Universidad de las Ciencias Informáticas
Facultad 3

Título: Diseño e Implementación de un Mercado de Datos para la Oficina Nacional de Estadísticas.

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

Autor

Julio Ernesto Ortiz Sierra

Tutores:

Ing. Asnioby Hernández López

Ing. Alberto Limia Navarro

Dr. C. Pedro Yobanis Piñero Pérez

Ciudad de la Habana, Junio de 2009

Lo que sabemos es una gota de agua; lo que ignoramos es el océano.

Isaac Newton

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

_____	_____
Julio Ernesto Ortiz Sierra	Dr C. Pedro Y. Piñero Pérez
Autor	Tutor
_____	_____
Ing. Asnioby Hernández López	Ing. Alberto Limia Navarro
Tutor	Tutor

AGRADECIMIENTOS

A mis padres, Julio y María, por ser mis ejemplos, mis amigos y apoyarme en todas las decisiones, que durante este largo andar, me ha impuesto el destino.

A mis tres hermanos, Alejandro, Ana Rosa y Lis María, por estar siempre a mi lado.

A mis queridas abuelas, Nuñú y Fefa, que tan especial son conmigo.

A mi novia Lissette por su certera ayuda en la investigación, su confianza y cariño.

A mis tíos Janet y Jorge por el apoyo y confianza que siempre me han depositado.

Al resto de mi familia, en especial a mis abuelos Miguel y Fidel, a todos gracias.

A mis Tutores por la consagración con que trabajaron junto a mí durante el desarrollo de la investigación.

A Yonelbys por su incalculable ayuda dentro de la investigación.

A mis amigos por haber estado siempre apoyándome en los momentos buenos y malos.

En fin, a todas las personas que de una forma u otra contribuyeron con el éxito del presente trabajo, les agradezco de todo corazón.

DEDICATORIA

Dedico esta investigación a quien fuera uno de mis grandes modelos a seguir, a la memoria de abuelo Tito.

RESUMEN

El presente trabajo de diploma se enmarca en el área de los almacenes de datos y las técnicas Procesamiento Analítico en Línea (OLAP, por sus siglas en inglés) para el análisis de información estadística. Comprende una revisión minuciosa y detallada de las metodologías, tendencias y mejores prácticas para el desarrollo de este tipo de soluciones. La investigación toma como referencia la metodología formulada por Ralph Kimball.

Como resultados se exponen las estructuras dimensionales para el modelo estadístico de “Indicadores Generales” que comprende las dimensiones, jerarquías, tablas de hechos y medidas necesarias para soportar los análisis estadísticos. Además se definen los mecanismos de extracción, transformación y cargas de los datos correspondientes al modelo. Otros aportes de la solución comprenden los procedimientos para agregar datos en función de alcanzar rendimientos aceptables ante peticiones que requieren el procesamiento de un número significativo de tuplas (del orden los cientos de miles de tuplas). Por otra parte se presentan las estrategias de indexado y particionamiento que constituyen un referente para incorporar nuevos modelos. Su principal novedad consiste en su implementación utilizando herramientas libres.

ÍNDICE

ÍNDICE

ÍNDICE DE FIGURAS Y TABLAS

INTRODUCCIÓN

La creciente evolución y desarrollo de las Ciencias de la Información le imponen al mundo una nueva forma de concepción para enfrentarse a los problemas que día a día se le presentan. Esta nueva forma de conceptualizar las soluciones a estos problemas se van fusionando indiscutiblemente al aumento de la explotación de las Tecnologías de la Información y las Comunicaciones (TIC) en la sociedad, mostrándose como un requisito indispensable para lograr un desarrollo sostenible e incremental.

El control de los datos estadísticos dentro de la infraestructura de un país constituye el eslabón principal para la toma de decisiones en los diferentes sectores socioeconómicos. Cuba posee una larga historia en materia de estadística. La entidad rectora de este tema en el país es la Oficina Nacional de Estadísticas (ONE) la cual mediante su Sistema Estadístico Nacional (SEN), organiza, dirige, controla y regula esta actividad.

La ONE tiene una estructura institucional distribuida territorialmente en las provincias y municipios del país. Existen 16 oficinas provinciales, una en cada provincia, una en el municipio especial Isla de la Juventud y otra adicional en Ciudad de la Habana; y 169 oficinas municipales, subordinadas a las ONE provinciales, las cuales son las encargadas de interactuar directamente con los Centros Informantes (CI) siendo estos el último eslabón de la cadena de la actividad estadística. Todas estas oficinas tienen atención administrativa y metodológica por la oficina nacional.

La información estadística en Cuba está agrupada en el SEN, que se divide en las siguientes áreas:

- Sistema de Información Estadística Nacional (SIEN): incluye los formularios estadísticos recopilados por la Oficina Nacional de Estadísticas a través de sus dependencias y los ministerios con fines nacionales.

- Sistema de Información Estadística Territorial (SIET): incluye los formularios estadísticos recopilados por la oficina de estadística territorial y entidades territoriales con fines territoriales aprobados por la ONE.

- Sistema de Información Estadística Complementaria (SIEC): incluye los formularios estadísticos recopilados por todos los ministerios y entidades para sus propios fines aprobados por la ONE.

La información estadística la brindan los CI, los que pueden ser: empresas, instituciones y organizaciones; según lo previsto en el programa de captación de datos convenido con todas las entidades a principio de cada año. La información que se recoge en esta institución se difunde principalmente en papel y en CD ROM provocando afectaciones para su digitación, pérdidas de información y aislamiento; complejizándose la situación si se consideran los 6833 centros informantes existentes a todo lo largo del país.

Los centros informantes por su parte han generado, con el pasar de los años, un histórico de datos disponibles en los más disímiles formatos. A medida que pasa el tiempo esa información se incrementa debido a la propia gestión estadística y, aunado a esto, los avances tecnológicos reflejados en las redes y las telecomunicaciones diversifican más esta situación.

La Oficina Nacional de Estadísticas gestiona los datos asociados a indicadores sociales y económicos; presentando limitaciones para su consulta e integración. Los datos históricos almacenados en archivos DBF, particionados por fecha de captación, año y modelo, provocan un retraso significativo en la entrega de la información a las organizaciones políticas y áreas vitales de la economía nacional impactando negativamente en las estrategias políticas, sociales y económicas del país.

El objeto social de la ONE conlleva a disponer de esta información de manera oportuna y con alta calidad para utilizarla como elemento de apoyo a la toma de decisiones a nivel nacional. La recuperación parcial o total de esta información, bajo las condiciones actuales, genera una gestión compleja y los resultados obtenidos no siempre se alcanzan en el tiempo y con la calidad requerida.

Problema

La insuficiente integración de los datos estadísticos está afectando la toma de decisiones ajustadas a la realidad del país.

Objeto de Estudio

Almacenes de Datos

Campo de Acción

Mercados de Datos

Objetivo General

Diseñar e Implementar un Mercado de Datos para la Oficina Nacional de Estadísticas.

Objetivos Específicos

1. Evaluar las tendencias de los Mercados de Datos y sus principales implementaciones.
2. Modelar e implementar el repositorio de datos, con un enfoque dimensional, para el Modelo Estadístico de Indicadores Generales.
3. Definir la estrategia de Extracción, Transformación y Carga de los datos históricos del Modelo Estadístico de Indicadores Generales.
4. Validar la solución desarrollada mediante la realización de pruebas de volumen y carga.

Hipótesis

La integración de los datos estadísticos contribuye a la toma de decisiones ajustadas a la realidad del país.

Tareas Científicas de la Investigación:

1. Estudiar los temas relacionados a las mejores prácticas en el desarrollo de Mercados de Datos.
2. Definir la metodología a utilizar en el desarrollo.
3. Seleccionar el proceso del negocio a modelar.
4. Definir la arquitectura del sistema.
5. Elegir la granularidad del proceso del negocio.
6. Definir las dimensiones del Mercado de Datos (MD).
7. Definir los hechos mensurables asociados a las dimensiones definidas.
8. Estructurar el modelo dimensional.

9. Transformar del modelo dimensional al diseño físico.
10. Definir estrategia de Extracción, Transformación y Carga de los datos.
11. Validación de la solución.

Estructura del Trabajo

El presente trabajo está compuesto por 3 capítulos, de los cuales el primero aborda los temas relacionados con la fundamentación teórica, el segundo trata sobre la descripción de la solución y el tercero está orientado al análisis y validación de los resultados.

En el Capítulo 1 los puntos implicados están referidos a un estudio sobre los Sistemas de Almacenes de Datos, sus principales metas y características, los principales elementos que los componen, un estudio del estado del arte tanto a nivel mundial como nacional de sus desarrollos, las metodologías existentes y las principales herramientas para el desarrollo de los mismos, así como la justificación de su uso.

En el Capítulo 2 se abordan aspectos concernientes a la descripción de la solución, específicamente, a la descripción de las fuentes a integrar, definición de las áreas de análisis, la arquitectura, el diseño, la interacción entre los componentes, la implementación del Mercado de Datos, el cubo de datos y el modelo dimensional propuesto.

Finalmente en el Capítulo 3 se detallan las temáticas referidas a la normalización, calibrado de la base de datos (BD), análisis del rendimiento, pruebas, así como la validación general del Sistema y el análisis de los resultados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

Desde un principio, las bases de datos se convirtieron en una herramienta fundamental de control y manejo de operaciones comerciales. De ahí que en un corto período de tiempo las grandes empresas y negocios acumularan un cuantioso número de información que ya alcanzaba una dimensión considerablemente voluminosa. Con la acumulación de esta información se presentó la problemática de cómo darle un fin útil debido a que en ella estaba almacenada la mayor parte de las operaciones comerciales de las mismas.

La solución sería unificar las diferentes fuentes de información de las cuales disponían, en un único lugar, al que sólo se le incorporaría información relevante, sobre la base de una estructura organizada, integrada, lógica, dinámica y de fácil explotación. La respuesta a esto fueron los Almacenes de Datos o Data Warehouse (DW), como se conocen mundialmente. (Kimball, y otros, 2002), (Inmon, 2005).

Desde su aparición en la década de los 90's hasta la fecha la tecnología de warehousing ha venido madurándose y posicionándose como la variante más acertada para la realización de análisis de información histórica. Se ha convertido, para quienes lo explotan, en una potente herramienta para la recuperación efectiva de las más complejas consultas o el más grande o engorroso reporte, además de servir como base para la toma de decisiones, en un mundo empresarial cada vez más necesitado de oportunas estrategias de marketing y de sistemas que provean de la información necesaria para la definición de objetivos y metas reales ajustadas a la dirección eficiente de las empresas, como exige el mercado global.

1.1 Los Sistemas de Almacenes de Datos

Haciendo una referencia a la justificación del surgimiento de los Sistemas de Almacenes de Datos (DWS, por sus siglas en inglés) puede referirse que, con la aparición de la computación en los años 70's se presenta el primer problema que consistía en el almacenamiento de toda la información que se generaba día tras día. Los Sistemas de Ficheros Relacionados constituyen el primer acercamiento; este es sucedido por los Sistemas de Bases de Datos que prestaban un servicio con mayor rapidez. Con el transcurso de

los años y el aumento de la complejización de las tecnologías la información que se almacenaba comenzó a aumentar a tal punto que en ocasiones resultaba demasiado complicado analizarla.

Con la aparición de los DWS se abrió una puerta para proporcionar respuesta a esta situación. Ha llevado casi 20 años la estandarización y madurez de esta nueva tecnología. En la actualidad se puede afirmar que los avances alcanzados confirman que ya es una tecnología madura, estable y soluciona la problemática presentada, lo que no significa que no continúe en constante evolución.

Para adentrarse en el tema de los DW se considera necesario hacer referencia a las principales personalidades que se desenvuelven en este campo. Existiendo diversas tendencias y formas de conceptualizar está terminología que, aunque difieren en algunos aspectos, todas giran sobre el mismo eje central.

En el libro "*Mastering Data Warehouse Desing, Relational and Dimentional Techniques*" (Imhoff, Galemmo & Geiger, 2003) se plantea que la definición universalmente aceptada es la desarrollada en los 90's por William H. Inmon, más conocido como Bill Inmon, "Son un conjunto de datos o*rientados a un tema, integrados, de tiempo variante y no volátiles usados en la estrategia de toma de decisiones administrativas.*"

En 1996 se esboza que "Los Almacenes de Datos entraron en existencia para satisfacer las necesidades, consolidando e integrando la información de fuentes internas o externas en función de organizarla en un formato útil, para soportar a las decisiones empresariales". (Wang, 2006)

Además en el citado libro más adelante se enuncia que "Los Almacenes de Datos se han venido reconociendo cada vez más como una herramienta efectiva de las organizaciones para transformar los datos en información útil y estratégica para la toma de decisiones."

"Si el objetivo de los sistemas operacionales es la **ejecución** del proceso del negocio, los DWS soportan la **evaluación** del proceso del negocio." (Adamson, 2006)

Existen otros autores, como es el caso de Ralph Kimball, mundialmente reconocido por sus libros sobre el tema que plantea "…los Almacenes de Datos son una copia de los datos de la transacción estructurados específicamente para la pregunta y el análisis." (Kimball, y otros, 2002)

Como se puede apreciar los autores rondan sobre el mismo núcleo acerca del concepto de DW. Todos brindan desde su punto de vista su percepción sobre el tema. Aunque se expresan de forma diferente queda claro que los Almacenes de Datos son estructuras que se definen en función de temas específicos donde la información histórica debe estar integrada, robusta ante los cambios que puedan afectar a la organización y que su objetivo principal, y es lo que define su razón de ser, es servir de ayuda a la toma de decisiones empresariales.

1.1.1 Mercados de Datos

Al hacer un análisis de la bibliografía existente sobre la tecnología de data warehousing se puede constatar que existen autores que utilizan los conceptos de “Almacén de Datos” y “Mercados de Datos” indistintamente refiriéndose al mismo tema aunque realmente no definen un concepto idéntico.

Los Mercados de Datos, también conocidos como Data Mart, son un subconjunto de datos de un DW donde se almacenan la mayoría de las actividades de análisis que en el entorno de Inteligencia de Negocio se llevará a cabo. (Imhoff, y otros, 2003)

Los MD son DW orientados a temas específicos o aplicaciones específicas y contienen datos de sólo una línea del negocio como puede ser ventas o marketing. La mayor diferencia entre ellos es el ámbito de la información que contienen debido a que en los DM es más pequeño y los datos se obtienen de un menor número de fuentes y comúnmente el tiempo de desarrollo es menor. (Hobbs, y otros, 2005)

Un DM es una alternativa de solución al igual que DW a los problemas antes planteados porque el diseño y construcción son similares, además de poseer una secuencia común. La diferencia entre estas dos estructuras se basa principalmente en que los DM están enfocados en un área de negocio específica, mientras que un DW entrega información a nivel corporativo. (Peñaloza, 2008)

Un concepto más amplio sería: “Un conjunto flexible de datos, idealmente basado en el dato más atómico posible (granular) para ser extraído de las fuentes operacionales y presentado en un modelo simétrico (dimensional) que es más resistente cuando se enfrentan con las más inesperadas consultas de los usuarios (…) Podemos decir que los MD están conectados con la arquitectura de los DW en su forma más simple y que representan los datos de un sólo proceso del negocio a la vez.” (Kimball, y otros, 2002)

1.1.2 Almacenes de Datos o Mercados de Datos

En 1998 Bill Inmon declaró “El elemento más importante que enfrentan los directores de tecnologías de la información en este año es si construir primero los Data Warehouse”. (Ponniah, 2001) Esta afirmación todavía está vigente en la actualidad.

Existen un conjunto de interrogantes que resultan imprescindibles evaluar antes de decidir realizar la construcción de un Almacén de Datos (Ponniah, 2001):

- ¿La aproximación se realizará de arriba hacia abajo (top-down) o de abajo hacia arriba (bottom-up)?
- ¿Empresarial o departamental?
- ¿Cuál primero el DW o el DM?
- ¿Construir un piloto o directamente el DW completo?
- ¿Mercados de Datos dependientes o independientes?

Las respuestas de estas preguntas conllevan a una planificación profunda de lo que realmente se va a desarrollar. Cada respuesta es crucial acerca de la decisión correcta sobre si utilizar un Almacén de Datos Corporativo o utilizar un Mercado de Datos Departamental.

En su libro “*Data Warehouse Fundamentals*” Poulraj Ponniah determina un conjunto de elementos que dan claridad sobre las diferencias existentes entre ambos conceptos.

Estas diferencias se muestran en la Tabla 1 siguiente: (Ponniah, 2001)

Tabla 1: Comparación entre Almacén de Datos y Mercado de Datos

Almacén de Datos	Mercado de Datos
Corporativo o red empresarial	Departamental
Es la unión de todos los MD	Un simple proceso del negocio
Los datos son recibidos desde el área de procesamiento	Unión en forma de estrella (hechos y dimensiones)
Consultas sobre la presentación de recursos	Tecnología óptima para el acceso a los datos y el análisis
Estructura para vista corporativa de los datos	Estructura para adaptarse a la vista de los datos departamentales

Tomando como base lo expresado por Ponniah, los MD, como estructura, son un Almacén de Datos pero reducido a un departamento específico sirviendo como fuente de análisis del tema que concierne a dicho departamento. La unión de todos los MD de la organización es lo que conformaría la vista global de los datos, es decir, el Almacén de Datos Corporativo.

1.2 Metas de los Almacenes de Datos

Con la complejización de las soluciones empresariales y la necesidad de que la información cada día sea de mayor utilidad para los directivos de las organizaciones, obliga a la realización de sistemas más abiertos y dinámicos que permitan su adaptación ante los cambios que se imponen en la actualidad. Ante esta disyuntiva, con la maduración de las técnicas de desarrollo de Almacenes de Datos y Mercados de Datos se han definido un conjunto de metas que deben cumplir estos sistemas (Kimball, y otros, 2002).

1. **Deben hacer fácilmente accesible la información.**

   La información que se almacena debe ser accesible en todo momento, diseñándose mediante datos que sean intuitivos y obvios, para los usuarios del negocio y no para los desarrolladores. Se debe etiquetar significativamente el contenido que almacena. El diseño de las estructuras debe soportar que los usuarios combinen y separen los datos en un sinfín de combinaciones (a este proceso se le denomina en este mundo como slicing y dicing). La información que está almacenada debe ser recuperada con un tiempo mínimo de espera.

2. **La información de la organización debe ser presentada de forma consistente.**

   Refiere que la información almacenada debe ser creíble. Los datos deben ser ensamblados o agregados de las fuentes que existen alrededor de la organización, limpiados, con calidad asegurada y descargado sólo cuando es de ayuda para el consumo de los usuarios.

   Se debe permitir que la información de un proceso determinado del negocio pueda ser comparada con la de otro proceso. Cuando se habla de consistente es equivalente a decir de alta calidad, la consistencia también implica que la definición común del contenido almacenado está disponible para los usuarios.

3. **Deben ser adaptables y resistentes al cambio.**

   El diseño de los almacenes de datos debe ocuparse del inevitable cambio que proponen las condiciones del negocio, los datos, la tecnología, etc. Los cambios deberán ser elegantes, significando que no invalidan datos existentes o aplicaciones.

4. **Deben ser un baluarte seguro que apoye los recursos de información.**

   La información no siempre puede ser consultada por todos, lo que implica tener un control de acceso efectivo para la información confidencial de la organización. Se deben establecer niveles de seguridad para cada funcionalidad que va a brindar el almacén.

5. **Debe servir como base para mejorar la toma de decisiones.**

   Los datos almacenados deben ser correctos y a la vez ser útiles para dar soporte a la toma de decisiones empresariales. La organización de la información debe ser lo suficientemente dinámica y efectiva para que pueda ser servida oportunamente a los usuarios.

6. **La comunidad del negocio debe aceptar el DW para poder ser juzgado como exitoso.**

   No importa que tan elegante sea la solución usando los mejores productos y plataformas si la comunidad de la organización no depende del DW y no es explotado activamente después del entrenamiento entonces ha fallado la prueba de aceptación.

Las metas definidas por Kimball para los Almacenes de Datos son adaptables totalmente a los Mercados de Datos debido a que según el mismo autor referencia los MD son la unidad básica de los DW corporativos. Cada MD que se diseñe debe cumplir los objetivos planteados pero adaptándolo específicamente a un proceso del negocio, lo que significa, mirándolo de este punto de vista, el universo de información útil y necesaria para el soporte a la toma de decisiones empresariales estaría enmarcada en un departamento específico.

1.3 Características Generales

Los DW son el corazón del ambiente arquitectónico y la base de la toma de decisiones. Según Inmon existen 4 características fundamentales que deben cumplir (Inmon, 2005).

1. **Orientado al tema**

   Los sistemas operacionales son clasificados dentro de las compañías según las funcionalidades que realicen, por ejemplo, en una compañía de seguro podrían ser: tratamientos de autos, vida, salud, entre otros. En el ambiente operacional las bases de datos se diseñan orientadas a las funcionalidades que cada sistema ofrezca.

   Los Almacenes de Datos se diseñan orientados a los aspectos que son de interés para la organización. Temas como ventas, clientes, vendedores constituyen ejemplos de estos aspectos. Cada compañía posee un conjunto único de temas a analizar. Esta información generalmente es almacenada en fuentes operacionales pero orientadas a un programa específico y dan respuesta a este operativamente. Aquí es donde actúan las estructuras del DW, organizando toda esta información para que sea útil desde el punto de vista de análisis para la organización y que la información cobre valor a nivel empresarial.

   La selección de los temas dentro de la organización tipifica el proceso más crítico para el diseño e implementación de las estructuras del DW. Cada MD debe englobar un tema específico con el fin de permitir su análisis.

2. Integrado

De todos los aspectos de la tecnología warehousing, la integración es lo más relevante. Los datos, en el almacén de datos, se alimentan de múltiples y dispares fuentes. Estas fuentes almacenan la información con el formato que es entendido por el sistema que los va a utilizar lo que evidencia claramente las disímiles variantes que se puede almacenar una misma información. Problemas tales como la codificación, medidas de atributos, múltiples fuentes, conflicto de llaves, entre otros, concurren cotidianamente a la hora de realizar la integración.

Los datos, en el momento en que van a ser almacenados; son convertidos, reformateados, resecuenciados y resumidos lográndose crear una sola imagen física de ellos. El objetivo es que cuando se referencien tengan una misma identidad dentro de las estructuras y puedan ser analizados desde diversos puntos de vista.

3. No volatilidad

Dentro de la teoría de DW está definido que la información que es almacenada no se puede modificar directamente. Las únicas dos acciones que se permiten residen en la carga inicial de los datos y el acceso a la información. Todos los procesos establecidos dentro del concepto “gestión” como son inserción, modificación y actualización se realizan en el ambiente operacional, es decir, todo error que se almacene en el Mercado de Datos se corrige en la fuente de donde vino y se realiza nuevamente el proceso de carga de los datos.

4. Tiempo variante

Tiempo variante significa que cada unidad de dato almacenada es exacta a partir de un momento en el tiempo. En algunos casos el registro está sellado en el tiempo y en otros posee una fecha de transacción, pero en cualquier caso existe una forma de mostrar el momento en el tiempo cuando se realizó el registro.

Generalmente el horizonte de tiempo que se establece en este tipo de estructura es de años, de 5 a 10 años mínimo a diferencia del ambiente operacional donde la información que se maneja es del momento en que se hizo la transacción, entre 2 y 3 meses aproximadamente. (Inmon, 2005)

1.4 Componentes de los Almacenes de Datos

Los Almacenes de Datos están compuestos por una serie de elementos que definen, en su conjunto, el ambiente que estos poseen. Aunque cada desarrollo de DWS son diferentes debido a la especificidad de las organizaciones, generalmente, cumplen con la realización de los componentes que a continuación se proponen.

La Figura 1 muestra los componentes del Almacén de Datos y la relación entre ellos: (Kimball, y otros, 2002)

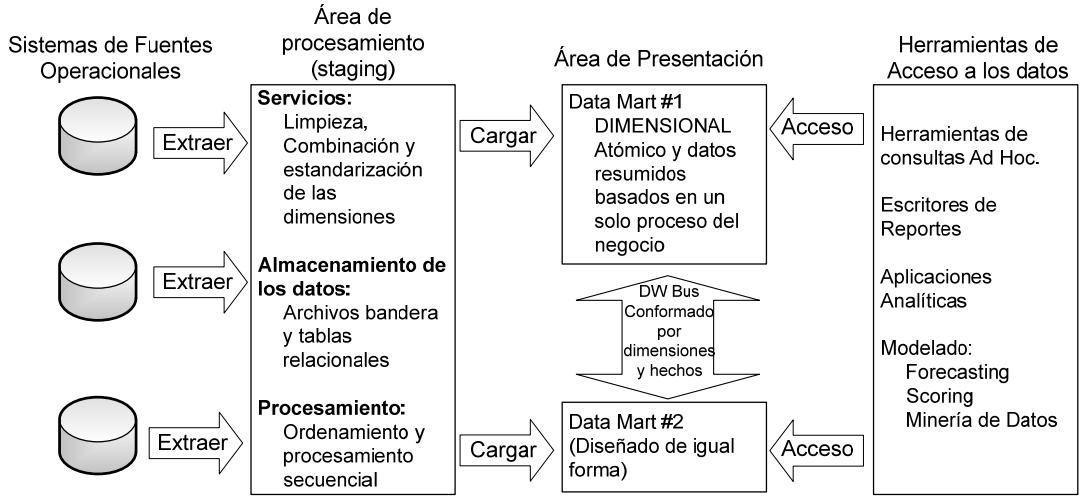

Figura 1 Relación entre los componentes de un Almacén de Datos

1.4.1 Sistema de fuentes operacionales

Estos son los sistemas que poseen las compañías o empresas para la gestión de sus transacciones diarias. Estas transacciones son almacenadas en los más diversos formatos, desde una base de datos relacional hasta cualquier tipo de ficheros, ya sea Excel, XML, DBF, texto plano, entre otros.

Se encuentran localizados fuera del repositorio debido a que se tiene poco o ningún control sobre el volumen y formato de los datos de estas fuentes. Las prioridades principales de este componente es el procesamiento, el rendimiento y la disponibilidad. Generalmente realizan salvas de la información que

gestionan y sólo trabajan con los datos generados en un período corto de tiempo para hacer las recuperaciones de forma más óptima. También existe la posibilidad de que sean fuentes creadas manualmente debido a que no posean un sistema que las procese.

1.4.2 Área de Procesamiento (staging)

Es el área que almacena los datos temporalmente y realiza un conjunto de procesos comúnmente llamados de Extracción, Transformación y Carga (ETL, por sus siglas en inglés). Realiza la función de interfaz entre las fuentes operacionales y el área de presentación.

En este componente es donde se invierte la mayor cantidad de tiempo y esfuerzo durante el desarrollo del almacén. Se realiza el proceso de extracción de los datos de las diversas fuentes operacionales que se deseen integrar, teniendo como principal tarea la de almacenar toda esa información en bases de datos relacionales, generalmente, para realizar el análisis y procesamiento de los datos. Una vez los datos almacenados en bases de datos temporales se procede a su limpieza donde se detectan inconsistencias, duplicaciones, errores de formato e inexistencias, estandarizándose la información almacenada en diferentes fuentes. Estas transformaciones son las que sirven de apoyo para realizar la carga de los datos, hacia el DW, en el área de presentación.

1.4.3 Área de Presentación

En este componente los datos se encuentran organizados, almacenados y disponibles para ser consultados, reportados o analizados por parte de los usuarios finales. Es donde se encuentra la información, diseñada mediante esquemas dimensionales, que ha sido definida por los usuarios como útil para la toma de decisiones.

Generalmente esta área es referenciada como una serie de Mercados de Datos integrados donde cada uno se encuentra representando a un proceso específico del negocio.

1.4.4 Herramientas de Acceso a Datos

En este componente se usa la palabra herramientas para referirse a la variedad de capacidades que pueden ser provistos a los usuarios del negocio para el soporte a la toma de decisiones. Su actividad principal es la de consultar el área de presentación del Almacén de Datos.

El mismo puede abarcar desde una simple o personalizada herramienta de consulta hasta una compleja y sofisticada aplicación de modelado o de minería de datos.

1.5 Modelo Entidad-Relación y Modelo Dimensional

1.5.1 Modelo Entidad-Relación

Un diagrama o modelo entidad-relación (a veces denominado por su siglas, *E-R* "Entity relationship", o, "DER", Diagrama de Entidad Relación) es un lenguaje para el modelado de datos de un sistema de información. Estos modelos expresan entidades más relevantes para el sistema, sus inter-relaciones y propiedades. Trabajan dividiendo los datos en muchas entidades discretas donde cada una se convierte en una tabla física en la base de datos operacional.

Los sistemas de información que se realizan bajo estas directrices comúnmente se denominan sistemas OLTP (Online Transaction Processing). Su principal función es reflejar el estado y funcionamiento de las empresas mediante el registro de las operaciones que realizan diariamente.

Los modelos entidad – relación no son recomendables para el diseño de los almacenes de datos debido a que no garantizan la recuperación óptima del gran cúmulo de información que se almacena. Además estos diagramas tienden a resultar en un diseño normalizado mientras que en un almacén de datos este aspecto no es un requisito a tener muy en cuenta. (Hobbs, y otros, 2005)

1.5.2 Modelo Dimensional

A diferencia de los clásicos sistemas de bases de datos que presentan sus estructuras diseñadas mediante el modelo Entidad-Relación los Almacenes de Datos se diseñan mediante un Modelo Dimensional. Poseen la misma información que el DER pero la organiza de forma diferente para

garantizar la velocidad y eficiencia en la recuperación de la misma. Una de sus características principales es que no necesita una predefinición de los reportes debido a que se diseñan de forma tal que cubra el universo de variantes que los usuarios necesiten consultar la información almacenada. En la Figura 2 se muestra la estructura espacial que posee este tipo de diseño.

