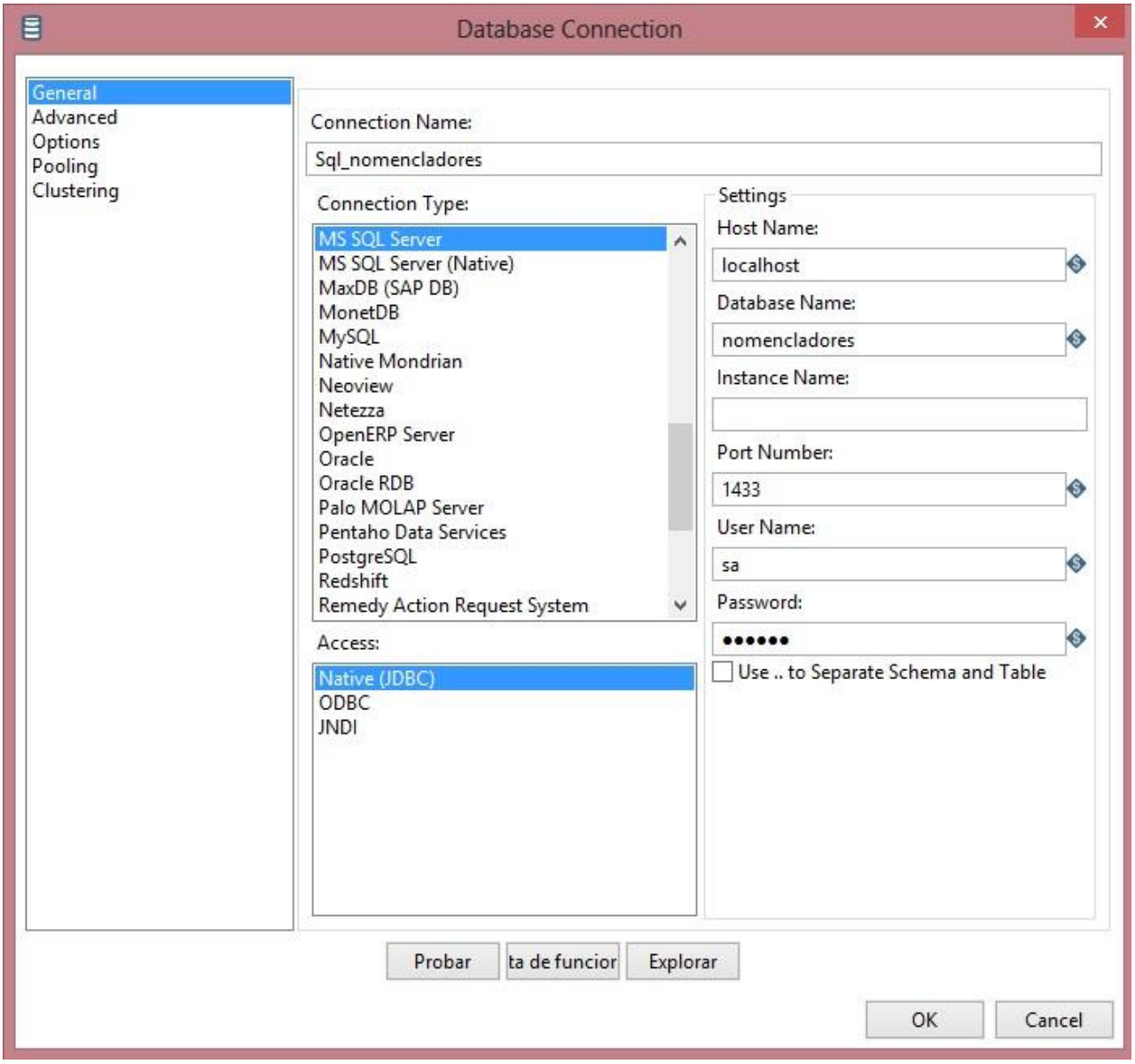


Figura 7 Conexión a la base de datos fuente

- ✓ Posteriormente se especifica la entrada y salida, utilizando las funciones “Entrada Tabla” y “Insertar/Actualizar”, para determinar la tabla que se utilizará y los campos a transformar, así como el destino de los mismos, de modo que cada campo de la tabla coincida, Figuras 8 y 9:

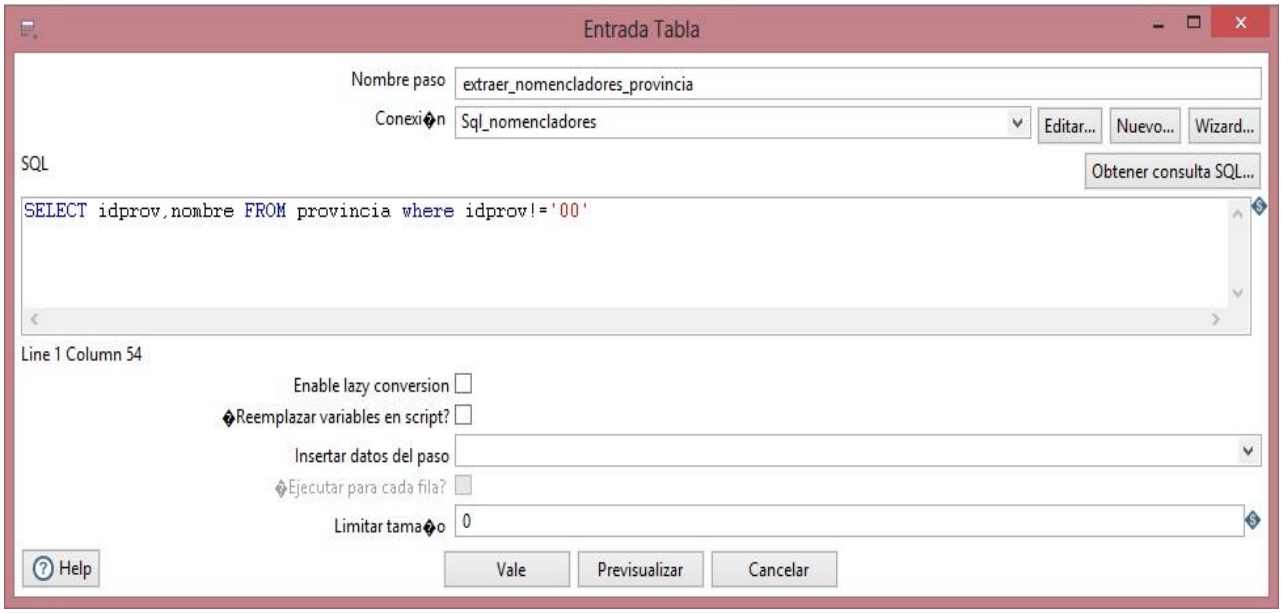


Figura 8 Entrada Tabla

En el caso de la funcionalidad “Insertar Actualizar” inserta solo los campos que no existan en la tabla destino. En caso contrario se le pueden especificar los campos de la fila que se quieran actualizar, permitiendo realizar modificaciones en las filas de la tabla.

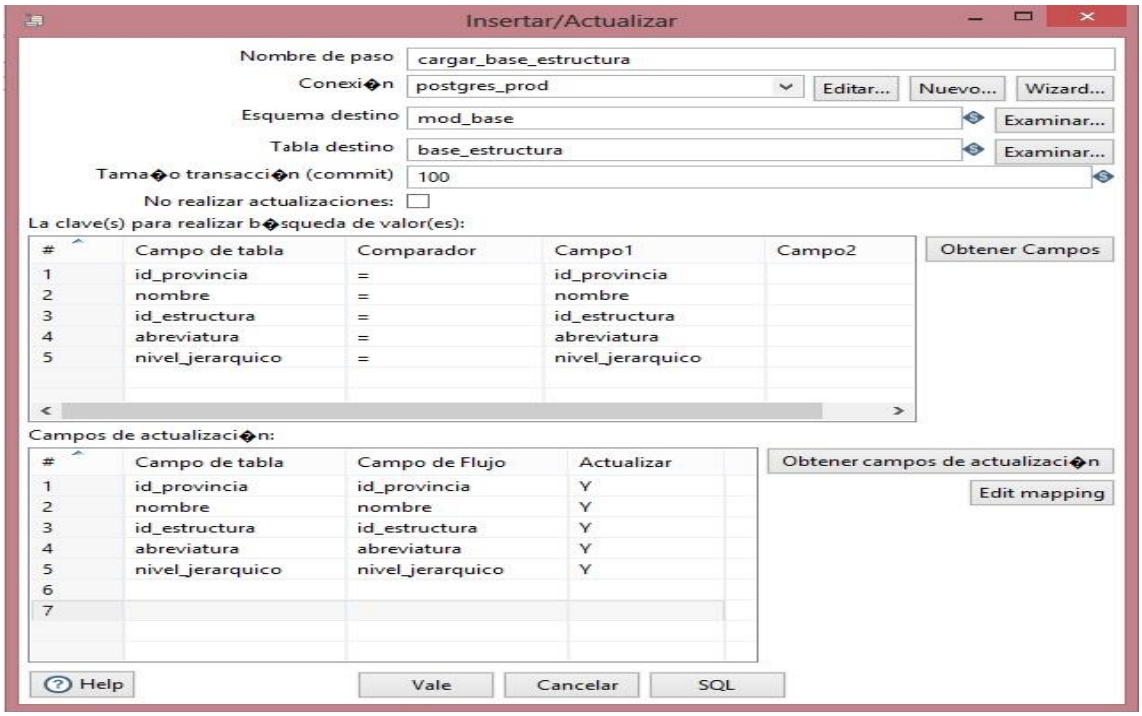


Figura 9 Insertar Actualizar

- ✓ En el caso de la tabla base_estructura se crean las variables id_estructura y nivel_jerarquico utilizando la funcionalidad “Valor Java Script Modificado”. El id_estructura siempre se conforma 0+idprov y la variable nivel_jerarquico con valor 0, Figura 10.

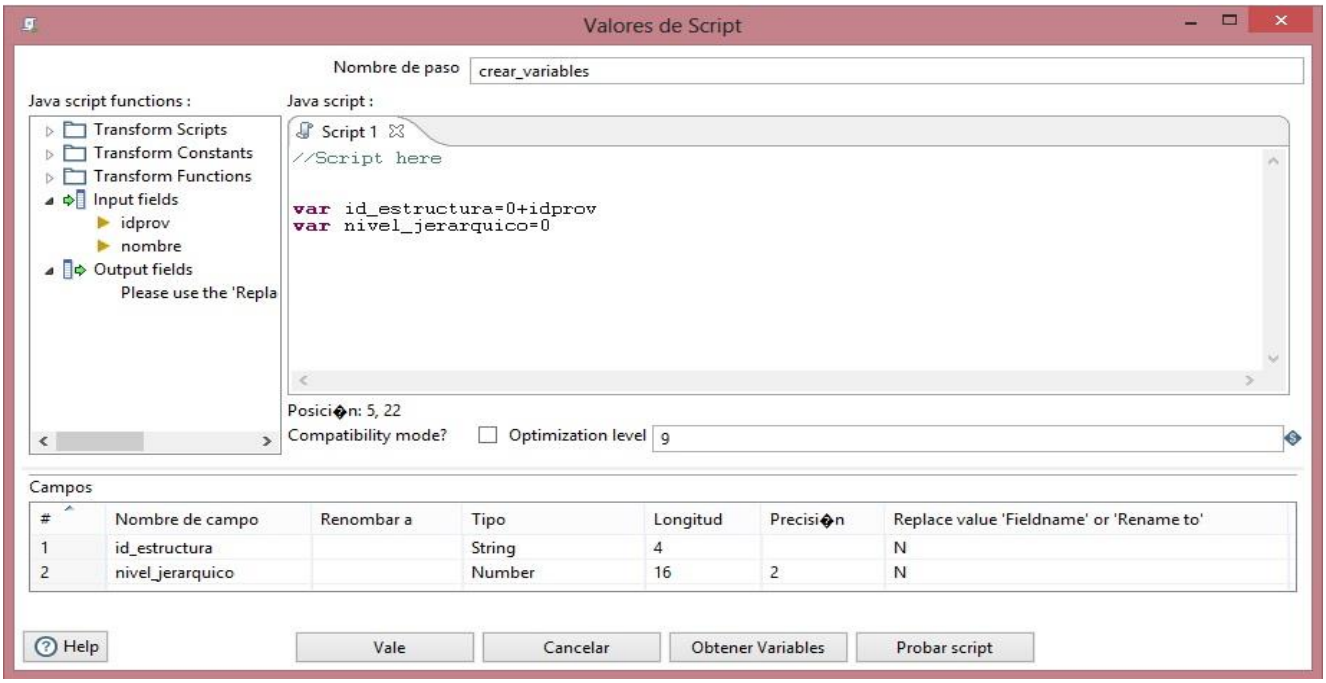


Figura 10 Valor Java Script Modificado

Luego se realiza un mapeo de valores para crear las abreviaturas para cada una de las estructuras. Cada estructura posee un idprov que es el número que la identifica. Este se utiliza para asignar dichas abreviaturas, Figura 11.

