Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Desarrollo de un Datamart que soporte las actividades de control y prevención de enfermedades endémicas del Centro de Investigación en Salud Pública "Dr. Jacinto Convit"

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela

Por los bachilleres:
César A. D'Suze S.
C.I. 17.583.758
César A. Hernández U.
C.I. 21092216

Tutora:
Profa. Concettina Di Vasta

Caracas, Abril de 2015

Agradecimientos

A la **Universidad Central de Venezuela** por brindarnos toda una formación y preparación integral.

A toda mi familia, tíos, primos, abuelos y especialmente a mis padres y hermana quienesfueron, son y serán mi impulso fundamental.

Al Centro **de investigación de Salud Pública "Dr Jacinto Convit"** por permitirnos desarrollar el trabajo de Grado.

A nuestra tutora **Concettina Di Vasta**, por toda la ayuda brindada durante el desarrollo de esta Investigación, y su tiempo invertido en nosotros durante la carrera.

César Hernández

Agradezco la confianza y apoyo brindado por parte de mi familia, que sin duda alguna en el trayecto de mi vida me han demostrado su amor, corrigiendo mis faltas y celebrando mis triunfos.

A nuestra tutora Concettina di Vasta por su valiosa tutoría en todo el proceso de realización de esta tesis.

Agradezco mucho por la ayuda de mis maestros, mis compañeros, y a nuestra querida Universidad Central de Venezuela.

César D'Suze

Dedicatoria

A mi **mama** que es mi mejor amiga y mi ejemplo a seguir.
A mi **papá** que siempre me ha dado aliento en los momentos difíciles.
A mi **hermana** que siempre ha sido mi compañera.
A **Moisés** que siempre ha sido como un hermano menor.
A **mis tías y abuelas** que siempre me han brindado su mano cuando las he necesitado.
A mi mejor amigo **Emmanuel** que siempre ha sido mi mano derecha.
A todos los **profesores** que han puesto su grano de arena en mi formación académica.

César Hernández

A mi abuela, que eres la persona que nos ha hecho como somos y que gracias a ti hemos afrontando nuestras vidas con seguridad y alegría. Es imposible que algún día te devolvamos todo lo que haces por nosotros.

A mi madre, tus palabras guiaron mis pasos, tu cariño suavizó mi carácter y tu amor ilumina mi vida.

A mi padre porque tus consejos me ayudan a tomar las decisiones correctas, porque tu esfuerzo ha hecho que no me falte nada.

César D'Suze

Resumen

En la actualidad el Centro de Investigación en Salud Pública (CISP) "Dr. Jacinto Convit", posee entre sus manos todas las historias de los pacientes egresados de los hospitales del estado Lara. Dicha información es almacenada en archivos Access que son generados a lo largo del año.

A partir de esta data recolectada de distintas parroquias y municipios del Estado Lara, los epidemiólogos del CISP realizando una ardua labor, construyen boletines epidemiológicos semanales, mensuales y anuales, que se componen de un sinfín de reportes, análisis, gráficos y cálculos estadísticos. Dicho proceso de recolección, análisis y generación de boletines es denominado vigilancia epidemiológica.

Para cumplir con dichas tareas los epidemiólogos se apoyan en herramientas ofimáticas, sin embargo, se torna realmente complicado la ejecución de todo este proceso de vigilancia de forma eficiente, redundando en intensas jornadas de trabajo y numerosas horas invertidas en el proceso de estructuración y agregación de la data, que a la larga disminuyen el tiempo que realmente se debería invertir en análisis.

Por lo tanto, teniendo en mente las premisas anteriores, se propone el desarrollo de un Datamart mediante el cual se centralicen y se estandaricen los datos provenientes de los diversos repositorios, permitiendo mediante una herramientas de reporteo, el análisis de dichos datos, en función de automatizar la generación de reportes, garantizando de esta forma reducir el tiempo invertido por parte de los epidemiólogos en el proceso de estructuración y agregación de la data y permitir una mayor holgura para el proceso de análisis de los mismos.

Palabras claves: Datamart, Inteligencia de negocio, Bases de datos, Epidemiologia, Pentaho.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Introducción

Hoy en día el conocimiento y la aplicación de las Tecnologías de Información y Comunicación (TIC) está ampliamente desarrollado en el ámbito mundial, ocasionando transformaciones en todos los espacios de la sociedad. En el sector salud venezolano específicamente, las ventajas de incluir sistemas de información eficientes que generen un impacto positivo en la gerencia pública de salud, redundaría en beneficios sociales, psicológicos, espirituales y hasta económicos en la población.

El sector salud venezolano presenta deficiencias en el manejo de la información, ya que en pocos casos se cuenta con sistemas automatizados que almacenen los datos que se generan en los centros de salud. El área de Epidemiología, se ha visto afectada por esta falta de automatización, ya que no se puede obtener mayor provecho de la información, por las dificultades de acceso o la poca disponibilidad para analizarlos. Esto representa un problema debido a que justamente, el trabajo del Epidemiólogo se refiere al estudio y control de las enfermedades endémicas, para poder luego tomar las decisiones acertadas para su prevención.

El presente trabajo de tesis tiene como objetivo desarrollar un Datamart como herramienta de apoyo a las actividades de prevención y control de enfermedades endémicas del Centro de Investigación en Salud Pública "Dr. Jacinto Convit". A continuación, se presenta una breve descripción del contenido abarcado en este trabajo, el cual está dividido en 3 capítulos:

Inicialmente se presenta una introducción donde se describe de manera general la relación de los sistemas de información y el área de epidemiología en Venezuela, lo que permite dar un acercamiento al tema a investigar en el trabajo especial de grado.

Capítulo 1: Problema de Investigación. En este capítulo se aborda el problema de investigación, así como también los objetivos tanto

específicos como generales de la tesis y se describe la justificación e importancia del trabajo.

Capítulo 2: Marco Conceptual: En este capítulo se cubren todas las bases conceptuales que dan soporte al diseño de un Data Warehouse, bases de datos, sistemas de información e inteligencia de negocio. Además, se describen las principales tecnologías existentes en el mercado para el desarrollo de un Datamart, dentro de tres (3) categorías principales: Sistema Manejador de Bases de Datos, herramientas ETL y herramientas de inteligencia de negocio.

Capítulo 3: Marco Aplicativo: Se explica de manera detallada, la metodología seleccionada, aplicada al diseño y construcción del Datamart. Adicionalmente se muestran cada uno de los procesos y fases utilizadas, haciendo referencia a la investigación realizada y las herramientas involucradas.

Capítulo 1: Problema de Investigación

1. Planteamiento del Problema

El personal del Centro de Investigación en Salud Pública "Dr. Jacinto Convit" (CISP-UCV) debe realizar estudios y análisis de gran cantidad de datos para cumplir efectivamente con los objetivos Generales del CISP-UCV, los cuales son:

• Reducir la incidencia y el impacto de las enfermedades endémicas, que afecte a las poblaciones del área de influencia del CISP-UCV, desarrollando actividades de control de la Leishmaniasis, enfermedad de Chagas, dengue, parasitosis intestinales, tuberculosis, lepra, micosis, cisticercosis; y cualquier otra endemia prevalente en el área.

• Fortalecer la capacidad resolutiva de los niveles regional y local, con el desarrollo de servicios de diagnóstico y tratamiento, investigaciones operacionales, sistemas de información, capacitación y entrenamiento del personal, fortalecimiento en gerencia y administración y organización y participación comunitaria.

A continuación se describen algunas de las tareas realizadas el CISP-UCV en el control de enfermedades endémicas:

1.-Recopilacion de información.
2.-Procesamiento de los datos.
3.-Generacion de informes y reportes.
4.-Vigilancia de enfermedades.

En primera instancia una recopilación de información que proviene de diferentes fuentes y formatos. Esos datos son procesados por el personal del CISP-UCV siguiendo las directrices de los epidemiólogos,

utilizando la herramienta Microsoft Excel para crear tablas y organizar los datos de manera conveniente.

A partir estas estructuras, los epidemiólogos proceden a realizar un análisis de los datos y realizar un conjunto de cálculos estadísticos, gráficos, identificación de tendencias, entre otros; esto permite tomar medidas oportunas y necesarias para controlar o modificar la situación problema. Para hacer un procesamiento adecuado de los datos realizan cuatro (4) actividades principales:

- Evaluación de la confiabilidad de los datos.
- Tabulación.
- Análisis e interpretación.
- Preparación de informes.

Luego se generan informes y reportes particulares, que en conjunto se utilizan para desarrollar el Boletín Epidemiológico con la intención de difundir la información acerca del control y la prevención de una o más enfermedades endémicas.

La vigilancia resulta esencial para las actividades de prevención y control de enfermedades y es una herramienta en la asignación de recursos del sistema de salud, así como en la evaluación del impacto de programas y servicios de salud. El enfoque de la vigilancia requiere equilibrio entre las necesidades de información y las limitaciones para la recolección de datos. El análisis e interpretación de los datos de la vigilancia debe someterse a los límites de la oportunidad, el tiempo, la cobertura geográfica y número de individuos requeridos para que estos sean útiles.

De lo anterior, se desprende la idea de que existe una variada gama de inconvenientes y problemas a la hora de llevar a cabo estos procesos relativos al control y prevención de enfermedades endémicas.

Se puede observar que todas las actividades son realizadas de manera manual, por ello, surgen un conjunto de problemas:

- Gran cantidad de tiempo invertido al llevar a cabo todo el proceso que debe realizarse durante el estudio de las enfermedades endémicas.
- Los datos son manejados utilizando herramientas que no garantizan la integridad.
- El extravío o pérdida de información, debido a la cantidad de documentos que deben ser manipulados o la falta de consistencia de los mismos.
- No es posible llevar un control eficiente de los análisis y productos que son generados debido a que son particulares de cada documento.
- La información no está centralizada.
- La situación actual del Centro de Investigación en Salud Pública "Dr. Jacinto Convit" hace engorrosa y tediosa la ejecución de las tareas necesarias para llevar a cabo el proceso de control y prevención de enfermedades endémicas.

2. Objetivos

2.1 Objetivo General

- Desarrollar un Datamart como herramienta de apoyo a las actividades de prevención y control de enfermedades endémicas del Centro de Investigación en Salud Pública "Dr. Jacinto Convit"

2.2 Objetivos Específicos

- Identificar condiciones políticas, sociales, históricas, económicas y culturales (determinantes de la salud) que favorecen o limitan la situación de la salud.

- Identificar las fuentes de datos (Registros de Morbilidad Hospitalaria, Registros de Mortalidad, Registros de Enfermedades de Denuncia Obligatoria).
- Identificar los indicadores básicos de salud.
- Identificar los reportes a ser generados y graficados.
- Automatizar el procesamiento de los datos e indicadores.
- Aplicar el método de desarrollo de un datawarehouse.

3. Justificación e Importancia

El presente trabajo de investigación se enfoca en el análisis, diseño, construcción e implementación de un Datamart como herramienta de apoyo a las actividades de prevención y control de enfermedades endémicas del Centro de Investigación en Salud Pública "Dr. Jacinto Convit", específicamente se trabaja con los datos correspondientes a egreso hospitalario, los cuales son almacenados en archivos Access. En primera instancia lo que se busca es centralizar los datos en un repositorio único que garantice integridad, y estructurar dichos datos con la finalidad de transformarlos en información que permita generar conocimiento a los epidemiólogos. Una vez que ya los datos son estructurados entonces se integran herramientas de acceso a la información como reporteadores y herramientas de análisis OLAP que garantizan la disponibilidad de la información a los epidemiólogos, facilitan el análisis, y permiten automatizarla generación de los distintos reportes que forman parte de los boletines epidemiológicos.

Capítulo 2: Marco Conceptual

1. SISTEMAS DE INFORMACIÓN

1.1 Definición

Es apropiado definir conceptos que por su importancia en el tema son imprescindibles conocer. Entre ellos, el dato:

> *"El antecedente necesario para llegar al conocimiento exacto de algo o para deducir las consecuencias legítimas de un hecho" (REAL ACADEMIA ESPAÑOLA, 2001)*

En el contexto computacional, se puede entender por dato como la unidad mínima de información, la cual no ha sido procesada y que carece de cualquier valor significativo. Una vez que estos datos son recolectados, manipulados e interpretados podrán llamarse entonces información.

Por otra parte, se emplea la palabra "sistema" con bastante frecuencia en la cotidianidad. Se habla de sistemas políticos, sistemas de transporte, sistema linfático, sistema digestivo, sistema educativo, entre otros. Cada uno de ellos tiene algo en común y es que se conforman por un conjunto de elementos que relacionados entre sí, contribuyen a alcanzar un objetivo.

Es momento para definir formalmente Sistemas de Información, para (LAUDON, K & LAUDON, J, 2012) un Sistema de Información (SI):

> *"Es un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización"*(p.15).

El principal propósito de un Sistema de Información es el procesamiento de información en todas sus etapas: recolección, organización, almacenamiento, proceso y despliegue, y en todas sus formas que va desde información primaria, información procesada e interpretada hasta el conocimiento.

Los principales componentes de un SI son los siguientes:

- Datos: Unidad mínima de un SI, que se utiliza para alimentar programas y producir información.
- Hardware:

  *"Conjunto de los componentes que integran la parte material de una computadora"(*REAL ACADEMIA ESPAÑOLA, 2001).
- Software:

  "Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora" (REAL ACADEMIA ESPAÑOLA, 2001).

  Es el equipamiento lógico e intangible de un ordenador.
- Telecomunicaciones: El término telecomunicación hace referencia a todas las formas de comunicación a distancia. Es una técnica que consiste en la transmisión de un mensaje desde un punto hacia otro, usualmente con la característica adicional de ser bidireccional.
- Recurso Humano: Se define como todo el conjunto de personas que tiene relación e interacción con el Sistema de Información, valiéndose de los recursos Hardware y Software para producir, almacenar o recuperar datos.

1.2 Tipos de Sistemas Información

Los Sistemas de información se desarrollan con diferentes objetivos y sobre todo según las necesidades de la organización (Figura 2). Según LAUDONK&LAUDON, J. (2012) los SI se clasifican en:

1.2.1 Sistemas de procesamiento de transacciones (TPS por sus siglas en inglés Transactional Processing System)

Este tipo de sistema de información recolecta, almacena, modifica y recupera toda la información generada por las transacciones producidas

en una organización. Una transacción es un evento que genera o modifica los datos que se encuentran eventualmente almacenados en un sistema de información. KENDALL & KENDALL. (2011) agrega que:

> *"Son creados para procesar grandes cantidades de datos relacionadas con transacciones rutinarias de negocios, como las nóminas y los inventarios. Un TPS elimina la molestia que representa la realización de transacciones operativas necesarias y reduce el tiempo que una vez fue requerido para llevarlas a cabo de manera manual"* (p.2).

1.2.2 Sistemas en el nivel de conocimiento de la organización

Existen dos tipos de sistemas en este nivel, Los sistemas de automatización de la oficina (OAS, Office Automation Systems) que consisten en aplicaciones destinadas a ayudar al trabajo administrativo diario de una organización, forman parte de este tipo de sistemas: los procesadores de textos , las hojas de cálculo, los editores de presentaciones, los clientes de correo electrónico, entre otros.

Los sistemas de trabajo del conocimiento (KWS, Knowledge Work Systems) dan soporte a los trabajadores profesionales, tales como científicos, ingenieros y doctores, ayudándoles a crear nuevos conocimientos que contribuyan a mejorar la organización.

1.2.3 Sistemas de información gerencial

Los sistemas de información gerencial (MIS, Management Information Systems) tienen como propósito general cooperar a la correcta interacción entre los usuarios y las computadoras.

Producen información que es usada para la toma de decisiones, basándose en los sistemas de procesamiento de transacciones. En otras palabras, dan soporte a un espectro más amplio de tareas organizacionales que los sistemas de procesamiento de transacciones, incluyendo el análisis de decisiones y la toma de decisiones.

1.2.4 Sistemas de apoyo a la toma de decisiones

Los sistemas de apoyo a la toma de decisiones (DSS, Decisión Support Systems) son sistemas de información interactivos que ayudan al tomador de decisiones a utilizar datos y modelos para resolver problemas. Estos sistemas se ajustan más al gusto de la organización que los utiliza que a los sistemas de información gerencial tradicionales. En ocasiones se hace referencia a ellos como sistemas que se enfocan en la inteligencia de negocio.

1.2.5 Sistemas de soporte a ejecutivos

Los sistemas de soporte a ejecutivos (ESS) se encuentran en el nivel estratégico de la administración, ayudan a los ejecutivos a organizar sus interacciones con el ambiente externo proporcionando datos resumidos, indicadores, gráficos, entre otros. Estos se apoyan en la información generada por los TPS y los MIS y ayudan a las decisiones no estructuradas.

A continuación se ilustra (ver Figura 1) la clasificación de los sistemas de información.

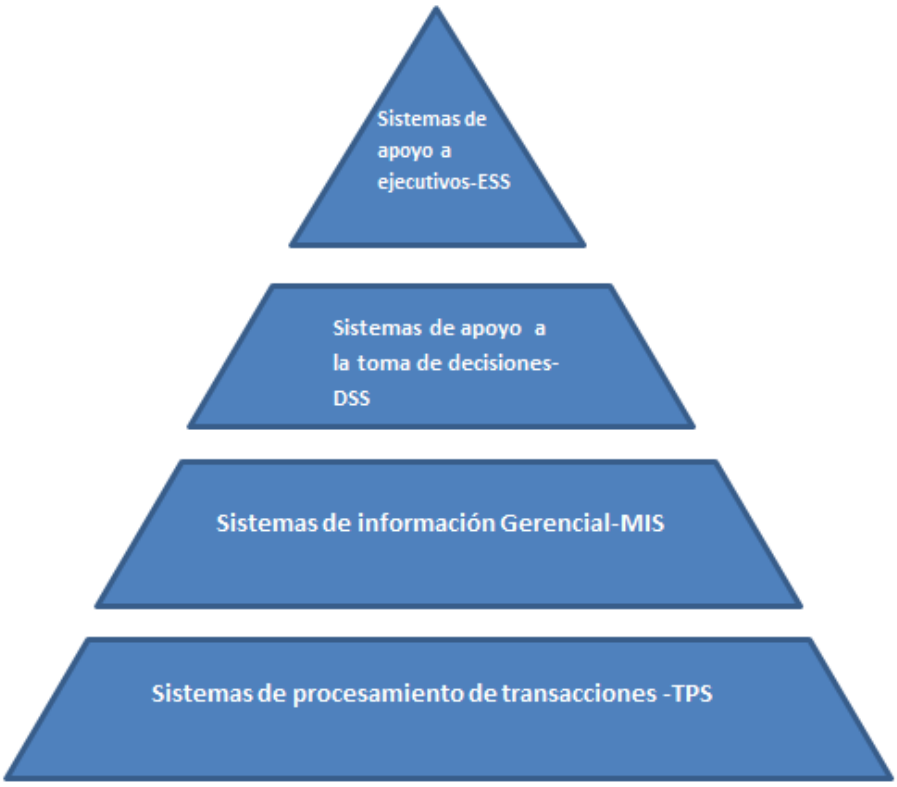


Figura1: Pirámide de los sistemas de información.

Según esta pirámide, las organizaciones se pueden dividir en 4 niveles; a medida que subimos en la pirámide, las decisiones son cada vez menos estructuradas, es decir, que son más heurísticas y menos mecánicas. Por lo tanto, la información que concierne a la toma de decisiones difiere en los distintos niveles, lo que requiere la existencia de diferentes tipos de sistemas que provean dicha información.

1.3 Sistemas OLTP

Originalmente los sistemas computarizados se fueron utilizando para automatizar las operaciones rutinarias que se llevaban a cabo en las empresas, apareciendo el término Online Transaction Processing (OLTP), con el cual se manejan las transacciones y operaciones de la empresa de una manera rápida, eficiente y precisa.(Gonzales R., año desconocido)

Una transacción genera un proceso atómico (que debe ser validado con un commit, o invalidado con un rollback), y que puede involucrar operaciones de inserción, modificación y borrado de datos. El proceso transaccional es típico de las bases de datos operacionales.

El acceso a los datos está optimizado para tareas frecuentes de lectura y escritura. (Por ejemplo, la enorme cantidad de transacciones que tienen que soportar las BD de bancos o hipermercados diariamente).

Los datos se estructuran según el nivel aplicación (programa de gestión a medida, ERP o CRM implantado, sistema de información departamental...).

Los formatos de los datos no son necesariamente uniformes en los diferentes departamentos (es común la falta de compatibilidad y la existencia de islas de datos).

El historial de datos suele limitarse a los datos actuales o recientes.

1.4 Sistemas OLAP

Según Gamboa y Berroteran (2007), la tecnología OLAP (On-line Analytical Processing) facilita el análisis de los datos en línea en un almacén de datos, proporcionando respuestas rápidas a consultas complejas. OLAP es generalmente utilizado para ayudar a la toma de decisiones y presenta los datos a los usuarios a través de un modelo de datos intuitivo y natural. Por lo tanto los usuarios finales pueden ver y entender con mayor facilidad la información de sus bases de datos.

Dicho análisis suele implicar, generalmente, la lectura de grandes cantidades de datos para llegar a extraer algún tipo de información útil: tendencias de ventas, patrones de comportamiento de los consumidores, elaboración de informes complejos, etc. Este sistema es típico de los datamarts.

El acceso a los datos suele ser de sólo lectura. La acción más común es la consulta, con muy pocas inserciones, actualizaciones o eliminaciones.

Los datos se estructuran según las áreas de negocio, y los formatos de los datos están integrados de manera uniforme en toda la organización.

El historial de datos es a largo plazo, normalmente de dos a cinco años. Las bases de datos OLAP se suelen alimentar de información procedente de los sistemas operacionales existentes, mediante un proceso de extracción, transformación y carga (ETL - Extract, Transformation and Load)

1.5 Diferencias entre Sistemas OLAP y Sistemas OLTP

En la tabla 1 se muestran las principales características entre OLAP y OLTP:

Tabla 1. Comparativa entre OLAP y OLTP

Tomado de http://l3n6iwmeza.wordpress.com/2012/02/14/tabla-comparativa-entre-olap-y-oltp/

CARACTERÍSTICA	OLAP	OLTP
Significado	Análisis de Procesamiento en Línea	Análisis de Transacciones en Línea
Objetivos	Ayudar con la planificación, resolución de problemas y apoyo a las decisiones	Controlar y ejecutar las tareas fundamentales del negocio

Alineación de Datos	Están alineados por dimensión, los datos son organizados definiendo dimensiones del negocio, se focaliza en el cumplimiento de requerimientos del análisis del negocio.	Están alineados por aplicación, se focaliza en el cumplimiento de requerimientos de una aplicación especial o una tarea específica.
Integración de Datos	Los datos deben ser integrados, son conocidos como datos derivados o DSS.	Los datos no están integrados, son calificados como datos primitivos u operacionales, son estructurados independientemente uno de otros, son almacenados en diferentes formatos de archivos, pueden residir en diferentes plataformas de hardware o RDBMS

Historia	Como una cuestión de hecho, el origen de la tecnología OLAP puede rastrearse camino de vuelta en 1962. No fue sino hasta 1993 que el término OLAP fue acuñado en el documento de Codd blanco escrito por el investigador de la base de datos en alta estima Ted Codd, que también estableció las 12 reglas para un producto OLAP.	Retienen datos para 60 o 90 días después son resguardados por administradores de B.D en almacenamientos secundarios.
Acceso y Manipulación de Datos	Tienen una carga y acceso masivo de datos, la carga y refresco es batch (bulkcopy), la validación de datos se realiza antes o después de la carga, se realizan sentencias de Select sobre varios registros y tablas.	Realizan manipulación de datos registro por registro con inserts, updates y deletes, necesitan rutinas de validación y transacciones a nivel de registro.

En general, podemos suponer que los sistemas OLTP deben proporcionar los datos de origen a los almacenes de datos, mientras que los sistemas OLAP ayudan a analizar.

2. INTELIGENCIA DE NEGOCIO

2.1 Definición

Inteligencia de Negocio (BI, Business Intelligence) pretende dar una visión de información y conocimiento en las organizaciones.
Para definir Business Intelligence se partirá de la definición del glosario de términos de Gartner:

> *"Es una variedad de aplicaciones y tecnologías para obtener, almacenar, analizar, compartir y proveer acceso a los datos para ayudar a los usuarios de las empresas a tomar mejores decisiones."*

Tomando como base la definición anterior y después de haber analizado el significado de Business Intelligence, se cita una definición más general y desde un punto de vista más práctico:

Inteligencia de Negocio es un proceso, el cual plantea soluciones de negocio, utilizando un conjunto de metodologías, aplicaciones y tecnología, capaz de transformar los datos e información desestructurada, en información estructurada, generando conocimiento para soportar la toma de decisiones en beneficio del negocio y de la organización.

2.2 Antecedentes

La inteligencia de negocio no es un tema nuevo, es algo que se viene tratando desde los años 50, cuando Hans Peter Luhn científico de IBM, escribió un artículo donde hablaba de un sistema dedicado a la Inteligencia de Negocio. En dicho artículo describe la Inteligencia como la habilidad para percibir la interrelación de los hechos presentados de manera tal que orienten la acción hacia una meta deseada.

Para los años 60, se trabajaba con la información almacenada en archivadores, existían miles de carpetas, con datos de todo tipo, y en muchas ocasiones se perdía información importante, manejar esta información para aquel entonces era una tarea compleja. Para esta década, con la aparición del computador, también surgieron las bases de

datos creadas por Edgar Frank Codd lo cual cambio el concepto de almacenar la información en carpetas físicas para hacerlo ahora en el computador.

A principio de los años 70, varias empresas empezaron a crear aplicaciones empresariales empleando esas bases de datos que muchas empresas ya habían comenzado a utilizar. Además, las aplicaciones permitieron realizar "dataentry" en los sistemas aumentando la información disponible pero no tenían un fácil y rápido acceso a esa información.

Luego para la de década de los 80 por primera vez aparece el término DataWarehouse desarrollado por Ralph Kimball y Bill Inmon y los primeros sistemas (SAP) que podían generar reportes para el usuario, pero seguía siendo algo difícil de utilizar por parte de los usuarios finales y surgió un nuevo problema, potentes sistemas de bases de datos sin aplicativos que permitieran una fácil explotación de las mismas.

Luego, en 1989, Howard Dresner utiliza el término Inteligencia de Negocio para describir que son un conjunto de métodos y procesos para mejorar la toma de decisiones de las organizaciones con el uso de sistemas de apoyo basado en hechos.

No es hasta finales de la década de los 90s en donde el uso de las herramientas de Inteligencia de Negocio comienza su popularización. Existían múltiples aplicaciones para acceder a la información y el uso de los almacenes de datos estaba centrado en el soporte de los datos.

Finalmente, a partir del año 2000 y hasta la actualidad, las aplicaciones de Inteligencia de Negocio se han consolidado en pocas plataformas como Oracle, SAP, IBM y Microsoft. Además, el uso de los almacenes de datos se utiliza para proveer datos y facilitar su análisis más que para el soporte de los mismos.

2.3 Arquitectura

La Inteligencia de Negocio incorpora las mejores prácticas tecnológicas, orientadas a entregar una solución completa y eficiente.

A continuación se muestra en la Figura 2, la Arquitectura de una solución de Business Intelligence.

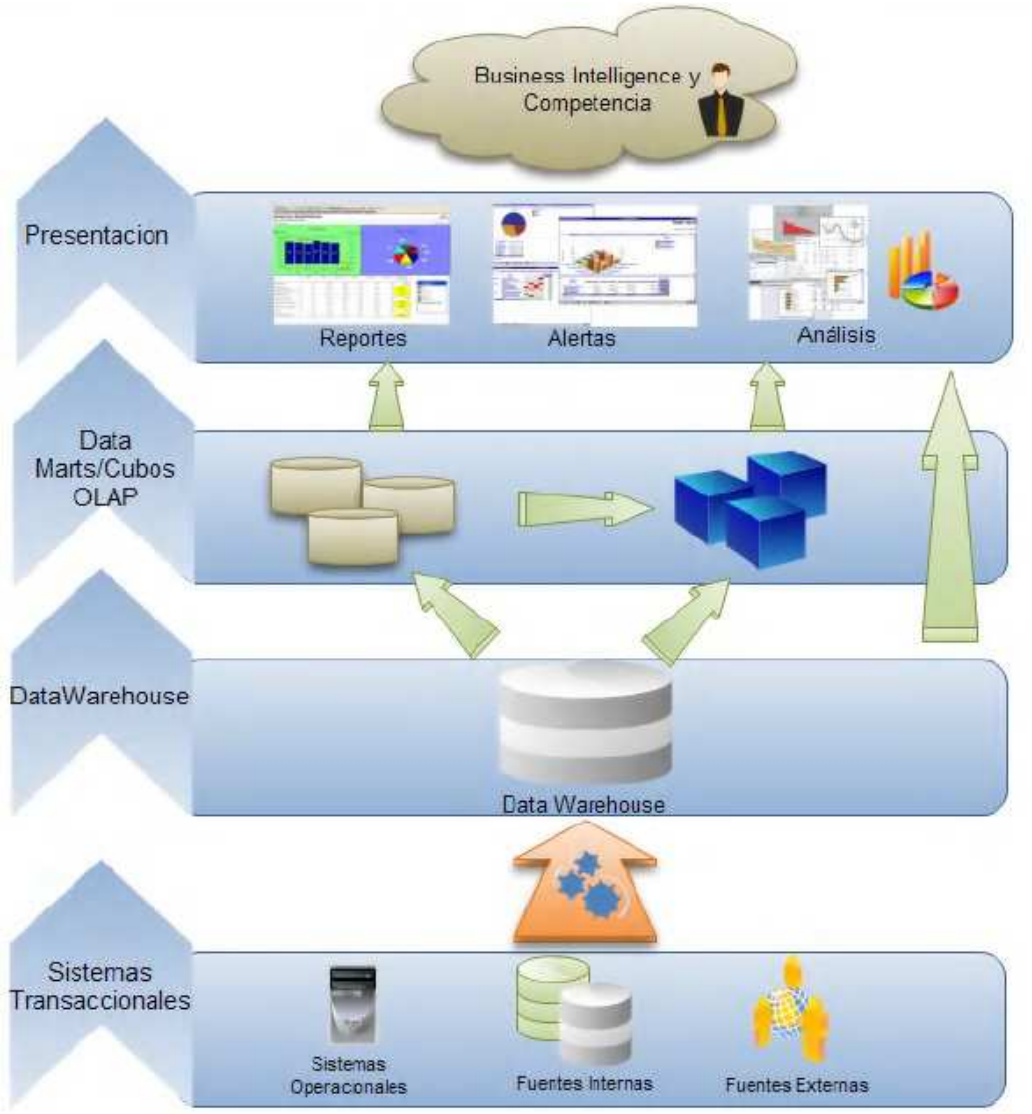


Figura 2: Arquitectura de BI.

Como se observa en la figura anterior una solución de Inteligencia de Negocio inicia desde los sistemas de origen de una organización (bases de datos, ERPs, archivos de texto...), sobre los que suele ser necesario aplicar una transformación estructural para optimizar su proceso analítico, posteriormente estos datos son almacenados, después pueden ser estructurados específicamente y finalmente ser presentados de diferente manera.

3. DATA WAREHOUSE (DW)

3.1Definición

Según Ralph Kimball y Bill Inmon un Data Warehouse es:

"Una copia de la transacción de datos específicamente estructurado para consulta y análisis." (Ralph Kimball, 1996)

"Orientado al tema, integrado, de tiempo variante, de colección de datos no volátil en apoyo a la gestión del proceso de toma de decisiones." (Bill Inmon, 1995)

El Data Warehouse organiza y aloja los datos necesarios, para ser utilizados en el procesamiento analítico dentro de una perspectiva de tiempo.

Un Data Warehouse es una base de datos accesible por los usuarios el cual tiene un registro de datos históricos y actuales acerca de todas las entidades importantes que se encuentran en la empresa y de acuerdo a negocios específicos.

3.2 Características

A continuación se muestran las características principales de un Data Warehouse en cuanto a sus datos.

- Orientado al tema.
- Integrado.

- De tiempo Variante.
- No volátil.

Orientado al tema: Se debe esta característica debido a que en el DataWarehouse la información se clasifica de acuerdo a aspectos de interés para la empresa, como por ejemplo cliente, vendedor, producto, venta.

Integrado: En el Data Warehouse la información se encuentra integrada, esta integración puede ser vista a problemas de datos con que se puede encontrar en una base de datos operacional, errores como de inconsistencia de datos, uniformidad, diferente codificación de datos en múltiples fuentes. Con la integración, cualquier tipo de dato será estandarizado de manera general y así será alojado en el almacén.

De tiempo variante: Se refiere así debido al horizonte de tiempo con que funciona un DataWarehouse.

No Volátil: Se debe a que el DataWarehouse no sufre las operaciones como inserción, eliminación, modificación, tan solo realizan dos operaciones que son la carga de datos y el acceso a los mismos

Un DataWarehouse como producto presenta las siguientes características:

- Fácil accesibilidad a la información organizacional.
- Información sumarizada y detallada.
- Presentación consistente de la información organizacional.
- Permite realizar análisis rápidamente.

3.3 Arquitectura

A continuación, se muestra en la Figura 3la Arquitectura de una solución de Business Intelligence.

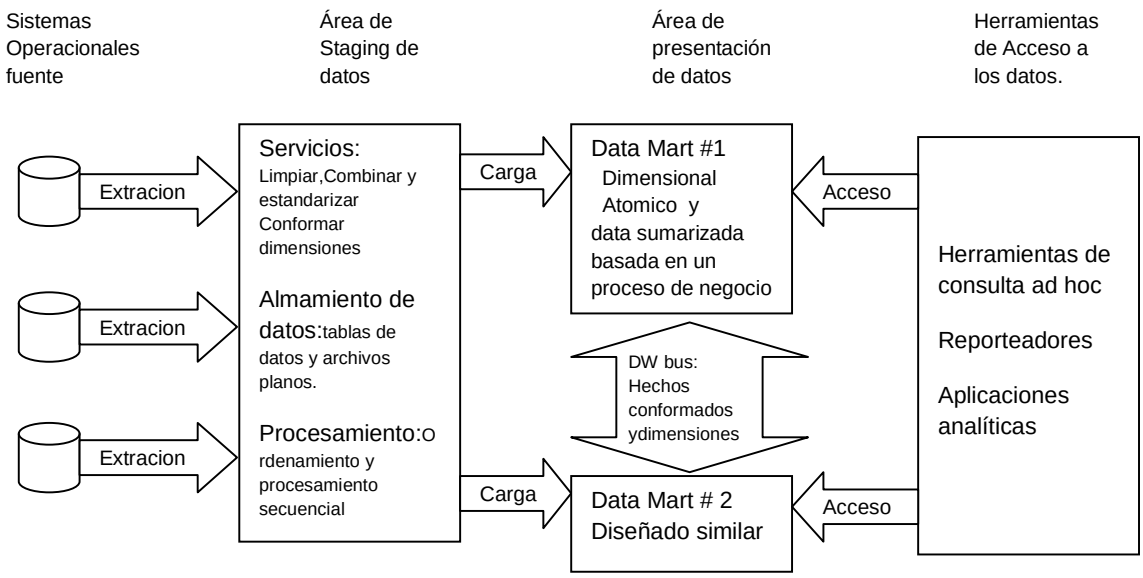


Figura 3:Elementos Básicos de un Data Warehouse.

En la figura anterior se aprecia la arquitectura de un Data Warehouse, la cual se basa en la arquitectura de Bussiness Intelligence (Figura 2). Inicia con los sistemas origen, de donde se extrae datos, posteriormente se realiza un proceso de estandarización de la data a través de un proceso ETL para obtener datos de calidad. Luego estos datos son almacenados en un Data warehouse, el cual se puede estructurar en varios datamarts orientados a un área de negocio de la organización. Finalmente el DataWarehouse es accedido a través de diversas herramientas, como aplicaciones analíticas, de reportes, minería de datos, etc. Las cuales representan la capa de presentación con la cual el usuario final puede interactuar.

3.3.1Sistema Origen

En cuanto a los Sistemas Origen, son en donde se encuentra los datos de interés y que serán objeto de extracción, estos orígenes pueden ser desde múltiples lugares como por ejemplo las bases de datos, así como también hojas de cálculo, archivos planos, Sistemas ERP (Enterprise

Resource Planning), entre otros que serán los que proveerán los datos de interés.

3.3.2 StagingArea (Almacenamiento Temporal)

Es un área en la que se almacenará de manera temporal todos los datos que son extraídos de los sistemas origen según las necesidades planteadas del negocio. La función primordial es minimizar la afectación a los sistemas origen, porque cuando se realiza la carga al Data Warehouse, los orígenes de datos no intervendrán hasta que se realice una próxima carga.

3.3.3 ODS (Operational Data Store)

ODS o Almacén de datos operativos es un área que da soporte a los sistemas operacionales. El modelo de datos del ODS sigue una estructura relacional y normalizada, característica por la cual se la trata de diferente manera a la de un DW, y con lo que le permite que cualquier herramienta de reporte o sistema operacional pueda consultar sus datos. El ODS forma parte de un Data Warehouse, en el sentido de que el Almacén de Datos aprovecha los datos que previamente se encuentran integrados en él, permitiendo también dar soporte a todas las transacciones operacionales.

Los ODS se lo puede considerar como un sistema origen desde el punto de vista que va a ser sujeto a la extracción de los datos que serán cargados en el Data Warehouse. Si el Almacén no es demasiado grande, o el nivel de exigencia no es muy elevado en cuanto a los requerimientos o consultas operacionales, se puede prescindir de la intervención del ODS.

Los datos a ser cargados en el Almacén provendrán del Staging Área así como también puede ser desde el ODS. Aquí los modelos de datos no serán tan normalizados como lo es en los sistemas origen y en el ODS, aquí sucede lo contrario y se realizan técnicas de desnormalización.

3.3.4Datamart

Son considerados pequeños almacenes ya que poseen información específica que se obtiene desde el Data Warehouse, un Datamart es más personal ya que puede llegar a construirse a partir de las necesidades en particular de un usuario o a un tema en específico, así como también por ejemplo los cubos OLAP sobre cierta información que el usuario requiera, dándole una perspectiva analítica sobre los datos.

3.4Proceso de ETL (Extracción, Transformación y Carga)

Existe un conjunto básico de procesos de suma importancia detrás de una arquitectura de DW, que garantizan la calidad de datos que en ella se almacenarán.

Este proceso de Extracción, Transformación y carga, también conocido como ETL o proceso de integración de datos, cumple con la función principal de organizar e integrar el flujo de datos desde múltiples fuentes, hacia un destino que es el almacén de datos (DataWarehouse). El proceso ETL brinda soporte a la Gestión de Datos que se va a realizar, obteniendo calidad de los mismos dentro de un almacén.

Todo el proceso que se lleva a cabo, se debe especificar los tiempos en los cuales se deberá realizar el mismo, lo que garantiza que se mantenga al día los datos en el almacén. Aunque esto va a ser definido de acuerdo a las necesidades de la Organización, ya que por ejemplo se puede definir cargas diarias, así como semanales o mensuales.

Este proceso general se encuentra subdividido en 3 subprocesos fundamentales como se detalla a continuación.

3.4.1 Extracción

La extracción se refiere a la adquisición de los datos, los cuales pueden ser recogidos de diferentes fuentes, como archivos planos, hojas de cálculo ó bases de datos.

La extracción de los datos se almacenarán en una área temporalmente o Staging Area, vale recalcar que solo se extraerán datos necesarios, es decir, de acuerdo a lo que se haya especificado en los requerimientos, ya que en el ambiente transaccional se encuentra gran cantidad innecesaria de datos, por lo que es indispensable la extracción de los mismos, y que serán útiles en el ambiente del DataWarehouse.

3.4.2 Transformación

Es el subproceso más laborioso con respecto a los otros dos, debido a que en esta etapa se realiza el refinamiento de los datos que han sido extraídos de las diferentes fuentes, por lo que aquí se especificará pasos de acuerdo a los datos que van a ser tratados dando valor para los usuarios. Este proceso incluye corrección de errores, decodificación, borrado de campos que no son de interés, generación de claves, agregación de información, etcétera, lo que es más conocido como limpieza de los datos fuentes.

3.4.3 Carga

El último subproceso se caracteriza por realizar la carga hacia el Data Warehouse, los datos que previamente han sido extraídos y tratados en los dos subprocesos anteriores para contar con datos de calidad, ahora se procederá a realizar la carga de los mismos a un nuevo ambiente que es el de almacén de datos, para ello es importante implementar métodos y/o maneras de carga de datos con el fin de controlar por ejemplo datos actualizados o históricos.

Finalizado todo el proceso ETL, lo que se pretende es contar con datos relevantes para el negocio, los mismos que deben ser de valor sin ningún tipo de codificación, es decir datos transparentes y entendibles por los usuarios finales. Ya contado con la calidad de datos en el ambiente del almacén se termina un ciclo del proceso ETL.

3.5 Modelo Multidimensional de un Data Warehouse

Un DataWarehouse adopta un Modelo Dimensional en su estructura de almacenamiento, caracterizado por ser un esquema en estrella o copo de nieve, lo que permite maximizar el rendimiento de las consultas.

Según Wolff(2002) un modelo dimensional:

> *"Es una técnica para modelar bases de datos simples y entendibles al usuario final. La idea fundamental es que el usuario visualice fácilmente la relación que existe entre los distintos componentes del negocio"*

Un diseño dimensional en estrella o copo de nieve es muy diferente del diseño de un esquema de base de datos operacional, en este último, los datos están altamente normalizados para soportar frecuentes actualizaciones y para mantener la integridad referencial, en cambio en un diseño de Data Warehouse, los datos están desnormalizados o redundantes para proporcionar acceso inmediato a los datos sin tener que realizar una gran cantidad de relaciones.

