Universidad de las Ciencias Informáticas

Facultad 3

Título: Diseño e Implementación de un Almacén de Datos Operacionales para la Corporación CIMEX.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Darián González Ochoa

Tutor: Ing. Yonelbys Iznaga González

Ciudad de la Habana

Mayo de 2009

La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.

Aristóteles

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 _____.

_____	_____
Darián González Ochoa	Ing. Yonelbys Iznaga González
Autor	Tutor

AGRADECIMIENTOS

Resulta bastante difícil concentrar en tan solo una página a tantas personas que de una manera u otra han dado a mi vida un sentido. Aún así haré el noble intento de escribir.

En primer lugar quisiera agradecer a mis padres que me han apoyado en todo momento, y han estado a mi lado en todas y cada una de mis decisiones. A mi mami querida, que ha sido y será mi sostén, que ha sido mi sol, incluso en las noches más tormentosas, por su amor, mil gracias. A mi papá por su nobleza y entrega en cada una de las cosas que hace, por ser mi maestro y mi ejemplo, por recordarme que no existe dificultad alguna cuando a tu lado la familia está. Quiero agradecer al destino por dotarme de dos padres, un privilegio, un lujo, que no todos pueden llegar a comprender, por esto quiero agradecer a Pancho, por todos lo momentos, por inculcarme el estudio, el carácter y el respeto por las cosas que hago, por estar al tanto de cada paso que doy y por darme los ánimos para dar los siguientes.

Quisiera agradecer a mis hermanos Panchito e Iria, por simplemente existir, por inspirarme a que todos los días tengo que ser mejor como persona, para poder ser un ejemplo ante ustedes; mis hermanitos del alma.

A mis abuelos; dondequiera que estén, por la guía, el ejemplo y la confianza depositada en mí; gracias les doy. A mi querido abuelo Pepe por enseñarme que ser revolucionario no es una opción, y que es una manera de ser realmente útil. A mis tíos y primos; por demostrarme que la grandeza de la familia no está en la cantidad de sus miembros, sino en el tamaño de sus corazones, gracias por esta gigante familia que me han dado.

A Dai; por permitirme encontrarla y colocarme allí donde no se suele llegar, gracias por la compañía, por la espera o por tan solo callar, mil gracias por todo ese amor.

A mis amigos, a todos ustedes gracias por tenderme la mano cuando lo necesité, a Tico, a Eliober, a "La Embajada" por abrirme las puertas; a Julio, a Frank, a la flaca, a la negra de Songo; a todos gracias por ser lo que son.

No podía faltar Yonelbys, a usted tutor, gracias por la enseñanza y la confianza depositada.

Con admiración y respeto les agradezco a todos.

Dedico muy especialmente este trabajo a mi madre Norma Ochoa Rodríguez, quién por su constancia, dedicación, ternura y amor; soy, lo que he llegado a ser. Este también es el fruto de su quehacer. Te amo mamita.

Cada día son mayores las necesidades de información sobre la cual soportar el sistema de apoyo a la toma de decisiones. Los procesamientos analíticos que desean realizar la mayoría de los usuarios son prácticamente imposibles sobre los sistemas transaccionales, por lo cual, generalmente, se asume un entorno separado destinado a consultas. Los Almacenes de Datos Operacionales pretenden unificar la información operacional de la organización para tales propósitos. Sin embargo, el proceso de desarrollo merece ser formalizado debido a que son numerosas las decisiones a tomar en este largo camino. Por esto es nuestro objetivo principal proponer una formalización de los aspectos más relevantes a tener en cuenta que sirvan de soporte durante el diseño y posterior puesta en funcionamiento de un Almacén de Datos Operacionales. Para corroborar la validez de la propuesta realizada, nada mejor que aplicarla en la práctica, por lo cual durante el desarrollo de un Almacén de Datos Operacionales para la Corporación CIMEX[1], se aplica la metodología sugerida desde sus primeras etapas.

[1] Corporación Importadora y Exportadora.

ÍNDICE

FIGURAS

TABLAS

GRÁFICOS

INTRODUCCIÓN

En nuestros días, manejar correctamente la información y realizar análisis sobre ésta es una necesidad primaria en prácticamente cualquier ámbito de la vida social. Los avances tecnológicos ocurridos en las últimas décadas han facilitado la manipulación y almacenamiento de modo eficiente de grandes volúmenes de datos. Sin embargo, las necesidades crecientes de los propios usuarios, en su deseo de obtener la mayor riqueza posible a partir de la información acumulada, evidenciaron la incapacidad de realizar, en los entornos transaccionales, procesos analíticos de gran envergadura.

Esto provocó que surgiera una división en la línea del manejo de la información: por una parte quedaban los ambientes transaccionales, encargados de la entrada de datos; por otra, los destinados a análisis, especializados en la obtención y buen aprovechamiento de los mismos. Uno de los resultados más notorios en esta separación ha sido la concepción de una nueva arquitectura destinada a apoyar los procesos de toma de decisiones: los Almacenes de Datos. Con ellos se pretende concentrar y homogeneizar la información para brindar una visión global del comportamiento del negocio.

Sin embargo, existe otra capa que merece especial atención: los Almacenes de Datos Operacionales, con los cuales se pretende satisfacer las necesidades de información operacional en la organización. Estos entornos asimilan algunas de las cualidades de los Almacenes de Datos y las adaptan a sus objetivos dentro de la arquitectura de los sistemas de información. Existen diversas fuentes bibliográficas que abordan algunos de los detalles de mayor interés relacionados con este tipo de sistemas: su importancia, diseño e implementación. Sin embargo, los intentos por formalizar el proceso de desarrollo de un Almacén de Datos Operacionales son escasos.

Es por esto que se ha trazado como principal objetivo elaborar una propuesta de formalización del proceso de desarrollo de este entorno, para la cual se debe, en primer lugar, brindar una definición propia y precisar los objetivos del sistema dentro de la arquitectura, analizar posteriormente cada uno de los pasos a ejecutar durante su diseño, y ofrecer un conjunto de métodos recomendables a seguir para obtener mejores prácticas durante su desarrollo y puesta en funcionamiento.

Marco Teórico

Situación Problémica:

La Corporación Cimex consta de un sistema transaccional llamado Sentai, el mismo está presentando demoras a la hora de dar respuestas a consultas personalizadas por parte de los usuarios finales del negocio, se necesita un sistema que sea capaz de brindar información de datos de manera operacional y que nutra de información al Datawarehouse(DW), ya que el proceso de consultas operacionales sobre este Datawarehouse existente, se ralentiza, pues no fue concebido con tal motivo, además de esto, el nuevo sistema que se cree debe ser desarrollado con herramientas y tecnologías libres, llevando a cabo el proceso de independencia tecnológica en el proceso de informatización de la sociedad cubana.

Problema Científico:

Después de un análisis de la situación problémica, queda conformado el problema científico mediante el cuestionamiento de:

¿Cómo lograr la integración de datos que almacena la Corporación CIMEX en un ambiente operacional, de manera que facilite el análisis de la información y la toma de decisiones gerenciales?

Objeto de Estudio:

Almacenes de Datos

Campo de Acción:

Almacenes de Datos Operacionales

Objetivo General:

Desarrollar un Almacén de Datos Operacionales para la corporación CIMEX basado en herramientas de software libre.

Hipótesis:

Si se desarrolla un Almacén de Datos capaz de integrar los datos de la Corporación CIMEX en un ambiente operacional, esto facilitará el acceso a los mismos y una mejora en el proceso de toma de decisiones gerenciales.

Tareas de la Investigación:

1. Realizar un estudio del arte con el fin de caracterizar como se llevan a cabo los diseños e implementación de ODS en otros sistemas.
2. Definir la metodología a utilizar en el desarrollo del almacén de datos.
3. Realizar un estudio y selección de las herramientas Software Libre que brinden los servicios y las funcionalidades necesarias para montar un Almacén de Datos Operacionales.
4. Definir las dimensiones del almacén a partir de las especificaciones realizadas por DataCimex.
5. Construir el diseño del modelo multidimensional del almacén operacional de datos.
6. Realizar el esquema del diseño físico del almacén operacional de datos.
7. Implementar el modelo multidimensional del almacén operacional de datos.
8. Evaluar la velocidad de respuesta de los reportes y la validez de los datos del ODS de CIMEX.

Con tal propósito se ha estructurado el presente trabajo de la siguiente forma:

En el Capítulo 1, se presenta una breve reseña histórica sobre el surgimiento de los Almacenes de Datos como componente dentro de la arquitectura de los sistemas de información, la aparición del término Almacén Operacional de Datos, motivaciones de su surgimiento y características principales del mismo. Se realiza un reconocimiento de las principales tecnologías y herramientas involucradas en el mundo de los almacenes de datos, preferentemente las de corte open source, para dar cumplimiento así una de las necesidades imperantes en la Corporación.

Ya entrando en el Capítulo 2, se encuentra un guión sobre cómo se realizará la construcción del ODS, los detalles de su modelado e implementación, según la metodología tratada en el documento, se tratarán temas de vital importancia como las estrategias de indexado y optimización, ya que como el sistema está destinado al análisis se hace sumamente importante prestarle atención especial a la optimización de consultas.

Por último, en el Capítulo 3, en un intento por validar la propuesta teórica realizada, se comentan algunas de las experiencias al aplicar la metodología sugerida durante el desarrollo del ODS para CIMEX.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Este capítulo brindará un acercamiento a los principales sistemas de almacenamiento de datos, sus antecedentes y principales características, el surgimiento de Almacenes de Datos Operacionales (ODS), sus utilidades e importancia en el nuevo orden informático mundial, y cómo influyen estos en la toma de decisiones de las empresas que por necesidad de su funcionamiento trabajan con excesivas cantidades de datos. Se ofrecerá una panorámica general del uso y estudio de las principales herramientas de trabajo que se requieren para la puesta en marcha y el correcto funcionamiento de un ODS, así como una breve comparación y análisis para la selección del óptimo Gestor de Sistemas de Bases de Datos, dada las características enunciadas en la situación problémica anteriormente descrita en la introducción de este trabajo.

Desde inicios de los años 60 con la aparición de los primeros conceptos de Bases de Datos se escucha ya hablar de almacenamiento de datos en las empresas, hoy en día es cada vez mayor el cúmulo y el procesamiento de los datos que utiliza cualquier entidad por pequeña que esta sea. Surge la necesidad de manejar la información de manera eficiente, que facilite la operatividad, rapidez y efectividad de las decisiones que se tomen, esto es de manera tangible, importantes elementos para poder estar a la altura de la competencia empresarial. Procesos que hoy ocurren automáticamente, y que, seguramente se ven con total naturalidad, encierran por detrás una cadena de sucesos que desenlazan en la constante evolución de los sistemas de información.

Los Almacenes de Datos

Desde sus inicios, los sistemas de bases de datos se convirtieron en una herramienta fundamental de control y manejo de las operaciones comerciales. Fue así, como en unos pocos años en grandes empresas y negocios existía un considerable cúmulo de información, almacenada en diferentes fuentes de datos y estas ya habían alcanzado un tamaño excesivamente grande.