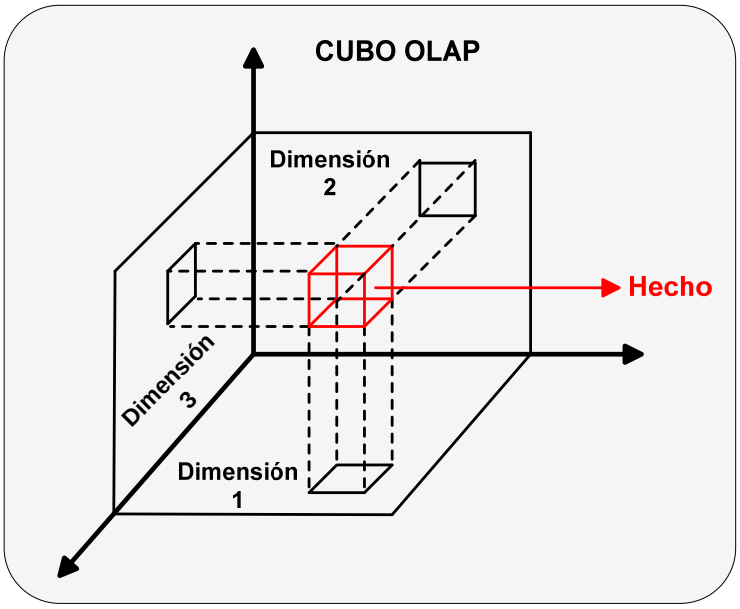

Figura 2 Estructura de un Cubo OLAP

Para la materialización física de este tipo de modelo se utiliza comúnmente la propuesta realizada por Ralph Kimball llamada “esquema estrella” que consiste en una tabla central denominada “tabla de hechos” y un conjunto de pequeñas tablas, llamadas “dimensiones”, que se relacionan a esta tabla central. Se le denomina estrella por su similitud con una estrella natural debido a que las dimensiones poseen entre sus relaciones una con la tabla de hechos, ver Figura 3.

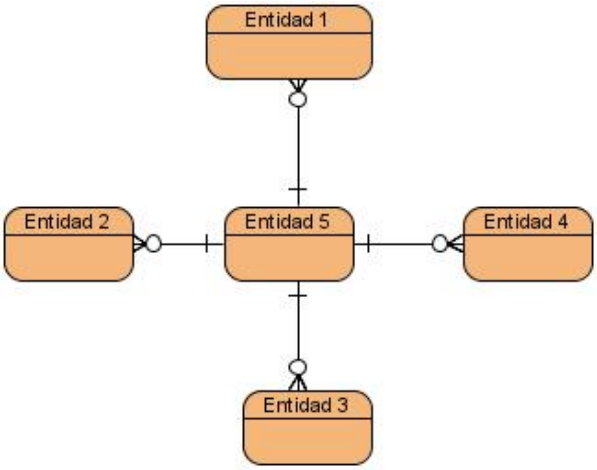

Figura 3 Representación del Esquema Estrella

Existen otras estructuras que surgen producto a modificaciones realizadas al esquema estrella. En este sentido se tiene el Copo de Nieve (*Snowflake*, en inglés)*;* la citada estructura tiene como objetivo primordial su uso para el ahorro de espacio de almacenamiento. Se dice que una dimensión se encuentra "snowflaked" cuando los atributos de baja calidad se llevan a tablas separadas. La utilización de este tipo de estructura posee algunas deficiencias debido a que hace las presentaciones más complejas y afecta el rendimiento de la recuperación de las consultas.

Para uso similar al del Copo de Nieve se puede utilizar las Subdimensiones pero sólo es recomendable utilizarlas cuando existe un conjunto de atributos, dentro de las dimensiones, que son necesarios aislar. A este conjunto de estructuras anteriormente descritas se les puede añadir la Constelación de Hechos donde su principal característica es que múltiples tablas poseen las mismas dimensiones, de esta forma se pueden utilizar diversas medidas, separadas en diferentes tablas de hechos, definidas por las mismas dimensiones.

El modelo dimensional divide el mundo de los datos en dos grandes conjuntos: las medidas y las descripciones del entorno de estas medidas. Las medidas, que generalmente son numéricas, se almacenan en las tablas de hechos y las descripciones de los entornos que son textuales se almacenan en las tablas de dimensiones. Las tablas de hechos son las tablas primarias en el modelo dimensional y contiene los valores del negocio. Los hechos más comunes son valores numéricos. Cada tabla representa

una interrelación **muchos – muchos** y contiene dos o más llaves extranjeras que acoplan con sus respectivas tablas de dimensiones. (Ponniah, 2001)

Tablas de Hechos

La tabla de hechos es la tabla primaria en el modelo dimensional donde el rendimiento de las mediciones numéricas del negocio es almacenado. (Kimball, y otros, 2002) Generalmente cada tabla de hecho define un mercado de datos determinado debido a que en ellas se almacena la información concerniente al tema en cuestión, ejemplo ventas, clientes, vendedores, etc.

Al utilizarse la palabra “hecho” se refiere a la medida del negocio. Cada fila de esta tabla corresponde a un hecho determinado y a su vez cada conjunto de hechos dentro de la tabla de hechos referencian a la misma granularidad. La principal condición que debe cumplir las tablas de hechos es que el hecho debe almacenarse de tal forma que su valor sea numérico y a su vez sea aditivo para así poder realizar cálculos sobre él, ya sea por ciento, sumas, igualdades, etc.

Tablas de Dimensiones

Las tablas de dimensiones son las compañeras integrales de las tablas de hechos, ellas contienen la descripción textual del negocio. En el modelo dimensional, las tablas de dimensiones poseen varios atributos que en su conjunto definen una fila en la tabla de dimensión.

Los atributos de las dimensiones sirven como fuente primaria de las restricciones de las consultas, agrupaciones y las etiquetas de los reportes. Ellos desempeñan un rol de vital importancia dentro del Almacén de Datos debido a que son las llaves que hacen el DW usable y entendible. Estos atributos son las llaves de entrada a los hechos o medidas almacenadas.

La calidad de todo Almacén de Datos se mide por la definición de los atributos de las dimensiones. Su poder es directamente proporcional a la calidad y profundidad de estos atributos. (Kimball, y otros, 2002)

1.6 Modos de Almacenamiento de Datos

Existen tres modelos para el proceso analítico en línea (OLAP, por sus siglas en inglés) de la información ROLAP, MOLAP y HOLAP. El proceso de análisis se realiza de igual forma lo que varía en uno y otro caso es la metodología de almacenamiento. La forma de almacenamiento es crítica para garantizar la velocidad de recuperación de la información, las zonas de ubicación de las agregaciones y el procesamiento de los datos en general.

1.6.1 ROLAP

En el Procesamiento Analítico Relacional en Línea (*Relational Online Analitical Process*, en inglés) los datos son almacenados en filas y columnas de forma relacional. Este modelo presenta los datos a los usuarios en forma de dimensiones de negocio. Con el fin de ocultar las estructuras de almacenamiento y presentar los datos dimensionalmente es creada la semántica de las etiquetas de los metadatos. Ellas soportan el mapeo de las dimensiones a las tablas relacionales. Estos metadatos también son almacenados en tablas relacionales.

El modelo ROLAP es usado fundamentalmente sobre información que no se consulta frecuentemente debido a que no es muy óptimo en este sentido. Por ejemplo información histórica de muchos años de antigüedad.

La **¡Error! No se encuentra el origen de la referencia.** muestra la arquitectura que presenta el modelo ROLAP (Ponniah, 2001)

1.6.2 MOLAP

Por su parte el Procesamiento Analítico Multidimensional en Línea (*Multidimensional Online Analitical Process*, en inglés) almacena los datos dimensionalmente a diferencia del ROLAP. Aquí las estructuras de los datos están fijas para que la lógica, al procesar la información, pueda estar basada en métodos bien definidos para establecer las coordenadas del almacenamiento de los datos.

Las estructuras de almacenamiento son grandes arreglos dimensionales que son una copia de la fuente de datos y persisten físicamente en la misma estación de trabajo donde está instalada la herramienta Data

Warehousing. Esto provoca que el acceso a la información almacenada se realice de forma más rápida y efectiva utilizándose en depósito donde el tiempo en la velocidad de respuesta es crítico.

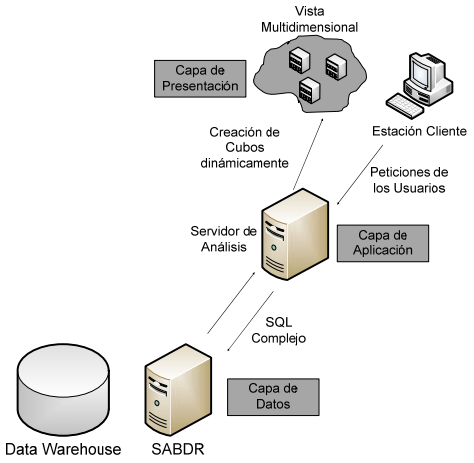

Figura 4 Modelo de almacenamiento ROLAP

La Figura 5 muestra la arquitectura que presenta el modelo MOLAP (Ponniah, 2001)

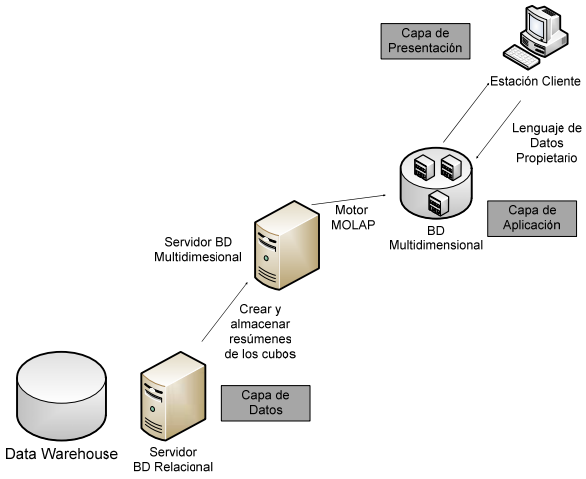

Figura 5 Modelo de almacenamiento MOLAP

1.6.3 HOLAP

El modo de almacenamiento HOLAP (*Hybrid Online Analitical Process, por sus siglas en inglés),* como su nombre lo indica, es un híbrido entre los métodos ROLAP y MOLAP. Permite almacenar una parte de los datos como en un sistema MOLAP y el resto como en uno ROLAP. El grado de control que el operador de la aplicación tiene sobre este particionamiento varía de unos productos a otros. Posee dos tipos de particionamiento:

Particionamiento Vertical

Almacena las agregaciones como un MOLAP para mejorar la velocidad de las consultas, y los datos se detallan en ROLAP para optimizar el tiempo en que se procesa el cubo.

Particionamiento Horizontal

En este modo HOLAP se almacena una sección de los datos, normalmente los más recientes (por ejemplo particionando por la dimensión *tiempo*) en modo MOLAP para mejorar la velocidad de las consultas, y los datos más antiguos en ROLAP. Además, se pueden almacenar algunos cubos en MOLAP y otros en ROLAP.

1.6.4 MOLAP versus ROLAP

La selección de uno u otro modelo depende de cuán importante sea el rendimiento de las consultas para los usuarios y de la tecnología disponible a utilizar. En la Tabla 1 se muestra una comparación entre ambos modelos basándose en Almacenamiento de los Datos, Tecnologías Subyacentes y Funciones y Características. (Ponniah, 2001)

Tabla 2 Comparación entre ROLAP y MOLAP

	Almacenamiento de Datos	Tecnologías Subyacentes	Funciones y Características
ROALP	Almacenamiento como tablas relacionales. Resumen detallado de los datos disponibles Volúmenes altos de datos Todos los datos de acceso están en la bodega almacenados.	Uso de SQL complejo para obtener los datos del depósito. Motor ROLAP en el servidor de análisis crea los cubos de datos sobre la marcha. Vistas multidimensionales en la capa de presentación.	Ambiente conocido y disponibilidad de herramientas Limitación en funciones de análisis complejas Realización de agregaciones no siempre son fáciles.
MOLAP	Almacenamiento como tablas relacionales. Diversos resúmenes de datos se mantienen en las BD propietarias Volúmenes de datos moderados. Resúmenes de acceso a datos detallados en BD multidimensionales.	Creación de cubos de datos prefabricados por el motor MOLAP. Tecnología propietaria para almacenar las vistas multidimensionales en arreglos no en tablas. Matriz de alta velocidad para la recuperación de los datos. Escasa tecnología de matriz de datos para gestionar la escasez de los resúmenes.	Acceso rápido. Fuerte librería de funciones para cálculos complejos Facilita el análisis independientemente de la cantidad de dimensiones. Amplitud en la capacidad de acción en el Drill-Down, Roll-Up y Slicing-Dicing.

1.7 Estado actual de los Almacenes de Datos

En la actualidad a nivel global se ha impuesto una gran competencia entre las grandes compañías donde se obliga a los directivos proponer ideas nuevas e innovadores dentro del campo gerencial. Esto provoca que en las empresas se haya venido elevando la necesidad de incorporar tecnologías de última

generación para crear nuevas estrategias de comercialización y mercado, basándolas principalmente, en los clientes debido a que la actual economía mundial así lo amerita. El análisis de la información que se utiliza y los datos históricos es la principal herramienta que se ha encontrado para contrarrestar esta problemática.

Esto provoca cambiar de concepción de almacenamiento de los datos. El tradicional modelo relacional no brinda facilidades en este sentido debido a que los sistemas OLTP trabajan con las transacciones diarias que manejan los diferentes departamentos pero no es común almacenar el histórico en ellos. Las consultas dentro de estos sistemas se complejizan exponencialmente al almacenar junto con la información diaria todo el histórico que posea la organización. Además que el modelo no está acorde con la percepción que poseen los usuarios sobre la gestión comercial dentro del negocio.

Basándose en esta experiencia las compañías de todo el mundo han comenzado a migrar hacia su modernización para posicionarse en la delantera en este sentido.

1.7.1 En el mundo

El mundo avanza y cada día las empresas tienen mayor número de aplicaciones automatizadas, almacenan la información diaria en grandes bases de datos, y necesitan conocer al momento su stock de inventarios, su volumen de ventas del día, tener sus precios actualizados, etc.

Con el transcurso del tiempo las empresas fueron almacenando un gran número de información en diferentes fuentes de datos (archivos, documentos de texto, bases de datos, etc.), y los directivos de las empresas se dieron cuenta de que ésta, podría ser útil pues reflejaba la mayoría de las operaciones diarias del negocio.

En 1994 el 90% de las empresas, según la revista Fortune 2000, planeaba implementar un Almacén de Datos entre 1994 y 1996. En 1996 el 90% de las grandes corporaciones consideraba adoptar la tecnología del DW. Hill Hostian, de la empresa Gartner, estimó que para el 2007, el 50% de los proyectos de inteligencia de negocio (Almacén de Datos), requerirán de un proveedor de servicios para librar los obstáculos debidos a falta de personal capacitado y recursos. (Introducción a los Datawarehouses, 2007)

En los mercados empresariales existe mayor competencia, por lo que las empresas requieren mayor celeridad en su rapidez y eficiencia en sus procesos, y precisión en la información para tomar decisiones adecuadas. Por esto, se pensó que lo ideal sería unificar las diferentes fuentes de información de las cuales se disponían, almacenándolas en un único lugar, de tal forma que solo se le incorporara información relevante. Este nuevo repositorio de datos debería tener una estructura organizada, integrada, lógica y dinámica, y además ser de fácil explotación.

A partir de esta problemática un gran número de compañías se dieron a la tarea de implementar la tecnología data warehousing para convertir a sus datos en información útil. En el mercado minorista es donde más incidencia ha tenido esta novedosa forma de utilizar los datos debido a la necesidad de estudiar los clientes potenciales y estar actualizado ante la gran competencia que existe en este sentido. En América Latina existen empresas como Telefónica de Argentina, Visa, Arcor, todas de Argentina; en México existen algunas como Wallmart, Procter & Gamble, Whirpol, Tv Azteca, Baxter, GNP, Warner Lambert y Sabre que también han venido incorporando el uso de los almacenes de datos para la toma de decisiones a nivel gerencial.

También se puede mencionar a American Stores (Estados Unidos), Canadian Tyre (Canadá), Owens Corning Glass (Estados Unidos), Karsten Ping Golf Clubs que han obtenido grandes avances en este sentido.

En Europa, pionero en este campo, existen empresas tales como Carrefour (España) WH Smith Books (Gran Bretaña), BonPreu (España), SAR Group (España), Great Universal (Gran Bretaña), Corte Inglés (Francia), Cortefiel (Francia), Eroski (Francia), Supermercados Casino (Francia), Migrosgenossenschaftsbund (Suiza), Otto Versand (Alemania), Helene Curtis (España).

También, grandes trasnacionales como, Coca Cola, Walt Disney, Nike, Maybelline, Adidas, 3M, Bosh Siemens se han incorporado a la utilización de los DW para la realización de estudios de mercado y de inteligencia de negocio.

La esfera bancaria y de seguros también han dado pasos concretos en este sentido con entidades como Banco Galicia en Argentina y Banco de México, Banorte y Banamex todos de México, Banco París de Francia, el European Central Bank perteneciente a la Unión Europea, Bancard en Paraguay, BBVA el

segundo banco más grande de España, Caja Madrid, Caja Extremadura, el Banco Guipuzcoano radicado en la ciudad de San Sebastián en España, la compañía de seguros ARAG en España son ejemplos palpables de estos pasos donde realizan estudios sobre inflación, población, monetarios, tendencias, etc.

El tema estadístico no ha estado exento a estas necesidades, en países como México, específicamente en el INEGI (Instituto Nacional de Estadísticas e Informática), se tiene la información almacenada en almacenes de datos y de esta forma es servido para la toma de decisiones a nivel gubernamental. También el Instituto Nacional de Estadística de Venezuela posee un conjunto de Mercados de Datos para realizar el análisis de las operaciones estadísticas más importantes que utilizan. El Instituto Nacional de Estadística de España es otro ejemplo de entidades de este tipo que actualmente están utilizando esta tecnología.

En el campo del transporte de carga es también muy útil la utilización de los almacenes de datos para el análisis de los datos históricos referentes al estudio sobre ventas, clientes, transportaciones, monitoreo de ganancias y futuras proyecciones. En este caso aparecen empresas de la envergadura de Cornrail, Union Pacific, Norfolk Southern, American President Lines, Delta, Lufthansa, QANTAS, British Airways, Air France, American Airlines, Canadian Airlines y SNFC.

En las telecomunicaciones, debido a que es una esfera bien dinámica y completamente competitiva, los DW se han utilizado para la monitorización de clientes, la prestación de servicios, los cobros y pagos, utilidades, marketing, en fin, para la realización de estudios necesarios para mantenerse en la preferencia de los clientes. Representante de estas empresas se encuentran Bouygues Telecom, la tercera compañía operadora más grande en telecomunicaciones inalámbricas en Francia, Jazztel, Vodafone, France Telecom y CRTVG la Compañía de Radio Televisión de Galicia.

En la medicina se utilizan para hacer estudios de patrones de comportamiento en pacientes con diferentes patologías, dar seguimiento a tratamientos y pacientes. Uno de los principales ejemplares que posee un almacén de datos dentro de su infraestructura tecnológica es La Congregación de Hermanas Hospitalarias, organización internacional de asistencia médica, activa en 24 países en Europa, América, África y Asia.

Otras organizaciones como Bacardí Martini (distribución de bebidas) utiliza la información de ventas existente en el DW para optimizar la utilización de recursos con el fin de lograr el máximo de ventas con un coste preestablecido de antemano.

Pierre Fabré Ibérica (laboratorio multinacional cosmético y farmacéutico) utiliza un DW comercial para el seguimiento de ventas por zona geográfica, organización comercial, por producto, cliente, cadena y campaña etc., integrado en la aplicación de red de ventas, produce también un extenso informe mensual requerido por la casa matriz francesa.

Pastas La Familia (producción y distribución de alimentos) cuenta con un DW comercial que se destaca por la integración de la información presupuestaria en el ámbito de familia de producto y cadena, genera hojas electrónicas con información real del año en curso, sobre las cuales el departamento correspondiente calcula los presupuestos del próximo año.

SEUR (empresa de mensajería y transporte de paquetes) posee un DW de más de 80 millones de registros para seguimiento estadístico de los movimientos operativos, que permite realizar unos análisis mucho más detallados y precisos de envíos por ejemplo por origen y destino, por volumen, peso o precios de envío.

Por último, en el ámbito informativo, el diario El Mundo cuenta con un DW cuyo objetivo es obtener información completa sobre la contratación de publicidad en sus medios.

1.7.2 En Cuba

En Cuba ha venido asentándose una cultura tecnológica sobre el tema. Aunque todavía faltan muchos aspectos por mejorar ya se han visto algunos ejemplos que han dado pasos firmes dentro de la rama. El ejemplo más elocuente de esto es el DWS comercial de la Corporación CIMEX. La cual se dedica fundamentalmente a la Exportación e Importación de mercancías. Forman parte de ella un conjunto de empresas que se encuentran enfocadas en diversos negocios, aquí se puede citar la red de Comercio Minorista y la Dirección de Logística, esta última dedicada al Comercio Mayorista. El mismo centra su atención en la actividad del comercio, principalmente en la gestión de inventario, permitiendo una gestión

de compra–venta eficiente, con una finalidad de disminuir los costos, sin afectar al cliente, permitiendo prestaciones eficientes y con la calidad requerida, aumentando las utilidades de las mismas.

En el XIII Concurso Nacional de Computación y en la Feria de Informática del 2002 se presentó un Almacén de Datos para CUBACEL desarrollado sobre plataforma Oracle con grandes resultados obtenidos a partir de su implantación. Existen otras entidades como UNION CUPET y COPEXTEL que en la actualidad se encuentran en el proceso de diseño e implementación de sus respectivos almacenes.

1.8 Metodologías para el desarrollo

El vocablo metodología, es la ciencia que estudia los métodos del conocimiento. Se refiere a los métodos o procedimientos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia. Además puede categorizarse como el conjunto de métodos que se rigen en una investigación científica o en una exposición doctrinal.

En múltiples disciplinas existen diferentes enfoques para abordar un mismo concepto o problema. La existencia de dichos enfoques enriquece de sobremanera la propia disciplina. Siendo más generalista, eso mismo sucede entre diferentes áreas del conocimiento y produce que el avance de la ciencia no se enquiste.

El diseño de un Almacén de Datos, como disciplina que ha alcanzado ya un grado de madurez considerable a lo largo de estos años, también presenta diferentes enfoques. En esta tecnología se ha destacado un conjunto de metodologías que definen y guían todo el ciclo de vida del desarrollo concreto. Existen 2 criterios bien identificados y que han marcado claramente su tendencia sirviéndole de guía a la comunidad mundial en cuanto a este tema.

Estas tendencias son la conocida como Metodología Kimball en honor a su creador Ralph Kimball e igualmente mencionada Metodología de Inmon dada a su creador William H. Inmon. Bill Inmon y Ralph Kimball son dos de las personalidades referentes y más influyentes en el área de data warehousing, y responsables de los dos enfoques a los que se hace referencia. En la web, hay muchos debates relativos Inmon vs Kimball. Ambos autores tienen aficionados que parecen creer que la elección entre los dos enfoques para un proyecto de almacén de datos es mucho más una guerra religiosa que una decisión

técnica. Inmon es el creador del término Data Warehouse así como del CIF (*Corporate Information Factory*), conjuntamente con Claudia Imhoff. Es considerado por todos el padre de la disciplina. Por su parte, Ralph Kimball es un gurú del diseño de almacenes de datos y creador del enfoque MD (*Multidimensional Architecture*).

La principal diferencia que existe entre ambas tendencias está basada en la forma de enfrentar el problema. La visión de Inmon se basa principalmente en un enfoque descendente (top-dpwn) planteando la creación de un repositorio de datos corporativo como fuente de información consolidada, persistente, histórica y de calidad. Al ser construido descendentemente los MD se nutren del DW corporativo convirtiéndose en un complejo empresarial de bases de datos relacionales.

Inmon afirma que la creación de una base de datos relacional con una leve normalización es la que nutre los mercados de datos. Por lo que no se crea los MD directamente desde el sistema OLTP a través de un área de ensayo. En lugar de ello, se crean a partir de la arquitectura relacional de los datos corporativos.

La propuesta de Kimball de dividir el mundo de Inteligencia de Negocio (*Business Intelligence*, BI) entre el hecho y las dimensiones es muy eficaz y conduce a una solución completa en una cantidad muy pequeña de tiempo. Además, la técnica de Kimball tiene una gran cantidad de documentación y se puede encontrar una respuesta a casi todas las preguntas que se posean.

Entre sus características principales es que su arquitectura es ascendente (bottom-up) debido a que plantea que se debe crear por cada departamento un conjunto de mercados de datos independientes orientados a los temas que estén relacionados con él. Y "*El Almacén de Datos es la unión de todos los Mercados de Datos de una entidad*". (CIF vs MD Dos enfoques clásicos en el diseño de la arquitectura de un Data Warehouse, 2008)

La Tabla 3 muestra las principales diferencias existentes entre ambas tendencias. (CIF vs MD Dos enfoques clásicos en el diseño de la arquitectura de un Data Warehouse, 2008)

Tabla 3 Comparación Arquitectura CIF y MD

CIF	MD
Enfoque Top-Down	Enfoque Bottom-Up
Basado en un Almacén de Datos	Basado en Mercados de Datos
Fases: Getting Data In & Getting Information Out	Fases: Back Room & Front Room
MD basados en el DW	MD a partir de la Staging Area u otros MD
Las herramientas de consulta pueden atacar el DW (no recomendado) o los MD	Las herramientas de consulta atacan los MD de la back room
Diseñado para cruzar la información de toda la organización a través el proceso de conformación	Posibles problemas en el momento de cruzar la información entre MD independientes
Información Corporativa en un mismo repositorio	Información Corporativa repartida en varios repositorios
Focalizado en la visión corporativa	Focalizado por la visión de las unidades de negocio
Diseño teniendo en cuenta las necesidades de todos los usuarios de la organización y una puesta en común para tener una visión única	Diseño teniendo en cuenta las necesidades de los usuarios de cada departamento de forma independiente
El Almacén de Datos es normalmente, una estructura relacional	Basado exclusivamente en estructuras dimensionales

Existen situaciones en las que una de las arquitecturas clásicas proporciona ventajas competitivas sobre la otra. Hecho que hará escoger dicha arquitectura. Normalmente la realidad es que ambas arquitecturas se combinan para proporcionar la mejor respuesta a las necesidades del cliente o incluso es factible encontrarse otros enfoques más o menos afortunados. Lo importante, en definitiva, es conocer todos los enfoques posibles para no tener que reinventarse una propia en el momento de hacer un diseño de un almacén de datos. Es decir, sólo a partir del conocimiento profundo se puede ir más allá de las propias fronteras.

Basados en estas propuestas se han desarrollado un conjunto de metodologías que no siguen obligatoriamente una específica sino que realizan una selección de lo mejor de cada una y definen su propia metodología. Ejemplo de esto se tiene la avalada por Microsoft llamada Metodología SQLBI orientada totalmente a las herramientas que proponen el gigante del software como son Microsoft SQL

Server, SQL Server Analysis Services y su oferta más completa en este campo que es Microsoft Suite for Business Intelligence.

La metodología DM2 se basa en las necesidades de información a nivel gerencial, donde la información debe ser encarada como patrimonio de la empresa, accesible a quien la necesite. Por la propia naturaleza del ambiente, el modelo cumple con su objetivo (atender las necesidades de información del nivel gerencial y ejecutivo de una empresa), esta metodología se asemeja a la forma top-down que propone Inmon, y acorta en función razonable el tiempo entre el inicio del análisis y la implantación. Esta rapidez no solo es buena para el cliente sino que también es exigida y necesaria por el propio ambiente que lo rodea.

En 1996 se propuso a la comunidad interesada en el desarrollo de DW una metodología llamada CRISP-DM como herramienta industrial y de aplicación neutral. Está descrita en términos de un modelo de proceso jerárquico, consistente en un conjunto de tareas definidas en cuatro niveles de abstracción (de lo general a lo específico): fase, tarea genérica, tarea especializada e instancia de procesos. Principalmente ha sido propulsada por el GIS (Grupo Interesado Especialmente en CRISP-DM, por sus siglas en inglés)

Otra metodología para este tema es la denominada HEFESTO que entre sus principales directrices plantea que la construcción e implementación de un DW puede adaptarse muy bien a cualquier ciclo de vida de desarrollo de software, con la salvedad de que para algunas fases en particular, las acciones que se han de realizar serán muy diferentes. Lo que se debe tener muy en cuenta, es no entrar en la utilización de metodologías que requieran fases extensas de reunión de requerimientos y análisis, fases de desarrollo monolítico que conlleve demasiado tiempo y fases de despliegue muy largas. Lo que se busca, es entregar una primera implementación que satisfaga una parte de las necesidades, para demostrar las ventajas del DW y motivar a los usuarios. (Bernabeu, 2007)

Esta metodología se sustenta en la definición de 4 fases principales que son: Análisis de los requerimientos, Análisis de los OLTP, elaboración del modelo lógico de la estructura del DW y procesos ETL, limpieza de datos y sentencias SQL. Dentro de cada uno de estas fases se definen un conjunto de pasos que dan cumplimiento, en su conjunto, al desarrollo del depósito.

Como se evidencia existen diversas metodologías que pretenden dar un acercamiento a una propuesta ideal para el desarrollo de almacenes. Cada autor la orienta a la optimización del rendimiento y a su visión de los principales procesos que se deben tener en cuenta para construir un almacén de datos flexible y dinámico. En dependencia de la problemática presentadas es la política de selección por una y otra metodología. Queda a decisión del equipo de desarrollo la selección de la mejor opción.