3.5.1 Esquema Estrella

Es la técnica de diseño más popular usada para un Data Warehouse. Es un paradigma en el cual un único objeto en el centro (conocido como tabla de hecho) se encuentra conectado radialmente con otros objetos circundantes llamados tabla de dimensiones formando una estrella.

Este tipo de esquema es la más utilizada en un ambiente de Data Warehouse debido a que los datos se encuentran desnormalizados y por ende las consultas que se pueden realizar son menos complejas ya que no existe la necesidad de realizar muchas relaciones entre tablas.

A continuación, en la figura 4 se representa gráficamente un esquema estrella.

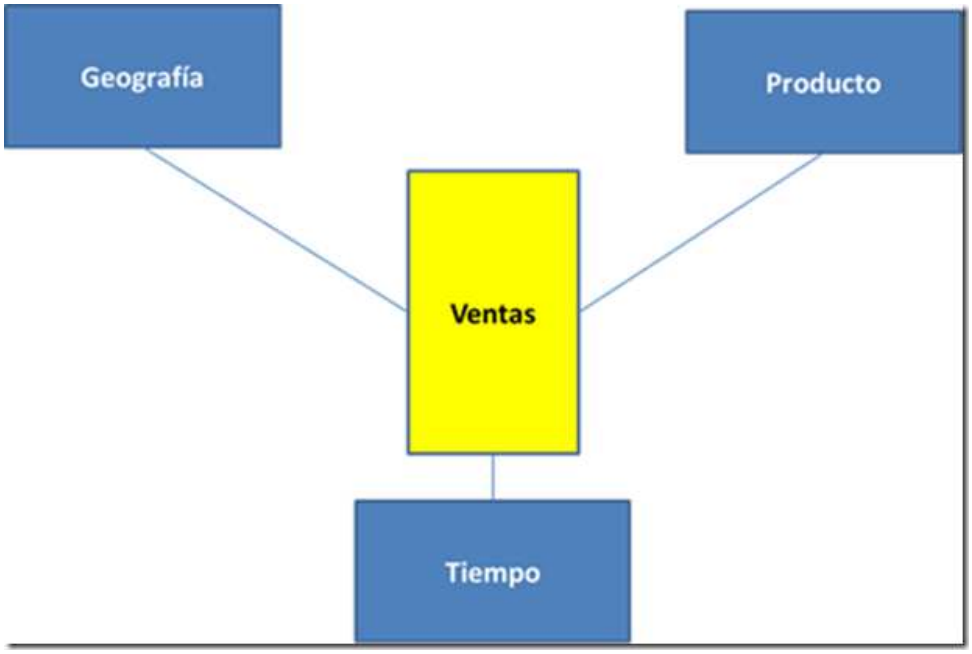


Figura4: Ejemplo de esquema estrella.

En la figura anterior se visualiza un esquema estrella básica, conformado por 3 dimensiones (Geografía, Producto y Tiempo), que están conectadas de forma circundante a un único objeto central denominado tabla de hechos, representado por ventas. Dichos elementos en conjunto dan la percepción de una estrella.

3.5.2 Esquema Copo de Nieve

El esquema copo de nieve es una extensión del esquema estrella, donde cada punta se explota en más puntas y su denominación se debe a que el diagrama se asemeja a un copo de nieve.

En el esquema copo de nieve se normalizan dimensiones para eliminar redundancia, permitiendo que los datos de las dimensiones se agrupen en múltiples tablas en lugar de una tabla grande.

A continuación se aprecia en la figura 5, un esquema copo de nieve.

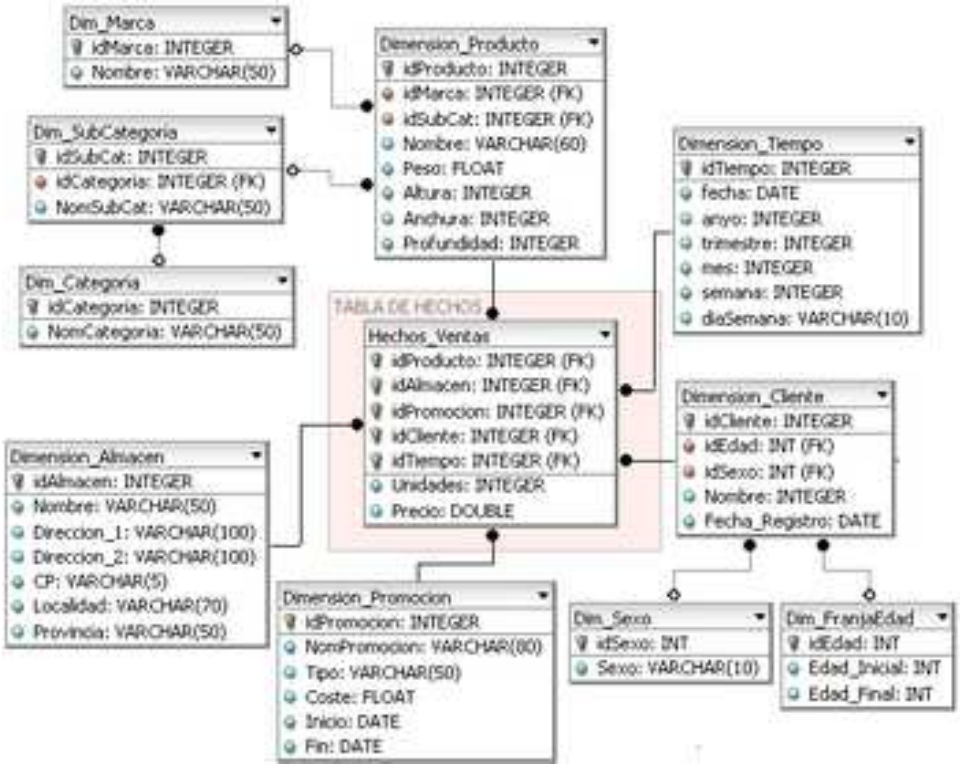


Figura5: Ejemplo de copo de nieve.

En la figura anterior, se observa como la dimensión producto se normaliza para el caso de la marca del producto, con esto se busca eliminar la redundancia que se genera al almacenar el nombre de la marca como atributo de la dimensión producto, es por ello, que se crea una nueva dimensión (Dim_Marca), que contiene todas las marcas existentes de manera univoca y en la dimensión producto se almacena la clave foránea idMarca que representa la relación entre ambas dimensiones.

3.5.3 Tablas

3.5.3.1 Hechos

Tabla de hechos (fact table), es la tabla central de un esquema dimensional (en estrella o en copo de nieve), que contiene los valores de las medidas de negocio a ser analizadas en un modelo de datos tipo

estrella. Este tipo de tabla representa el hecho o actividad del negocio, como por ejemplo, Ventas, Movimientos, Pedidos, etc. Por lo general estos datos son numéricos y pueden agruparse (agregación) en un valor total, las medidas pueden ser por ejemplo, cantidad vendida, costo, precio unitario, etc. Es decir son los indicadores que permitirán medir los hechos que se realizan en el negocio.

Una tabla de hecho está dividida por 2 tipos de Atributos, estos son: las claves foráneas provenientes de las Dimensiones, y por los indicadores o medidas del Hecho.

En la figura 6, se representa gráficamente una tabla de hechos.

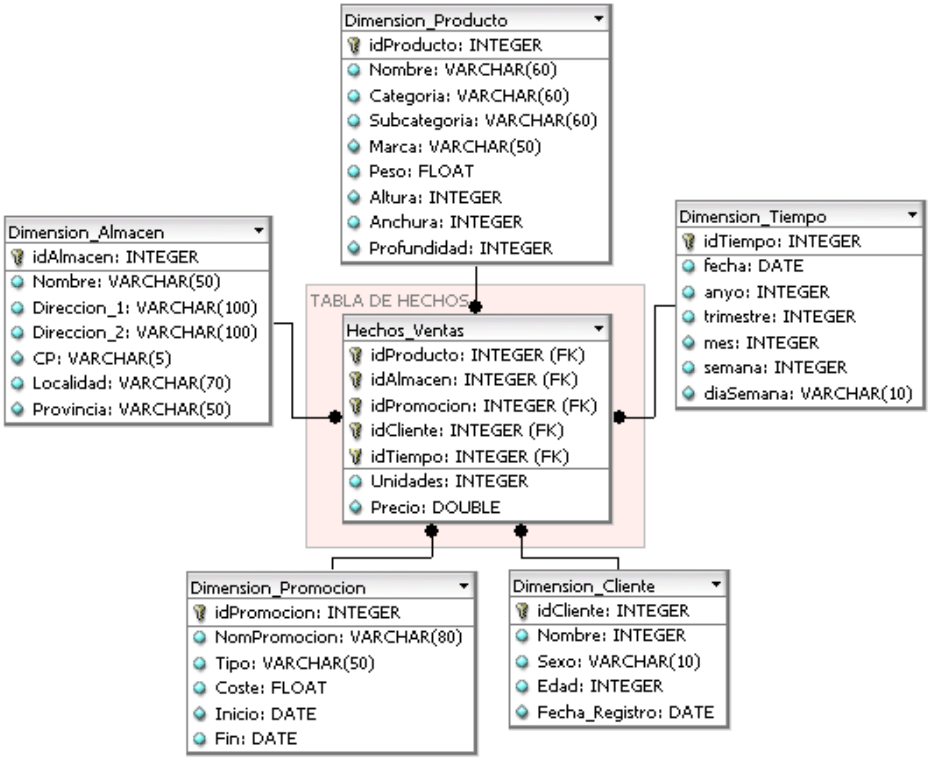


Figura6.Ejemplo de tabla de hecho

En la figura anterior, se aprecia en el centro la tabla de hechos Hechos_Ventas,que contiene los valores de las medidas de negocio a ser analizadas, en este ejemplo las medidas de negocio son representadas por los atributos Unidades y Precio. Mientras que los atributos idTiempo,

idCliente, idPromocion, idAlmacen, idProducto son claves foráneas provenientes de las dimensiones.

3.5.3.2Dimensiones

Son tablas que describen a la tabla de hecho, mediante atributos descriptores que poseen de acuerdo a un tema específico del negocio como por ejemplo: Clientes, Productos, Ubicación Geográfica, etc. Esto se puede apreciar de mejor manera a continuación.

Al igual que la tabla de hecho, en la dimensión se cuenta con 2 tipos de Atributos estos son: la clave primaria, única para cada registro, y de los atributos descriptores de ellas.

Los atributos por lo general son campos textuales o descriptores numéricos cortos.

A continuación se representa gráficamente una dimensión en la figura 7.

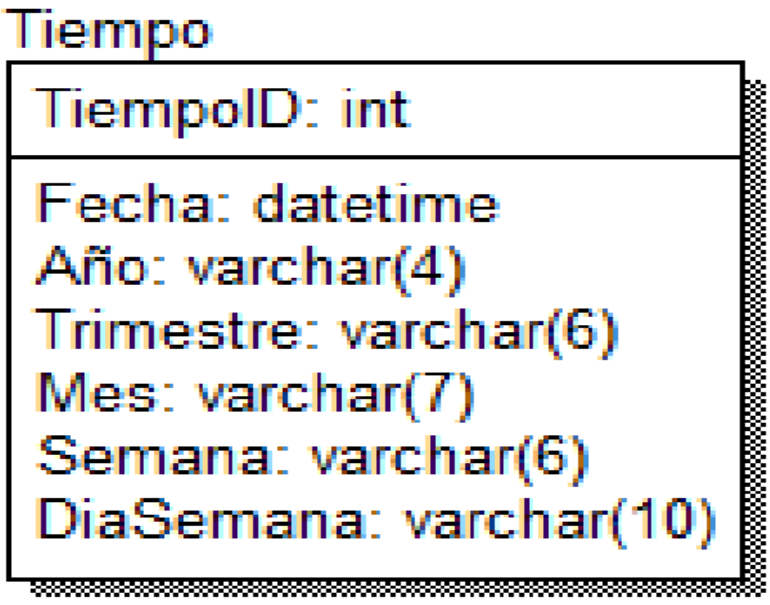


Figura 7: Ejemplo dimensión fecha.

En la figura anterior, se observa la dimensión Tiempo que describe una tabla de hecho mediante atributos descriptores como lo son Fecha, año, trimestre, mes, semana, día semana. Con esto se persigue brindar diferentes perspectivas de análisis a los tomadores de decisiones.

3.5.4 Granularidad

La granularidad es el nivel de detalle en que se almacena la información.

A mayor nivel de detalle, mayor posibilidad analítica, ya que los mismos podrán ser resumidos o sumarizados. Los datos con granularidad fina (nivel de detalle) podrán ser resumidos hasta obtener una granularidad media o gruesa. No sucede lo mismo en sentido contrario. La granularidad está centrada en las medidas es decir en la tabla de hecho.

Por ejemplo, no es lo mismo contar el tiempo por horas (grano fino) que por semanas (grano grueso); o en el caso de los productos, se puede considerar cada variante de un mismo artículo como un producto (por ejemplo, en una empresa textil, cada talla y color de pantalón podría ser un producto) o agrupar todos los artículos de una misma familia considerándolos como un único producto (por ejemplo, el producto pantalón genérico).

Como se puede observar, la granularidad afecta a la cardinalidad, tanto de las dimensiones como de la tabla de hechos, a mayor granularidad (grano más fino) mayor será el número de registros final de la tabla de hechos.

Cuando la granularidad es mayor, es frecuente que se desee disponer de subtotales parciales, es decir, si tenemos una tabla de hechos con las ventas por días, podría interesar disponer de los totales semanales o mensuales, estos datos se pueden calcular haciendo sumas parciales, pero es frecuente añadir a la tabla de hechos registros donde se almacenan dichos cálculos para no tener que repetirlos cada vez que se

requieran y mejorar así el rendimiento de la aplicación. En este caso se dispondrá en la misma tabla de hechos de datos de grano fino y de grano más grueso aumentando aún más la cardinalidad de la tabla.

3.6Explotacióndel Data Warehouse

La explotación de un Data Warehouse, es prácticamente en donde se puede identificar los resultados que puede llegar a brindar una implementación de este tipo, ya que como se ha podido identificar un almacén de datos no es el fin en sí de una solución de BI, y es solamente el medio para dar como resultado a la Inteligencia de Negocio, objetivo que se lleva a cabo con la adecuada explotación de los datos almacenados en el Data Warehouse. Es decir mediante una interfaz gráfica se puede apreciar y monitorear el Negocio, como por ejemplo revisar cómo van las ventas en diferentes sucursales, o analizar cuál es el producto más vendido, o cantidad de artículos a comprar, realizadas por la organización. Entre los medios más utilizados para aprovechar los datos de un Data Warehouse, se citarán los siguientes:

3.6.1 Reportes y Consultas

Dentro de toda Organización, es de suma importancia los reportes que se pueden obtener acerca de lo que está pasando con el negocio a la cual esté especificada, así como también acerca de consultas que los usuarios y empleados, como también directivos, desean realizar sobre algo en específico lo que es también conocido como Consulta ad-hoc.

Las consultas o informes libres trabajan tanto sobre el detalle como sobre las agregaciones de la información.

Realizar este tipo de explotación en un almacén de datos supone una optimización del tradicional entorno de informes (reporting), dado que el Data Warehouse mantiene una estructura y una tecnología mucho más apropiada para este tipo de solicitudes.

Los sistemas de "Consultas & Reportes", no basados en almacenes de datos se caracterizan por la complejidad de las consultas, los altísimos

tiempos de respuesta y la interferencia con otros procesos informáticos que compartan su entorno, esto es debido a la alta normalización de sus datos.

3.6.2 Análisis OLAP

OLAP (On-Line Analytical Processing) o Procesamiento Analítico en Línea, surge como contraste a OLTP (On-Line Transactional Processing) que define a los sistemas de ambientes transaccionales.

El análisis multidimensional (Análisis OLAP), parte de una visión de la información como dimensiones de negocio, en la que hay que tomar en cuenta que se debe de olvidar lo que son las tablas y campos, dando mayor énfasis a lo que son las dimensiones y medidas.

Las herramientas OLAP ofrecen a los usuarios, acceso a los almacenes de datos (Data Warehouse) para que puedan obtener las funcionalidades de análisis avanzados que requieren los usuarios finales y analistas, por lo que es considerada la herramienta ideal para sacar el mayor provecho a los datos almacenados en un repositorio.

Las herramientas OLAP se caracterizan por subdividirse en 3 tipos de acuerdo a la manera de almacenar los datos, estos son:

3.6.2.1 ROLAP (Relational OLAP)

Arquitectura en la que se almacenan los datos en un motor de base de datos relacional, pero de igual manera se proporciona la funcionalidad analítica. A través de esta implementación se soporta de mejor manera las capacidades OLAP con respecto a las bases de datos relacionales, en el sentido que realiza consultas directas a la base de datos, e igualmente presenta los datos de la manera multidimensional caracterizada por la Arquitectura. Los esquemas más comunes sobre los que se trabaja son estrella o copo de nieve. La arquitectura está compuesta por un servidor de datos relacional y el motor OLAP.

3.6.2.2 MOLAP (Multidimentional OLAP)

En este tipo de Arquitectura los datos se almacenan de manera dimensional en un servidor de base de datos multidimensional, permitiendo optimizar los tiempos de respuesta en la información, ya que al ser sumarizada y/o agregada ayuda mucho a los datos calculados por adelantado como por ejemplo los totales, lo que aumenta el desempeño de análisis.

3.6.2.3 HOLAP (Hybrid OLAP)

La arquitectura OLAP híbrida (HOLAP), se caracteriza por combinar las arquitecturas ROLAP y MOLAP para brindar una solución con las mejores características de ambas: desempeño superior y gran escalabilidad. Un tipo de HOLAP mantiene los registros de detalle (los volúmenes más grandes) en la base de datos relacional, mientras que mantiene las agregaciones en un almacén MOLAP separado.

4. Indicadores

Para lograr una gestión más eficiente en las instituciones, es necesario aplicar una serie de transformaciones importantes en el funcionamiento de las mismas. Para ello, se requiere, entre otras cosas, el desarrollo de nuevos liderazgos que impulsen el cambio, incorporar técnicas modernas de gestión y establecer metas cuantificables de desempeño.

Algunos mecanismos que contribuyen al desarrollo de una gestión de calidad, son las mediciones y evaluaciones de los servicios o productos que provee cada unidad u organización, a través de un conjunto de indicadores claves.

Existen sistemas que permiten definir los indicadores claves para las instituciones, así como la medición de los mismos, por lo tanto las diversas instancias de la organización pueden contar con bases

sustentables de información para la toma de decisiones, que ayudarán amejorar su desempeño.

Los indicadores se definen como fórmulas matemáticas y/o estadísticas que se aplican a los volúmenes de los valores y pueden ser de diversa índole(Mozarrain,2000), para ello se definirán cada uno de ellos.

4.1 Tipos de Indicadores

Según Mozarrain (2000) los indicadores se clasifican en:

- Indicadores de cumplimiento

Dado que la palabra cumplir implica la culminación de una tarea, se puede decir que los indicadores de cumplimiento se refieren a los tiempos que indican el grado de consecución de las tareas.

- Indicadores de evaluación

El término evaluación se refiere al rendimiento que se obtiene de una tarea, trabajo o proceso. Los indicadores de evaluación se relacionan íntimamente con los mecanismos que ayudan a la identificación de las fortalezas, oportunidades de mejora y debilidades de las instituciones.

- Indicadores de eficiencia

Dado que el término eficiencia implica realizar un trabajo o una tarea en el menor tiempo posible, se podría inferir que los indicadores de eficiencia están relacionados con las expresiones matemáticas que indican el tiempo invertido en la realización consecuente de las mismas.

- Indicadores de eficacia

El concepto de eficacia se refiere al grado de cumplimiento de los objetivos planteados, es decir, en qué medida el área, o la institución como un todo, está cumpliendo con sus objetivos fundamentales, sin considerar necesariamente los recursos asignados para ello.

- Indicadores de gestión

La gestión está íntimamente relacionada con la administración y el establecimiento de acciones para concretar o llevar a cabo las tareas o trabajos planificados y programados. Los indicadores de gestión hacen referencia a las medidas que permiten administrar realmente un proceso.

4.2 Indicadores Epidemiológicos

Los indicadores de salud pueden ser divididos en dos amplias categorías (EPI-CENTRO, 2003):

Los indicadores epidemiológicos se utilizan para estimar la magnitud y trascendencia de una situación determinada. Siempre deben estar referidos a: La población a partir de la cual se calculan, el periodo de tiempo que representan, el lugar geográfico del cual proviene la información (variables de persona, de tiempo y de lugar). Se puede medir el impacto o los efectos de los programas de salud comparando un mismo indicador epidemiológico antes y después de la ejecución de las actividades de un programa determinado. Estos indicadores tienen gran utilidad en la etapa de formulación diagnóstica y en la de evaluación del programa. La evaluación es en cierto modo, un diagnóstico de situación actualizado.

Los indicadores operacionales miden el trabajo realizado, ya sea en función de la cantidad o de la calidad de él. Miden la cantidad de actividades y procedimientos realizados, en relación con metas o estándares establecidos previamente.

5. Vigilancia Epidemiológica

En términos prácticos, la vigilancia se entiende como la observación sistemática y continuada de la frecuencia, la distribución y los determinantes de los eventos de salud y sus tendencias en la población. Todo sistema de vigilancia debe estar amparado por un marco legal propio del Estado que garantice la operación eficiente de dicho sistema.

Este concepto tiene dos componentes prácticos:

- La medición sistemática de problemas prioritarios de salud en la población, el registro y la transmisión de datos.
- La comparación e interpretación de datos con el fin de detectar posibles cambios en el estado de salud de la población y su ambiente.

Esta definición destaca tres características de la vigilancia: i) es un proceso continuo y sistemático, es decir, no es una actividad aislada en el tiempo, ni se puede ejecutar sin métodos; ii) es un proceso de escrutinio de tendencias; y, iii) es un proceso de comparación, entre lo que se observa y lo que se espera, para detectar o anticipar cambios en la frecuencia, distribución o determinantes de la enfermedad en la población.

5.1 Objetivos de la Vigilancia

- Detectar cambios agudos en la ocurrencia y distribución de las enfermedades.
- Identificar, cuantificar y monitorear las tendencias y patrones del proceso salud-enfermedad en las
- poblaciones.
- Observar los cambios en los patrones de ocurrencia de los agentes y huéspedes para la presencia de
- enfermedades.
- Detectar cambios en las practicas de salud.
- Investigar y controlar las enfermedades.
- Planear los programas de salud.
- Evaluar las medidas de prevención y control

5.2 Etapas básicas de los sistemas de vigilancia

1. Análisis de datos

El análisis involucra principalmente un proceso de descripción y comparación de datos con relación a características y atributos de tiempo, lugar y persona, así como entre los diferentes niveles organizativos del sistema de salud y tiene el propósito de:

- Establecer las tendencias de la enfermedad a fin de detectar y anticipar la ocurrencia de cambios en su comportamiento.
- Sugerir los factores asociados con el posible incremento o descenso de casos y/o defunciones e identificar los grupos sujetos a mayor riesgo.
- Identificar las áreas geográficas que requieren medidas de control.

Tiempo

La distribución de los casos en el tiempo permite el establecimiento de hipótesis acerca del comportamiento de una enfermedad. En general son de principal interés, tres tipos de tendencias de enfermedad:

- Secular. patrón de variación (regular o no) o comportamiento general por largos periodos de tiempo).
- Estacional patrón regular de variación entre estaciones del año.

Lugar

Los datos de la vigilancia también pueden ser analizados o comparados según el lugar en que ocurrieron. Un buen apoyo es la descripción gráfica de las notificaciones (mapeo) según espacios y población, especialmente a nivel local. El uso de sistemas de información geográfica (SIG) no solamente puede mejorar la descripción gráfica de los eventos bajo vigilancia con relación a la variable lugar, sino también el análisis geoespacial de dichos eventos y la identificación de conglomerados y brotes. Se debe tratar de localizar el lugar en el que se originó la enfermedad así como el lugar en el que se encontraba el paciente al momento de detección de la enfermedad. Al igual que para el análisis en el tiempo, es importante utilizar tasas, ya que un alto número

de casos puede deberse a un tamaño poblacional mayor y no necesariamente a una alta incidencia o riesgo.

El análisis epidemiológico de los datos de vigilancia se orienta a la identificación de un aparente exceso en la ocurrencia o el riesgo de ciertas exposiciones, enfermedades o muerte con relación a un grupo de personas, un periodo en el tiempo o un área geográfica específica.

Persona

El análisis de los datos de vigilancia por las características de las personas afectadas es valioso para identificar los grupos de riesgo. La mayoría de los sistemas de vigilancia proporcionan información por edad y sexo. Otras variables utilizadas o que pueden estar disponibles son: nacionalidad, nivel de inmunidad, nutrición, estilos de vida, escolaridad, área de trabajo, hospitalización, factores de riesgo y nivel socio económico.

2. Curva epidémica

Para la identificación de una epidemia es necesario conocer la frecuencia precedente de la enfermedad. Una de las maneras más simples y útiles es construir una curva epidémica, que consiste en la representación gráfica de las frecuencias diarias, semanales o mensuales de la enfermedad en un eje de coordenadas, en el cual el eje horizontal representa el tiempo y el vertical las frecuencias. Las frecuencias pueden expresarse en números absolutos o en tasas y el tiempo puede corresponder a días, semanas, meses o años. El gráfico puede ser un histograma.

La curva epidémica tiene usualmente distribución asimétrica y presenta los siguientes elementos:

- La curva ascendente, que representa la fase de crecimiento de la epidemia y cuya pendiente o grado de inclinación indica la velocidad de propagación de la epidemia, que está asociada al modo de transmisión de la gente y al tamaño de la población susceptible.

- El punto máximo o meseta, que puede ser alcanzado naturalmente o truncado por una intervención temprana.
- La curva descendente, que representa la fase de agotamiento de la epidemia y cuya pendiente o grado de inclinación descendente indica la velocidad de agotamiento de la población susceptible, sea naturalmente o por efecto o impacto de las medidas de control establecidas.

3. Corredor endémico

Una segunda forma de identificar una tendencia epidémica es a través de un corredor endémico (también llamado canal endémico). El corredor endémico es también una representación gráfica de las frecuencias de la enfermedad en un eje de coordenadas, en el cual el eje horizontal representa el tiempo y el vertical las frecuencias.

Sin embargo, a diferencia de la curva epidémica, el corredor endémico describe en forma resumida la distribución de frecuencias de la enfermedad para el periodo de un año, basada en el comportamiento observado de la enfermedad durante varios años previos yen secuencia. El corredor endémico suele ser representado gráficamente por tres curvas: la curva endémica y otras dos curvas límite, que indican los valores máximos y mínimos, a fin de tomar en cuenta la variación inherente a las observaciones de la frecuencia de la enfermedad a través del tiempo. Así, el corredor endémico expresa, en forma gráfica, la distribución típica de una enfermedad durante un año cualquiera, captura la tendencia estacional de la enfermedad y representa el comportamiento esperado de dicha enfermedad en un año calendario.

En los servicios locales de salud, el corredor endémico es un instrumento útil para el análisis de la situación epidemiológica actual de una enfermedad, la determinación de situaciones de alarma epidémica y la predicción de epidemias.

Para ello, básicamente, se debe superponer la curva epidémica actual (frecuencia observada) al corredor endémico (frecuencia esperada).

El corredor endémico expresa la tendencia estacional de una enfermedad y tiene los siguientes elementos:

- La curva endémica propiamente dicha o nivel endémico, que corresponde a la línea central del gráfico y representa la frecuencia esperada promedio de casos encada unidad de tiempo del año calendario; expresa una medida resumen de tendencia central de la distribución de datos observados (mediana, promedio, etc.).
- El límite superior, o umbral epidémico, que corresponde a la línea superior del gráfico y representa la frecuencia esperada máxima de casos en cada unidad de tiempo del año calendario; expresa una medida resumen de dispersión de la distribución de los datos observados (cuartil superior, desviación estándar, etc.).
- El límite inferior, o nivel de seguridad, que corresponde a la línea inferior del gráfico y representa la frecuencia esperada mínima de casos en cada unidad de tiempo del año calendario; expresa una medida resumen de dispersión de la distribución de datos observados (cuartil inferior, desviación estándar, etc.).
- El corredor o canal endémico, que corresponde a la franja delimitada por los límites inferior y superior del gráfico y representa el rango de variación esperado de casos en cada unidad de tiempo del año calendario.
- La zona de éxito, que corresponde a la franja delimitada por la línea basal (línea de frecuencia cero) y el límite inferior en cada unidad de tiempo del año calendario.
- La zona de seguridad, que corresponde a la franja delimitada por el límite inferiorylacurvaendémicapropiamentedichaencadaunidaddetiempodelañocalendario.

- La zona de alarma, que corresponde a la franja delimitada por la curva endémica propiamente dicha y el límite superior en cada unidad de tiempo del año calendario.
- La zona de epidemia, que corresponde a la zona localizada por encima del límite superior o umbral epidémico en cada unidad de tiempo del año calendario.

En general, al monitorear el comportamiento actual de los casos notificados en función del respectivo corredor endémico, cada cambio de una zona a otra debería acompañarse de una acción correspondiente sobre el sistema de vigilancia, desde la revisión de la validación de los datos de vigilancia y las visitas de supervisión a las unidades notificadoras hasta la implementación de medidas de emergencia.

4. Interpretación de Información

La interpretación de los hallazgos del análisis sirve para la generación de hipótesis, para lo cual debe tenerse en consideración una serie de posibles explicaciones alternativas.

Factores tales como el aumento de la población, la migración, la introducción de nuevos métodos diagnósticos, el mejoramiento de los sistemas de notificación, el cambio en la definición de casos, la aparición de nuevos y efectivos tratamientos y la posibilidad de problemas con la validez de los datos de vigilancia, por subregistro, sesgos o duplicación de notificaciones pueden producir resultados espurios o falsos. Esto deberá guiar el grado y extensión de las recomendaciones de acción dirigidas al control del problema, así como la necesidad de realizar estudios epidemiológicos específicos y de evaluar el sistema de vigilancia.

5. Difusión de información

La difusión periódica de la información que resulte del análisis e interpretación de los datos recolectados y de las medidas de control tomadas, constituye una de las etapas cruciales de lavigilancia.Dadoqueelanálisisdedatosdeberealizarseentodoslosnivelesdelsi

stema, la retroalimentación del sistema debe también llegar a esos mismos niveles.

Los datos de la vigilancia tienen una jerarquía de flujo; ellos fluyen desde el nivel más periférico, que es donde se generan (médico, personal de enfermería, personal auxiliar, servicio de urgencias, laboratorio, comunidad) hacia el nivel regional. Una vez consolidados, se remiten al nivel nacional.

6. Herramientas de desarrollo

6.1. Selección de Herramientas

A continuación se muestra la arquitectura de la solución(Ver figura 8) la cual se basa en el análisis de herramientas que se realizó en el trabajo de seminario.

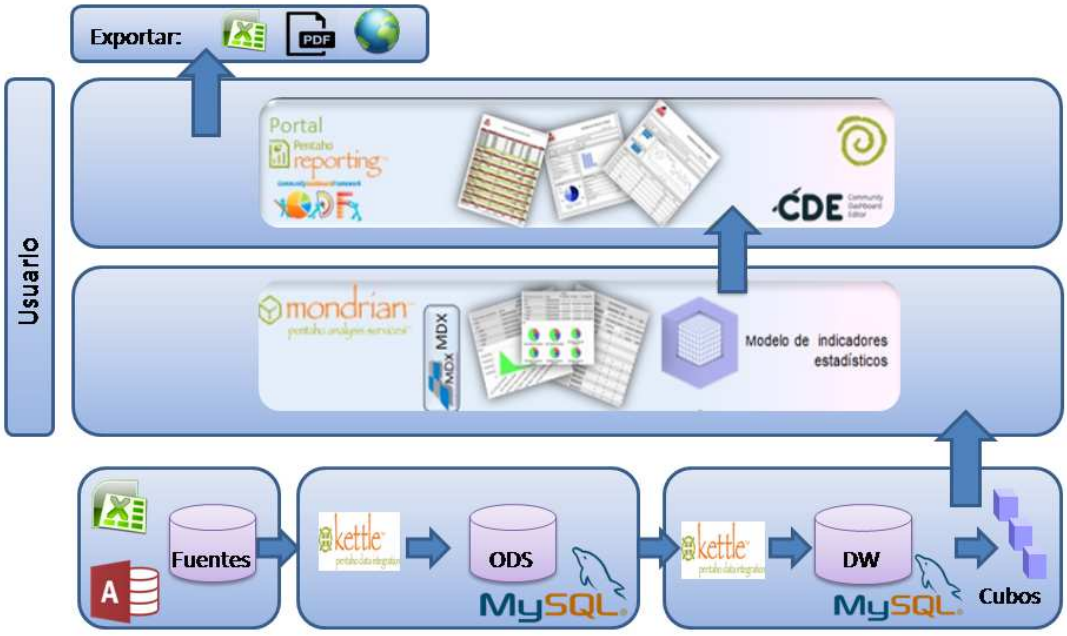


Figura 8: arquitectura de la solución.

1.Los procesos de extracción y transformación y carga (ETL) se realizan a través de la herramienta Pentaho Data Integration Kettle, debido la

usabilidad que la herramienta aporta mediante su interfaz gráfica y el concepto de drag and drop en la cual está basada.

2.La Arquitectura está compuesta por un ODS, donde se integra la data proveniente de documentos Access, y servirá de apoyo para los sistemas transaccionales en la carga inicial. Para las cargas futuras se hace uso de un Staging área llamado staging area.

3.Se integra la data proveniente de las fuentes de datos a través de un primer proceso ETL hacia el ODS.

4.Una vez que termina el primer proceso ETL, se extrae la data integrada y estandarizada desde el ODS hacia el datamart a través de un segundo proceso de ETL.

5.El ODS, el datamart y el staging area serán implementados físicamente en Mysql.

6. Se hace uso del motor OLAP Mondrian para implementar los cubos multidimensionales.

7. Una vez definidos los cubos, estos son cargados al servidor de BI pentaho, el cual permite el despliegue de los cubos definidos en Mondrian y generación de gráficos.

8. A través del uso de la herramienta Jpivot se podrán realizar operaciones OLAP como drill down, Roll up, slice and dice y pivot, que permitirá un análisis de la data.

9. Se hace usos de la Pentaho Report Designer que permite la creación de reportes parametrizados.

Se puede extraer los reportes en formatos Excel, pdf, y html.

6.2 MySQL

MySQL es sistema de administración de bases de datos relacional (SMBDR), que almacena y distribuye una gran cantidad de datos, típicos de una aplicación.

Está basado en la arquitectura cliente-servidor, por lo que el servidor de BD puede estar asociado a múltiples clientes. Utiliza el Lenguaje de Consulta Estructurado o SQL, por sus siglas en inglés Structured Query Language para el acceso y manipulación de los datos.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

6.2.2 Características principales

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.

- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos. Posee un buen enfoque en cuanto a seguridad, ya que ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- El servidor soporta mensajes de error en distintos idiomas.
- Permite escoger entre múltiples motores de almacenamiento para cada tabla que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones. En MySQL 5.0 éstos debían añadirse en tiempo de compilación, a partir de MySQL 5.1 se pueden agregar dinámicamente en tiempo de ejecución. Los hay nativos como MyISAM, Falcon, Merge, InnoDB, BDB, Memory/heap, MySQL Clúster, Federated, Archive, CSV, Blackhole y Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

6.2.3 Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

- Conectividad y seguridad

6.3 Pentaho

6.3.1 Pentaho BI Suite Community Edition

Pentaho es una herramienta de Inteligencia de Negocio desarrollada bajo la filosofía del software libre para la gestión y toma de decisiones empresariales. Es una plataforma compuesta de diferentes programas que satisfacen los requisitos de BI. Ofreciendo soluciones para la gestión y análisis de la información, incluyendo el análisis multidimensional OLAP, presentación de informes, minería de datos y creación de cuadros de mando para el usuario.

La plataforma ha sido desarrollada bajo el lenguaje de programación Java y tiene un ambiente de implementación también basado en Java, haciendo así que Pentaho sea una solución muy flexible al cubrir una alta gama de necesidades empresariales.

6.3.2 Características de la plataforma

- Servidor: puede correr en servidores compatibles con J2EE como JBOSS AS, WebSphere, Tomcat, WebLogic y Oracle AS.
- Base de datos: vía JDBC, IBM DB2, Microsft SQL Server, MySQL, Oracle, PostgreSQL, NCR Teradata, Firebird.
- Sistema operativo: no hay dependencia. Lenguaje interpretado.
- Lenguaje de programación: Java, Javascript, JSP, XSL (XSLT/XPath/XSL-FO).
- Interfaz de desarrollo: Java SWT, Eclipse, Web-based.
- Repositorio de datos basado en XML.
- Todos los componentes están expuestos vía Web Services para facilitar la integración con Arquitecturas Orientadas a Servicios (SOA).

6.3.3 Arquitectura

En la figura 12 se puede ver la arquitectura de la plataforma Pentaho:

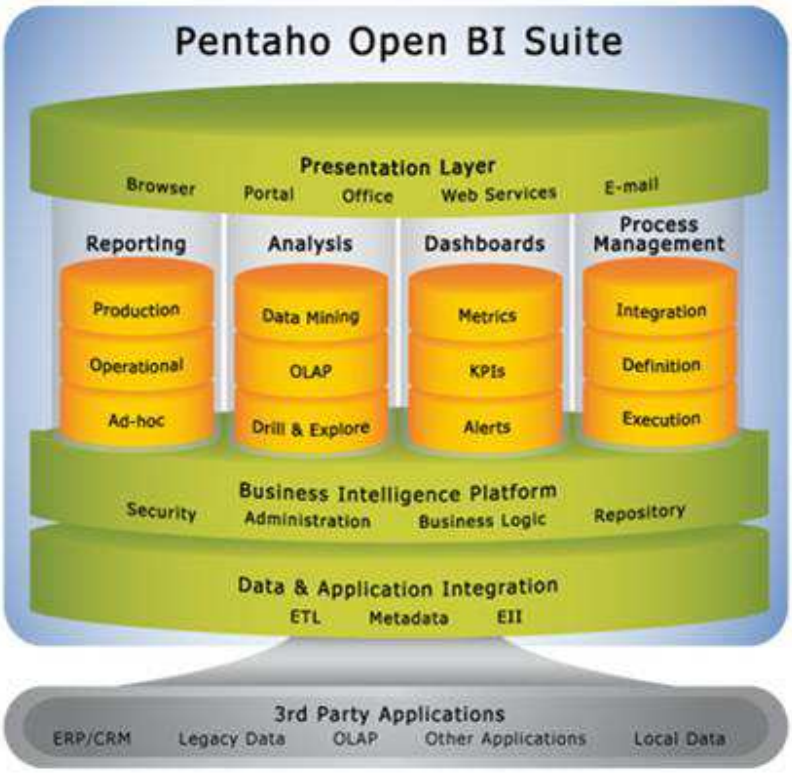


Figura 9: Arquitectura Pentaho BI Suite.

Los productos destacados ofertados en la Suite de Business Intelligence son los siguientes:

Pentaho Data Integration: herramienta que proporciona mediante una interfaz de usuario sencilla e intuitiva la posibilidad de manipulación de los datos desde una fuente externa e independiente a la herramienta.

Pentaho Analisys Services: herramienta para crear cubos multidimensionales OLAP. Soporta el lenguaje de consulta MDX (expresiones multidimensionales) y lenguaje XML para el análisis y especificaciones. Pentaho Analysis suministra a los usuarios un sistema

avanzado de análisis de información. Con uso de las tablas dinámicas (pivot tables, crosstabs) generadas por Mondrian y JPivot, el usuario puede navegar por los datos ajustando la visión de los datos y los filtros de visualización añadiendo o quitando los campos de agregación.

Las características generales son:

- Vista dimensional de datos (por ventas, por periodo).
- Navegar y explorar.
- Análisis Ad Hoc.
- Drill-down.
- Seleccionar un específico miembro para el análisis.
- Interactuar con alto rendimiento.
- Tecnología optimizada para rápida respuesta interactiva.

Pentaho Reporting: herramienta con la cual el usuario será capaz de crear informes usando datos de fuentes externas. Estos informes son generados en XML y pueden ser exportados a diversos tipos de archivos finales, como puede ser PDF, HTML o documentos de texto. Una de las características es que dispone de un menú interactivo que guía al usuario paso por paso en la creación de los informes.

La solución proporcionada por la plataforma Business Intelligence OpenSource pentaho e integrada en su suite para el desarrollo de informes se llama pentaho Reporting

Existen tres productos con diferentes enfoques y dirigidos a diferentes tipos de usuarios.

Pentaho Report Designer:Editor basado en eclipse con prestaciones profesionales y de calidad y con capacidad de personalización de informes a las necesidades de negocio destinado desarrolladores.
Incluye Asistentes para facilitar la configuración de propiedades.

Está estructurado de forma que los desarrolladores pueden acceder a sus prestaciones de forma rápida:

Incluye un editor de consultas para facilitar la confección de los datos que serán utilizados en un informe.

Pentaho Report Design Wizard: Herramienta de diseño de informes, que facilita el trabajo y permite a los usuarios obtener resultados de forma inmediata. Está destinada a usuarios con menos conocimientos técnicos.

A través de pasos sencillos permite:

- Conectarse a todo tipo de bases relacionales.
- Integrar el resultado dentro del portal pentaho.
- Posibilidad de montar codificación semafórica.

Web ad-hoc reporting: Es el similar a la herramienta anterior pero via web. Extiende la capacidad de los usuarios finales para la creación de informes a partir de plantillas preconfiguradas y siguiendo un asistente de creación.

Las características generales se describen a continuación.

Proporciona funcionalidad crítica para usuarios finales como:

- Acceso vía web.
- Informes parametrizados.
- Scheduling.
- Suscripciones.
- Distribucíon (bursting).

Proporciona claras ventajas a especialistas en informes:

- Acceso a fuentes de datos heterogeneos: relacional (vía jdbc), OLAP, XML, transformaciones de pentaho data integration.