Con esta gran acumulación de información, los principales directivos de estas empresas y negocios se dieron cuenta que esta podría tener un fin útil, al estar reflejada la mayoría de sus operaciones comerciales durante los llamados ciclos de negocios, propios del mercado.

A su vez, los mercados empresariales han experimentado una metamorfosis radical. Las empresas demandan mayor rapidez y eficiencia en la entrega de productos, y mejora en todos los servicios que presta, por lo que se hace imprescindible encontrar formas más eficaces de distribuir los productos, más facilidades para hacer estudios de mercado basados en la información de las operaciones comerciales de las empresas y de sus clientes y, en definitiva, mayor rapidez y efectividad a la hora de tomar decisiones.

A medida que se hacía notable el incremento de los datos almacenados en los pequeños sistemas creados para dar solución a estas crecientes necesidades, se hizo excesivo, a la par de esto, también la necesidad de realizar consultas y extraer información guardada con mayor rapidez y efectividad, para decidir de manera estratégica que pasos debía seguir la empresa. Los sistemas creados hasta el momento fueron decayendo ante la creciente necesidad de realizar análisis exhaustivos de los datos, ya que estas operaciones se tornaban altamente costosas y atentaban contra el funcionamiento de estos sistemas.

Se fueron creando otros sistemas y programas que realizaban estas funciones, pero lo hacían de forma independiente, ya que extraían y analizaban parte de la información, donde esta era sometida a criterios de especialistas en determinada área de la empresa. Estos sistemas tomaron un gran auge por un tiempo, ya que separaba la información y cada usuario analizaba solo la parte de los datos conveniente a su departamento, esto traía consigo que al trasladar los datos en ambientes separados se afectaba los Sistemas Operacionales cuando se requería de estos, un análisis sobre gran cantidad de datos.

Con el tiempo esta ventaja de ver los datos por separados se convirtió en un gran problema, ya que se extraía información, de la que ya anteriormente se había extraído y así sucesivamente, por lo que se formaba una especie de “tela de araña” de la cuál resultaba difícil escapar, si esta, cada vez aumentaba más. Esta forma de concebir la extracción y análisis de la información se conoció como “Evolución Natural de la Arquitectura”(Inmon 2002)

Este sistema de almacenamiento de los datos, o por lo menos la arquitectura que él encierra trajo consigo problemas que no demoraron en darse a conocer, diferencias en la información de los datos, provocado a la hora de extraer los mismos, dado que este proceso de extracción se realizaba en varios niveles, y la información podía ser manipulada. Todo esto provocó falta de credibilidad en los datos. En conclusiones

acceder a la información en estas condiciones resultaba costoso, ineficaz y prolongado, todo esto sin contar con que se dificultaba el análisis histórico de los datos.

Esta forma de llamarle “evolución natural”, solo se hizo efectiva para resolver problemas específicos y puntuales.

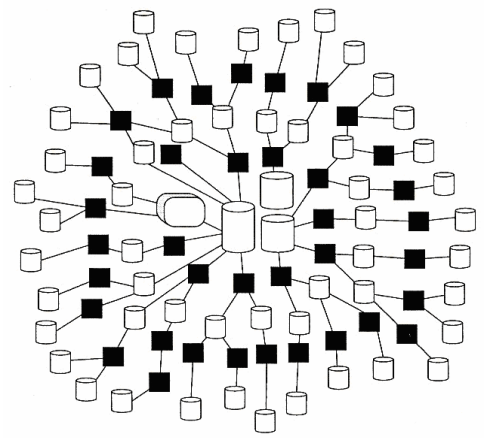

Figura 1: Evolución Natural de la Arquitectura

Por tanto, se pensó en lo ideal que 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[2].

Esta nueva arquitectura que fue surgiendo daba solución eficientemente a los problemas presentados por causa de la ramificación de los datos. Está basada fundamentalmente, en la unificación de la información a nivel central y está destinado al apoyo en el proceso de toma de decisiones de una empresa cualquiera que esta sea. Con esta nueva manera de concebir el almacenamiento de los datos se persigue, a pesar de las incompatibilidades que puedan existir entre diferentes sistemas y sus contenidos, unificar la información y transformarla, haciéndola claramente legible y que pueda llegar al usuario un conocimiento general de los procesos que ocurren en su entrono de trabajo. La misma intenta garantizar rápidas

[2] El término más difundido para Almacén de Datos es *Datawarehouse* (DW), del inglés.

respuestas de las demandas que impliquen reportes de grandes cantidades de datos, se asegura además conservar una visión histórica de los datos, posibilitando esto que se realice un profundo análisis de las decisiones tomadas en el pasado y la influencia de las mismas en el ulterior negocio de la empresa.

Sin embargo, para hacer un uso eficiente de la información histórica almacenada en un Almacén de Datos para la ayuda a la toma de decisiones, era vital garantizar que estos datos fueran fáciles de obtener, estandarizados y confiables.

Varias personas en el mundo entero han investigado sobre las principales potencialidades que brindan los almacenes de datos y en su particularidad más conocida, los Datawarehouse. Los primeros conceptos de de almacenes de datos aparecen a inicios de la década del 90´, cuando Bill Inmon[3], lo define de la siguiente manera: "*Un Datawarehouse es una colección de datos* ***orientado a sujeto, integrado, variante en el tiempo y no volátil*** *para ayudar al proceso de toma de decisiones gerenciales"*(Inmon 2002)

Obviamente esta definición de Inmon, ya clásica, se puede tomar como una definición "pura", teniendo en cuenta la constante evolución en este campo. Actualmente, este concepto propiciado por quien es considerado uno de los padres fundadores de la arquitectura de almacenes de datos, ha sido manejado en dependencia de las necesidades y capacidades del mercado, esto conlleva a que se originen otros conceptos como el de Data Mart, que surge como solución a la imposibilidad de almacenar toda la información histórica que brindan los datos y lo que realmente hace es que almacena una foto de un período determinado de tiempo.

Ralph Kimball[4] define Almacén de Datos de una forma más práctica y simple, pero igual de importante, un Datawarehouse es: *"Una copia de los* ***datos transaccionales, específicamente estructurados*** *para consultas y análisis".*(Kimball and Ross 2002)

Se puede decir que un Datawarehouse es una base de datos, orientada al análisis de la información histórica contenida en ella. Dependiendo de las necesidades de análisis de la organización, puede

[3] **William Harvey Inmon** (1945), experto reconocido mundialmente, es el creador de la llamada *Corporate Information Factory*.

[4] **Ralph Kimball**, Doctor en Filosofía, ha sido uno de los mayores visionarios en la industria del Almacén de Datos desde 1982, actualmente, reconocido conferencista, consultante y profesor.

almacenarse desde unos meses hasta varios años la información. El modelo que soporta la información que contiene, se encuentra diseñado, estructurado e implementado con la finalidad y propósito del análisis y navegación de los datos.

Los Almacenes de Datos Operacionales (ODS)

Los Sistemas Datawarehouse, están enfocados fundamentalmente a ofrecer una panorámica integral de los datos, que sostengan las funciones generales de la gerencia a la hora de la toma de decisiones. A estos niveles, generalmente, las solicitudes de reportes van encaminadas a ver la información agrupada por diferentes criterios, donde se pueda observar el comportamiento general que siguen ciertos y determinados datos, pero solo eso, de forma general, no hechos detallados.

En diversas ocasiones, cuando se está realizando el proceso de carga hacia los Almacenes de Datos, se puede manejar de una forma u otra, cierto nivel de atomicidad de los mismos, pero a medida que este proceso avanza, y se incrementa el nivel de almacenamiento, se pierden los detalles de los datos que están siendo cargados. Como es lógico, pues la arquitectura de los Almacenes de Datos, lo describe de esta manera y para ello está concebida.

Surge entonces, cierta necesidad, pues la práctica ha confirmado que en muchas circunstancias ha resultado necesario realizar un análisis exhaustivo a un nivel altamente detallado como son el caso de las transacciones. ¿Entre tal dilema, entonces, qué hacer?, es evidente que no se podría buscar en los sistemas transaccionales pues esto atentaría contra el buen funcionamiento de los sistemas operacionales, ya que estos deben estar libre de la mayor cantidad de procesamientos analíticos fuertes. Tampoco sería recomendable torcer nuestra mirada hacia la antagónica frase “que cada cual extraiga lo que necesite”, pues esto sería entonces retroceder al pasado y sería tangible el desastroso resultado, ya demostrado en el epígrafe anterior. Y por último, la solución tampoco cabría en alterar la arquitectura del Almacén, para que contuviese datos operacionales, esto sería alterar por completo la arquitectura general de los Sistemas de Información.

Un Almacén de Datos Operacionales[5] es "*una de las arquitecturas más dominantes que se pueden encontrar hoy en día en los sistemas de información*"(Inmon 1998). Surge como respuesta a las necesidades de contar con un sistema integrador de datos que brinde la información con un alto nivel de detalle operacional. ¿Pero, realmente que es un ODS?

Muchos estudiosos del tema se han referido a estas cuestiones aunque algunos otros no han querido reconocer al ODS como una estructura existente.

Dos personas, las cuales se consideran padres fundadores en las materias de los Almacenes de Datos, han establecido, cada uno de ellos y por separado definiciones de conceptos, sería muy oportuno citar las mismas:

- Según **Ralph Kimball**, un ODS es un almacén de información detallada orientado a temas, integrado, aumentado con frecuencia, dentro del Almacén de Datos de una empresa.(Kimball 1997)
- De acuerdo a lo planteado por **William H. Inmon,** un ODS es una colección de datos orientada a temas, integrada, volátil, actualizada, sólo detallada, que sustenta las necesidades de información reciente, operacional, integrada y colectiva de la organización.(Inmon 1995, Febrero)

Estas dos definiciones son a grandes rasgos como dos aristas, pero que convergen en un punto en común, ¿por qué razón?, pues ambos coinciden en que estos sistemas de información son contenedores de datos orientados a temas específicos, lo que proporciona una visión global del negocio donde se esté trabajando; y en ambas la información se encuentra en un nivel detallado. Pero hay otros aspectos que se deben considerar.

A simple vista después de haber leído las definiciones dadas por estos científicos se puede creer que en determinado momento divergen en cuanto a la permanencia de los datos en el sistema; es decir en cuanto a lo que almacenamiento histórico se refiere. Inmon por su parte ofrece un carácter de mayor volatilidad a los datos almacenados, pareciendo que le temiese al incremento de datos con ese alto nivel de detalle,

[5] Conocido en la mayoría de las literaturas referentes al tema como **O**perational **D**ata **S**tore (ODS), del ingles.

que es básicamente lo que propone Kimball. Pero es relevante destacar que hoy en día es cada vez más necesario realizar análisis de información cada vez más detallada y de un período histórico especifico según lo requiera el negocio de la empresa o entidad en cuestión; esto evidentemente traerá consigo una mejor organización a la hora de realizar una observación de datos operacionales históricos.