1.8.1 Justificación de la metodología a utilizar

La Oficina Nacional de Estadísticas basándose en su papel como órgano rector en materia estadística en Cuba amerita la utilización de una metodología robusta y madura que garantice el éxito de la integración de la información que actualmente disponen. De todo el conjunto de metodologías existentes para enfrentar el desarrollo del MD la decisión es adecuar, la mundialmente conocida, metodología de Kimball, adaptándola, claro está, a la realidad de la Universidad de las Ciencias Informáticas (UCI), por las siguientes razones:

- La técnica de Kimball posee una gran cantidad de documentación y generalmente se puede encontrar una respuesta a casi todas las problemáticas que puedan presentar.
- Su creador Ralph Kimball es una figura emblemática en el mundo de warehousing teniendo publicados alrededor de 100 artículos científicos proponiendo mejoras al proceso, además de innumerables libros que se han posicionado como guías de obligatoria consulta para el desarrollo, ejemplo de esto es su libro Técnicas de Diseño Dimensional que en la actualidad se ha convertido en un “Best Seller” dentro del campo.
- Claridad de las actividades a realizar por cada rol propuesto.
- Esta metodología de dividir el mundo de BI entre el hecho y las dimensiones es muy eficaz y conduce a una solución completa en un tiempo razonable.
- Es iterativo, donde se construye una pieza a la vez (mercado de datos) garantizando mayor velocidad de respuesta a los clientes.

- La forma de almacenar la información es de fácil entendimiento por parte del usuario lo que permite mayor comprensión para el análisis de los datos que se encuentran integrados.
- Es una metodología resistente y adaptable ante los cambios.

1.9 Herramientas Existentes

En la actualidad se han desarrollado diversas herramientas con el fin de dar un acercamiento a la automatización del diseño, construcción, implementación y mantenimiento de los DW. Existen compañías que han marcado hitos en este sentido, un ejemplo de ellas es el gigante Oracle, que en la actualidad es el número uno mundial en el desarrollo de aplicaciones de este tipo, tanto para la optimización y mantenimiento de las bases de datos en función de las mejoras de las consultas complejas y reportes, como en el desarrollo de cuadros de mandos integrales y sistemas estratégicos de negocios empresariales enfocados a la toma de decisiones a nivel empresarial.

Una de las principales ventajas que presenta esta compañía es su arquitectura multiplataforma en el desarrollo de sus aplicaciones. Con su producto Oracle Warehouse Builder 10g puso al mercado una poderosa herramienta que soporta todo el ciclo de vida del desarrollo de almacenes de datos desde su diseño inicial hasta su implantación y mantenimiento. Con la aparición de la segunda versión de esta herramienta se mejoró importantes funciones como administración, integración y calidad de los datos además de una interfaz de usuario más amigable. Estas nuevas características facilitan su uso y la proponen como una herramienta eficaz para el desarrollo de Sistemas Data Warehouses y de BI.

También permite la realización de auditorías de los datos, el modelado relacional y dimensional, la gestión de datos y metadatos, opción avanzada de carga de datos, soporte de dimensiones lentamente cambiantes, traza lineal de principio a fin. Incluye el diseño de estructuras OLAP y relacionales permitiendo la integración entre ambas funciones. La total integración con el servidor ORACLE le ofrece una solidez incalculable debido a que dentro de los servidores de base de datos actualmente es el número en prestaciones.

La posibilidad de mantener un entorno integrado y el almacenamiento centralizado el servidor Oracle brinda a los usuarios la ventaja de poder utilizar la herramienta Oracle Business Intelligence Suite,

actualmente en su versión mejorada Oracle Business Intelligence Enterprise Edition Plus, para la realización de estudios de mercados e Inteligencia de Negocio.

Entre sus opciones permiten mejorar el desarrollo y escalabilidad de los procesos de ETL, los cuales son el 70 porciento, aproximadamente, en el desarrollo de los almacenes de datos, y administración de metadatos. Sus conexiones permiten a los clientes extraer datos rápida y fácilmente, y en algunos casos, dirigir los datos hacia sus aplicaciones centrales de CRM y ERP, incluidas Oracle E-Business Suite y PeopleSoft Enterprise. Incluye soporte para apuntar a bases de datos que no pertenecen a Oracle, característica que permite a los usuarios elegir el lugar donde sus datos serán almacenados finalmente.

La funcionalidad “Experts” permite el ahorro de costo y tiempo para los usuarios. Encapsula mediante wizard los estándares de desarrollo de la organización además que una amplia gama de usuarios, incluidos los usuarios finales, accedan a la funcionalidad declarativa de Oracle Warehouse Builder.

Garantizar la alta calidad de los datos es clave para las empresas que buscan optimizar el desempeño comercial y garantizar una mejor toma de decisiones. Para abordar este requerimiento, Oracle Warehouse Builder 10g ofrece mejores funciones diseñadas para mejorar la calidad de datos, incluidas la depuración de direcciones y nombres, y la funcionalidad de asociación-fusión (duplicación).

Otro gigante del desarrollo de software, Microsoft Corporation, no se ha quedado atrás en este sentido y dispuso en el mercado la herramienta Microsoft SQL Server 2000 Analysis Services, también disponible en su versión 2005, sucesores directos del paquete de componente OLAP Services que incluye el procesamiento analítico en línea (OLAP) y el tratamiento de minería de datos sobre los cubos OLAP para la extracción de conocimiento de la información almacenada ya sea relacional o dimensional.

Microsoft SQL Server 2005 Analysis Services (SSAS) ofrece funciones de procesamiento analítico en línea (OLAP) y minería de datos para aplicaciones de Business Intelligence. SSAS admite OLAP y permite diseñar, crear y administrar estructuras multidimensionales que contienen datos agregados desde otros orígenes de datos, como bases de datos relacionales. En el caso de las aplicaciones de minería de datos, permite diseñar, crear y visualizar modelos que se construyen a partir de otros orígenes de datos mediante el uso de una gran variedad de algoritmos estándares del sector.

Entre sus principales mejorías está la creación de las dimensiones con nuevos tipos y características. Permitiendo la asociación jerárquica y por niveles para la organización de los datos. También esta herramienta incluye características que proveen de mayor flexibilidad y control de acceso al cubo de datos, con métodos adicionales de autentificación de usuarios y roles.

En el plano Open Source la herramienta más significativa es Mondrian, la cual es una de las aplicaciones más importantes de la Suite Pentaho BI. Mondrian es un servidor OLAP open source que gestiona la comunicación entre una aplicación OLAP y la base de datos con los datos fuente. Es desarrollado en Java/Servlets/JSPs que permite ser instalado en servidores de aplicaciones como JBoss. Entre sus principales características se encuentra la facilidad para el análisis de grandes volúmenes de información que se encuentren almacenados en bases de datos que soporten JDBC.

Mondrian soporta el lenguaje Microsoft's Multidimensional Expressions (MDX). También soporta los APIs: Java OLAP (JOLAP) y XML for Analysis application programming. Sin embargo existen otras herramientas Open Source, orientadas a publicar reportes interactivos y gráficos mediante interfaces Web ejemplo BIRT, perteneciente al grupo Eclipse, iReports (de Jasper), Pentaho Reporting (de Pentaho), OpenReports, entre otros. También existen otras herramientas homólogas a Mondrian como son: Jedox, FreeAnalysis, JRubik con funcionalidades similares.

Como es conocido, el proceso de ETL, es el más importante y complicado dentro del desarrollo del data warehouse, para esto existen propuestas como: Octopus, Xineo, CloverETL, BabelDoc, Joost , CB2XML, mec-eagle, Transmorpher, XPipe, DataSift, Xephyrus Flume, Smallx, Nux, KETL, Kettle, OpenDigger, ServingXML, Talend, Scriptella, ETL Integrator, Jitterbit, Apatar, Spring Batch, JasperETL, Pentaho Data Integration.

En general no existen grandes diferencias entre las propuestas open source y las propietarias. Sólo diferenciándose en la completitud de las mismas, es decir, las privativas generalmente poseen todas las funcionalidades integradas y en el caso de las libres se obtienen mediante herramientas aisladas. Pero las experiencias definidas en proyectos de Almacenamiento de Datos e Inteligencia de Negocio es que si existe capacidad de integración entre las herramientas es permisible.

1.9.1 Justificación de las herramientas a utilizar

La Oficina Nacional de Estadísticas es una de las entidades que se encuentran en el país en franca migración hacia la independencia tecnológica. Actualmente está exportando todos sus dispositivos de almacenamiento hacia plataforma PostgreSQL por lo que queda seleccionado como Sistema Gestor de Base de Datos el PostgreSQL. Esta decisión ha sido previamente colegiada y aceptada por parte del cliente final debido a que dentro de sus políticas de migración se encuentran las de llevar a todas sus bases de datos hacia dicha plataforma. En este sentido la versión que se utilizará es la 8.3.7 por ser lo suficientemente estable y segura. Entre las principales características que avalan esta decisión se encuentran:

- La cantidad máxima de BD que permite es ilimitada.
- El tamaño máximo de las tablas es de 32 Tb.
- El tamaño máximo de registro es de 1.6 Tb.
- El máximo de tamaño del campo es de 1 Gb.
- El máximo de registros por tablas es ilimitado.
- El máximo de campos por tabla es 250 a 1600 en dependencia de los tipos de datos usados.
- El máximo de índices por tablas es ilimitado.
- Se permite programar funciones en lenguajes como: Pl/pgsql, Pl/java, Pl/perl, Pl/pyton, TCL, pl/PHP, C, C++, Ruby, entre otros.
- Alta capacidad de almacenar información, con una alta velocidad de respuesta ante consultas complejas y/o extensas.

La Suite de Pentaho está compuesta por 5 componentes principales que se especializan en los procesos definidos dentro del ciclo de vida del desarrollo de Almacenes de Datos. En este sentido no todos los componentes pertenecen a la categoría de software libre razón por la cual de las 5 solamente se va a

utilizar uno sólo dentro del alcance de la presente investigación debido a que la capa de Inteligencia de Negocio no se encuentra dentro del mismo.

Pentaho Data Integration v 3.1.0 (PDI):

- Es de formato abierto y de fácil lectura para los xml que recogen transformaciones, tareas programadas y un repositorio relacional de metadatos ETL.
- Es aplicable a diversos tipos de bases de datos (SQL server, PostgreSQL, MySQL, Microsoft Access, etc.).
- Posee facilidad para la importación y exportación de datos de un formato a otro cualquiera.
- Su principal fortaleza es la posibilidad que brinda de ser extensible mediante pluggins.

Otros aspectos a tener en cuenta de la selección:

- Alta compatibilidad con la herramienta ETL con gestor PostgreSQL en los llamados Paquetes, donde se ofrecen servicios de importación, exportación, transporte, y transformación de datos.
- Rendimiento eficiente: Alta velocidad de respuesta de las consultas. El servidor PostgreSQL posee un potente motor de recuperación de los datos y permite la optimización de las complejas consultas enviadas o preparadas desde el PDI.
- Capacidad del PDI para extraer y cargar datos utilizando el PostgreSQL.
- Con estas herramientas se tiene la posibilidad de no sólo almacenar los datos al más atómico de los detalles, sino que también se puedan guardar solamente los agregados necesarios.

Conclusiones del Capítulo

A partir del estudio del estado del arte realizado se concluyó lo siguiente:

- La tecnología apropiada para la problemática en cuestión es el Mercado de Datos.
- La metodología de desarrollo adoptada es la de Kimball.

- El diseño dimensional cumple con los requerimientos necesarios para la estructuración del Mercado de Datos.
- Las herramientas a utilizar son el Sistema Gestor de Base de Datos PostgreSQL en su versión 8.3.7 y el Pentaho Data Integration 3.1.0.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN

Introducción

El desarrollo de Mercados de Datos no es una tarea fácil ni sencilla debido a que está compuesta por un conjunto de componentes que interactúan entre sí, como un sistema, para lograr un diseño robusto y adaptable a las necesidades reales de los usuarios finales. Cada componente posee sus responsabilidades específicas y trabajan de forma semi-independiente aportando resultados tangibles dentro de la solución física.

Estos componentes están enmarcados dentro de 3 grandes grupos:

- Diseño dimensional de las estructuras.
- Extracción, Transformación y Carga (ETL, *por sus siglas en inglés*) de los datos de la(s) fuente(s).
- Inteligencia de Negocio

La definición del proceso del negocio a modelar, la identificación de granularidad, dimensiones, medidas, cubos de datos y estructuras dimensionales son hitos decisivos para realizar un efectivo proceso de ETL donde se realizan desde simples transformaciones, como cambio de formato de campos, hasta complejas búsquedas de datos incompletos o insignificantes, todo esto, para garantizar la calidad en los datos a integrar.

2.1 Descripción de las Fuentes de Datos

Las Fuentes de Datos son el punto de partida dentro la arquitectura general de los Almacenes de Datos. Estas están agrupadas en cuatro categorías principales: Datos de Producción, Datos Internos, Datos Archivados y Datos Externos. (Ponniah, 2001)

Datos de Producción:

Son los datos de interés para el DW que se encuentran almacenados en los diferentes sistemas operacionales y que son utilizados dentro de la organización en sus funciones diarias.

Datos Internos

Son los datos que cada departamento, dentro de la organización, poseen almacenados en archivos o bases de datos internas para auxiliarse en sus actividades. Esta información es generalmente útil para el DW.

Datos Archivados

Son los datos provenientes de sistemas operacionales que se almacenan con el objetivo de llevar un histórico de la información de la organización.

Datos Externos

Son datos que provienen de fuentes externas a la organización. Generalmente son informaciones compartidas entre competidores o entre proveedores y clientes.

Las fuentes identificadas en este caso están divididas en dos grupos: La información histórica, que pertenecen a la categoría de Datos Archivados, y el conjunto de clasificadores establecidos por la ONE para la clasificación de los Centros Informantes que se subordinan a la estructura nacional, que se enmarca en la categoría de Datos Internos.

2.1.1 Fuente 1: Información Histórica 2000-2008

Esta fuente está conformada por un conjunto de archivos, en formato DBF, que posee la Oficina Nacional de Estadísticas para la confección de los diferentes análisis y reportes que se les solicitan. El proceso para la creación de los mismos nace en un servidor FTP (Protocolo de Transferencia de Archivos, por sus siglas en inglés) donde cada provincia posee una carpeta para que coloquen la información captada durante el mes, de los modelos estadísticos definidos. A partir de la recogida de esta información de todas las provincias se comienza con el proceso de limpieza de los datos recibidos donde un especialista de la ONE recoge toda la información y elabora un archivo general con los datos exactamente iguales a como viene de las provincias.

Cuando está listo el archivo es revisado por la dirección de Estadísticas Sociales que mediante cálculos matemáticos y reportes tipos que ya tienen predefinidos realizan un documento con los cambios a realizar sobre el archivo original debido a problemas o deficiencias encontrados durante el análisis realizado. La especialista de la ONE a cargo realiza los cambios y da por terminado el proceso de creación del mismo.

Estos archivos poseen una estructura similar donde se almacena toda la información suministrada por las provincias, organizada por mes y año de la información. Dentro de su estructura se relacionan los números de las filas del modelo, los códigos de los clasificadores asociados y los valores de la información captada de todo el país. Ver Figura 6 donde se muestra una porción de la información almacenada perteneciente al mes de Octubre del año 2008 del Modelo de Indicadores Generales.

MOD	EMP	V01	V02	ORG	PRO	DPA	SEC	RAM	CAE	UNI	FF1	SUB	EP	SIN	ESF	FPR	FOR	NAE	FF2	EAT	FIL	C01	C02	C03
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00000100	41350,0	50491,1	41147,7
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00000400	41350,0	50491,1	41147,7
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00000410	37200,0	47067,0	36036,6
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00001700	150,0	129,0	118,0
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00001800	0,0	126,0	129,0
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00001900	0,0	82,0	82,0
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00002100	1027,0	921,5	837,8
000508	00259	0	00	259	03	0302	14	1400	140000	999	1	0	01	01	2	1	4	116519	1	3	00002300	22,0	18,2	15,6
000508	00315	0	00	311	01	0105	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00000100	78,0	78,3	54,4
000508	00315	0	00	311	01	0105	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00000400	78,0	78,3	54,4
000508	00315	0	00	311	01	0105	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001400	78,0	78,3	54,4
000508	00315	0	00	311	01	0105	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001700	607,0	607,0	531,0
000508	00315	0	00	311	01	0105	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001800	0,0	628,0	592,0
000508	00315	0	00	311	01	0105	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001900	0,0	227,0	196,0
000508	00315	0	00	311	01	0105	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00002100	1821,0	1821,1	1693,2
000508	00319	0	00	311	01	0109	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00000100	16,7	34,1	36,6
000508	00319	0	00	311	01	0109	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001400	16,7	34,1	36,6
000508	00319	0	00	311	01	0109	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001700	379,0	385,0	347,0
000508	00319	0	00	311	01	0109	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001800	0,0	385,0	348,0
000508	00319	0	00	311	01	0109	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00001900	0,0	180,0	172,0
000508	00319	0	00	311	01	0109	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00002100	1250,3	1250,3	1080,5
000508	00322	0	00	311	01	0111	15	1501	150103	999	2	2	00	01	2	1	5	137511	2	3	00000100	97,5	97,5	78,1

Figura 6 Estructura de un archivo de información

2.1.2 Fuente 2: Clasificadores Estadísticos

Los Clasificadores son elementos que identifican la procedencia de la información que se ha captado. Son diferentes tipos de agrupación que organizan a los Centros Informantes que se encuentran registrados bajo las directrices de la ONE.

La fuente está conformada por un conjunto de archivos DBF que almacenan, en forma de tablas, los valores que pueden tomar los diferentes clasificadores.

Relación de Clasificadores:

- Clasificador Actividad Económica (CAE).
- Nomenclador Actividad Económica (NAE).
- Codificador de la División Político-Administrativa (DPA).
- Codificador de Organismo.
- Clasificador de Empresa.
- Clasificador de Forma Financiamiento.
- Clasificador Forma Organizativa.
- Clasificador Único de Indicador (CUI).
- Clasificador de Subordinación.
- Clasificador de Esfera Económica.
- Clasificador del Órgano de Atención del Gobierno.

Si bien no todos son consecuentes con las definiciones internacionales, se formulan utilizando como base las mismas y adaptándolas a las particularidades del país.

2.2 Definición de las Áreas de Análisis

La definición de las Áreas de Análisis (AA) son de vital importancia para del desarrollo del MD. La realización del mismo enfoca el desarrollo hacia el buen cumplimiento de las metas trazadas y garantiza la factibilidad, utilidad y el éxito de las estructuras que se están diseñando. En la solución propuesta se orientan en función de los diferentes cortes a la información que comúnmente la ONE realiza. En este sentido se definió un AA que está en concordancia con las necesidades de información identificadas:

1. Publicación de Indicadores sobre Producción e Ingresos
2. Publicación de indicadores sobre Exportación e Importación de Servicios.
3. Publicación de indicadores sobre Fuerza de Trabajo y Salarios.

El Área de Análisis identificado se denominó:

1. Comportamiento de los Indicadores Generales.

2.3 Arquitectura de los Componentes del Sistema

De manera general, la arquitectura, dentro del desarrollo de software, es el diseño de más alto nivel de la estructura de un sistema o producto basado en reglas, objetivos y restricciones. Más específicamente, en la tecnología warehousing, es una forma de representar la estructura total de datos, comunicación, procesamiento y presentación, en función de los usuarios finales.

Ponniah la define como la estructura que unifica los componentes del DW, donde provee un marco general para su desarrollo y despliegue. Además plantea que define los estándares, mediciones, diseño general y técnicas de soporte. (Ponniah, 2001)

En la Figura 7 se presenta la arquitectura de la solución que va a tener el Mercado de Datos.

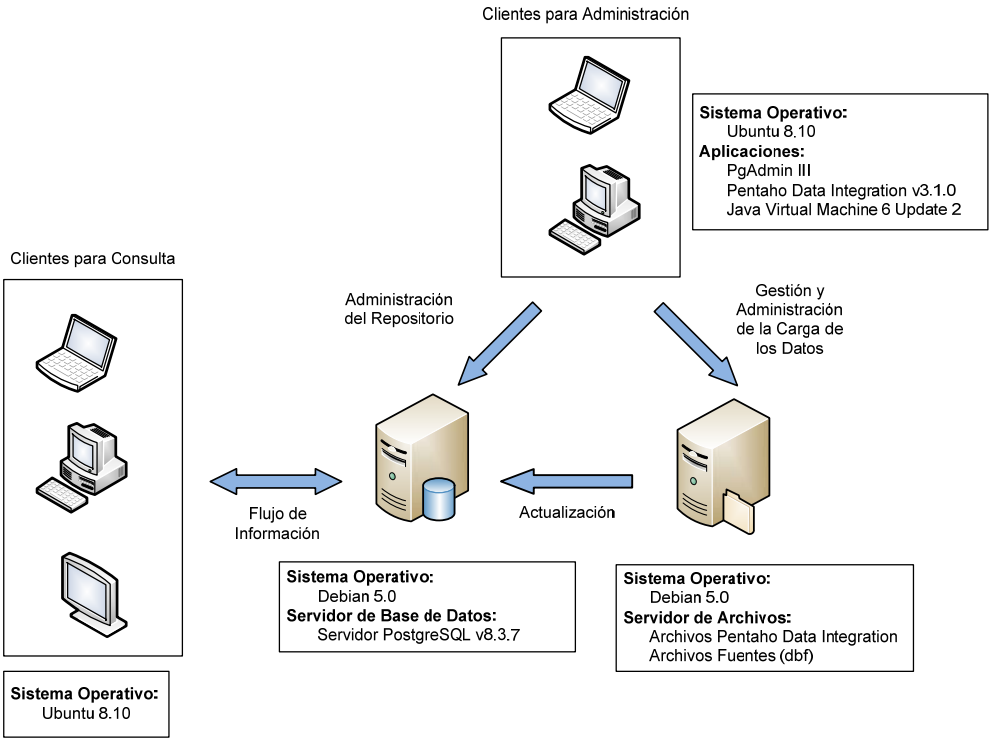

Figura 7 Arquitectura de la Solución

2.3.1 Arquitectura de la Solución

Para describir la arquitectura se va a dividir en 3 secciones: el componente de presentación (front end, en inglés), el repositorio central y el de carga de datos. En cada una de las secciones existen un conjunto de herramientas que soportan el proceso.

En primer lugar el componente más importante y es sobre el cual se basa el sistema es el repositorio central, la estructura del mismo está compuesta por el Gestor de Base de Datos PostgreSQL 8.3.7 que es donde va a estar desplegado el sistema sobre el Sistema Operativo Debian 5.0. Al igual que el servidor de ficheros que almacenará las estructuras diseñadas en el Pentaho Data Integration v3.1.0 y las fuentes de los DBF que se deseen integrar al MD permitiendo además su sincronización con los datos históricos. Las

estaciones para consulta y administración se soportarán sobre Sistema Operativo Ubuntu 8.10 como elemento de migración.

Las estructuras diseñadas están definidas en dos niveles de agrupación y concentrada en una misma instancia de la base de datos. Por un lado están las estructuras con los datos detallados relacionados con las dimensiones propuestas y en un segundo nivel se encuentran las agregaciones diseñadas en función de los reportes más comunes. Vale destacar que la carga de los datos hacia estas estructuras se realizará mediantes funciones definidas dentro del mismo gestor y sólo estará almacenada la información referente a los últimos 2 años. En la Figura 8 se especifica la arquitectura interna dentro del repositorio.

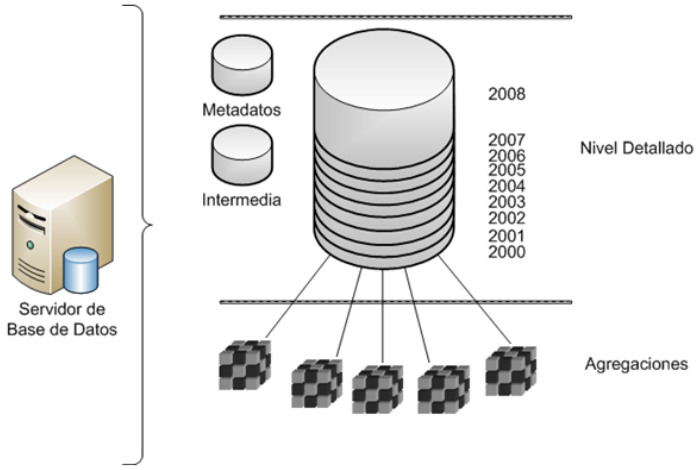

Figura 8 Arquitectura interna de la BD

Además de esta base de datos se encuentran dos más dentro del servidor, una llamada INTERMEDIA que es la encargada de almacenar la información de los procesos de ETL y donde aparecerán las estructuras necesarias para realizar la carga de los datos hacia el Mercado de Datos y otra llamada METADATOS donde se guardarán todos los metadatos de las estructuras dimensionales, esta base de datos es gestionada por la herramienta utilizada para los procesos de ETL.

Por último y no menos importante se encuentra el componente de administración y carga de datos. Para el cual se propone la utilización de la herramienta Pentaho Data Integration para todo el proceso de extracción, transformación y carga de los datos desde las fuentes y el PgAdmin III para la administración y

mantenimiento del servidor de base de datos. Además las rutinas de ETL se encontrarán diseñadas para ser utilizadas cada vez que se desee adicionar datos al repositorio siempre y cuando esos datos se encuentren en un formato similar al utilizado para la carga del histórico.

2.4 Pasos para el diseño lógico de la solución

Como se especificó en el Capítulo 1, la metodología que se utilizará para el diseño e implementación es la propuesta por Ralph Kimball. En la bibliografía relacionada con el diseño de estructuras dimensionales perteneciente a él se propone una guía bien explícita para su diseño lógico. Específicamente se agrupa en cuatro pasos fundamentales. (Kimball, y otros, 2002)

1. **Seleccionar el proceso a modelar.**

Es una actividad que brinda como salida la identificación de los procesos del negocio, los cuales se priorizan en función de las necesidades informacionales identificadas durante las entrevistas realizadas a los clientes.

2. **Declarar el grano del proceso del negocio.**

El grano significa específicamente la representación individual de las tablas de hechos. Existe una pregunta que ayuda a la definición del grano: ¿Cómo se puede describir una simple fila en la Tabla de Hechos?

3. **Seleccionar las dimensiones aplicables en cada tabla de hecho.**

Es la definición de las dimensiones propuestas. Igualmente existe una pregunta que auxilia en este sentido: ¿Cómo las personas del negocio describen los datos que resultan del proceso del negocio?

4. **Identificar el hecho numérico que puede poblar cada fila de la tabla de hechos.**

Es la identificación del valor numérico que se va a registrar en el Mercado de Datos. En este caso le pregunta es: ¿Cuáles son los valores medibles a almacenar?

Para la implementación de los MD es necesario utilizar una matriz para representar la relación entre las dimensiones y los procesos del negocio. Destacando que los Mercados de Datos se realizan en base a las fuentes no a los departamentos existentes en la organización.

2.5 Diseño del Sistema

El diseño de la solución está enfocado a dos niveles de detalle, un nivel la información más detallada y otro orientado a un nivel menos atómico pero más cercano a la información que comúnmente reportan, a esto se le denomina agregaciones. Para lograr el mismo se siguieron los pasos propuestos por la metodología.

2.5.1 Proceso del Negocio a modelar

La Oficina Nacional de Estadísticas debido al papel que juega como ente rector y coordinador de los temas estadísticos en Cuba posee el objetivo de funcionar como repositorio central donde confluyen y vierten un conjunto de procesos que ejecutan y supervisan la gestión estadística del país. En este sentido el mecanismo de captación que se encuentra en vigor está compuesto por diferentes procesos orientados a los distintos tipos de modelos estadísticos existentes.

Los modelos de Estadísticas Continuas y Encuestas Periódicas son los principales afluentes de la ramificación de procesos definidos. Las diferencias entre unos y otros se basan principalmente en los periodos de captura (mensual, trimestral, semestral, anual, etc.), características específicas de la recogida de información, estructuras de plantillas, entre otras. Cada uno se ajusta a los indicadores relacionados a cada centro informante en dependencia de la labor que realice, ya sea, económica, sociales, organismos estatales, etc.

El dato estadístico es sustraído de cada centro informante en los modelos anteriormente mencionados, subordinados a diferentes niveles de la estructura nacional que posee la ONE en una fecha determinada y donde cada uno de ellos representa un conjunto de indicadores.

Después de haber descrito brevemente el negocio de la ONE se resalta que su proceso principal es la captación de la información estadística, a todos los niveles y en todos los sectores, almacenándola desde

el mayor nivel de detalle hasta los consolidados más densos y complejos con el objetivo de permitir la disponibilidad de la misma para su consulta con la rapidez y validez requerida.

2.5.2 Grano Identificado

Las mejores prácticas sugieren que se desarrollen los modelos dimensionales sobre la información más detallada, capturada por el proceso del negocio, denominándose grano del proceso a modelar. La definición del grano es la que determina el alcance dimensional de las estructuras impactando significativamente en el tamaño del sistema en desarrollo.

En concordancia con las necesidades del negocio el grano queda definido como la información estadística mensual perteneciente a todos los Centros Informantes, registrados bajo la estructura de Organismos, Subordinación, Forma Organizativa, Forma de Financiamiento, Esfera, Clasificador de Actividades Económicas, Nomenclador de Actividades Económicas y Órgano de Atención del Gobierno, en todos los municipios, de los Indicadores Estadísticos captados en el Modelo Indicadores Generales.

2.5.3 Dimensiones Identificadas

Después de haber declarado el grano del proceso a modelar se comienza la definición de las dimensiones candidatas que posteriormente, después de un profundo análisis, se convertirán en las dimensiones que contendrá la solución. Las dimensiones poseen entre sus características principales la definición de jerarquías entre sus atributos las que poseen como objetivo plasmar explícitamente la forma en que se puede consolidar, ya sea mediante el uso de sumas, porcientos, máximos, mínimos, etc; la realización del proceso de análisis en línea de la información.

A continuación se describen las dimensiones y jerarquías que están relacionadas con el repositorio principal donde se va a almacenar la información atómicamente. En el Anexo 1 Especificación de las Dimensiones, se describen cada uno de los campos propuestos dentro de las dimensiones.

Dimensión NAE

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo al Nomenclador de Actividades Económicas.

Jerarquía:

1. Sector -> Rama ->NAE

Dimensión EAT

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo al Órgano de Atención del Gobierno.

Jerarquía:

1. EAT

Dimensión EMPRESA

Esta dimensión describe el universo de Empresas existentes.