- Capacidad de integración en aplicaciones o portales: jsp, portlet, web service.
- Definición modular de informes (distinción entre presentación y consulta).

Diseño de informes flexible:

- Entorno de diseño gráfico.
- Capacidad de uso de templates.
- Acceso a datos relacionaes, OLAP y XML.

Desarrollado para:

- Ser embebible.
- Ser fácil de extender.
- No consumir muchos recursos.
- 100% Java: portabilidad, escalabilidad e integración.

Multiplataforma (tanto a nivel de cliente como servidor):

- Mac.
- Linux/unix.
- Windows.

Pentaho BI Server: herramienta que proporciona el servidor y plataforma web del usuario final. Este podrá interactuar con la solución Business Intelligence previamente creada con las herramientas anteriormente comentadas.

Plataforma 100% J2EE, asegurando la escalabilidad, integración y portabilidad.

Mondrian: En integración con el módulo Pentaho Analytics se incorpora Mondrian, el cual es un motor OLAP escrito en JAVA. El mismo,

ejecuta consultas haciendo uso del lenguaje MDX, lo que permite la lectura de datos desde la base de datos relacional (RDBMS), y presenta los resultados de dicha consulta en un formato multidimensional a través del API Java. Por lo tanto permite a los usuarios de negocio analizar largas y complejas cantidades de datos en tiempo real.

Pentaho Data Integration (Kettle): muchas organizaciones tienen información disponible en aplicaciones y base de datos separados. Pentaho Data Integration abre, limpia e integra esta valiosa información y la pone en manos del usuario. Provee una consistencia, una sola versión de todos los recursos de información, que es uno de los más grandes desafíos para las organizaciones TI hoy en día. Pentaho Data Integration permite un poderoso proceso de ETL. El uso de kettle permite evitar grandes cargas de trabajo manual frecuentemente difícil de mantener y de desplegar.

Las características básicas de esta herramienta son:

- Entorno gráfico de desarrollo.
- Uso de tecnologías estándar: Java, XML, Java Script.
- Fácil de instalar y configurar.
- Multiplataforma: Windows, Macintosh, Linux.
- Basado en dos tipos de objetos: Transformaciones (colección de pasos en un proceso ETL) y trabajos (colección de transformaciones).

Incluye cuatro herramientas:

- Spoon: para diseñar transformaciones o trabajos ETL usando el entorno gráfico.
- PAN: para ejecutar transformaciones diseñadas con spoon mediante líneas de comando.
- CHEF: para crear trabajos mediante líneas de comando.

- Kitchen: para ejecutar trabajos mediante líneas de comando.

A continuación se muestra la arquitectura de una solución de un Datamart(Ver figura 10).

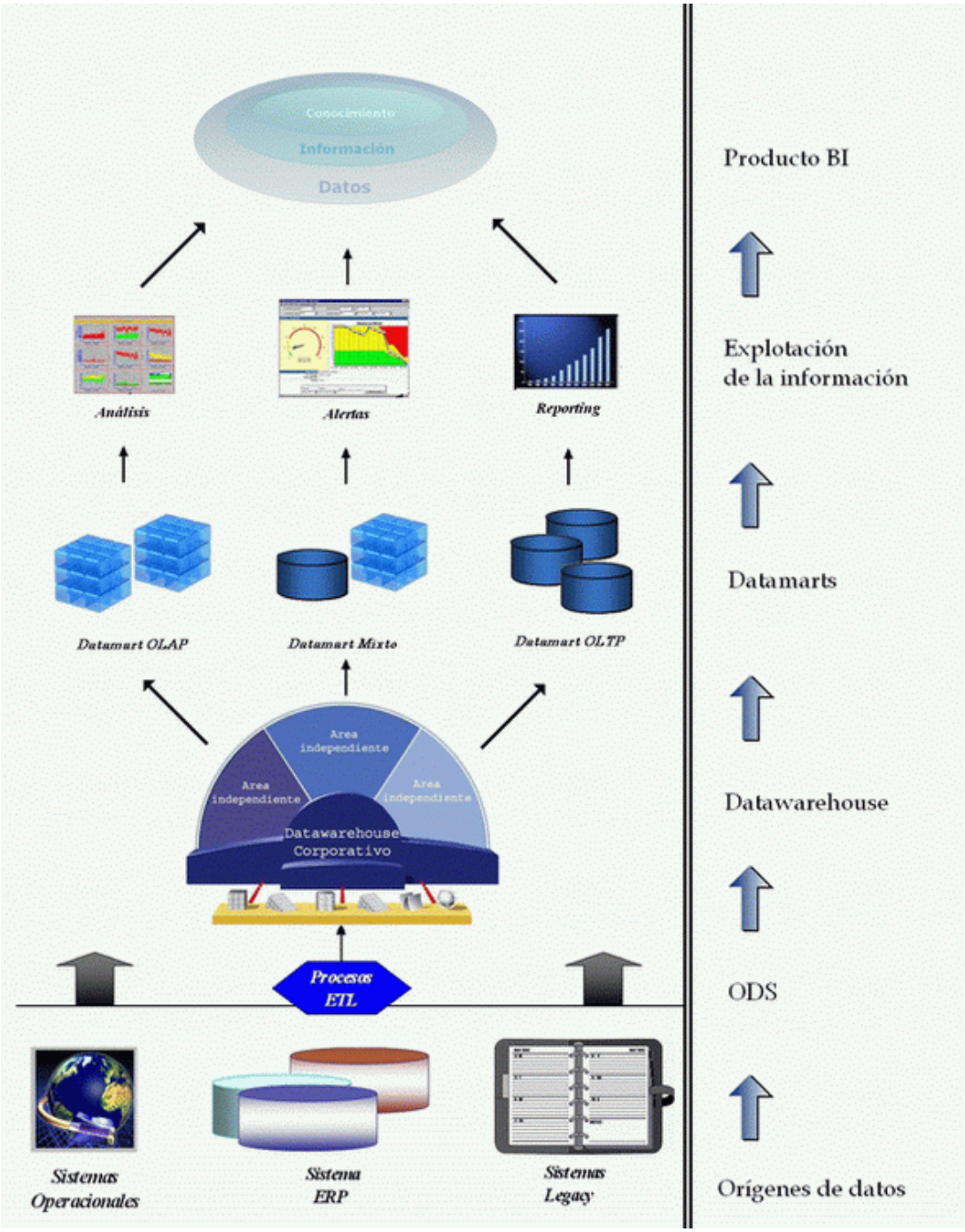


Figura 10: arquitectura de una solución de un Datamart.

Partiendo de la imagen anterior entonces se especifica la tecnología utilizada para cada componente:

ODS, DataWarehouse

Ambos son implementados en el manejador de bases de datos Mysql.

Procesos ETL

Los procesos son implementados en la herramienta Spoon.

DataMarts OLAP

Son construidos mediante el motor OLAP Mondrian.

Análisis

Mediante Jpivot se lleva a cabo el análisis OLAP.

Reporting

Se hace uso de la herramienta Pentaho Report Designer para la generación de reportes parametrizados.

7. METODO DE DESARROLLO DE UN DATA WAREHOUSE SELECCIONADO.

Partiendo del análisis de metodologíasque se llevó a cabo en el trabajo de seminario a continuación se describe la metodología escogida para la implementación del Datamart.

7.1 Metodología Ascendente (Bottom-up)

La metodología de Kimball proporciona una base empírica y metodológica adecuada para las implementaciones de almacenes de datos pequeños y medianos, dada su gran versatilidad y su enfoque ascendente, que permite construir los almacenes en forma escalonada.

Además presenta una serie de herramientas, tales como planillas, gráficos y documentos, que proporcionan una gran ayuda para iniciarse en el ámbito de la construcción de un DataWarehouse.

La metodología se basa en lo que Kimball denomina Ciclo de Vida Dimensional del Negocio (Business Dimensional Lifecycle) (Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., & Becker, B. (2008)). Este ciclo de vida del proyecto de DW, está basado en cuatro principios básicos:

Centrarse en el negocio: Hay que concentrarse en la identificación de los requerimientos del negocio y su valor asociado, y usar estos esfuerzos para desarrollar relaciones sólidas con el negocio, agudizando el análisis del mismo y la competencia consultiva de los implementadores.

Construir una infraestructura de información adecuada: Diseñar una base de información única, integrada, fácil de usar, de alto rendimiento donde se reflejará la amplia gama de requerimientos de negocio identificados en la empresa.

Realizar entregas en incrementos significativos: crear el almacén de datos (DW) en incrementos entregables en plazos de 6 a 12 meses. Hay que usar el valor de negocio de cada elemento identificado para determinar el orden de aplicación de los incrementos. En esto la metodología se parece a las metodologías ágiles de construcción de software.

Ofrecer la solución completa: proporcionar todos los elementos necesarios para entregar valor a los usuarios de negocio. Para comenzar, esto significa tener un almacén de datos sólido, bien diseñado, con calidad probada, y accesible. También se deberá entregar herramientas de consulta ad hoc, aplicaciones para informes y análisis avanzado, capacitación, soporte, sitio web y documentación.

La construcción de una solución de DW/BI (Datawarehouse/Business Intelligence) es sumamente compleja, y Kimball nos propone una metodología que nos ayuda a simplificar esa

complejidad. Las tareas de esta metodología (ciclo de vida) se muestran en la figura 11.

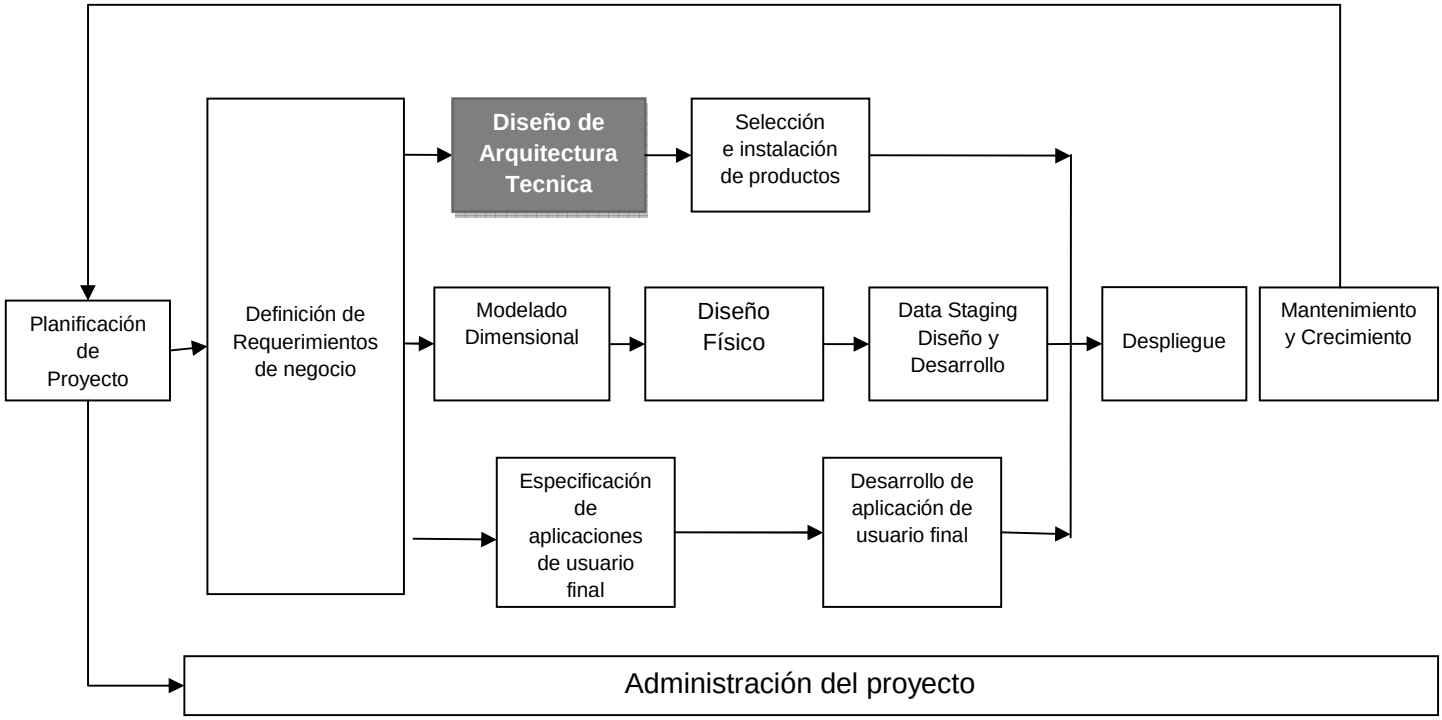


Figura 11: Tareas de la metodología de Kimball, denominada Business Dimensional Lifecycle.

A continuación se describe cada una de las fases del ciclo de vida de kimball que se visualiza en la figura 11.

Planificación

En este proceso se determina el propósito del proyecto de DW/BI, sus objetivos específicos y el alcance del mismo, los principales riesgos y una aproximación inicial a las necesidades de información.

En la visión de programas y proyectos de Kimball, Proyecto, se refiere a una iteración simple del KLC (Kimball Life Cycle), desde el lanzamiento hasta el despliegue.

Esta tarea incluye las siguientes acciones típicas de un plan de proyecto:

- Definir el alcance.
- Identificar las tareas
- Programar las tareas
- Planificar el uso de los recursos.
- Asignar la carga de trabajo a los recursos
- Elaboración de un documento final que representa un plan del proyecto.

Además en esta parte se define cómo realizar la administración o gestión de esta subfase que es todo un proyecto en sí mismo, con las siguientes actividades:

- Monitoreo del estado de los procesos y actividades.
- Rastreo de problemas
- Desarrollo de un plan de comunicación comprensiva que direccione la empresa y las áreas de TI

Análisis de requerimientos

La definición de los requerimientos es en gran medida un proceso de entrevistar al personal del negocio y técnico. Se debe aprender tanto como se pueda sobre el negocio, los competidores, la industria y los clientes del mismo. Hay que leer todos los informes posibles de la organización; rastrear los documentos de estrategia interna; entrevistar a los empleados, analizar lo que se dice en la prensa acerca de la organización, la competencia y la industria. Se deben conocer los términos y la terminología del negocio.

Parte del proceso de preparación es averiguar a quién se debe realmente entrevistar. Esto normalmente implica examinar cuidadosamente el organigrama de la organización. Hay básicamente cuatro grupos de personas con las que hablar desde el principio: el directivo responsable de tomar las decisiones estratégicas; los administradores intermedios y de negocio responsables de explorar

alternativas estratégicas y aplicar decisiones; personal de sistemas, si existen, la gente que realmente sabe qué tipos de problemas informáticos y de datos existen; y por último, la gente que se necesita entrevistar por razones políticas.

Por otra parte, a partir del análisis se puede construir una herramienta de la metodología denominada matriz de procesos/dimensiones (Bus Matrix en inglés).

Una dimensión es una forma, vista o criterio por medio de cual se pueden sumarizar, cruzar o cortar datos numéricos a analizar, datos que se denominan medidas (measures en inglés).

Esta matriz tiene en sus filas los procesos de negocio identificados, y en las columnas, las dimensiones identificadas.

Modelado Dimensional

La creación de un modelo dimensional es un proceso dinámico y altamente iterativo. Un esquema general se puede ver en la figura 12.

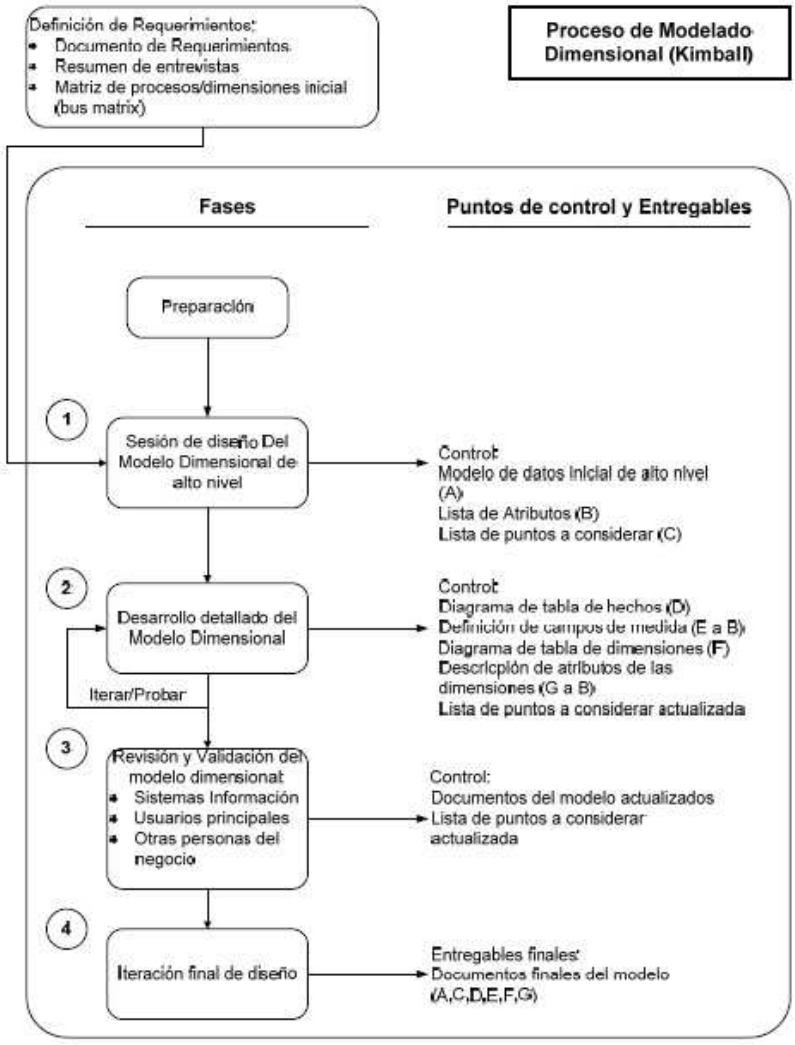


Figura 12: Diagrama de flujo del proceso dimensional de Kimball.

El proceso de diseño comienza con un modelo dimensional de alto nivel obtenido a partir de los procesos priorizados de la matriz descrita en el punto anterior.

El proceso iterativo consiste en cuatro pasos:

1. Elegir el proceso de negocio.
2. Establecer el nivel de granularidad.
3. Elegir las dimensiones.
4. Identificar medidas y las tablas de hechos.

1. Elegir el proceso de negocio

El primer paso es elegir el área a modelar. Esta es una decisión de la dirección, y depende fundamentalmente del análisis de requerimientos y de los temas analíticos anotados en la etapa anterior.

2. Establecer el nivel de granularidad

La granularidad significa especificar el nivel de detalle. La elección de la granularidad depende de los requerimientos del negocio y lo que es posible a partir de los datos actuales.

3. Elegir las dimensiones

Las dimensiones surgen naturalmente de las discusiones del equipo, y facilitadas por la elección del nivel de granularidad y de la matriz de procesos/dimensiones. Las tablas de dimensiones tienen un conjunto de atributos (generalmente textuales) que brindan una perspectiva o forma de análisis sobre una medida en una tabla hechos.

Una forma de identificar las tablas de dimensiones es que sus atributos son posibles candidatos para ser encabezado en los informes, tablas pivot, cubos, o cualquier forma de visualización, unidimensional o multidimensional.

4. Identificar las tablas de hechos y medidas

El último paso consiste en identificar las medidas que surgen de los procesos de negocio. Una medida es un atributo (campo) de una tabla que se desea analizar, sumarizando o agrupando sus datos, usando los criterios de corte conocidos como dimensiones. Las medidas habitualmente se vinculan con el nivel de granularidad, y se encuentran en tablas que denominamos tablas de hechos (fact table en inglés). Cada tabla de hechos tiene como atributos una o más medidas de un proceso organizacional, de acuerdo a los requerimientos. Un registro contiene una medida expresada en números, como cantidad, tiempo, dinero, etc., sobre la cual se desea realizar una operación de agregación (promedio, conteo, suma, etc.) en función de una o más dimensiones. La

granularidad es el nivel de detalle que posee cada registro de una tabla de hechos.

Diseño Físico

En esta parte, se intenta contestar las siguientes preguntas:

¿Cómo puede determinar cuán grande será el sistema de DW/BI?

¿Cuáles son los factores de uso que llevarán a una configuración más grande y más compleja?

¿Cómo se debe configurar el sistema?

¿Cuánta memoria y servidores se necesitan? ¿Qué tipo de almacenamiento y procesadores?

¿Cómo instalar el software en los servidores de desarrollo, prueba y producción?

¿Qué necesitan instalar los diferentes miembros del equipo de DW/BI en sus estaciones de trabajo?

¿Cómo convertir el modelo de datos lógico en un modelo de datos físicos en la base de datos relacional?

¿Cómo conseguir un plan de indexación inicial?

¿Debe usarse la partición en las tablas relacionales?

Diseño del sistema de Extracción, Transformación y Carga (ETL).

El sistema de Extracción, Transformación y Carga (ETL) es la base sobre la cual se alimenta el Datawarehouse. Si el sistema ETL se diseña adecuadamente, puede extraer los datos de los sistemas de origen de datos, aplicar diferentes reglas para aumentar la calidad y consistencia de los mismos, consolidar la información proveniente de distintos sistemas, y finalmente cargar (grabar) la información en el DW en un formato acorde para la utilización por parte de las herramientas de análisis.

Especificación y desarrollo de aplicaciones de usuario final.

Una parte fundamental de todo proyecto de DW/BI está en proporcionarles a una gran comunidad de usuarios una forma más

estructurada y por lo tanto, más fácil, de acceder al almacén de datos. Se proporciona éste acceso estructurado a través de lo que se llama aplicaciones de inteligencia de negocio (Business Intelligence Applications).

Las aplicaciones de BI son la cara visible de la inteligencia de negocio: los informes y aplicaciones de análisis proporcionan información útil a los usuarios. Las aplicaciones de BI incluyen un amplio espectro de tipos de informes y herramientas de análisis, que van desde informes simples de formato fijo a sofisticadas aplicaciones analíticas que usan complejos algoritmos e información del dominio. Kimball divide a estas aplicaciones en dos categorías basadas en el nivel de sofisticación, y les llama informes estándar y aplicaciones analíticas.

Informes estándar: Los informes estándar son la base del espectro de aplicaciones de BI. Por lo general son informes relativamente simples, de formato predefinido, y parámetros de consulta fijos. En el caso más simple, son informes estáticos previamente almacenados. Los informes estándar proporcionan a los usuarios un conjunto básico de información acerca de lo que está sucediendo en un área determinada de la empresa. Este tipo de aplicaciones son el caballo de batalla de la BI de la empresa. Son informes que los usuarios usan día a día. La mayor parte de lo que piden las personas durante el proceso de definición de requisitos se clasificaría como informes estándar. Por eso es conveniente desarrollar un conjunto de informes estándar en el ciclo de vida del proyecto.

Aplicaciones analíticas: Las aplicaciones analíticas son más complejas que los informes estándar. Normalmente se centran en un proceso de negocio específico y resumen cierta experiencia acerca de cómo analizar e interpretar ese proceso de negocio. Estas aplicaciones pueden ser muy avanzadas e incluir algoritmos y modelos de minería de datos, que ayudan a identificar oportunidades o cuestiones subyacentes en los datos. Otra característica avanzada en algunas aplicaciones

analíticas es que el usuario puede pedir cambios en los sistemas transaccionales basándose en los conocimientos obtenidos del uso de la aplicación de BI. En el otro extremo del espectro, algunas aplicaciones analíticas se venden como soluciones cerradas o enlatados, y son independientes de las aplicaciones particulares de la empresa.

Capítulo 3: Marco Aplicativo

2. Fase de Desarrollo del Data warehouse

En la actualidad, las instituciones poseen diversos sistemas que recolectan datos diarios, cabe destacar, que por este mismo motivo, la cantidad de datos obtenidos de estos sistemas es de gran tamaño, por lo que se vuelve casi imposible el uso de los mismos. Aunado a esto, se presentan otra gran variedad de inconvenientes que obstruyen el proceso de toma de decisiones, entre las cuales tenemos, inconsistencias en los datos, información no estructurada, poco personal conocedor de las estructuras de almacenamiento, lo que implica retraso en los tiempos de generación de los reportes, por ende la toma de decisiones incorrectas. En nuestro caso (sector salud), se hace imprescindible que la información histórica, esté disponible para el análisis, en un sólo repositorio con el fin de tomar decisiones acertadas en el momento preciso y poder construir herramientas de análisis y vigilancia endémica como los son los corredores endémicos y las curvas epidemiológicas.

El objetivo del Data warehouse es permitir que los usuarios puedan realizar los análisis que crean convenientes, en función de los datos almacenados. Estos usuarios principalmente serán los epidemiólogos, investigadores y personal encargado de la vigilancia endémica dentro del CISP.

Para comprender las necesidades de los usuarios, se realizó una serie de reuniones con ellos. La información suministrada por los epidemiólogos, permite comprender la situación y los procesos que se llevan actualmente el instituto y posibilita la definición de los requerimientos para el diseño del Data warehouse.

En función de lo anteriormente planteado, se tomó la decisión de enfocar el análisis en los principales entes que conforman el proceso de vigilancia como lo son los diagnósticos, las procedencias, las causas y los casos. Aunado a esto y en concordancia con el tipo de análisis planteado por los usuarios en los diversos requerimientos, es necesario que los datos recolectados sean definidos a nivel fecha y de semana epidemiológica, ya que los epidemiólogos realizan seguimiento a las diferentes enfermedades evaluando su comportamiento en las diferentes semanas epidemiológicas de un año especifico.

2.1. Análisis y recolección de requerimientos

El Data warehouse debe ser creado con el fin de solventar algunas inquietudes persistentes en los usuarios finales, es decir, son ellos los únicos capaces de responder preguntas como ¿Qué quieren ver? ¿Cómo lo quieren ver? ¿Qué análisis desean hacer, sobre qué datos?, para ello los desarrolladores deben identificar claramente los requerimientos de los usuarios, conocer los datos con los que se cuenta y en función de esto, verificar que todas esas solicitudes puedan ser contestadas en función de los datos almacenados en las diversas fuentes de datos origen, ya que de no ser así, se podría descartar ese requerimiento.

Durante esta etapa del Desarrollo del Data warehouse se llevó a cabo el levantamiento de información referente a las necesidades propias de los epidemiólogos, es por ello que principalmente el proceso se centró en el análisis de los boletines epidemiológicos que son generados semanal, mensual y anualmente por parte del equipo de vigilancia.

Dentro de los requerimientos que fueron recolectados, tenemos los distintos tipos de reportes requeridos por los usuarios finales, dentro de los cuales se encuentran:

- Casos de ENO
- Casos por grupo etario

- Casos por procedencia para un diagnostico especifico.
- Casos por sexo y grupo etario para una enfermedad especifica
- Causas de morbilidad
- Consolidado semanal de enfermedades y eventos de notificación obligatoria.
- Defunciones distribuidas por semanas epidemiológicas y procedencias
- Distribución porcentual de la mortalidad infantil por grupos etarios
- Enfermedades de Notificación obligatoria acumulativa
- Enfermedades de Notificación obligatoria
- Morbilidad y mortalidad especifica
- Morbilidad y mortalidad por accidentes agrupados por precedencia
- Morbilidad y mortalidad por accidentes agrupados por sexo
- Morbilidad y mortalidad por grupo
- Muertes infantiles por causas y grupo etario
- Muertes infantiles por semana y grupo etario
- Muertes por grupo etario y semana epidemiológica para un año especifico
- Vigilancia especializada de las enfermedades de notificación obligatoria

2.2 Modelado dimensional

Según el análisis efectuado a partir de la información que fue levantada en la fase de análisis y requerimientos, se pudo identificar que

la agrupación de las diferentes medidas y dimensiones, particulares para este caso de estudio, sugieren que el diseño del Data warehouse siga un esquema estrella, en el cual, como se mencionó con anterioridad, una tabla dimensión puede relacionarse con una o más tablas de hechos.

A continuación se muestra el modelo dimensional diseñado (ver figura 14).

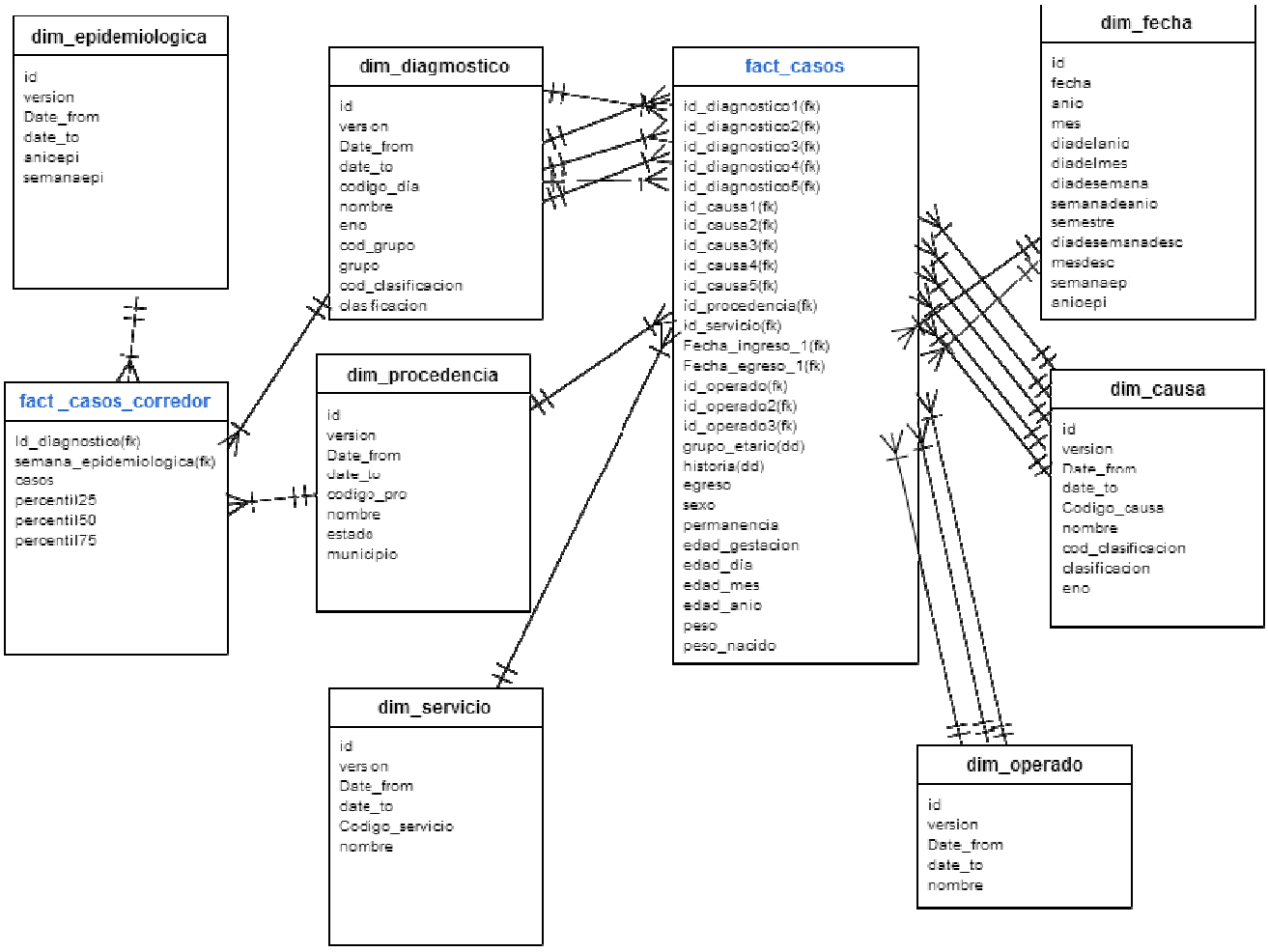


Figura 13: Modelo dimensional diseñado.

En la figura 13 se muestra el modelo dimensional que constituye la estructura de almacenamiento del Datamart.Para poder construir el

modelo, se llevó a cabo una serie de pasos, como sugiere la metodología de Kimball, que son descritos a continuación:

2.2.1. Elegir el proceso de negocio

El proceso elegido es la vigilancia epidemiológica.

2.2.2. Establecer el nivel de granularidad

Se observó que para responder a los requerimientos fue necesario establecer dos granularidades:

La primera granularidad viene definida por la ocurrencia de un caso, que pertenece a una procedencia, es atendido en un servicio específico, así mismo, el caso puede tener hasta 5 diagnósticos y 5 causas, además, puede haber sido sometido a hasta 3 operaciones, dicho caso posee un ingreso al hospital y un egreso del hospital en una fecha determinada.

La segunda granularidad viene definida por la ocurrencia de un número de casos que poseen un mismo diagnóstico, en una semana epidemiológica de un año específico.

2.2.3. Elegir las dimensiones

Tabla 2: Dimension diagnostico

Dim_diagnostico	
Descripción: Almacena las características de los diagnósticos o enfermedades.	
Campo	**Descripción**
id	Clave primaria dentro del modelo dimensional.
version	Versión de la inserción del registro SCD tipo 2.
Date_from	Fecha de la inserción del registro.
date_to	Fecha de validez del registro.

codigo_dia	Código de diagnóstico según CIE 10
nombre	Nombre del diagnóstico según CIE 10
eno	Caracter que indica si es una enfermedad de denuncia obligatoria (S o N)
cod_grupo	Código de grupo de diagnóstico al que pertenece el diagnostico según el CIE 10.
grupo	Nombre del grupo de diagnóstico según CIE 10
cod_clasificacion	Código de clasificación del diagnóstico según CIE 10
clasificacion	Nombre de la clasificación del diagnóstico según CIE 10

Tabla 3: Dimension causa

Dim_causa	
Descripción: Almacena las características de las causas de los diagnósticos.	
Campo	**Descripción**
id	Clave primaria dentro del modelo dimensional.
version	Versión de la inserción del registro SCD tipo 2.
Date_from	Fecha de la inserción del registro.
date_to	Fecha de validez del registro.
Codigo_causa	Código de la causa según CIE 10
nombre	Nombre de la causa según CIE 10
cod_clasificacion	Código de clasificación de la causa según CIE 10
clasificacion	Nombre de la clasificación de la causa según CIE 10

eno	Caracter que indica si la causa es denuncia obligatoria (S o N)

Tabla 4: Dimensión procedencia.

Dim_procedencia	
Descripción: Almacena las características de la procedencia de los casos.	
Campo	**Descripción**
id	Clave primaria dentro del modelo dimensional.
version	Versión de la inserción del registro SCD tipo 2.
Date_from	Fecha de la inserción del registro.
date_to	Fecha de validez del registro.
codigo_pro	Código de procedencia.
nombre	Nombre de la procedencia.
estado	Nombre del estado al que pertenece la procedencia.
municipio	Nombre del municipio al que pertenece la procedencia

Tabla 5: Dimensión servicio.

Dim_servicio	
Descripción: Almacena las características del servicio donde se atiende el caso.	
Campo	**Descripción**
id	Clave primaria dentro del modelo dimensional.
version	Versión de la inserción del registro SCD tipo 2.
Date_from	Fecha de la inserción del registro.

date_to	Fecha de validez del registro.
Codigo_servicio	Código del servicio.
nombre	Nombre del servicio.

Tabla 6: Dimensión operado.

Dim_operado	
Descripción: Almacena las características del servicio donde se atiende el caso.	
Campo	**Descripción**
id	Clave primaria dentro del modelo dimensional.
version	Versión de la inserción del registro SCD tipo 2.
Date_from	Fecha de la inserción del registro.
date_to	Fecha de validez del registro.
nombre	Nombre de la operación.

Tabla 7: Dimensión fecha.

Dim_fecha	
Descripción: Almacena las características de la fecha incluyendo la semana epidemiológica.	
Campo	**Descripción**
id	Clave primaria dentro del modelo dimensional.
fecha	Contiene la fecha completa.
anio	Año de la fecha.
mes	Numero de mes de la fecha.

diadelanio	Numero de dia del año. (1-365)
diadelmes	Numero de dia del mes.(1-31)
diadesemana	Numero de dia de la semana(1-7)
semanadeanio	Numero de semana del año (1-53)
semestre	Numero de semestre del año (1 o 2)
diadesemanadesc	Nombre del dia de la semana(Lunes-Domingo)
mesdesc	Nombre del mes (Enero-Diciembre)
semanaepi	Numero de semana epidemiológica del año(1-53)
anioepi	Año epidemiológico al que pertenece la semana epidemiológica.

Tabla 8: Dimensión epidemiológica.

Dim_epidemiologica		
Descripción: Almacena las características del servicio donde se atiende el caso.		
Campo	**Tipo de dato**	**Descripción**
id	bigint	Clave primaria dentro del modelo dimensional.
version	int	Versión de la inserción del registro SCD tipo 2.
Date_from	datetime	Fecha de la inserción del registro.
date_to	datetime	Fecha de validez del registro.
anioepi	int	Año epidemiológico.
semanaepi	int	Semana epidemiológica (1-53).

Dimensiones Degeneradas

- Grupo_etario

- Historia

2.2.4. Identificar las tablas de hechos y medidas

Tabla 9: tabla de hechos Fact_casos.

Fact_casos	
Descripción: Almacena la ocurrencia de un caso, y los atributos correspondientes para el análisis de la vigilancia.	
Campo	**Descripción**
id_diagnostico1	Clave foránea del diagnóstico principal dado.
id_diagnostico2	Clave foránea del segundo diagnostico dado.
id_diagnostico3	Clave foránea del tercero diagnostico dado.
id_diagnostico4	Clave foránea del cuarto diagnostico dado.
id_diagnostico5	Clave foránea del quinto diagnostico dado.
id_causa1	Clave foránea de la causa principal .
id_causa2	Clave foránea de la segundo causa dado.
id_causa3	Clave foránea de la tercero causa dado.
id_causa4	Clave foránea de la cuarto causa dado.
id_causa5	Clave foránea de la quinto causa dado.
id_procedencia	Clave foránea que indica la procedencia a la que pertenece el caso.
id_servicio	Clave foránea que indica el servicio donde se atendió el caso.
Fecha_ingreso_1	Clave foránea que indica la fecha de ingreso del caso al hospital.

Fecha_egreso_1	Clave foránea que indica la fecha de egreso del caso al hospital.
id_operado	Clave foránea que indica si el caso fue sometido a una operación o que operación se llevó a cabo.
id_operado2	Clave foránea que indica si el caso fue sometido a una segunda operación o que operación se llevó a cabo.
id_operado3	Clave foránea que indica si el caso fue sometido a una tercera operación o que operación se llevó a cabo.
grupo_etario	Dimensión degenerada que indica la clasificación de grupo etario al que pertenece el caso.
historia	Dimensión degenerada que indica el número de historia que se le asignó al caso en el hospital.
egreso	Indica si el caso egresó vivo o muerto del hospital.
sexo	Indica el género de del caso.(F O M)
permanencia	Indica el número de días que el caso permaneció en el hospital.
edad_gestacion	Indica la edad de gestación del caso.
edad_dia	Indica la edad en días el caso.
edad_mes	Indica la edad en meses del caso
edad_anio	Indica la edad en años que posee el caso.
peso	Indica el peso del caso al momento del ingreso al hospital.
peso_nacido	Indica el peso al momento de nacer del caso.

Tabla 10: tabla de hechos Fact_casos_corredor.

Casos_corredor
Descripción: Almacena el número de casos ocurridos con un mismo diagnóstico, en una procedencia y una semana epidemiológica especifica.

Además almacena los percentiles 25, 50, 75 teniendo en cuenta los casos correspondientes a 7 años anteriores.	
Campo	**Descripción**
Id_diagnostico	Clave foránea a la dimensión diagnóstico.
semana_epidemiologica	Clave foránea a la dimensión epidemiológica.
casos	Número total de casos ocurridos en una semana epidemiológica específica, una procedencia y una semana epidemiológica de un año especifico.
percentil25	Percentil 25 tomando en cuenta los casos ocurridos en los 7 años anteriores.
percentil50	Percentil 50 tomando en cuenta los casos ocurridos en los 7 años anteriores.
percentil75	Percentil 75 tomando en cuenta los casos ocurridos en los 7 años anteriores.

2.3 Diseño Físico

En esta fase, se describe como se implementó físicamente cada una de las tablas que conforman el ODS, el datamart y el Staging área.

ODS

A continuación se describen cada una de las tablas que conforman el ODS CISP.

Tabla 11: historia ODS CISP diseño físico.

Historia		
Descripción: Almacena los atributos relacionados con el caso como servico donde fue atendido, procedencia, diagnosticos, causas, entre otros.		
Campo	**Tipo de dato**	**Descripción**

cod_servicio	Varchar(255)	Código univoco del servicio
edad_gestacion	double	Edad de gestación del caso.
peso_nacido	double	Peso al momento de nacer del caso.
fecha_ingreso	datetime	Fecha de ingreso al hospital.
fecha_egreso	datetime	Fecha de egreso al hospital.
edad_anio	int	Años de edad del caso.
edad_mes	int	Meses de edad del caso.
edad_dia	int	Días de edad del caso.
peso	double	Peso en kg. del caso.
cod_procedencia	Varchar(255)	Codigo de procedencia del caso.
cod_diagnostico1	Varchar(255)	Código univoco del diagnóstico principal.
cod_diagnostico2	Varchar(255)	Código univoco del segundo diagnóstico.
cod_diagnostico3	Varchar(255)	Código univoco del tercer diagnóstico.
cod_causa1	Varchar(255)	Código de la primera causa.
cod_causa2	Varchar(255)	Código de la segunda causa.
cod_causa3	Varchar(255)	Código de la tercera causa.
egreso	Varchar(255)	Indica si el caso egresó vivo o muerto (V o M).
sexo	Varchar(2)	Indica el sexo del caso masculino o femenino(F o M).
cod_diagnostico4	Varchar(255)	Código univoco del cuarto diagnóstico.

cod_diagnostico5	Varchar(255)	Código univoco del quinto diagnóstico.
cod_causa4	Varchar(255)	Código de la cuarta causa.
cod_causa5	Varchar(255)	Código de la quinta causa.
operado	Varchar(255)	Indica si el caso ha sido sometido a una operación o cual operación.
operado2	Varchar(255)	Indica si el caso ha sido sometido a una segunda operación o el tipo de operación.
operado3	Varchar(255)	Indica si el caso ha sido sometido a una tercera operación o el tipo de operación
anio	int	Indica el año en que ocurre el caso.
permanencia	int	Número de días de permanencia en el hospital(fecha de egreso menos fecha de ingreso)
historia	Varchar(255)	Número de la historia del caso.
grupo_etario	Varchar(255)	Grupo etario al que pertenece el caso.