Ahora solo cabría preguntarse si las transacciones podrían analizarse estando al nivel de detalle que se describe en el ODS. Quizá, hace algún tiempo atrás, esto no sería posible; pues el soporte tecnológico no lo permitiría y entonces todo indicaba a la tendencia de ir disminuyendo en cuantía de detalles y atomicidad de datos. Hoy sucede lo contrario ya que potencialmente existen los soportes de hardware y software que permitan la incorporación y almacenamiento por períodos históricos de datos altamente granulados.

Desde otro ángulo de análisis, no se podría obviar totalmente lo planteado por Inmon, pues esto pudiera interpretarse como: ¿con qué período de tiempo es factible el almacenamiento de estos datos?, ¿es necesario conservarlo todo? Se podría claramente centrar en los ambientes operacionales, donde por lo general no hace falta retener los datos por períodos históricos prolongados, y es aquí, tal vez una de las interpretaciones fundamentales que se le pudiera dar a la definición de Bill Inmon cuando hablaba de volatilidad de datos. Por lo que ambas variantes tanto Kimball como Inmon son favorables a la hora de concebir el ODS, o por lo menos en esto de la conservación de los datos.

Estado Actual de los ODS

Contradictoriamente, desde que aparecieron los primeros conceptos de Almacén de Datos y Almacén Operacional de Datos en la década de los 90 y más específicamente en el año 1992, cuando referente a esto Bill Inmon proyecta uno de sus principales artículos, no se realizó la proyección práctica del tema, pues estos aparecen gradualmente a mediados de los 90´ y principios del nuevo milenio. Grandes empresas manejadoras de datos han implementado ODS como complemento de sus propios Datawarehouse. Algunas de las más importantes de acuerdo a la gran cantidad de recursos que manejan son: la gigante de circuitos electrónicos, dedicados a la telecomunicación AT&T, quién utiliza el ODS como soporte a la toma de decisiones de situaciones operacionales que se puedan presentar como parte del negocio.

En nuestro propio patio América Latina muchas empresas dedicadas a los medios de comunicación, han optado por la construcción de un Almacén de Datos Operacionales que nutra a sus propios Data Warehouses, tales como TV Azteca, Wal-Mart, Visa, Telefónica de Argentina, Ipostel, GNP, Baxter entre otras. En países desarrollados como Estados Unidos y Canadá, las compañías American Stores, Canadian Tyre, Owens Corning Glass, han obtenido resultados tangibles en la implementación de este tipo de estructura de almacenamiento de datos.

Transnacionales como la Mc Donald, la Samsung, HTL, Nike, Coca Cola, Pilsen, entre otras, han interiorizado en la importancia de cómo realizar trabajo sobre datos históricos y operacionales. La mayoría de los Bancos Mundiales hoy implementan este tipo de tecnología de la información, tal es el caso de Banco de Argentina, Banfoandes, Banorte, Banco de Venezuela, Banco Nacional de España, Caja Extremadura.

En Cuba, producto a factores económicos que influyen directamente sobre las cuestiones de desarrollo de las tecnologías no se ha visto un sustancial avance en este tema específico. Muchas entidades que manejan volúmenes cuantiosos de datos ha implementado su propio almacén de datos, sin comportarse de la misma manera con los almacenes operacionales de datos, estos últimos no están concebidos dentro de la arquitectura de los sistemas de información. Es por ello que sólo se conoce por fuentes documentables que la Corporación Cimex sea la única que trabaje con un ODS en su negocio interno. Resultando la utilización del mismo de una utilidad impresionante a la hora de realizar el proceso de toma de decisiones de la empresa.

Características de los ODS.

Frecuentemente se puede observar en muchas literaturas referentes al tema, que los sistemas de información están concebidos de *forma piramidal*[6], esta está dividida por varios niveles y cado uno de ellos con sus propósitos bien definidos y argumentados. En la base de esta pirámide se encuentran los datos de forma bruta, es decir sin ningún tipo de tratamiento, los mismos son manejados por los Sistemas

[6] También conocida como **"Pirámide del Conocimiento"**, la jerarquía de Dato, Información, Conocimiento, Sabiduría o Inteligencia (*Data Information Knowledge Wisdom* o DIKW) tiene sus orígenes, en el campo de la gestión del conocimiento, por el año 1987, cuando Milan Zeleny la describe en su artículo "*Management Support Systems: Towards Integrated Knowledge Management*"

Transaccionales (OLTP)[7], que como principal funcionamiento de estos, se encuentra la correcta entrada de los datos a los sistemas de almacenamiento. Para no atentar contra el funcionamiento de estos sistemas, en ellos se deben realizar consultas bastante puntuales y especificas, no procesos de análisis. En un nivel superior se encuentra la información, que no son más que los datos que fueron transformados y unificados para lograr una mejor comprensión de estos. Y por último, ya casi en la cúspide de la pirámide se encuentra en conocimiento que no es más que aquello que se extrae de la información y que una vez acumulado este se convierte en sabiduría o inteligencia. ¿Y el ODS, donde se podría ubicar?

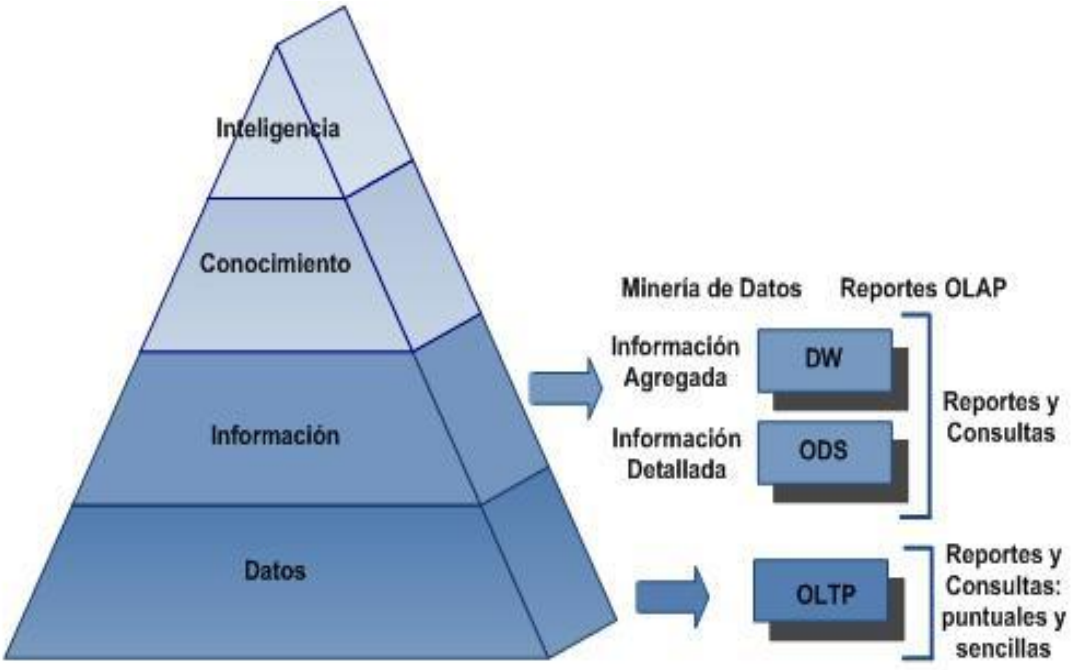

Figura 2: Ubicación del ODS dentro de la Pirámide del Conocimiento

Se tiene que definir y tener en cuenta como considerar al ODS, si como parte de un Almacén de Datos en general o de manera independiente a este. Se hace tentadora la propuesta que una vez integrado los datos en el ODS se pudiera levantar fácilmente un DW, este punto de vista puede estar propiciado por la

[7] Procesamiento de Transacciones en Línea, en la mayoría de las literaturas se trata como OLTP de sus siglas en inglés(**O**n**L**ine **T**ransaction **P**rocessing)

idea de constar de información detallada en el ODS y mediante procesos de extracción, cargarlas hacia el Almacén de Datos. Esto último depende en gran medida de los negocios propios de la empresa, pues en ocasiones solo se quiere que el ODS contenga información detallada de parte del negocio que se esté tratando y no de una totalidad, pero esto no significa que no se valore la real importancia del ODS como intermediario entre dos ambientes totalmente diferentes. *Ver Figura 3*

Tal pareciera que no existen diferencias entre un ODS y un DW, pues coinciden entre ellos; la estructura y los datos almacenados en los mismos, ya que están orientados por tema, homogenizados y se logran integrar[8]. Entonces se podrá resumir que la principal característica que los distingue es el tipo de consulta que sobre ellos se realice. En un DW, por lo general, se efectúan análisis de tendencias para comprender mejor comportamiento del negocio, de manera global e histórica; una tendencia no se puede definir con la recopilación de los datos que se brindan de unos días, es un proceso más abarcador, en el cuál se necesita información incluso de años. Sin embargo en un ODS el análisis que se realiza es operacional, de forma más detallada, de último momento, y como se señaló anteriormente, esta información posee un período de vigencia que resulta interesante para el analista. Esto marca de cierta manera la diferencia de los datos almacenados en estos entornos, en uno datos operacionales, en otro, datos informativos.

Esta diferencia de objetivos, ocasiona que se cree una diferencia de agregación en cada sistema, en el ODS se encontrará información atómica e indivisible con un alto nivel de detalle, mientras que en el Almacén de Datos se mantendrán los datos agregados.

Los diferentes tipos de análisis que se realicen sobre el ODS y el DW como sistemas contenedores de la información, determinan en la mayoría de los casos que la frecuencia de actualización de los datos en el ODS debe ser mayor o en el peor de los casos igual a la del Almacén de Datos, ya que el primero debe contener las últimas modificaciones ocurridas en los sistemas transaccionales (OLTP).

De igual forma resulta el comportamiento de estos sistemas ante las actualizaciones, pues no es conveniente realizar análisis con el objetivo de búsqueda de tendencias del negocio sobre un ODS ya que resultaría una pérdida total de tiempo, pues como se explicó anteriormente las tendencias surgen con el

[8] Ver Anexo 1. Tabla comparativa entre ODS y DW.

cúmulo de la información en el tiempo, los datos no operacionales en el ODS por lo general son sobrescritos.

Otra diferencia sustancial es la probabilidad de acceso de los diferentes usuarios al ODS y el DW. En cualquiera de los dos sistemas pueden hacer consultas tanto personas como otros sistemas propiamente dicho; las personas con el objetivo de intentar descubrir, a simple vista, comportamientos en el negocio que los conlleve a tomar cierta y determinada decisión; los sistemas por su parte accederán al usar un subconjunto de la información, reestructurarla a conveniencia y servir de soporte a los analistas en el proceso de la toma de decisiones. Debido que la información contenida en el ODS es de gran riqueza y de fácil restructuración es más frecuente el acceso de otros sistemas al ODS, mientras que los analistas querrán acceder al DW, pues les hace falta tener una visión global y utilizar datos totalizados, y es en este ambiente donde podrán lograr esto recorriendo de lo más general hasta el nivel de detalle deseado.