Jerarquía:

1. Empresa

Dimensión ESFERA

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo a la esfera bajo la cual se subordina el Centro Informante que suministra la información.

Jerarquía:

1. Esfera

Dimensión FORMA DE FINANCIAMIENTO

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo a la Forma de Financiamiento que posea el Centro Informante que suministra la información.

Jerarquía:

1. Forma de Financiamiento 2 -> Forma de Financiamiento 1

Dimensión FORMA ORGANIZATIVA

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo a la Forma Organizativa que posea el Centro Informante que suministra la información.

Jerarquía:

1. Forma Organizativa

Dimensión TEMPORAL

Esta es la dimensión más común e importante en los diseños de Mercados de Datos debido a que define una línea de tiempo para enmarcar la información almacenada. Además organiza jerárquicamente cuando fue captada la información.

Jerarquías:

1. Año -> Semestre -> Trimestre -> Mes
2. Año -> Mes

Dimensión CAE

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo al Clasificador de Actividades Económicas.

Jerarquía:

1. Sector -> Rama ->CAE

Dimensión DPA

Esta dimensión almacena los valores pertenecientes a la División Política Administrativa que presenta el país.

Jerarquía:

1. Provincia -> Municipio

Dimensión ORGANISMO

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo al Organismo al cual pertenece el Centro Informante que suministra la información.

Jerarquía:

1. Organismo

Dimensión SUBORDINACIÓN

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo a la estructura de Subordinación que presenta el Centro Informante que suministra la información.

Jerarquía:

1. Subordinación

Dimensión MODELO

Esta dimensión describe el universo de valores bajo los cuales puede clasificarse la información atendiendo a los modelos establecidos para captar la información. En esta dimensión hay que especificar que dentro del marco de la presente investigación no se relaciona con más de una tabla de hechos debido a que solamente integra un sólo modelo estadístico pero que el diseño es flexible para la integración de los demás modelos utilizados por la ONE.

Jerarquía:

1. Dirección -> Modelo

2.5.4 Tabla de Hechos Identificada

Las tablas de hechos son las que almacenan las medidas numéricas. En este caso se definieron como medidas numéricas los tres valores que se captan en el Modelo Indicadores Generales que son lo concernientes al valor del Plan y Real del año actual y al Real del año anterior debido a que es un modelo estadístico netamente económico. La tabla de hechos identificada se describe a continuación:

Tabla de Hechos INDICADORES GENERALES

En esta tabla es donde va a residir, como repositorio central, toda la información existente del modelo estadístico en cuestión. Es la tabla que servirá como fuente de información principal para la realización de las estructuras que soporten los reportes más comunes de la institución.

En la Tabla 4 se relacionan las AA identificadas con las dimensiones propuestas con el fin de presentar los Mercados de Datos candidatos del proceso en cuestión.

Tabla 4 Relación entre Áreas de Análisis y Dimensiones

AA/DIM	Subordinación	Esfera	Organismo	Indicadores	Modelo	Forma Financiamiento	Forma Organizativa	DPA	EAT	Temporal	Empresa	CAE	NAE
Indicadores Generales	X	X	X	X	X	X	X	X	X	X	X	X	X

2.6 Mercado de Datos

En la solución se ha definido un MD donde convergen todas las dimensiones propuestas, Modelo, Temporal, Forma de Financiamiento, Forma Organizativa, Organismo, CAE, NAE, DPA, Subordinación, EAT, Esfera, Indicador y Empresa; y 5 estructuras adicionales en función de los reportes más comunes

identificados. Los hechos mensurables del MD son los aspectos que se evalúan en el modelo estadístico en cuestión: el valor del Plan del Año Actual, del Real de Año Actual y del Real del Año anterior.

Algunas características que presenta la solución desarrollada es que se actualizará mensualmente mediante configuraciones realizadas a rutinas desarrolladas en PDI y el modo de almacenamiento que se utilizará es el ROLAP.

Una de las razones por la cuales se propuso como metodología de desarrollo la de Kimball es la posibilidad que brinda de ir desarrollando un proceso a la vez. En este sentido la línea que se va a seguir es que las iteraciones se hagan en función de los modelos estadísticos que la ONE posee integrándose uno a la vez. En la versión actual del sistema solamente contempla la integración del modelo estadístico 0005 o Indicadores Generales, como también se le conoce, debido a que es el modelo más estable y con mayor cantidad de Centros Informantes involucrados con él.

2.6.1 Granularidad del Proceso

La granularidad en la tabla de hechos se determina después de identificar las columnas que existirán en dichas tablas. La granularidad, como concepto, es una medida del nivel de detalle enfocada a cada ocurrencia que exista en la Tabla de Hechos. Por esta razón se puede inferir la estrecha relación existente entre las dimensiones y la granularidad.

Es recomendable no mezclar varias granularidades en una misma tabla de hechos, ni almacenar, en dicha tabla, sumas, promedios, porcientos o resúmenes, debido a que contradicen la filosofía de almacenar el mínimo detalle de la información, en estos casos se deben almacenar dichos resúmenes o agregados en tablas separadas con sus respectivos niveles de granularidad.

Además de la importancia que reviste el mantenimiento de la mínima granularidad dentro del diseño de los almacenes de datos, en ocasiones también es aconsejable almacenar la información en niveles de detalles intermedios, es decir, con altos niveles de granularidad debido a que podría ser beneficioso para empresas que no requieran altos niveles de detalles para el análisis de su información.

Después del análisis anterior se puede concluir que la granularidad del repositorio central de la solución propuesta está dada por el registro del dato estadístico captado en un mes determinado, con un indicador

asociado, por un centro informante; y este último está asociado a un municipio, un organismo, un clasificador de actividades económicas, un nomenclador de actividades económicas, una forma de financiamiento, una forma organizativa, una esfera, un órgano de atención del gobierno y una subordinación.

2.7 Modelo Dimensional

Una vez definido dentro del negocio las dimensiones, medidas, el cubo y la granularidad, se procede a la estructuración del modelo o los modelos dimensionales que existirán. En tal sentido se puede destacar que por las necesidades actuales del negocio existen varios modelos que unifican las dimensiones definidas y las medidas que se han especificado hasta el momento.

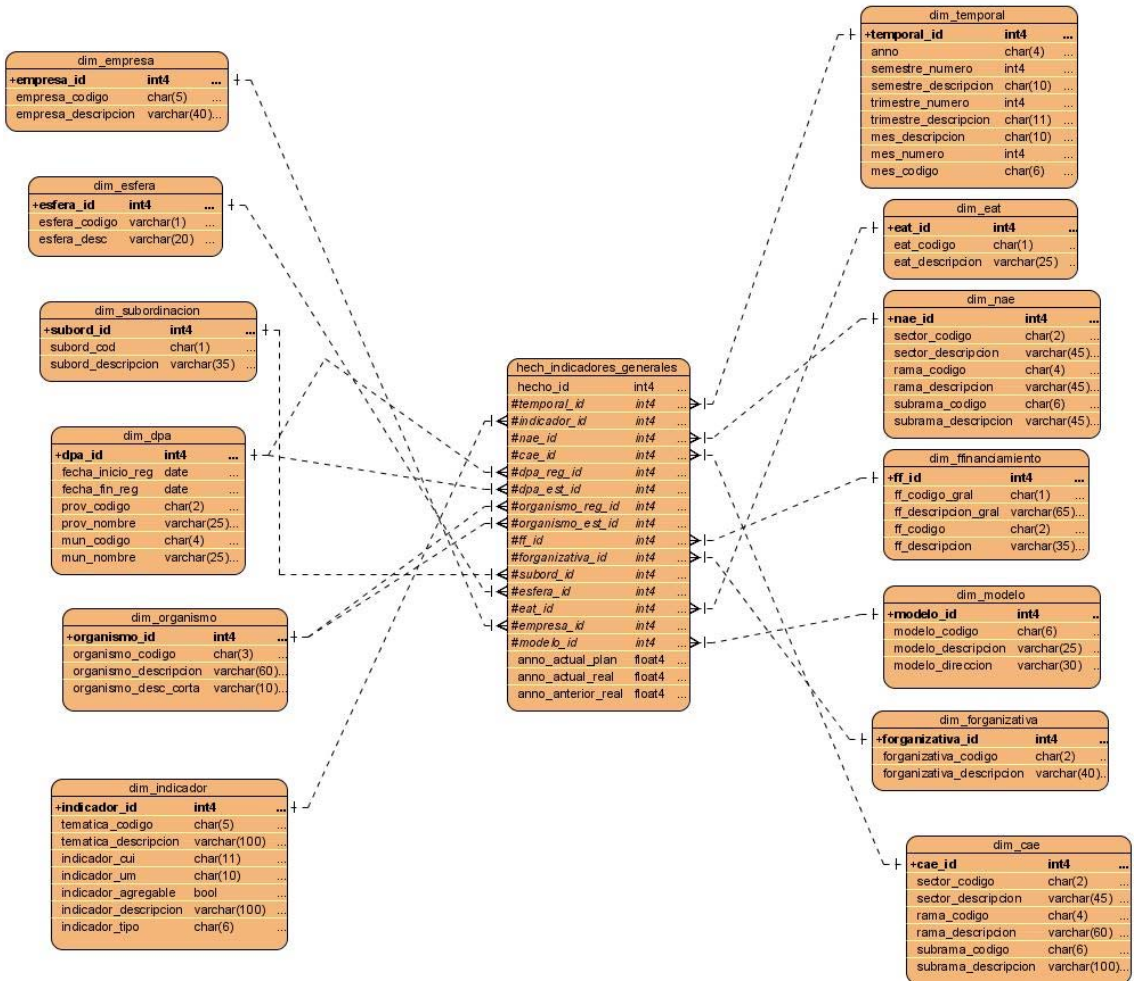

Figura 9 Modelo Dimensional de la Solución

La idea fundamental del modelo dimensional es que los datos de negocio queden representados en forma de cubo de datos. En los cubos cada celda contiene un valor y las aristas del cubo definen dimensiones naturales de análisis. En la solución propuesta se seleccionó el Modelo Tipo Estrella, Figura 9, para el desarrollo de la misma, donde existirá una tabla central llamada INDICADORES GENERALES la cual estará relacionada con las 13 dimensiones propuestas. Con este fin cada dimensión posee una llave

primaria que es la encargada de mantener la integridad referencial entre ellas y la tabla de hechos. Vale aclarar que esta llave primaria no posee ningún tipo de significado dentro del negocio, simplemente, es un número que garantiza las uniones.

2.8 Implementación del Mercado de Datos

Según la metodología que se está utilizando posterior al diseño dimensional de los MD se procede a la implementación física de los mismos. En el modelo dimensional, tanto el diseño lógico como el físico poseen una gran semejanza. El modelo físico se diferenciará del lógico en términos de los detalles específicos del gestor incluyendo tipos de datos, índices, relación entre tablas, etc. Para la implementación del modelo físico Kimball propone un conjunto de pasos que ayudan a desarrollar el mismo. (Kimball, y otros) A continuación se desarrollan cada uno de ellos.

2.8.1 Desarrollo de la BD y Estandarización de los Nombres

Esencialmente existen 3 componentes básicos de Nombres de Objetos de Bases de Datos: palabra primaria (*prime words*), clases de palabras (*class words*) y calificadores (*qualifiers*). Definiciones de elementos de datos, nombres de elementos lógicos de datos y nombres de elementos físicos de datos están todos compuestos por los tres componentes básicos.

Palabra primaria: Describe el elemento de datos de la materia. Algunos ejemplos de esta categoría son: clientes, productos, cuentas, ciudades, regiones, etc. Cada palabra primaria debe ser clara, definición inequívoca que va la derecha en la información de catálogo.

Clases de palabras: Describe la mayor clasificación de los datos asociados con cada elemento. Algunos ejemplos son: total, cantidad, fecha, bandera, nombre, descripción y número.

Clasificadores: Los clasificadores son elementos opcionales que pueden describir o definir más las Clases o Palabras Primarias. Existen algunos ejemplos como: inicio, final, primario, secundario, etc.

El objetivo principal de este paso es organizar la forma en que se van a denominar las estructuras con el fin de que quede documentado para su utilización por los inmersos en la arquitectura de desarrollo explicada en el Capítulo 1. Generalmente queda estructurado de la siguiente manera:

Primaria_Clasificador_Clase ejemplo *cuenta_inicio_fecha.* Además es conveniente mantener una nomenclatura estándar en el nombrado para un mejor entendimiento de las estructuras por los desarrolladores.

En la solución propuesta, a nivel global, se mantuvo la misma estructura en cuanto a la Clasificación, específicamente en lo referente a si la estructura es una dimensión, una tabla de hecho o una tabla de hecho agregada. Si la tabla es una dimensión al nombre le precede las letras “dim” ejemplo dim_CAE, en caso de ser una tabla de hecho se le antepone las siglas “hech”, ejemplo, hech_indicadores_generales y cuando es una agregación se le especifica con las letras “hech_agg” ejemplo hech_agg_subordinacion.

En el caso de los atributos de las dimensiones se siguió la misma política en todas. Cuando se refiere a la llave de las dimensiones se le denominó “*dimensión*_**id**”, en caso que fuera algún código del negocio se le especificó “*dimensión*_**codigo**”. Así mismo con respecto a las descripciones: “*dimensión*_**descripcion**”. A modo de generalizar se puede decir que todos los atributos se nombraron como “*dimensión_clase*”, excepto cuando la dimensión posee más de un nivel jerárquico, en este caso se siguió la estrategia de nombrarlos “*nombre_de_la_jerarquía_clase*”, ejemplo de esto es en la dimensión dim_Indicador el atributo “tematica_descripcion”.

Con respecto a las medidas se nombraron mediante la agrupación de la clase “anno”, los clasificadores “actual”, “real”, “anterior” y las palabras primarias “plan”, “real”. Esta estructura es semejante a su nombre especificado dentro del modelo estadístico en cuestión. Un ejemplo de esto es la medida “anno_actual_plan” que dentro del modelo aparece como Año Actual Plan.

Igualmente se nombraron los scripts ya sea para la creación como para el llenado de cada una de las dimensiones y la tabla de hechos. Estos nombres están ordenados por las siglas “DDL” (*data definition language*, en ingles) y el nombre de la acción ejemplo DDL_metadatos_MD, al referirse a la estructura del Mercado de Datos.

Al finalizar este paso queda completamente estructurado la nomenclatura utilizada para la denominación de las tablas, atributos, medidas y scripts dentro de la base de datos. Ya después de tenerlos definidos es que se comienza con la implementación de las estructuras físicas.

2.8.2 Desarrollo del Modelo Físico

El punto de partida del modelo físico es el modelo lógico, este debe reflejar al modelo lógico en la mayor medida posible, sin embargo algunos cambios en las estructuras de las tablas y las columnas son necesarios para ajustarla a las características propias del Sistema Gestor de Base de Datos Relacionales (SGBDR) y las herramientas de acceso seleccionadas. Además, el modelo físico, contiene tablas de mantenimiento que usualmente no son incluidas en el modelo lógico.

La mayor diferencia entre ambos modelos es la especificación exhaustiva y detallada de las características físicas de la base de datos, comenzando por los tipos de datos hasta las tablas de segmentación, parámetros de almacenamiento de tablas y bandas de discos.

Para desarrollar el modelo físico generalmente es necesario la utilización de herramientas que automaticen el proceso, en el caso de la solución desarrollada se utilizó el Visual Paradim en su versión 6.1. Entre las ventajas que facilita el uso de esta herramienta se muestran claramente la integración del MD con otro modelos de datos corporativos, la ayuda de asegurar la consistencia en el nombrado y en las definiciones de tablas y columnas, la generación de los objetos físicos mediante el lenguaje DDL, entre otras.

Ya con el modelo físico cargado en la herramienta se comienzan las tareas de personalización de las estructuras físicas en función de la estandarización de formatos, nombres de objetos, la corrección de relaciones, en fin, todas las tareas concernientes a dejar a punto las estructuras dimensionales para ser desplegada en el SGBDR. En este sentido en la solución se corrigieron la utilización de algunos nombres de atributos de dimensiones y la corrección de tipos de datos, específicamente en la cantidad de caracteres que soportaban, además de la documentación de cada estructura y atributo con el fin de que quede descrito el modelo de datos completo.

La actividad final de este paso es la estimación inicial del tamaño de la base de datos. Para los desarrolladores de almacenes de datos es realmente crítico el saber cuánto va a almacenar con el fin de utilizar el impacto de esta variable en el rendimiento del sistema. Para esto la metodología utilizada propone un conjunto de tareas que a continuación se irán desarrollando.

Las longitudes de las filas afectan significativamente el tamaño de una base de datos debido a que ejemplos como las cadenas VARCHAR no siempre son explotadas en su totalidad y el SGBDR si le almacena espacio como si estuvieran totalmente llenas. Igualmente pasa con los campos vacíos o null, la inclusión de estos campos en la base de datos aumenta su tamaño y generalmente no es información útil para el usuario final. En el sistema propuesto se redujo a cero la cantidad de campos null dentro de la BD no siendo así con los campos VARCHAR debido a que la información de las dimensiones, en ocasiones, son oraciones completas y se hace muy difícil acotar el tamaño, como alternativa se establecieron los tamaños lo más pequeño posible para reducir al máximo la cantidad de espacios sin utilizar dentro del campo.

La actividad siguiente es estimar por cada tabla la cantidad de filas que podrá tener cuando el histórico esté cargado completamente en el sistema. Esta sección será abordada en profundidad en el capítulo 3.

2.8.3 Estrategia Inicial de Indexado

Las demoras del sistema ante operaciones que involucren grandes volúmenes de datos pueden ser reducidas. Es posible lograr optimizaciones ya sea por el tipo específico de gestor con que se manejen los datos y sus configuraciones puntuales, por el modelo de datos seleccionado, por configuraciones que se realicen sobre la base de datos, como por las optimizaciones en las consultas. Las técnicas de optimización pueden enfocarse entonces tanto en el nivel físico, por ejemplo, distribuyendo la información en distintos ficheros, discos, o incluso servidores, realizando un mayor número de operaciones en paralelo; como a la hora de proponer un diseño conceptual, seleccionando un modelo con el que se prevean realizar menos operaciones costosas. En la práctica se deciden aplicar varias de estas alternativas juntas. Aunque la intensión no es mencionar cada una de las opciones disponibles para optimizar, sí se quisiera poner a consideración algunas de ellas.

Sobre un Mercado de Datos se realizarán, muchas veces, consultas de gran complejidad que solicitarán información que cumpla determinados criterios, es decir, los usuarios frecuentemente querrán especificar los valores con los cuales se filtrarán los datos que deberán ser retornados. La mayoría de estas consultas incluirán, probablemente, operaciones de join entre tablas muy grandes, lo cual puede resultar extremadamente costoso. Para ganar en eficiencia a la hora de realizar estas operaciones se han investigado y creado técnicas especializadas que hoy ofrecen varios gestores, como los índices.

Para entender qué es un índice y cuál es su utilidad, se puede hacer un símil con los índices de los libros. Si un libro no tuviera índice y se necesitara leer sobre un tema en específico, tendría que recorrerse cada página hasta encontrar lo buscado, llegando necesariamente hasta el final, pues no se podría determinar cuándo se ha encontrado la última referencia al tópico de interés. Este proceso, definitivamente, haría consumir una cantidad considerable de tiempo. De manera similar, en las consultas que incluyen filtrar de acuerdo a uno o varios valores, se deben recorrer todas las filas de manera secuencial, buscando las que cumplen la condición. Si se tuviera una estructura que, al igual que un índice en un libro, guíe hasta encontrar las páginas de interés más rápido, serían más eficientes las búsquedas en el sistema.

Una de las técnicas de las que se dispone en SQL39 para reducir los tiempos de respuesta es, precisamente, los índices. Sin embargo, para usarlos de forma efectiva primero se debe saber cómo funcionan.

Un índice es una estructura física que permite un tipo de acceso alternativo al secuencial. Es creado a partir de una o varias columnas de una tabla, y, por lo general, es construido en forma de árbol balanceado (B-Tree). Al ser estructuras físicas, los índices van a tener un fichero asociado, en cuyas páginas se pueden almacenar uno o varios nodos del árbol. Cada uno de ellos apunta hacia otros nodos del árbol o hace referencia a las filas de la tabla. En cada nodo, los valores están ordenados, y los que se encuentran en un nodo hijo son menores o iguales que el valor en el nodo padre que le hace referencia. Los nodos que apuntan hacia las filas reciben el nombre de “páginas hojas”, y están enlazados entre sí: una página hoja apunta a otra hoja que contiene el próximo conjunto de valores.

Existe un tipo de índice con el cual se impone que los datos de la tabla estén ordenados en el nivel físico, y reciben el nombre de índices clusterizados (clustered index). Para cada tabla sólo se puede especificar un índice clusterizado, pues este afecta la forma en que son almacenadas las filas. Aquellos que no influyen en la organización física se denominan índices no clusterizados y varios pueden ser creados para una misma tabla. (England, y otros, 2007)

Las ventajas que tiene el uso de los índices están dadas, precisamente, por su estructura. Por ejemplo, las búsquedas de filas en las que un valor en particular aparezca no implican recorrer toda la tabla, sino que se utiliza la estructura arbórea del índice que se haya definido. Bajando desde la raíz del árbol, sólo es necesario desprenderse por una de las ramas hasta encontrar, en las páginas hojas, las referencias a

las filas en el fichero. Con esto se consume menos tiempo en hallar el resultado y es menor la cantidad de veces que se accede al disco para leer.

Se podría pensar entonces que la mejor opción es crear un índice por cada combinación de columnas. Sin embargo, sobre todas ellas en la práctica no se definen buenos criterios de búsqueda, por lo que no deberían crearse estas estructuras innecesariamente. Además, la creación de demasiados índices puede traer consecuencias no deseadas:

- Si se modifican valores en la tabla asociados a columnas sobre las que se hayan creado índices, o se insertan o eliminan filas, la estructura del índice se actualiza, pues el árbol asociado debe ser consistente con respecto a la información de la tabla. Esto va a influir, por tanto, en el comportamiento del gestor, pudiendo reducir la velocidad de procesamiento a la hora de realizar dichas operaciones. Aunque las operaciones que mayormente serán realizadas en un Mercado de Datos son de lectura, esto se debe tener en cuenta a la hora de realizar las cargas hacia el sistema, donde las operaciones de inserción y modificación son abundantes. Una alternativa que se podría analizar es eliminar los índices antes de comenzar la carga y volverlos a crear después.

- Como los índices se almacenan en ficheros al igual que los datos de una tabla, van a ocupar espacio de almacenamiento físico. Mientras más grande sea una tabla, mayores serán los índices asociados a ella. Por lo tanto, se debe analizar la capacidad de almacenamiento de que se dispone.

La solución más apropiada es decidir cuáles índices implicarán una mejora significativa en el rendimiento del sistema ante consultas. Algunas instrucciones que se pueden seguir son:

- Crear índices para las llaves primarias y foráneas: debido a que las operaciones de *join* consumen mucho tiempo, y para la mayoría de ellos las columnas por las que se realiza la unión son llaves foráneas, crear índices en las llaves implicadas en la unión puede ser ventajoso.

- Definir índices para las columnas incluidas en criterios de selección: si frecuentemente se deben seleccionar las filas de una tabla, filtrando por valores de una columna, es conveniente que dicha columna tenga definido un índice. Pero un criterio más fuerte que la frecuencia de consulta, lo

brindan el número de filas en la tabla (cardinalidad de la tabla) y el número de valores diferentes en la columna (cardinalidad de la columna): el impacto de un índice es generalmente mayor mientras mayor sea la cardinalidad de la tabla y/o de la columna.

La mayoría de los Sistemas Gestores de Bases de Datos proporcionan herramientas de prueba y evaluación para determinar la efectividad de un índice, con las cuales, luego de creado, se puede determinar si traerá mejoras significativas en el sistema.

Debido a la complejidad de muchas consultas que involucran realizar operaciones de *joins* entre tablas grandes, algunas formas especiales de índices han sido desarrolladas para agilizar este tipo de consultas. Algunos gestores los han incorporado, permitiendo lograr mayor eficiencia en los tiempos de respuesta ante solicitudes con propósitos analíticos.

Los índices multitabla o índices *join*, por ejemplo, permiten definir índices sobre columnas de dos o más tablas. Desde el punto de vista físico, la modificación con respecto a los índices antes explicados es que las referencias de las páginas hojas apuntan a varias filas en tablas diferentes. Esto mejora notoriamente las operaciones de unión donde participen dichas columnas. Otros índices son los de columnas virtuales, también denominados índices basados en funciones, que dan la posibilidad de definir índices sobre una expresión más allá que sobre columnas. En lugar de almacenar en el árbol los valores que aparecen en la columna, primero la expresión especificada es calculada y luego guardada en dicho árbol. Las hojas apuntan a las filas en las cuales el resultado de la expresión es igual al almacenado. La principal ventaja que ofrecen es la mejoría de la velocidad de procesamiento en consultas donde se utilice la expresión para filtrar. Existen también otras formas especiales de índices que se basan en estructuras diferentes del **B-Tree**, como el *índice Hash* y *el Bitmap*, este último utilizado generalmente cuando la cardinalidad de la columna es baja.

El estudio de los índices ha sido y continúa siendo un campo en desarrollo. A medida que surgen nuevas necesidades informativas y las consultas van ganando en complejidad, se hacen necesarias estas técnicas de optimización, con el fin de mejorar el comportamiento de los sistemas ante las solicitudes. Cada tipo de índice generalmente está enfocado a hacer eficientes las consultas, pero teniendo en cuenta los datos almacenados, su cantidad y variabilidad, factores que influyen a la hora de tomar la decisión de qué índices definir. Se recomienda, para un Mercado de Datos:

Considerar la posibilidad de añadir índices a varias tablas, atendiendo a factores como la frecuencia con que es consultada dicha tabla, la cantidad de filas que posee y las operaciones de join en las que puede verse involucrada. Tener en cuenta, para su creación, las capacidades de almacenamiento que se posea. Se recomienda, además, desactivar los índices durante los procesos de carga.

La solución posee implementado como indexado el que trae por defecto el gestor PostgreSQL, para la búsqueda de datos utilizando las llaves primarias, foráneas y campos únicos. Todas las llaves primarias, que son llaves subrogadas además, poseen índices de tipo “b-tree” (Árboles-B) lo que implica que cualquier búsqueda que se realice utilizando las llaves se optimizará mediante este método. Para las dimensiones el indexado propuesto es sobre la llave primaria de cada una de ellas, al igual que la tabla de hecho hech_indicadores_generales, sobre su llave hecho_id. Como en las agregaciones las búsquedas se hacen en función del tiempo y los indicadores asociados a él, se definieron 3 índices, uno para el atributo temporal_id, otro para el atributo indicador_id y uno final que relaciona los 2 anteriormente presentados con los demás tributos que conforman la llave de cada una de ellas.

2.8.4 Diseño y Construcción de la Instancia de la BD

Los Mercados de Datos existen dentro de una instancia de un SGBDR y este a su vez dentro de un servidor físico para que todo el sistema de parámetros, como la memoria, pueda ser optimizado en función de los requerimientos del almacén. Aunque los ajustes varían debido al SGBDR y por DWS en sí, algunos parámetros son absolutamente vitales para su desempeño. Generalmente el establecimiento de cada parámetro, por qué, y que parámetros tienen más probabilidad de requerir un ajuste en la base de datos crece y evoluciona.

El objetivo principal de este paso es el de garantizar la existencia de los requerimientos físicos mínimos necesarios para el buen funcionamiento del Sistema. El parámetro más necesario y vital para un MD es la adecuada disponibilidad de memoria debido a que en él se realizan complejas consultas y la cantidad de tablas físicas que se tienen que unir (*join*) para recuperar una consulta específica no son pocas, almacenándose en memoria para ser ofrecidas al usuario final. Esto implica que la solución propuesta necesite de al menos 2Gb de memoria para garantizar un rendimiento óptimo a las peticiones de información que se le soliciten. Otro parámetro a tener en cuenta es el procesador que tendrá el servidor físico pero para la solución en cuestión con un procesador Pentium IV es suficiente debido a que la

información almacenada sólo será la del Modelo Estadístico de Indicadores Generales ya cuando comiencen a crecer la cantidad de modelos se requerirá aumentar dicho parámetro pero esto queda fuera del alcance de la investigación actual.

2.8.5 Desarrollo de la estructura física de almacenamiento

Existe un nivel que se sitúa por debajo de las estructuras de datos en el cual se encuentran los archivos, discos, particiones, espacios de tablas, etc. La utilización adecuada de estos elementos y el dominio de los mismos inciden significativamente en el éxito del Mercado de Datos. En este sentido en la solución la utilización del tipo de disco duro queda a responsabilidad de los especialistas de la ONE debido a que se deben minimizar los costos de despliegue aunque aumente el riesgo de afectación sobre el sistema. Los elementos que si se van a tener en cuenta durante el desarrollo son el particionamiento de las tablas, en función de lograr una mayor organización de la información y velocidad en su recuperación, y estructuras de control de cambios con el fin de minimizar la utilización de recursos físicos cuando se refresquen las agregaciones. Para garantizar este fin se definieron 3 esquemas para almacenar las tablas:

1. “esq_dim” donde se ubicarán las dimensiones propuestas.
2. “esq_hech” donde aparecerán las tablas de hechos, ya sea, el repositorio central o la de las agregaciones.
3. “esq_part” en él se encontrarán las tablas de metadatos relacionadas con el control de cambio, el particionamiento y las funciones de refrescamiento de las agregaciones.

Además de los esquemas definidos se utilizaron un conjunto de tablespace para separar la utilización de los recursos físicos. Los tablespace utilizados son: tb_medida (para agrupar todas la tablas de hechos), tb_dimension (para todas las tablas de dimensiones), tb_agg (para las tablas de hechos creadas en las agrupaciones), tb_indices (para que sea utilizado por los índices b-tree creados) y tb_part (para los metadatos utilizados en el particionamiento y control de cambios).