Tabla 12: Causas ODS CISP diseño físico.

Causas		
Descripción: Almacena los atributos relacionados a las causas como su nombre, clasificación y si es de denuncia obligatoria.		
Campo	**Tipo de dato**	**Descripción**
Codigo_causa	tinytext	Código univoco de la causa y clave

		primaria.
Nombre_causa	tinytext	Nombre de la causa.
codigo_clasificacion	Varchar(255)	Código de clasificación de la causa.
clasificacion	Varchar(255)	Descripción de la clasificación de la causa.
Eno	Varchar(2)	Indica si la causa es de denuncia obligatoria(S o N).

Tabla 13: Causas_faltan ODS CISP diseño físico.

Causas_faltan		
Descripción: Almacena los códigos de causas de 3 digitos dentro de la tabla historia que no se encuentran en la tabla causa.		
Campo	**Tipo de dato**	**Descripción**
codigo_causa	Varchar(255)	Codigo de causa

Tabla 14: Diagnostico ODS CISP diseño físico.

Diagnostico		
Descripción: Almacena los atributos relacionados a los diagnosticos como su nombre,grupo, clasificación y si es de denuncia obligatoria.		
Campo	**Tipo de dato**	**Descripción**
nombre_dia	tinytext	Nombre del diagnostico
codigo_dia	tinytext	Código univoco del diagnóstico. Clave primaria.
Eno	varchar	Indica si la causa es de denuncia obligatoria(S o N).

cod_grupo	varchar	Código univoco del grupo de diagnóstico.
grupo	varchar	Descripción del grupo de diagnóstico.
cod_clasificacion	varchar	Codigo de clasificación del diagnóstico.
clasificacion	varchar	Descripción de la clasificación del diagnóstico.

Tabla 15: Diagnosticos_faltan ODS CISP diseño fisico.

Diagnosticos_faltan		
Descripción: Almacena los códigos de diagnostico de 3 dígitos dentro de la tabla historia que no se encuentran en la tabla causa.		
Campo	**Tipo de dato**	**Descripción**
codigo_dia	Varchar(255)	Codigo de diagnosatico

Tabla 16: Procedencia ODS CISP diseño fisico.

Procedencia		
Descripción: Almacena los atributos relacionados a las procedencias y los municipios.		
Campo	**Tipo de dato**	**Descripción**
nombre_pro	tinytext	Nombre de la procedencia
codigo_pro	tinytext	Código de la procedencia. Clave primaria.
cod_municipio	int	Clave foránea a la tabla municipio.

Tabla 17: Servicios ODS CISP diseño fisico.

Servicios		
Descripción: Almacena los distintos servicios de los hospitales.		
Campo	**Tipo de dato**	**Descripción**
nombre_ser	tinytext	Nombre del servicio
codigo_ser	tinytext	Código del servicio. Clave primaria.

Tabla 18: Grupos_etarios ODS CISP diseño físico.

Grupos_etarios		
Descripción: Almacena la clasificación de los distintos grupos etarios.		
Campo	**Tipo de dato**	**Descripción**
nombre	Varchar(255)	Nombre del grupo etario
mínimo	int	Indica el mínimo de años del grupo.
máximo	int	Indica el máximo de años del grupo.
Id	int	Código univoco del grupo etario. Clave primaria

Tabla 19: Estado ODS CISP diseño físico.

Estado		
Descripción: Almacena los estados de las distintas procedencias.		
Campo	**Tipo de dato**	**Descripción**
cod_estado	int	Código de estado.
nombre	text	Nombre del estado

Tabla 20: Municipio ODS CISP

Municipio		
Descripción: Almacena los municipios de las distintas procedencias.		
Campo	**Tipo de dato**	**Descripción**
cod_municipio	int	Código de Municipio. Clave primaria.
nombre	text	Nombre del Municipio.
cod_estado	int	Clave foránea a la tabla estado.

Datamart

A continuación se describen cada una de las tablas que conforman el Datamart Cisp_olap.

Tabla 21: Dimension diagnostico diseño físico.

Dim_diagnostico		
Descripción: Almacena las características de los diagnósticos o enfermedades.		
Campo	**Tipo de dato**	**Descripción**
Id	int	Clave primaria dentro del modelo dimensional.
version	int	Versión de la inserción del registro SCD tipo 2.
Date_from	datetime	Fecha de la inserción del registro.
date_to	datetime	Fecha de validez del registro.
codigo_dia	Varchar(255)	Código de diagnóstico según CIE 10
nombre	Varchar(255)	Nombre del diagnóstico según CIE 10

eno	Varchar(2)	Caracter que indica si es una enfermedad de denuncia obligatoria (S o N)
cod_grupo	Varchar(255)	Código de grupo de diagnóstico al que pertenece el diagnostico según el CIE 10.
grupo	Varchar(255)	Nombre del grupo de diagnóstico según CIE 10
cod_clasificacion	Varchar(255)	Código de clasificación del diagnóstico según CIE 10
clasificacion	Varchar(255)	Nombre de la clasificación del diagnóstico según CIE 10

Tabla 22: Dimensión causa diseño físico.

Dim_causa		
Descripción: Almacena las características de las causas de los diagnósticos.		
Campo	**Tipo de dato**	**Descripción**
Id	bigint	Clave primaria dentro del modelo dimensional.
version	int	Versión de la inserción del registro SCD tipo 2.
Date_from	datetime	Fecha de la inserción del registro.
date_to	datetime	Fecha de validez del registro.
Codigo_causa	Varchar(255)	Código de la causa según CIE 10
nombre	Varchar(255)	Nombre de la causa según CIE 10
cod_clasificacion	Varchar(255)	Código de clasificación de la causa según CIE 10
clasificacion	Varchar(255)	Nombre de la clasificación de la causa según CIE 10

eno	Varchar(2)	Caracter que indica si la causa es denuncia obligatoria (S o N)

Tabla 23: Dimensión procedencia diseño fisico.

Dim_procedencia		
Descripción: Almacena las características de la procedencia de los casos.		
Campo	**Tipo de dato**	**Descripción**
Id	bigint	Clave primaria dentro del modelo dimensional.
version	int	Versión de la inserción del registro SCD tipo 2.
Date_from	datetime	Fecha de la inserción del registro.
date_to	datetime	Fecha de validez del registro.
codigo_pro	Varchar(255)	Código de procedencia.
nombre	Varchar(255)	Nombre de la procedencia.
estado	Varchar(255)	Nombre del estado al que pertenece la procedencia.
municipio	Varchar(255)	Nombre del municipio al que pertenece la procedencia

Tabla 24: Dimensión servicio diseño fisico.

Dim_servicio		
Descripción: Almacena las características del servicio donde se atiende el caso.		
Campo	**Tipo de dato**	**Descripción**
Id	bigint	Clave primaria dentro del modelo

		dimensional.
version	int	Versión de la inserción del registro SCD tipo 2.
Date_from	datetime	Fecha de la inserción del registro.
date_to	datetime	Fecha de validez del registro.
Codigo_servicio	Varchar(255)	Código del servicio.
nombre	Varchar(255)	Nombre del servicio.

Tabla 25: Dimensión operado diseño fisico.

Dim_operado		
Descripción: Almacena las características del servicio donde se atiende el caso.		
Campo	**Tipo de dato**	**Descripción**
Id	bigint	Clave primaria dentro del modelo dimensional.
version	int	Versión de la inserción del registro SCD tipo 2.
Date_from	datetime	Fecha de la inserción del registro.
date_to	datetime	Fecha de validez del registro.
nombre	Varchar(255)	Nombre de la operación.

Tabla 26: Dimensión fecha diseño fisico.

Dim_fecha
Descripción: Almacena las características de la fecha incluyendo la semana

epidemiologica.		
Campo	**Tipo de dato**	**Descripción**
Id	int	Clave primaria dentro del modelo dimensional.
fecha	datetime	Contiene la fecha completa.
anio	int	Año de la fecha.
mes	int	Numero de mes de la fecha.
diadelanio	int	Numero de dia del año. (1-365)
diadelmes	int	Numero de dia del mes.(1-31)
diadesemana	int	Numero de dia de la semana(1-7)
semanadeanio	int	Numero de semana del año (1-53)
semestre	char()	Numero de semestre del año (1 o 2)
diadesemanadesc	Varchar(30)	Nombre del dia de la semana(Lunes-Domingo)
mesdesc	Varchar(30)	Nombre del mes (Enero-Diciembre)
semanaepi	int	Numero de semana epidemiológica del año(1-53)
anioepi	int	Año epidemiológico al que pertenece la semana epidemiológica.

Tabla27: Dimensión epidemiológica diseño fisico.

Dim_epidemiologica		
Descripción: Almacena las características del servicio donde se atiende el caso.		
Campo	**Tipo de dato**	**Descripción**
Id	bigint	Clave primaria dentro del modelo dimensional.

version	int	Versión de la inserción del registro SCD tipo 2.
Date_from	datetime	Fecha de la inserción del registro.
date_to	datetime	Fecha de validez del registro.
anioepi	int	Año epidemiológico.
semanaepi	int	Semana epidemiológica (1-53).

Tabla 28: tabla de hechos Fact_casos diseño fisico.

Fact_casos		
Descripción: Almacena la ocurrencia de un caso, y los atributos correspondientes para el análisis de la vigilancia.		
Campo	**Tipo de dato**	**Descripción**
id_diagnostico1	bigint	Clave foránea del diagnóstico principal dado.
id_diagnostico2	bigint	Clave foránea del segundo diagnostico dado.
id_diagnostico3	bigint	Clave foránea del tercero diagnostico dado.
id_diagnostico4	bigint	Clave foránea del cuarto diagnostico dado.
id_diagnostico5	bigint	Clave foránea del quinto diagnostico dado.
id_causa1	bigint	Clave foránea de la causa principal .
id_causa2	bigint	Clave foránea de la segundo causa dado.
id_causa3	bigint	Clave foránea de la tercero causa dado.
id_causa4	bigint	Clave foránea de la cuarto causa dado.

id_causa5	bigint	Clave foránea de la quinto causa dado.
id_procedencia	bigint	Clave foránea que indica la procedencia a la que pertenece el caso.
id_servicio	bigint	Clave foránea que indica el servicio donde se atendió el caso.
Fecha_ingreso_1	int	Clave foránea que indica la fecha de ingreso del caso al hospital.
Fecha_egreso_1	int	Clave foránea que indica la fecha de egreso del caso al hospital.
id_operado	bigint	Clave foránea que indica si el caso fue sometido a una operación o que operación se llevó a cabo.
id_operado2	bigint	Clave foránea que indica si el caso fue sometido a una segunda operación o que operación se llevó a cabo.
id_operado3	bigint	Clave foránea que indica si el caso fue sometido a una tercera operación o que operación se llevó a cabo.
grupo_etario	Varchar(255)	Dimensión degenerada que indica la clasificación de grupo etario al que pertenece el caso.
historia	Varchar(255)	Dimensión degenerada que indica el número de historia que se le asignó al caso en el hospital.
egreso	Varchar(255)	Indica si el caso egresó vivo o muerto del hospital.
sexo	Varchar(2)	Indica el género de del caso.(F O M)
permanencia	int	Indica el número de días que el caso permaneció en el hospital.

edad_gestacion	double	Indica la edad de gestación del caso.
edad_dia	Int	Indica la edad en días el caso.
edad_mes	Int	Indica la edad en meses del caso
edad_anio	Int	Indica la edad en años que posee el caso.
peso	double	Indica el peso del caso al momento del ingreso al hospital.
peso_nacido	double	Indica el peso al momento de nacer del caso.

Tabla 29: tabla de hechos casos_completos diseño fisico.

Fact_casos_corredor		
Descripción: Almacena el número de casos ocurridos con un mismo diagnóstico, en una procedencia y una semana epidemiológica especifica. Además almacena los percentiles 25, 50, 75 teniendo en cuenta los casos correspondientes a 7 años anteriores.		
Campo	**Tipo de dato**	**Descripción**
Id_diagnostico	bigint	Clave foránea a la dimensión diagnóstico.
semana_epidemiologica	datetime	Clave foránea a la dimensión epidemiológica.
casos	datetime	Número total de casos ocurridos en una semana epidemiológica específica, una procedencia y una semana epidemiológica de un año especifico.
percentil25	int	Percentil 25 tomando en cuenta los casos ocurridos en los 7 años anteriores.

percentil50	int	Percentil 50 tomando en cuenta los casos ocurridos en los 7 años anteriores.
percentil75	int	Percentil 75 tomando en cuenta los casos ocurridos en los 7 años anteriores.

Staging Area

Tabla 30: Historia stagingarea diseño fisico.

Historia		
Descripción: Almacena los atributos relacionados con el caso como servico donde fue atendido, procedencia, diagnosticos, causas, entre otros.		
Campo	**Tipo de dato**	**Descripción**
cod_servicio	Varchar(255)	Código univoco del servicio
edad_gestacion	double	Edad de gestación del caso.
peso_nacido	double	Peso al momento de nacer del caso.
fecha_ingreso	datetime	Fecha de ingreso al hospital.
fecha_egreso	datetime	Fecha de egreso al hospital.
edad_anio	int	Años de edad del caso.
edad_mes	int	Meses de edad del caso.
edad_dia	int	Días de edad del caso.
peso	double	Peso en kg. del caso.
cod_procedencia	Varchar(255)	Codigo de procedencia del caso.

cod_diagnostico1	Varchar(255)	Código univoco del diagnóstico principal.
cod_diagnostico2	Varchar(255)	Código univoco del segundo diagnóstico.
cod_diagnostico3	Varchar(255)	Código univoco del tercer diagnóstico.
cod_causa1	Varchar(255)	Código de la primera causa.
cod_causa2	Varchar(255)	Código de la segunda causa.
cod_causa3	Varchar(255)	Código de la tercera causa.
egreso	Varchar(255)	Indica si el caso egresó vivo o muerto (V o M).
sexo	Varchar(2)	Indica el sexo del caso masculino o femenino(F o M).
cod_diagnostico4	Varchar(255)	Código univoco del cuarto diagnóstico.
cod_diagnostico5	Varchar(255)	Código univoco del quinto diagnóstico.
cod_causa4	Varchar(255)	Código de la cuarta causa.
cod_causa5	Varchar(255)	Código de la quinta causa.
operado	Varchar(255)	Indica si el caso ha sido sometido a una operación o cual operación.
operado2	Varchar(255)	Indica si el caso ha sido sometido a una segunda operación o el tipo de operación.
operado3	Varchar(255)	Indica si el caso ha sido sometido a una tercera operación o el tipo de operación
anio	int	Indica el año en que ocurre el caso.

permanencia	int	Número de días de permanencia en el hospital(fecha de egreso menos fecha de ingreso)
historia	Varchar(255)	Número de la historia del caso.
grupo_etario	Varchar(255)	Grupo etario al que pertenece el caso.

Tabla 31: tabla de hechos Fact_casos Stagingarea.

Fact_casos		
Descripción: Almacena la ocurrencia de un caso, y los atributos correspondientes para el análisis de la vigilancia.		
Campo	**Tipo de dato**	**Descripción**
id_diagnostico1	bigint	Clave foránea del diagnóstico principal dado.
id_diagnostico2	bigint	Clave foránea del segundo diagnostico dado.
id_diagnostico3	bigint	Clave foránea del tercero diagnostico dado.
id_diagnostico4	bigint	Clave foránea del cuarto diagnostico dado.
id_diagnostico5	bigint	Clave foránea del quinto diagnostico dado.
id_causa1	bigint	Clave foránea de la causa principal .
id_causa2	bigint	Clave foránea de la segundo causa dado.
id_causa3	bigint	Clave foránea de la tercero causa dado.
id_causa4	bigint	Clave foránea de la cuarto causa dado.
id_causa5	bigint	Clave foránea de la quinto causa dado.

id_procedencia	bigint	Clave foránea que indica la procedencia a la que pertenece el caso.
id_servicio	bigint	Clave foránea que indica el servicio donde se atendió el caso.
Fecha_ingreso_1	int	Clave foránea que indica la fecha de ingreso del caso al hospital.
Fecha_egreso_1	int	Clave foránea que indica la fecha de egreso del caso al hospital.
id_operado	bigint	Clave foránea que indica si el caso fue sometido a una operación o que operación se llevó a cabo.
id_operado2	bigint	Clave foránea que indica si el caso fue sometido a una segunda operación o que operación se llevó a cabo.
id_operado3	bigint	Clave foránea que indica si el caso fue sometido a una tercera operación o que operación se llevó a cabo.
grupo_etario	Varchar(255)	Dimensión degenerada que indica la clasificación de grupo etario al que pertenece el caso.
historia	Varchar(255)	Dimensión degenerada que indica el número de historia que se le asignó al caso en el hospital.
egreso	Varchar(255)	Indica si el caso egresó vivo o muerto del hospital.
sexo	Varchar(2)	Indica el género de del caso.(F O M)
permanencia	int	Indica el número de días que el caso permaneció en el hospital.
edad_gestacion	double	Indica la edad de gestación del caso.

edad_dia	Int	Indica la edad en días el caso.
edad_mes	Int	Indica la edad en meses del caso
edad_anio	Int	Indica la edad en años que posee el caso.
peso	double	Indica el peso del caso al momento del ingreso al hospital.
peso_nacido	double	Indica el peso al momento de nacer del caso.

Tabla 32: tabla de hechos fact_casos_corredorStagingarea.

Casos_completos		
Descripción: Almacena el número de casos ocurridos con un mismo diagnóstico, en una procedencia y una semana epidemiológica especifica. Además almacena los percentiles 25, 50, 75 teniendo en cuenta los casos correspondientes a 7 años anteriores.		
Campo	**Tipo de dato**	**Descripción**
Id_diagnostico	bigint	Clave foránea a la dimensión diagnóstico.
semana_epidemiologica	datetime	Clave foránea a la dimensión epidemiológica.
casos	datetime	Número total de casos ocurridos en una semana epidemiológica específica, una procedencia y una semana epidemiológica de un año especifico.
percentil25	int	Percentil 25 tomando en cuenta los casos ocurridos en los 7 años anteriores.
percentil50	int	Percentil 50 tomando en cuenta los

		casos ocurridos en los 7 años anteriores.
percentil75	int	Percentil 75 tomando en cuenta los casos ocurridos en los 7 años anteriores.

Tabla 33: Causas_faltan Stagingarea.

Causas_faltan		
Descripción: Almacena los códigos de causas de 3 digitos dentro de la tabla historia que no se encuentran en la tabla causa.		
Campo	**Tipo de dato**	**Descripción**
codigo_causa	Varchar(255)	Codigo de causa

Tabla 34: Diagnosticos_faltan Stagingarea.

Diagnosticos_faltan		
Descripción: Almacena los códigos de diagnostico de 3 dígitos dentro de la tabla historia que no se encuentran en la tabla causa.		
Campo	**Tipo de dato**	**Descripción**
codigo_dia	Varchar(255)	Codigo de diagnosatico

Tabla 35: Grupos_etarios Stagingarea

Grupos_etarios		
Descripción: Almacena la clasificación de los distintos grupos etarios.		
Campo	**Tipo de dato**	**Descripción**
nombre	Varchar(255)	Nombre del grupo etario

mínimo	int	Indica el mínimo de años del grupo.
máximo	int	Indica el máximo de años del grupo.
Id	int	Código univoco del grupo etario. Clave primaria

2.4Diseño del sistema de Extracción, Transformación y Carga (ETL).

2.4.1 Identificación de fuentes de datos

En primera instancia, para poder iniciar el diseño del proceso ETL, se tuvo que identificar las fuentes de datos, producto de esto, se pudo identificar 13 archivos Access que se listan a continuación.

- Pediatria(2002)
- Pediatria(2003)
- Pediatria(2004)
- Pediatria(2005)
- Pediatria(2006)
- Pediatria(2007)
- Pediatria(2008)
- Pediatria(2009)
- Pediatria(2010)
- Pediatria(2011)
- Pediatria(2013)
- Pediatria

Dichos archivos Access contienen la data histórica del CISP, desde el año 2002 hasta el 2014. Dentro de los mismos se identificaron como principales las siguientes tablas.

Tabla 36: Fuente externa: Tabla servicios.

Servicios		
Descripción: Almacena el código y el nombre del servicio.		
Campo	**Tipo de dato**	**Descripción**
nombre_ser	text	Nombre del servicio
codigo_ser	text	Código univoco del servicio

Tabla 37: Fuente externa: Tabla procedencia.

Procedencia		
Descripción: Almacena el código y el nombre de la procedencia.		
Campo	**Tipo de dato**	**Descripción**
nombre_pro	text	Nombre de la procedencia
codigo_pro	text	Código univoco del servicio

La tabla historia varía entre los diferentes archivos Access es por ello que continuación se describe las diferentes variaciones de la tabla historia.

Para los archivos Pediatria(2002), Pediatria(2003), Pediatria(2004), Pediatria(2005), Pediatria(2006) y Pediatria(2007), se encuentra definida la tabla historia de la siguiente forma:

Tabla 38: Fuente externa: Tabla historia tipo 1

Historia		
Descripción: Almacena el código y el nombre de la procedencia.		
Campo	**Tipo de dato**	**Descripción**
codser	text	Código univoco del servicio
historia	text	Número de la historia del caso.
servicio	text	Nombre del servicio
edadges	Number	Edad de gestación del caso.
pesonac	Number	Peso al momento de nacer del caso.
fechaing	Date/Time	Fecha de ingreso al hospital.
fechaegr	Date/Time	Fecha de egreso al hospital.
edadaño	Number	Años de edad del caso.
edadmes	Number	Meses de edad del caso.
edaddia	Number	Días de edad del caso.
peso	Number	Peso en kg. del caso.
codpro	Text	Código univoco de la procedencia
procedencia	Text	Nombre de la procedencia.
coddia1	Text	Código univoco del diagnóstico principal.
diagnostico1	Text	Nombre del diagnóstico principal
coddia2	Text	Código univoco del segundo diagnóstico.
diagnostico2	Text	Nombre del segundo diagnóstico.

coddia3	Text	Código univoco del tercer diagnóstico.
diagnostico3	Text	Nombre del tercer diagnóstico.
codcausa	Text	Código de la causa principal.
causa	Text	Nombre de la causa principal.
codcausa2	Text	Código de la segunda causa.
causa2	Text	Nombre de la segunda causa.
codcausa3	Text	Código de la tercera causa.
causa3	Text	Nombre de la tercera causa.
egreso	Text	Indica si el caso egreso vivo o muerto.
sexo	Text	Indica el sexo del caso.

Para los documentos Pediatria(2008), se encuentra definida la tabla historia de la siguiente forma:

Tabla 39: Fuente externa: Tabla Historia tipo 2.

Historia		
Descripción: Almacena el código y el nombre de la procedencia.		
Campo	**Tipo de dato**	**Descripción**
Codser	text	Código univoco del servicio
Historia	text	Número de la historia del caso.
servicio	text	Nombre del servicio
edadges	Number	Edad de gestación del caso.
pesonac	Number	Peso al momento de nacer del caso.

fechaing	Date/Time	Fecha de ingreso al hospital.
fechaegr	Date/Time	Fecha de egreso al hospital.
edadaño	Number	Años de edad del caso.
edadmes	Number	Meses de edad del caso.
edaddia	Number	Días de edad del caso.
Peso	Number	Peso en kg. del caso.
Codpro	Text	Código univoco de la procedencia
procedencia	Text	Nombre de la procedencia.
coddia1	Text	Código univoco del diagnóstico principal.
diagnostico1	Text	Nombre del diagnóstico principal
coddia2	Text	Código univoco del segundo diagnóstico.
diagnostico2	Text	Nombre del segundo diagnóstico.
coddia3	Text	Código univoco del tercer diagnóstico.
Diagnostico4	Text	Nombre del tercer diagnóstico.
Coddia4	Text	
diagnostico3	Text	
Coddia5	Text	
Diagnostico5	Text	
codcausa	Text	Código de la causa principal.
Causa	Text	Nombre de la causa principal.
codcausa2	Text	Código de la segunda causa.

causa2	Text	Nombre de la segunda causa.
codcausa3	Text	Código de la tercera causa.
causa3	Text	Nombre de la tercera causa.
codcausa4	Text	Código de la cuarta causa.
causa4	Text	Nombre de la cuarta causa.
codcausa5	Text	Código de la quinta causa.
causa5	Text	Nombre de la quinta causa.
Egreso	Text	Indica si el caso egreso vivo o muerto.
Sexo	Text	Indica el sexo del caso.

Para los documentos Pediatria(2009) en adelante se define la tabla historia de la siguiente forma:

Tabla 40: Fuente externa: Tabla Historia tipo 3.

Historia		
Descripción: Almacena el código y el nombre de la procedencia.		
Campo	**Tipo de dato**	**Descripción**
Codser	text	Código univoco del servicio
Historia	text	Número de la historia del caso.
servicio	text	Nombre del servicio
edadges	Number	Edad de gestación del caso.
pesonac	Number	Peso al momento de nacer del caso.
fechaing	Date/Time	Fecha de ingreso al hospital.
fechaegr	Date/Time	Fecha de egreso al hospital.

edadaño	Number	Años de edad del caso.
edadmes	Number	Meses de edad del caso.
edaddia	Number	Días de edad del caso.
Peso	Number	Peso en kg. del caso.
Codpro	Text	Código univoco de la procedencia
procedencia	Text	Nombre de la procedencia.
coddia1	Text	Código univoco del diagnóstico principal.
diagnostico1	Text	Nombre del diagnóstico principal
coddia2	Text	Código univoco del segundo diagnóstico.
diagnostico2	Text	Nombre del segundo diagnóstico.
coddia3	Text	Código univoco del tercer diagnóstico.
Diagnostico4	Text	Nombre del tercer diagnóstico.
Coddia4	Text	
diagnostico3	Text	
Coddia5	Text	
Diagnostico5	Text	
codcausa	Text	Código de la causa principal.
Causa	Text	Nombre de la causa principal.
codcausa2	Text	Código de la segunda causa.
causa2	Text	Nombre de la segunda causa.
codcausa3	Text	Código de la tercera causa.

causa3	Text	Nombre de la tercera causa.
codcausa4	Text	Código de la cuarta causa.
causa4	Text	Nombre de la cuarta causa.
codcausa5	Text	Código de la quinta causa.
causa5	Text	Nombre de la quinta causa.
Egreso	Text	Indica si el caso egreso vivo o muerto.
Sexo	Text	Indica el sexo del caso.
operado	Text	Indica si el caso ha sido sometido a una operación o cual operación.
operado2	Text	Indica si el caso ha sido sometido a una segunda operación o el tipo de operación.
operado3	Text	Indica si el caso ha sido sometido a una tercera operación o el tipo de operación

Para poder obtener los datos correspondiente a los diagnósticos y las causas, se optó por generar dos documentos llamados diagnosticosA..Z.xlsx y Causas.xlsx, con los códigos y nombres tanto de diagnosticos como causas según el CIE 10. A continuación se muestra la estructura de ambos archivos.

Tabla 41: Fuente externa: Archivo causas.xlsx.

Causas.xlsx		
Descripción: Almacena el código y el nombre de la procedencia.		
Campo	**Tipo de dato**	**Descripción**
COD_3	General	Código de clasificación de la causa

DESC_3	General	Nombre de la clasificación de la causa.
COD_4	General	Código de la causa
DESC_4	General	Nombre de la causa
ENO	General	Indica si es de denuncia obligatoria o no.

Tabla 42: Fuente externa: Archivo diagnosticosA..Z.xlsx

DiagnosticosA..Z.xlsx		
Descripción: Almacena el código y el nombre de la procedencia.		
Campo	**Tipo de dato**	**Descripción**
COD_GRUPO	General	Código de grupo al que pertenece el diagnóstico.
DESC_GRUPO	General	Código de grupo al que pertenece el diagnóstico.
COD_3	General	Código de clasificación de la diagnostico.
DESC_3	General	Nombre de la clasificación de la diagnostico.
COD_4	General	Código del diagnóstico.
DESC_4	General	Nombre del diagnóstico.
ENO	General	Indica si es de denuncia obligatoria o no.

2.4.2 Transformaciones y Jobs

Los script esenciales para la creación de las tablas dimensión y tablas de hechos correspondientes al Data warehouse en función del diseño planteado con anterioridad, son creados usando lenguaje SQL. Dicha

creación se encuentra inmersa en las transformaciones y Jobs de Pentaho data Integration.

Para comenzar el proceso automatizado de extracción, transformación y carga, se deben crear los jobs (desde Spoon), los cuales están conformados por diversas transformaciones mediante los cuales se especificará el mapeo de los diversos sistemas fuentes al Data warehouse y se realizaran las transformaciones pertinentes para estructurar y estandarizar la data.

Antes de comenzar a explicar la estructura de las transformaciones y jobs creados, es necesario que se mencionen algunos de los iconos referenciados por Pentaho data integration (Kettle), así como su descripción de manera tal de que se haga más fácil su comprensión. Se debe recalcar que Spoon es la herramienta permite el diseño y creación gráfica de transformaciones y jobs.

En la figura 14, se muestra la interfaz de Spoon, la cual consta de un área de trabajo para la creación de transformaciones y jobs, barras de herramientas que permiten la inclusión de elementos o tareas, que se pueden arrastrar a la hoja de diseño.

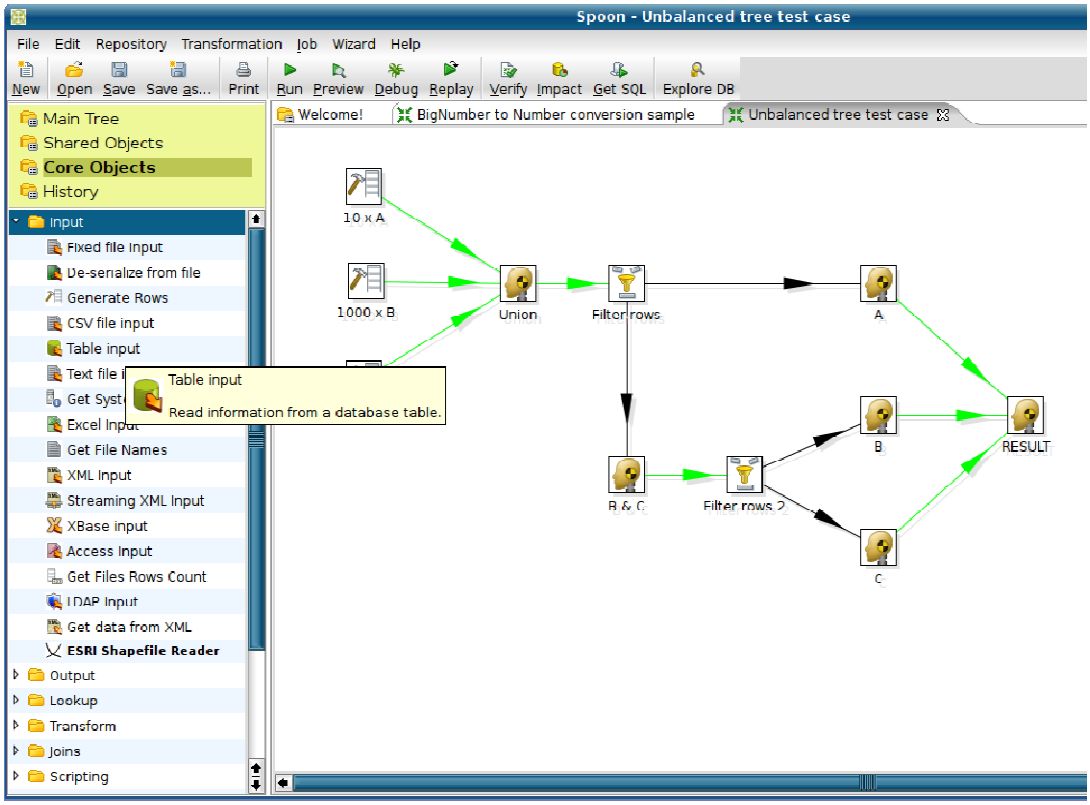


Figura 14: Interfaz de Spoon.

Al arrastrar las conexiones y tareas a la hoja de diseño y especificar el orden de ejecución de los flujos de trabajo, se pueden desarrollar fácilmente eficaces procesos de ETL. Las secciones siguientes definen las tareas, los flujos de trabajo, las conexiones y las transformaciones que tienen lugar e ilustran la facilidad de utilización Spoon para implementar transformaciones y Jobs.

2.4.2.1 Definición de los pasos para la creación de Transformaciones y Jobs

Spoon permite establecer conexión con un gran número de sistemas fuentes, entre los cuales podemos mencionar: archivos de texto, otras bases de datos SQL Server, Mysql, Oracle, etc. Dependiendo del sistema con el que se desee establecer una conexión, se deben especificar ciertos

parámetros de configuración, de manera tal de hacer exitosa dicha comunicación, algunos de ellos pueden ser: rutas y nombres de archivos (en caso de conexiones con archivos planos), nombre del servidor y base de datos específica dentro del mismo (en caso de conexiones con otras bases de datos SQL Server), entre otros. Adicionalmente, una transformación suele incluir una o varias tareas, cada una de las cuales define un elemento de trabajo que se puede llevar a cabo durante la ejecución de la transformación.

A continuación se describen algunas de las tareas:

Tabla 43: tareas de spoon.

	**Table output**	Permite insertar datos en una tabla de una base de datos y la creación de la tabla.
	**Table input**	Permite importar datos de una base de datos mediante sql.
	**Dimension lookup/update**	Permite crear, insertar y actualizar una dimensión.
	**filter**	Permite filtrar los datos de un flujo mientras cumplan alguna condición.
	**Database lookup**	Permite realizar un join de un flujo de entrada con una tabla en una base de datos.
	**dummy**	Permite visualizar la salida de los datos
	**start**	Inicia un job.
	**transformation**	Permite incluir una transformación a un job.
	**Sql script**	Permite ejecutar sentencias SQL tanto en transformaciones como en jobs.

	**Excel input file**	Permite importar datos de un archivo Excel.
	**Unique rows**	Elimina las filas duplicadas de las entradas.
	**Access input**	Permite importar datos de un archivo Access.
	**Generate rows**	Permite generar filas.
	**Add sequence**	Agrega un valor de secuencia.
	**calculator**	Crea nuevos campos mediante la realización de cálculos matemáticos.
	**Select values**	Selección, cambio de nombre, cambio de tipo de datos y la configuración de la longitud y precisión de los campos.
	**Replace in string**	Reemplazar todas las apariciones de una palabra en una cadena con otra palabra.
	**Sort rows**	Ordena las filas basándose en los campos especificados y si deben ser ordenados en orden ascendente o descendente.
	**Stream lookup**	Permite hacer join de dos flujos de entrada.
	**String operaions**	Aplicar operaciones como recorte, relleno y otros para el valor de la cadena.
	**Value mapper**	Mapea de un valor a otro.

2.4.2.2 Creación de las transformaciones y Jobs

Con basamento en lo explicado con anterioridad, a continuación se explicarán cada una de transformaciones y jobs creados para solventar la

problemática planteada en lo que refiere a la extracción, transformación y carga de los datos.

Sin embargo, se debe hacer énfasis en que se llevó a cabo dos procesos ETL. El primer proceso ETL denominado fuentes a ODS, surge por la necesidad de centralizar y estandarizar la data histórica encontrada en los diferentes archivos Access y Excel mencionados con anterioridad, en un ODS, que pretende servir de apoyo a los sistemas fuentes y el segundo proceso ETL. EL segundo proceso ETL es el encargado de extraer la data desde el ODS hacia el Data Mart CISP y estandarizar la data en función de garantizar la respuesta a los requerimientos.

ETL fuentes a ODS

A continuación se describen cada una de las transformaciones implementadas para poder llevar a cabo el primer proceso ETL fuentes a ODS. Sin embargo, es necesario describir de antemano la estructura de datos del ODS para facilitar la comprensión de los mismos.

A continuación en la figura 15 se visualiza el modelo relacional del ODS cisp.

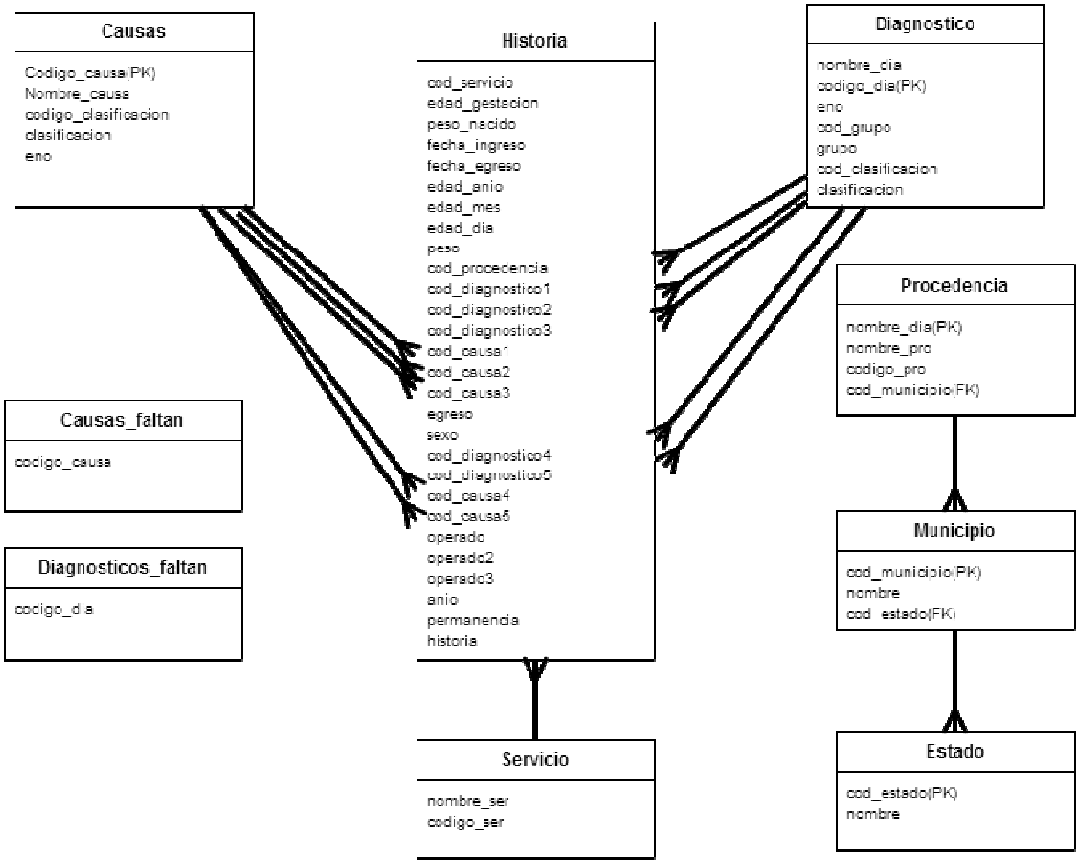


Figura 15: ODS CISP.

En la figura anterior se describe la estructura de almacenamiento del ODS cisp, el cual sirve de apoyo a los sistemas fuente permitiendo la centralización de los datos.

En la figura 17, se puede visualizar la transformación **carga de historias 2002 a 2007.ktr**, con la cual se extraen los casos desde 2002 a 2007, encontrados en los archivos Access, para ser cargados en el ODS.

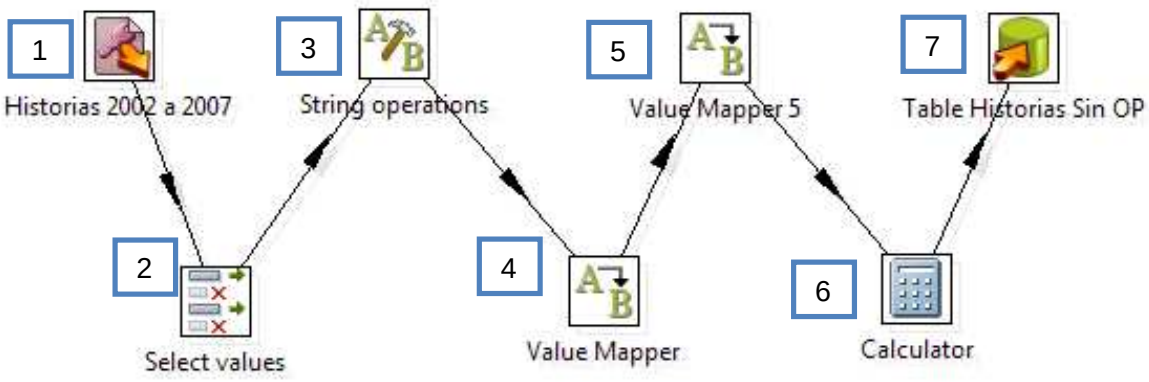


Figura 16: carga de historias 2002 a 2007.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extrae la tabla historia que se encuentran en los archivos Access.

2.-Se asignan los tipos de datos, longitud y precisión al flujo.

3.-Se eliminan los espacios a los códigos y convierten en mayúscula.

4.- Se asigna NA(no asignado) a los campos egreso vacíos y se remplazan los caracteres w,W y f por el carácter M (Muerto) en el campo egreso.

5. Se asigna NA(no asignado) a los campos sexo vacíos.

6.-Se calcula el tiempo de permanencia en el hospital restando la fecha de egreso menos la de ingreso.

7.-Se crea la tabla historia dentro el ODS llamado CISP y se inserta el flujo de datos en dicha tabla.