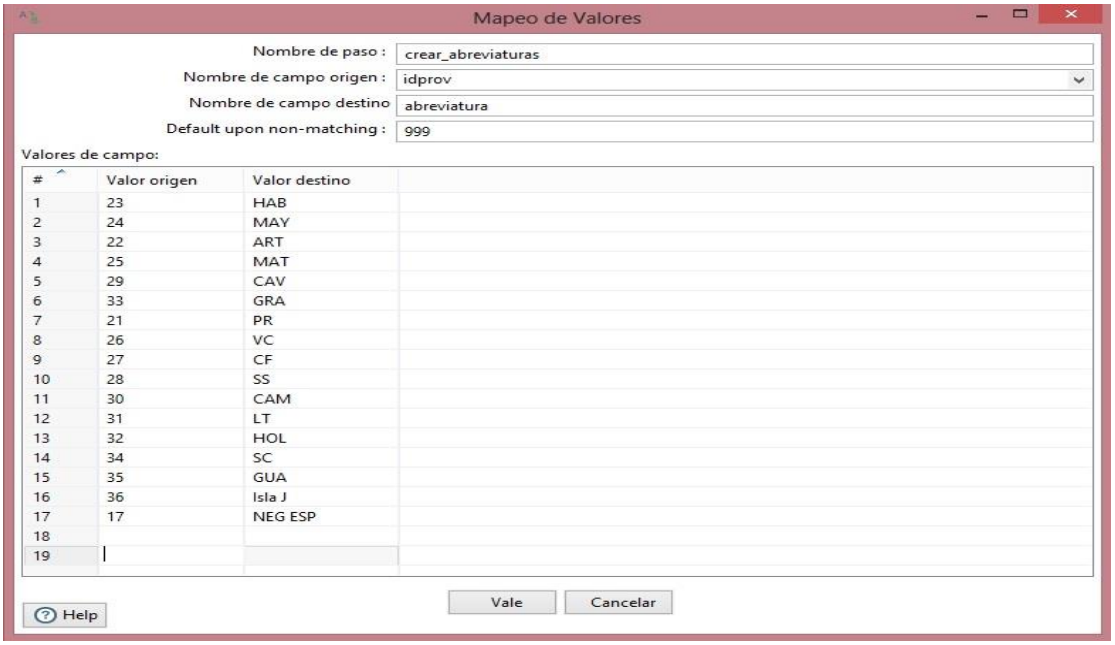


Figura 11 Mapeo de Valores

- ✓ Después se realiza un mapeo de valores para cambiar los números que vienen en el campo idprov extraídos del origen por los que deben insertarse, Figura 12.

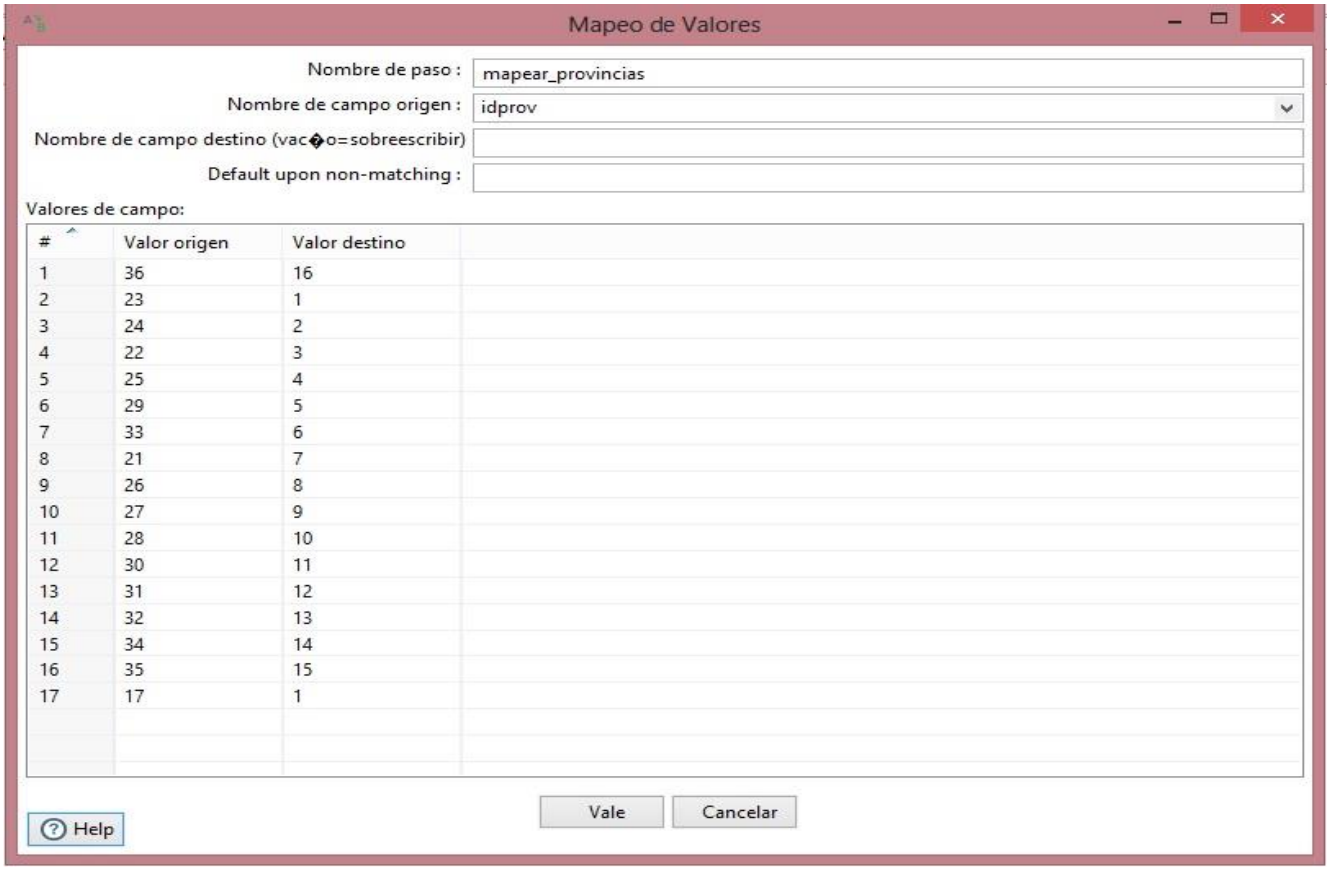


Figura 12 Mapeo de Valores

- ✓ Posteriormente se renombra el campo idprov a id_provincia utilizando la funcionalidad "Selecciona/Renombra valores" que posee Kettle, Figura 13.

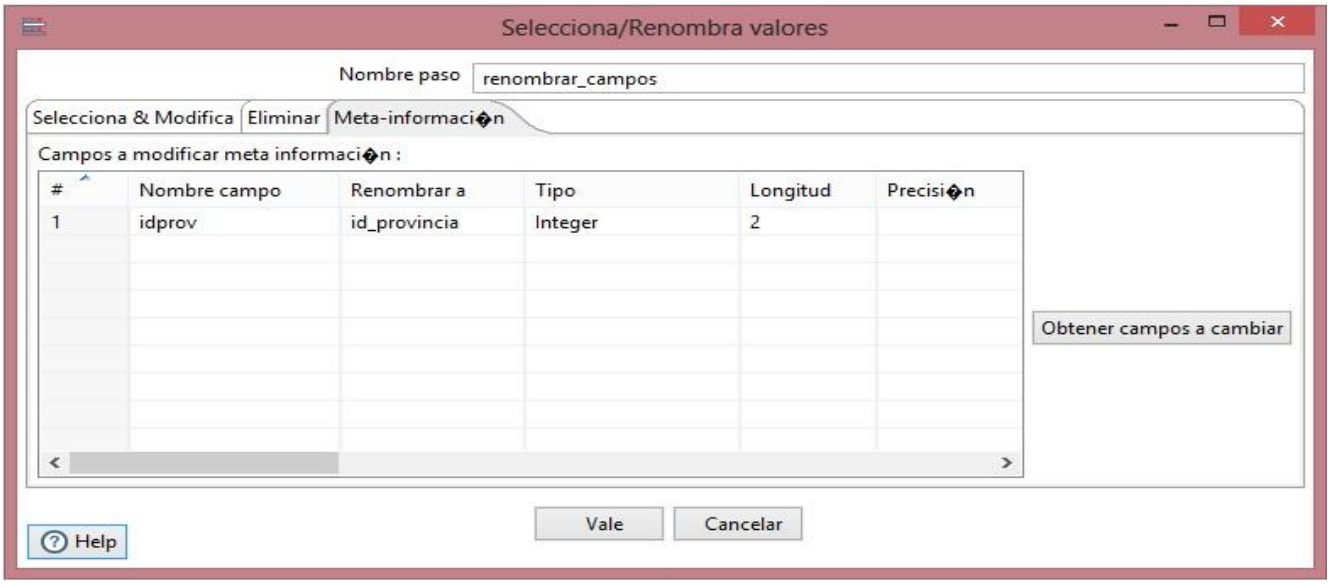


Figura 13 Renombrar campo

- ✓ Por último se configura la funcionalidad de “Insertar Actualizar” como se muestra en la Figura 9 y se ejecuta la transformación.

El esquema resultante de la transformación queda de la siguiente forma:

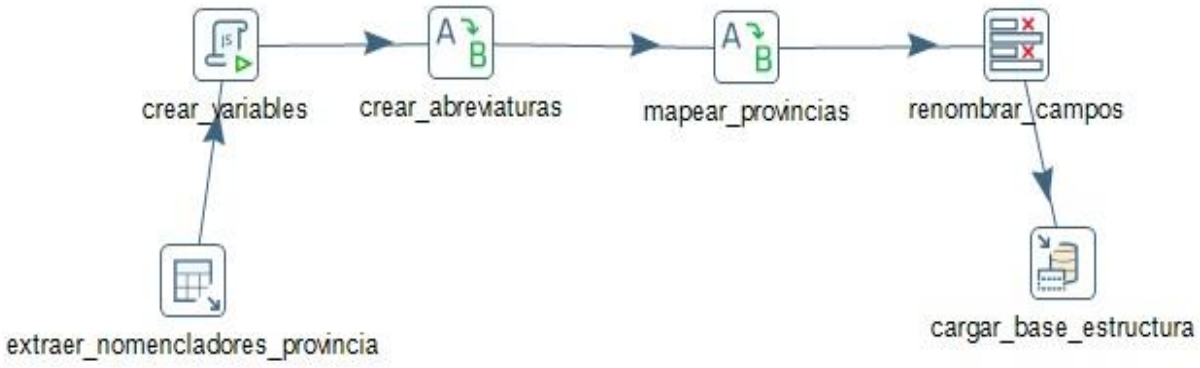


Figura 14 Esquema de la transformación base_estructura

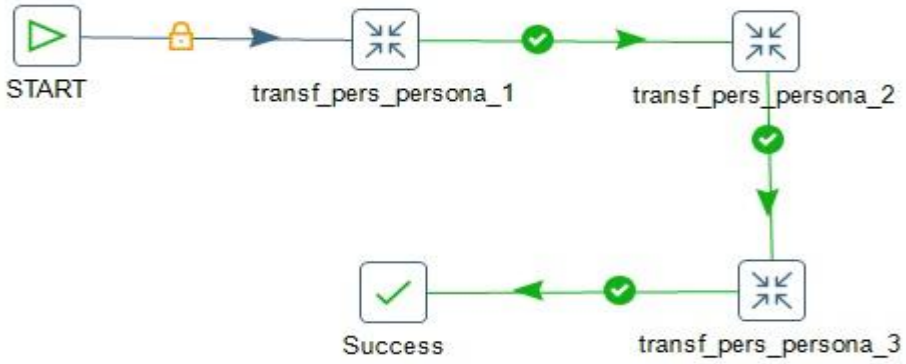


Figura 15 Diseño de ejecución del trabajo trab_pers_persona

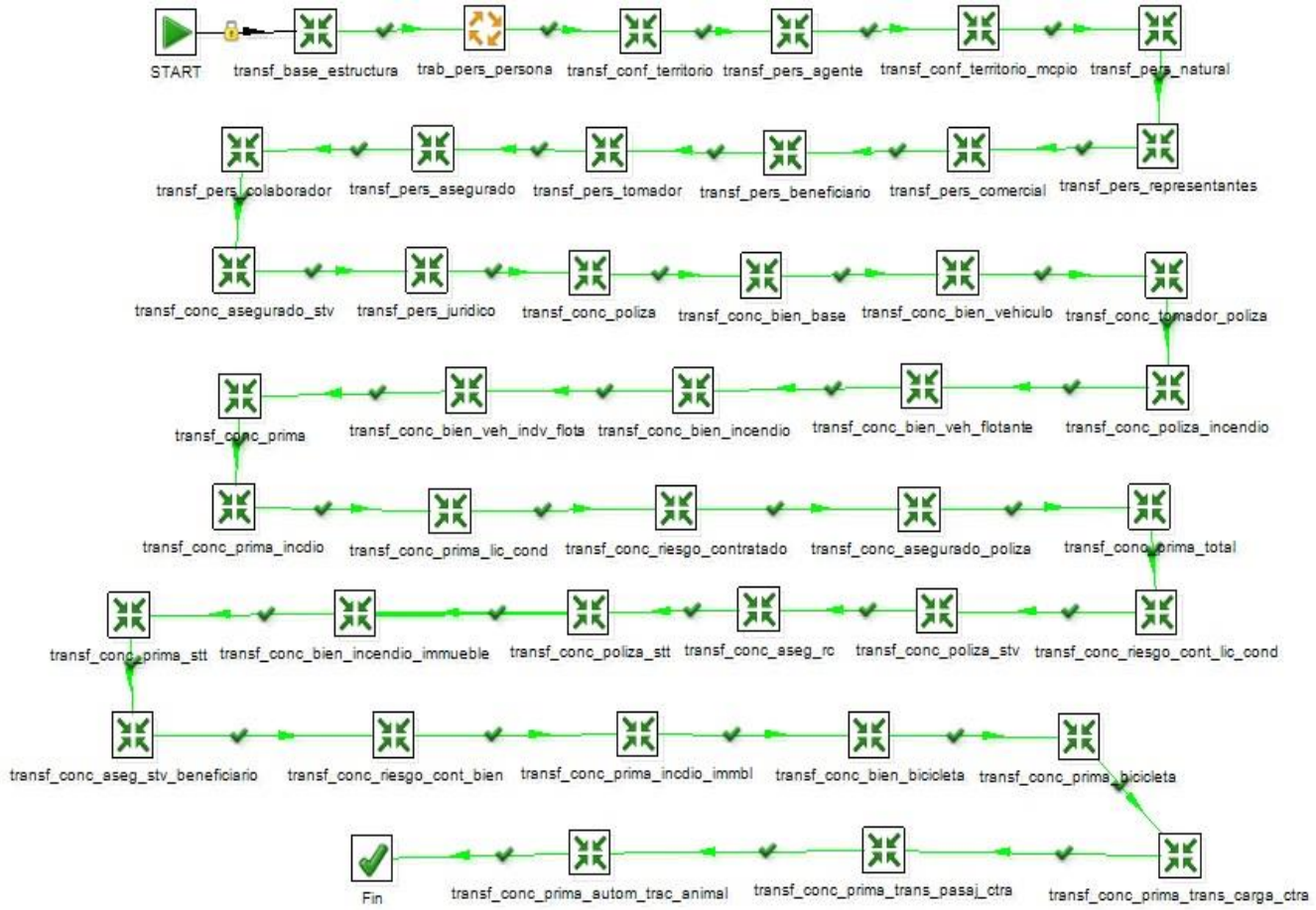


Figura 16 Ejecución del trabajo general

3.4 Pruebas de integración de los datos

“Factor crítico para el éxito de la migración de la base de datos es la realización de pruebas , las cuales inicialmente, pueden ser a pequeña escala para validar o modificar la arquitectura final y el plan de migración, así como para comprobar que las aplicaciones que harán uso de la base de datos funcionan correctamente y optimizar los tiempos y recursos necesarios” (36).

En la realización de las pruebas de integración se tomaron juegos de datos aleatorios de las tablas fuente y se determinaron los posibles valores que se debían obtener. Los datos fueron cargados en una BD de prueba, para determinar si eran cargados correctamente.

Para finalmente comprobar si tuvo éxito la migración, se realizaron consultas a las tablas fuente después de la migración y a las tablas destino, para verificar si los datos obtenidos eran iguales, garantizando así el éxito de la estrategia.

Las siguientes tablas muestran lo antes explicado para las tablas conc_poliza y conc_prima:

Atributo	Entrada	Transformación	Salida esperada	Salida real
id_poliza	000000000012	Se eliminaron los espacios en blanco del comienzo y el final	000000000012	000000000012
f_inicio	1900-01-01 00:00:00.000	-	1900-01-01 00:00:00.000	1900-01-01
f_fin	-	-	-	-
id_moneda	1	Se realizó el mapeo de datos para asignar el id correspondiente	2	2
id_deducible	8	Se realizó una búsqueda en base de datos para encontrar el id_deducible dado el producto	10	10
valor_asegurado	17000.00	-	17000.00	17000.00
id_forma_contratacion	Línea 101 Modalidad 4	Se obtuvo mediante la implementación de código javascript, dependiendo de la modalidad y la línea se asigna un valor u otro	2	2
id_producto	49	Se mapean los productos y se asigna su valor correspondiente	18	18

id_agente	1812	Se realiza una búsqueda en BD dado el número de inscripción del agente, para obtener su identificador	59050609237	59050609237
id_vias_financiamiento	2	Se mapean los campos y se asigna su valor correspondiente	4	4
bonificada	1	Se obtiene usando código javascript, en caso de ser 1 se asigna **true** y de ser 0 se asigna **false**	true	true
id_ramo	-	Se crea con valor 1	1	1
id_modalidad_producto	14	Se mapean las modalidades y se asigna su valor correspondiente	6	6
id_estado	5	Se mapean los valores del campo **tipopoliza**, se asigna su valor correspondiente y se renombra el campo a **id_estado**	2	2
bonificación	50.00	-	50.00	50.00

cod_poliza	611000021	Se eliminan los espacios en blanco del inicio y del final	611000021	611000021
f_inicio_vigencia	2007-07-10 00:00:00.000	-	2007-07-10 00:00:00.000	2007-07-10
cod_historico	-	Se genera a partir de una secuencia	1	1
id_estructura	3206	Se obtiene del campo dpa, utilizando javascript se corta el código a la mitad, luego se agrega 0+los dos primeros dígitos	032	032
f_fin_vigencia	2008-07-10 00:00:00.000	-	2008-07-10 00:00:00.000	2008-07-10
bien	-	Se obtiene dependiendo el valor del campo producto, usando javascript, si el producto no es de tipo vida se asigna **true** y **false** en caso contrario	True	True
persona	1	Se obtiene usando código javascript, en caso de ser 1 se asigna **true** y de	True	True

		ser 0 se asigna **false**		
id_tipo_persona	-	Se crea usando javascript, dependiendo del valor que se obtenga en la búsqueda en BD sobre la tabla **pers_juridico,** si el campo **reane** es igual a -1 se asigna 2, en otro caso se asigna el valor 1	-	1

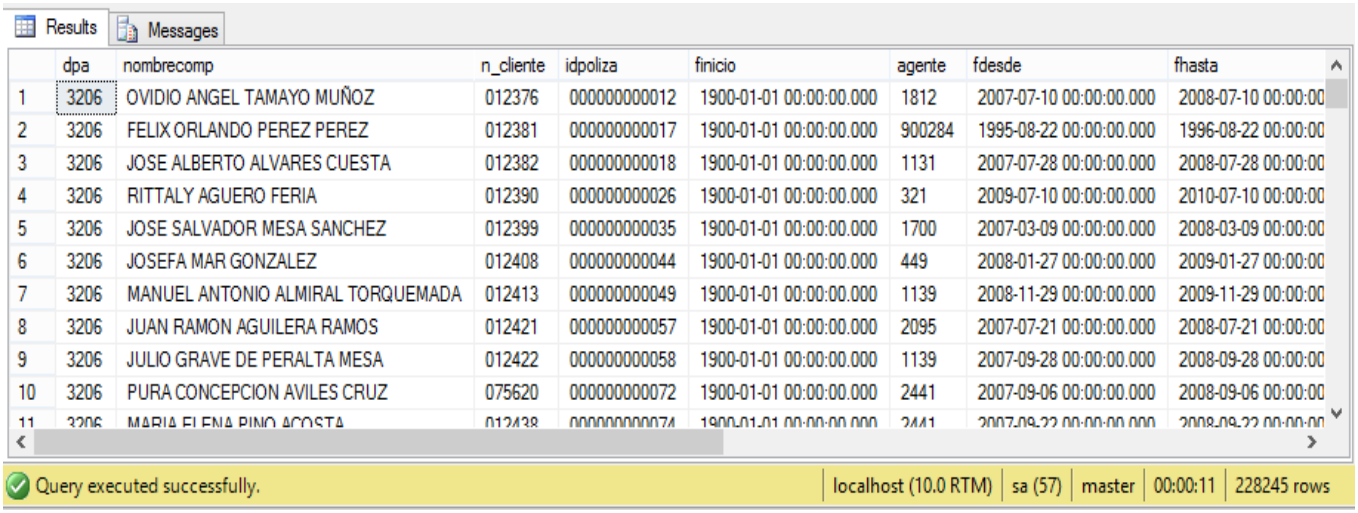
Tabla 10 Tabla conc_poliza

Posteriormente se muestra las consultas realizadas para comprobar la integración de los datos y la existencia de los mismos en ambas BD:

Consulta SQL para la BD fuente:

SELECT distinct dpa, nombrecomp, n_cliente, p.idpoliza, finicio, agente, fdesde, fhasta, tipopoliza, formapago, p.femicion, moneda, pr.modalida, valoraseg, primas, p_bon_rec, p_desc as deducible, producto, pr.lineas FROM sis_hol.dbo.cliente c join sis_hol.dbo.poliza p on (c.n_cliente=p.cliente) join nomencladores.dbo.producto pr on pr.id=p.producto;

Se muestra la salida que arrojó el SGBD con la consulta SQL anterior:



Results | Messages

	dpa	nombrecomp	n_cliente	idpoliza	finicio	agente	fdesde	fhasta
1	3206	OVIDIO ANGEL TAMAYO MUÑOZ	012376	000000000012	1900-01-01 00:00:00.000	1812	2007-07-10 00:00:00.000	2008-07-10 00:00:00
2	3206	FELIX ORLANDO PEREZ PEREZ	012381	000000000017	1900-01-01 00:00:00.000	900284	1995-08-22 00:00:00.000	1996-08-22 00:00:00
3	3206	JOSE ALBERTO ALVARES CUESTA	012382	000000000018	1900-01-01 00:00:00.000	1131	2007-07-28 00:00:00.000	2008-07-28 00:00:00
4	3206	RITTALY AGUERO FERIA	012390	000000000026	1900-01-01 00:00:00.000	321	2009-07-10 00:00:00.000	2010-07-10 00:00:00
5	3206	JOSE SALVADOR MESA SANCHEZ	012399	000000000035	1900-01-01 00:00:00.000	1700	2007-03-09 00:00:00.000	2008-03-09 00:00:00
6	3206	JOSEFA MAR GONZALEZ	012408	000000000044	1900-01-01 00:00:00.000	449	2008-01-27 00:00:00.000	2009-01-27 00:00:00
7	3206	MANUEL ANTONIO ALMIRAL TORQUEMADA	012413	000000000049	1900-01-01 00:00:00.000	1139	2008-11-29 00:00:00.000	2009-11-29 00:00:00
8	3206	JUAN RAMON AGUILERA RAMOS	012421	000000000057	1900-01-01 00:00:00.000	2095	2007-07-21 00:00:00.000	2008-07-21 00:00:00
9	3206	JULIO GRAVE DE PERALTA MESA	012422	000000000058	1900-01-01 00:00:00.000	1139	2007-09-28 00:00:00.000	2008-09-28 00:00:00
10	3206	PURA CONCEPCION AVILES CRUZ	075620	000000000072	1900-01-01 00:00:00.000	2441	2007-09-06 00:00:00.000	2008-09-06 00:00:00
11	3206	MARIA ELENA PINO ACOSTA	012438	000000000074	1900-01-01 00:00:00.000	2441	2007-09-22 00:00:00.000	2008-09-22 00:00:00

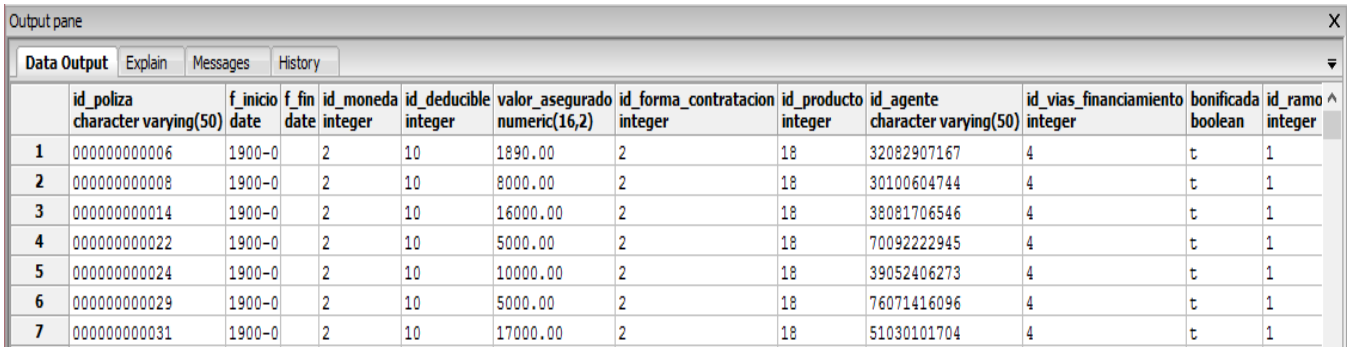
Query executed successfully. | localhost (10.0 RTM) | sa (57) | master | 00:00:11 | 228245 rows