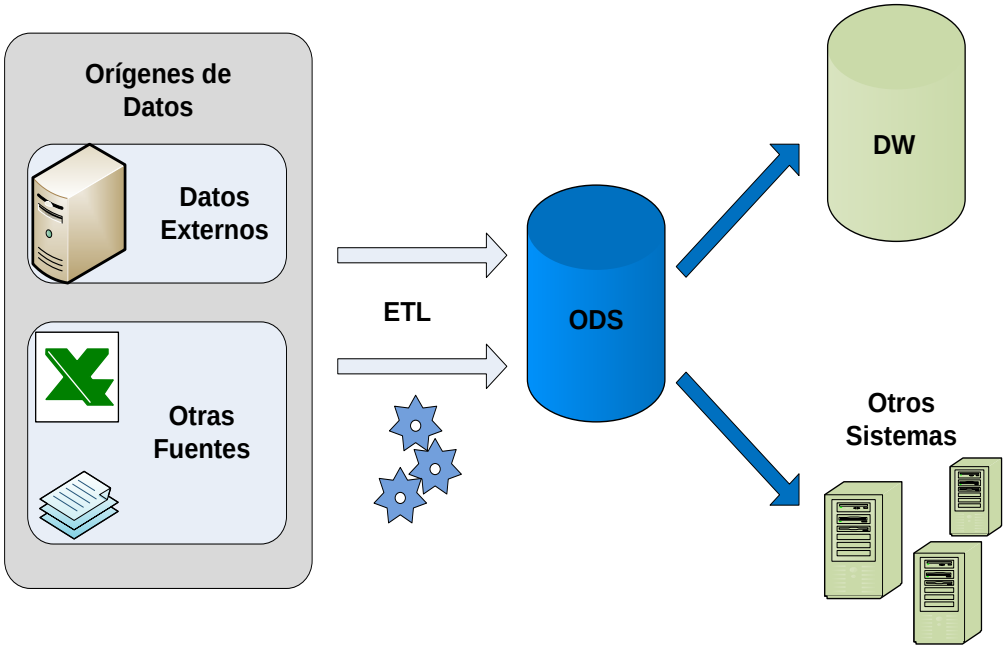

Figura 3: Flujo de datos en el Almacén de Datos Operacionales. Propuesta de Arquitectura

Las características y objetivos de cada entorno están bien delimitados, y no es intención de uno sustituir las funcionalidades del otro, sino que cada cual tiene su finalidad bien definida dentro de la arquitectura de los sistemas de información.

Múltiples responsabilidades del ODS

Como se ha visto, un ODS tiene otras responsabilidades además de mantener la información actualizada mediante procesos de carga periódicamente. Administrar los metadatos, realizar cuadres de datos y responder eficientemente ante las solicitudes son algunas de las tareas que se suman a la lista. A esta se pueden añadir otras, como pueden ser la creación automática de salvas por cuestiones de seguridad, purgas de datos que no se consideren dentro del período de vigencia válido para el analista y actualización de tablas de agregación, con las que se puede almacenar el resultado de consultas frecuentes, usadas para aumentar la eficiencia en el sistema. Debe estar preparado, por tanto, para los procesos con diversos propósitos que se ejecutarán sobre él, pero muchas veces sus múltiples responsabilidades deben ocurrir en cortos intervalos de tiempo, en el mejor de los casos, en un mismo día. ¿Cómo dividir entonces 24 horas entre tantas tareas?

Su flujo de trabajo deberá ser dividido en varias fases a lo largo de las 24 horas del día. Los límites entre estas fases o intervalos se determinarán de acuerdo al momento del día en que resulte más estratégico poner en funcionamiento los diferentes procedimientos. Generalmente, durante la madrugada se realizarán procesos de carga para realizar actualizaciones. Es recomendable realizar dichas cargas en horarios en los que el tráfico de datos en el nivel transaccional sea relativamente bajo. De esta forma se evita que su rendimiento se vea afectado por la coincidencia de procesos que requieren gran cantidad de recursos en el momento de su ejecución. También, a estas horas, pueden ejecutarse otros procesos, como la preparación de las tablas de agregación correspondientes a los datos recientes, cuadre de datos y administración de los metadatos asociados a las cargas.

Un detalle a tener en cuenta, es que otros sistemas que soportan la toma de decisiones pueden solicitar información al ODS durante la madrugada, razón por la cual el proceso de actualización de nuestro entorno debe comenzar y terminar antes que las solicitudes externas ocurran. Se deberá planificar, por tanto, el horario en que inicia la carga, así como tener un estimado de la duración de la misma e informarla a otros desarrolladores para evitar solapamientos. Una vez comenzado el horario regular de trabajo, los datos están listos para la realización de un fuerte procesamiento analítico. Pero se debe tener en cuenta que, de acuerdo a la frecuencia de actualización que se elija, pueden ocurrir, conjuntamente, procesos de carga con las modificaciones ocurridas en el entorno transaccional.

El ODS debe incorporar tanto la capacidad de realizar procesamiento analítico de alto rendimiento, como manejar los procesos de actualización y administración del sistema. Por esto se sugiere:

1.A

Planificar concienzudamente el "horario de trabajo de 24 horas" del ODS, así como realizar una correcta elección de las herramientas que permitan el buen desempeño de sus múltiples funciones.

Actualización de los ODS. Clasificación

El ODS puede asumir diferentes frecuencias para actualizar su contenido, y esto puede determinar, muchas veces, los métodos que se utilicen para el tránsito de la información desde los sistemas transaccionales, así como el nivel de integración y transformación de los datos. Las diferentes categorías o clases asociadas a dichas frecuencias han surgido como resultado del propio proceso evolutivo de los ODS.

Haciendo un poco de historia, a inicios de la década del 90, los sistemas ODS que fueron desarrollados eran utilizados como una herramienta de reportes con propósitos administrativos. Se actualizaban diariamente y ofrecían resúmenes sobre las transacciones empresariales del día. Este tipo de sistemas recibe hoy el nombre de ODS de Clase III. Posteriormente, con el aumento de las necesidades empresariales, el ODS evolucionó para convertirse en lo que actualmente se conoce como ODS de Clase II, siendo capaz de manejar información más compleja, integrar de múltiples fuentes y actualizarse más frecuentemente, por ejemplo, cada una hora. Los ODS de Clase I surgieron con la llegada de los sistemas de Administración de Relaciones con el Cliente[9]. Los CRM requerían la creación de un ODS enfocado a los clientes, con actualizaciones sincrónicas o casi sincrónicas con los sistemas transaccionales, de tal manera que se pudiera ofrecer información consistente y organizada inmediatamente después de la ocurrencia de cambios.(Imhoff 2000, Julio)

[9] El término Administración de Relaciones con el Cliente (*Customer Relationship Management* o CRM) es usado para describir herramientas de personalización sofisticadas, desarrolladas por algunos distribuidores para definir grupos de clientes y clasificarlos correctamente de acuerdo a los productos y servicios que se le ofrecen.

Para el ODS de Clase I, como ya se ha visto, la actualización es sincrónica o casi sincrónica. Los cambios aparecen a los 2 ó 3 segundos después de ocurridos en los sistemas fuente. Un ODS con actualización instantánea es difícil de implementar, por lo que sólo debe ser creado cuando la tecnología de que se dispone lo permita.

Generalmente, el intercambio de información se realiza a través de la transmisión de mensajes por capas intermedias[10]. Raras veces se utilizan herramientas de Extracción, Transformación y Carga[11] debido a que las transformaciones sobre los datos tienden a ser escasas. El modelo de datos usualmente es parecido en algunos aspectos al de los sistemas fuente. Aún así, la integración, por lo general, es poca. En esta clase de ODS, las sumarizaciones instantáneas pueden ser difíciles de realizar, por lo que se recomienda efectuarlas en diferentes intervalos durante el transcurso del día, por ejemplo, cada una hora. Los ODS de Clase I utilizan muchos recursos, son considerables los gastos en su mantenimiento y en sus inicios es muy difícil su sincronización. En resumen, desarrollar un sistema así puede resultar extremadamente costoso.

En los ODS de Clase II las actualizaciones pueden ocurrir varias veces al día. Los intervalos entre una carga y otra pueden ir desde 15 minutos a varias horas, por lo que no tiene la inmediatez de un ODS de Clase I. El método más utilizado para el traspaso es el de almacenar los datos y enviarlos más tarde (técnica conocida como store and forward), donde los cambios, por lo general, se escriben a un fichero y luego son cargados al sistema. Se puede realizar algo de integración y transformación mientras los datos son cargados, debido a que los requerimientos de actualidad de la información son menos estrictos. Por esto se sugiere utilizar herramientas ETL para este proceso. Pueden realizarse sumarizaciones de manera instantánea, pero se recomienda hacerlas sólo una vez al día, por ejemplo, en la madrugada. Los ODS de Clase II son más fáciles de desarrollar que los de Clase I y, por tanto, menos costosos. Su estructura es relativamente sencilla y requiere de menos recursos en línea para la carga.

---

[10] Conocido también como *messaging middleware*

[11] El término más difundido para Extracción, Transformación y Carga es del inglés *Extraction Transformation and Load* (ETL)

El ODS de Clase III es el caso de menor sincronía. El movimiento de datos hacia el sistema se realiza una sola vez al día, generalmente durante la noche. Para el traspaso se utiliza comúnmente el método store and forward descrito, realizando la carga por lotes. Son posibles una mayor integración y transformaciones complejas sobre los datos, por lo que las herramientas ETL son ampliamente usadas. Las sumarizaciones para reportes son realizadas una vez al día. Los ODS de Clase III son los más fáciles de desarrollar y de mantener, por lo que es recomendable comenzar con esta categoría e ir incrementando la frecuencia a medida que el sistema evolucione.

Otra categoría más reciente para los ODS es la Clase IV, donde la información se recibe desde el Almacén de Datos, realizando un proceso de retroalimentación para chequear el estado de la información actual y completar análisis tácticos.(Imhoff 2000, Julio)

El movimiento de datos, puede realizarse en intervalos regulares o irregulares. Cualquiera de las clases I, II, ó III pueden convertirse en una Clase IV. El requerimiento es, por supuesto, que debe existir un Almacén de Datos antes de asumir esta categoría. El proceso de traspaso de información de un entorno a otro se puede realizar de manera sencilla, haciendo uso, por ejemplo, de herramientas ETL.

De las categorías vistas, la I es un caso que se encuentra muy pocas veces, debido a que su costo tecnológico y operacional es mucho más elevado que en las otras variantes. Las clases II y III son las más comúnmente usadas, ya que pueden mantenerse con tecnología estándar.

Más allá de regirse por una clase u otra, una opción aún más rica es la combinación de varias categorías. La unión de clases en un entorno puede dar como resultado una estructura muy poderosa, que puede satisfacer los más difíciles requerimientos de los usuarios en cuanto a frecuencia de actualización. Esto es factible si se pueden identificar en el negocio procesos que ocurran de forma diferente, es decir, si es posible otorgar categorías de “volatilidad” a la información vinculada a distintos temas. Pero se debe recalcar que tan poderosa puede ser esta opción como difícil de desarrollar y de mantener.