En la figura 18, se puede visualizar la transformación **carga de historias 2008.ktr**, con la cual se extraen los casos del 2008, encontrados en el archivo Access Pediatria(2008), para ser cargados en el ODS.

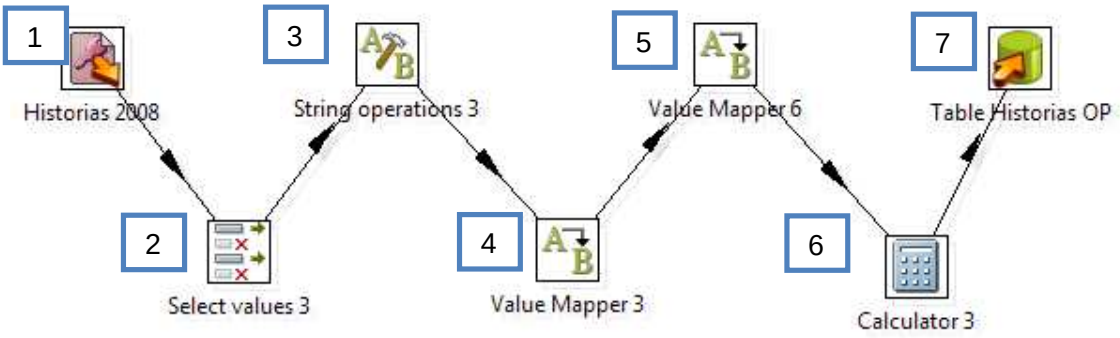


Figura 17: carga de historias 2008.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla historia que se encuentran en los archivos Access.

2.-Se asignan los tipos de datos, longitud y precisión al flujo.

3.-Se eliminan los espacios a los códigos y convierten en mayúscula.

4.- Se asigna NA(no asignado) a los campos egreso vacíos y se remplazan los caracteres w,W y f por el carácter M (Muerto) en el campo egreso.

5. Se asigna NA(no asignado) a los campos sexo vacíos.

6.-Se calcula el tiempo de permanencia en el hospital restando la fecha de egreso menos la de ingreso.

7.-Se crea la tabla historia dentro el ODS llamado CISP y se inserta el flujo de datos en dicha tabla.

En la figura 19, se puede visualizar la transformación **carga de historias 2009 a 2014.ktr**, con la cual se extraen los casos del 2009 al 2014, encontrados en los archivo Access, para ser cargados en el ODS.

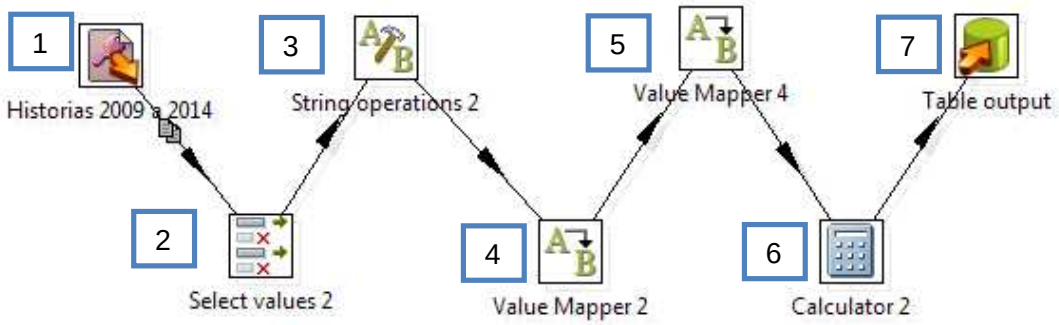


Figura 18: carga de historias 2009 a 2014.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla historia que se encuentran en los archivos Access.

2.-Se asignan los tipos de datos, longitud y precisión al flujo.

3.-Se eliminan los espacios a los códigos y convierten en mayúscula.

4.- Se asigna NA(no asignado) a los campos egreso vacíos y se remplazan los caracteres w,W y f por el carácter M (Muerto) en el campo egreso.

5. Se asigna NA(no asignado) a los campos sexo vacíos.

6.-Se calcula el tiempo de permanencia en el hospital restando la fecha de egreso menos la de ingreso.

7.-Se crea la tabla historia dentro el ODS llamado CISP y se inserta el flujo de datos en dicha tabla.

En la figura 19, se puede visualizar la transformación **quitar puntos en códigos de diagnóstico y causa en historia.ktr**, con la cual se eliminan los puntos en los códigos de diagnóstico y causa en función de estandarizarlos según el CIE 10.

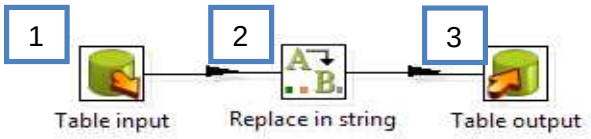


Figura 19: quitar puntos en códigos de diagnóstico y causa en historia.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla historia en el ODS.

2.-Se eliminan los puntos en cod_diagnostico1, cod_diagnostico2, cod_diagnostico3, cod_diagnostico4, cod_diagnostico5, cod_causa1, cod_causa2, cod_causa3, cod_causa4, cod_causa5 y cod_causa5.

3.-Se inserta el flujo en la tabla historia del ODS truncando la tabla.

En la figura 20 se visualiza la transformación **llamar a procedimiento que asigna grupo etario.ktr.**

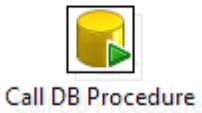


Figura 20: llamar a procedimiento que asigna grupo etario.ktr

En la transformación anterior se asigna el grupo etario a los casos de la tabla historia el ODS. A groso modo lo que se lleva a cabo es unir la tabla grupos_etarios dentro del ODS con los registros de la tabla historia comparando la edad en años. En los anexos se encuentra el procedimiento a detalle.

En la figura 21, se puede visualizar la transformación **carga de procedencias 2002 a 2014.ktr**, con la cual se extraen los datos de la tabla procedencia de los archivos Access y se cargan en el ODS.

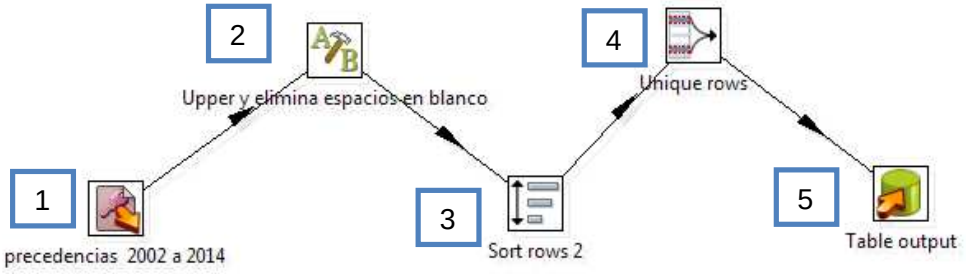


Figura 21: carga de procedencias 2002 a 2014.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla procedencia en todos los archivos Access.
2.-Se eliminan los espacios y se convierten a mayúscula los códigos de procedencia.
3.-Se ordena el flujo por código de procedencia.
4.-Se eliminan los registros duplicados.
5.-Se crea la tabla Procedencia en el ODS CISP y se inserta el flujo en la tabla.

En la figura 22, se puede visualizar la transformación **carga de servicios 2002 a 2014.ktr**, con la cual se extraen los datos de la tabla servicio de los archivos Access y se cargan en el ODS.

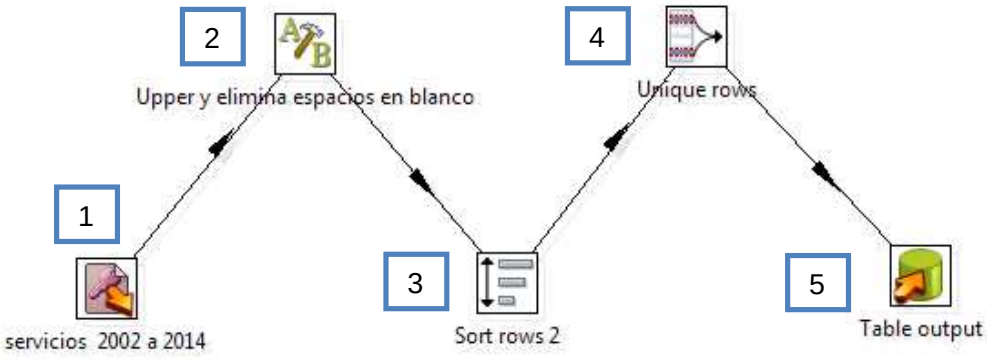


Figura 22: carga de servicios 2002 a 2014.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla servicio en todos los archivos Access.
2.-Se eliminan los espacios y se convierten a mayúscula los códigos de servicio.
3.-Se ordena el flujo por código de servicio.
4.-Se eliminan los registros duplicados.
5.-Se crea la tabla Servicio en el ODS CISP y se inserta el flujo en la tabla.

En la figura 23, se puede visualizar la transformación **carga de causas excel.ktr**, con la cual se extraen los datos del archivo Excel y se cargan en el ODS.

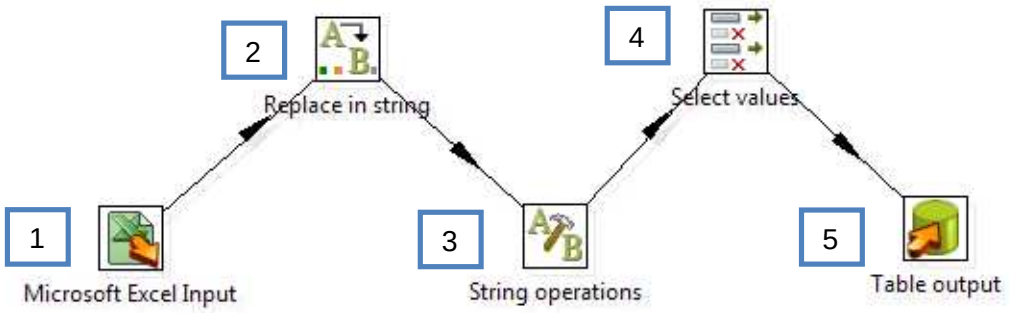


Figura 23: carga de causas excel.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla en Excel.
2.-Se eliminan los caracteres * en los campos COD_3 y COD_4.
3.-Se convierten a mayúscula todos los códigos y descripciones. (COD_3,COD_4 ,DESC_4,DESC_3 y ENO).
4.-Se renombra COD_3 como código_clasificacion, DESC_3 como clasificación, COD_4 como código_causa y DESC_4 como nombre_causa.
5.-Se crea la tabla Causa en el ODS CISP y se inserta el flujo en la tabla.

En la siguiente figura 24, se puede visualizar la transformación **carga de diagnosticos excel.ktr**, con la cual se extraen los datos del archivo Excel diagnosticosA..Z.xlsx y se cargan en el ODS.

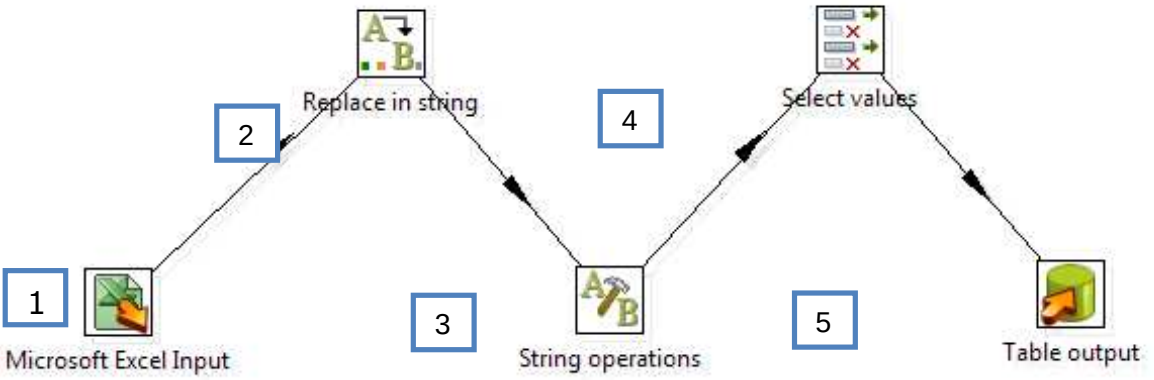


Figura 24: carga de diagnosticos excel.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior

1-Se extraen los registros de la tabla en Excel.
2.-Se eliminan los caracteres * y † en los campos COD_3, COD_4 COD_GRUPO.
3.-Se convierten a mayúscula todos los códigos y descripciones. (COD_3, COD_4, DESC_4, DESC_3, COD_GRUPO, DESC_GRUPO y ENO).
4.-Se renombra COD_3 como código_clasificacion, DESC_3 como clasificación, COD_4 como código_dia, DESC_GRUPO como grupo y DESC_4 como nombre_dia.
5.-Se crea la tabla Diagnostico en el ODS CISP y se inserta el flujo en la tabla.

En la figura 25, se muestra la transformación **obtener los códigos de diagnóstico de 3 dígitos faltantes.ktr**, con la cual se obtienen los códigos de 3 dígitos que no están en la tabla diagnóstico.

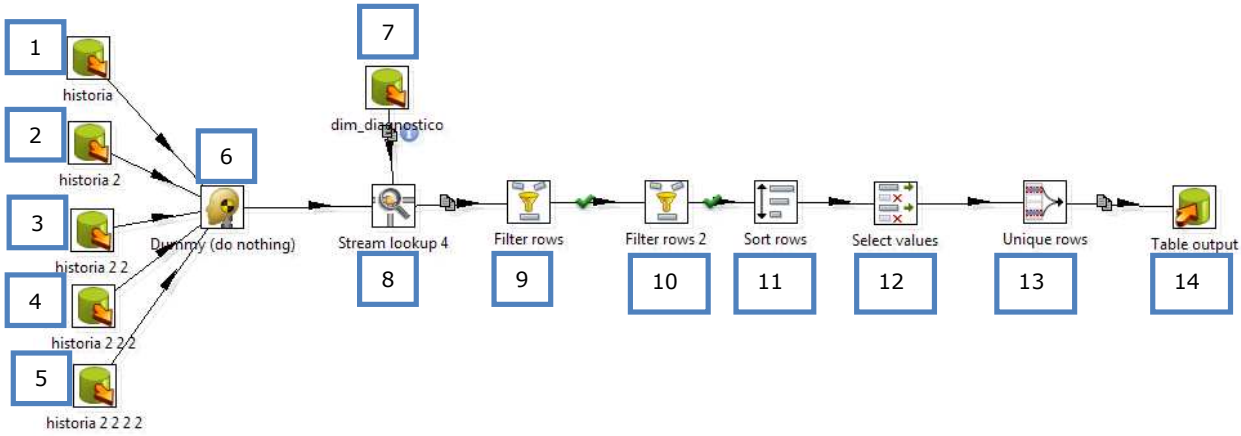


Figura 25: obtener los códigos de diagnóstico de 3 dígitos faltantes.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1.-Se extrae el cod_diagnostico1 de la tabla historia del ODS y la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico1) y se renombra como longitud.

2.-Se extrae el cod_diagnostico2 de la tabla historia del ODS , se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico2) y se renombra como longitud.

3.- Se extrae el cod_diagnostico3 de la tabla historia del ODS , se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico3) y se renombra como longitud.

4.- Se extrae el cod_diagnostico4 de la tabla historia del ODS de la tabla historia del ODS, se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico4) y se renombra como longitud.

5.- Se extrae el cod_diagnostico5 de la tabla historia del ODS de la tabla historia del ODS, se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico5) y se renombra como longitud.
6.-Se unen los 5 flujos en una sola tabla.
7.-Se extrae el campo código_dia de la tabla diagnóstico del ODS.
8.-Se realiza un join entre ambos flujos por el campo código_dia y el campo cod_diagnostico.
9.-Filtra los registros cuyo codigos_dia no sean vacio.
10.-Obtiene los registros que tengan longitud 3.
11.-Ordena los registros por cod_diagnostico.
12.-Remueve el campo longitud.
13.-Elimina los duplicados.
14.- Crea la tabla diagnosticos_faltan con los códigos de 3 dígitos que no se encuentran en la tabla diagnóstico.

A continuacion (ver figura 26) se aprecia una transformación llamada **corregir diagnosticos.ktr.**

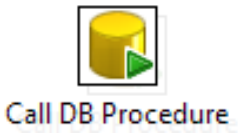


Figura 26: corregir diagnosticos.ktr

En la imagen anterior se llama al procedimiento completar_diagnosticos() el cual agrega un 0 a los códigos de 3 dígitos que se encuentran en la tabla diagnosticos_faltan y verifica que el nuevo código exista en la tabla diagnóstico del ODS y si existe modifica todos los registros que contengan

el código de 3 dígitos dentro de la tabla historia y les asigna el código de cuatro dígitos. Posteriormente elimina el registro en la tabla diagnosticos_faltan.

Dicho procedimiento se realiza para que los códigos de diagnósticos que están errados en la tabla historia sean modificados para que concuerden con los códigos del CIE 10 que se encuentran en la tabla diagnóstico. Dicho procedimiento se encuentra en los anexos.

En la figura 27 se visualiza la transformacion **obtener los codigos de causa de 3 digitos faltantes.ktr.**

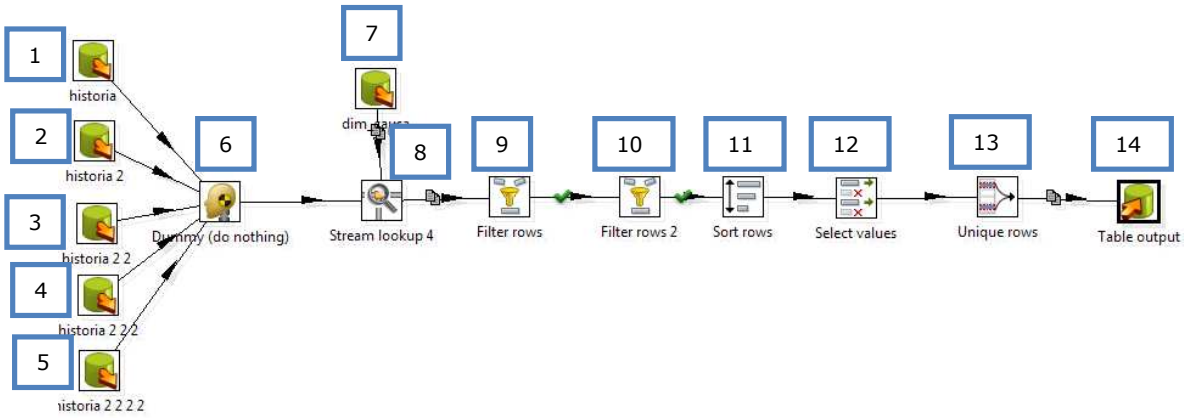


Figura 27: obtener los codigos de causa de 3 digitos faltantes.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1.-Se extrae el cod_causa1 de la tabla historia del ODS , se renombra como como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa1) y se renombra como longitud.

2.-Se extrae el cod_causa2 de la tabla historia del ODS, se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa2) y se renombra como longitud.

3.- Se extrae el cod_ cod_causa 3 de la tabla historia del ODS , se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa 3) y se renombra como longitud.
4.- Se extrae el cod_causa4 de la tabla historia del ODS de la tabla historia del Staging area, se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa4) y se renombra como longitud.
5.- Se extrae el cod_causa5 de la tabla historia de la tabla historia del ODS, se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa5) y se renombra como longitud.
6.-Se unen los 5 flujos en una sola tabla.
7.-Se extrae el campo código_causa de tabla causa en el ODS.
8.-Se realiza un join entre ambos flujos por el campo código_causa y el campo cod_causa .
9.-Filtra los registros cuyo campo cod_causa no esté vacío.
10.-Obtiene los registros que tengan longitud 3.
11.-Ordena los registros por cod_causa.
12.-Remueve el campo longitud.
13.-Elimina los duplicados.
14.- Crea la tabla causas_faltan en el ODS área con los códigos de 3 dígitos que no se encuentran en la tabla diagnóstico del ODS Cisp.

De esta forma culmina el primer proceso ETL, encargado de centralizar la data proveniente de los distintos archivos fuentes.

ETL ODS al Datamart

A continuación se describen cada una de las transformaciones implementadas para poder llevar a cabo el segundo proceso ETL ODS a al Data Warehouse.

En la figura 28, se puede visualizar la transformación **dim_diagnostico.ktr**, con la cual se extraen los diagnósticos del ODS y se cargan en el DW.

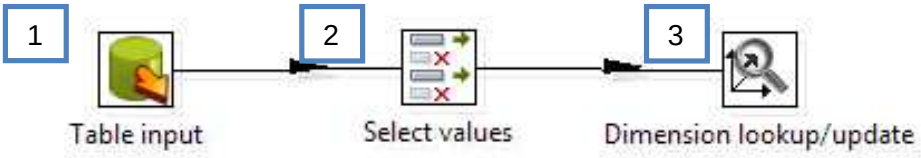


Figura 28: dim_diagnostico.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla Diagnostico que se encuentra en el ODS CISP.
2.-Se seleccionan los campos nombre_dia,código_dia, eno, cod_grupo, grupo, cod_clasificacion y clasificación.
3.-Se crea la dimensión Diagnostico en el Data warehouse Cisp_olap y se insertan los diagnósticos en la dimensión creada.

En la figura 29, se puede visualizar la transformación **dim_causa.ktr**, con la cual se extraen las causas del ODS y se cargan en el DW.

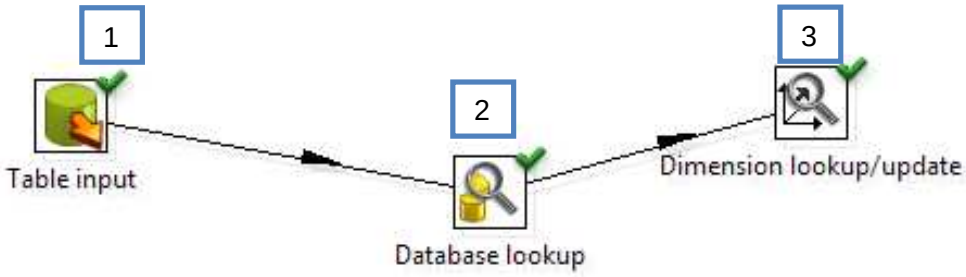


Figura 29: dim_causa.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior

1-Se extraen los registros de la tabla Causa que se encuentra en el ODS CISP.
2.-Se realiza un join con la dimensión diagnóstico para obtener los grupos y sus respectivos códigos de acuerdo al CIE10.
3.-Se crea la dimensión Causa en el Data warehouse CISP_olap y se inserta el flujo en la dimensión creada.

En la figura 30, se puede visualizar la transformación **dim_procedencia.ktr**, con la cual se extraen las procedencias del ODS y se cargan en el DW.

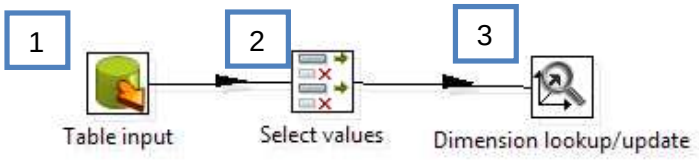


Figura 30: dim_procedencia.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla Procedencia que se encuentra en el ODS CISP.
2.-Se seleccionan los campos estados, municipio, nombre_pro y se les asigna varchar(255) como tipo de dato.
3.-Se crea la dimensión Procedencia en el Data warehouse CISP_olap y se inserta el flujo en la dimensión creada.

En la figura 31, se puede visualizar la transformación **dim_servicio.ktr**, con la cual se extraen los servicios del ODS y se cargan en el DW.

Figura 31: dim_servicio.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la tabla Servicio que se encuentra en el ODS CISP.
2.-Se crea la dimensión Servicio en el Data warehouse CISP_olap y se inserta el flujo en la dimensión creada.

En la figura 32, se puede visualizar la transformación **dim_operado.ktr**, con la cual se extraen las operaciones del ODS y se cargan en el DW.

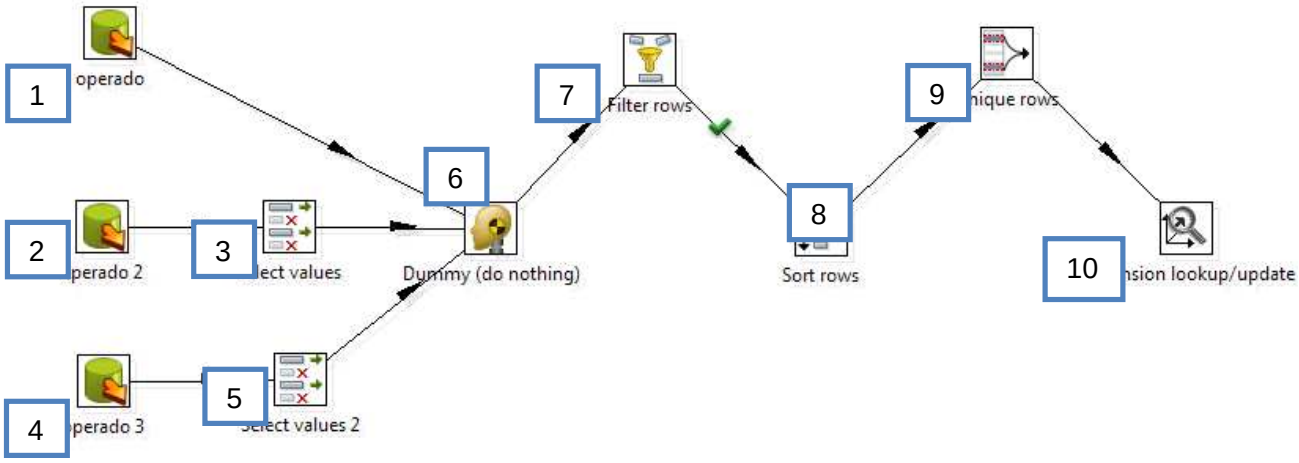


Figura 32: dim_servicio.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1-Se extraen los registros de la columna operado de la tabla Historia en el ODS.

2.- Se extraen los registros de la columna operado2 de la tabla Historia en el ODS.

3.-Se renombra el campo operado2 como operado.

4.- Se extraen los registros de la columna operado3 de la tabla Historia en el ODS.

5.- Se renombra el campo operado3 como operado.

6.-Se unifican los 3 flujos en un solo campo operado.

7.-Se filtran los registros con el campo operado NULL.

8.-Se ordenan los registros.

9.-Se eliminan los registros duplicados.

10.-Se crea la dimensión Operado y se inserta el flujo en la dimensión.

En la figura 33, se puede visualizar la transformación **dim_fecha.ktr**, con se genera la dimensión fecha.

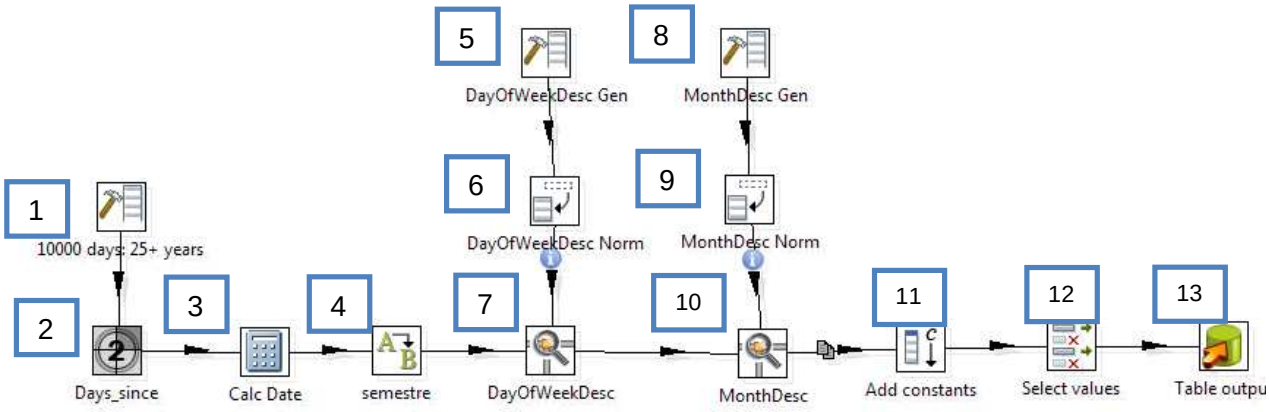


Figura 33: dim_fecha.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1.-Se generan 21000 registros con el valor 20000101 de tipo fecha y se le asigna como nombre del campo start_day.

2.-Se crea una secuencia llamada days_since del 0 al 20999.

3.-Se suma el campo start_day + days_since, para obtener las fechas desde 01-01-2000 hasta 01-01-2000 más 20999 dias.

Se obtiene el mes (Month), el año, dia del año, dia del mes, dia de la semana (Dayofweek), semana del año, de todas las fechas generadas.

Se le asigna un identificador a cada fecha llamado id_tk.

4.-Se crea un campo semestre y se le asigna 1 si el mes de la fecha está entre 1 y 6 o se le asigna 2 si el mes esta entre 7 y 12.

5.-Se crean una columna que contiene el número de dia de todos los días de la semana (1 al 7), la descripción de todos los días de la semana (LUNES a DOMINGO) y la descripción abreviada(LUN a DOM).
6.-Se utiliza el row normalizer para separar los números de dia en un campo llamado DayNr, la descripción en un campo llamado DayDesc,y la descripción abreviada en un campo llamado DayDescShort, formando una tabla.
7.-Se realiza un join de ambos flujos por el campo DayNr y Dayofweek.
8.- Se crean una columna que contiene la descripción de los meses (ENERO a DICIEMBRE), y la descripción abreviada (ENE a DIC).
9.- Se utiliza el row normalizer para separar la descripción de los meses en un campo llamado MonthDesc y la descripción abreviada en un campo llamado MonthDescShort, y crear un identificador de mes del 1 al 12 denominado MonthNr, formando una tabla .
10.- Se realiza un join de ambos flujos por el campo MonthNr y Month.
11.-Se agregan dos campos al flujo semanaepi y anioepi y se establecen en 0 por defecto. Estos dos campos indicaran la semana epidemiológica y el año epidemiológico al que pertenece la fecha.
12.-Se seleccionan los campos que conformaran la dimensión fecha.
13.-Se crea la dimensión fecha y se inserta el flujo de fechas hasta el 2057.

En la figura 34, se puede visualizar la primera parte de la transformación **fact_casos.ktr**, con se genera la tabla de hechos fact_casos.

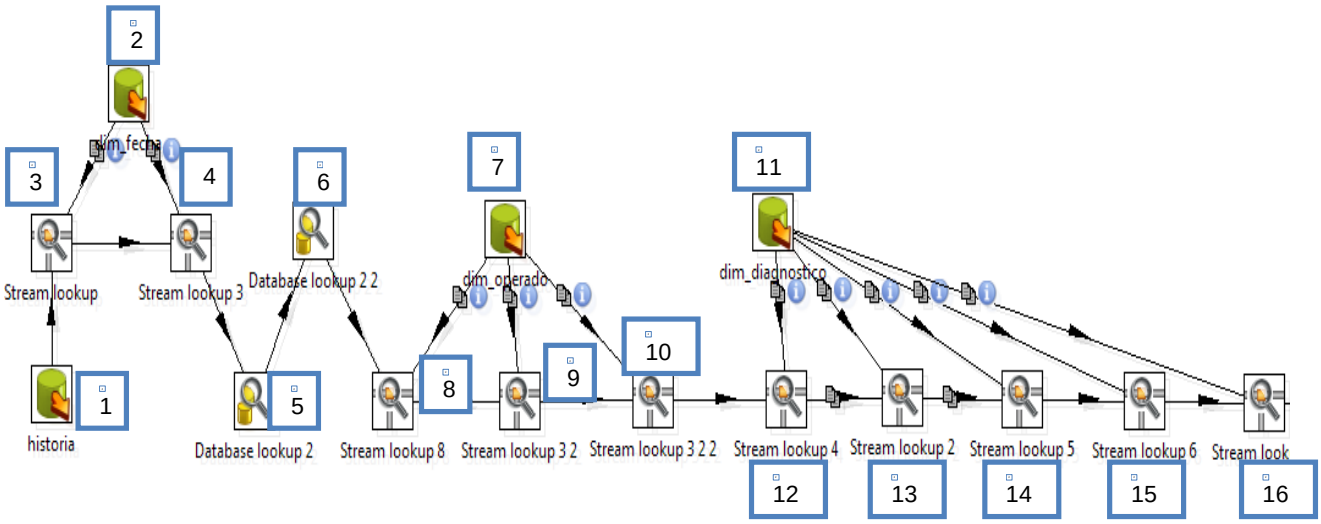


Figura 34:fact_casos.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1.-Se extrae la tabla historia del ODS.

2.-Se el campo fecha y el campo id de la dimensión fecha.

3.-Se realiza un join de ambos flujos por el campo fecha y el campo fecha_ingreso proveniente de la tabla historia.Se renombra el campo id como fecha_ingreso_1.

4.- Se realiza un join de ambos flujos por el campo fecha y el campo fecha_egreso proveniente de la tabla historia.Se renombra el campo id como fecha_egreso_1.

5.-Se realiza un join con la dimension procedencia por el cod_procedencia y el código_pro y se obtiene el campo id de la dimensión procedencia el cual se renombra como id_procedencia.

6.- Se realiza un join con la dimensión servicio por el cod_servicio y el cod_servicio y se obtiene el campo id de la dimensión procedencia el cual se renombra como id_servicio.

7.- Se extrae el campo nombre y el campo id de la dimensión operado.

8.-Se realiza un join de ambos flujos por el campo operado y el campo nombre del flujo proveniente de la dimension operado y se renombra el campo id por id_operado.
9.-Se realiza un join de ambos flujos por el campo operado y el campo nombre del flujo proveniente de la dimension operado2 y se renombra el campo id por id_operado2.
10.-Se realiza un join de ambos flujos por el campo operado3 y el campo nombre del flujo proveniente de la dimension operado y se renombra el campo id por id_operado3.
11.- Se extrae el campo codigo_dia y el campo id de la dimensión operado.
12.- Se realiza un join de ambos flujos por el campo cod_diagnostico1 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico1.
13.- Se realiza un join de ambos flujos por el campo cod_diagnostico2 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico2.
14.- Se realiza un join de ambos flujos por el campo cod_diagnostico3 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico3.
15.- Se realiza un join de ambos flujos por el campo cod_diagnostico4 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico4.
16.- Se realiza un join de ambos flujos por el campo cod_diagnostico5 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico5.

En la figura 35, se puede visualizar la segunda parte de la transformación **fact_casos.ktr**, con se genera la tabla de hechos fact_casos.

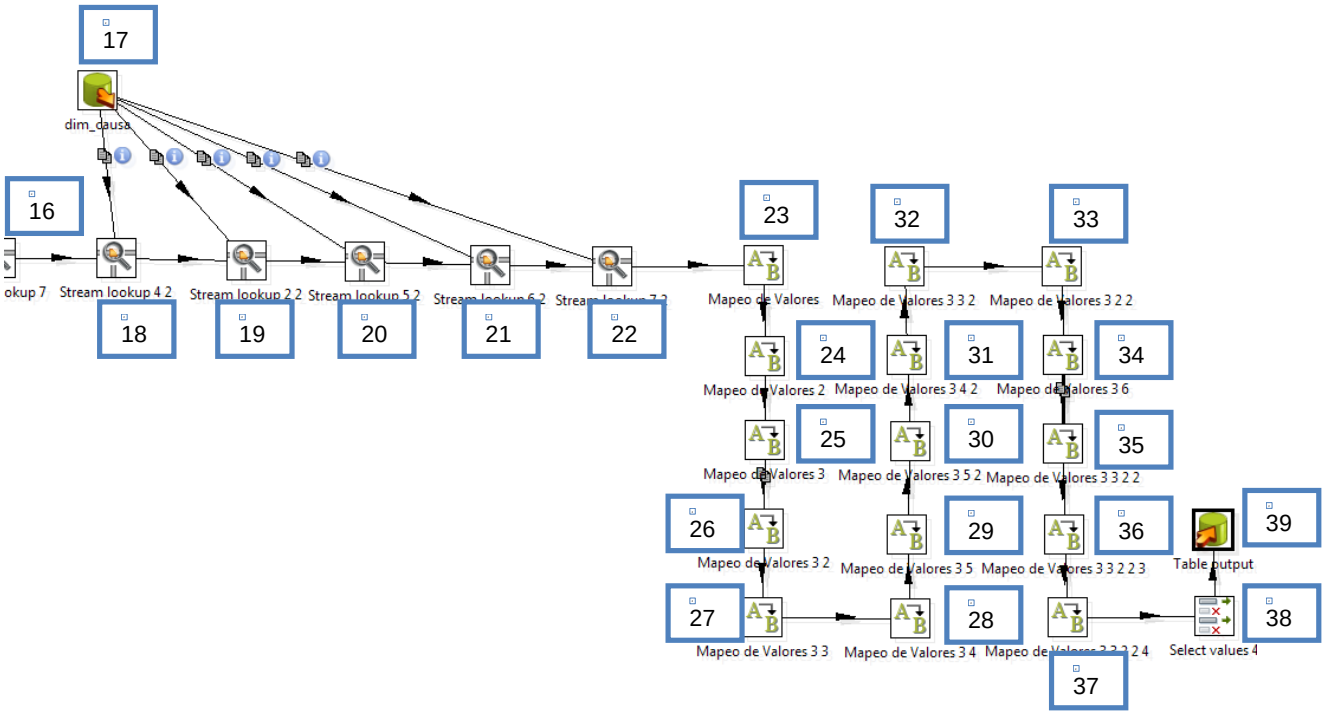


Figura 35: fact_casos.ktrparte2.

16.- Se realiza un join de ambos flujos por el campo cod_diagnostico5 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico5.

17.- Se extrae el campo codigo_causa y el campo id de la dimensión causa.

18.- Se realiza un join de ambos flujos por el campo cod_causa1 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa1.

19.- Se realiza un join de ambos flujos por el campo cod_causa2 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa2.

20.- Se realiza un join de ambos flujos por el campo cod_causa3 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa3.

21- Se realiza un join de ambos flujos por el campo cod_causa4 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa4.

21- Se realiza un join de ambos flujos por el campo cod_causa4 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa4.

22- Se realiza un join de ambos flujos por el campo cod_causa5 y el campo codigo_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa5.

23.-Si encuentra un registro en el campo fecha_ingreso_1 establece el valor 0.

24.-Si encuentra un registro en el campo fecha_egreso_1 establece el valor 0.

25.- Si encuentra un registro en el campo id_diagnostico1 establece el valor 0.

26.- Si encuentra un registro en el campo id_diagnostico2 establece el valor 0.

27.- Si encuentra un registro en el campo id_diagnostico3 establece el valor 0.

28.- Si encuentra un registro en el campo id_diagnostico4 establece el valor 0.

28.- Si encuentra un registro en el campo id_diagnostico5 establece el valor 0.

29.- Si encuentra un registro en el campo id_causa1 establece el valor 0.

30.- Si encuentra un registro en el campo id_causa2 establece el valor 0.

31.- Si encuentra un registro en el campo id_causa3 establece el valor 0.

32.- Si encuentra un registro en el campo id_causa4 establece el valor 0.

33.- Si encuentra un registro en el campo id_causa5 establece el valor 0.

34.- Si encuentra un registro en el campo id_operado establece el valor 0.

35.- Si encuentra un registro en el campo id_operado2 establece el valor 0.

36.- Si encuentra un registro en el campo id_operado3 establece el valor 0.

37.-Se remueven los campos sobrantes cod_servicio, fecha_ingreso, fecha_egreso, cod_procedencia, cod_diagnostico1, cod_diagnostico2, cod_diagnostico3, cod_diagnostico4, cod_diagnostico5, cod_causa1, cod_causa2, cod_causa3, cod_causa4, cod_causa5,anio, operado, operado2, operado3.

38.- Se crea la tabla de hechos fact_table y se inserta el flujo en dicha tabla.

A continuación se puede visualizar la transformación **casos_sumarizados.ktr(ver figura 36)**, cuyo objetivo es sumarizar los casos por procedencia, diagnóstico y semana epidemiológica de un año especifico, en función de formar los nuevos registros de la tabla de hechos casos_completos en el data warehouse.

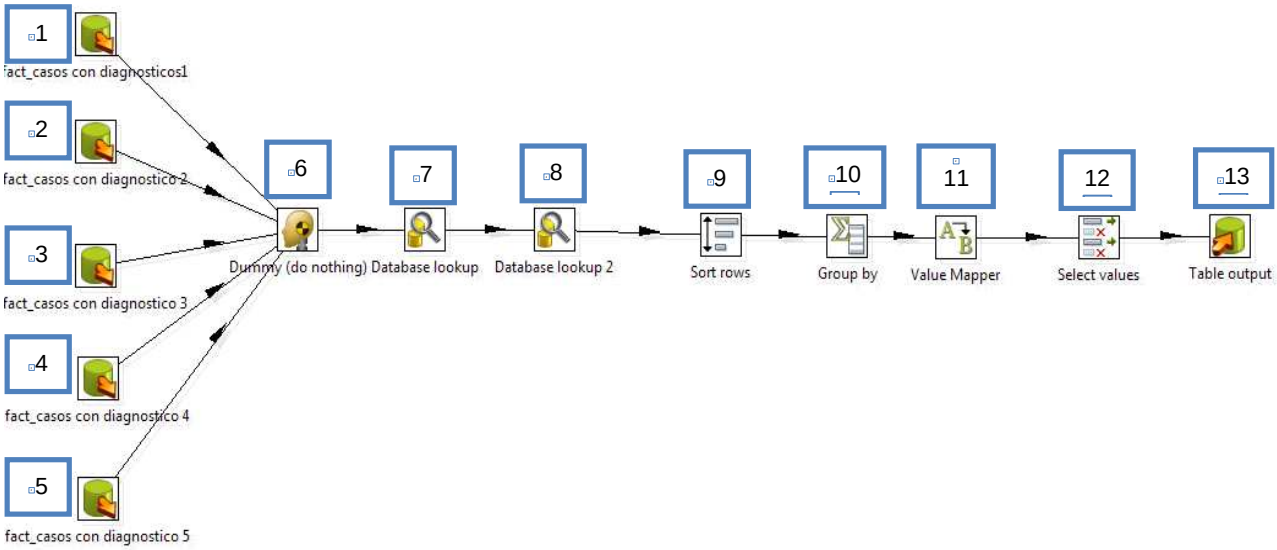


Figura 36: casos_sumarizados futuros.ktr

1.-Se extraen los registros de los campos id_diagnostico1 y fecha_ingreso_1 de la tabla fact_casos del data warehouse. Se renombra el campo id_diagnostico1 como id_diagnostico.