Figura 17 Salida de la consulta SQL en la fuente

Consulta SQL para la BD destino:

SELECT id_poliza, f_inicio, f_fin, id_moneda, id_deducible, valor_asegurado, id_forma_contratacion, id_producto, p.id_agente, id_vias_financiamiento, bonificada, id_ramo, id_modalidad_producto, id_estado, bonificacion, cod_poliza, f_inicio_vigencia, cod_historico, p.id_estructura, f_fin_vigencia, bien, persona, id_tipo_persona, a.*, e.*
FROM mod_comercial.conc_poliza p join mod_base.pers_agente a on (p.id_agente=a.id_agente) join mod_base.base_estructura e on (p.id_estructura=e.id_estructura);

Se muestra la salida del SGBD destino después de realizada la consulta SQL, permitiendo conocer el total de datos que existen en la fuente y el destino. En este caso fueron insertados 228245, cumpliendo con el 100% de las filas:



Output pane

Data Output | Explain | Messages | History

	id_poliza character varying(50)	f_inicio date	f_fin date	id_moneda integer	id_deducible integer	valor_asegurado numeric(16,2)	id_forma_contratacion integer	id_producto integer	id_agente character varying(50)	id_vias_financiamiento integer	bonificada boolean	id_ramo integer
1	000000000006	1900-0		2	10	1890.00	2	18	32082907167	4	t	1
2	000000000008	1900-0		2	10	8000.00	2	18	30100604744	4	t	1
3	000000000014	1900-0		2	10	16000.00	2	18	38081706546	4	t	1
4	000000000022	1900-0		2	10	5000.00	2	18	70092222945	4	t	1
5	000000000024	1900-0		2	10	10000.00	2	18	39052406273	4	t	1
6	000000000029	1900-0		2	10	5000.00	2	18	76071416096	4	t	1
7	000000000031	1900-0		2	10	17000.00	2	18	51030101704	4	t	1

Figura 18 Salida de la consulta SQL en el destino

Se muestra la tabla con el juego de datos utilizado para conc_prima:

Atributo	Entrada	Transformación	Salida esperada	Salida real
id_prima	-	Se genera a partir de una secuencia	-	2
suma_asegurada	100115.00	-	100115.00	100115.00
prima_parcial	384.80	-	384.80	384.80
prima_total	481.00	-	481.00	481.00
id_poliza	170690054072	Se eliminaron los espacios en blanco del comienzo y el final	170690054072	170690054072
id_asegurado	32069180766	Se realiza una búsqueda en BD dado el número del cliente, para obtener su id_asegurado en caso de existir en la tabla pers_asegurado	32069180766	32069180766
id_forma_pago	3	Se realiza un mapeo de datos para asignar los valores correspondientes en cada caso	1	1
bonificacion	20.00	-	20.00	20.00
f_inicio	2017-01-30 00:00:00.000	-	2017-01-30 00:00:00.000	2015-02-27
f_fin	2018-01-30 00:00:00.000	-	2018-01-30 00:00:00.000	2018-01-30

cod_historico	-	Se crea a partir de una secuencia	-	2

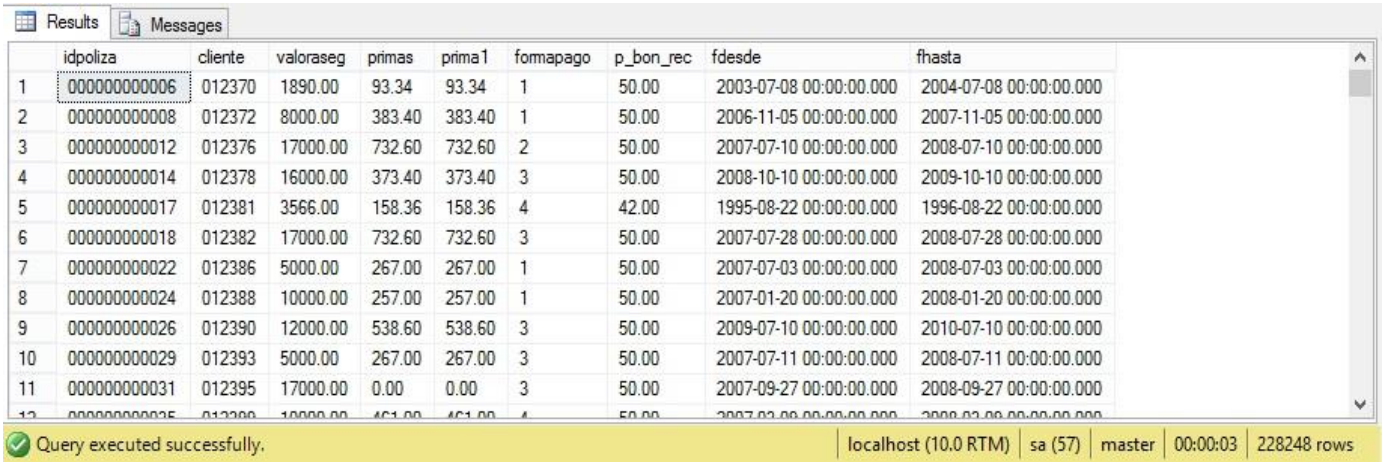
Tabla 11 Tabla conc_prima

Posteriormente se muestra las consultas realizadas para comprobar la integración de los datos y la existencia de los mismos en ambas BD:

Consulta SQL para la BD fuente:

SELECT distinct p.idpoliza, p.cliente, p.valoraseg, p.primas, p.prima1, p.formapago, p.p_bon_rec, p.fdesde, p.fhasta FROM sis_hol.dbo.poliza p;

Se muestra la salida obtenida después de realizar la consulta SQL anterior sobre la fuente:


Results Messages

	idpoliza	cliente	valoraseg	primas	prima1	formapago	p_bon_rec	fdesde	fhasta
1	000000000006	012370	1890.00	93.34	93.34	1	50.00	2003-07-08 00:00:00.000	2004-07-08 00:00:00.000
2	000000000008	012372	8000.00	383.40	383.40	1	50.00	2006-11-05 00:00:00.000	2007-11-05 00:00:00.000
3	000000000012	012376	17000.00	732.60	732.60	2	50.00	2007-07-10 00:00:00.000	2008-07-10 00:00:00.000
4	000000000014	012378	16000.00	373.40	373.40	3	50.00	2008-10-10 00:00:00.000	2009-10-10 00:00:00.000
5	000000000017	012381	3566.00	158.36	158.36	4	42.00	1995-08-22 00:00:00.000	1996-08-22 00:00:00.000
6	000000000018	012382	17000.00	732.60	732.60	3	50.00	2007-07-28 00:00:00.000	2008-07-28 00:00:00.000
7	000000000022	012386	5000.00	267.00	267.00	1	50.00	2007-07-03 00:00:00.000	2008-07-03 00:00:00.000
8	000000000024	012388	10000.00	257.00	257.00	1	50.00	2007-01-20 00:00:00.000	2008-01-20 00:00:00.000
9	000000000026	012390	12000.00	538.60	538.60	3	50.00	2009-07-10 00:00:00.000	2010-07-10 00:00:00.000
10	000000000029	012393	5000.00	267.00	267.00	3	50.00	2007-07-11 00:00:00.000	2008-07-11 00:00:00.000
11	000000000031	012395	17000.00	0.00	0.00	3	50.00	2007-09-27 00:00:00.000	2008-09-27 00:00:00.000

Figura 19 Salida de la consulta SQL en la fuente

Consulta SQL para la BD destino:

SELECT id_prima, suma_asegurada, prima_parcial, prima_total, pr.id_poliza, pe.id_asegurado, id_forma_pago, pr.bonificacion, pr.f_inicio,pr.f_fin, pr.cod_historico, pe.*,p.* FROM mod_comercial.conc_prima pr join mod_comercial.conc_poliza p on (pr.id_poliza=p.id_poliza) join mod_base.pers_asegurado pe on (pe.id_asegurado=pr.id_asegurado);

Se muestra la salida obtenida después de realizar la consulta SQL sobre el destino, permitiendo conocer el total de datos que existen en la fuente y el destino. En este caso fueron insertados 203253, de un total de 228248, en el caso de las filas que no fueron insertadas, se explica posteriormente:

Output pane

Data Output | Explain | Messages | History

	id_prima integer	suma_asegurada numeric(16,2)	prima_parcial numeric(16,2)	prima_total numeric(16,2)	id_poliza character varying(50)	id_asegurado character varying(50)	id_forma_pago integer	bonificacion numeric(4,2)	f_inicio date	f_fin date	cod_historico bigint	id_asegurado character varying(50
1	202315	34000.00	460.76	460.76	000000002556	014920	1	38.00	2008-1	2009	202315	014920
2	200697	53000.00	0.00	0.00	000000005613	017977	3	0.00	2010-1	2011	200697	017977
3	200516		0.00	0.00	000000005952	018316	3	15.00	2004-0	2005	200516	018316
4	195871		0.00	0.00	000000013323	025687	3	10.00	2002-0	2003	195871	025687
5	195426		0.00	0.00	000000014529	024934	3	0.00	2007-1	2008	195426	024934
6	192315	13000.00	26.00	26.00	000000019052	031416	3	10.00	2011-0	2012	192315	031416
7	189545	80000.00	252.81	337.08	000000034052	055501	1	0.00	2007-0	2008	189545	055501
8	186357	1583296.98	6084.64	7158.37	000000042631	005763	2	20.00	2009-0	2010	186357	005763

OK. | Unix | Ln 4, Col 1, Ch 341 | 203253 rows. | 44926 ms

Figura 20 Salida de la consulta SQL en el destino

Después de realizadas las consultas SQL, con el objetivo de verificar que los datos fueron migrados correctamente, se pudo constatar la efectividad de la carga inicial, implementando la estrategia propuesta y obteniéndose el resultado esperado. En el caso de las filas que no fueron cargadas en el destino por no cumplir con las especificaciones del negocio, se almacenaron en ficheros de texto, para su posterior tratamiento por los especialistas de la ESEN. Por ejemplo, la fila que hace referencia al id_poliza “170690055132” no fue cargada en el destino por no existir en la tabla cliente de la fuente el número de id_cliente, lo que trajo consigo que cuando fueron cargados los datos a la tabla, no fuera insertado porque los datos no tenían sentido sin el cliente.

3.5 Resultado de la migración

Con la ejecución del trabajo, que realiza cada una de las transformaciones de manera automatizada, los datos que se encontraban en la base de datos del SIGES fueron trasladados al Sistema Gestor de Bases de Datos PostgreSQL mediante el uso de la herramienta Kettle, con la excepción de los datos que presentaban problemas que no fueron cargados. Comprobando la integración de los datos mediante el uso de las pruebas.

3.6 Conclusiones parciales

En este capítulo se describió detalladamente todo el proceso de extracción, transformación y carga usando la herramienta Pentaho Data Integration (Kettle ETL). Además, se aplicaron pruebas de integración de datos para verificar el éxito de la migración y se ejecutaron consultas SQL para

comprobar la integración de los datos, obteniéndose los resultados esperados en cada uno de los casos.

CONCLUSIONES GENERALES

A partir del presente trabajo se diseñó e implementó una estrategia de carga inicial de dato, que permite realizar la carga de los datos que se encuentran almacenados en el SIGES hacia el SISEN, haciendo las limpiezas y transformaciones necesarias para que la información mantenga su integridad, garantizando que el SISEN cuente con la información que necesita.

Para lograr lo antes planteado:

- ✓ Se efectuó un estudio sobre las principales herramientas, tecnologías, metodologías y métodos actuales propuestos para la realización de transformaciones y migraciones de datos, con el objetivo de identificar y seleccionar las posibles a utilizar en la solución del problema planteado.
- ✓ Se definió una estrategia que utilizando la MDPAD permitió generar una serie de artefactos como: perfilado de los datos, reglas del negocio y mapeo de datos, que permitieron llevar a cabo la carga.
- ✓ Se realizó el proceso de extracción, transformación y carga de datos para filtrar, limpiar, homogenizar y agrupar la información proveniente de la fuente de datos.
- ✓ Por último, se efectuaron las pruebas de integración de datos para verificar el resultado de la carga inicial.

RECOMENDACIONES

Después de terminado este trabajo y teniendo en cuenta que cumple con el objetivo planteado, se recomienda:

- ✓ Someter a consideración de la ESEN la aplicación de la estrategia de carga inicial propuesta.
- ✓ Motivar el estudio de la MDPAD para el desarrollo de soluciones que incluyan migraciones de datos sobre las cuales sea necesario realizar algún tipo de transformación.