Se debe tener en cuenta, sin embargo, a la hora de elegir una frecuencia de actualización u otra, el tiempo que demoran tanto los sistemas transaccionales en entregar los datos, como el que demora el propio ODS en realizar la carga, pues no es posible comenzar una carga si los datos aún no están listos o si la carga anterior no ha terminado.

Se puede concluir que además de las estrategias y necesidades de renovación de la información que tenga la organización, de igual manera otros factores van a determinar asumir una frecuencia u otra, como las posibilidades económicas de la empresa para brindar el respaldo tecnológico y de mantenimiento que implica cada clase de ODS, así como el tiempo que demoran los sistemas operacionales en generar los datos a cargar, y la complejidad del procesamiento que el ODS realice sobre la información durante la carga. Se debe tener en cuenta que a cada clase le corresponden distintas tecnologías y tienen propósitos diferentes[12]. Es recomendable, por tanto:

1.B

Definir una frecuencia de actualización que se pueda asumir en la empresa. Preferentemente, comenzar desarrollando un ODS de Clase III y luego ir variando su frecuencia o combinando varias categorías de acuerdo a la volatilidad de cada proceso.

Aceptación de la Metodología a utilizar.

Para el diseño y la construcción de cualquier empresa que se propondrá, se debe seguir un grupo de pasos, que más que simples pasos podría llamárseles procesos que abarcan una metodología confiable por la cual se guiará para levantar desde la arcilla la meta propuesta. En el mundo del diseño e implementación de los almacenes de datos, se pueden encontrar estas metodologías respaldadas cada una de ellas por todo el arsenal de quien las creara y las pusiera en práctica.

El diseño de Datawarehouse en el mundo entero, hoy, ha adquirido un nivel de madurez mayor que el de hace algún tiempo atrás, cuando no se pensaba siquiera obtener el resultado que tiene hoy gracias a las buenas prácticas de diseño de almacenes de datos y cuanto ha beneficiado esto a las empresas.

En las tecnologías data warehousing se puede evidenciar dos grandes tendencias a nivel mundial, ambas sustentadas en las manos de sus principales creadores, la primera conocida como metodología de Inmon y la segunda, conocida como metodología Kimball, ambas con un sin número de seguidores que han

[12] Ver Anexo 2 tabla resumen de las principales características de cada Clase de ODS.

concebido que el hecho de escoger una de las dos, sea, más que cualquier decisión de especialistas, una cruzada de orgullo personal.

Inmon es considerado el padre de los almacenes de datos, pues fue quien primero introdujo el término, este, trata la construcción de los almacenes con un enfoque descendente (top-down) donde los pequeños almacenes departamentales (Data Mart) se nutrirán del DW, donde se encuentren los datos de forma consistente e histórica. Por su parte resulta tentadora la propuesta que ofrece Kimball quién es considerado un *gurú* en el tema de tecnologías de almacenamiento de datos, cuando de interesante manera separa la Inteligencia del Negocio (BI)[13] entre el hecho y sus dimensiones, esto de manera general es muy eficaz y conduce a respuestas en muy pequeñas cantidades de tiempo.

La Corporación CIMEX, en particular característica de manejar grandes volúmenes de datos, realización de análisis de tendencias del negocio, superación y optimización del proceso de toma de decisiones, conlleva a la utilización de la metodología Kimball, ya que se hace necesario, dada la confiabilidad por su madurez, experiencia existente en la empresa en la utilización de la misma, robustez y documentación necesaria y suficiente sobre ella. Además que su creador, Ralph Kimball es una personalidad en el mundo del desarrollo de los almacenes de datos, el mismo es Doctor en Ciencias Filosóficas y posee trabajos relacionados con el tema en alrededor de un centenar de artículos, por lo que se infiere la gran cantidad de bibliografía existente, referente a su metodología. Propone con claridad cada actividad en cada uno de los períodos de construcción que deben realizar los roles, involucrados en el proyecto. La forma de almacenar la información es de fácil entendimiento para los usuarios finales, lo que permite mayor comprensión en la realización del análisis de los datos que se encuentran integrados y detallados. Es una metodología dúctil, es decir que puede resistir y adaptarse a los cambios.

Modelo Conceptual de Datos.

Ahora bien, ¿qué Modelo Conceptual de Datos se podrá usar para reflejar el negocio de forma intuitiva y qué Modelo Lógico de Datos respaldará, desde el punto de vista computacional, la necesidad de eficiencia?

[13] **B**usiness **I**ntelligence, del ingles.

El Modelo Entidad – Relación Extendido (MER/X)[14] es uno de los Modelos Conceptuales de Datos más difundidos en el mundo. Está enfocado a determinar las relaciones entre las entidades practicantes del negocio en cuestión. Este Modelo Relacional es uno de los modelos mayormente utilizados en el mundo, por su extensa riqueza teórica, logra reflejar de manera formal la descripción hecha por el MER/X. Entonces aparece aquí, la noción de normalizar, basado en teorías de Bases de Datos, la misma impide o reduce de manera eficiente la redundancia de los datos tratando los mismos como tablas de manera independiente. Esto provoca que el modelado del MER/X se convierta en una especie de laberinto y resulte incomprensible para el usuario, siendo poco natural y sencillo para el posterior análisis. Al tratar los datos como tablas independientes, a la hora de realizar consultas a los sistemas de bases de datos se genera una intensa actividad de *joins* para enlazar toda la información requerida, por lo que ralentiza el proceso y lo hace demasiado costoso. Por lo tanto el MER/X no se adecua correctamente a las necesidades de construcción de un Almacén de Datos Operacionales.(Hoobs 2005)

Se hace necesario organizar de una mejor manera o más intuitiva los datos y reducir considerablemente los costos por concepto de normalización. Podría especularse que si se reducen las normalizaciones se podría caer en errores tales como la redundancia de los datos, y de esta forma se introduciría al almacén datos inconsistentes. Esto no debe ser obstáculo alguno ya que nuestro ODS se va alimentar de sistemas transaccionales de datos, lo que posibilita que no sean necesarios los chequeos de consistencia e integridad de los datos, ya que estos últimos garantizan eso. Si se seleccionara un modelo que no realizará tantas normalizaciones entonces se apartarían del camino, cuantiosos y costosos *joins,* se ganará en eficiencia y simplicidad, por lo que la redundancia no sería un inconveniente sino una valiosa ventaja.

El Modelo Dimensional que ha ganado gran popularidad en el mundo de los Almacenes de Datos en los últimos años, es un candidato que requiere especial atención. Como bien lo indica su nombre, este modelo permite realizar un modelado y representar las estructuras de manera multidimensional, siendo esto consecuente con las especificaciones y exigencias del negocio. A diferencia del MER/X, identifica de

[14] El **MER/X** combina la propuesta inicial de P. Chen de su modelo Entidad – Relación con conceptos de modelación orientada a objetos.

manera correcta los procesos generales y luego analiza las particularidades de cada elemento en cuestión, es decir, va de lo general a lo particular (top - down).

Para la materialización de este modelo se utiliza la propuesta de Ralph Kimball llamada “esquema de estrella” que consiste a grandes rasgos, en una tabla central denominada “tabla de hechos” y un conjunto de pequeñas tablas llamadas “dimensiones” que se relacionan con esta tabla central se les denomina esquema de estrella por su similitud a una estrella natural. *Ver Figura 1.3*

Existen otras estructuras de modelos que surgen como modificaciones a este esquema de estrella. En sentido se puede encontrar al esquema de “copo de nieve”[15], y su principal uso en los almacenes de datos está dado para el ahorro en costo del almacenamiento en la base de datos. La utilización de este tipo de estructura posee algunas deficiencias, debido que hace las representaciones más complejas y afecta de manera directa el nivel de respuesta de las consultas.

El modelo dimensional divide el mundo de los datos en dos grandes grupos: las medidas y las dimensiones del entorno a estas medidas. Las medidas que generalmente son numéricas, se almacenan en tablas de hechos y las descripciones de los entornos que son textuales, se almacenan en las tablas dimensiones. Las tablas de hechos son tablas primarias en el modelo dimensional y contiene los valores del negocio. Los hechos más comunes son valores numéricos, cada tabla representa una relación de muchos a muchos, y contiene dos o más llaves extranjeras que se enlazan con sus respectivas tablas dimensiones.

[15] En numerosas literaturas se hace referencia a **Snowflake**, del inglés.

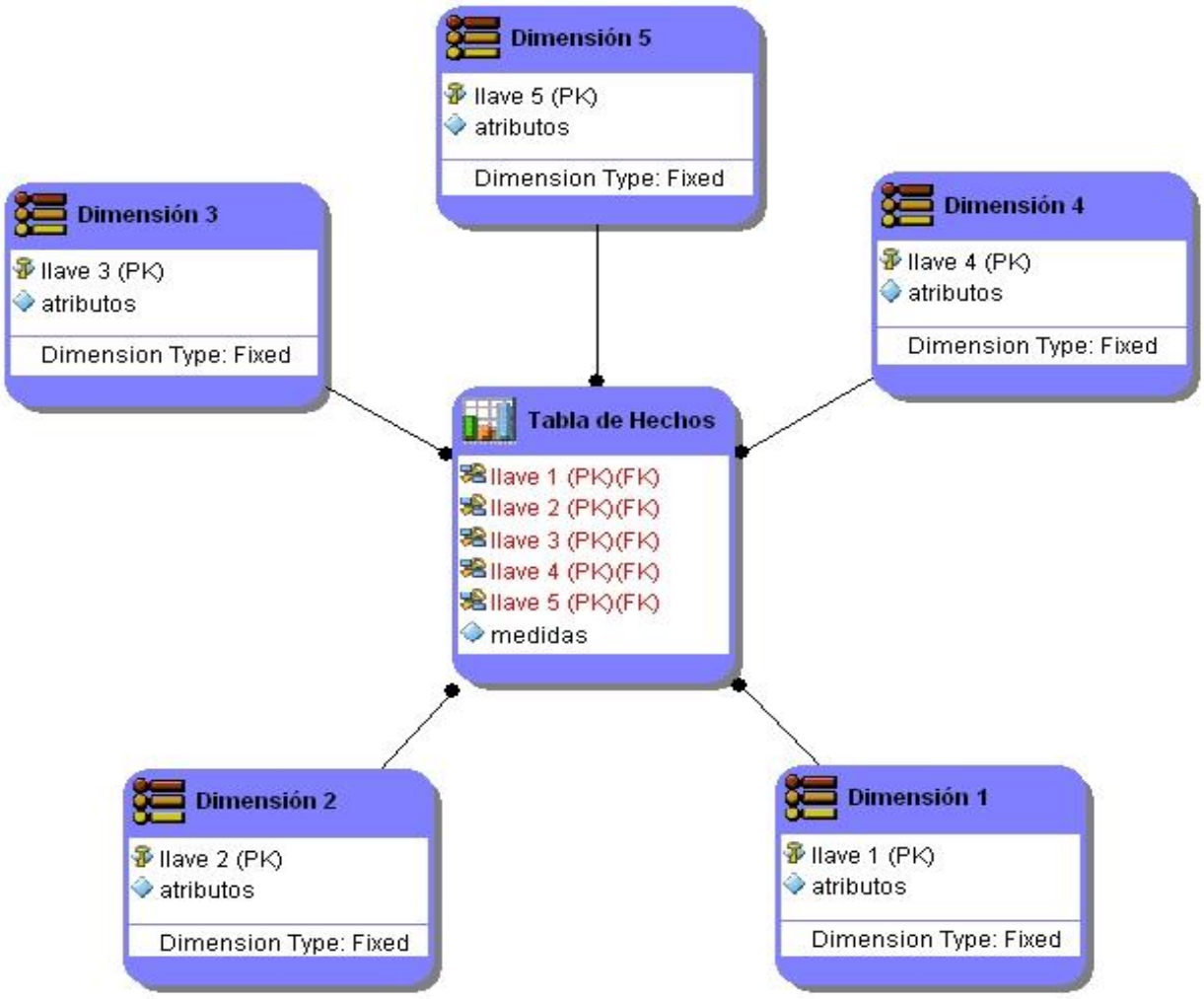

Figura 4: Representación del Esquema de Estrella

Cada proceso genera un diagrama dimensional, y su representación recibe el nombre de cubo. Cada cubo va a contener un conjunto de dimensiones y hechos. Las dimensiones son los objetos o elementos que, relacionados entre sí, determinan el proceso en el que participan. Los hechos son las medidas numéricas que caracterizan las acciones que ocurren entre los distintos objetos o dimensiones.