La estrategia de particionado está basada en la orientación a objeto de postgreSQL utilizando la herencia para lograr dicho fin. El criterio que se consideró es el año debido a que los principales pedidos de información que se le realizan a la ONE están referidos a la fecha de captación. De esta forma se logra optimizar la gestión de almacenamiento y se facilita la recuperación y consulta. La funcionalidad se

sustenta en un conjunto de funciones que automatizan el proceso de particionamiento de la forma más genérica posible. Con la utilización de la función "*fn_crear_particion"* pasándole por parámetros el nombre la nueva tabla, el nombre de la tabla base, de la cual se va a heredar, y el año de la misma se crea la tabla física en la base de datos, asociándole una regla, con el fin de garantizar la inserción de la información correspondiente a dicho año, y un trigger que gestionará los cambios que se realicen sobre ella. El fin de esta función es la inserción dentro de la dimensión temporal los datos asociados al año introducido, en caso de no existir. Para cumplir con este proceso se adicionaron 2 tablas de metadatos: una primera llamada "*particion_base*" con el fin de almacenar los nombres de las tablas bases definidas y otra llamada "*particiones*" que es la que relaciona las tablas bases con sus particiones correspondientes.

Las estructuras diseñadas para el control de cambios está basada específicamente en 2 tablas de metadatos, similar a las diseñadas para el particionamiento, pero en este caso con el fin de almacenar la acción que se realizó, sobre que tabla y los valores asociados. Garantizando que quede almacenada esta información y cuando se ejecuten las funciones de refrescamiento sólo se refrescan los valores nuevos o modificados minimizándose de esta forma el tiempo y la utilización del servidor. Las tablas en cuestión son "*control_cambios*" y "*control_cambios_tuplas*", en el caso de la segunda es una especialización de la primera con el fin de almacenar todos los datos de la tupla modificada o eliminada, no siendo así en el caso de las inserciones. En el Anexo 2 Funciones Implementadas se describe cada una de ellas.

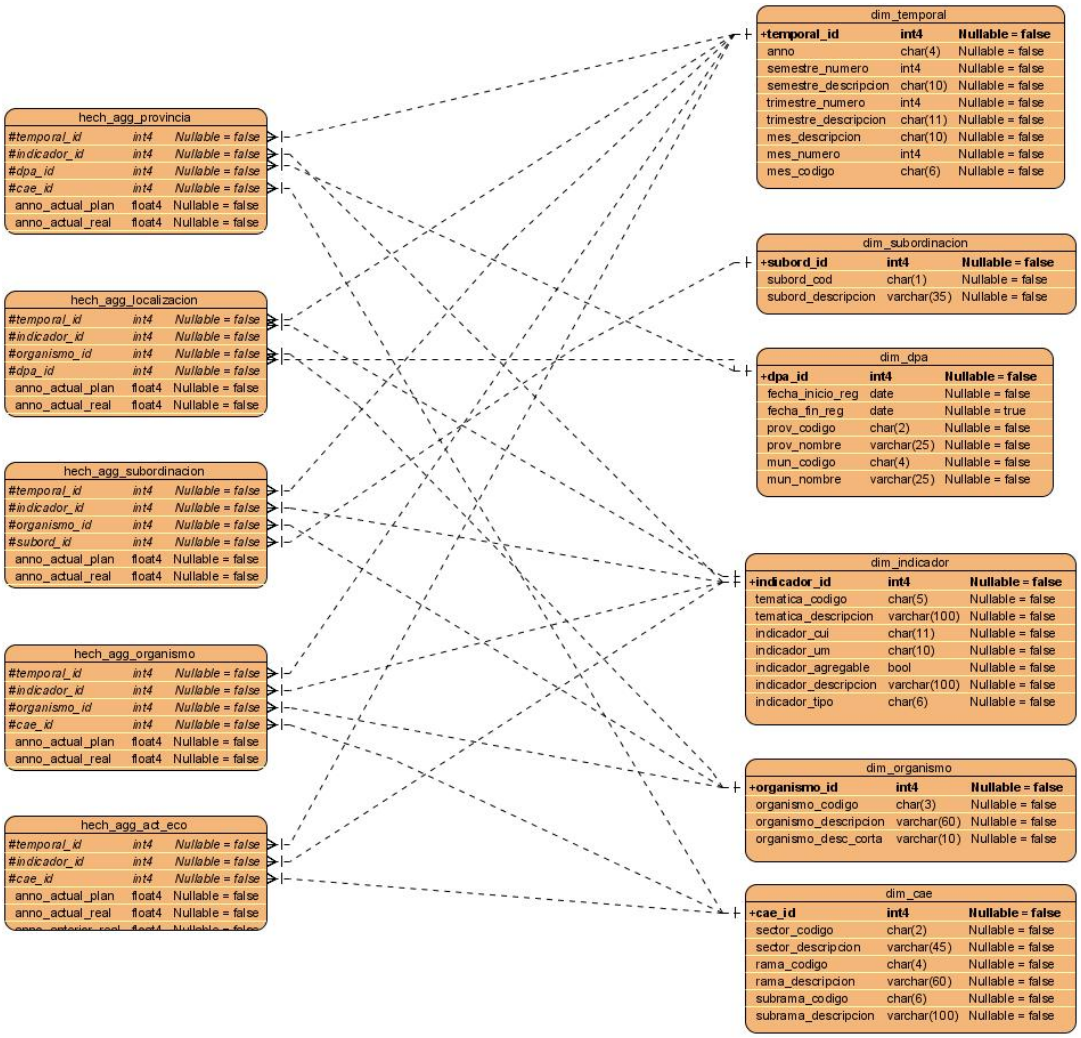

Figura 10 Modelo Dimensional de las Agregaciones

Debido al gran volumen de información que se manipulan se le deposita gran importancia a la utilización de agregaciones, que no son más que consolidados de la información, orientados a los reportes más comunes y que relacionan la información en dependencia de las dimensiones que interactúan en ellas. Ver Figura 10.

A continuación se relacionan las Tablas de Hechos que se conformaron sobre la base a las necesidades de información detectadas en la ONE. Vale aclarar que el llenado de dichas tablas no se realizará desde

las fuentes sino desde la información almacenada en el repositorio central y que ha sido previamente validada por los especialistas de la Dirección de Estadísticas Sociales de la ONE.

Tabla de Hechos SUBORDINACIÓN

En esta tabla es donde se almacena la información existente del modelo estadístico orientada a la subordinación que presentan los Centros Informantes que facilitaron la información.

Dimensiones relacionadas:

1. Organismos (dim_organismo)
2. Subordinación (dim_subordinacion)
3. Indicadores (dim_indicador)
4. Temporal (dim_temporal)

Medidas:

1. Plan
2. Real
3. Año Anterior

Tabla de Hechos ORGANISMO

En esta tabla es donde se almacena la información existente orientada a los organismos bajo los cuales se subordinan Centros Informantes que facilitaron la información.

Dimensiones relacionadas:

1. Organismos (dim_organismo)
2. Clasificador de Actividades Económicas (dim_cae)

3. Indicadores (dim_indicador)
4. Temporal (dim_temporal)

Medidas:

1. Plan
2. Real
3. Año Anterior

Tabla de Hechos CLASIFICADOR DE ACTIVIDADES ECONÓMICAS

En esta tabla es donde se almacena la información existente del modelo estadístico orientada al Clasificador de Actividades Económicas (CAE) bajo el cual se clasifican los Centros Informantes que facilitaron la información.

Dimensiones relacionadas:

1. Clasificador de Actividades Económicas (dim_cae)
2. Indicadores (dim_indicador)
3. Temporal (dim_temporal)

Medidas:

1. Plan
2. Real
3. Año Anterior

Tabla de Hechos PROVINCIA

En esta tabla es donde se almacena la información existente del modelo estadístico orientada a la localización geográfica donde están ubicados los Centros Informantes, agrupados por el CAE.

Dimensiones relacionadas:

1. División Político Administrativa (dim_dpa)
2. Clasificador de Actividades Económicas (dim_cae)
3. Indicadores (dim_indicador)
4. Temporal (dim_temporal)

Medidas:

1. Plan
2. Real
3. Año Anterior

Tabla de Hechos LOCALIZACIÓN

En esta tabla es donde se almacena la información existente del modelo estadístico orientada a la localización geográfica donde están ubicados los Centros Informantes, agrupados por el organismo.

Dimensiones relacionadas:

5. División Político Administrativa (dim_dpa)
6. Organismo (dim_organismo)
7. Indicadores (dim_indicador)

8. Temporal (dim_temporal)

Medidas:

4. Plan
5. Real
6. Año Anterior

2.8.6 Monitorización del Uso

La Monitorización del Uso es el último paso planteado por la metodología para el completamiento del Diseño Físico. La implantación de este paso dentro del Sistema contribuye al seguimiento de las respuestas a consultas, reportes y la carga de los datos hacia el almacén. Esta información es particularmente vital durante su desarrollo y mantenimiento. Este paso se encuentra regido por tres áreas fundamentales: rendimiento, soporte a usuarios y planificación.

Los datos obtenidos de la monitorización son usados para identificar cuáles son las tablas y columnas más expensas a ser unidas, seleccionadas, agregadas y filtradas. Esta información permite la inclusión de índices y esquemas con el fin de mejorar el tiempo de respuesta del sistema.

Se tiene previsto el soporte a los usuarios de la ONE durante los siguientes 6 meses después de implantada la solución, para dar seguimiento a la evolución del MD y a la experticia de los usuarios finales en su uso. Esto ocurriría siempre después de haber realizado una correcta transferencia tecnológica y capacitación a los DBA (*Database Administrators,* por sus siglas en inglés) y usuarios finales que lo administrarán y utilizarán respectivamente.

El seguimiento al crecimiento, promedio de tiempo de consulta, cuentas de usuarios concurrentes, variación del tamaño de la base de datos, tiempos de carga, proporcionarán estadísticas necesarias para ayudar a cuantificar los aumentos de capacidad y oportunidad de la solución. Al terminar este paso quedan las condiciones necesarias listas para la carga de los datos y comenzar con los procesos de

pruebas que validen el éxito de la solución propuesta. En el Capítulo 3 se desarrolla con más profundidad los resultados arrojados del crecimiento, las pruebas de volumen y carga realizadas.

2.9 Procesos de Extracción, Transformación y Carga

Debido a que los datos deberán ser extraídos, transformados, limpiados y cargados desde el conjunto de archivos DBF hacia el Mercado de Datos, es imprescindible conocer como se realizarán cada una de estas actividades. En este sentido, controlar desde dónde y en qué tiempo se realizará la extracción de los datos, que herramienta se utilizará así como dominar sus potencialidades, de qué tablas y cuáles datos se extraerán, es de vital importancia. En cuanto a la transformación de los datos, esta se hará de acuerdo a las reglas que se definieron en el negocio.

La extracción, transformación y carga de los datos en este sistema en particular no va a ser de la envergadura que normalmente estos procesos conllevan debido a que la información histórica se encuentra con un alto grado de limpieza, estandarización y calidad. Esto ocurre como consecuencia de los pasos previos que se realizan antes de la creación de los DBF. Sólo habría que centrar los esfuerzos en dos áreas específicas: en la extracción de los datos de las fuentes y en la carga hacia el MD. Vale aclarar que no se descarta la posibilidad que aparezca algún tipo de transformación sencilla, principalmente de formato y conversión, que se requiera hacer pero no sería el grueso para la realización del proceso.

Para los procesos de ETL se utilizarán las ventajas que brinda la herramienta Pentaho Data Integration, como se explicó en el Capítulo 1, en el cual se definen un conjunto de *jobs* con distintas características que posibilitarán este proceso. Con este fin el repositorio fue preparado con un conjunto de tablas que auxiliarían las estrategias definidas.

Tabla: *staging_indicador*

Es la tabla utilizada para la conversión de los códigos de los indicadores la cual almacena por cada fila, modelo y año, el código que posee ese indicador en el Clasificador Único de Indicadores vigente. Esta conversión es de gran importancia debido a que define la base para el uso de los códigos vigentes asociados a los indicadores del modelo estadístico.

Tabla: *staging_mes*

Esta tabla mantiene una relación entre los meses y su forma abreviada existente en la ONE debido a que es una forma bien específica dentro de su entorno de trabajo. Almacena un identificador del mes, su número dentro del año, el nombre del mes en cuestión y la abreviatura correspondiente. Esta conversión se utiliza mucho debido a que en los DBF la fecha viene mediante su abreviatura y en el MD se almacena con su nombre completo, ejemplo: la ONE utiliza JN para referenciarse a Junio.

Tabla: *hech_llaves_huerfanas*

Es la tabla que almacena todas las tuplas que se intenten insertar y posean incoherencias de referencia con las dimensiones asociadas debido a que poseían códigos que no estaban presentes en las tablas dimensionales. Su uso es muy útil para agrupar las incongruencias dentro de la información que se está almacenando y realizar procesos de corrección de la misma ya sea de forma manual o automática.

La última estrategia definida fue la de **llaves nulas** que consiste en crear una tupla en las tablas dimensionales con un valor que nunca será utilizado y cuando se vaya a insertar una fila en la tabla de hechos con valores dimensionales nulos se relacionarán con el valor definido con ese fin.

La carga hacia el MD se realizará directamente, es decir, evitando las tablas intermedias para el almacenado temporal de la información debido a que no son profundas las transformaciones ni la existencia de mejoras a la calidad de los datos. Este modelo dimensional, físicamente concluido en pasos anteriores, constituirá una base de datos separada, donde los datos estarán poblados sin inconsistencia, con limpieza, sin duplicaciones ni anulaciones.

Para el refrescamiento de las estructuras de agregaciones se realizará después de tener poblado completamente el repositorio central mediante rutinas en lenguaje SQL que serán ejecutadas a decisión del usuario, es decir, sin periodicidad definida. Generalmente se utilizará cuando se haya adicionado, al menos, la información perteneciente a un mes de captación.

2.10 Estrategia de Copias de Respaldo

Para garantizar la persistencia de la información y la contribución a no tener almacenada información que no sea útil a los analistas estadísticos de la ONE se definieron las siguientes directrices.

Se realizarán backups, con periodicidad mensual, de la información total que posea la BD garantizando en todo momento que exista una copia exacta de la información que está vigente en el servidor. Se realizará en periodos mensuales debido a que la ONE tiene definido que este modelo estadístico se capte a las entidades de esta manera. La estructura de carpetas definidas con este sentido será con la jerarquía Modelo -> Año -> Mes y el nombre del scripts será de la misma forma anno_modelo_mes logrando dejar plasmado claramente la fecha de la copia realizada.

Debido a que no es necesario que el tamaño del histórico crezca indefinidamente se propone, en concordancia con lo especificado por los especialistas funcionales, que el máximo de años a almacenar sea de 10 años y que a partir de su aumento la información sea almacenada anualmente en una estructura similar que las copias de respaldo pero llegando solamente hasta el nivel de año Modelo -> Año. Esto sucedería cuando se lleguen a completar los 2 años faltantes, es decir, 2009 y 2010. El nombre de los scripts para este caso sería anno_modelo.

Conclusiones del Capítulo

En este capítulo se describió el Sistema. Se detalló la arquitectura, el diseño lógico y físico, la estrategia de extracción, transformación y carga, así como la de copia de respaldo. Todo esto posibilitó llegar a las siguientes conclusiones:

- La Arquitectura propuesta sobre Plataforma Libre es flexible y escalable, adaptándose a las políticas vigentes en la ONE.
- Las estructuras dimensionales definidas abarcan los indicadores y parámetros necesarios para el análisis de las variables que recoge el Modelo de Indicadores Generales.
- La estrategia de particionamiento utilizada garantiza el almacenamiento organizado y eficiente de la información, permitiendo mejoras en la velocidad de refrescamiento de las agregaciones.

- La utilización de agregaciones e indexado mejoran los tiempos de respuesta en los pedidos de información.
- Con la implantación de la estrategia de respaldo se garantiza la seguridad, integridad y disponibilidad de los datos.

.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

Introducción

Al concluir el proceso de construcción del Mercado de Datos, el análisis de algunos aspectos tales como la normalización, las pruebas de volumen y carga, para el análisis de los tiempos de respuesta, la validación del rendimiento con la concurrencia de usuarios, y la validación del sistema; resultan tan importantes como el diseño y la implementación misma. El Mercado de Datos al entrar en contacto con los usuarios finales, entra en un ciclo iterativo e incremental, de lo simple a lo complejo, donde el sistema nunca descansará puesto que a él son adheridos, con el transcurso del tiempo, nuevos años de información, procesos de negocios de la empresa, nuevas necesidades o insatisfacciones del cliente. En el momento en que se implanta en la empresa y al entrar en plena explotación, el MD crece ilimitadamente, al ser alimentado con los datos históricos, a la vez que se vuelve complejo, y es cuando comienzan a observarse los beneficios de los tiempos de respuesta, el dinamismo en la elaboración de los reportes, los conocimientos que puedan ser extraídos de la información almacenada y la efectiva preparación de los usuarios finales, garantizan así el éxito del Sistema.

3.1 Normalización

La normalización, dentro del universo de los diseños relacionales, se justifica y adquiere un valor incalculable debido a que garantizan el éxito conceptual y lógico de la base de datos. Cuando se refiere a estructuras dimensionales la bibliografía especializada plantea que con el fin de garantizar el rendimiento, debido a que almacenan millones de tuplas, no se recomienda la normalización, ya sea, a nivel de dimensiones como de tablas de hechos. En el sistema desarrollado la tabla de hechos propuesta es la que se relaciona con cada una de las 13 dimensiones identificadas y, a su vez, es la única comunicación existente entre las dimensiones.

Ralph Kimball en su famoso libro "*The Data Warehouse Toolkit*" plantea claramente que las tablas dimensionales no tienen que estar normalizadas sino deben permanecer como tablas planas puesto que las tablas dimensionales normalizadas destruyen la habilidad de la presentación tabulada. Los espacios en disco salvados por la normalización de las tablas dimensionales, son típicamente menores que un

porciento del espacio total de disco necesario para el esquema completo. Los esfuerzos para normalizar cualquiera de las tablas en una base de datos dimensional solamente con el objetivo de salvar espacio en disco, son una pérdida de tiempo.

3.2 Calibrado de la Base de Datos

A partir de un estimado razonable que se realizara en cuanto al tamaño de la base de datos, se tendrá una concepción aproximada de la dimensión espacial total que alcanzaría el Mercado de Datos. Por tal razón, se realizará un análisis de cada una de las dimensiones propuestas para calcular la cantidad de unidades, la cantidad de filas implicadas en cada una de las tablas hasta llegar al número de bytes que serán ocupados por concepto de tamaño. Es necesario destacar que este análisis se hará con la información referente al Modelo Estadístico 0005 “Indicadores Generales” que se encuentra almacenada entre el rango de años 2000-2008, años de especial interés para la Oficina Nacional de Estadísticas. Este cálculo se realizará a partir de una propuesta hecha por el MSc Rosendo Moreno Rodríguez de la Universidad Central de las Villas “Marta Abreu” en su traducción del libro “*Data Warehouse Toolkit*” de Ralph Kimball.

3.2.1 Caso Crítico

Filas aproximadas por cada dimensión:

- Dimensión Tiempo: 9 años * 12 meses = 108 meses para el repositorio central y 24 para las agregaciones debido a que en las agregaciones solo se almacenan 2 años.
- Dimensión CAE: 216 elementos de los cuales 180 están relacionados con el modelo estadístico y aproximadamente 146 captan información mensualmente con una cardinalidad con la dimensión Empresa de 1 y con la de Organismo de 14 como promedio.
- Dimensión NAE: 269 elementos de los cuales 215 están relacionados con el modelo estadístico y aproximadamente 50 captan información mensualmente con una cardinalidad con la dimensión Empresa de 2.

- Dimensión Organismo: 216 organismos de los cuales 82 están relacionados con el modelo estadístico y aproximadamente 48 capta información mensualmente con una cardinalidad con la dimensión Empresa de 1 como promedio.

- Dimensión DPA: 170 municipios que tienen empresas asociadas de los cuales 147, aproximadamente, captan información mensualmente.

- Dimensión Forma de Financiamiento: 5 formas que tienen empresas asociadas de las cuales 3, aproximadamente, captan información mensualmente con una cardinalidad con la dimensión Empresa de 1.

- Dimensión Forma Organizativa: 18 formas que tienen empresas asociadas de las cuales 8, aproximadamente, captan información mensualmente con una cardinalidad con la dimensión Empresa de 1.

- Dimensión Subordinación: 7 subordinaciones que tienen empresas asociadas de las cuales 5 aproximadamente, captan información mensualmente con una cardinalidad con la dimensión Empresa de 1 y con la de Organismo de 2.

- Dimensión Esfera: 3 esferas que tienen empresas asociadas de las cuales 3, aproximadamente, captan información mensualmente con una cardinalidad con la dimensión Empresa de 1.

- Dimensión EAT: 5 estructuras bajo las cuales el gobierno subordina las empresas de las cuales 4, aproximadamente, captan información mensualmente con una cardinalidad con la dimensión Empresa de 1.

- Dimensión Empresa: 72 170 Empresas han reportado información a la ONE de las cuales 6057 están relacionadas con este modelo estadístico y aproximadamente 3542 captan información mensualmente.

- Dimensión Indicadores: 33 indicadores posee el modelo estadístico de los cuales 24, aproximadamente, se captan mensualmente con una cardinalidad con la dimensión Empresa de 11.

- Dimensión Modelo: 1 modelo estadístico almacenado.

Filas aproximadas por cada tabla de hechos:

- Tabla de Hechos Indicadores Generales: 108*1*2*1*147*1*1*1*1*1*3542*11*1 = 1 237 121 424
- Agregación Subordinación: 24*24*2*48 = 248 832
- Agregación Organismo: 24*24*48*14 = 387 072
- Agregación Provincia: 24*24*147*146 = 12 362 112
- Agregación Localización: 24*24*48*147 = 4 064 256
- Agregación Actividad Económica: 24*24*146 = 84 096

Total de Campos Claves en las tablas de hechos:

- Tabla de Hechos Indicadores Generales: 1
- Agregación Subordinación: 4
- Agregación Organismo: 4
- Agregación Provincia: 4
- Agregación Localización: 4
- Agregación Actividad Económica: 3

Total de Campos Foráneos en la tabla de hechos:

- Tabla de Hechos Indicadores Generales: 14

Total de Campos Medidas en las tablas de hechos:

- Tabla de Hechos Indicadores Generales: 3
- Agregación Subordinacion:3

- Agregación Organismo: 3
- Agregación Provincia: 3
- Agregación Localización: 3
- Agregación Actividad Económica: 3

Total de Campos en las tablas de hechos:

- Tabla de Hechos Indicadores Generales: 18
- Agregación Subordinacion:7
- Agregación Organismo: 7
- Agregación Provincia: 7
- Agregación Localización: 7
- Agregación Actividad Económica: 6

Tamaño de las Tablas de Hechos:

- Tabla de Hechos Indicadores Generales: 1 237 121 424*18*4(byte) = 83 Gb
- Agregación Subordinación: 248 832*7*4 = 0.006 Gb
- Agregación Organismo: 387 072*7*4 = 0.01 Gb
- Agregación Provincia: 12 362 112*7*4 = 0.3 Gb
- Agregación Localización: 4 064 256*7*4 = 0.1 Gb
- Agregación Actividad Económica: 84 096*6*4 = 0.001 Gb

Crecimiento Anual:

- Tabla de Hechos Indicadores Generales: 83/9 (cantidad de año) = 9.2 Gb
- Agregación Subordinación: 0.006/2 = 0.003 Gb.
- Agregación Organismo: 0.01/2 = 0.005 Gb.

- Agregación Provincia: 0.3/2 = 0.15 Gb.
- Agregación Localización: 0.1/2 = 0.2 Gb.
- Agregación Actividad Económica: 0.001/2 = 0.0005 Gb

De tal forma se puede concluir que el Mercado de Datos tendría un crecimiento anual de aproximadamente 9.6 Gb de información, y todo el histórico almacenaría un cúmulo de aproximadamente de 83.4 Gb. La Oficina Nacional de Estadísticas cuenta con la tecnología y la infraestructura necesaria para almacenar dicha información en sus servidores, por lo que su despliegue no requerirá la adquisición de equipamiento ni dispositivos adicionales de almacenamiento.

3.2.1 Caso Real

Después de haber cargado los datos de los 9 años que se tienen actualmente se evidenció que solamente el 0.4 % del caso pesimista fue el real almacenado provocando que el volumen de información almacenada en el dispositivo físico sea de aproximadamente 500 Mb.

3.3 Pruebas y Análisis del Rendimiento

Dentro del desarrollo de estos tipos de sistemas la realización de las pruebas es un paso importante para garantizar el éxito de la solución informática. El mecanismo que se seguirá para esto será la realización de pruebas de volumen y carga las cuales validarán la utilización del Mercado de Datos. En este punto se analizan los rendimientos del Sistema que se ha construido, al dar respuesta a distintos pedidos de información accediendo a la base de datos que se encuentra en el servidor PostgreSQL.

En este análisis se propone determinar los tiempos que demora en recuperar la información almacenada en el Mercado de Datos mediante consultas de distintos grados de complejidad, sobre una cantidad determinada de filas. El objetivo de este análisis es demostrar cuán óptimo, fácil y rápido se recupera la información de las estructuras dimensionales.

De tal forma se da comienzo al análisis ofreciendo algunos datos del caso de estudio:

- Fuente de datos a reportar: Base de Datos con diseño dimensional, en PostgreSQL, con información que ha sido facilitada por la ONE referente los datos estadísticos entre los años 2000-2008, del Modelo Indicadores Generales, de todo el país.
- Tipo de consulta a realizar: Consultas SQL.
- Agregaciones implicadas: Subordinación, Organismo, Actividad Económica, Provincia y Localización.
- Cantidad de Filas en la Tabla de Hechos: 3 070 863 filas.
    - 2000: 332 192 filas.
    - 2001: 404 065 filas.
    - 2002: 399 578 filas.
    - 2003: 367 326 filas.
    - 2004: 399 318 filas.
    - 2005: 324 774 filas.
    - 2006: 222 615 filas.
    - 2007: 362 504 filas.
    - 2008: 258 491 filas.
- **Características del Hardware del Servidor:**
    - Hardware: 1 Gb de memoria RAM, 120 Gb de capacidad de disco duro SATA, procesador Intel Pentium IV a 3.0 GHz de velocidad.
    - Software: SO Debian 5.0, PostgreSQL 8.3.7.

3.3.1 Pruebas de Volumen y Carga

Existen un conjunto de posibles pruebas que se le pueden realizar a un sistema informático para validar su uso, ejemplo de ellas se pueden mencionar: pruebas de unidad, integración, sistema, funcionalidad, volumen, carga, stress, etc; las que más impactan en el desarrollo de almacenes de datos son las pruebas que tengan relación con el rendimiento, capacidad y concurrencia. En este sentido las pruebas que se realizarán al Mercado de Datos serán las de volumen y carga.

Las pruebas de volumen son pruebas típicas de entornos que utilicen bases de datos. Las mismas se realizan para analizar el comportamiento del sistema o base de datos con volúmenes de datos almacenados lo más similar posible a los esperados en la explotación real del sistema. Para el sistema en cuestión la BD se pobló con los datos reales suministrados por los especialistas de la ONE, lo que implica que el tamaño es muy similar al real esperado.

Al introducir los datos no se presentaron problemas de límite de capacidad, ni de volumen de datos. Tampoco se detectaron desbordamientos de matrices, columnas, atributos, tipos de datos, ni peticiones excesivas de memoria. Las llaves autogeneradas no se salieron del rango especificado, ni se detectaron problemas con los tipos de datos definidos en el paso de diseño. Lo anteriormente planteado garantiza que el gestor utilizado y el diseño de las estructuras de la base de datos implementadas soportan completamente el almacenamiento de los niveles de información requeridos para la puesta en producción del Mercado de Datos.

Por otro lado, las pruebas de carga consisten en someter a una aplicación y/o base de datos a un régimen de carga de trabajo (habitualmente por simulación de concurrencia) similar al esperado en la explotación real del sistema. El objetivo de estas pruebas es buscar consultas mal diseñadas, consultas candidatas a optimización, la necesidad de índices adicionales, código mal diseñado, tiempo de demora de respuesta de magnitudes inaceptables, hardware insuficiente, problemas de control de concurrencia, etc.

Para la realización de las Pruebas de Carga existen diversos mecanismos y herramientas que automatizan dicho proceso. Se pueden utilizar desde navegadores ordinarios, trazas del servidor de base de datos, una aplicación simplificada, con consultas de la aplicación real, con un mínimo de código y sin complejidad algorítmica ni iteraciones, la utilización de herramientas diseñadas con este fin, entre otras.

Para realizar las pruebas se utilizarán las bondades que brinda le herramienta Jmeter por las facilidad de su uso y las funcionalidades que brinda. A continuación se argumentan dichas funcionalidades.

Apache-Jakarta Jmeter es un generador de carga diseñado para la realización de pruebas de carga y stress. Corre sobre la máquina virtual de java por lo que es multiplataforma. Genera carga por diversos protocolos, ya sea, FTP, HTTP, HTTPS, SQL, etc. Maneja cookies y autenticación. Realiza carga variable, en niveles de concurrencia, número de veces, tiempo, etc; y su característica principal radica en que pertenece a la familia de software libre.

La herramienta posee dos tipos de generación de carga, indirecta, es decir, a través de una aplicación y directa que basa fundamentalmente su utilización en consultas grabadas en la traza o log del servidor de base de datos. La que se va a utilizar para las pruebas del sistema es la directa configurada específicamente para la realización de consultas sobre el servidor de base de datos.