REFERENCIAS BIBLIOGRÁFICAS

1. *TIC.pdf* [online]. [Accessed 11 June 2017]. Available from: http://www.uv.es/~bellochc/pdf/pwtic1.pdf

2. Revista Digital Universitaria. [online]. [Accessed 13 December 2016]. Available from: http://www.revista.unam.mx/vol.10/num11/art79/int79.htm

3. Definición de TIC. [online]. [Accessed 13 December 2016]. Available from: http://www.serviciostic.com/las-tic/definicion-de-tic.html

4. *Definicion de SGBD* [online]. [Accessed 11 June 2017]. Available from: http://www.alegsa.com.ar/Dic/sgbd.php

5. En Informática ¿Que es ETL? | ddavalos. [online]. [Accessed 13 December 2016]. Available from: https://ddavalos99.wordpress.com/2010/10/05/en-informatica-%C2%BFque-es-etl/

6. *Kimball & Caserta -The Data Warehouse ETL Toolkit [Wiley 2004].pdf* [online]. [Accessed 6 June 2017]. Available from: http://users.itk.ppke.hu/~szoer/DW/Kimball%20&%20Caserta%20-The%20Data%20Warehouse%20ETL%20Toolkit%20%5BWiley%202004%5D.pdf

7. Procesos ETL: Extracción. ¿En qué consiste? [online]. [Accessed 11 June 2017]. Available from: http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/312587/Procesos-ETL-Extracci-n-En-qu-consiste

8. DISPARADORES O TRIGGERS EN UNA BASE DE DATOS - Uneweb Instituto. [online]. [Accessed 11 June 2017]. Available from: http://tecnologiaenvivo.com/%EF%BB%BF%EF%BB%BFdisparadores-o-triggers-en-una-base-de-datos/

9. Qué es un Trigger o Desencadenador. [online]. [Accessed 11 June 2017]. Available from: http://developerji.com/Post/Que-es-un-Trigger-o-Desencadenador/1031

10. *triggers-y-procedimiento-almacenado.pdf* [online]. [Accessed 11 June 2017]. Available from: https://elenahzz.files.wordpress.com/2012/03/triggers-y-procedimiento-almacenado.pdf

11. JORG and STEFAN DESSLOCH. *Formalizing ETL Jobs for Incremental Loading of Data Warehouses* [online]. 2009. [Accessed 7 June 2017]. Available from: https://pdfs.semanticscholar.org/f0f0/00add2c77c9e31ca8b6832731ee07db160b5.pdf

12. *The Data Warehouse Toolkit, 3rd Edition.pdf* [online]. [Accessed 6 June 2017]. Available from: http://www.essai.rnu.tn/Ebook/Informatique/The%20Data%20Warehouse%20Toolkit,%203rd%20Edition.pdf

13. JORGE BUSTILLOS. Comparativa herramientas ETL. [online]. [Accessed 9 March 2017]. Available from: https://es.slideshare.net/JorgeCarlos3/comparativa-herramientas-etl

14. LEVIN, Jonathan. *ETL Tools Comparison*. March, 2008.

15. Datapump. [online]. [Accessed 11 June 2017]. Available from: https://es.scribd.com/document/80773564/Wiki-Datapump

16. Oracle Datapump. [online]. [Accessed 11 June 2017]. Available from: https://es.scribd.com/doc/238541787/Oracle-Datapump

17. ORACLE-BASE - Oracle Transportable Tablespaces. [online]. [Accessed 11 June 2017]. Available from: https://oracle-base.com/articles/misc/transportable-tablespaces

18. Cómo gestionar un proyecto de migración de datos. [online]. [Accessed 10 April 2017]. Available from: http://www.powerdata.es/migracion-de-datos

19. Migración de datos - Migración de datos. [online]. [Accessed 12 June 2017]. Available from: http://www.mailxmail.com/curso-migracion-datos/migracion-datos

20. *Kimball & Caserta -The Data Warehouse ETL Toolkit [Wiley 2004].pdf* [online]. [Accessed 13 December 2016]. Available from: http://users.itk.ppke.hu/~szoer/DW/Kimball%20&%20Caserta%20-The%20Data%20Warehouse%20ETL%20Toolkit%20%5BWiley%202004%5D.pdf

21. *Lenzerini-pods02.pdf* [online]. [Accessed 13 December 2016]. Available from: https://www.cs.ubc.ca/~rap/teaching/534a/readings/Lenzerini-pods02.pdf

22. HERNÁNDEZ SAMPIERI, Roberto, FERNÁNDEZ COLLADO, Carlos and BAPTISTA LUCIO, Pilar. Metodología de la investigación. *La Habana: Editorial Félix Varela*. 2003. Vol. 2.

23. SAMPIERI, Roberto Hernández, COLLADO, Carlos Fernández, LUCIO, Pilar Baptista and PÉREZ, Ma de la Luz Casas. *Metodología de la investigación*. Mcgraw-hill México, 1998.

24. HERNÁNDEZ, Yanisbel González. Repositorio Digital: METODOLOGÍA DE DESARROLLO PARA PROYECTOS DE ALMACENES DE DATOS. [online]. [Accessed 12 June 2017]. Available from: https://repositorio_institucional.uci.cu/jspui/handle/ident/8094

25. ER/Studio - Danysoft. [online]. [Accessed 24 January 2017]. Available from: http://shop.danysoft.com/Embarcadero-ER/Studio

26. Sobre PostgreSQL | www.postgresql.org.es. [online]. [Accessed 24 January 2017]. Available from: http://www.postgresql.org.es/sobre_postgresql

27. pgAdmin: PostgreSQL administration and management tools. [online]. [Accessed 24 January 2017]. Available from: https://www.pgadmin.org/

28. Información general de SQL Server. [online]. [Accessed 24 January 2017]. Available from: https://technet.microsoft.com/es-es/library/ms166352(v=sql.90).aspx

29. About: Java Database Connectivity. [online]. [Accessed 12 June 2017]. Available from: http://es.dbpedia.org/page/Java_Database_Connectivity

30. The Premier Open Source Data Quality Solution | DataCleaner. [online]. [Accessed 2 May 2017]. Available from: https://datacleaner.org/

31. ¿Qué es estrategia? [online]. [Accessed 12 June 2017]. Available from: http://planeacion-estrategica.blogspot.com/2008/07/qu-es-estrategia.html

32. Concepto de estrategia - Definición, Significado y Qué es. [online]. [Accessed 12 June 2017]. Available from: http://definicion.de/estrategia/

33. Definición de Estrategia - Qué es y Concepto. [online]. [Accessed 12 June 2017]. Available from: https://definicion.mx/estrategia/

34. IBM Knowledge Center. [online]. [Accessed 12 June 2017]. Available from: https://www.ibm.com/support/knowledgecenter/es/SSFPJS_8.5.7/com.ibm.wbpm.wid.bpel.doc/busrules/topics/cundbus.html

35. Catálogo Biblioteca UCI en línea › Detalles para: Mercado de datos gestión académica para la sala situacional de la Universidad de las Ciencias Informáticas. [online]. [Accessed 22 February 2017]. Available from: http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=12463

36. SUSANA CORONA CORREA. Enter@te. [online]. [Accessed 20 May 2017]. Available from: http://www.enterate.unam.mx/Articulos/2006/agosto/migracion.htm

ANEXOS

En el caso de los anexos en los que no aparecen las tablas de mapeo de datos se muestran en el capítulo 2 en el epígrafe 2.5 Mapa lógico.

Anexo 1: Mapeo de datos, trabajo y transformaciones de la tabla pers_persona

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_persona	id_persona	varchar					
pers_persona	nombre	varchar	sis_hol	dbo	cliente, Agentes	nombre, NOMBRES	varchar
pers_persona	f_inicio	date					
pers_persona	f_fin	date					
pers_persona	cod_historico	BigInt					
pers_persona	id_estructura	varchar	sis_hol	dbo	cliente	dpa	varchar

Tabla 12 Mapeo de datos de la tabla pers_persona

Posteriormente se muestra el trabajo pers_persona y las tres transformaciones que lo componen:

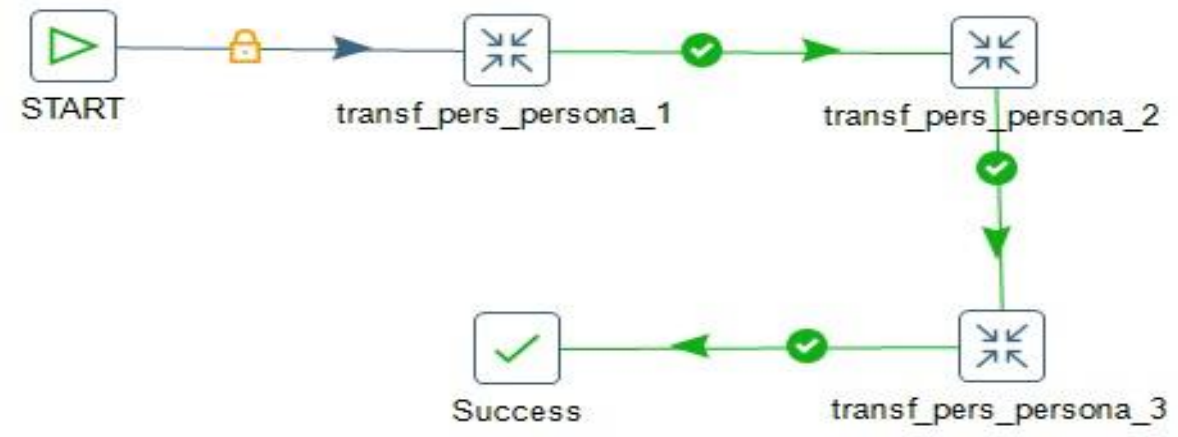


Figura 21 Trabajo pers_persona

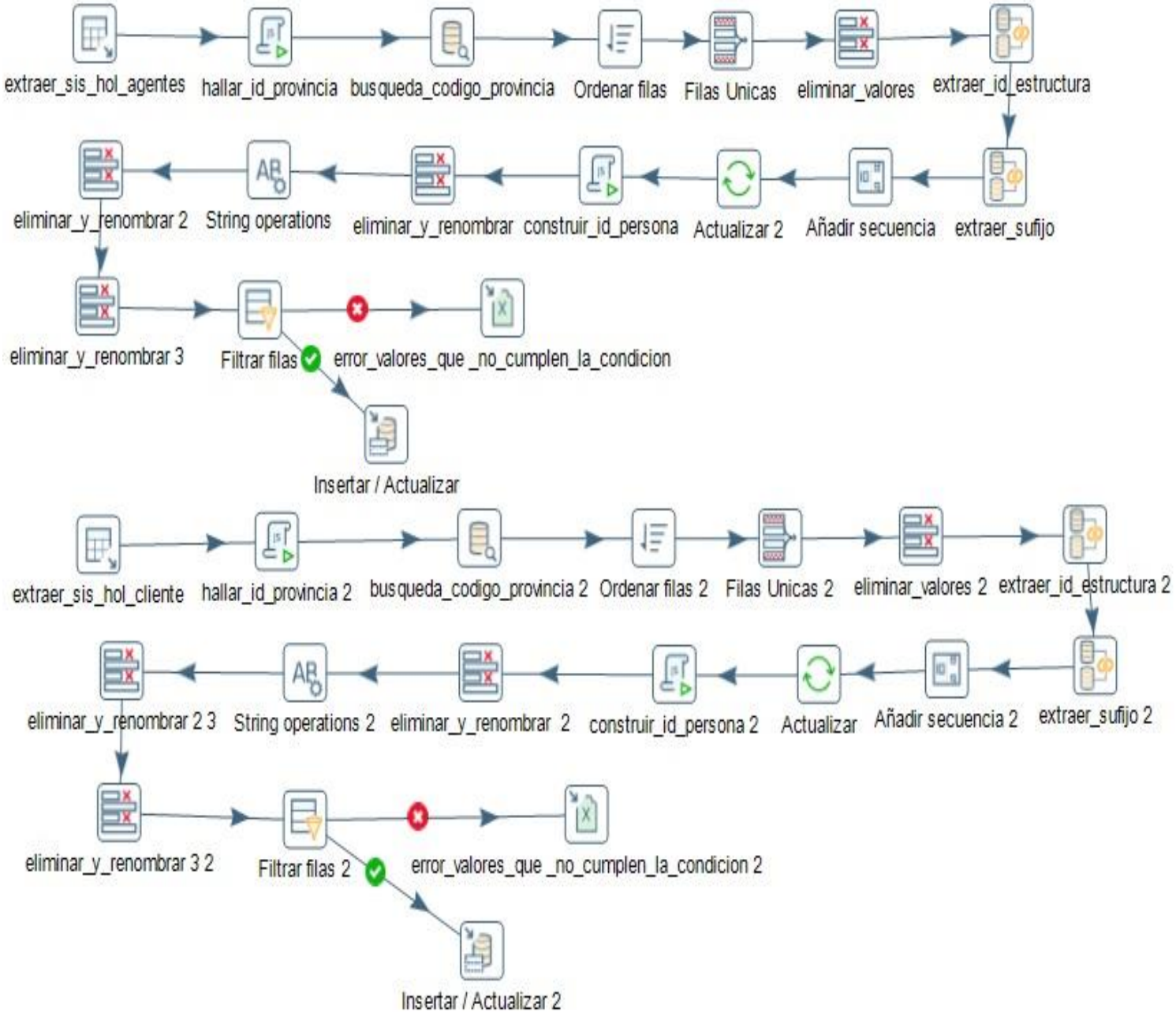


Figura 22 Transformación pers_persona 1

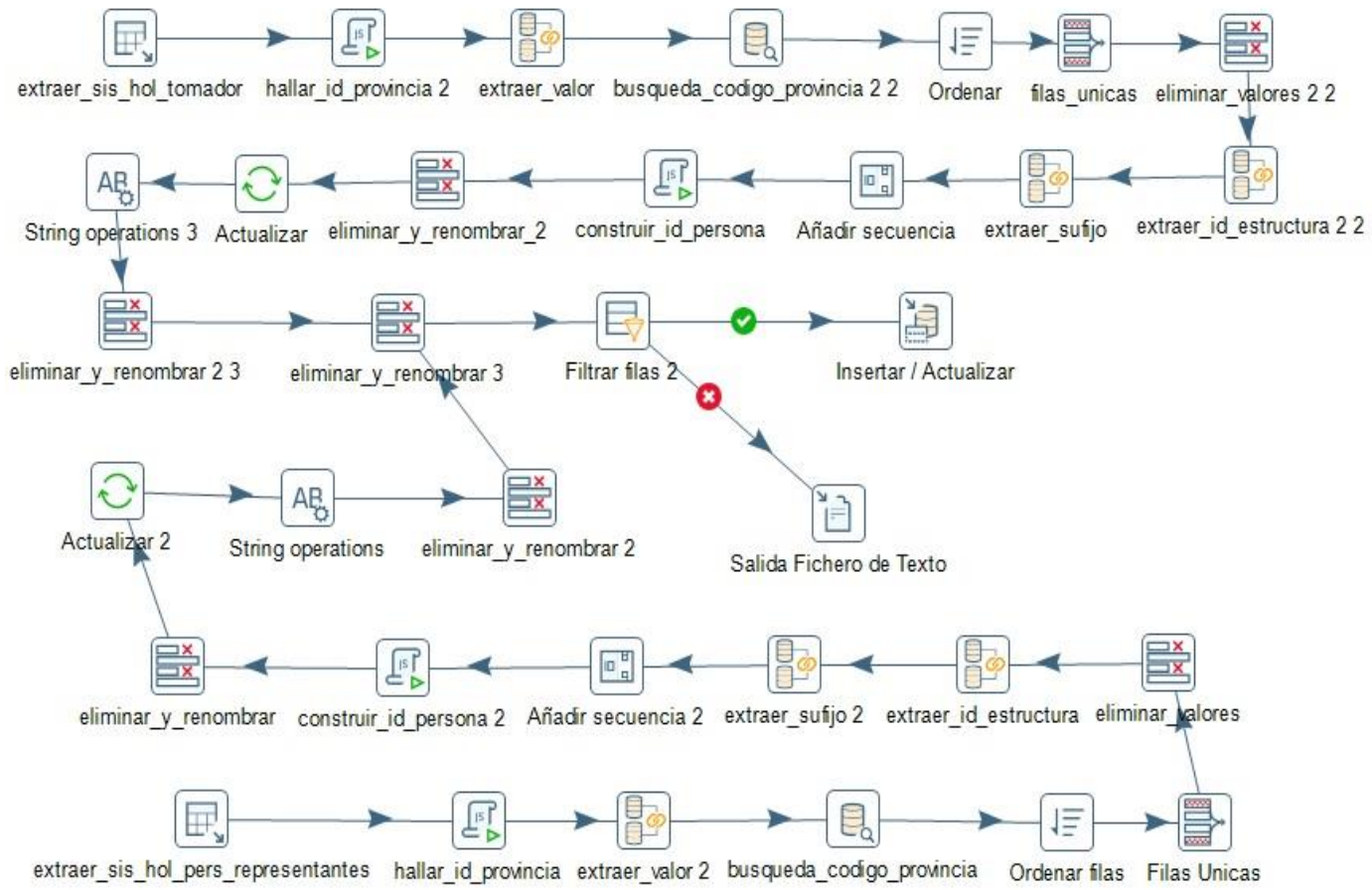


Figura 23 Transformación pers_persona 2

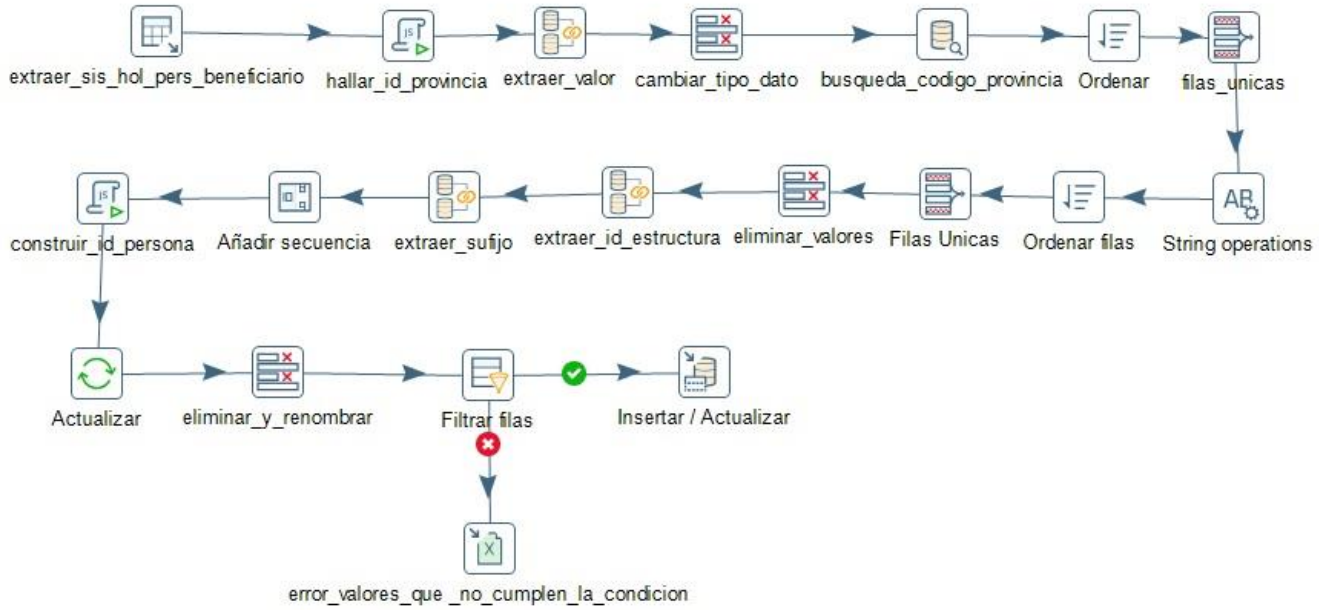


Figura 24 Transformación pers_persona 3

Anexo 2: Mapeo y transformación de la tabla conf_territorio

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
conf_ territorio	id_territorio	integer	nomencladores	dbo	dpa	id_tabla	integer
conf_ territorio	denominacion	varchar	nomencladores	dbo	dpa	newnombre	char
conf_ territorio	codigo	varchar	nomencladores	dbo	dpa	newiddpa	char
conf_ territorio	f_inicio	date					
conf_ territorio	f_fin	date					
conf_ territorio	cod_historico	bigInt					
conf_ territorio	id_estructra	varchar	prod	mod_base	base_ estructura	id_estructura	varchar

Tabla 13 Mapeo de datos de la tabla conf_territorio

A continuación se puede observar la transformación de la tabla conf_territorio:

Figura 25 Transformación conf_territorio

Anexo 3: Transformación de la tabla pers_agente

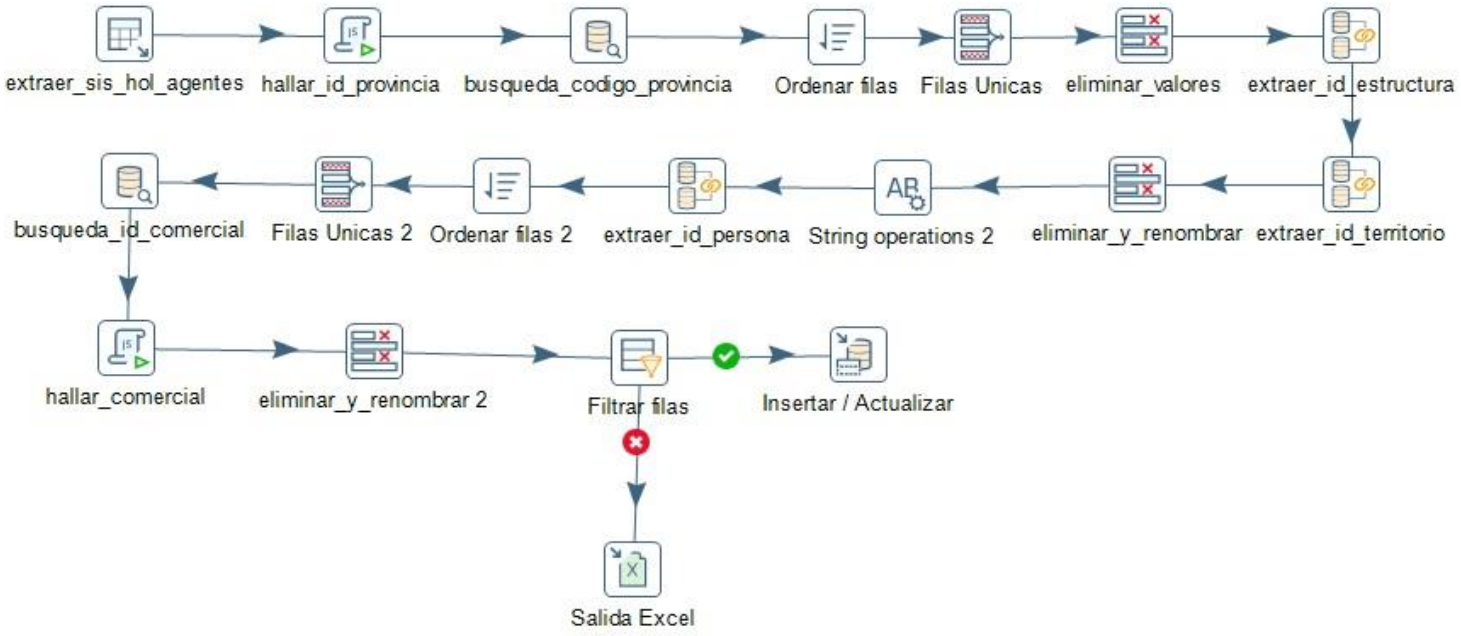


Figura 26 Transformación pers_agente

Anexo 4: Transformación de la tabla pers_natural

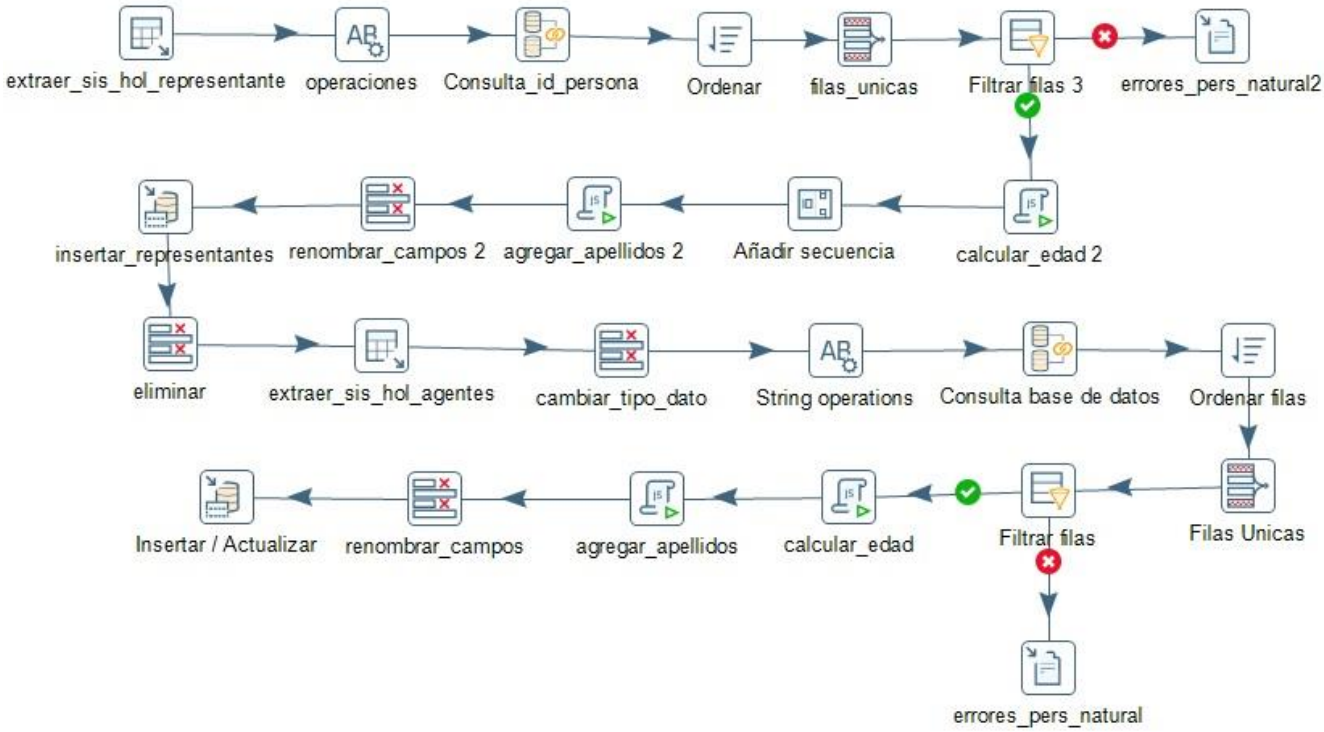


Figura 27 Transformación pers_natural

Anexo 5: Transformación de la tabla pers_representante

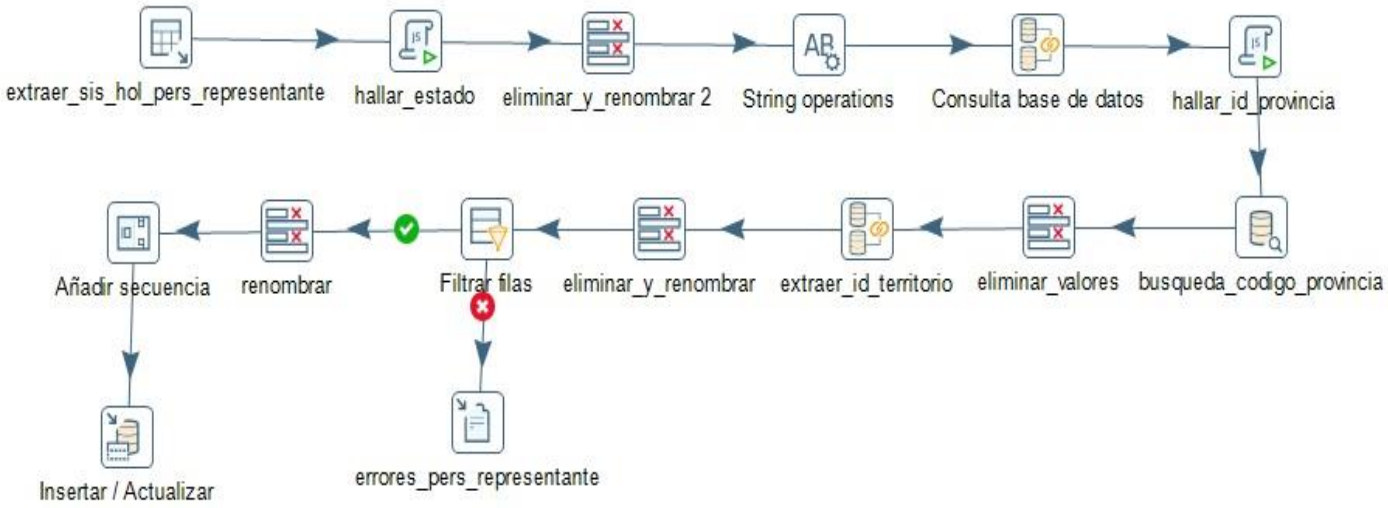


Figura 28 Transformación pers_representante

Anexo 6: Transformación de la tabla pers_comercial

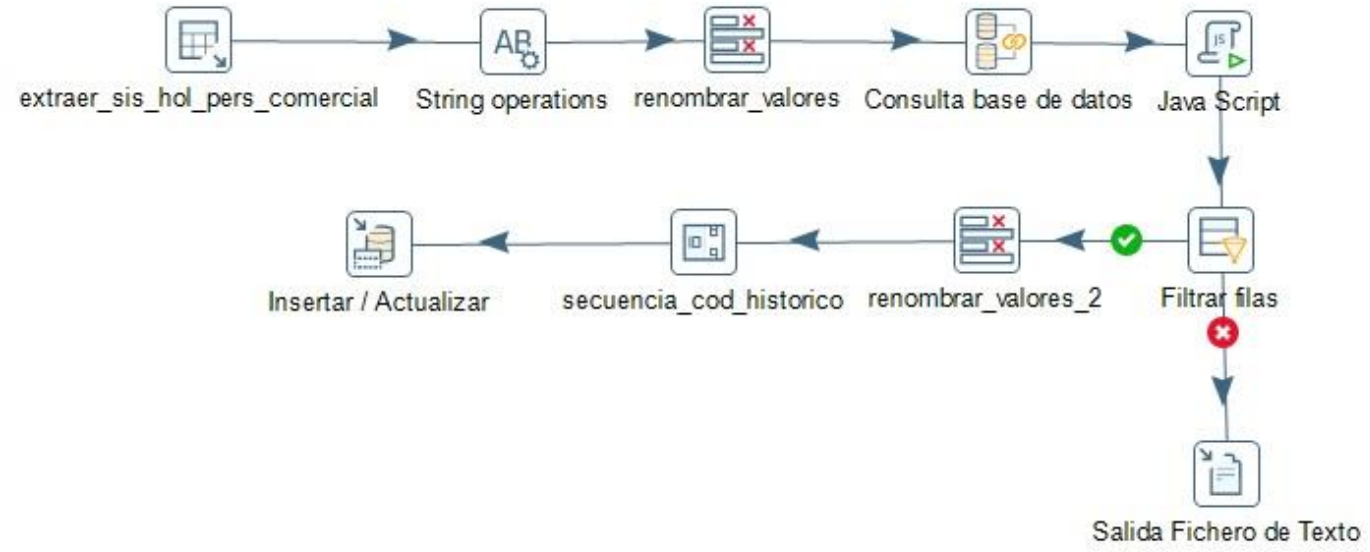


Figura 29 Transformación pers_comercial

Anexo 7: Mapeo de datos y transformación de la tabla pers_beneficiario

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_beneficiario	id_beneficiario	varchar	sis_hol	dbo	v_benefivida	idbenef	char
pers_beneficiario	id_persona	varchar	prod	mod_base	pers_persona	id_persona	varchar
pers_beneficiario	por_ciento	numeric	sis_hol	dbo	v_benefivida	porciento	float

Tabla 14 Mapeo de datos de la tabla pers_beneficiario

Posteriormente se puede observar la transformación correspondiente a la tabla pers_beneficiario:

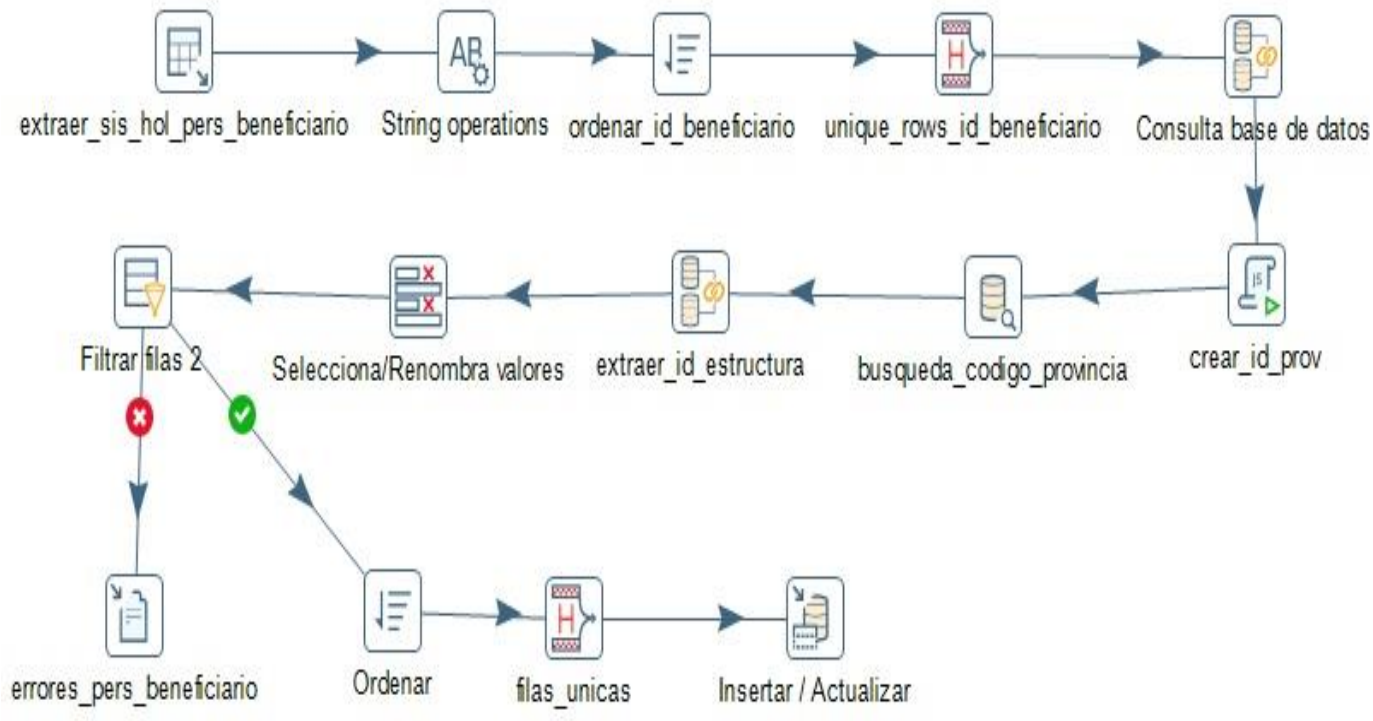


Figura 30 Transformación pers_beneficiario

Anexo 8: Mapeo de datos y transformación de la tabla pers_tomador

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_tomador	id_tomador	varchar	sis_hol	dbo	cliente	n_cliente	char
pers_tomador	id_persona	varchar	prod	mod_base	pers_persona	id_persona	varchar
pers_tomador	direccion	varchar	sis_hol	dbo	cliente	direccion	char
pers_tomador	cargo	varchar					
pers_tomador	f_inicio	date					
pers_tomador	f_fin	date					
pers_tomador	cod_historico	bigInt					

Tabla 15 Mapeo de datos de la tabla pers_tomador

A continuación se muestra la transformación realizada para la tabla anterior:

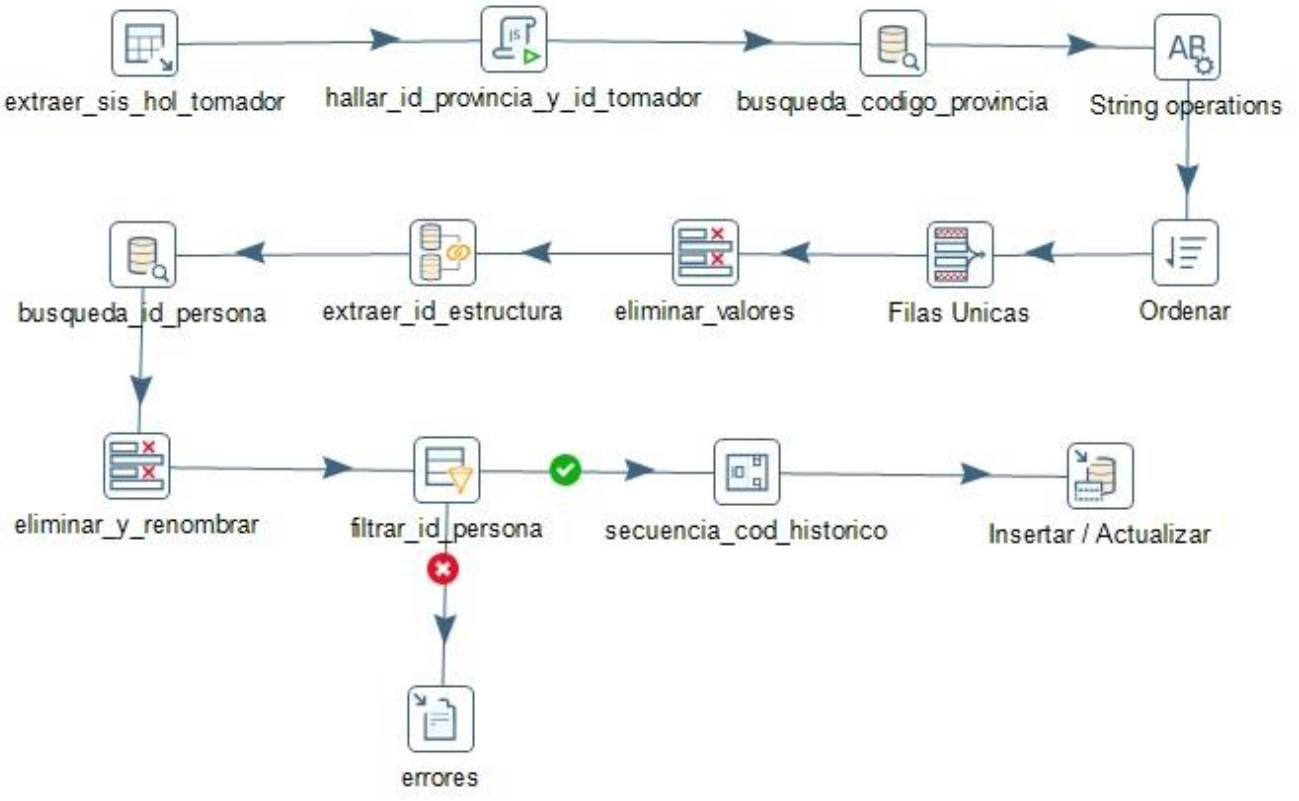


Figura 31 Transformación pers_tomador

Anexo 9: Mapeo de datos y transformación de la tabla pers_asegurado

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_asegurado	id_asegurado	varchar	sis_hol	dbo	cliente	n_cliente	char
pers_asegurado	id_persona	varchar	prod	mod_base	pers_persona	id_persona	varchar
pers_asegurado	id_municipio	integer	prod	public	tmp_tabla_municipio	id_municipio	integer
pers_asegurado	id_provincia	integer	prod	mod_nomencladores	nom_provincia	id_provincia	integer
pers_asegurado	id_organismo	integer	sis_hol	dbo	cliente	id_organismo	char
pers_asegurado	id_sector	integer	sis_hol	dbo	cliente	sector	char
pers_asegurado	direccion	varchar	sis_hol	dbo	cliente	direccion	char
pers_asegurado	cta_bancaria	varchar					
pers_asegurado	id_clasificacion_asegurado	integer					
pers_asegurado	f_inicio	date					
pers_asegurado	f_fin	date					
pers_asegurado	cod_historico	bigInt					