Viéndolo desde el punto de vista geométrico, se puede establecer una correspondencia entre los hechos y los puntos que pueden determinar en un espacio tridimensional[16], si se toma de cada arista de un cubo, coordenadas exactas.

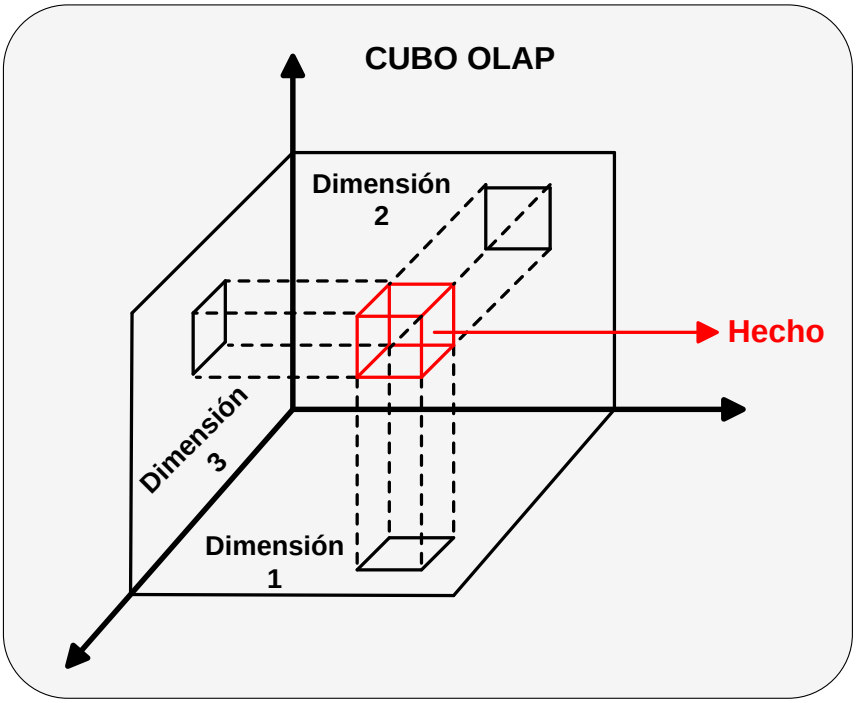

Figura 5: Relación entre el cubo, sus dimensiones y hechos

Las Tablas de Hechos, son como anteriormente se había descrito, las tablas primarias en el modelo dimensional, donde es almacenado el rendimiento de las dimensiones numéricas del negocio.(Kimball and Ross 2002). Generalmente cada tabla de hecho define un departamento determinado debido a que en ellas se almacena la información concerniente al tema en cuestión, ejemplo ventas, compra, inventario, etc. Al utilizarse el acrónimo “hecho”, se hace referencia a la medida del negocio. Cada fila de esta tabla corresponde a un hecho determinado dentro del negocio de la empresa y a su vez el conjunto de hechos

[16] Esto no implica que los cubos en un esquema dimensional tengan a lo sumo tres dimensiones. Sólo se hace la analogía con el fin de lograr una mejor comprensión por parte del lector.

dentro de esta tabla referencian a la misma granularidad. La condición principal que debe cumplir una tabla de hecho es que el hecho que se almacene tiene que ser medible, es decir, numérico y a su vez aditivo para que se puedan realizar operaciones sobre él.

Las Tablas Dimensiones son los complementos integrales de estas tablas de hechos que anteriormente se hacía referencia, estas contienen las descripciones textuales del negocio, las tablas dimensiones poseen un conjunto numeroso de atributos que definen una fila en dicha tabla. Los atributos en las tablas dimensiones sirven como fuente primaria para las restricciones de consultas y etiquetas de reportes, entre otras. Estas tablas juegan un papel fundamental dentro de la arquitectura del ODS y de cualquier almacén de datos, ya que hacen el depósito usable y entendible. Estos atributos son las llaves de entrada a las medidas almacenadas en las tablas de hechos.

La claridad y profundidad de los atributos de las dimensiones es directamente proporcional al poder del almacén de datos.(Kimball and Ross 2002)

El Modelo Dimensional debe estar preparado para almacenar el estado real de cada dimensión en cada momento, por lo que debe definirse, de acuerdo a las necesidades que se tengan, el comportamiento a seguir ante la ocurrencia de cambios. Para manejar las modificaciones en las dimensiones existen diversas técnicas, siendo una de ellas sobrescribir la información almacenada. Otras más especializadas y frecuentemente usadas para los Almacenes de Datos son la de crear un nuevo record con la información duplicada, excepto por el o los campos que se modifiquen, o agregar un campo en la misma fila, manteniendo un valor anterior y uno actual. Estas técnicas son conocidas como *Slowly Changing Dimension*[17] *(Kimball 1996,Abril)*

Una ventaja de utilizar el Modelo Dimensional es que admite la adición de dimensiones y hechos que no se habían previsto, sin que esto implique volver a cargar los datos ya almacenados. Es posible adicionar nuevos hechos en la tabla de hechos siempre que se mantenga el mismo nivel de granularidad, y se pueden adicionar dimensiones si ellas no generan más de un valor para cada fila en la tabla de hechos.

[17] Se denominan así pues los cambios en una dimensión, por lo general, no suceden con mucha frecuencia, sino que ocurren de manera aislada en el tiempo, se producen lentamente.

También es posible agregar nuevos atributos a las dimensiones. Esta característica de gran adaptabilidad es muy deseable, pues a medida que los analistas añaden nuevos requerimientos al sistema, se pueden ir incorporando las modificaciones sin que esto implique demasiados cambios.

El Modelo Dimensional, por tanto, se ajusta bastante bien a las aspiraciones que se tienen, al disminuir la normalización, resultar sencillo e intuitivo a los usuarios y bastante adaptable. Se recomienda para el ODS:

1.C

Basarse en el Modelo Dimensional para modelar conceptualmente los distintos procesos del negocio que se quieren reflejar, debido a la simplicidad que posee y por el mejor rendimiento predecible al desnormalizar sus datos.

Procesamiento analítico en línea. Aspectos Fundamentales.

Cualquiera que sea el Almacén de Datos, pretende como objetivo final de su razón de ser: preparar su contenido para presentarlo de manera útil a los usuarios finales del negocio. Pero después de reunir toda la información, ¿cómo visualizarla?

En los primeros años de la década del noventa Edgar Frank Cood[18] define el término OLAP (OnLine Analytical Processing o procesamiento analítico en línea) como una nueva tecnología que permite a los usuarios finales realizar complejos análisis de datos multidimensionales de una manera más eficiente. Una herramienta OLAP debe permitir una accesible y buena navegación de los usuarios, donde cada analista pueda visualizar de manera comprensible cada proceso del negocio en cuestión, dando esto la posibilidad de conservar análisis histórico de las diferentes dimensiones y hechos correspondientes al negocio. Según las características de un Almacén de Datos, ellos son idóneos para crear una punto de partida utilizando herramientas OLAP, ya que brinda la posibilidad de acuerdo con su arquitectura de ofrecer una fuente de información heterogénea, la cual debe ser traducida realizando las conversiones lógicas

[18] **Edgar Frank Cood** matemático graduado de la Universidad de Oxford en Reino Unido, Doctor en Ciencias de la Computación y premio Turing del año 1981.

necesarias para presentar en una vista simple y coherente de los datos finales de los cuales se extraerá la información necesaria para adquirir el conocimiento que llevará a una correcta decisión en la empresa. Cood en la tercera de sus 12 reglas, define de manera clara lo anteriormente descrito, esta se conoce como regla de Accesibilidad.[19] Hoy en día han resultado tan convenientes los Almacenes de Datos para esta tecnología, que es casi imposible hablar de una de ellas sin mencionar a la otra, estos almacenes han encontrado en las herramientas OLAP la interfaz perfecta de cara a los usuarios-analistas finales del negocio.

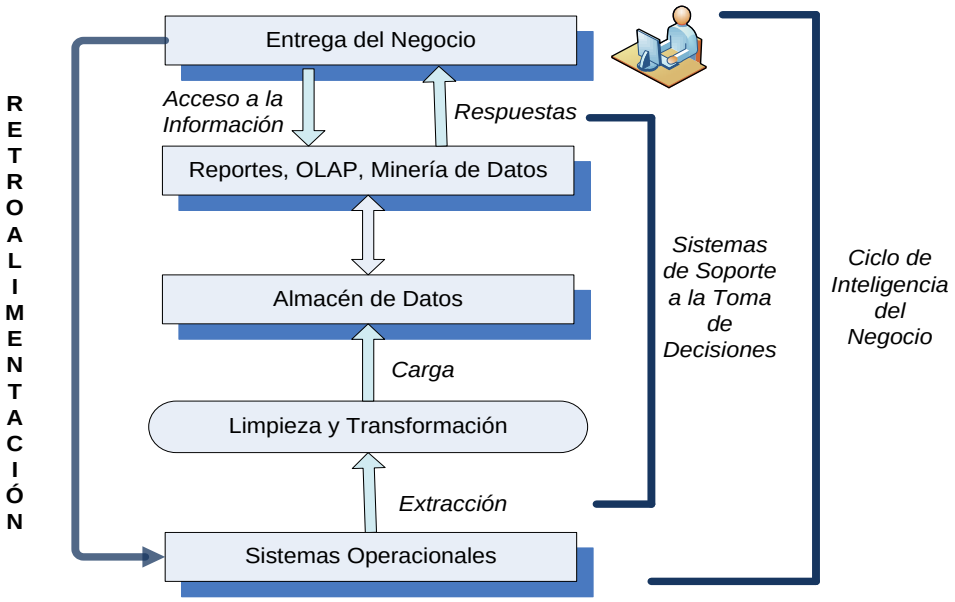

Figura 6: Almacén de Datos dentro de la Inteligencia del Negocio

La tecnología de Procesamiento Analítico en Línea –OLAP- permite un uso más eficaz de los almacenes de datos para el análisis de datos en línea, lo que proporciona respuestas rápidas a consultas analíticas complejas e iterativas utilizada generalmente para sistemas de ayuda para la toma de decisiones. Primero y más importante, OLAP presenta los datos a los usuarios a través de un modelo de datos intuitivo y natural. Con este estilo de navegación, los usuarios finales pueden ver y entender más efectivamente la

[19] En el Anexo 3 se presentan las 12 reglas que según E.F. Cood deben cumplir las herramientas OLAP.

información de sus bases de datos, permitiendo así a las organizaciones reconocer mejor el valor de sus datos.