La arquitectura general que se utilizará para la realización de las pruebas serán 3 estaciones clientes con el Jmeter configurado directamente con el servidor de BD. Dos de las estaciones clientes sólo limitarán su uso a realizar peticiones indefinidamente al servidor y la otra para llevar las estadísticas con el número de muestras definido en 50. Se le realizarán pruebas con cantidades diferentes de usuarios concurrentes, 5 y 10 respectivamente, para realizar el análisis de los resultados debido a que según los especialistas de la ONE nunca existirán más de 25 usuarios registrados en el servidor y, en general, la concurrencia será mínima. Las consultas se realizarán sobre las agregaciones definidas para este fin. Se considera necesario aclarar que el servidor utilizado para las pruebas no posee todas las prestaciones de un servidor profesional debido a que se utilizó una estación cliente con características mejoradas. Esto afecta la calidad de las pruebas pero su objetivo es dar una idea del rendimiento de la solución. Ver Figura 11

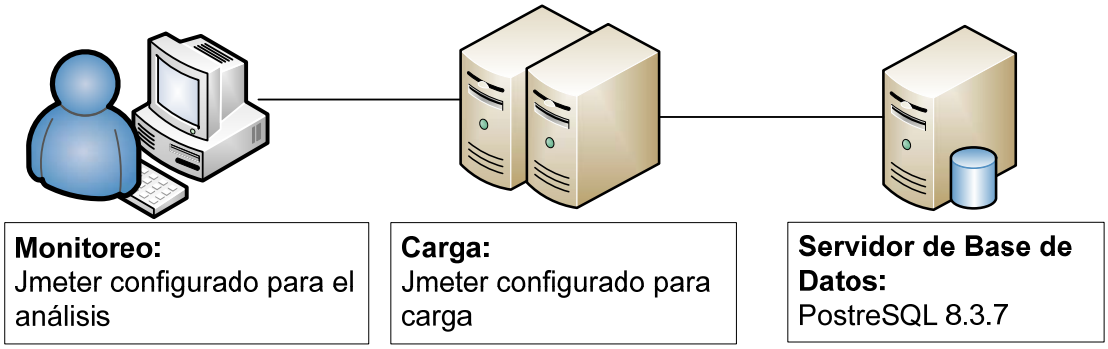

Figura 11 Configuración para las pruebas de carga

Prueba No 1: Comportamiento de los indicadores económicos por tipo de subordinación en el año 2008.

- Dimensiones involucradas: Temporal, Indicadores, Organismo y Subordinación.
- Cantidad Total de Filas en la agregación: 28 659 filas
- Cantidad Total de Filas recuperadas: 8191 filas
- Consulta: SELECT * FROM esq_hech.hech_agg_subordinacion WHERE temporal_id >= 100
- Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	0,62	0,71	0,19	0,88

  - 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	1,28	1,46	0,65	1,62

  - Gráfico de relación entre las pruebas

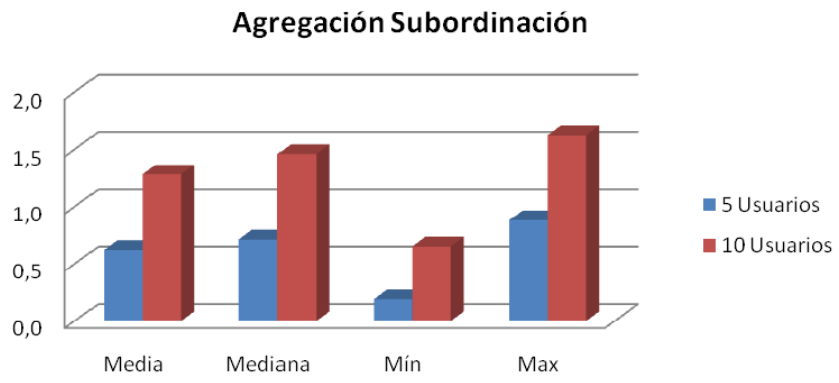

Gráfico 1 Representación de la Prueba 1

Prueba No 2: Comportamiento de los indicadores económicos por organismo en el año 2008.

- Dimensiones involucradas: Temporal, Indicadores, Organismo y CAE.
- Cantidad Total de Filas en la agregación: 143 425 filas
- Cantidad Total de Filas recuperadas: 40 289 filas
- Consulta: SELECT * FROM esq_hech.hech_agg_organismo WHERE temporal_id >= 100
- Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	2,96	3,29	0,72	4,14

- 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	6,75	6,96	3,36	7,27

- Gráfico de relación entre las pruebas

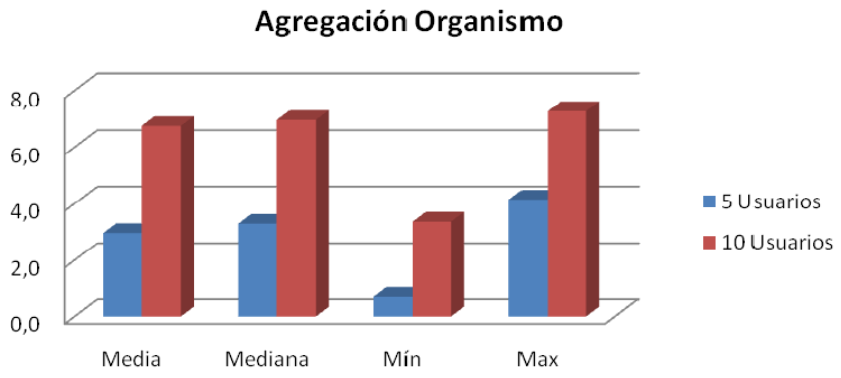

Gráfico 2 Representación de la Prueba 2

Prueba No 3: Comportamiento de los indicadores económicos por actividad económica en el año 2008.

- Dimensiones involucradas: Temporal, Indicadores y CAE.
- Cantidad Total de Filas en la agregación: 41 764 filas
- Cantidad Total de Filas recuperadas: 11 986 filas
- Consulta: SELECT * FROM esq_hech.hech_agg_act_eco WHERE temporal_id >= 100

- Cantidad de Usuarios:

  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	0,87	0,99	0,21	1,13

  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	1,41	1,50	0,30	2,12

  - Gráfico de relación entre las pruebas

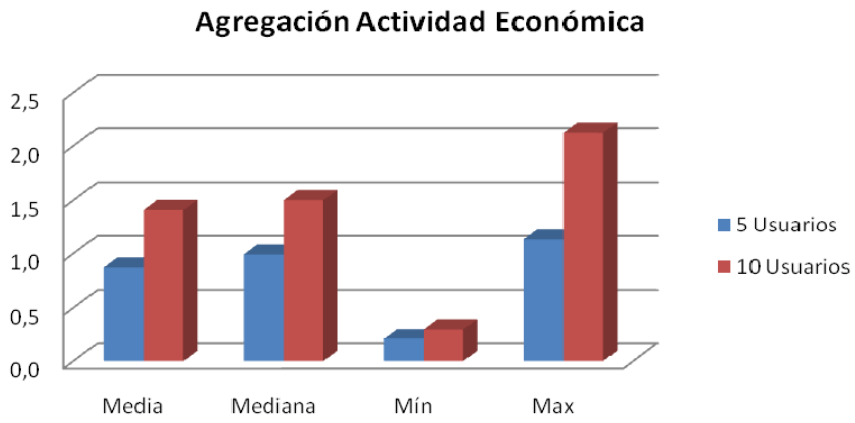

Gráfico 3 Representación de la Prueba 3

Prueba No 4: Comportamiento de los indicadores económicos por provincia en el año 2008.

- Dimensiones involucradas: Temporal, Indicadores, DPA y CAE.
- Cantidad Total de Filas en la agregación: 421 402 filas
- Cantidad Total de Filas recuperadas: 119 349 filas

- Consulta: SELECT * FROM esq_hech.hech_agg_provincia WHERE temporal_id >= 100
- Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	7,49	7,67	3,37	11,35

  - 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	19,49	20,70	10,15	21,62

  - Gráfico de relación entre las pruebas

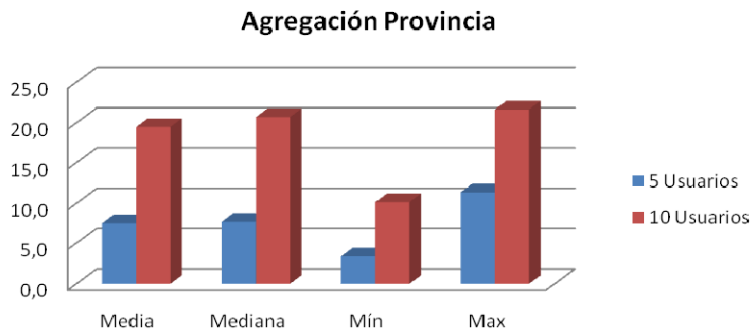

Gráfico 4 Representación de la prueba 4

Prueba No 5: Comportamiento de los indicadores económicos por localización en el año 2008.

- Dimensiones involucradas: Temporal, Indicadores, Organismo y DPA.
- Cantidad Total de Filas en la agregación: 195 683 filas
- Cantidad Total de Filas recuperadas: 55 843 filas

- Consulta: SELECT * FROM esq_hech.hech_agg_localizacion WHERE temporal_id >= 100
- Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	3,98	4,29	1,63	5,32

  - 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	9,59	9,71	5,19	10,23

  - Gráfico de relación entre las pruebas

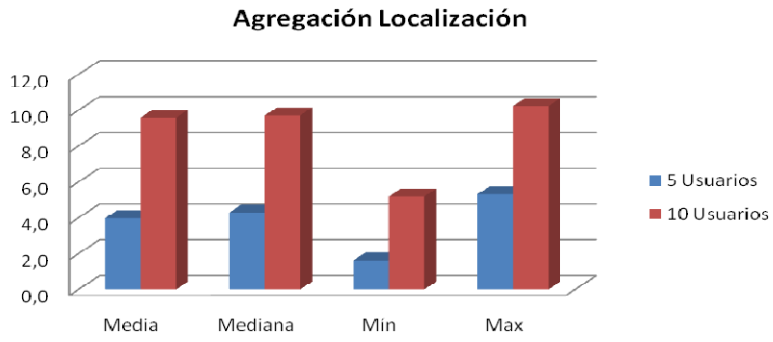

Gráfico 5 Representación de la prueba 5

Como puede apreciarse los resultados se enmarcan en 4 variables, aportadas por la herramienta Jmeter, de significativo valor para la presentación de los resultados de las pruebas: la media, valor de la suma aritmética de los tiempos de respuesta dividido entre 2, la mediana, valor de la variable que deja el mismo número de datos antes y después que él, una vez ordenados estos, de acuerdo con esta definición el conjunto de datos menores o iguales que la mediana representarán el 50% de los datos, y los que sean mayores que la mediana representarán el otro 50% del total de datos de la muestra; el valor mínimo, que

se refiere al tiempo de respuesta menor de todos los usuarios que hicieron peticiones concurrentes y el valor máximo que, similarmente, es el tiempo mayor de respuesta a todos los usuarios.

En las gráficas anteriores los tiempos de respuestas oscilaron en dependencia de la cantidad de filas que se recuperen en la consulta. En general los resultados obtenidos son satisfactorios evidenciándose un aumento casi simétrico entre cada una de las configuraciones realizadas. Destacándose los mayores tiempos en la agregación Provincia debido a que la consulta devuelve más cantidad de tuplas a cada usuario, convirtiéndose en un proceso un poco más lento.

3.4 Validación del Sistema

Después de haber concluido una primera iteración de ciclo de desarrollo del futuro Almacén de Datos Estadístico, en el cual quedan definidos aspectos importantes referentes al diseño como a la implementación del mismo, corresponde evaluar y validar el Sistema, donde la participación de los clientes es de suma importancia en la etapa

Resulta de gran importancia que los clientes estén inmersos en esta etapa de evaluación y validación del sistema, pues:

- Pueden ser encontradas discrepancias con los requerimientos identificados en la etapa de análisis.
- La familiarización con el ambiente de explotación de la información.
- Para refinar el Sistema en función de que quede lo más completo posible.

En el proyecto esta etapa duró aproximadamente 2 meses donde estuvo involucrada la principal cliente del mismo que es la Jefa de Informática de la ONE Ing. Elena Fernández García mediante un sistema de chequeo semanal donde se le presentaba los principales resultados y se definían las posibles funcionalidades a agregar.

Uno de los puntos fundamentales en este proceso fueron los tipos de reportes que la ONE realiza en su quehacer diario, por lo que los especialistas se encargaron de recopilar un variado y amplio conjunto de

tablas de salida, que ayudaron de sobremanera para enfocar el diseño de las estructuras hacia los principales pedidos de información.

Algunos de los detalles más significativos detectados por los clientes, y que han sido solucionados satisfactoriamente, se enuncian a continuación:

- La necesidad de disposición de la información necesaria para el cálculo de forma automática de indicadores autocalculados como: crecimiento, varianza, etc.
- El almacenamiento de la relación existente entre los códigos y los nombres, ya sea, de las empresas, organismos, CAE, NAE, etc.
- Posibilidad de obtener totales, promedios, porcientos, máximos, mínimos durante la realización de consultas OLAP.
- Eliminación de miembros precalculados, definidos en un inicio en el cubo de datos, por resultar innecesarios en el mismo.
- Necesidad de diseñar una arquitectura capaz de soportar, en futuras iteraciones, la incorporación de los 52 modelos estadísticos existentes en la ONE.

Para el logro del éxito en esta etapa ha tenido un valor significativo la disposición, colaboración y asistencia de los especialistas de la ONE que en todo momento han brindado la ayuda necesaria para la resolución de los problemas que fueron apareciendo a medida que se refinaba el Sistema.

El hecho de que el Sistema haya cumplido los objetivos propuestos inicialmente y satisfecho total o mayoritariamente los requisitos definidos, no significa que ya esté apto para ser montado e instalado en la entidad, sino que es importante también la preparación de los especialistas que interactuarán con el mismo, el cual fue comenzado con un taller impartido a todos los informáticos de las Oficinas de Estadísticas pertenecientes a las provincias habaneras y a la ONE en general. Habiéndole presentado el Sistema tanto sus características internas como la visualización de los reportes para su familiarización con el ambiente de explotación.

Finalmente se evidencia que el proceso de validación del Sistema se ha realizado satisfactoriamente, a pesar que el Mercado de Datos no abarca todas las áreas de la Oficina Nacional de Estadísticas, detalle que será solucionado en próximas iteraciones, pero en general los clientes se encuentran satisfechos con el trabajo realizado, hecho que ha quedado plasmado en la carta de aceptación por parte de ellos, la cual se pueden encontrar en el Anexo 9, por tal motivo se puede decir que los objetivos propuestos han sido cumplidos y las expectativas que se tenían con el Mercado de Datos han sido superadas.

Conclusiones del Capítulo

Después de analizar los resultados obtenidos en la etapa de validación se arribaron a las siguientes conclusiones:

- Las pruebas de calibrado garantizan una acertada política de mantenimiento y evolución de las estructuras creadas en función de mantener los niveles de servicio a partir de los requerimientos de almacenamiento.
- Las pruebas de volumen validaron la infraestructura de hardware y software propuestas garantizando la capacidad de gestión de los datos almacenados.
- Las pruebas de carga resultaron una herramienta eficaz en el proceso de optimización y demostraron que las técnicas de indexado y agregación cumplen con los tiempos de respuestas aceptables para este tipo de soluciones.

CONCLUSIONES

La investigación cumplió los objetivos planteados y se arribaron a las siguientes conclusiones:

- Los Mercados de Datos son una solución viable para el almacenamiento de datos pertenecientes al Modelo de Indicadores Generales.
- El Gestor de Base de Datos PostgreSQL garantiza el manejo de los volúmenes de información que necesita este tipo de solución.
- Se modelaron e implementaron las estructuras dimensionales necesarias que abarcan la información relevante del Modelo Estadístico de Indicadores Generales y soportan el proceso de toma de decisiones.
- La estrategia general de extracción, transformación y carga de los datos asegura la integración de los datos históricos hacia el Mercado de Datos.
- Las pruebas realizadas permitieron validar la solución propuesta obteniendo resultados satisfactorios en cada una de ellas.

RECOMENDACIONES

- La incorporación, sobre la arquitectura propuesta, de los modelos estadísticos restantes según la prioridad de los especialistas de la ONE.
- Migrar el Sistema a la plataforma NOVA.

BIBLIOGRAFÍA

Adamson, Christopher. 2006. *Mastering Data Warehouse Aggregates.* EUA : Wiley Publishing Inc, 2006.

Bernabeu, Ricardo Dario. 2007. *Hefesto: Metodología propia para la construcción de un Data Warehouse.* Argentina : s.n., 2007.

CIF vs MD Dos enfoques clásicos en el diseño de la arquitectura de un Data Warehouse. **Curto, Josep. 2008.** España : s.n., 2008.

England, Ken y Powell, Gavy. 2007. *Performance, Optimization and Tunning handbook.* eua : ElSEVIER Inc, 2007.

Hobbs, Lilian, y otros. 2005. *Oracle Database 10g Data Warehousing.* EUA : ELSEVIER Digital Press, 2005.

Huamantumba, Rayner. 2007. *Manual para diseño y desarrollo de Data Mart .* 2007.

Hurtado, y otros. 2003. *Base de Datos y Data Warehouse: Herramientas Estratégicas para la eficacia comercial.* Granada : s.n., 2003.

Imhoff, Claudia, Galemmo, Nicholas y Geiger, Jonathan G. 2003. *Mastering Data Warehouse Desing, Relational and Dimentional Techniques.* EUA : Wiley Publishing Inc, 2003.

Inmon, William H. 2005. *Building the Data Warehouse.* EUA : Wiley Publishing Inc, 2005.

Inmon, William H., Strauss y Neushloss. 2007. *DW 2.0 The Architecture for de Next Generation.* EUA : Wiley Publishing Inc, 2007.

Introducción a los Datawarehouses. **Chuc-Durán, Diana Graciela. 2007.** México : s.n., 2007.

Iznaga, Yonelbys. 2008. *Tesis: Sistema Data Warehouse.* Cuba : s.n., 2008.

Kimball, Ralph y Ross, Margy. 2002. *The Data Warehouse Toolkit.* EUA : Wiley Publishing Inc, 2002.

Kimball, Ralph, y otros. *The Data Warehouse Lifecycle Toolkit.* EUA : Wiley Publishing Inc.

Lockhart, Thomas. 2000. *Tutorial de PostgreSQL.* 2000.

Peñaloza, Lucía Victoria Hernández. 2008. *Tesis para logra el título de Magíster: Diseño y Construcción de un Data Mart para la mantención de Indicadores de Sostenibilidad de la Industria del Salmón.* Chile : s.n., 2008.

Ponniah, Paulraj. 2001. *Data Warehousing Fundamentals.* EUA : Wiley Publishing Inc, 2001.

Poole, y otros. 2003. *Common Warehouse Metamodel.* EUA : Wiley Publishing Inc, 2003.

Wang, John. 2006. *Encyclopedia of Warehousing and Mining.* EUA : Idea Group Reference, 2006.

Zenaido, Rosendo. 2008. *Borrador Tesis de Doctorado: Metodología para el Diseño de Almacenes de Datos.* España : s.n., 2008.

GLOSARIO DE TÉRMINOS

ONE: Oficina Nacional de Estadísticas

Centros Informantes: Los Centros Informantes son las empresas u organismos que suministran información a las oficinas de estadísticas en sus diferentes niveles.

Modelo Estadístico: Especie de planilla diseñada de forma matricial que almacena la información estadística.

Indicador: Se dice de la variable que puede tomar un valor de una determinada unidad de medida y de un determinado tipo de datos (generalmente numérico). Los indicadores de la ONE están bien definidos y tienen un código único que los identifica.

Clasificador: Es un instrumento que asigna un código a elementos ya definidos por otras vías.

EAT: Es un clasificador que organiza los Centro Informantes para ser atendidos por el gobierno.

NAE: Es un nomenclador, emitido a nivel mundial, para la organización de las actividades económicas de las empresas.

CAE: Es un clasificador, basado en el NAE, que organiza las actividades económicas de las empresas cubanas.

Ralph Kimball: Conocido innovador, escritor, educador y consultor en el campo de Almacenes de Datos. En la actualidad posee más de 100 artículos sobre inteligencia empresarial. Es Vicepresidente de *Metaphor Cumputer Systems*, pionera en software para ayuda a la toma de decisiones y proveedora de servicios de esta índole. La asociación Ralph Kimball fue creada en 1992 para proveer consultoría y educación sobre la tecnología de warehousing.

William H. Inmon: Conocido como “El padre de la tecnología de Almacenes de Datos”, es el creador de la metodología CIF (*Corporate Information Factory*) y más recientemente GIF (*Government Information Factory*). Posee más de 35 años de experiencia en tecnología de administración de base de datos y

diseño de almacenes de datos. Ha escrito más de 650 artículos sobre construcción, uso y mantenimiento de almacenes de datos. Posee la autoría de más de 46 libros de temas relacionados a tecnologías de base de datos.

ANEXOS

Anexo 1 Especificación de las Dimensiones

Tabla 5 Descripción Dimensión CAE

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
cae_id	Es la llave primaria de la dimensión. No posee significado para el negocio	250	2088, 2089, 3000	
sector_codigo	Almacena el código oficial de los sectores establecidos dentro del CAE	20	01, 02, 03	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
sector_descripcion	Almacena la descripción de los sectores establecidos dentro del CAE	20	Cultura y Arte, Finanzas y Seguros, Administración.	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
rama_codigo	Almacena el código oficial de las ramas establecidos dentro del CAE	85	0101, 0102, 0103	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva

				versión del dato guardado
rama_descripcion	Almacena la descripción de las ramas establecidas dentro del CAE	85	Cine, Hidroeconomía, Energía Eléctrica.	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
subrama_codigo	Almacena el código oficial del CAE.	220	Metalurgia ferrosa, Minería de la Sal, Industria Gráfica	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 6 Descripción Dimensión NAE

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
nae_id	Es la llave primaria de la dimensión. No posee significado para el negocio	270	2088, 2089, 3000	
sector_codigo	Almacena el código oficial de los sectores establecidos dentro del NAE	20	01, 02, 03	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva

				versión del dato guardado
sector_descripcion	Almacena la descripción de los sectores establecidos dentro del NAE	20	Pesca, Educación, Construcción.	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
rama_codigo	Almacena el código oficial de las ramas establecidos dentro del NAE	75	0101, 0102, 0103	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
rama_descripcion	Almacena la descripción de las ramas establecidas dentro del NAE	75	Ganadería, Pesca, Suministro de Agua	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
subrama_codigo	Almacena el código oficial del NAE.	270	Cría de Aves, Caza, Cultivo de Tabaco	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 7 Descripción Dimensión DPA

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
dpa_id	Es la llave primaria de la dimensión. No posee significado para el negocio	270	2088, 2089, 3000	
fecha_inicio_reg	Almacena la fecha de registro de la DPA en cuestión.	1	01/01/1976	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
fecha_fin_reg	Almacena la descripción de los sectores establecidos dentro del NAE	1	01/01/2009	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
prov_codigo	Almacena el código oficial de la provincia	15	01, 02, 03	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
prov_descripcion	Almacena la descripción del código de las provincias	15	Holguín, Villa Clara, Ciudad de la Habana	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
mun_codigo	Almacena el código oficial del	170	0101,0102,0301	Tipo1- sobrescrito

	municipio.			Tipo3- se registran la vieja y la nueva versión del dato guardado
mun_descripcion	Almacena la descipción oficial del municipio.	170	Holguín, Banes, Cacocún	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 8 Descripción Dimensión EAT

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
eat_id	Es la llave primaria de la dimensión. No posee significado para el negocio	5	1897, 1898, 1899	
eat_codigo	Almacena el código oficial establecido para la EAT	5	1, 2, 3	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
eat_descripcion	Almacena la descripción de las EAT establecidas	5	Comerciales, de servicio, inversionista.	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 9 Descripción Dimensión Empresa

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
empresa_id	Es la llave primaria de la dimensión. No posee significado para el negocio	100 000	2088, 2089, 3000	
empresa_codigo	Almacena el código oficial de los centros informantes establecidos a nivel nacional	100 000	00103, 00105, 00106	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
empresa_descripcion	Almacena el nombre de los centros informantes establecidos a nivel nacional	100 000	SIME, INDER, MINVEC	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 10 Descripción Dimensión Esfera

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
esfera_id	Es la llave primaria de la dimensión. No posee	3	1897, 1898, 1899	

	significado para el negocio			
esfera_codigo	Almacena el código oficial de los centros informantes establecidos a nivel nacional	3	1,2,9	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
esfera_desc	Almacena el nombre de las esferas establecidas a nivel nacional	3	MATERIAL, NO MATERIAL, OTRA	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 11 Descripción Dimensión Forma de Financiamiento

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
ff_id	Es la llave primaria de la dimensión. No posee significado para el negocio	3	1897, 1898, 1899	
ff_codigo_gral	Almacena el código oficial de la forma de financiamiento 2 establecida a nivel nacional	3	1,2,9	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato

				guardado
ff_descripcion_gral	Almacena la descripción oficial de la forma de financiamiento 2 establecida a nivel nacional	3	EMPRESAS y OEE PARCIALMENTE AUTOFINANCIADA, UNIDADES PRESUPUESTADAS y OEE PARCIALMENTE PRESUPUESTADAS, OTRA	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
ff_codigo	Almacena el código oficial de la forma de financiamiento 1 establecida a nivel nacional	5	11,22,12	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
ff_descripcion	Almacena la descripción oficial de la forma de financiamiento 1 establecida a nivel nacional	5	EMPRESAS, UNIDADES PRESUPUESTADAS, OTRA	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 12 Descripción Dimensión Forma Organizativa

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
forganizativa_id	Es la llave primaria de la dimensión. No	18	1897, 1898, 1899	

	posee significado para el negocio			
forganizativa_codigo	Almacena el código oficial de las formas organizativas establecidos a nivel nacional	18	01,02,99	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
forganizativa_descripcion	Almacena el nombre de las formas organizativas establecidas a nivel nacional	18	UBPC, CCS, EMPRESAS MIXTAS	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 13 Descripción Dimensión Indicador

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
indicador_id	Es la llave primaria de la dimensión. No posee significado para el negocio	18	1897, 1898, 1899	
tematica_codigo	Almacena el código oficial de las temáticas establecidos a	100	00401	Tipo1- sobrescrito Tipo3- se registran la vieja

	nivel nacional			y la nueva versión del dato guardado
tematica_descripcion	Almacena el nombre de las temáticas establecidas a nivel nacional	100	Países	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
indicador_cui	Almacena el código oficial de los indicadores establecidos a nivel nacional en el Clasificador Único de Indicadores	7000	00401010000, 00401044100, 00401046000	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
indicador_descripcion	Almacena el nombre oficial de los indicadores establecidos a nivel nacional en el Clasificador Único de Indicadores	7000	Producción Mercantil, Ingreso Turístico, Ventas Mayoristas	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
indicador_um	Almacena la unidad de medida oficial de los indicadores	20	U, Ton, Kg	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva

	establecidos a nivel nacional en el Clasificador Único de Indicadores			versión del dato guardado
indicador_agregable	Almacena la si un indicador determinado puede es aditivo.	2	0,1	Tipo1- sobrescrito
indicador_tipo	Almacena el tipo de indicadores establecidos a nivel nacional en el Clasificador Único de Indicadores	2	Físico, Valor	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 14 Descripción Dimensión Modelo

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
modelo_id	Es la llave primaria de la dimensión. No posee significado para el negocio	52	1897, 1898, 1899	
modelo_codigo	Almacena el código oficial de los modelos	52	0005	Tipo1- sobrescrito Tipo3- se

	estadísticos establecidos a nivel nacional			registran la vieja y la nueva versión del dato guardado
modelo_descripcion	Almacena el nombre de los modelos estadísticos establecidos a nivel nacional	52	Indicadores Generales	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
modelo_direccion	Almacena la dirección dentro de la ONE a la cual pertenece el modelo estadísticos	10	Economía	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 15 Descripción Dimensión Organismo

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
organismo_id	Es la llave primaria de la dimensión. No posee significado para el negocio	120	1897, 1898, 1899	
organismo_codigo	Almacena el código oficial de los organismos	120	100, 103, 105	Tipo1- sobrescrito Tipo3- se

	establecidos a nivel nacional			registran la vieja y la nueva versión del dato guardado
organismo_descripcion	Almacena el nombre de los organismos establecidos a nivel nacional	120	Industria de materiales de construcción, Banco Central de Cuba	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
organismo_desc_corta	Almacena las siglas de los organismos establecidos a nivel nacional	120	CIMEX, BNC, BCC	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 16 Descripción Dimensión Subordinación

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
subord_id	Es la llave primaria de la dimensión. No posee significado para el negocio	7	1897, 1898, 1899	
subord_codigo	Almacena el código oficial de las subordinaciones	7	1,2,9	Tipo1- sobrescrito Tipo3- se

	establecidas a nivel nacional			registran la vieja y la nueva versión del dato guardado
subord_descripcion	Almacena el nombre de las subordinaciones establecidas a nivel nacional	7	NACIONAL, PROVINCIAL, EMPRESAS MIXTAS	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Tabla 17 Descripción Dimensión Temporal

Nombre del atributo	Descripción	Cardinalidad	Dato Simple	Política de actualización
temporal_id	Es la llave primaria de la dimensión. No posee significado para el negocio	270	2088, 2089, 3000	
anno	Almacena el año de la información	10	2000, 2001, 2003	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
semestre_numero	Almacena el número del semestre de la información	2	1, 2	Tipo1- sobrescrito Tipo3- se registran la vieja

				y la nueva versión del dato guardado
semestre_descripcion	Almacena el nombre del semestre de la información	2	Semestre 1, Semestre 2	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
trimestre_numero	Almacena el número del trimestre de la información	4	1, 2, 3, 4	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
trimestre_descripcion	Almacena el nombre del trimestre de la información	4	Trimestre 1, Trimestre 2, Trimestre 3, Trimestre 4	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado
mes_numero	Almacena el número del mes de la información	12	1,2,3	Tipo1- sobrescrito
mes_descripcion	Almacena el nombre del mes de la información	12	Enero, Febrero, Marzo	Tipo1- sobrescrito

mes_codigo	Almacena el código del mes de la información	108	200001, 200203, 200812	Tipo1- sobrescrito Tipo3- se registran la vieja y la nueva versión del dato guardado