Tabla 16 Mapeo de datos de la tabla pers_asegurado

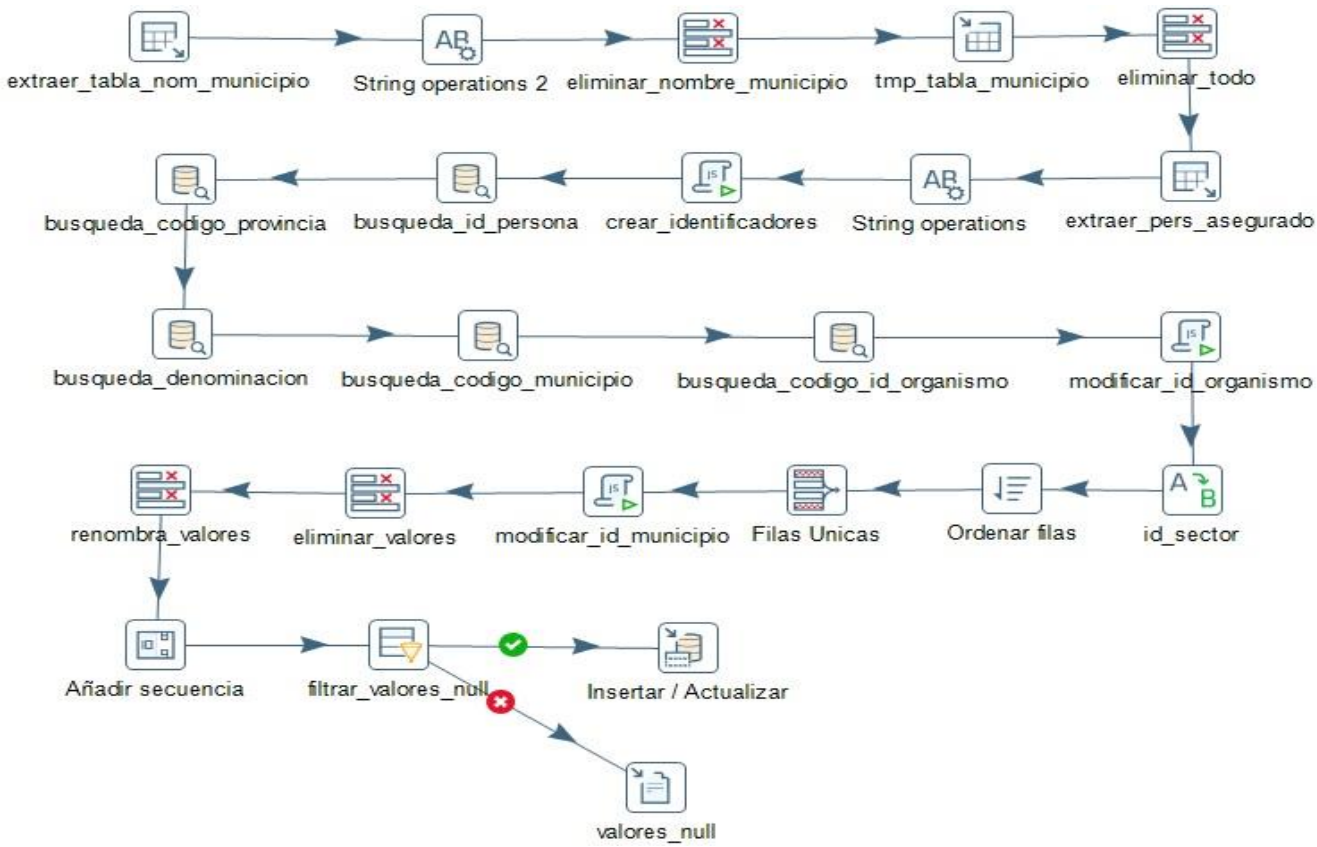


Figura 32 Transformación pers_asegurado

Anexo 10: Mapeo de datos y transformación de la tabla pers_colaborador

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_colaborador	id_colaborador	integer	sis_hol	dbo	v_detalles	idcliente	char
pers_colaborador	nombre	varchar	sis_hol	dbo	cliente	nombrecomp	char
pers_colaborador	apellidos	varchar	sis_hol	dbo	cliente	nombrecomp	char

Tabla 17 Mapeo de la tabla pers_colaborador

A continuación se muestra la transformación para la tabla pers_colaborador:

Figura 33 Transformación pers_colaborador

Anexo 11: Mapeo de datos y transformación de la tabla conc_asegurado_stv

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
conc_asegurado_stv	id_asegurado	varchar	sis_hol	dbo	cliente	n_cliente	char
conc_asegurado_stv	ingreso_mensual	numeric	sis_hol	dbo	v_detalles	salario	decimal
conc_asegurado_stv	zurdo	boolean	sis_hol	dbo	v_detalles	diestro	boolean
conc_asegurado_stv	id_grupo_ocupacional	integer					
conc_asegurado_stv	f_inicio	date					
conc_asegurado_stv	f_fin	date					
conc_asegurado_stv	cod_historico	bigInt					

Tabla 18 Mapeo de la tabla conc_asegurado_stv

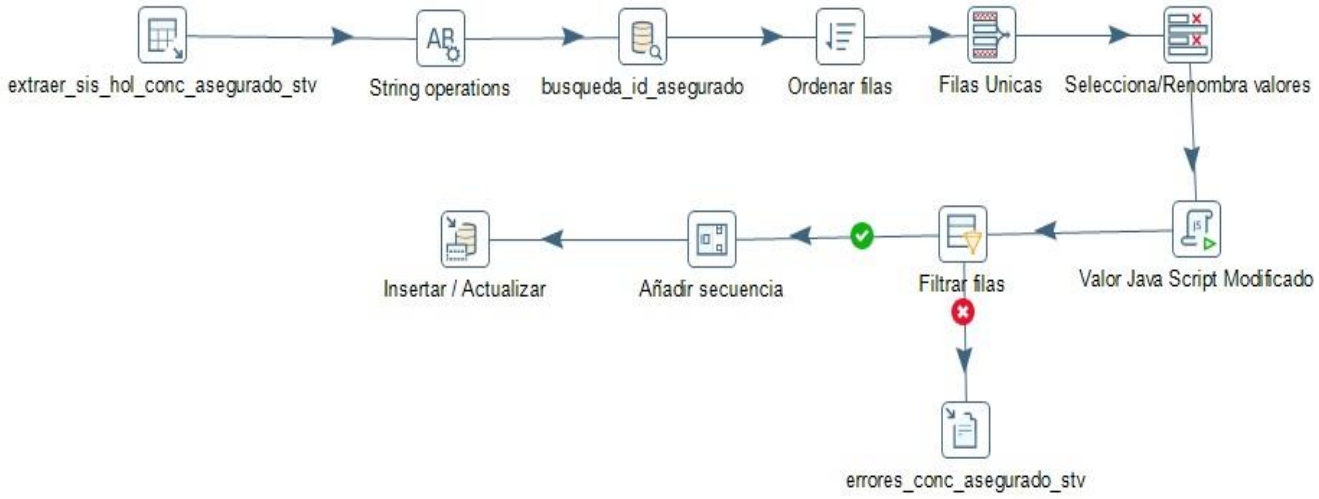


Figura 34 Transformación conc_asegurado_stv

Anexo 11: Mapeo de datos y transformación de la tabla pers_juridico

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
pers_juridico	id_persona	varchar	sis_hol	dbo	cliente	nombrecomp	char
pers_juridico	reane	varchar	sis_hol	dbo	cliente	cod	decimal

Tabla 19 Mapeo de la tabla pers_juridico

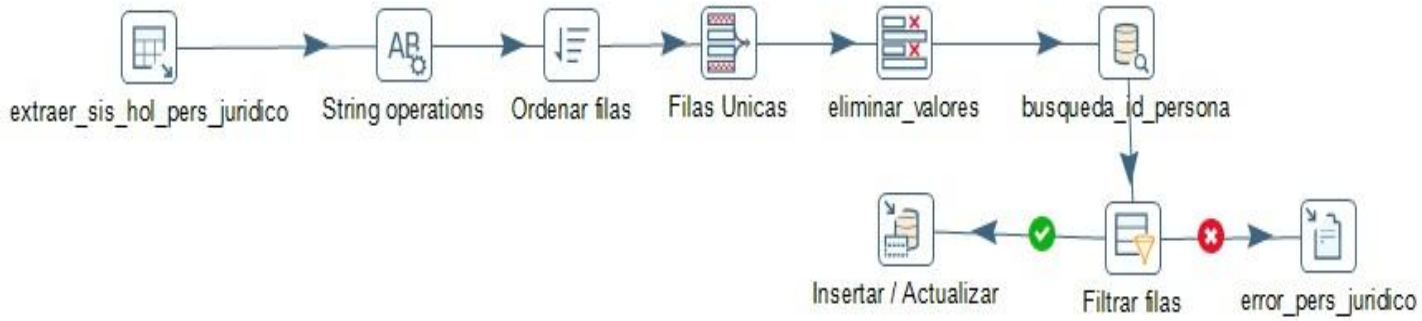


Figura 35 Transformación pers_juridico

Anexo 12: Mapeo de datos de la tabla conc_poliza

Destino			Fuente				
Nombre de la tabla	**Nombre de la columna**	**Tipo de dato**	**Nombre de la base de datos**	**Nombre del Esquema Fuente**	**Nombre de la tabla**	**Nombre de la columna**	**Tipo de dato**
conc_poliza	id_poliza	varchar	sis_hol	dbo	poliza	idpoliza	varchar
conc_poliza	f_inicio	date	sis_hol	dbo	poliza	finicio	date
conc_poliza	f_fin	date	sis_hol	dbo	poliza	fecha_end	date
conc_poliza	id_moneda	integer	sis_hol	dbo	poliza	moneda	integer
conc_poliza	id_deducible	integer	sis_hol	dbo	poliza	p_desc	decimal
conc_poliza	valor_asegurado	numeric	sis_hol	dbo	poliza	valoraseg	numeric
conc_poliza	id_forma_contratacion	integer	sis_hol				
conc_poliza	id_producto	integer	sis_hol	dbo	poliza	producto	integer
conc_poliza	id_agente	varchar	sis_hol	dbo	poliza	agente	integer
conc_poliza	id_vias_financiamiento	integer	sis_hol				
conc_poliza	bonificada	boolean	sis_hol	dbo	poliza	p_bon_rec	numeric
conc_poliza	id_ramo	integer					
conc_poliza	id_modalidad_producto	integer	sis_hol	dbo	poliza	modalidad	integer
conc_poliza	id_estado	integer	sis_hol	dbo	poliza	tipopoliza	integer
conc_poliza	bonificacion	numeric	sis_hol	dbo	poliza	p_bon_rec	numeric
conc_poliza	cod_poliza	varchar	sis_hol	dbo	poliza	poliza	varchar
conc_poliza	f_inicio_vigencia	date	sis_hol	dbo	poliza	fdesde	date
conc_poliza	cod_historico	bigint	sis_hol				
conc_poliza	id_estructura	varchar	sis_hol	dbo	dpa	nombre	varchar
conc_poliza	f_fin_vigencia	date	sis_hol	dbo	poliza	fhasta	date
conc_poliza	bien	boolean					
conc_poliza	persona	boolean					
conc_poliza	id_tipo_persona	integer					

Tabla 20 Mapeo de la tabla conc_poliza