En segundo lugar, OLAP acelera la entrega de información a los usuarios finales que ven estas estructuras de datos como cubos denominados multidimensionales debido a que la información es vista en varias dimensiones. Esta entrega es optimizada ya que se prepararan algunos valores calculados en los datos por adelantado, en vez de realizar el cálculo al momento de la solicitud. La combinación de navegación fácil y rápida le permite a los usuarios ver y analizar información más rápida y eficientemente que lo que es posible con tecnologías de bases de datos relacionales solamente. El resultado final: se pasa más tiempo analizando los datos y menos tiempo analizando las bases de datos.

A pesar del proceso de almacenamiento de datos de preparar información para el consumo del usuario final se debe facilitar la búsqueda de la información. Generalmente, las estructuras de datos de las bases tienen cierta complejidad para el usuario final, principalmente para responder a preguntas tales como:

¿Quiénes fueron los mejores vendedores de cada región durante el año pasado, mensualmente? son complejas cuando se expresan en lenguaje SQL.

Estos retos son enfrentados con herramientas avanzadas de peticiones (*queries*), las cuáles ocultan al usuario final la complejidad de las base de datos. Esta es la función de las herramientas OLAP.

Todas las organizaciones tienen datos multidimensionales y la complejidad no es necesariamente una función del tamaño de la compañía. Aún a las más pequeñas compañías les gustaría poder rastrear sus ventas por producto, vendedor, geografía, cliente y tiempo. Las organizaciones han buscado durante mucho tiempo herramientas para acceder, navegar y analizar información multidimensional de una manera fácil y natural.

Las aplicaciones OLAP deberían proporcionar análisis rápidos de información multidimensional compartida. Las características principales del OLAP son:(Pence and Creeth 2002)

- **Rápido:** proporciona la información al usuario a una velocidad constante. La mayoría de las peticiones se deben de responder al usuario en cinco segundos o menos.

- **Análisis:** realiza análisis estadísticos y numéricos básicos de los datos, predefinidos por el desarrollador de la aplicación o definido “ad hoc” por el usuario.
- **Compartida:** implementa los requerimientos de seguridad necesarios para compartir datos potencialmente confidenciales a través de una gran población de usuarios.
- **Multidimensional:** llena la característica esencial del OLAP, que es ver la información en determinadas vistas o dimensiones.
- **Información:** acceden a todos los datos y a la información necesaria y relevante para la aplicación, donde sea que ésta resida y no esté limitada por el volumen.

OLAP es un componente clave en el proceso de almacenamiento de datos y los servicios OLAP proporcionan la funcionalidad esencial para una gran variedad de aplicaciones que van desde reportes corporativos hasta soporte avanzado de decisiones.

En los primeros días de la tecnología OLAP, la mayoría de las compañías asumían que la única solución para una aplicación OLAP era un modelo de almacenamiento no relacional. Después, otras compañías descubrieron que a través del uso de estructuras de base de datos (esquemas de estrella y de copo de nieve), índices y el almacenamiento de agregados, se podrían utilizar sistemas de administración de bases de datos relacionales (RDBMS) para OLAP.

Estos vendedores llamaron a esta tecnología OLAP relacional (ROLAP)[20]. Las primeras compañías adoptaron entonces el término OLAP multidimensional (MOLAP)[21], estos conceptos, MOLAP y ROLAP, se explican con más detalle en los siguientes párrafos. Las implementaciones MOLAP normalmente se desempeñan mejor que la tecnología ROLAP, pero tienen problemas de escalabilidad. Por otro lado, las implementaciones ROLAP son más escalables y son frecuentemente atractivas a los clientes debido a que aprovechan las inversiones en tecnologías de bases de datos relacionales preexistentes.

---

[20] **R**elational **O**n-**L**ine **A**nalytical **P**rocessing, del ingles.

[21] **M**ultidimentional **O**n-**L**ine **A**nalytical **P**rocessing, del ingles.

Sistemas ROLAP

La arquitectura ROLAP, accede a los datos almacenados en un Almacén de Datos para proporcionar los análisis OLAP. La premisa de los sistemas ROLAP es que las capacidades OLAP se soportan mejor contra las bases de datos relacionales.

El sistema ROLAP utiliza una arquitectura de tres niveles. La base de datos relacional maneja los requerimientos de almacenamiento de datos, y el motor ROLAP proporciona la funcionalidad analítica. El nivel de base de datos usa bases de datos relacionales para el manejo, acceso y obtención del dato. El nivel de aplicación es el motor que ejecuta las consultas multidimensionales de los usuarios.

El motor ROLAP se integra con niveles de presentación, a través de los cuáles los usuarios realizan los análisis OLAP. Después de que el modelo de datos para el Almacén de Datos se ha definido, los datos se cargan desde el sistema operacional. Se ejecutan rutinas de bases de datos para agregar el dato, si así es requerido por el modelo de datos. Se crean entonces los índices para optimizar los tiempos de acceso a las consultas.

Los usuarios finales ejecutan sus análisis multidimensionales, a través del motor ROLAP, que transforma dinámicamente sus consultas a consultas SQL. Se ejecutan estas consultas SQL en las bases de datos relacionales, y sus resultados se relacionan mediante tablas cruzadas y conjuntos multidimensionales para devolver los resultados a los usuarios.

La arquitectura ROLAP es capaz de usar datos precalculados si estos están disponibles, o de generar dinámicamente los resultados desde los datos elementales si es preciso. Esta arquitectura accede directamente a los datos del almacén, y soporta técnicas de optimización de accesos para acelerar las consultas. Estas optimizaciones son, entre otras, particionado de los datos a nivel de aplicación, soporte a la desnormalización y joins múltiples.

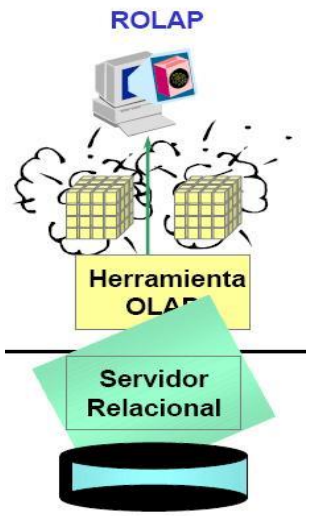

Figura 7: Arquitectura del Sistema ROLAP

Sistemas MOLAP

La arquitectura MOLAP usa unas bases de datos multidimensionales para proporcionar el análisis, su principal premisa es que el OLAP está mejor implantado almacenando los datos multidimensionalmente. Por el contrario, la arquitectura ROLAP cree que las capacidades OLAP están perfectamente implantadas sobre bases de datos relacionales. Un sistema MOLAP usa una base de datos propietaria multidimensional, en la que la información se almacena multidimensionalmente, para ser visualizada en varias dimensiones de análisis.

El sistema MOLAP utiliza una arquitectura de dos niveles: la bases de datos multidimensionales y el motor analítico. La base de datos multidimensional es la encargada del manejo, acceso y obtención del dato.

El nivel de aplicación es el responsable de la ejecución de los requerimientos OLAP. El nivel de presentación se integra con el de aplicación y proporciona un interfaz a través del cual los usuarios finales visualizan los análisis OLAP. Una arquitectura cliente/servidor permite a varios usuarios acceder a la misma base de datos multidimensional.

La información procedente de los sistemas operacionales, se carga en el sistema MOLAP, mediante una serie de rutinas por lotes. Una vez cargado el dato elemental en la Base de Datos multidimensional

(MDDB), se realizan una serie de cálculos por lotes, para calcular los datos agregados, a través de las dimensiones de negocio, rellenando la estructura MDDB.

Tras rellenar esta estructura, se generan unos índices y algoritmos de tablas hash para mejorar los tiempos de accesos a las consultas. Una vez que el proceso de compilación se ha acabado, la MDDB está lista para su uso. Los usuarios solicitan informes a través del interface, y la lógica de aplicación de la MDDB obtiene el dato.

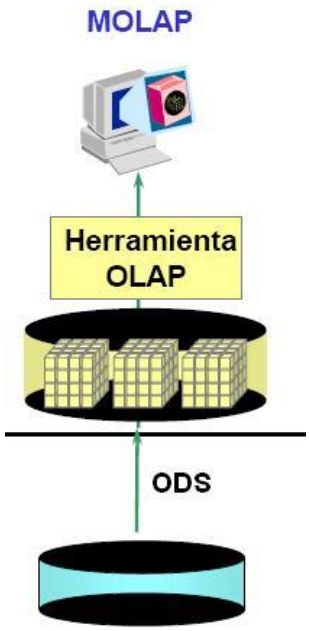

Figura 8: Arquitectura del Sistema MOLAP

Tecnología y tendencias actuales

Hoy en día, con el fin de automatizar procesos, se han creado nuevas herramientas y tecnologías capaces de llevar a cabo estos trabajos y hacerlo de forma eficiente y con un mínimo de costo sea cual fuese este. Alrededor del mundo numerosas empresas que se han dado la tarea de crear y revitalizar estas herramientas que brindan una mejor compenetración con temas de bases de datos, entre ella se puede citar a los titanes Microsoft y Oracle Corporation, los que de una manera u otra se han apoderado del mercado y del mundo de las bases de datos.

Si por cuestión de experiencia y trabajo consistente con resultados tangibles estas compañías son líderes, cabría analizar desde el punto de vista económico que impacto tendría la aplicación de sus herramientas en entidades cubanas, evidentemente el impacto sería negativo, dado los inaccesibles precios de sus privativas licencias. La tendencia hoy en el mercado del software indica una renovadora revolución e interesante propuesta, que convida a inclinarse por herramientas de software libre, que independientemente de los resultados que ya están ofreciendo se libra de cuantiosas sumas de capital y no da la posibilidad de desarrollar nuestras propias herramientas.

Sistemas Gestores de Base de Datos

Los Sistemas de Gestión de Base de Datos (SGBD)[22] son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos.