2.-Se extraen los registros de los campos id_diagnostico2 y fecha_ingreso_1 de la tabla fact_casos del data warehouse. Se renombra el campo id_diagnostico2 como id_diagnostico.

3.-Se extraen los registros de los campos id_diagnostico3 y fecha_ingreso_1 de la tabla fact_casos del data warehouse. Se renombra el campo id_diagnostico3 como id_diagnostico.

4.-Se extraen los registros de los campos id_diagnostico4 y fecha_ingreso_1 de la tabla fact_casos del data warehouse. Se renombra el campo id_diagnostico4 como id_diagnostico.

5.-Se extraen los registros de los campos id_diagnostico5 y fecha_ingreso_1 de la tabla fact_casos del data warehouse. Se renombra el campo id_diagnostico5 como id_diagnostico.

6.-Se unifican los 5 flujos formando una sola tabla con los campos id_diagnostico y fecha_ingreso_1.

7.-Se lleva a cabo un join con el campo id de la dimension tiempodel data warehouse y el campo fecha_ingreso_1 del flujo, y se traen los campos anioepi y semanaepi.

8.-Se lleva a cabo un join por los campos anioepi y semanaepi de la dimensión epidemiológica del data warehouse y se trae el campo id de la dimension epidemiológica.

9.- Se ordena el flujo por los campos por id_diagnostico,id_procedencia y id.

10.- Se agrupa el flujo por los campos por id_diagnostico,id_procedencia y id.Se cuentan los casos, formando un nuevo campo llamado casos.

11.-Se establece en cero el campo id que se encuentren vacios.

12.-Se renombra el campo id por semana_epidemiologica

13.-Se crea la tabla de hechos casos_completos en el data warehouse y se inserta el flujo.

En la siguiente imagen (ver figura 37) se observa la transformación **llamar a procedimiento que inserte los casos cero.ktr**

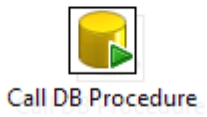


Figura 37: Llamar a procedimiento que inserte los casos cero.ktr

En esta transformación se llama al procedimiento insertar_casos() que inserta las semanas epidemiológicas que faltan en la tabla casos_completos para poder graficar el canal endémico. Cabe destacar que se insertan los registros con el campo casos establecido con valor cero. En los anexos se encuentra el procedimiento de una forma más detallada.

En la figura 38 se observa la transformación **llamar a procedimiento que calcula los percentiles.ktr**

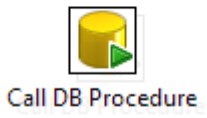


Figura 38: Llamar a procedimiento que calcula los percentiles.ktr

En esta transformación se llama al procedimiento calcular_percentiles() que calcula los percentiles 25,50 y 75, para cada uno de los registros de la tabla casos_completos del Datamart, tomando en cuenta los registros que se encuentren en la tabla de hecho fact_casos del Datamart ,

correspondiente a los casos ocurridos los 7 años anteriores de un diagnostico especifico en una procedencia y semana epidemiológica de un año especifico.

ETL Staging Area a Datamart

Con las distintas transformaciones explicadas anteriormente se logró realizar la primera carga de datos desde el 2002 hasta el 2014. Sin embargo, es necesario que exista un proceso automatizado que permita realizar la extracción, transformación y carga de datos generados en el futuro, es por ello que se decidió hacer uso de los Jobs.

Además, es necesario apoyarse de una estructura de datos que permita almacenar de forma temporal los datos que serán cargados a futuro. Es por ello que se creó una base de datos llamada stagingarea que contiene las siguientes tablas(ver figura 39).

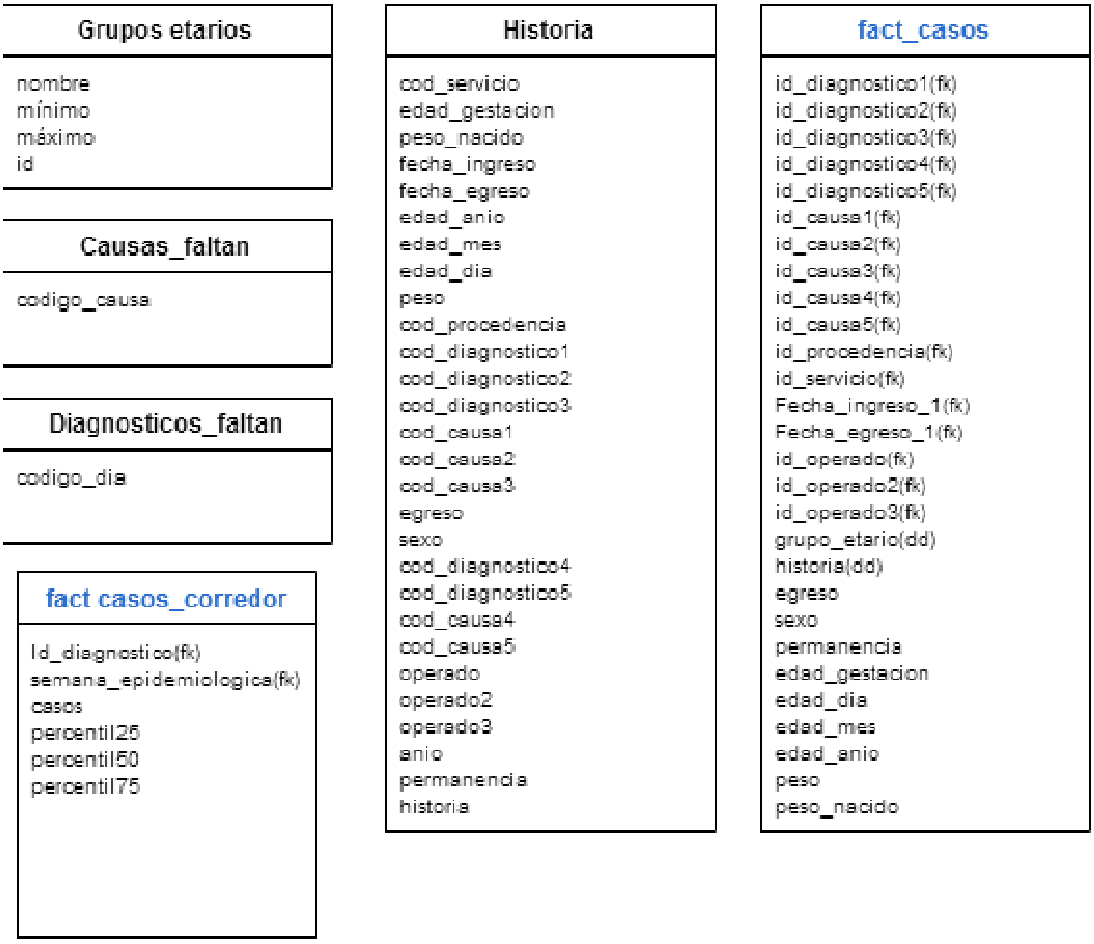


Figura 39: Stagingarea.

Es importante destacar que los Jobs permiten incluir diferentes transformaciones, generando un flujo de trabajo con el cual es posible monitorear el proceso completo, mediante captura de errores y notificación de errores. Así mismo Spoon contiene un Scheduler o planificador que brinda la posibilidad de configurar la frecuencia de ejecución de los Jobs.

A continuación se muestra una imagen que permite visualizar el Job creado para generar las cargas futuras(ver figura 40).

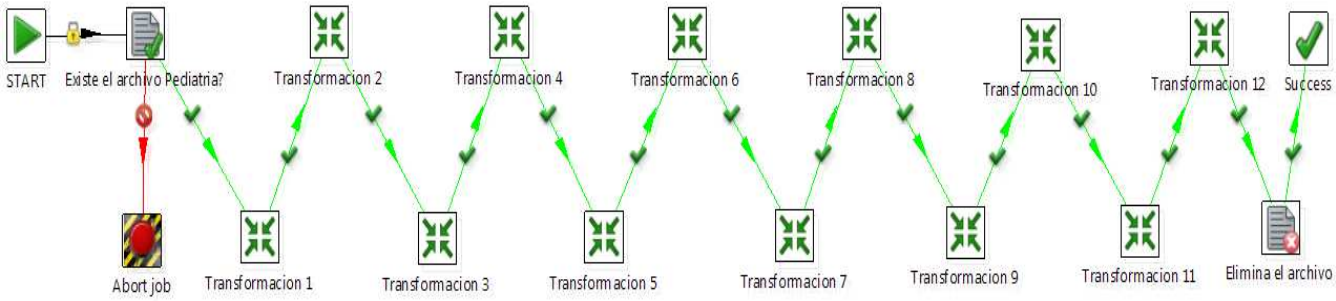


Figura 40: Job carga futura.

El job comienza con la verificación de que exista el nuevo archivo Acess pediatria.mdb, si el archivo no existe se aborta el job. Si el archivo existe en el directorio, entonces ejecuta las transformaciones que se explican a continuación.

Transformacion1

En la figura 41 se visualiza la transformación **carga futura.ktr**

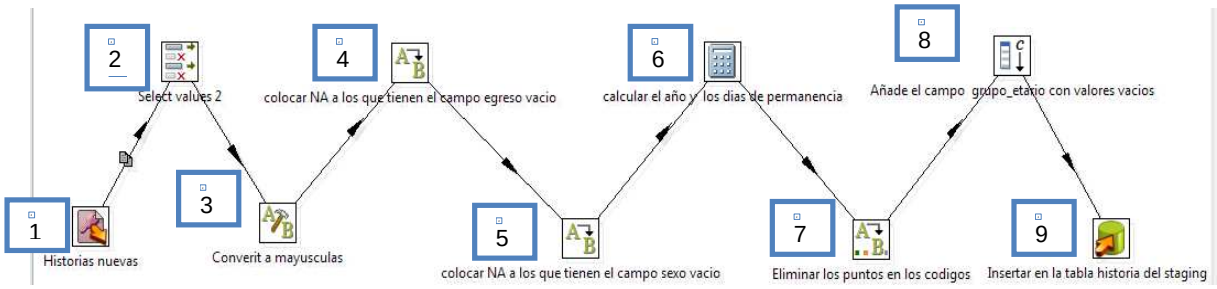


Figura 41: carga futura.ktr.

A continuación se describe la transformación anterior.

1.-Se extrae el archivo Access pediatría.

2.-Se establecen los tipos de datos y se ordenan los campos para que concuerde con la tabla de hechos fact table.
3.-Se convierten todos los códigos de diagnóstico, causa, operado, servicio y procedencia a mayúscula.
4.-Se reemplazan los registros caracteres f, w, W por el carácter M en el campo egreso. Se coloca NA (no asignado) a los registros vacios del campo egreso.
5.- Se coloca NA (no asignado) a los campos vacíos del campo sexo.
6.-Se calcula la permanencia de los casos en el hospital restando la fecha de egreso menos la fecha de ingreso.
7.-Se eliminan los puntos en los 5 códigos de diagnóstico y los 5 códigos de causa para que concuerden con los códigos establecidos por el CIE 10.
8.-Se añade el campo grupo_etario que contendrá la clasificación de los casos por grupo de edad. Se establecen en NULL por defecto.
9.-Se crea la tabla historia en el staging área y se inserta el flujo en la tabla.

Transformación 2

En la figura 42 se visualiza la transformación llamar a procedimiento que asigna grupo etario.ktr

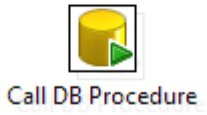


Figura 42: llamar a procedimiento que asigna grupo etario.ktr

En la transformación anterior se asigna el grupo etario a los casos de la tabla historia en la staging área. A groso modo lo que se lleva a cabo es unir la tabla grupos_etarios dentro del ODS con los registros de la tabla

historia del staging area comparando la edad en años. En los anexos se encuentra el procedimiento a detalle.

Transformación 3

En la figura 43 se puede observar la transformacion **obtener los codigos de diagnóstico de 3 digitos faltantes.ktr**, cuya finalidad es obtener los códigosdiagnósticos de 3 dígitos que no se encuentran en la tabla diagnóstico del ODS.

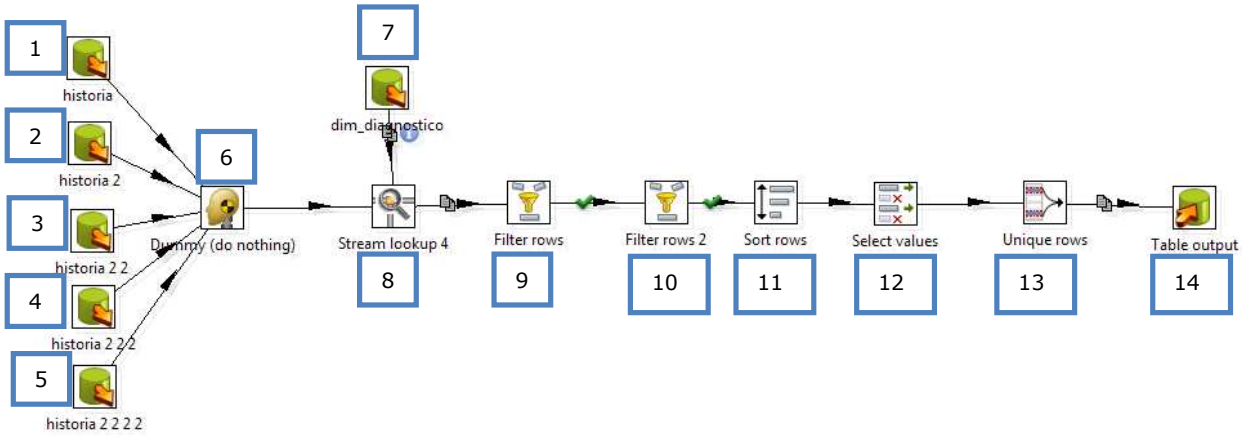


Figura 43: obtener los codigos de diagnóstico de 3 digitosfaltantes.ktr

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1.-Se extrae el cod_diagnostico1 de la tabla historia del staging area y la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico1) y se renombra como longitud.

2.-Se extrae el cod_diagnostico2 de la tabla historia del staging area , se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico2) y se renombra como longitud.

3.- Se extrae el cod_diagnostico3 de la tabla historia del staging area , se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico3) y se renombra como longitud.
4.- Se extrae el cod_diagnostico4 de la tabla historia de la tabla historia del staging area, se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico4) y se renombra como longitud.
5.- Se extrae el cod_diagnostico5 de la tabla historia del staging area de la tabla historia del staging area, se renombra como cod_diagnostico y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_diagnostico5) y se renombra como longitud.
6.-Se unen los 5 flujos en una sola tabla.
7.-Se extrae el campo id y código_dia de la dimensión diagnóstico del Data warehouse.
8.-Se realiza un join entre ambos flujos por el campo código_dia y el campo cod_diagnostico y se renombra el campo id como id_diagnostico1.
9.-Filtra los registros cuyo campo id_diagnostico1 no sean vacio.
10.-Obtiene los registros que tengan longitud 3.
11.-Ordena los registros por cod_diagnostico.
12.-Remueve el campo longitud.
13.-Elimina los duplicados.
14.- Crea la tabla diagnosticos_faltan en el staging área con los códigos de 3 dígitos que no se encuentran en la dimension diagnóstico del data warehouse Cisp_olap.

Transformación 4

En la figura 44, se aprecia la transformación llamar a procedimiento que corrige los codigos de 3 digitos.ktr, cuya finalidad es corregir los codigos de diagnosticos que vienen con errores desde el archivo fuente, de manera que concuerden con los codigos establecidos por el CIE 10.

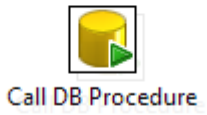


Figura 44: llamar a procedimiento que corrige los codigos de 3 digitos.ktr

En esta transformación se llama al procedimiento completar_diagnosticos() que corrige los diagnósticos de 3 dígitos que fueron almacenados en la tabla diagnósticos faltan agregándoles un cero al final. Si existen los códigos dentro de la dim_diagnostico del Data warehouse Cisp_olap , entonces actualiza los campos de diagnóstico en la tabla historia dentro del staging área con los nuevos códigos de cuatro dígitos.

Transformación 5

En la siguiente imagen se visualiza la transformación obtener los **códigos de causa de 3 digitos faltantes.ktr (ver figura 45).**

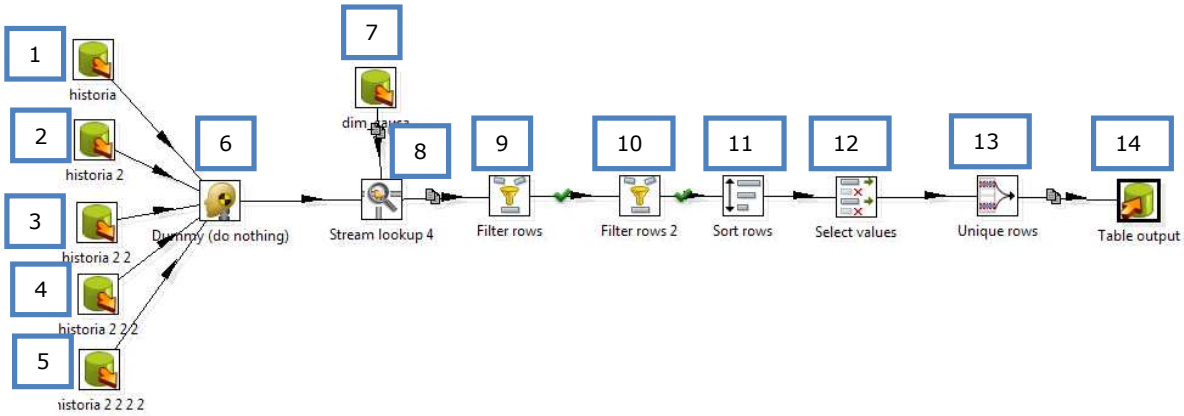


Figura 45: códigos de causa de 3 dígitos faltantes.ktr.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1.-Se extrae el cod_causa1 de la tabla historia del Staging area , se renombra como como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa1) y se renombra como longitud.

2.-Se extrae el cod_causa2 de la tabla historia del Staging area , se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa2) y se renombra como longitud.

3.- Se extrae el cod_ cod_causa 3 de la tabla historia del Staging area , se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa 3) y se renombra como longitud.

4.- Se extrae el cod_causa4 de la tabla historia del ODS de la tabla historia del Staging area, se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa4) y se renombra como longitud.

5.- Se extrae el cod_causa5 de la tabla historia de la tabla historia del Staging area, se renombra como cod_causa y se extrae la longitud del código usando la función CHARACTER_LENGTH(cod_causa5) y se renombra como longitud.

6.-Se unen los 5 flujos en una sola tabla.

7.-Se extrae el campo id y código_causa de la dimensión diagnóstico del Data warehouse.

8.-Se realiza un join entre ambos flujos por el campo código_causa y el campo cod_causa y se renombra el campo id como id_causa1.

9.-Filtra los registros cuyo campo id_causa1 no sean vacio.

10.-Obtiene los registros que tengan longitud 3.

11.-Ordena los registros por cod_causa.

12.-Remueve el campo longitud.

13.-Elimina los duplicados.

14.- Crea la tabla causas_faltan en el staging área con los códigos de 3 dígitos que no se encuentran en la dimensión diagnóstico del data warehouse Cisp_olap.

Transformación 6

En la figura 46se aprecia la transformación **llamar a procedimiento que corrige los codigos de causa de 3 digitos.ktr**, cuya finalidad es corregir los códigos de diagnósticos que vienen con errores desde el archivo fuente, de manera que concuerden con los códigos establecidos por el CIE 10.

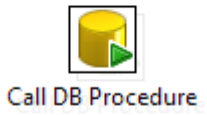


Figura 46: llamar a procedimiento que corrige los codigos de causa de 3 digitos.ktr

En esta transformación se llama al procedimiento completar_causas() que corrige los códigos de causa de 3 dígitos que fueron almacenados en la tabla causas_faltan agregándoles un cero al final. Si existen los códigos dentro de la dim_causa del Data warehouse Cisp_olap , entonces actualiza los campos de ids de causas en la tabla historia dentro del staging área con los nuevos códigos de causa de cuatro dígitos.

Transformación 7

En la imagen a continuación, se aprecia la primera parte de la transformación**fact_casos futura.ktr (ver figura 47).**

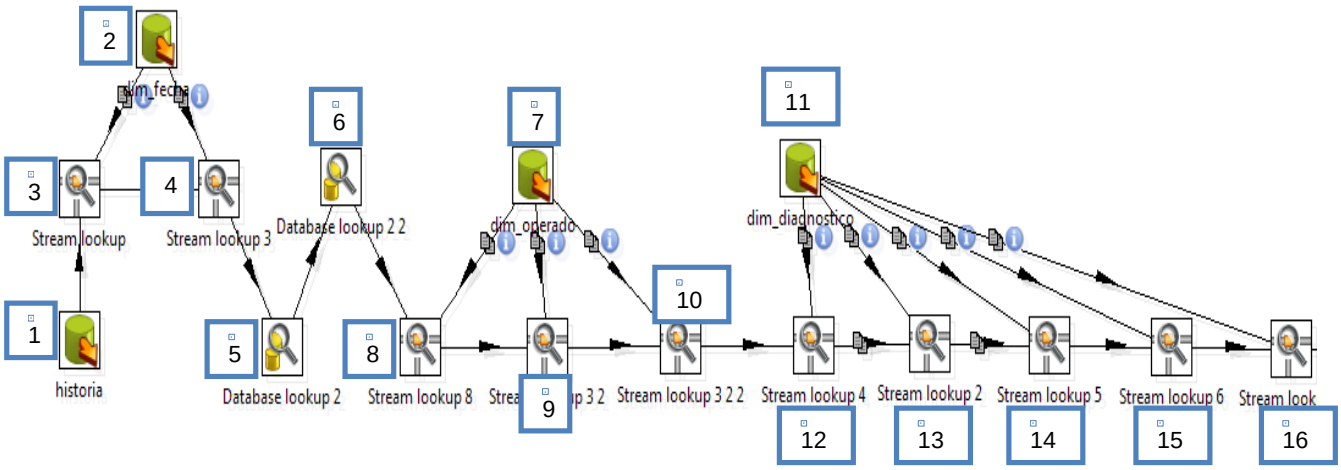


Figura 47: fact_casos futura.ktr primera parte.

A continuación se explican los pasos de la transformación que se aprecia en la figura anterior.

1.-Se extrae la tabla historia del Staging area.

2.-Se extrae el campo fecha y el campo id de la dimensión fecha.

3.-Se realiza un join de ambos flujos por el campo fecha y el campo fecha_ingreso proveniente de la tabla historia. Se renombra el campo id como fecha_ingreso_1.

4.- Se realiza un join de ambos flujos por el campo fecha y el campo fecha_egreso proveniente de la tabla historia.Se renombra el campo id como fecha_egreso_1.

5.-Se realiza un join con la dimension procedencia por el cod_procedencia y el código_pro y se obtiene el campo id de la dimensión procedencia el cual se renombra como id_procedencia.

6.- Se realiza un join con la dimensión servicio por el cod_servicio y el cod_servicio y se obtiene el campo id de la dimensión procedencia el cual se renombra como id_servicio.

7.- Se extrae el campo nombre y el campo id de la dimensión operado.

8.-Se realiza un join de ambos flujos por el campo operado y el campo nombre del flujo proveniente de la dimension operado y se renombra el campo id por id_operado.
9.-Se realiza un join de ambos flujos por el campo operado y el campo nombre del flujo proveniente de la dimension operado2 y se renombra el campo id por id_operado2.
10.-Se realiza un join de ambos flujos por el campo operado3 y el campo nombre del flujo proveniente de la dimension operado y se renombra el campo id por id_operado3.
11.- Se extrae el campo codigo_dia y el campo id de la dimensión operado.
12.- Se realiza un join de ambos flujos por el campo cod_diagnostico1 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico1.
13.- Se realiza un join de ambos flujos por el campo cod_diagnostico2 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico2.
14.- Se realiza un join de ambos flujos por el campo cod_diagnostico3 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico3.
15.- Se realiza un join de ambos flujos por el campo cod_diagnostico4 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico4.
16.- Se realiza un join de ambos flujos por el campo cod_diagnostico5 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico5.

En la figura 48, se puede visualizar la segunda parte de la transformación **fact_casos futura.ktr**, con se genera la tabla de hechos fact_casos.

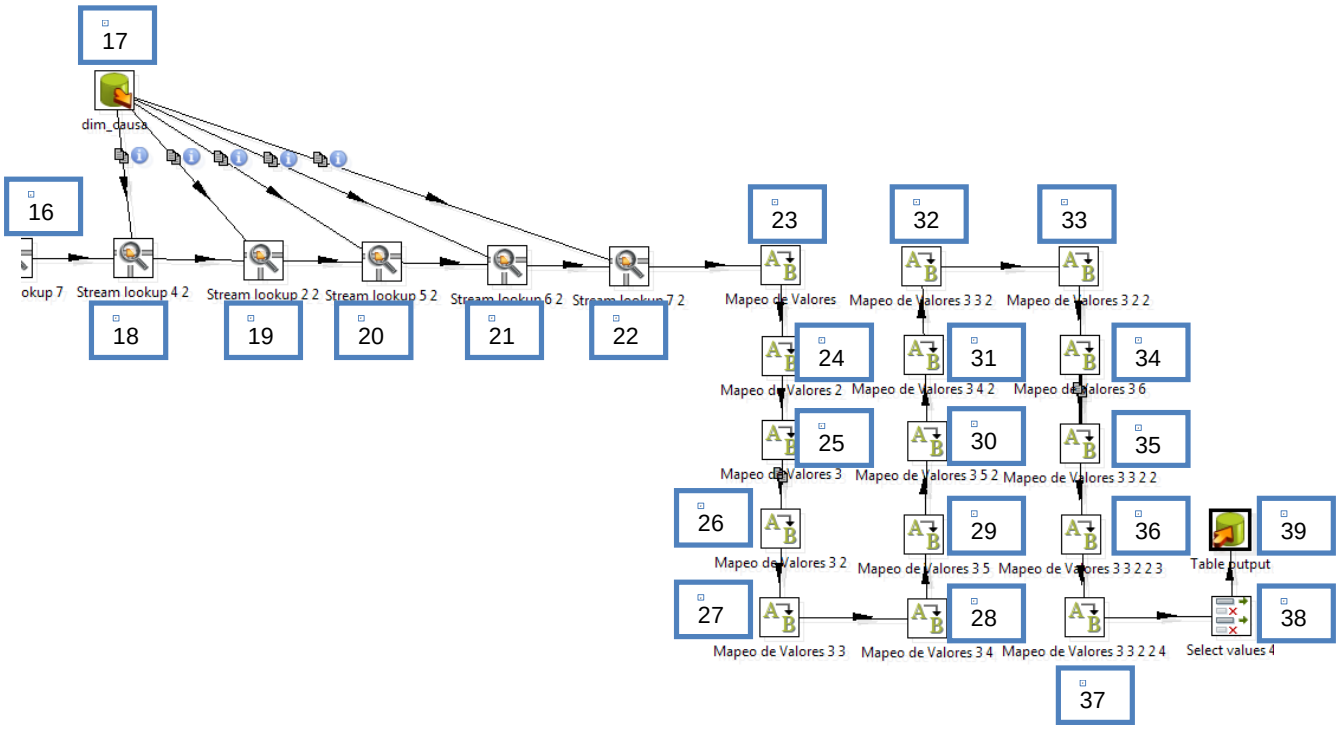


Figura 48:fact_casos futura.ktr segunda parte.

16.- Se realiza un join de ambos flujos por el campo cod_diagnostico5 y el campo código_dia del flujo proveniente de la dimensión diagnóstico y se renombra el campo id por id_diagnostico5.

17.- Se extrae el campo codigo_causa y el campo id de la dimensión causa.

18.- Se realiza un join de ambos flujos por el campo cod_causa1 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa1.

19.- Se realiza un join de ambos flujos por el campo cod_causa2 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa2.

20.- Se realiza un join de ambos flujos por el campo cod_causa3 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa3.

21- Se realiza un join de ambos flujos por el campo cod_causa4 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa4.

21- Se realiza un join de ambos flujos por el campo cod_causa4 y el campo código_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa4.

22- Se realiza un join de ambos flujos por el campo cod_causa5 y el campo codigo_causa del flujo proveniente de la dimensión causa y se renombra el campo id por id_causa5.

23.-Si encuentra un registro en el campo fecha_ingreso_1 establece el valor 0.

24.-Si encuentra un registro en el campo fecha_egreso_1 establece el valor 0.

25.- Si encuentra un registro en el campo id_diagnostico1 establece el valor 0.

26.- Si encuentra un registro en el campo id_diagnostico2 establece el valor 0.

27.- Si encuentra un registro en el campo id_diagnostico3 establece el valor 0.

28.- Si encuentra un registro en el campo id_diagnostico4 establece el valor 0.

28.- Si encuentra un registro en el campo id_diagnostico5 establece el valor 0.

29.- Si encuentra un registro en el campo id_causa1 establece el valor 0.

30.- Si encuentra un registro en el campo id_causa2 establece el valor 0.

31.- Si encuentra un registro en el campo id_causa3 establece el valor 0.

32.- Si encuentra un registro en el campo id_causa4 establece el valor 0.

33.- Si encuentra un registro en el campo id_causa5 establece el valor 0.

34.- Si encuentra un registro en el campo id_operado establece el valor 0.

35.- Si encuentra un registro en el campo id_operado2 establece el valor 0.

36.- Si encuentra un registro en el campo id_operado3 establece el valor 0.

37.-Se remueven los campos sobrantes cod_servicio, fecha_ingreso, fecha_egreso, cod_procedencia, cod_diagnostico1, cod_diagnostico2, cod_diagnostico3, cod_diagnostico4, cod_diagnostico5, cod_causa1, cod_causa2, cod_causa3, cod_causa4, cod_causa5,anio, operado, operado2, operado3.

38.- Se crea la tabla de hechos fact_table en el staging área y se inserta el flujo en dicha tabla.

Transformación 8

A continuación se puede visualizar la transformación **casos_sumarizados futuros.ktr**, cuyo objetivo es sumarizar los casos por procedencia, diagnóstico y semana epidemiológica de un año especifico, en función de formar los nuevos registros de la tabla de hechos casos_completos(ver figura 49).

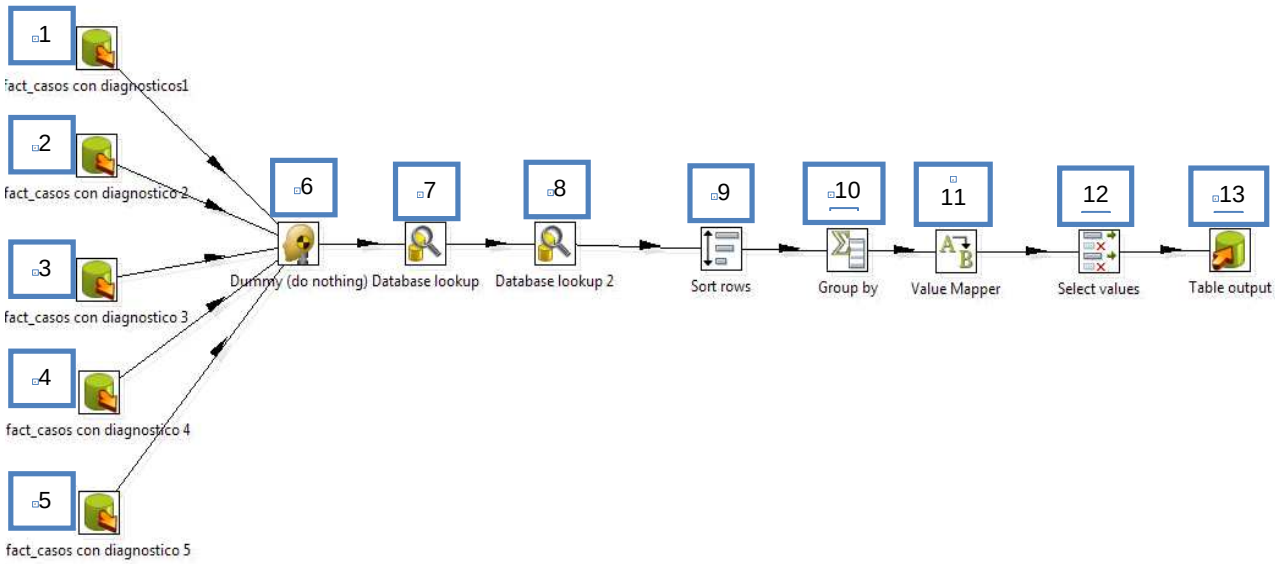


Figura 49: casos_sumarizados futuros.ktr

1.-Se extraen los registros de los campos id_diagnostico1 y fecha_ingreso_1 de la tabla fact_casos de la staging área. Se renombra el campo id_diagnostico1 como id_diagnostico.
2.-Se extraen los registros de los campos id_diagnostico2 y fecha_ingreso_1 de la tabla fact_casos de la staging área. Se renombra el campo id_diagnostico2 como id_diagnostico.
3.-Se extraen los registros de los campos id_diagnostico3 y fecha_ingreso_1 de la tabla fact_casos de la staging área. Se renombra el campo id_diagnostico3 como id_diagnostico.
4.-Se extraen los registros de los campos id_diagnostico4 y fecha_ingreso_1 de la tabla fact_casos de la staging área. Se renombra el campo id_diagnostico4 como id_diagnostico.
5.-Se extraen los registros de los campos id_diagnostico5 y fecha_ingreso_1 de la tabla fact_casos de la staging área. Se renombra el campo id_diagnostico5 como id_diagnostico.
6.-Se unifican los 5 flujos formando una sola tabla con los campos id_diagnostico y fecha_ingreso_1.
7.-Se lleva acabo un join con el campo id de la dimension tiempodel data warehouse y el campo fecha_ingreso_1 del flujo, y se traen los campos anioepi y semanaepi.
8.-Se lleva a cabo un join por los campos anioepi y semanaepi de la dimensión epidemiológica del data warehouse y se trae el campo id de la dimension epidemiológica.
9.- Se ordena el flujo por los campos por id_diagnostico,id_procedencia y id.
10.- Se agrupa el flujo por los campos por id_diagnostico y id.Se cuentan los casos, formando un nuevo campo llamado casos.
11.-Se establece en cero el campo id que se encuentren vacios.
12.-Se renombra el campo id por semana_epidemiologica
13.-Se crea la tabla casos_completos en la staging área y se inserta el flujo.

Transformación 9

En la figura 50, se observa la transformación **llamar a procedimiento que inserte los casos cero.ktr**

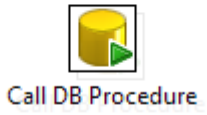


Figura 50: Llamar a procedimiento que inserte los casos cero.ktr

En esta transformación se llama al procedimiento insertar_casos() que inserta las semanas epidemiológicas que faltan en la tabla fact_casos_corredor para poder graficar el canal endémico. Cabe destacar que se insertan los registros con el campo casos, establecido con valor cero. En los anexos se encuentra el procedimiento de una forma más detallada.

Transformación 10

En la figura 51, se observa la transformación **llamar a procedimiento que calcula los percentiles.ktr**

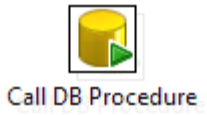


Figura 51: Llamar a procedimiento que calcula los percentiles.ktr

En esta transformación se llama al procedimiento calcular_percentiles() que calcula los percentiles 25,50 y 75, para cada uno de los registros de

la tabla fact_casos_corredor del staging area, tomando en cuenta los registros que se encuentren en la tabla de hecho fact_casos del Datamart , correspondiente a los casos ocurridos los 7 años anteriores de un diagnostico especifico en una procedencia y semana epidemiológica de un año especifico.

Transformación 11

En la figura 52, se observa la transformación **insertar casos en la tabla fact_casos.ktr**

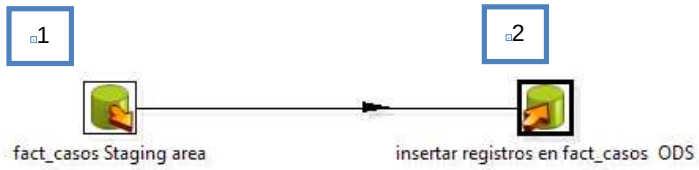


Figura 52: insertar casos en la tabla fact_casos.ktr

1.-Se extrae la tabla fact_casos del staging área.
2.-Se insertan los registros en la tabla fact_casos del Datamart cisp_olap.

Transformación 12

En la figura 53 se observa la transformación **llamar a procedimiento que inserta los nuevos fact_casos_completos en la tabla casos_completos.ktr**

Figura 53: llamar a procedimiento que inserta los nuevos casos completos en la tabla fact_casos_corredor.ktr

En esta transformación se llama al procedimiento insertar_casos_completos () que inserta los casos_completos del staging área en la tabla casos_completos del Datamart.

2.5 Especificación y desarrollo de aplicaciones de usuario final.

Las aplicaciones de BI son la cara visible de la inteligencia de negocios: los informes y aplicaciones de análisis proporcionan información útil a los usuarios. Las aplicaciones de BI incluyen un amplio espectro de tipos de informes y herramientas de análisis, que van desde informes simples de formato fijo a sofisticadas aplicaciones analíticas que usan complejos algoritmos e información del dominio. Kimball divide a estas aplicaciones en dos categorías basadas en el nivel de sofisticación, y les llama informes estándar y aplicaciones analíticas.

A continuación se especifican las herramientas utilizadas para generación de reportes y análisis.

2.5.1 Informes estándar

Los informes estándar son la base del espectro de aplicaciones de BI. Por lo general son informes relativamente simples, de formato predefinido, y parámetros de consulta fijos. Este tipo de aplicaciones son el caballo de batalla de la BI de la empresa.

Son informes que los usuarios usan día a día. La mayor parte de lo que piden las personas durante el proceso de definición de requisitos se clasificaría como informes estándar. Por eso es conveniente desarrollar un conjunto de informes estándar en el ciclo de vida del proyecto.

Es por ello que para la realización de los mismos se escogió la herramienta Pentaho Report Designer, el cual aporta facilidad de diseño y parametrización de reportes.

A continuación se muestran cada uno de los reportes generados.

Casos de ENO

Lista el número de casos de diagnósticos de ENO (enfermedades de notificación obligatoria) que ocurren en una semana epidemiológica determinada correspondiente a un año especifico y los casos registrados para la semana epidemiológica anterior y sus respectivos porcentajes de ocurrencia (ver figura 54).

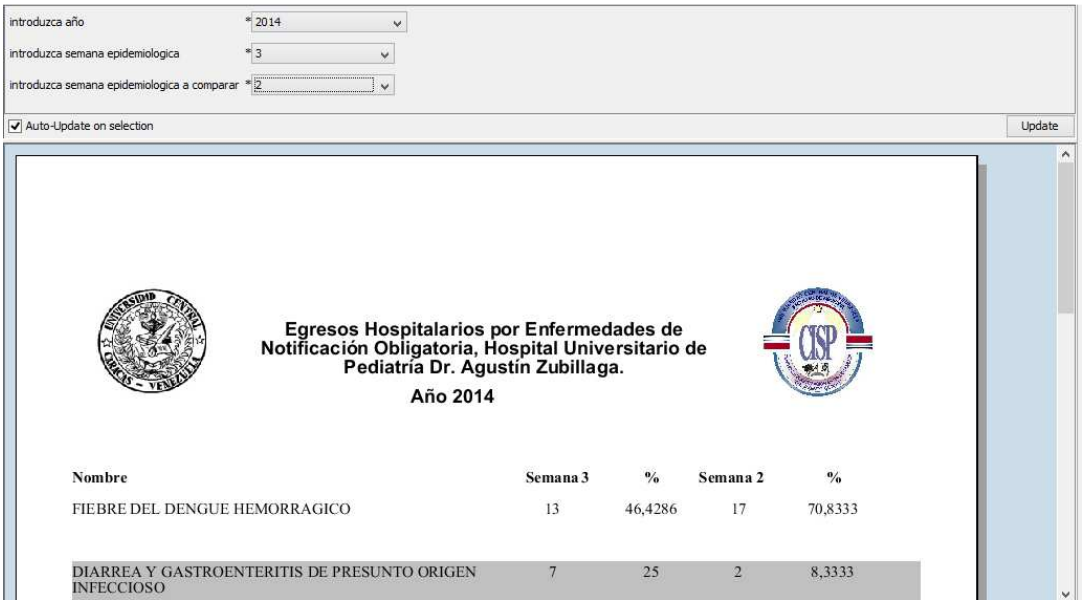


Figura 54: Casos de ENO

Número de egresos por grupos de edad

Permite conocer el número de egresos por grupos de edad al introducir un año específico. Además muestra un histograma que permite visualizar de mejor forma el reporte (ver figura 55).