Anexo 2 Funciones Implementadas

- Función para registrar los cambios realizados sobre el repositorio central

```
CREATE OR REPLACE FUNCTION "esq_part"."fn_control_cambios" () RETURNS SETOF trigger AS
$body$
DECLARE
    acc varchar(6);
BEGIN
   IF(TG_OP = 'INSERT') THEN
     INSERT INTO esq_part.control_cambios (tabla_asociada, accion, fecha, hecho_id) VALUES
('esq_hech.hech_indicadores_generales', TG_OP::text, now(), NEW.hecho_id);
     RETURN NEW;
   ELSIF(TG_OP = 'DELETE') THEN
     IF(NOT EXISTS (SELECT * FROM esq_part.control_cambios WHERE esq_part.control_cambios.hecho_id =
OLD.hecho_id)) THEN
          INSERT INTO esq_part.control_cambios_tuplas VALUES
(nextval(('"esq_part"."control_cambios_cambios_id_seq"'::text)::regclass),'esq_hech.hech_indicadores_generales',
TG_OP::text, now(), OLD.hecho_id, OLD.temporal_id, OLD.indicador_id, OLD.nae_id, OLD.cae_id,
OLD.dpa_reg_id, OLD.dpa_est_id, OLD.organismo_reg_id, OLD.organismo_est_id, OLD.ff_id, OLD.forganizativa_id,
OLD.subord_id, OLD.esfera_id, OLD.eat_id, OLD.empresa_id, OLD.modelo_id, OLD.anno_actual_plan,
OLD.anno_actual_real, OLD.anno_anterior_real);
     ELSE
          SELECT esq_part.control_cambios.accion INTO acc FROM esq_part.control_cambios WHERE
esq_part.control_cambios.hecho_id = OLD.hecho_id;
          IF(acc = 'INSERT') THEN
               DELETE FROM esq_part.control_cambios WHERE esq_part.control_cambios.hecho_id =
OLD.hecho_id;
          ELSIF(acc = 'UPDATE') THEN
               UPDATE esq_part.control_cambios_tuplas SET accion = TG_OP::text, fecha = now() WHERE
esq_part.control_cambios_tuplas.hecho_id = OLD.hecho_id;
          END IF;
     END IF;
     RETURN OLD;
   ELSIF(TG_OP = 'UPDATE') THEN
     IF(NOT EXISTS (SELECT * FROM esq_part.control_cambios WHERE esq_part.control_cambios.hecho_id =
NEW.hecho_id)) THEN
          INSERT INTO esq_part.control_cambios_tuplas VALUES
(nextval(('"esq_part"."control_cambios_cambios_id_seq"'::text)::regclass),'esq_hech.hech_indicadores_generales',
TG_OP::text, now(), OLD.hecho_id, OLD.temporal_id, OLD.indicador_id, OLD.nae_id, OLD.cae_id,
OLD.dpa_reg_id, OLD.dpa_est_id, OLD.organismo_reg_id, OLD.organismo_est_id, OLD.ff_id, OLD.forganizativa_id,
OLD.subord_id, OLD.esfera_id, OLD.eat_id, OLD.empresa_id, OLD.modelo_id, OLD.anno_actual_plan,
OLD.anno_actual_real, OLD.anno_anterior_real);
     ELSE
          SELECT esq_part.control_cambios.accion INTO acc FROM esq_part.control_cambios WHERE
esq_part.control_cambios.hecho_id = NEW.hecho_id;
          IF(acc = 'INSERT') THEN
```

```
            UPDATE esq_part.control_cambios SET fecha = now() WHERE esq_part.control_cambios.hecho_id =
NEW.hecho_id;
          ELSIF(acc = 'UPDATE') THEN
            UPDATE esq_part.control_cambios_tuplas SET fecha = now() WHERE
esq_part.control_cambios_tuplas.hecho_id = NEW.hecho_id;
          END IF;
      END IF;
      RETURN NEW;
    END IF;
    RETURN NULL;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;
```

- Función para crear dinámicamente las tablas de particiones

```
CREATE OR REPLACE FUNCTION "esq_part"."fn_crear_particion" (nueva_partic varchar, partic_base varchar,
anno_particion integer) RETURNS SETOF boolean AS
$body$
DECLARE
      tabla_base varchar(50);
      tabla_nueva varchar(50);
      nombre ALIAS FOR $1;
      base ALIAS FOR $2;
      anno_p ALIAS FOR $3;
      id_base integer;

BEGIN
    tabla_nueva := 'esq_hech.'||nombre;
    tabla_base := 'esq_hech.'||base;
    SELECT esq_part.tablas_base.id_tabla_base INTO id_base
    FROM esq_part.tablas_base
    WHERE esq_part.tablas_base.nombre_tabla = tabla_base;

    BEGIN
        EXECUTE 'INSERT INTO esq_part.particiones (id_tabla_base, nombre_particion, anno) VALUES
('||id_base||','||quote_literal(tabla_nueva)||','||anno_p||')';
        EXECUTE 'CREATE TABLE '||tabla_nueva||'() INHERITS ('||tabla_base||') WITHOUT OIDS TABLESPACE
tb_medida';
        EXECUTE 'CREATE RULE regla_'||nombre||' AS ON INSERT TO '||tabla_base||' WHERE (temporal_id IN
(SELECT esq_dim.dim_temporal.temporal_id FROM esq_dim.dim_temporal WHERE esq_dim.dim_temporal.anno =
'||quote_literal(anno_p)||')) DO INSTEAD INSERT INTO '||tabla_nueva||' VALUES(NEW.hecho_id, NEW.temporal_id,
NEW.indicador_id, NEW.nae_id, NEW.cae_id, NEW.dpa_reg_id, NEW.dpa_est_id, NEW.organismo_reg_id,
NEW.organismo_est_id, NEW.ff_id, NEW.forganizativa_id, NEW.subord_id, NEW.esfera_id, NEW.eat_id,
NEW.empresa_id, NEW.modelo_id, NEW.anno_actual_plan, NEW.anno_actual_real, NEW.anno_anterior_real)';
        EXECUTE 'CREATE TRIGGER tr_'||nombre||' AFTER INSERT OR UPDATE OR DELETE ON '||tabla_nueva||'
FOR EACH ROW EXECUTE PROCEDURE "esq_part"."fn_control_cambios"()';
```

```
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 1,
'Semestre 1', 1, 'Trimestre 1', 'Enero', 1, anno_p||'01');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 1,
'Semestre 1', 1, 'Trimestre 1', 'Febrero', 2, anno_p||'02');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 1,
'Semestre 1', 1, 'Trimestre 1', 'Marzo', 3, anno_p||'03');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 1,
'Semestre 1', 2, 'Trimestre 2', 'Abril', 4, anno_p||'04');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 1,
'Semestre 1', 2, 'Trimestre 2', 'Mayo', 5, anno_p||'05');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 1,
'Semestre 1', 2, 'Trimestre 2', 'Junio', 6, anno_p||'06');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 2,
'Semestre 2', 3, 'Trimestre 3', 'Julio', 7, anno_p||'07');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 2,
'Semestre 2', 3, 'Trimestre 3', 'Agosto', 8, anno_p||'08');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 2,
'Semestre 2', 3, 'Trimestre 3', 'Septiembre', 9, anno_p||'09');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 2,
'Semestre 2', 4, 'Trimestre 4', 'Octubre', 10, anno_p||'10');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 2,
'Semestre 2', 4, 'Trimestre 4', 'Noviembre', 11, anno_p||'11');
        INSERT INTO "esq_dim"."dim_temporal" ("anno", "semestre_numero", "semestre_descripcion",
"trimestre_numero", "trimestre_descripcion", "mes_descripcion", "mes_numero", "mes_codigo") VALUES (anno_p, 2,
'Semestre 2', 4, 'Trimestre 4', 'Diciembre', 12, anno_p||'12');
    END;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;
```

- Función para refrescar los cambios en la agregación Actividad Económica

```
CREATE OR REPLACE FUNCTION "esq_part"."fn_refrescar_act_econ" () RETURNS SETOF boolean AS
$body$
DECLARE
    fila_hecho esq_hech.hech_indicadores_generales%ROWTYPE;
    fila_control esq_part.control_cambios_tuplas%ROWTYPE;
```

```
    fila record;
BEGIN
    FOR fila IN SELECT * FROM esq_part.control_cambios LOOP
      SELECT * INTO fila_hecho FROM esq_hech.hech_indicadores_generales WHERE
(esq_hech.hech_indicadores_generales.hecho_id = fila.hecho_id);
      IF(fila.accion = 'INSERT')THEN
        IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_act_eco WHERE
esq_hech.hech_agg_act_eco."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_act_eco."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_act_eco."cae_id" =
fila_hecho.cae_id)) THEN
          INSERT INTO esq_hech.hech_agg_act_eco VALUES (fila_hecho.temporal_id, fila_hecho.indicador_id,
fila_hecho.cae_id, fila_hecho.anno_actual_plan, fila_hecho.anno_actual_real, fila_hecho.anno_anterior_real);
        ELSE
          UPDATE esq_hech.hech_agg_act_eco SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_act_eco."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_act_eco."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_act_eco."cae_id" =
fila_hecho.cae_id;
        END IF;
      ELSIF(fila.accion = 'UPDATE')THEN
        SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
        UPDATE esq_hech.hech_agg_act_eco SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_act_eco."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_act_eco."indicador_id" = fila_control.indicador_id AND esq_hech.hech_agg_act_eco."cae_id" =
fila_control.cae_id;
        IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_act_eco WHERE
esq_hech.hech_agg_act_eco."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_act_eco."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_act_eco."cae_id" =
fila_hecho.cae_id)) THEN
          INSERT INTO esq_hech.hech_agg_act_eco VALUES (fila_hecho.temporal_id, fila_hecho.indicador_id,
fila_hecho.dpa_est_id, fila_hecho.subord_id, fila_hecho.anno_actual_plan, fila_hecho.anno_actual_real,
fila_hecho.anno_anterior_real);
        ELSE
          UPDATE esq_hech.hech_agg_act_eco SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_act_eco."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_act_eco."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_act_eco."cae_id" =
fila_hecho.cae_id;
        END IF;
      ELSE
        SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
        UPDATE esq_hech.hech_agg_act_eco SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
```

```
esq_hech.hech_agg_act_eco."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_act_eco."indicador_id" = fila_control.indicador_id AND esq_hech.hech_agg_act_eco."cae_id" =
fila_control.cae_id;
        END IF;
    END LOOP;
    RETURN NEW;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE RETURNS NULL ON NULL INPUT SECURITY INVOKER;
```

- Función para refrescar los cambios en la agregación Localización

```
CREATE OR REPLACE FUNCTION "esq_part"."fn_refrescar_localizacion" () RETURNS SETOF boolean AS
$body$
DECLARE
    fila_hecho esq_hech.hech_indicadores_generales%ROWTYPE;
    fila_control esq_part.control_cambios_tuplas%ROWTYPE;
    fila record;
BEGIN
    FOR fila IN SELECT * FROM esq_part.control_cambios LOOP
        SELECT * INTO fila_hecho FROM esq_hech.hech_indicadores_generales WHERE
(esq_hech.hech_indicadores_generales.hecho_id = fila.hecho_id);
        IF(fila.accion = 'INSERT')THEN
            IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_localizacion WHERE
esq_hech.hech_agg_localizacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_localizacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_localizacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_localizacion."dpa_id" = fila_hecho.dpa_est_id)) THEN
                INSERT INTO esq_hech.hech_agg_localizacion VALUES (fila_hecho.temporal_id,
fila_hecho.indicador_id, fila_hecho.organismo_est_id, fila_hecho.dpa_est_id, fila_hecho.anno_actual_plan,
fila_hecho.anno_actual_real, fila_hecho.anno_anterior_real);
            ELSE
                UPDATE esq_hech.hech_agg_localizacion SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_localizacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_localizacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_localizacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_localizacion."dpa_id" = fila_hecho.dpa_est_id;
            END IF;
        ELSIF(fila.accion = 'UPDATE')THEN
            SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
            UPDATE esq_hech.hech_agg_localizacion SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_localizacion."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_localizacion."indicador_id" = fila_control.indicador_id AND
```

```
esq_hech.hech_agg_localizacion."organismo_id" = fila_control.organismo_est_id AND
esq_hech.hech_agg_localizacion."dpa_id" = fila_control.dpa_est_id;
        IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_localizacion WHERE
esq_hech.hech_agg_localizacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_localizacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_localizacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_localizacion."dpa_id" = fila_hecho.dpa_est_id)) THEN
            INSERT INTO esq_hech.hech_agg_localizacion VALUES (fila_hecho.temporal_id,
fila_hecho.indicador_id, fila_hecho.organismo_est_id, fila_hecho.subord_id, fila_hecho.anno_actual_plan,
fila_hecho.anno_actual_real, fila_hecho.anno_anterior_real);
        ELSE
            UPDATE esq_hech.hech_agg_localizacion SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_localizacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_localizacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_localizacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_localizacion."dpa_id" = fila_hecho.dpa_est_id;
        END IF;
      ELSE
        SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
        UPDATE esq_hech.hech_agg_localizacion SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_localizacion."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_localizacion."indicador_id" = fila_control.indicador_id AND
esq_hech.hech_agg_localizacion."organismo_id" = fila_control.organismo_est_id AND
esq_hech.hech_agg_localizacion."dpa_id" = fila_control.dpa_est_id;
      END IF;
    END LOOP;
    RETURN NEW;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE RETURNS NULL ON NULL INPUT SECURITY INVOKER;
```

- Función para refrescar los cambios en la agregación Organismo

```
CREATE OR REPLACE FUNCTION "esq_part"."fn_refrescar_organismo" () RETURNS SETOF boolean AS

$body$
DECLARE
    fila_hecho esq_hech.hech_indicadores_generales%ROWTYPE;
    fila_control esq_part.control_cambios_tuplas%ROWTYPE;
    fila record;
BEGIN
    FOR fila IN SELECT * FROM esq_part.control_cambios LOOP
```

```
        SELECT * INTO fila_hecho FROM esq_hech.hech_indicadores_generales WHERE
(esq_hech.hech_indicadores_generales.hecho_id = fila.hecho_id);
        IF(fila.accion = 'INSERT')THEN
           IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_organismo WHERE
esq_hech.hech_agg_organismo."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_organismo."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_organismo."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_organismo."cae_id" = fila_hecho.cae_id)) THEN
              INSERT INTO esq_hech.hech_agg_organismo VALUES (fila_hecho.temporal_id, fila_hecho.indicador_id,
fila_hecho.organismo_est_id, fila_hecho.cae_id, fila_hecho.anno_actual_plan, fila_hecho.anno_actual_real,
fila_hecho.anno_anterior_real);
           ELSE
              UPDATE esq_hech.hech_agg_organismo SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_organismo."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_organismo."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_organismo."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_organismo."cae_id" = fila_hecho.cae_id;
           END IF;
        ELSIF(fila.accion = 'UPDATE')THEN
           SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
           UPDATE esq_hech.hech_agg_organismo SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_organismo."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_organismo."indicador_id" = fila_control.indicador_id AND
esq_hech.hech_agg_organismo."organismo_id" = fila_control.organismo_est_id AND
esq_hech.hech_agg_organismo."cae_id" = fila_control.cae_id;
           IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_organismo WHERE
esq_hech.hech_agg_organismo."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_organismo."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_organismo."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_organismo."cae_id" = fila_hecho.cae_id)) THEN
              INSERT INTO esq_hech.hech_agg_organismo VALUES (fila_hecho.temporal_id,
fila_hecho.indicador_id, fila_hecho.organismo_est_id, fila_hecho.subord_id, fila_hecho.anno_actual_plan,
fila_hecho.anno_actual_real, fila_hecho.anno_anterior_real);
           ELSE
              UPDATE esq_hech.hech_agg_organismo SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_organismo."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_organismo."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_organismo."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_organismo."cae_id" = fila_hecho.cae_id;
           END IF;
        ELSE
           SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
```

```
        UPDATE esq_hech.hech_agg_organismo SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_organismo."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_organismo."indicador_id" = fila_control.indicador_id AND
esq_hech.hech_agg_organismo."organismo_id" = fila_control.organismo_est_id AND
esq_hech.hech_agg_organismo."cae_id" = fila_control.cae_id;
      END IF;
    END LOOP;
    RETURN NEW;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE RETURNS NULL ON NULL INPUT SECURITY INVOKER;
```

- Función para refrescar los cambios en la agregación Provincia

```
CREATE OR REPLACE FUNCTION "esq_part"."fn_refrescar_provincia" () RETURNS SETOF boolean AS

$body$
DECLARE
    fila_hecho esq_hech.hech_indicadores_generales%ROWTYPE;
    fila_control esq_part.control_cambios_tuplas%ROWTYPE;
    fila record;
BEGIN
    FOR fila IN SELECT * FROM esq_part.control_cambios LOOP
      SELECT * INTO fila_hecho FROM esq_hech.hech_indicadores_generales WHERE
(esq_hech.hech_indicadores_generales.hecho_id = fila.hecho_id);
      IF(fila.accion = 'INSERT')THEN
        IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_provincia WHERE
esq_hech.hech_agg_provincia."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_provincia."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_provincia."dpa_id"
= fila_hecho.dpa_est_id AND esq_hech.hech_agg_provincia."cae_id" = fila_hecho.cae_id)) THEN
          INSERT INTO esq_hech.hech_agg_provincia VALUES (fila_hecho.temporal_id, fila_hecho.indicador_id,
fila_hecho.dpa_est_id, fila_hecho.cae_id, fila_hecho.anno_actual_plan, fila_hecho.anno_actual_real,
fila_hecho.anno_anterior_real);
        ELSE
          UPDATE esq_hech.hech_agg_provincia SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_provincia."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_provincia."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_provincia."dpa_id"
= fila_hecho.dpa_est_id AND esq_hech.hech_agg_provincia."cae_id" = fila_hecho.cae_id;
        END IF;
      ELSIF(fila.accion = 'UPDATE')THEN
        SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
        UPDATE esq_hech.hech_agg_provincia SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
```

```
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_provincia."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_provincia."indicador_id" = fila_control.indicador_id AND esq_hech.hech_agg_provincia."dpa_id"
= fila_control.dpa_est_id AND esq_hech.hech_agg_provincia."cae_id" = fila_control.cae_id;
        IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_provincia WHERE
esq_hech.hech_agg_provincia."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_provincia."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_provincia."dpa_id"
= fila_hecho.dpa_est_id AND esq_hech.hech_agg_provincia."cae_id" = fila_hecho.cae_id)) THEN
            INSERT INTO esq_hech.hech_agg_provincia VALUES (fila_hecho.temporal_id, fila_hecho.indicador_id,
fila_hecho.dpa_est_id, fila_hecho.subord_id, fila_hecho.anno_actual_plan, fila_hecho.anno_actual_real,
fila_hecho.anno_anterior_real);
        ELSE
            UPDATE esq_hech.hech_agg_provincia SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_provincia."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_provincia."indicador_id" = fila_hecho.indicador_id AND esq_hech.hech_agg_provincia."dpa_id"
= fila_hecho.dpa_est_id AND esq_hech.hech_agg_provincia."cae_id" = fila_hecho.cae_id;
        END IF;
      ELSE
        SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
        UPDATE esq_hech.hech_agg_provincia SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_provincia."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_provincia."indicador_id" = fila_control.indicador_id AND esq_hech.hech_agg_provincia."dpa_id"
= fila_control.dpa_est_id AND esq_hech.hech_agg_provincia."cae_id" = fila_control.cae_id;
      END IF;
    END LOOP;
    RETURN NEW;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE RETURNS NULL ON NULL INPUT SECURITY INVOKER;
```

- Función para refrescar los cambios en la agregación Subordinación

```
CREATE OR REPLACE FUNCTION "esq_part"."fn_refrescar_subordinacion" () RETURNS SETOF boolean AS

$body$
DECLARE
    fila_hecho esq_hech.hech_indicadores_generales%ROWTYPE;
    fila_control esq_part.control_cambios_tuplas%ROWTYPE;
    fila record;
BEGIN
    FOR fila IN SELECT * FROM esq_part.control_cambios LOOP
      SELECT * INTO fila_hecho FROM esq_hech.hech_indicadores_generales WHERE
(esq_hech.hech_indicadores_generales.hecho_id = fila.hecho_id);
```

```
        IF(fila.accion = 'INSERT')THEN
            IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_subordinacion WHERE
esq_hech.hech_agg_subordinacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_subordinacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_subordinacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_subordinacion."subord_id" = fila_hecho.subord_id)) THEN
                INSERT INTO esq_hech.hech_agg_subordinacion VALUES (fila_hecho.temporal_id,
fila_hecho.indicador_id, fila_hecho.organismo_est_id, fila_hecho.subord_id, fila_hecho.anno_actual_plan,
fila_hecho.anno_actual_real, fila_hecho.anno_anterior_real);
            ELSE
                UPDATE "esq_hech"."hech_agg_subordinacion" SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_subordinacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_subordinacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_subordinacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_subordinacion."subord_id" = fila_hecho.subord_id;
            END IF;
        ELSIF(fila.accion = 'UPDATE')THEN
            SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
            UPDATE esq_hech.hech_agg_subordinacion SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_subordinacion."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_subordinacion."indicador_id" = fila_control.indicador_id AND
esq_hech.hech_agg_subordinacion."organismo_id" = fila_control.organismo_est_id AND
esq_hech.hech_agg_subordinacion."subord_id" = fila_control.subord_id;
            IF (NOT EXISTS(SELECT * FROM esq_hech.hech_agg_subordinacion WHERE
esq_hech.hech_agg_subordinacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_subordinacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_subordinacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_subordinacion."subord_id" = fila_hecho.subord_id)) THEN
                INSERT INTO esq_hech.hech_agg_subordinacion VALUES (fila_hecho.temporal_id,
fila_hecho.indicador_id, fila_hecho.organismo_est_id, fila_hecho.subord_id, fila_hecho.anno_actual_plan,
fila_hecho.anno_actual_real, fila_hecho.anno_anterior_real);
            ELSE
                UPDATE esq_hech.hech_agg_subordinacion SET anno_actual_plan = anno_actual_plan +
fila_hecho.anno_actual_plan, anno_actual_real = anno_actual_real + fila_hecho.anno_actual_real,
anno_anterior_real = anno_anterior_real + fila_hecho.anno_anterior_real WHERE
esq_hech.hech_agg_subordinacion."temporal_id" = fila_hecho.temporal_id AND
esq_hech.hech_agg_subordinacion."indicador_id" = fila_hecho.indicador_id AND
esq_hech.hech_agg_subordinacion."organismo_id" = fila_hecho.organismo_est_id AND
esq_hech.hech_agg_subordinacion."subord_id" = fila_hecho.subord_id;
            END IF;
        ELSE
            SELECT * INTO fila_control FROM esq_part.control_cambios_tuplas WHERE
(esq_part.control_cambios_tuplas.hecho_id = fila.hecho_id);
            UPDATE esq_hech.hech_agg_subordinacion SET anno_actual_plan = anno_actual_plan -
fila_control.anno_actual_plan, anno_actual_real = anno_actual_real - fila_control.anno_actual_real,
```

```
anno_anterior_real = anno_anterior_real - fila_control.anno_anterior_real WHERE
esq_hech.hech_agg_subordinacion."temporal_id" = fila_control.temporal_id AND
esq_hech.hech_agg_subordinacion."indicador_id" = fila_control.indicador_id AND
esq_hech.hech_agg_subordinacion."organismo_id" = fila_control.organismo_est_id AND
esq_hech.hech_agg_subordinacion."subord_id" = fila_control.subord_id;
        END IF;
    END LOOP;
    RETURN NEW;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE RETURNS NULL ON NULL INPUT SECURITY INVOKER;
```

Anexo 3 Diseño de las tablas para el Control de Cambio

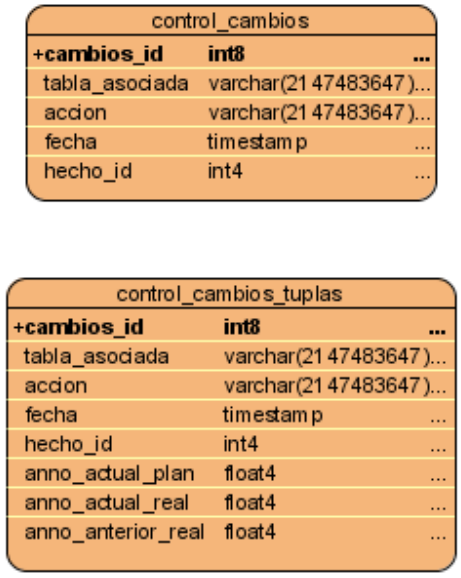

Figura 12 Tablas definidas para el control de cambio

Anexo 4 Diseño de las tablas para el Particionamiento

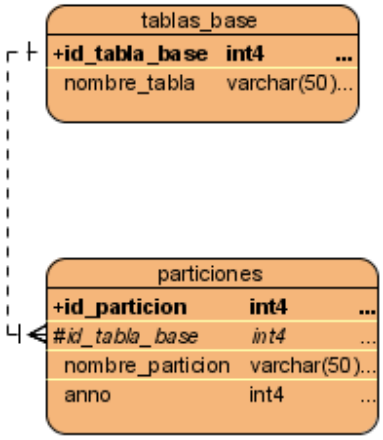

Figura 13 Tablas definidas para el particionamiento

Anexo 5 Transformación para el llenado de la tabla de hechos

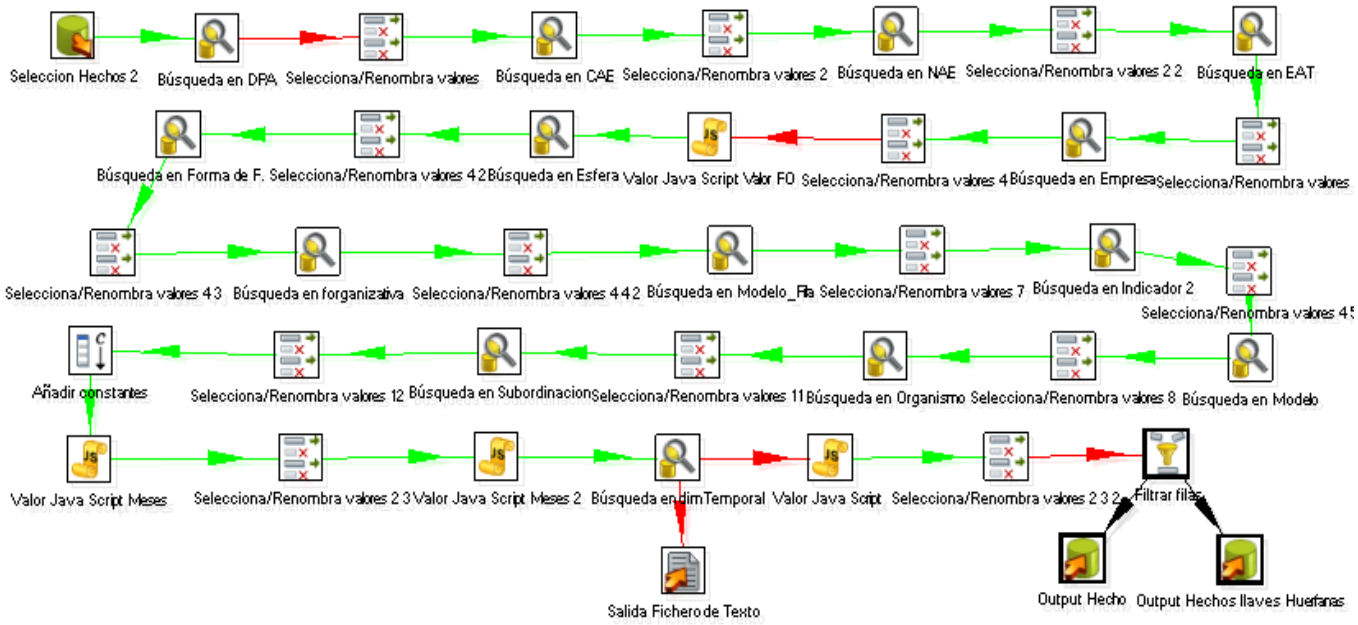

Figura 14 Jobs diseñado en el Pentaho Data Integration

Anexo 6 Transformación para la revisión de los elementos almacenados en las tablas huérfanas