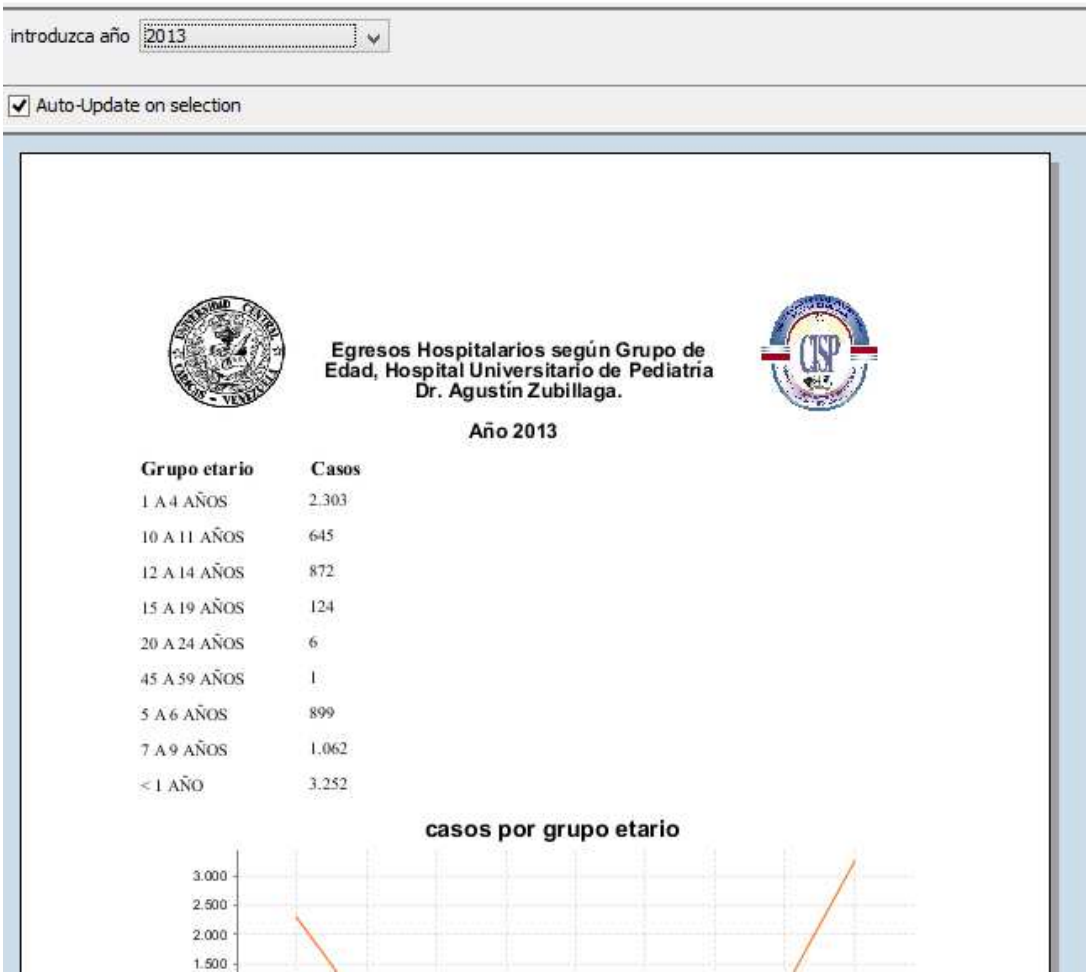


Figura 55:Número de egresos por grupos de edad.

Número de casos por procedencia.

Permite listar el número de casos por procedencia (Municipios del Edo. Lara) de una enfermedad (diagnostico) especifica en un año determinado(ver figura 56).

Figura 56: Número de casos por procedencia.

Morbilidad y mortalidad especifica

Permite listar las principales causas de morbilidad y mortalidad para un año determinado y el número egresos vivos, muertos y totales, por grupos etarios (ver figura 57).

Código	Descripcion	< 1 AÑO				1 A 4 AÑOS				5 A 6 AÑOS				7 A 9 AÑOS			
		F	M	NA	T	F	M	NA	T	F	M	NA	T	F	M	NA	T
A039	SHIGELOSIS DE TIPO NO ESPECIFICADO	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
A044	OTRAS INFECCIONES INTESTINALES DEBIDAS A ESCHERICHIA COLI	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
A049	INFECCION INTESTINAL BACTERIANA, NO ESPECIFICADA	4	4	0	8	9	11	0	20	0	0	0	0	1	1	0	2
A050	INTOXICACION ALIMENTARIA ESTAFILOCOCICA	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
A059	INTOXICACION ALIMENTARIA BACTERIANA, NO ESPECIFICADA	0	0	0	0	0	3	0	3	0	0	0	0	0	0	0	0

Figura 57: Morbilidad y mortalidad específica.

Morbilidad y mortalidad por grupo

Permite conocer las principales causas de morbilidad y mortalidad por grupo de diagnósticos, para un rango de fechas determinado y el número egresos vivos, muertos y totales, por grupos etarios(ver figura 58).

Grupo	Descripcion	< 1 AÑO				1 A 4 AÑOS				5 A 6 AÑOS				7 A 9 AÑOS				10 A 11 AÑOS				12 A 14 AÑOS			
		F	M	NA	T	F	M	NA	T	F	M	NA	T	F	M	NA	T	F	M	NA	T	F	M	NA	T
A00-A09	ENFERMEDADES INFECCIOSAS INTESTINALES	41	49	0	90	40	60	0	100	6	8	0	14	4	1	0	5	1	7	0	8	1	1	0	2
A15-A19	TUBERCULOSIS	2	0	0	2	1	4	0	5	0	2	0	2	2	0	0	2	0	0	0	0	0	0	0	0
A20-A28	CIERTAS ZOONOSIS BACTERIANAS	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	2
A30-A49	OTRAS ENFERMEDADES BACTERIANAS	172	203	0	375	32	40	0	72	6	5	0	11	3	6	0	9	2	3	0	5	1	3	0	4
A50-A64	INFECCIONES CON MODO DE TRANSMISIÓN PREDOMINANTEMENTE SEXUAL	12	19	0	31	0	1	0	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0

Figura 58: Morbilidad y mortalidad por grupo.

Morbilidad y mortalidad por accidentes, agrupados por precedencia.

Permite conocer el número de casos de causas externas (accidentes) por procedencia (Municipios del Edo. Lara). En la figura 59 y 60, se puede apreciar el reporte.

Figura 59: Morbilidad y mortalidad por accidentes, agrupados por precedencia.

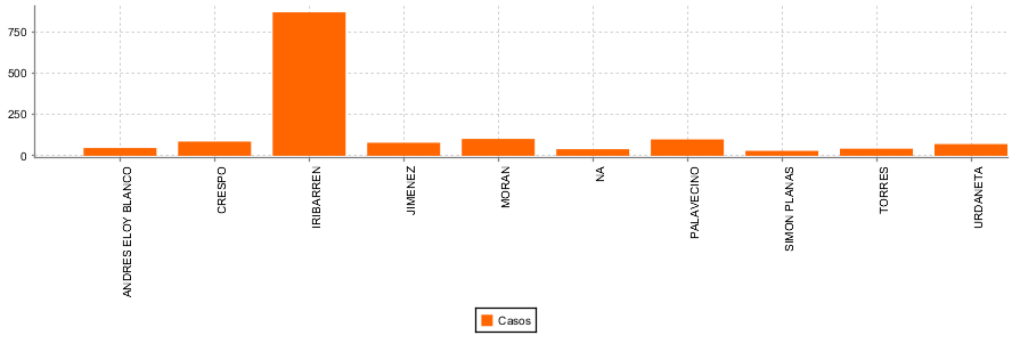


Figura 60: Grafico morbilidad y mortalidad por accidentes, agrupados por precedencia.

Morbilidad y mortalidad por accidentes agrupados por sexo.

Permite conocer el número de casos de causas externas (accidentes) por sexo para un año determinado (ver figura 61).

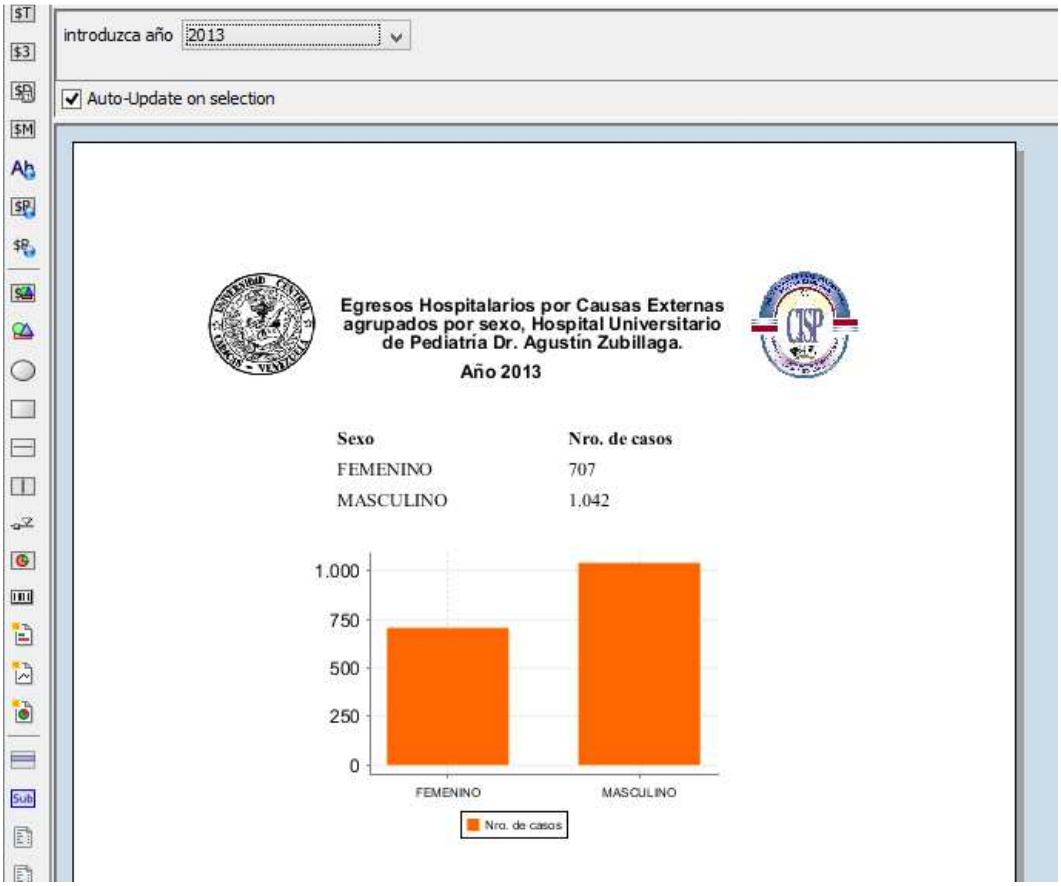


Figura 61: Morbilidad y mortalidad por accidentes agrupados por sexo.

Enfermedades de Notificación obligatoria

Permite conocer el número de casos acumulados, el porcentaje y el promedio de estancia y letalidad (número de muertes sobre el total de

casos, vivos y muertos), de las ENO hasta una semana epidemiológica de un año especifico (ver figura 62).

Diagnosticos	Casos	%	Promedio de Letalidad	Promedio de Estancia
FIEBRE DEL DENGUE HEMORRAGICO	193	25,9758	0,0052	5,0933
NEUMONIA, NO ESPECIFICADA	95	12,786	0,1474	13,0737
DIARREA Y GASTROENTERITIS DE PRESUNTO ORIGEN INFECCIOSO	92	12,3822	0,0326	9,1848
TOS FERINA, NO ESPECIFICADA	74	9,9596	0,0135	11,1757
OTRAS NEUMONIAS, DE MICROORGANISMO NO ESPECIFICADO	39	5,249	0,0256	20,7692

Figura 62: Enfermedades de Notificación obligatoria.

Consolidado semanal de enfermedades y eventos de notificación obligatoria.

Permite conocer el número de casos de las ENO por grupos etarios para una semana epidemiológica específica (ver figura 63).

Diagnosticos(ENO)	<1	1 a 4	5 a 6	7 a 9	10 a 11	12 a 14	15 a 19	20 a 24	25 a 44	45 a 59	60 a 64	65 y +	total
ABSCESO AMEBIANO DEL HIGADO	0	0	1	2	1	0	0	0	0	0	0	0	4
AMEBIASIS, NO ESPECIFICADA	2	3	0	1	0	0	0	0	0	0	0	0	6
AMIGDALITIS AGUDA, NO ESPECIFICADA	0	0	2	0	0	0	0	0	0	0	0	0	2
BRONCONEUMONIA, NO ESPECIFICADA	156	160	12	9	2	4	2	0	0	0	0	0	345
BRONQUIOLITIS AGUDA, NO ESPECIFICADA	11	0	0	0	0	0	0	0	0	0	0	0	11

Figura 63: Consolidado semanal de enfermedades y eventos de notificación obligatoria.

Enfermedades de Notificación obligatoria acumulativa

Permite conocer número de casos de las ENO acumulados hasta una semana epidemiológica de un año específico, el número de casos acumulados correspondientes a la semana anterior y el número de casos que han ocurrido en el transcurso del año. Además se comparara con el año anterior(ver figura 64).

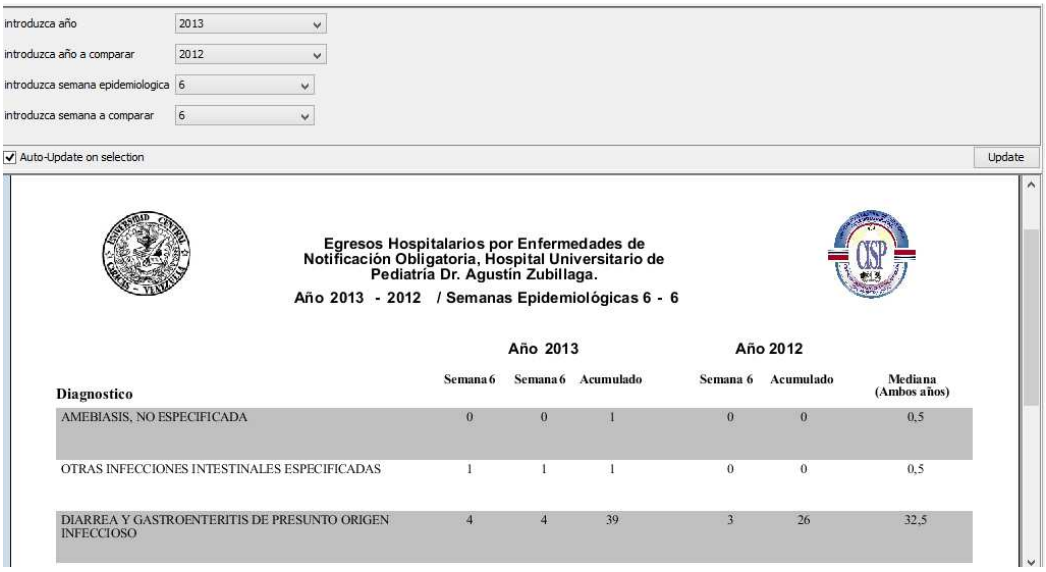
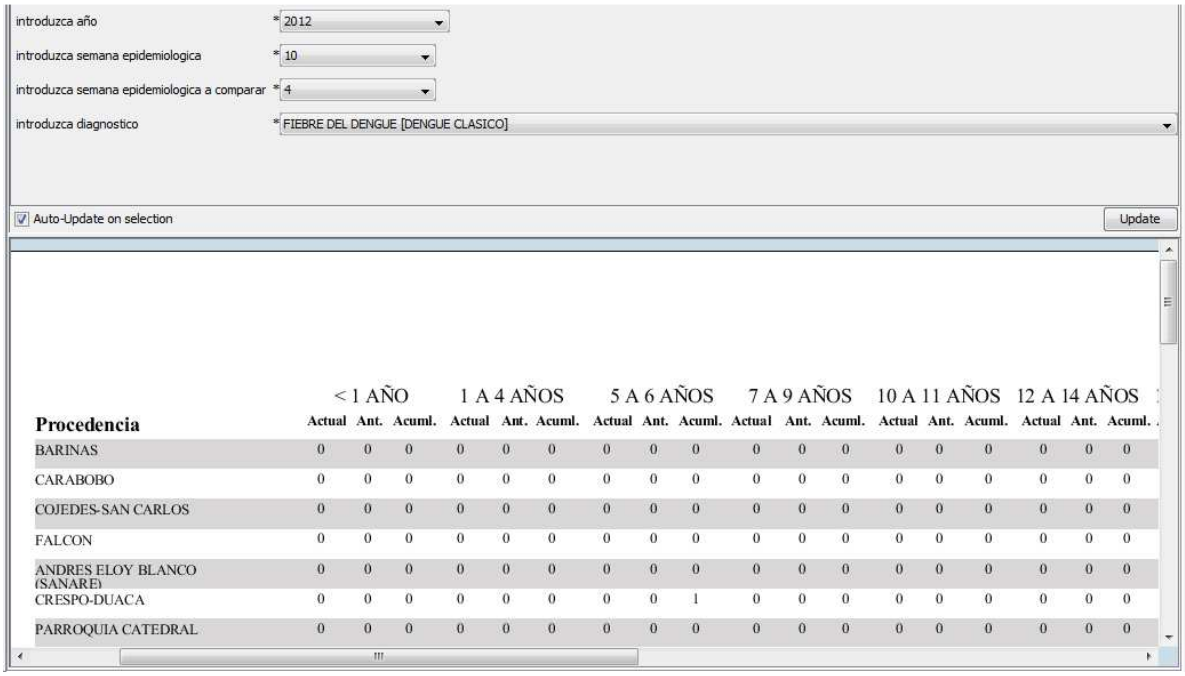


Figura 64: Enfermedades de Notificación obligatoria acumulativa.

Vigilancia especializada de las enfermedades de notificación obligatoria.

Permite conocer el número de casos de una ENO específica distribuidos por procedencia y grupos de edad en una semana epidemiológica específica, la semana anterior y los casos acumulados en el año(ver figura 65).

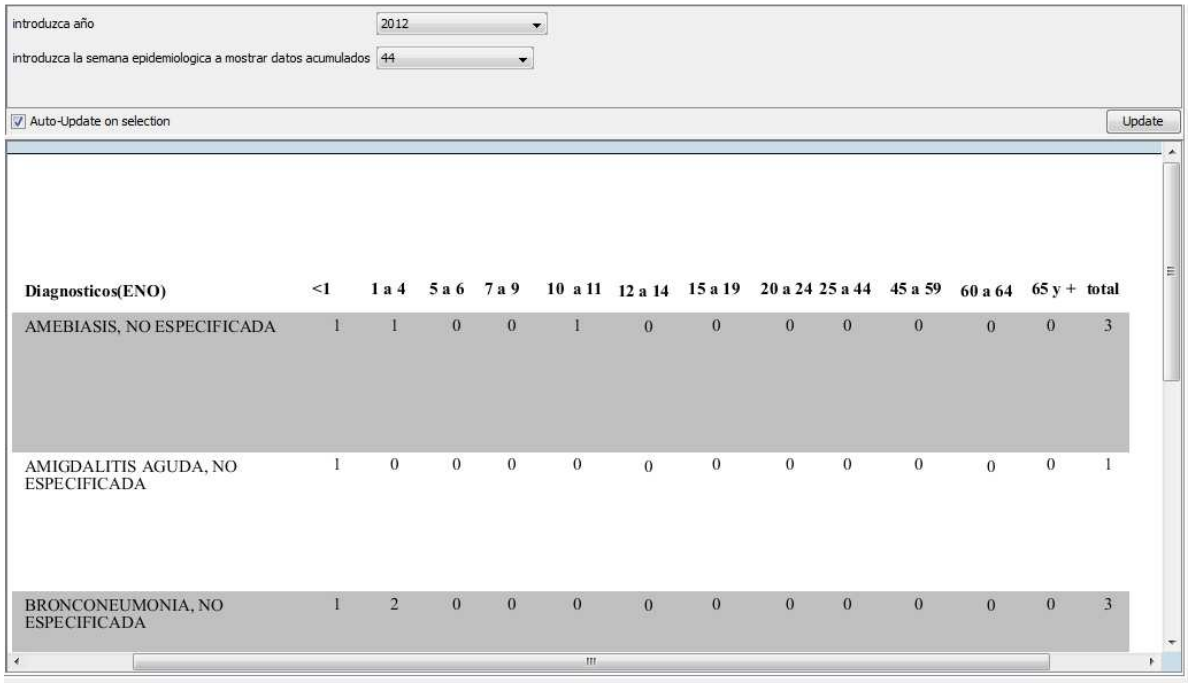


Procedencia	< 1 AÑO			1 A 4 AÑOS			5 A 6 AÑOS			7 A 9 AÑOS			10 A 11 AÑOS			12 A 14 AÑOS		
	Actual	Ant.	Acuml.	Actual	Ant.	Acuml.	Actual	Ant.	Acuml.	Actual	Ant.	Acuml.	Actual	Ant.	Acuml.	Actual	Ant.	Acuml.
BARINAS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CARABOBO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
COJEDES-SAN CARLOS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FALCON	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANDRES ELOY BLANCO (SANARE)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CRESPO-DUACA	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
PARROQUIA CATEDRAL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 65: Vigilancia especializada de las enfermedades de notificación obligatoria.

Consolidado semanal de enfermedades y eventos de notificación obligatoria.

Permite conocer el número de casos de las ENO por grupos etarios para una semana epidemiológica específica y un año especifico (ver figura 66).

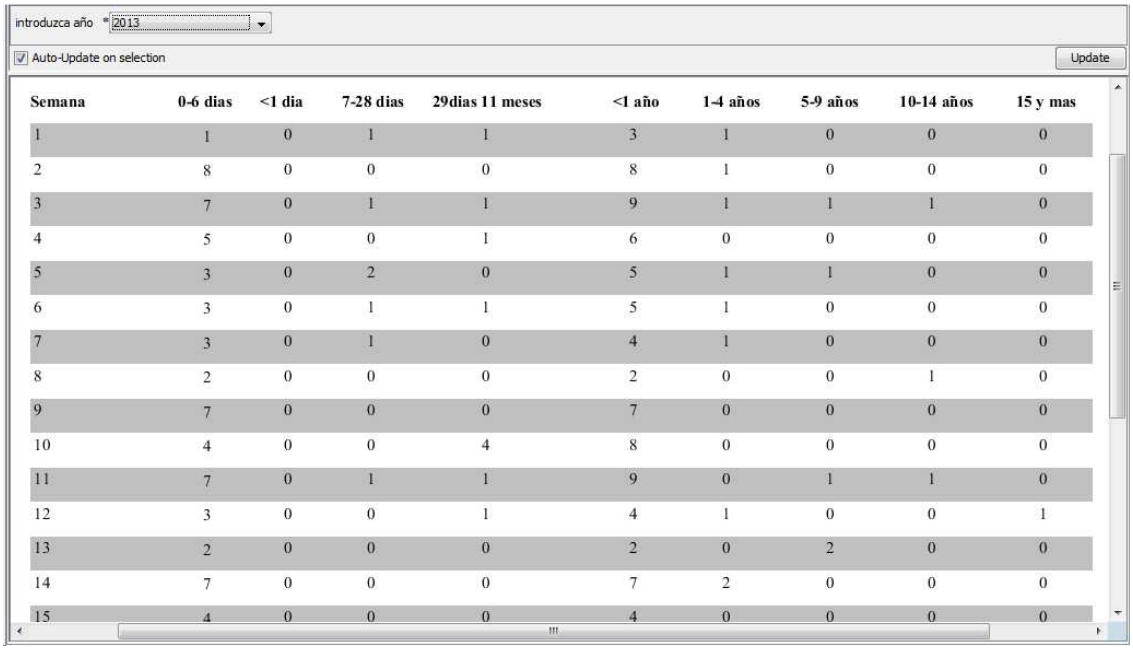


Diagnosticos(ENO)	<1	1 a 4	5 a 6	7 a 9	10 a 11	12 a 14	15 a 19	20 a 24	25 a 44	45 a 59	60 a 64	65 y +	total
AMEBIASIS, NO ESPECIFICADA	1	1	0	0	1	0	0	0	0	0	0	0	3
AMIGDALITIS AGUDA, NO ESPECIFICADA	1	0	0	0	0	0	0	0	0	0	0	0	1
BRONCONEUMONIA, NO ESPECIFICADA	1	2	0	0	0	0	0	0	0	0	0	0	3

Figura 66: Consolidado semanal de enfermedades y eventos de notificación obligatoria.

Muertes infantiles por semana.

Conocer el número de muertes infantiles por grupos de edad según la semana epidemiológica de un año especifico (ver figura 67).



Semana	0-6 dias	<1 dia	7-28 dias	29dias 11 meses	<1 año	1-4 años	5-9 años	10-14 años	15 y mas
1	1	0	1	1	3	1	0	0	0
2	8	0	0	0	8	1	0	0	0
3	7	0	1	1	9	1	1	1	0
4	5	0	0	1	6	0	0	0	0
5	3	0	2	0	5	1	1	0	0
6	3	0	1	1	5	1	0	0	0
7	3	0	1	0	4	1	0	0	0
8	2	0	0	0	2	0	0	1	0
9	7	0	0	0	7	0	0	0	0
10	4	0	0	4	8	0	0	0	0
11	7	0	1	1	9	0	1	1	0
12	3	0	0	1	4	1	0	0	1
13	2	0	0	0	2	0	2	0	0
14	7	0	0	0	7	2	0	0	0
15	4	0	0	0	4	0	0	0	0

Figura 67: Muertes infantiles por semana.

Distribución porcentual de la mortalidad infantil por grupos etarios.

Permite conocer el porcentaje de muertes infantiles (menores de un año) por grupos de edad en una semana epidemiológica específica (ver figura 68).

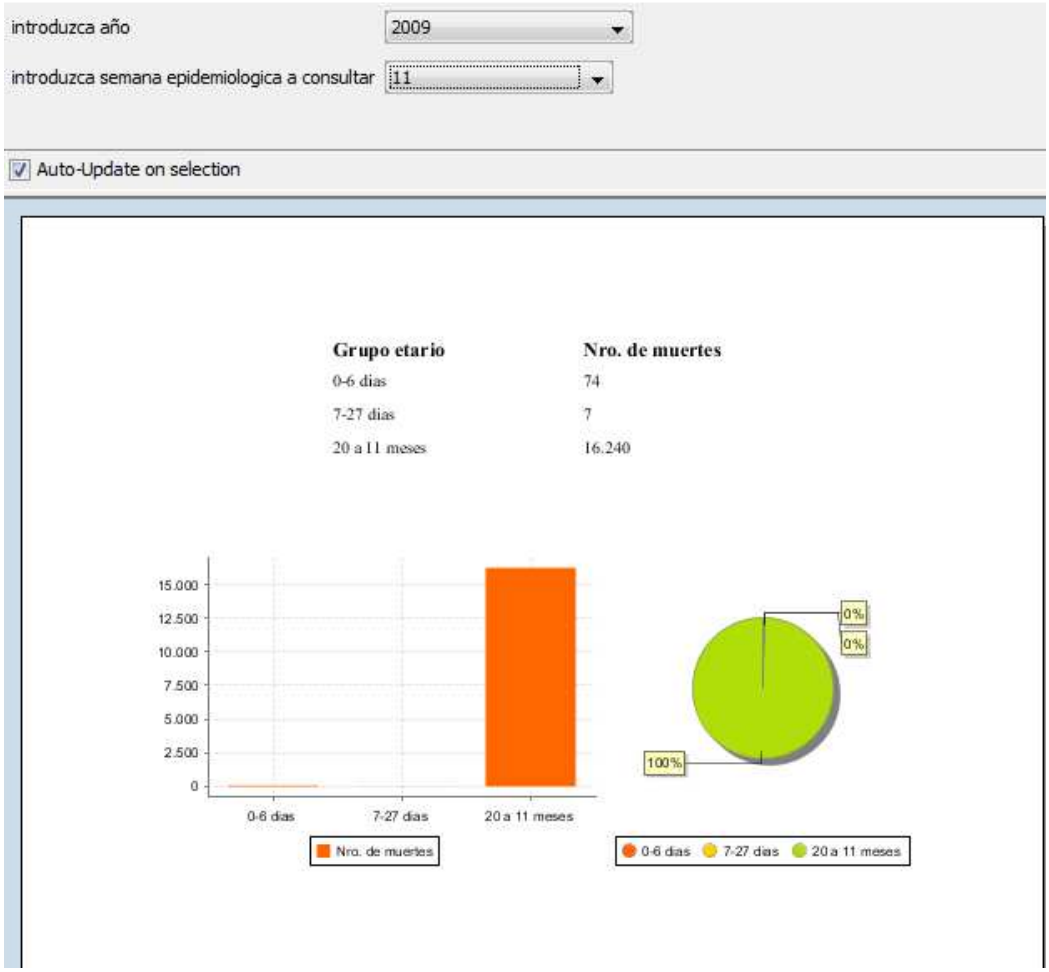
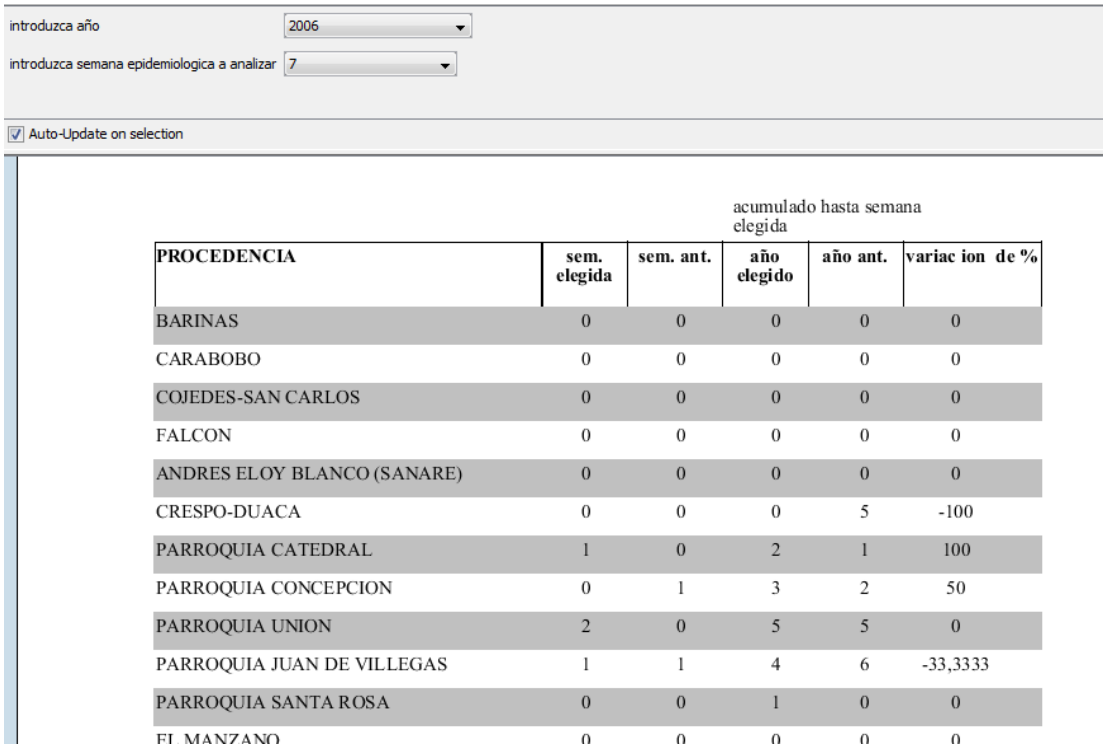


Figura 68: Distribución porcentual de la mortalidad infantil por grupos etarios.

Defunciones distribuidas por semanas epidemiológicas y procedencias.

Permite conocer el número de muertes infantiles (menores de un año) por semana epidemiológica y por procedencia. Además se desea conocer el acumulado del año actual y el anterior (ver figura 69).

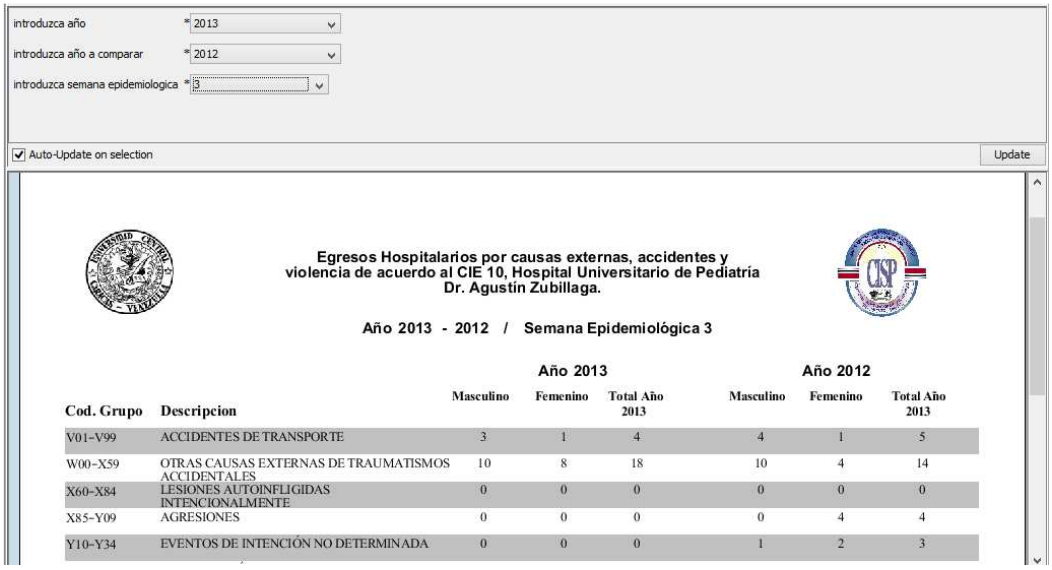


PROCEDENCIA	sem. elegida	sem. ant.	acumulado hasta semana elegida: año elegido	año ant.	variac ion de %
BARINAS	0	0	0	0	0
CARABOBO	0	0	0	0	0
COJEDES-SAN CARLOS	0	0	0	0	0
FALCON	0	0	0	0	0
ANDRES ELOY BLANCO (SANARE)	0	0	0	0	0
CRESPO-DUACA	0	0	0	5	-100
PARROQUIA CATEDRAL	1	0	2	1	100
PARROQUIA CONCEPCION	0	1	3	2	50
PARROQUIA UNION	2	0	5	5	0
PARROQUIA JUAN DE VILLEGAS	1	1	4	6	-33,3333
PARROQUIA SANTA ROSA	0	0	1	0	0
EL MANZANO	0	0	0	0	0

Figura 69: Defunciones distribuidas por semanas epidemiológicas y procedencias.

Causas de morbilidad

Permite listar las causas de morbilidad por causas externas, accidentes y violencia de acuerdo al CIE 10, los casos de la semana actual y los casos de la misma semana del año anterior y el número de casos por sexo (ver figura 70).

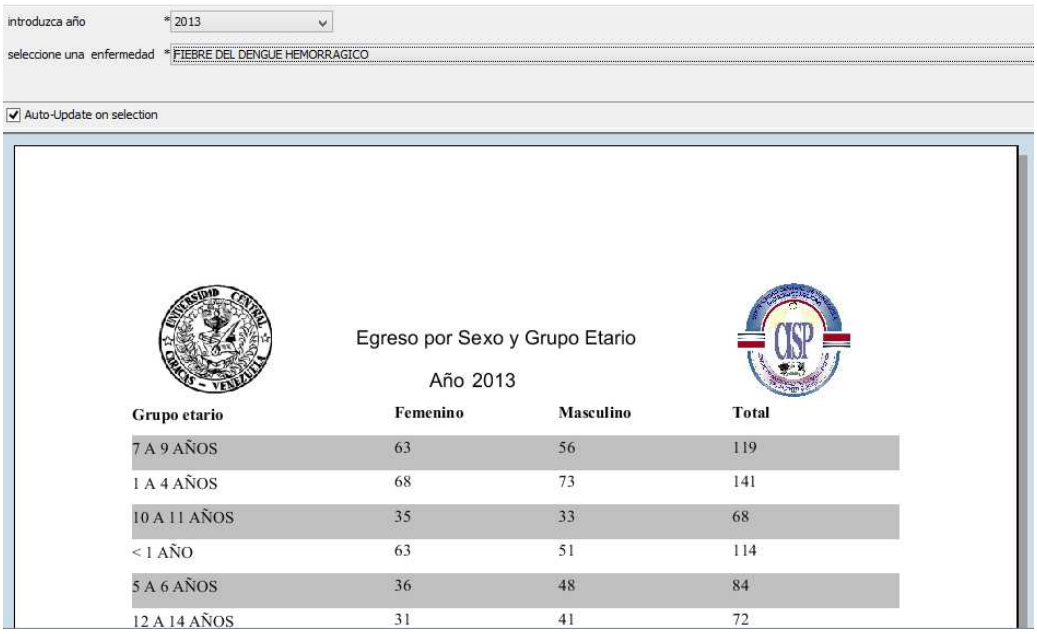


Cod. Grupo	Descripcion	Año 2013 Masculino	Año 2013 Femenino	Año 2013 Total Año 2013	Año 2012 Masculino	Año 2012 Femenino	Año 2012 Total Año 2013
V01-V99	ACCIDENTES DE TRANSPORTE	3	1	4	4	1	5
W00-X59	OTRAS CAUSAS EXTERNAS DE TRAUMATISMOS ACCIDENTALES	10	8	18	10	4	14
X60-X84	LESIONES AUTOINFLIGIDAS INTENCIONALMENTE	0	0	0	0	0	0
X85-Y09	AGRESIONES	0	0	0	0	4	4
Y10-Y34	EVENTOS DE INTENCIÓN NO DETERMINADA	0	0	0	1	2	3

Figura 70: Causas de morbilidad.

Casos por sexo y grupo etario para una enfermedad especifica

Permite conocer el número de casos de una enfermedad específica por grupos de edad y sexo que ocurren en un año especifico (ver figura 71).



Grupo etario	Femenino	Masculino	Total
7 A 9 AÑOS	63	56	119
1 A 4 AÑOS	68	73	141
10 A 11 AÑOS	35	33	68
< 1 AÑO	63	51	114
5 A 6 AÑOS	36	48	84
12 A 14 AÑOS	31	41	72

Figura 71: Casos por sexo y grupo etario para una enfermedad específica.

2.5.2 Aplicaciones analíticas

Las aplicaciones analíticas son más complejas que los informes estándar. Normalmente se centran en un proceso de negocio específico y resumen cierta experiencia acerca de cómo analizar e interpretar ese proceso de negocio.

Para llevar a cabo el análisis OLAP se escogió hacer uso del motor OLAP Mondrian y Jpivot.

A continuación se puede visualizar en la figura 72, el motor OLAP Mondrian, que permite la definición de los cubos .

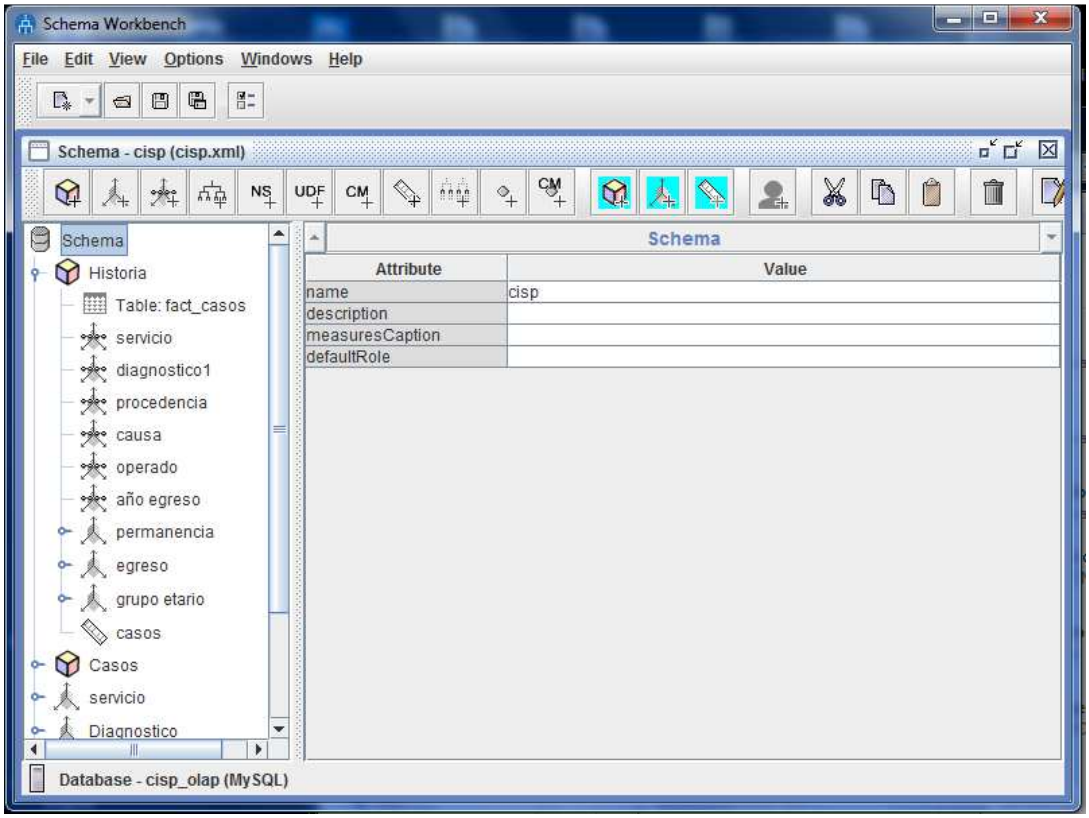


Figura 72: Herramienta Mondrian.

En la figura anterior se muestra la definición de dos cubos OLAP llamados Historia y Casos.

En la imagen que se muestra a continuación se puede observar el despliegue del cubo historia con Jpivot (ver figura 63).