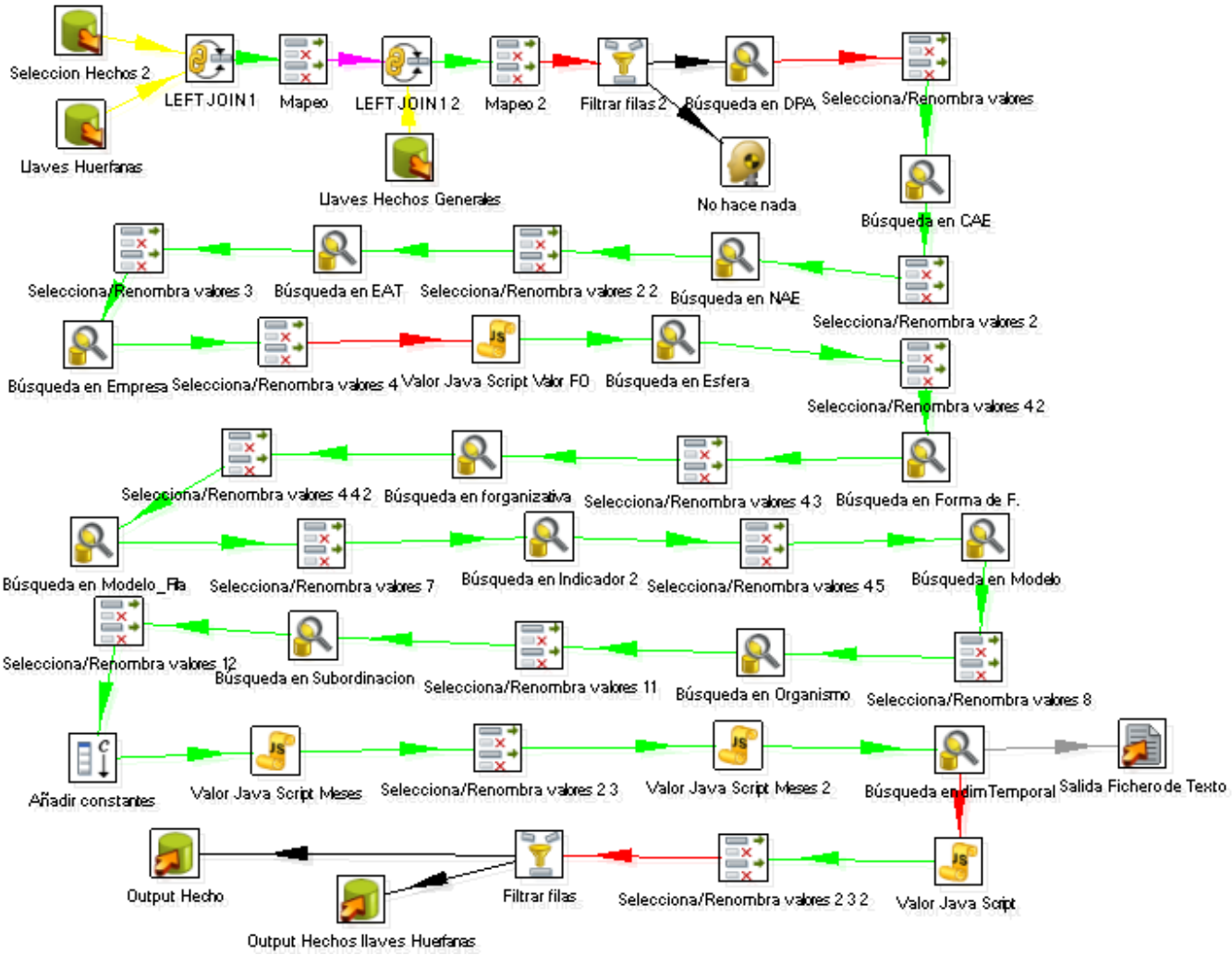

Figura 15 Jobs diseñado en Pentaho Data Integration

Anexo 7 Resultado de las Pruebas de Carga

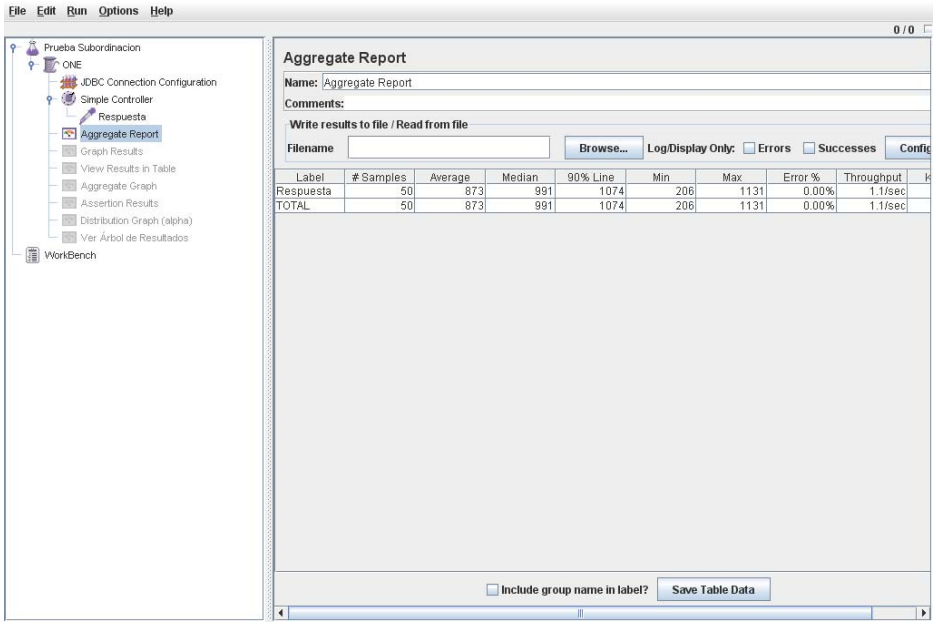

Figura 16 Prueba sobre la agregación Actividad Económica para 5 usuarios concurrentes

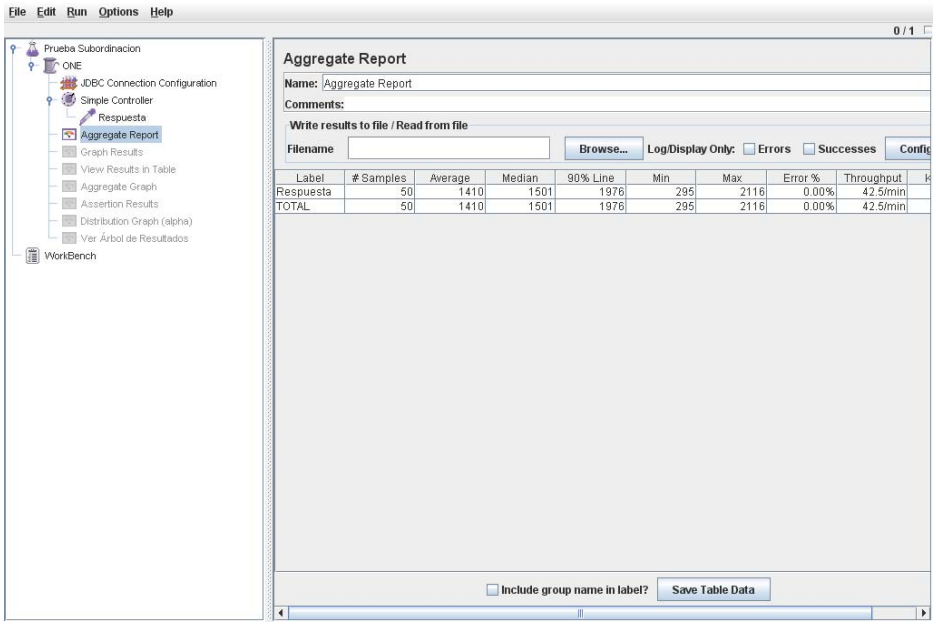

Figura 17 Prueba sobre la agregación Actividad Económica para 10 usuarios concurrentes

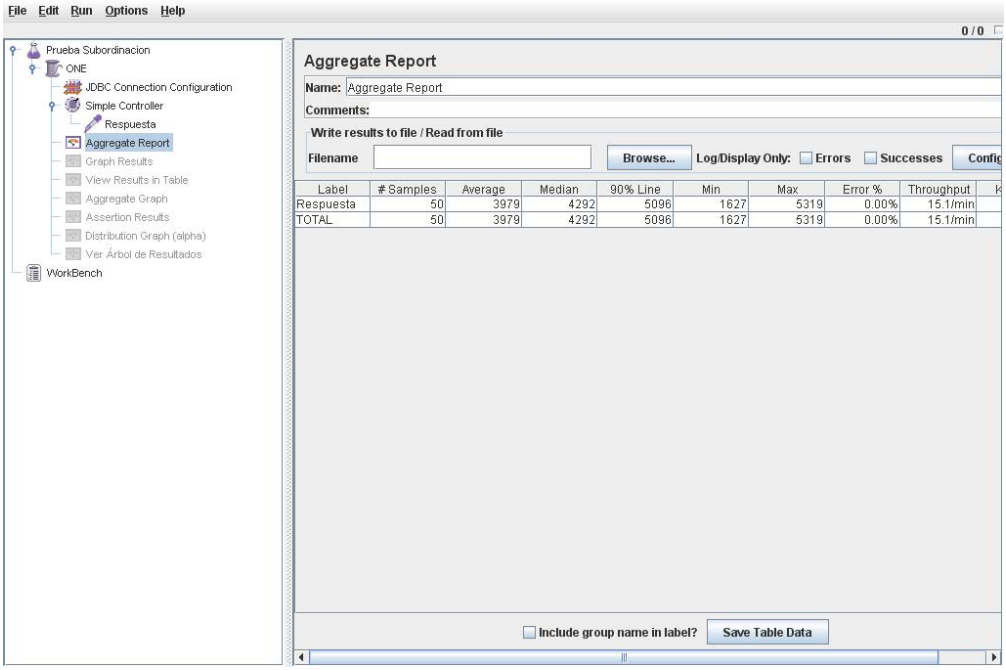

Figura 18 Prueba sobre la agregación Localización para 5 usuarios concurrentes

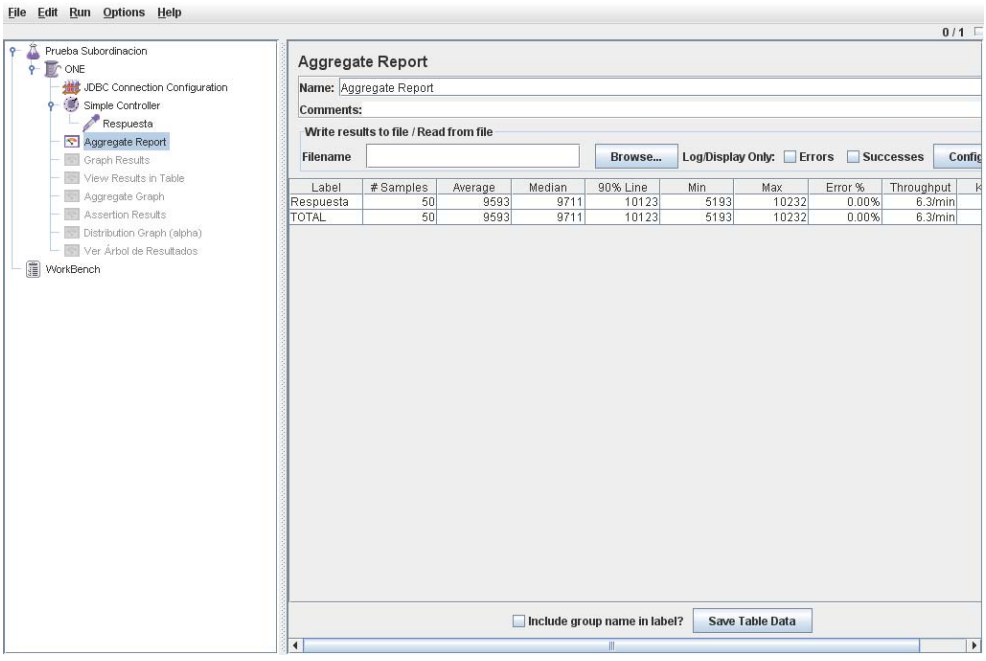

Figura 19 Prueba sobre la agregación Localización para 10 usuarios concurrentes

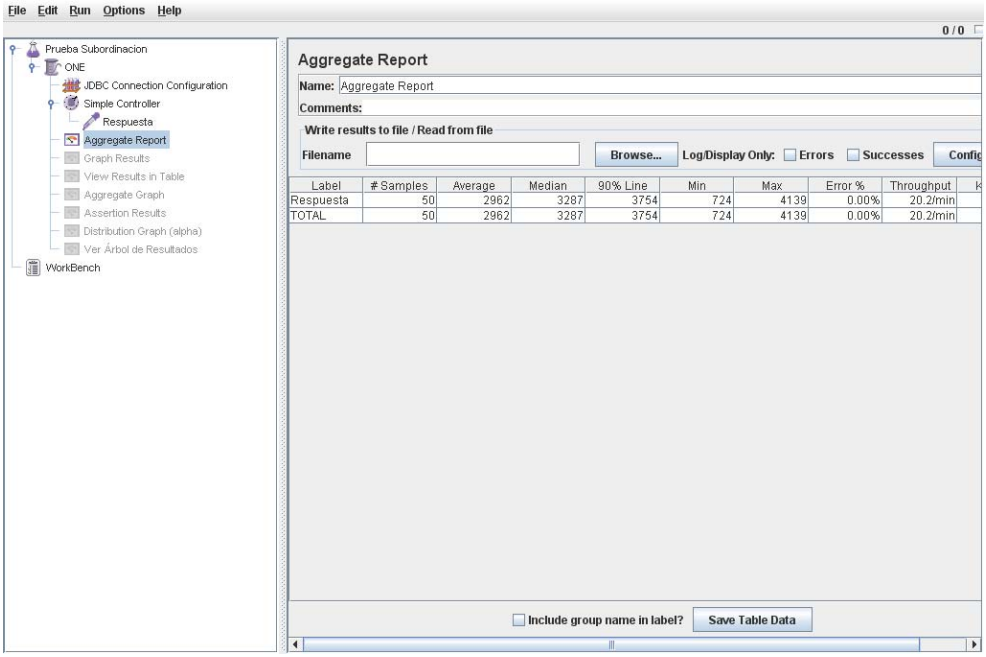

Figura 20 Prueba sobre la agregación Organismo para 5 usuarios concurrentes

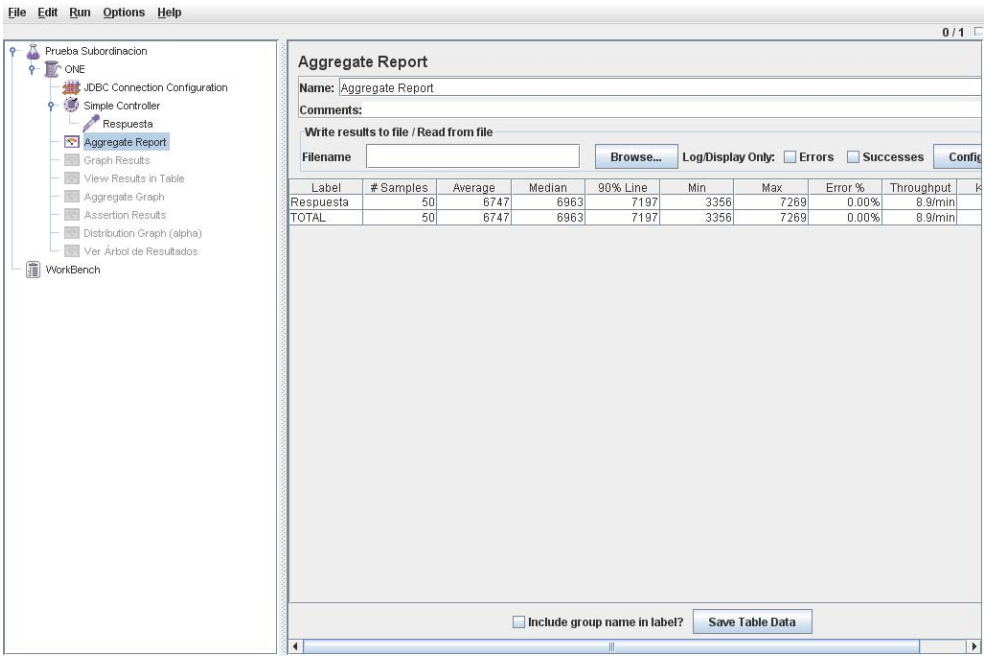

Figura 21 Prueba sobre la agregación Organismo para 10 usuarios concurrentes

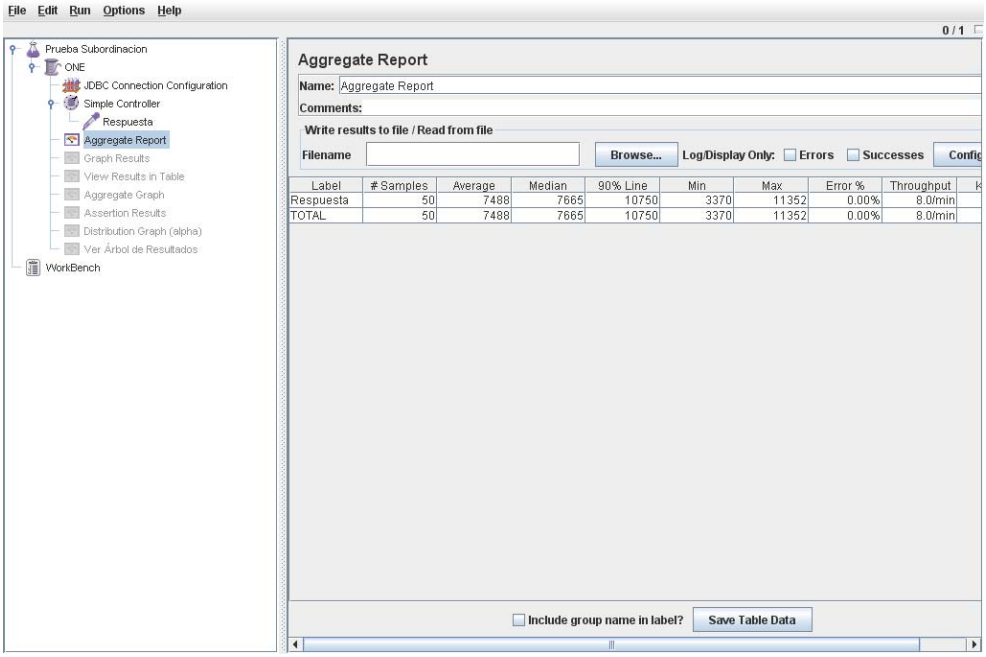

Figura 22 Prueba sobre la agregación Provincia para 5 usuarios concurrentes

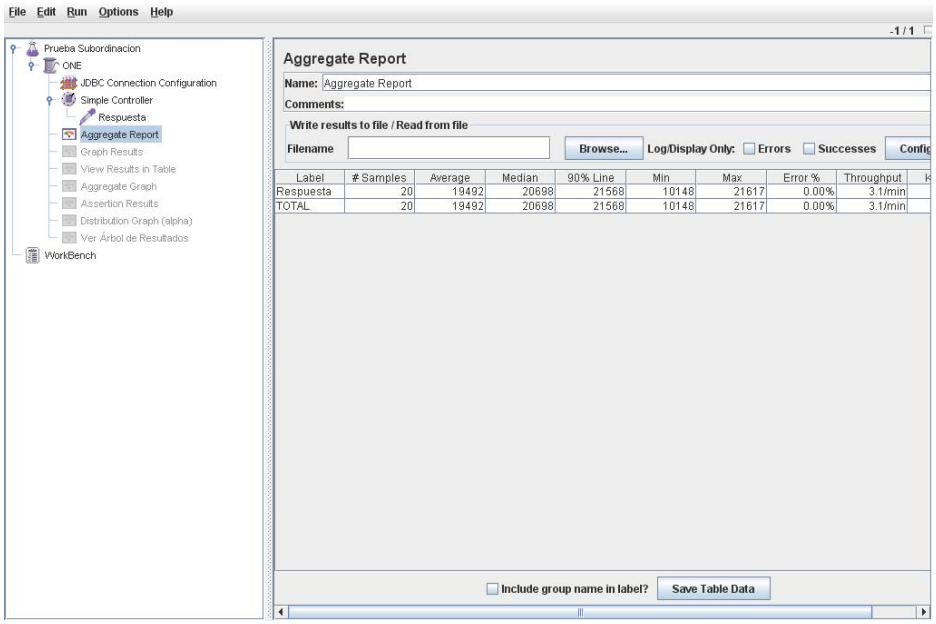

Figura 23 Prueba sobre la agregación Provincia para 10 usuarios concurrentes

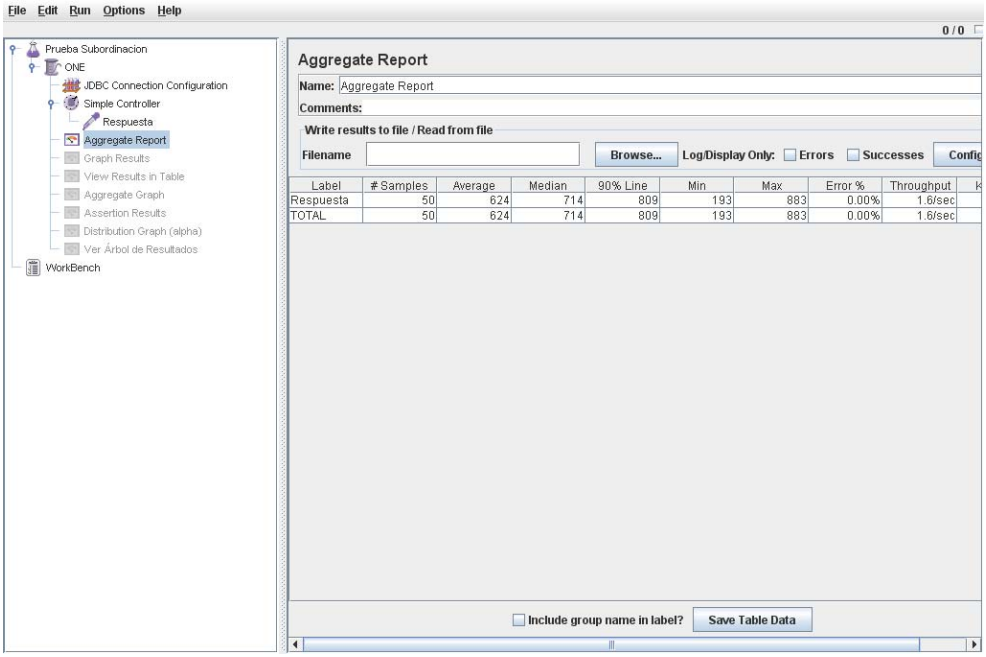

Figura 24 Prueba sobre la agregación Subordinación para 5 usuarios concurrentes

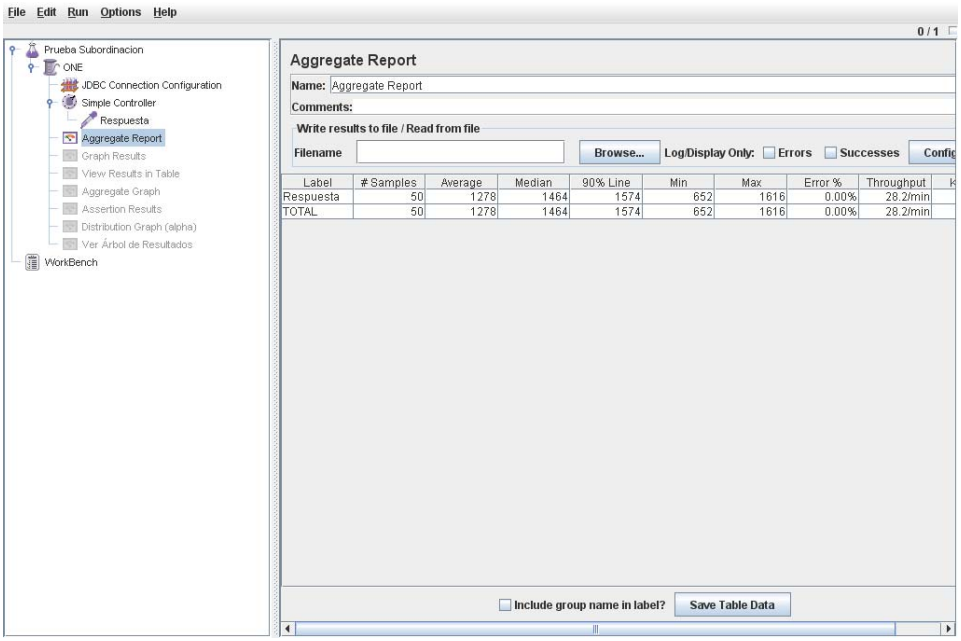

Figura 25 Prueba sobre la agregación Subordinación para 10 usuarios concurrentes

Anexo 8 Modelo Estadístico de Indicadores Generales

ONE OFICINA NACIONAL DE ESTADÍSTICAS	Sistema de Información de Estadística Nacional (SIEN)	INDICADORES GENERALES	MODELO No. 0005-10 Página 1 de 1 MENSUAL / TRIMESTRAL / ANUAL
INFORME ACUMULADO HASTA: Mes: __ Año: __		**UNIDAD DE MEDIDA:** Entero con un decimal	
Centro informante:		**Código del centro informante:**	

INDICADOR	UNIDAD DE MEDIDA	CÓDIGO	AÑO ACTUAL		AÑO ANTERIOR
			Plan	Real	
A	B	C	1	2	3
Indicadores de Producción e Ingresos					
Producción mercantil	MP	0100			
Ingreso turístico	MCUC	0200			
Ingreso de la Unidad Presupuestada (UP)	MP	0300			
Ventas netas de bienes y servicios	MP	0400			
De ellas: en divisa	MCUC	0410			
Del total: Producciones	MP	0420			
De ello: en divisa	MCUC	0421			
Ventas minoristas	MP	0430			
De ellas: en divisa	MCUC	0431			
Ventas de gastronomía	MP	0440			
De ellas: en divisa	MCUC	0441			
Ventas mayoristas	MP	0450			
De ellas: en divisa	MCUC	0451			
Ingresos por servicios a las personas	MP	0460			
De ellos: en divisa	MCUC	0461			
Ventas en comedores y merenderos	MP	1400			
Indicadores del sector externo					
Exportaciones de servicios	MCUC	1500			
Importaciones de servicios	MCUC	1600			
Indicadores de la fuerza de trabajo y los salarios					
Promedio de trabajadores total	U	1700			
Número de trabajadores al cierre del período	U	1800			
De ello: mujeres	U	1900			
Salarios y sueldos devengados	MP	2100			
De ellos: a mujeres	MP	2110			
Tiempo trabajado	Hombres- horas	2200			
Otros ingresos monetarios del trabajo en divisas	MCUC	2300			
Otros ingresos por distribución de utilidades	MP	2400			
Suma de control (página 1 de 1)		**9999**			

Certificamos que los datos contenidos en este modelo corresponden a los anotados en nuestros registros primarios y de acuerdo a las instrucciones vigentes para la elaboración del mismo.	VICEDIRECTOR ECONÓMICO Nombre y apellidos:_____ Firma:_____	DIRECTOR Nombre y apellidos:_____ Firma:_____	FECHA __ __ __ __ __ __ Día Mes Año

Figura 26 Modelo Estadístico de Indicadores Generales (0005)

Anexo 9 Acta de Aceptación de los Clientes Finales

AVAL

La Oficina Nacional de Estadísticas radicada en Paseo No. 60 e/ 3ra y 5ta, Vedado, Plaza de la Revolución, Ciudad de La Habana, Cuba, y en su nombre la Lic. Elena L. Fernández García con poderes suficientes para obligarle en este acto, según resulta de la verificación de la representación de la parte inferior de este documento.

AVALA

Al estudiante: Julio Ernesto Ortiz Sierra, miembro activo de la plantilla del Grupo de Desarrollo del **CENTALAD** y diplomante desarrollando la investigación titulada: Mercado de Datos para la Oficina Nacional de Estadísticas para optar por el título de Ingeniero en Ciencias Informáticas, por los resultados obtenidos con el diseño e implementación de un Mercados de Datos que integre la información perteneciente al rango de años 2000-2008 del Modelo Estadístico de Indicadores Generales (Modelo 0005) evidenciándose lo siguiente:

- Mejora apreciable en la integración de la información histórica necesaria para el análisis de la información.
- Mejora en rapidez y disponibilidad la preparación de reportes para el análisis estadístico.
- La solución integra las variables más relevantes del Modelo Estadístico en cuestión.

Además la solución fue presentada en un evento donde participaron los informáticos de las Oficinas Habaneras obteniendo resultados satisfactorios y la aceptación de los clientes finales con su diseño e implementación.

El presente aval estará en vigor hasta que el Centro de Tecnologías de Almacenamiento y Análisis de Datos (**CENTALAD**) o quien en su nombre sea habilitado para ello autorice su cancelación o devolución. Y a todos los efectos procedentes, se suscribe la presente, en un (1) ejemplar, en la Ciudad de La Habana a los 13 días del mes de Mayo del año 2009.

Directora de Informática de la ONE

Lic. Elena L. Fernández García

Anexo 10: 12 criterios definidos por E. F. Codd que deben cumplir los Sistemas OLAP.

En el año 1993, E. F. Codd en su artículo "Providing OLAP to User-Analysts: An IT Mandate" definió la tecnología OLAP haciendo uso de 12 reglas:

1. Vista conceptual multidimensional: Los analistas ven el negocio de manera dimensional por naturaleza. Los modelos de datos multidimensionales permiten a los usuarios una manipulación más simple e intuitiva de los datos, facilitando su filtrado ("slicing and dicing").

2. Transparencia: Debe ser transparente al usuario el hecho de que la herramienta OLAP forme parte de su hoja de trabajo habitual o de sus paquetes gráficos. OLAP debe formar parte de una arquitectura de sistemas abiertos, que pueda ser incluida en cualquier lugar que el usuario desee sin afectar la funcionalidad de la herramienta. Al usuario no se le debe presentar la fuente de datos suministrada a la herramienta OLAP, ya sea homogénea o heterogénea.

3. Accesibilidad: La herramienta OLAP debe ser capaz de aplicar su propia estructura lógica para acceder a fuentes de datos heterogéneas y realizar las conversiones necesarias para presentar una vista coherente al usuario. La herramienta (y no el usuario) debe saber de dónde vienen los datos físicos.

4. Desempeño constante ante el suministro de datos: El rendimiento de la herramienta OLAP no debe sufrir de manera significativa a medida que el número de dimensiones es aumentado.

5. Arquitectura Cliente-Servidor: El componente servidor de la herramienta OLAP debe ser lo suficientemente inteligente de forma que varios clientes puedan ser conectados con esfuerzo mínimo.

6. Dimensionalidad genérica: Todas las dimensiones de datos deben ser equivalentes en su estructura y posibilidades operacionales.

7. Manejo dinámico de matriz esparcida: La estructura física del servidor OLAP debe tener un manejo eficiente de la matriz esparcida.

8. Soporte multiusuario: las herramientas OLAP deben ofrecer acceso concurrente para la recuperación, integridad y seguridad.

9. Operaciones irrestrictas con dimensiones cruzadas: Las facilidades computacionales deben permitir el cálculo y la manipulación de datos a través de cualquier número de dimensiones sin restringir las relaciones entre las mismas.

10. Manipulación intuitiva de los datos: la manipulación de datos inherente a la vista multidimensional del negocio, como detallar y profundizar en diferentes niveles (drill-down) o generalizar y sacar conclusiones a niveles superiores (drill-up), debe hacerse de manera intuitiva y no requerir de demasiados pasos en la interfaz del usuario.

11. Suministro de información flexible: Las facilidades de reportes deben presentar la información en cualquier manera que el usuario la desee ver.

12. Niveles no limitados de dimensiones y agregaciones: El número de dimensiones soportadas debe ser ilimitado. Cada dimensión genérica debe permitir un número ilimitado de niveles de agregación definidos.