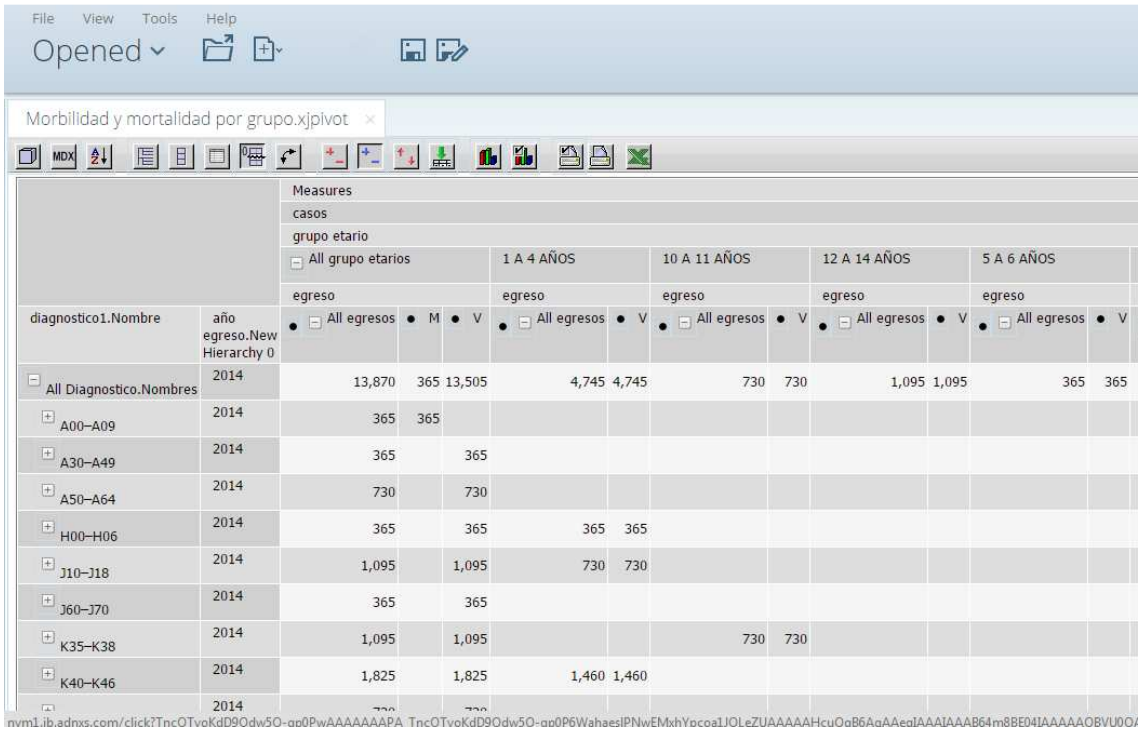


Figura 73: Jpivot cubo historia.

2.6Diseño de la Arquitectura Técnica

Los ambientes de data warehousing requieren la integración de numerosas tecnologías. Se debe tener en cuenta tres factores: los requerimientos del negocio, los actuales ambientes técnicos y las directrices técnicas estratégicas futuras planificadas para de esta forma

poder establecer el diseño de la arquitectura técnica del ambiente de data warehousing.

Teniendo en cuenta dichos factores se adoptó la arquitectura técnica que se puede apreciar en la figura 74.

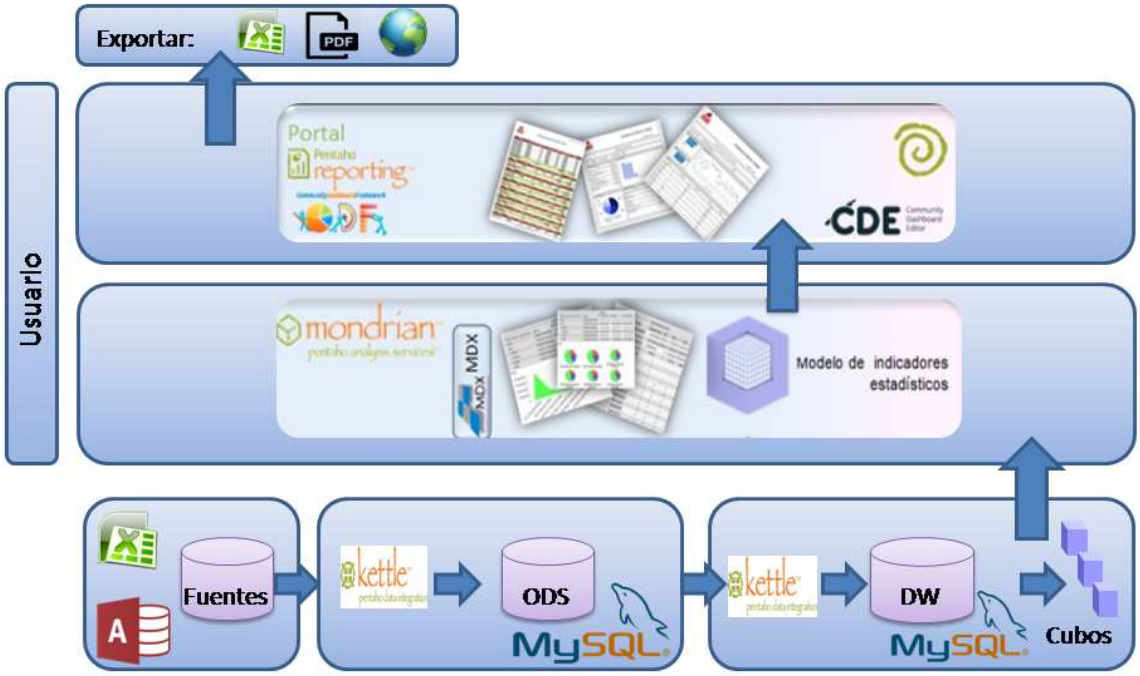


Figura 74: Arquitectura técnica.

Los procesos de extracción y transformación y carga (ETL) se realizan a través de la herramienta Pentaho Data Integration Kettle.

La Arquitectura va a estar compuesta por un ODS, donde se integrara la data proveniente de documentos Access, y servirá de apoyo para los sistemas transaccionales.

Se integra la data proveniente de las fuentes de datos a través de un primer proceso ETL hacia el ODS.

Una vez que termina el primer proceso ETL, se extrae la data integrada y estandarizada desde el ODS hacia el datamart a través de un segundo proceso de ETL.

EL ODS, el staging area y el datamart son implementados físicamente en Mysql.
Se hará uso del motor OLAP Mondrian para implementar los cubos multidimensionales.
Una vez definidos los cubos, estos son cargados al servidor de BI pentaho, el cual permite el despliegue de los cubos definidos en Mondrian y generación de gráficos.
A través del uso de la herramienta Jpivot se pueden realizar operaciones OLAP como drill down, Roll up, slice and dice y pivot, que permitirá un análisis de la data.
Se generan reportes con la herramienta report designer
Se podrá automatizar la generación de reportes en Excel, pdf, entre otros.

2.7 Selección de Productos e Instalación

Utilizando el diseño de arquitectura técnica como marco, es necesario evaluar y seleccionar componentes específicos de la arquitectura.

- Sistema Manejador de Bases de datos(SMBD)

Se seleccionó como manejador de base de datos para el ODS, el Staging área y el Data mart Mysql.

- Herramienta de ETL

Se eligió Pentaho data integration (Kettle), la cual contiene Spoon que permite de forma gráfica diseñar e implementar procesos ETL de forma gráfica.

- Herramientas de acceso

Para acceder a la información que se encuentra dentro del datamart, se eligió hacer uso de la plataforma Pentaho que incluye herramientas que permiten realizar análisis y reportes que son desplegados en un servidor.

Conclusiones

El presente trabajo de grado permite tener una visión más general de los diferentes inconvenientes que se presentan a la hora de tomar decisiones certeras en el ámbito de vigilancia epidemiológica, específicamente en el Centro de investigación de Salud Publica Dr. Jacinto Convit. A continuación se presentan algunas conclusiones obtenidas a la culminación del mismo:

- Se cumplieron los objetivos tanto generales como específicos que se plantearon.
- El subconjunto de datos provenientes de los sistemas transaccionales proporcionados por los epidemiólogos, para realizar el análisis, se encuentran incompletos y en ocasiones duplicados, lo que implica inconsistencias.
- Se centralizó la data proveniente de los distintos archivos Access correspondiente a los años 2002 hasta 2014 en el Datamart Cisp_olap.
- Adicionalmente, se diseñó y se implementó el proceso ETL encargado de realizar la carga de datos que se encuentren en los archivos Access que se generen en los años siguientes, hacia el datamart Cisp_olap.
- Se logró automatizar la generación de 15 reportes parametrizados que representan la piedra angular del proceso de vigilancia en el CISP.

Recomendaciones

A continuación se presentan una serie de recomendaciones, para que todo el proceso de implementación del Data warehouse se realice de manera exitosa y sin ningún tipo de problemas:

- Debido a las limitaciones del CISP, la captura de datos se realiza de manera manual por el personal. Por lo tanto se recomienda, que automaticen el proceso de captura de datos.
- Para que el proceso de ETL se lleve de manera exitosa, es necesario colocar todos los archivos de los sistemas fuente en un directorio predefinido. En nuestro caso, es necesario colocarlos dentro del directorio **"c:\cisp"** del sistema en donde se ejecutarán los paquetes DTS. Data warehouse, dentro del directorio **"c:\cisp\pediatria.dbm"**

Bibliografia

Albuera, P. (2009). Software Propietario. Recuperado el 12 de Junio de 2014, de http://www.slideshare.net/pabloalbuera/presentations

Berkelman RL, Buehler JW (1990). Public health surveillance of non-infectious chronic diseases: the potential to detect rapid changes in disease burden. International Journal of Epidemiology.

Buitrago, L. (2008). Concepto y Clasificación de Bases de Datos. Recuperado el 10 de Abril de 2014, de http://tbtsistemasdeinformacionybasesdedatos.blogspot.com/2008/09/concepto-y-clasificacion-de-bases-de.html

Cano, J. (2007). Business Intelligence: Competir con Información.

Connolly, T., &Begg, C. (2004). Database Systems: A Practical Approach to Design, Implementation and Management. Massachusetts: Addison-Wesley.

Costal, D. (s.f.). Introducción al Diseño de Bases de Datos. Recuperado el 10 de Abril de 2014, de http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02150.pdf

DaniSantia. (2012). Tema 2 - Modelo Relacional. Recuperado el 10 de Abril de 2014, de http://www.slideshare.net/DaniSantia/t2-modelo-relacional

Date, C. (2001). Introducción a los Sistemas de Bases de Datos. México: Pearson Education.

Duque, A. (2010). Implementación de un DataWarehouse para el Instituto Geográfico Militar.

Elmasri, R., &Navathe, S. (2007). Sistemas de Bases de Datos. Conceptos Fundamentales. México: Addison-Wesley Iberoamericana.

EPI-CENTRO (2003). Pontificia Universidad Católica de Chile. Tomado de http://escuela.med.puc.cl/Recursos/recepidem/insIntrod6.htm, http://escuela.med.puc.cl/Recursos/recepidem/insIntrod9c.htm.

Garmendia, L. (2009). Modelo Relacional. Recuperado el 10 de Abril de 2014, de http://www.fdi.ucm.es/profesor/lgarmend/FBD/Tema%202.2%20Modelo%20relacional%20v16.pdf

Hernando, R. (2004). El SGBDR Oracle. Recuperado el 19 de Abril de 2014, de http://www2.rhernando.net/modules/tutorials/doc/bd/oracle.html

I-test. (2005).Qué es un Benchmarck? Recuperado el 23 de Abril de 2014, de http://www.cretav.com/benchmark/bienvenida

Imhoff&Galemmo(2003), Mastering Data Warehouse Design: Relational and Dimensional Techniques, Wiley Publishing.

Inmon(2002), Building the Data Warehouse, (Third Edition). John Wiley&Sons.

Jorgenio. (2012). PostgreSQL vs MySQL: Cuál Elegir?. Recuperado el 20 de Abril de 2014, de http://blog.jorgenio.com/postgresql-vs-mysql-cual-elegir/

Kenneth S. Rubin (2012) Essential Scrum: A Practical Guide to the Most Popular Agile Process.

Kendall & Kendall (2011). Análisis y diseño de sistemas.

Laudon K. &Laudon J.(2012) Sistemas de información gerencial. Nueva York: Pearson.

Kimball &Caserta(2004), The Data Warehouse ETL Toolkit, Indianapolis, Wiley.

Kimball &Merz(2000), The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse, Wiley.

Kimball &Ross(2002), The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition), New York, Wiley.

Kimball &Ross(2010), The Kimball Group Reader; Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Indianapolis, Wiley.

Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., & Becker, B. (2008). The data warehouse lifecycle toolkit (2nd ed.). Wiley Publishing, Inc.

Gonzales R., Impacto de la datawarehouse e inteligencia de negocios en el desempeño de las empresas. Tesis Doctoral. Perú.

Gamboa E. &BerroteranD. (2007), Sistema de información de apoyo a la toma de decisiones relacionadas con la facturación, pedidos y necesidades de recursos humanos de la empresaFlaberge. Tesis de grado. Caracas.

López, A. (s.f.). El modelo Entidad-Relación. Recuperado el 10 de Abril de 2014, de http://hp.fciencias.unam.mx/~alg/bd/er.pdf

Newcomlab. (2013). Modelo de Tres Capas. Recuperado el 15 de Abril de 2014, de http://www.newcomlab.com/default.aspx?id_seccion=936

MySQL. (2013). AboutMySQL. Recuperado el 19 de Abril de 2014, de http://www.mysql.com/about/

Mozarrain, A.Gestión Indicadores
http://personales.jet.es/amozarrain/gestion_indicadores.htm

Oracle Technology Network. (2011). Características y Ventajas de Oracle. Recuperado el 19 de Abril de 2014, de http://docs.oracle.com/cd/E19593-01/E22994/gizfh.html

Ospina, M. (2009). Fundamentos, Conceptos Básicos y Diseño de Bases de Datos. Universidad Central de Venezuela. Caracas, Venezuela.

OMS (2001). Módulos de Principios de Epidemiología para el Control de Enfermedades, segunda edición.

Pecos, D. (2002). PostgreSQL vs MySQL. Recuperado el 20 de Abril de 2014, de http://www.netpecos.org/docs/mysql_postgres/x108.html

Pérez, D. (2007). ¿Qué son las Bases de Datos?. Recuperado el 02 de Abril de 2014, de http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/

PostgreSQL. (2010). Sobre PostgreSQL. Recuperado el 19 de Abril de 2014, de http://www.postgresql.org.es/sobre_postgresql

Rivadera, G. (2010) La metodología de Kimball para el diseño de almacenes de datos (Data Warehouses).

Silberschatz, A., Korth, H., &Sudarshan, S. (2002). Fundamentos de Bases de Datos. España: McGraw-Hill.

Sinnexus(2007). Bases de datos OLTP y OLAP. Recuperado el 12 de Abril de 2014, de http://www.sinnexus.com/business_intelligence/olap_vs_oltp.aspx

Sinnexus(2007). Bases de datos OLTP y OLAP. Recuperado el 12 de Abril de 2014, de http://www.sinnexus.com/business_intelligence/olap_avanzado.aspx

Universidad Fasta. (s.f.). Diseño de Bases de Datos. Recuperado el 10 de Abril de 2014, de http://www.ufasta.edu.ar/wp-content/uploads/04-Dise%C3%B1o-de-bases-de-datos.pdf

Wolff (2002). Modelamiento Multidimensional. Recuperado el 10 de Abril de 2014, de http://www.inf.udec.cl/~revista/ediciones/edicion4/modmulti.PDF

Datawarehouse (Almacenes de Datos) Prof. Concettina Di Vasta www.ciens.ucv.ve/portalasig/almacenamiento_de_datos/Herramientas OLAP_clase6.pdf

Karen Levy,Miguel Henríquez (2005)DESARROLLO DE UN DATA WAREHOUSE BASADO EN INDICADORES DE GESTIÓNEN EL ÁREA EDUCATIVA,PARA LA ESCUELA DE QUÍMICA, FACULTAD DE CIENCIAS DE LA UCV

http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/MySQL.pdf

http://www.mysql.com/

http://gravitar.biz/pentaho/

http://community.pentaho.com

Anexos

1.-Procedimientos del ODS Cisp.

1.1-Procedimiento almacenado que permite asignar los gripos etarios de la tabla historia en el ODS CISP.

```
DROP PROCEDURE  ifEXISTS ASIGNAR_GRUPOET;
CREATE  PROCEDURE ASIGNAR_GRUPOET()
BEGIN
DECLARE v_GRUPO varchar(255);
DECLARE v_ANIO INTEGER ;
DECLARE fin INTEGER DEFAULT 0;
DECLARE edades_cursor CURSOR FOR select  edad_anio,nombre
from historias,GRUPOS_ETARIOS WHERE  grupo_etario is NULL
AND edad_anio BETWEEN MINIMO AND MAXIMO
GROUP BY edad_anio,nombre;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN edades_cursor;
get_edades: LOOP
FETCH edades_cursor INTO v_ANIO, v_GRUPO;
IF fin = 1 THEN
        LEAVE get_edades;
END IF;
UPDATE historias SET      grupo_etario=v_GRUPO
where edad_anio=v_ANIO;

END LOOP get_edades;
CLOSE edades_cursor;
END;
```

1.2.-Procedimiento almacenado que completa con 0 si el código de causala tabla historia es de 3 dígitos y no se encuentra en la tabla diagnóstico. Si el nuevo código completado con 0 existe en la tabla diagnostico se actualiza la tabla historia.

```
DROP PROCEDURE  if EXISTS completar_causas;
CREATE  PROCEDURE completar_causas()
BEGIN
DECLARE v_cod3 VARCHAR(255);
DECLARE v_cod4 VARCHAR(255);
DECLARE v_result VARCHAR(255);
DECLARE v_anioepi INTEGER;

DECLARE fin INTEGER DEFAULT 0;

DECLARE diagnosticos_cursor CURSOR FOR select  codigo_causa
FROM causas_faltan1;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN diagnosticos_cursor;
get_diagnosticos: LOOP
FETCH diagnosticos_cursor INTO v_cod3;
IF fin = 1 THEN
	LEAVE get_diagnosticos;
END IF;

SELECT CONCAT(v_cod3, "0") into v_cod4;
SELECT codigo_causa FROM causas WHERE Codigo_causa=v_cod4
into v_result;
IF v_result is NOT NULL THEN
	update historias set cod_causa1=v_cod4 where
cod_causa1=v_cod3;
	update historias set cod_causa2=v_cod4 where
cod_causa2=v_cod3;
	update historias set cod_causa3=v_cod4 where
cod_causa3=v_cod3;
	update historias set cod_causa4=v_cod4 where
cod_causa4=v_cod3;
	update historias set cod_causa5=v_cod4 where
cod_causa5=v_cod3;
	DELETE FROM causas_faltan1 WHERE codigo_causa =
v_cod3;
ELSE
	SET v_cod4=NULL;
END IF;
```

```
END LOOP get_diagnosticos;
CLOSE diagnosticos_cursor;
END;
```

1.3.-Procedimiento almacenado que completa con 0 si el código de diagnóstico de la tabla historia es de 3 dígitos y no se encuentra en la tabla diagnóstico. Si el nuevo código completado con 0 existe en la tabla diagnostico se actualiza la tabla historia.

```
DROP PROCEDURE  if EXISTS completar_diagnosticos;
CREATE  PROCEDURE completar_diagnosticos()
BEGIN
DECLARE v_cod3 VARCHAR(255);
DECLARE v_cod4 VARCHAR(255);
DECLARE v_result VARCHAR(255);
DECLARE v_anioepi INTEGER;

DECLARE fin INTEGER DEFAULT 0;

DECLARE diagnosticos_cursor CURSOR FOR select  codigo_dia
FROM codigos_faltan ORDER BY codigo_dia DESC;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN diagnosticos_cursor;
get_diagnosticos: LOOP
FETCH diagnosticos_cursor INTO v_cod3;
IF fin = 1 THEN
        LEAVE get_diagnosticos;
END IF;

SELECT CONCAT(v_cod3, "0") into v_cod4;
SELECT codigo_dia FROM diagnostico WHERE codigo_dia=v_cod4
into v_result;
IF v_result is NOT NULL THEN
        update historias set cod_diagnostico1=v_cod4 where
cod_diagnostico1=v_cod3;
        update historias set cod_diagnostico2=v_cod4 where
cod_diagnostico2=v_cod3;
        update historias set cod_diagnostico3=v_cod4 where
cod_diagnostico3=v_cod3;
```

```
        update historias set cod_diagnostico4=v_cod4 where
cod_diagnostico4=v_cod3;
        update historias set cod_diagnostico5=v_cod4 where
cod_diagnostico5=v_cod3;
        DELETE FROM codigos_faltan WHERE codigo_dia = v_cod3;
ELSE
        SET v_cod4=NULL;
END IF;
END LOOP get_diagnosticos;
CLOSE diagnosticos_cursor;
END;
```

2.-Procedimientos del Data Mart Cisp_olap

2.1.Procedimiento que calcula el calendario epidemiológico.

```
CREATE  PROCEDURE calcular_semep ()
BEGIN
DECLARE v_id INTEGER;
DECLARE v_dia INTEGER;
DECLARE v_diasem INTEGER;
DECLARE v_diadelmes INTEGER;
DECLARE v_mes INTEGER;
DECLARE verifica INTEGER DEFAULT 0;
DECLARE miercoles INTEGER DEFAULT 0;
DECLARE bisiesto INTEGER DEFAULT 0;
DECLARE v_contsem INTEGER DEFAULT 1;
DECLARE fin INTEGER DEFAULT 0;
DECLARE v_year INTEGER DEFAULT 2000;
DECLARE fechas_cursor CURSOR FOR SELECT id, diadelanio,
diadesemana,diadelmes,mes FROM dim_fecha;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN fechas_cursor;
get_fechas: LOOP
FETCH fechas_cursor INTO v_id, v_dia, v_diasem, v_diadelmes,
v_mes;
IF fin = 1 THEN
        LEAVE get_fechas;
END IF;
```

```
IF v_contsem = 1 AND v_diasem=7 and v_dia=1 and v_year=2000
THEN
	UPDATE dim_fecha set semanaepi= 52 where id=v_id;
	UPDATE dim_fecha set anioepi= 1999 where id=v_id;
ELSE
	IF v_mes=1 and v_diadelmes=1 AND  v_diasem = 4  THEN
		SET miercoles=1;
	END IF;
	IF miercoles=1 and v_mes=2 and v_diadelmes=29   THEN
		SET bisiesto=1;
	END IF;
	IF bisiesto=1 AND v_mes=12 and v_diadelmes=19 AND
v_diasem = 7  THEN
		SET verifica=1;
	END IF;
	IF v_mes=12 and v_diadelmes=20 AND  v_diasem = 7
THEN
		SET verifica=1;
	END IF;
	UPDATE dim_fecha set semanaepi= v_contsem where
id=v_id;
	UPDATE dim_fecha set anioepi= v_year where id=v_id;
	IF v_diasem = 7 THEN
		SET v_contsem=v_contsem +1;
		IF verifica=1  THEN
			IF v_contsem = 54 THEN
			 SET v_contsem=1;
			 SET v_year=v_year +1;
			 SET verifica=0;
			 SET miercoles=0;
			 SET bisiesto=0;
			END IF;
		ELSE
			IF v_contsem = 53 THEN
			 SET v_contsem=1;
			 SET v_year=v_year +1;
			 SET miercoles=0;
			 SET bisiesto=0;
			END IF;
		END IF;
```

```
        END IF;
    END if;
    END LOOP get_fechas;
    CLOSE fechas_cursor;
    END;
```

2.2 Procedimientos para insertar los casos 0 en la tabla de hechos casos_completos para poder formar el canal endémico.

```
DROP  procedure if EXISTS insertar_casos;
CREATE  PROCEDURE insertar_casos()
BEGIN
DECLARE v_diagnostico INTEGER;

DECLARE v_anioepi INTEGER;
DECLARE v_idsemanaepi INTEGER;
DECLARE fin INTEGER DEFAULT 0;
DECLARE termino INTEGER DEFAULT 0;
DECLARE casos_cursor CURSOR FOR select id_diagnostico,
id_procedencia,anioepi
from (select id_diagnostico,casos,semana_epidemiologica,anioepi,
semanaepi from fact_casos_corredor,dim_epidemiologica WHERE
semana_epidemiologica=id) a
WHERE id_diagnostico <> 0 and anioepi is NOT NULL
GROUP BY id_diagnostico,anioepi;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;

OPEN casos_cursor;
get_casos: LOOP
FETCH casos_cursor INTO v_diagnostico,v_anioepi;
IF fin = 1 THEN
       LEAVE get_casos;
END IF;

CALL insertar_semanas(v_diagnostico,v_anioepi);
END LOOP get_casos;
CLOSE casos_cursor;
END
```

Procedimiento que es llamado el procedimiento insertar_casos descrito anteriormente. Este realiza la inserción de las semanas faltantes.

```
DROP PROCEDURE if EXISTS insertar_semanas ;
CREATE PROCEDURE insertar_semanas
(IN  v_diagnostico  INTEGER,IN  v_anioepi  INTEGER)
BEGIN

DECLARE fin INTEGER DEFAULT 0;
DECLARE v_idsemanaepi INTEGER;
DECLARE semanas_cursor CURSOR FOR SELECT id from
dim_epidemiologica WHERE anioepi=v_anioepi and id not in
(select semana_epidemiologica from
fact_casos_corredor,dim_epidemiologica WHERE
semana_epidemiologica=id and id_diagnostico = v_diagnostico AND
id_procedencia=v_procedencia and anioepi=v_anioepi);
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;

OPEN semanas_cursor;
get_semanas: LOOP
FETCH semanas_cursor INTO v_idsemanaepi;
IF fin = 1 THEN
        LEAVE get_semanas;
END IF;
if v_idsemanaepi is not NULL THEN
INSERT into fact_casos_corredor
VALUES(v_diagnostico,v_idsemanaepi,0,0,0,0);
END IF;
END LOOP get_semanas;
CLOSE semanas_cursor;



END;
```

2.4 Procedimientos que permiten calcular los percentiles 25, 50 y 75 que son almacenados en los registros de la tabla de hechos casos_completos.

```
DROP  procedure if EXISTS calcular_percentiles;
CREATE  PROCEDURE calcular_percentiles()
BEGIN
DECLARE v_id_diagnostico INTEGER DEFAULT 0;

DECLARE v_semana_epidemiologica INTEGER DEFAULT 0;
DECLARE v_anioepi INTEGER DEFAULT 0;
DECLARE v_p25 DOUBLE;
DECLARE v_p50 DOUBLE;
DECLARE v_p70 DOUBLE;
DECLARE fin INTEGER DEFAULT 0;
DECLARE v_semanaepi INTEGER DEFAULT 0;
DECLARE casos_cursor CURSOR FOR   select
fact_casos_corredor.id_diagnostico,fact_casos_corredor.semana_epidemi
ologica,dim_epidemiologica.semanaepi,dim_epidemiologica.anioepi
from fact_casos_corredor INNER JOIN dim_epidemiologica on
dim_epidemiologica.id=fact_casos_corredor.semana_epidemiologica
where anioepi>=2009
and casos.id_diagnostico<>0
ORDER BY anioepi;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN casos_cursor;
get_diagnosticos: LOOP
FETCH casos_cursor INTO
v_id_diagnostico,v_semana_epidemiologica,v_semanaepi, v_anioepi ;
IF fin = 1 THEN
        LEAVE get_diagnosticos;
END IF;

CALL
buscar_años_ant(@v_p25,@v_p50,@v_p75,v_id_diagnostico,v_semanae
pi,v_anioepi);
Update fact_casos_corredor SET percentil25=@v_p25,
percentil50=@v_p50, percentil75=@v_p75
where id_diagnostico=v_id_diagnostico
AND semana_epidemiologica= v_semana_epidemiologica;
```

```
END LOOP get_diagnosticos;
CLOSE casos_cursor;
END
```

Procedimiento llamado por el procedimiento calcular_percentiles, buscar_años_ant

```
DROP procedure IF EXISTS buscar_años_ant;
CREATE  PROCEDURE buscar_años_ant( INOUT v_p25 DOUBLE

                                        ,INOUT v_p50 DOUBLE

                                        ,INOUT v_p75 DOUBLE

                                    ,IN v_id_diagnostico INTEGER

                                    ,IN v_semanaepi INTEGER

                                     ,IN v_anioepiINTEGER )
BEGIN
DECLARE v_caso1 INTEGER DEFAULT 0;
DECLARE v_caso2 INTEGER DEFAULT 0;
DECLARE v_caso3 INTEGER DEFAULT 0;
DECLARE v_caso4 INTEGER DEFAULT 0;
DECLARE v_caso5 INTEGER DEFAULT 0;
DECLARE v_caso6 INTEGER DEFAULT 0;
DECLARE v_caso7 INTEGER DEFAULT 0;
DECLARE v_caso8 INTEGER DEFAULT 0;
DECLARE fin INTEGER DEFAULT 0;
DECLARE cont INTEGER DEFAULT 7;
DECLARE casos_cursor CURSOR FOR select
cisp_olap.fact_casos_corredor.casos
from cisp_olap.fact_casos_corredor INNER JOIN
cisp_olap.dim_epidemiologica on
cisp_olap.dim_epidemiologica.id=cisp_olap.fact_casos_corredor.semana_
epidemiologica
where id_diagnostico=v_id_diagnostico   AND semanaepi=v_semanaepi
AND  anioepi BETWEEN v_anioepi-7 and v_anioepi-1 and
cisp_olap.fact_casos_corredor.id_diagnostico<>0 ORDER BY
cisp_olap.fact_casos_corredor.casos DESC;
```

```
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN casos_cursor;
get_diagnosticos: LOOP

CASE cont
     WHEN 7 THEN FETCH casos_cursor INTO v_caso7;
     WHEN 6 THEN FETCH casos_cursor INTO v_caso6;
                    WHEN 5 THEN FETCH casos_cursor INTO v_caso5;
                    WHEN 4 THEN FETCH casos_cursor INTO v_caso4;
                    WHEN 3 THEN FETCH casos_cursor INTO v_caso3;
                    WHEN 2 THEN FETCH casos_cursor INTO v_caso2;
                    WHEN 1 THEN FETCH casos_cursor INTO v_caso1;
                    WHEN 0 THEN FETCH casos_cursor INTO v_caso8;
END CASE;
set cont=cont-1;
IF fin = 1 THEN
        LEAVE get_diagnosticos;
END IF;
END LOOP get_diagnosticos;
CLOSE casos_cursor;
set v_p25=v_caso2+(0.5*(v_caso3-v_caso2));
set v_p50=v_caso4;
set v_p75=v_caso5+(0.5*(v_caso6-v_caso5));

END
```

3.-Procedimientos del staging área.

3.1. Procedimiento que asigna los grupos etarios a los registros futuros.

```
CREATE PROCEDURE ASIGNAR_GRUPOET ()
BEGIN
DECLARE v_GRUPO varchar(255);
DECLARE v_ANIO INTEGER ;
DECLARE fin INTEGER DEFAULT 0;
```

```
DECLARE edades_cursor CURSOR FOR select  edad_anio,nombre from
historias,GRUPOS_ETARIOS WHERE  grupo_etario is NULL AND edad_anio
BETWEEN MINIMO AND MAXIMO
GROUP BY edad_anio,nombre;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN edades_cursor;
get_edades: LOOP
FETCH edades_cursor INTO v_ANIO, v_GRUPO;
IF fin = 1 THEN
        LEAVE get_edades;
END IF;
UPDATE historias SET      grupo_etario=v_GRUPO where
edad_anio=v_ANIO;

END LOOP get_edades;
CLOSE edades_cursor;
END
```

3.2 Procedimiento que completa los diagnosticos de 3 digitos de la tabla historia con 0 para que concuerden con el CIE 10.

```
CREATE  PROCEDURE completar_diagnosticos()
BEGIN
DECLARE v_cod3 VARCHAR(255);
DECLARE v_cod4 VARCHAR(255);
DECLARE v_result VARCHAR(255);
DECLARE v_anioepi INTEGER;

DECLARE fin INTEGER DEFAULT 0;

DECLARE diagnosticos_cursor CURSOR FOR select  codigo_dia FROM
diagnosticos_faltan ORDER BY codigo_dia DESC;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN diagnosticos_cursor;
get_diagnosticos: LOOP
FETCH diagnosticos_cursor INTO v_cod3;
IF fin = 1 THEN
        LEAVE get_diagnosticos;
END IF;
```

```
SELECT CONCAT(v_cod3, "0") into v_cod4;
SELECT codigo_dia FROM cisp.diagnostico WHERE codigo_dia=v_cod4
into v_result;
IF v_result is NOT NULL THEN
        update historias set cod_diagnostico1=v_cod4 where
cod_diagnostico1=v_cod3;
        update historias set cod_diagnostico2=v_cod4 where
cod_diagnostico2=v_cod3;
        update historias set cod_diagnostico3=v_cod4 where
cod_diagnostico3=v_cod3;
        update historias set cod_diagnostico4=v_cod4 where
cod_diagnostico4=v_cod3;
        update historias set cod_diagnostico5=v_cod4 where
cod_diagnostico5=v_cod3;
        DELETE FROM diagnosticos_faltan WHERE codigo_dia = v_cod3;
ELSE
        SET v_cod4=NULL;
END IF;
END LOOP get_diagnosticos;
CLOSE diagnosticos_cursor;
END
```

3.3 Procedimiento que inserta los casos 0 para formar el canal endémico.

```
DROP  procedure if EXISTS insertar_casos;
CREATE  PROCEDURE insertar_casos()
BEGIN
DECLARE v_diagnostico INTEGER;

DECLARE v_anioepi INTEGER;
DECLARE v_idsemanaepi INTEGER;
DECLARE fin INTEGER DEFAULT 0;
DECLARE termino INTEGER DEFAULT 0;
DECLARE casos_cursor CURSOR FOR select id_diagnostico,
id_procedencia,anioepi
from (select id_diagnostico,casos,semana_epidemiologica,anioepi,
semanaepi from fact_casos_corredor,dim_epidemiologica WHERE
semana_epidemiologica=id) a
WHERE id_diagnostico <> 0 and anioepi is NOT NULL
```

```
GROUP BY id_diagnostico,anioepi;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;

OPEN casos_cursor;
get_casos: LOOP
FETCH casos_cursor INTO v_diagnostico,v_anioepi;
IF fin = 1 THEN
        LEAVE get_casos;
END IF;

CALL insertar_semanas(v_diagnostico,v_anioepi);
END LOOP get_casos;
CLOSE casos_cursor;
END
```

Procedimiento llamado por insertar_semanas()

```
DROP PROCEDURE if EXISTS insertar_semanas ;
CREATE PROCEDURE insertar_semanas
(IN  v_diagnostico  INTEGER,IN  v_anioepi  INTEGER)
BEGIN

DECLARE fin INTEGER DEFAULT 0;
DECLARE v_idsemanaepi INTEGER;
DECLARE semanas_cursor CURSOR FOR SELECT id from
dim_epidemiologica WHERE anioepi=v_anioepi and id not in
(select semana_epidemiologica from
fact_casos_corredor,dim_epidemiologica WHERE
semana_epidemiologica=id and id_diagnostico = v_diagnostico AND
id_procedencia=v_procedencia and anioepi=v_anioepi);
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;

OPEN semanas_cursor;
get_semanas: LOOP
FETCH semanas_cursor INTO v_idsemanaepi;
IF fin = 1 THEN
        LEAVE get_semanas;
END IF;
if v_idsemanaepi is not NULL THEN
INSERT into fact_casos_corredor
VALUES(v_diagnostico,v_idsemanaepi,0,0,0,0);
```

```
END IF;
END LOOP get_semanas;
CLOSE semanas_cursor;



END;
```

3.4 Procedimiento que calcula los percentiles 25,50 y 75 para formar el canal endémico.

```
DROP  procedure if EXISTS calcular_percentiles;
CREATE  PROCEDURE calcular_percentiles()
BEGIN
DECLARE v_id_diagnostico INTEGER DEFAULT 0;

DECLARE v_semana_epidemiologica INTEGER DEFAULT 0;
DECLARE v_anioepi INTEGER DEFAULT 0;
DECLARE v_p25 DOUBLE;
DECLARE v_p50 DOUBLE;
DECLARE v_p70 DOUBLE;
DECLARE fin INTEGER DEFAULT 0;
DECLARE v_semanaepi INTEGER DEFAULT 0;
DECLARE casos_cursor CURSOR FOR   select
fact_casos_corredor.id_diagnostico,fact_casos_corredor.semana_epidemi
ologica,dim_epidemiologica.semanaepi,dim_epidemiologica.anioepi
from fact_casos_corredor INNER JOIN dim_epidemiologica on
dim_epidemiologica.id=fact_casos_corredor.semana_epidemiologica
where anioepi>=2009
and casos.id_diagnostico<>0
ORDER BY anioepi;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN casos_cursor;
get_diagnosticos: LOOP
FETCH casos_cursor INTO
v_id_diagnostico,v_semana_epidemiologica,v_semanaepi, v_anioepi ;
IF fin = 1 THEN
        LEAVE get_diagnosticos;
END IF;
```

```
CALL
buscar_años_ant(@v_p25,@v_p50,@v_p75,v_id_diagnostico,v_semanae
pi,v_anioepi);
Update fact_casos_corredor SET percentil25=@v_p25,
percentil50=@v_p50, percentil75=@v_p75
where id_diagnostico=v_id_diagnostico
AND semana_epidemiologica= v_semana_epidemiologica;
END LOOP get_diagnosticos;
CLOSE casos_cursor;
END
```

Procedimiento buscar_años_ant

```
DROP procedure IF EXISTS buscar_años_ant;
CREATE  PROCEDURE buscar_años_ant( INOUT v_p25 DOUBLE

                                        ,INOUT v_p50 DOUBLE

                                        ,INOUT v_p75 DOUBLE

                                    ,IN v_id_diagnostico INTEGER

                                    ,IN v_semanaepi INTEGER

                                      ,IN v_anioepiINTEGER )
BEGIN
DECLARE v_caso1 INTEGER DEFAULT 0;
DECLARE v_caso2 INTEGER DEFAULT 0;
DECLARE v_caso3 INTEGER DEFAULT 0;
DECLARE v_caso4 INTEGER DEFAULT 0;
DECLARE v_caso5 INTEGER DEFAULT 0;
DECLARE v_caso6 INTEGER DEFAULT 0;
DECLARE v_caso7 INTEGER DEFAULT 0;
DECLARE v_caso8 INTEGER DEFAULT 0;
DECLARE fin INTEGER DEFAULT 0;
DECLARE cont INTEGER DEFAULT 7;
DECLARE casos_cursor CURSOR FOR select
cisp_olap.fact_casos_corredor.casos
from cisp_olap.fact_casos_corredor INNER JOIN
cisp_olap.dim_epidemiologica on
```

```
cisp_olap.dim_epidemiologica.id=cisp_olap.fact_casos_corredor.semana_
epidemiologica
where id_diagnostico=v_id_diagnostico   AND semanaepi=v_semanaepi
AND  anioepi BETWEEN v_anioepi-7 and v_anioepi-1 and
cisp_olap.fact_casos_corredor.id_diagnostico<>0 ORDER BY
cisp_olap.fact_casos_corredor.casos DESC;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN casos_cursor;
get_diagnosticos: LOOP

CASE cont
     WHEN 7 THEN FETCH casos_cursor INTO v_caso7;
     WHEN 6 THEN FETCH casos_cursor INTO v_caso6;
                    WHEN 5 THEN FETCH casos_cursor INTO v_caso5;
                    WHEN 4 THEN FETCH casos_cursor INTO v_caso4;
                    WHEN 3 THEN FETCH casos_cursor INTO v_caso3;
                    WHEN 2 THEN FETCH casos_cursor INTO v_caso2;
                    WHEN 1 THEN FETCH casos_cursor INTO v_caso1;
                    WHEN 0 THEN FETCH casos_cursor INTO v_caso8;
END CASE;
set cont=cont-1;
IF fin = 1 THEN
       LEAVE get_diagnosticos;
END IF;
END LOOP get_diagnosticos;
CLOSE casos_cursor;
set v_p25=v_caso2+(0.5*(v_caso3-v_caso2));
set v_p50=v_caso4;
set v_p75=v_caso5+(0.5*(v_caso6-v_caso5));

END
```

3.5 Insertar casos en la tabla fact_casos_corredor

```
DROP PROCEDURE if EXISTS insertar_casos_completos2 ;
CREATE PROCEDURE insertar_casos_completos2()
BEGIN

DECLARE fin INTEGER DEFAULT 0;
DECLARE v_id_diagnostico INTEGER;
```

```
DECLARE v_semana_epidemiologica INTEGER;
DECLARE v_casos INTEGER;
DECLARE v_percentil25 FLOAT;
DECLARE v_percentil50 FLOAT;
DECLARE v_percentil75 FLOAT;
DECLARE cont INTEGER DEFAULT 0;
DECLARE semanas_cursor CURSOR FOR SELECT
id_diagnostico,semana_epidemiologica,casos,percentil25,percentil50,perc
entil75
FROM fact_casos_corredor;
DECLARE CONTINUE HANDLER for NOT FOUND SET  fin = 1;
OPEN semanas_cursor;
get_semanas: LOOP
FETCH semanas_cursor INTO
v_id_diagnostico,v_semana_epidemiologica,v_casos,v_percentil25,v_perc
entil50,v_percentil75;
IF fin = 1 THEN
        LEAVE get_semanas;
END IF;
SELECT COUNT(*) into cont from cisp_olap.fact_casos_corredor
WHERE cisp_olap.fact_casos_corredor.id_diagnostico=v_id_diagnostico
AND
cisp_olap.fact_casos_corredor.semana_epidemiologica=v_semana_epide
miologica;
if cont=1 THEN
UPDATE cisp_olap.fact_casos_corredor set
cisp_olap.fact_casos_corredor.casos=cisp_olap.fact_casos_corredor.casos
+ v_casos
WHERE cisp_olap.fact_casos_corredor.id_diagnostico=v_id_diagnostico
AND
cisp_olap.fact_casos_corredor.semana_epidemiologica=v_semana_epide
miologica;
SET cont=0;
ELSEIF cont=0 THEN
INSERT into fact_casos_corredor VALUES(v_id_diagnostico,
v_semana_epidemiologica,v_casos,v_percentil25,v_percentil50,v_percent
il75);
END IF;
END LOOP get_semanas;
CLOSE semanas_cursor;END;
```