Existen, de un modo por así decirlo, tres grandes agrupaciones de sistemas gestores de base de datos. La primera los SGBD considerados productos libres dentro de este grupo se pueden encontrar 6 principales gestores dentro de ellos se encuentra como uno de los más relevantes PostgreSQL. La segunda agrupación son el conjunto de los gestores no libres donde podemos encontrar 24 productos donde resaltan como principales ORACLE, Microsoft SQLServer y MySQL. Y por último la tercera agrupación y más pequeña los gestores considerados productos no libres y gratuitos con 2 principales gestores los mismos son Microsoft SQLServer Compact Edition Basica y Sybase ASE Express Edition para Linux. (The PostgreSQL Global Development 2009)

MySQL es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL fue un software de código abierto, licenciado bajo la GPL de la GNU, MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la

[22] En numerosas literaturas se conocen por las siglas **DBMS,** del inglés **D**ata **B**ase **M**anagement **S**ystem.

posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL. Es importante señalar que MySQL es comprado a principios de 2009 por la compañía productora de software propietario SUN MicroSystem, lo cual hace que el precio por la obtención de este producto sea casi inaccesible. El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre. Unas de las principales desventajas que presenta es que; un gran porcentaje de las utilidades de MySQL no están documentadas y no es intuitivo, como otros programas (ACCESS). (Yunko Nakamura 2007)

Microsoft SQLServer es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea, así como de otras ventajas. Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma *gratuita*. Es común desarrollar completos proyectos complementando *Microsoft SQL Server* y Microsoft Access a través de los llamados **ADP** (Access Data Project). De esta forma se completa la base de datos (*Microsoft SQL Server*), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows. Para el desarrollo de aplicaciones más complejas (tres o más capas), *Microsoft SQL Server* incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.(Yunko Nakamura 2007)

Oracle es un sistema de gestión de base de datos relacional, desarrollado por Oracle Corporation. Ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22

vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años. Oracle surge a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre Sistemas Gestores de Bases de Datos de George Koch. *Computer World* definió este estudio como uno de los más completos jamás escritos sobre bases de datos. Este artículo incluía una comparativa de productos que erigía a Relational Software como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las bases de datos relacionales, algo que por aquella época era todavía desconocido. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft, MySql y de la oferta de otros RDBMS con licencia libre como PostgreSQL.(Yunko Nakamura 2007)

¿Por qué PostgreSQL como gestor de Base de Datos?

PostgreSQL es un potente Sistema de Base de Datos Relacional libre (Open Source, su código fuente está disponible) liberado bajo licencia Berkeley software Distribución (BSD). Desarrollado en la Universidad de California, en el departamento de ciencias de la computación de Berkeley. Posee más de 15 años de activo desarrollo y arquitectura probada que se ha ganado una muy buena reputación por su confiabilidad e integridad de datos. Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX (AIX, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. El nombre como tal de PostgreSQL surge en el año 1996 para establecer una relación entre el nombre original PostgreSQL y las versiones más recientes con capacidades SQL.

Principales características

- Soporta casi toda la sintaxis SQL tiene soporte total para llaves extranjeras, uniones, vistas, triggers, y procedimientos almacenados (en múltiples lenguajes).
- Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- Lenguajes Procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje

procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido. además de en C, C++ y, Java.

- Interfaces con lenguajes de programación: La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas Interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, Pike, etc.
- Herencia de tablas.
- Incluye la mayoría de los tipos de datos SQL92 y SQL99 (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIM ESTAMP), soporta almacenamiento de objetos grandes binarios, además de tipos de datos y operaciones geométricas.
- Puntos de recuperación a un momento dado, tablespaces, replicación asincrónica, transacciones jerarquizadas (savepoints), copia de seguridad en línea.
- Un sofisticado analizador/optimizador de consultas.
- Soporta juegos de caracteres internacionales, codificación de caracteres multibyte. (Alarcón José Manuel 2006)

Ventajas

- Máximo tamaño de base de datos: ilimitado.
- Máximo tamaño de tabla: 32 TB.
- Máximo tamaño de tupla: 1.6 TB.
- Máximo tamaño de campo: 1 GB.
- Máximo tuplas por tabla: ilimitado.
- Máximo columnas por tabla: 250 - 1600 dependiendo de los tipos de columnas.
- Máximo de índices por tabla: ilimitado. (Alarcón José Manuel 2006)

Herramientas

En el terreno *Código Abierto*[23] la herramienta más significativa es el Mondrian, la cual es una de las aplicaciones más importantes de la plataforma Pentaho Business Intelligence. Mondrian es un servidor OLAP código abierto que gestiona la comunicación entre una aplicación OLAP y la base de datos con los datos fuente. Es desarrollado en Java/Servlets/JSPs 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 para el análisis de aplicaciones programadas.

De todas las herramientas existentes en el mundo para la administración y explotación de almacenes de datos e Inteligencia de Negocio la más potente Open Source es Pentaho Business Intelligence Suite Enterprise Edition la cual proporciona al usuario final una simplicidad y una escalabilidad mejoradas.

Pentaho es una plataforma de Business Intellince orientada a soluciones y centrada en procesos, que incluye todos los principales componentes requeridos para implementar soluciones basados en procesos y ha sido concebida desde el principio para estar basada en procesos.(PentahoCorporation 2005).

La plataforma será capaz de ejecutar las reglas de negocio necesarias, expresadas en forma de procesos y actividades y de presentar y entregar la información adecuada en el momento adecuado. Pentaho presenta informes en los formatos conocidos (html, excel, pdf, etc.) mediante JfreeReport u otras plataformas como BIRT, JasperReports o con su última propuesta Pentaho Report Designer. Incorpora la librería JPivot, gracias a la cual se puede ver tablas OLAP a través de un navegador y realizar las aplicaciones típicas de análisis OLAP

[23] En la mayoría de las literaturas es conocido como: Open Source, por su terminología en ingles.

Para obtener la funcionalidad de procesamiento analítico en línea (OLAP) se utilizan otras dos aplicaciones: el servidor OLAP Mondrian, que combinado con Jpivot, permiten realizar consultas al ODS, que los resultados sean presentados mediante un navegador y que el usuario pueda realizar *drill down* y el resto de las navegaciones típicas.

Esta Suite de Pentaho consta de 5 componentes fundamentales para el trabajo completo de un almacén de datos, cualquiera que fuese este, incluyendo el trabajo de Extracción, Transformación y Carga (ETL), pero por su importancia especifica en la construcción del almacén se utilizarán cuatro de estas cinco.

- **Pentaho Business Intelligence**

Esta herramienta es la encargada de orquestar todos los procesos de inteligencia de negocio. Configura el servidor de inteligencia de negocio (BI server) para poder realizar la explotación del ODS. Sobre esta plataforma se definen las áreas de interés que poseen los usuarios para la preparación de reportes, consultas dinámicas y la realización del análisis OLAP.(PentahoCorporation 2005)

- **Pentaho Mondrian:**

  Es el motor OLAP de Pentaho. Su núcleo es un JAR que actúa como "JDBC para OLAP", proporcionando conexiones y ejecutando consultas SQL contra la base de datos relacional que sirve los datos. Posee mejoras en la velocidad del despliegue y desarrollo de los modelos OLAP y la facilidad de uso y acceso a datos corporativos desde la interfaz del escritorio mediante tecnología web. Posee mejoras para la realización de cálculos de miembros. Realiza análisis de Expresiones Multi-Dimensional (MDX) dentro del Lenguaje Estructurado de Consultas (SQL) para recuperar las consultas dimensionales. Posee consultas de alta velocidad para el uso de tablas de agregación en el Sistema de Administración de Base de Datos Relacional (RDBMS). Los binarios de Mondrian vienen empaquetados de diferentes maneras:

  - ✓ Como un paquete WAR que contiene Jpivot, un *framework* para trabajo con aplicaciones web y tecnología OLAP, junto con un ejemplo de datos que pueden ser cargados en una base de datos de su elección.

- ✓ Como un paquete WAR que además de contener a Jpivot, incluye una base de datos Derby, con lo que no se requiere ninguna configuración extra, aparte del despliegue sobre el servidor de aplicaciones.(PentahoCorporation 2005)

- **Pentaho Report Designer:**

  El Pentaho Report Designer es una herramienta independiente que forma parte de la unidad de reportes de Pentaho (Pentaho Reporting), que simplifica el proceso de generación de reportes, permitiendo a los diseñadores de reportes crear rápidamente informes sofisticados y ricos visualmente basados en el proyecto de reportes de Pentaho JFreeReport. Posee un diseñador gráfico basado en "arrastrar y soltar" (drag & drop) que provee completo control de acceso a los datos, agrupaciones, cálculos, gráficas, formato para reportes de alta resolución.

  También está compuesto por un Asistente paso a paso integrado que guía a los diseñadores de reportes durante el proceso de diseño. Plantillas de reportes aceleran el proceso de generación, proporcionando un aspecto consistente y atractivo además de poseer opciones de salida flexibles que incluyen los populares formatos Adobe PDF, HTML, Microsoft Excel, entre otros. (PentahoCorporation 2005)

- **Pentaho Schema Workbench:**

  Esta es una herramienta para el desarrollo del esquema del modelo estrella en XML desarrollada en Java. Este programa recientemente publicado (2007) entrega todas las facilidades para poder realizar el modelo lógico del cubo OLAP al cual se le realizarán las consultas.

  Este programa se conecta directamente con la base de datos para así poder diseñar los cubos OLAP que se requieren para que el usuario final pueda visualizar los indicadores. Luego, el archivo generado se utiliza para definir la estructura de cubo en Jasper. (PentahoCorporation 2005)

Conclusiones del Capítulo 1

Luego del análisis realizado, se puede concluir que un ODS es un repositorio de datos operacionales orientado a temas, integrado, de frecuente actualización, a nivel de detalle, que mantiene una visión

histórica de las transacciones acotada por un período de vigencia y modifica sus datos no transaccionales según se modifiquen en los sistemas operacionales; que sustenta las necesidades de información operacional de una organización y que puede funcionar como potencial intermediario entre los sistemas transaccionales y el Almacén de Datos. Es un contenedor de datos operacionales que pretende brindar una visión uniforme y consolidada de información que sirva de ayuda a la toma de decisiones en una empresa.

La tarea de diseñar un ODS no es tan simple como parece. Es una labor ardua y ambiciosa a la vez: implica sumergirse en aguas transaccionales y entender el funcionamiento de los procesos que tienen lugar en este nivel; analizar cuál es el universo de datos a manejar y las fuentes de donde provienen; proponer un diseño que se ajuste a las necesidades de información operacional de la empresa, de manera que su rendimiento sea óptimo; desarrollar un proceso de homogeneización y unificación de los datos; automatizar este proceso teniendo en cuenta distintas frecuencias de actualización y monitorear su funcionamiento. Se recorrerá, pues, este largo camino.

CAPÍTULO 2: DISEÑO E IMPLEMENTACIÓN DEL ODS

Llevar a cabo la construcción de un ODS no se puede considerar una tarea simple. Pudieran surgir en torno a ello numerosas expectativas dada las potencialidades que ofrece este tipo de almacén de datos, estas expectativas no deben ser frustradas por posibles decisiones apresuradas que podrían convertirse en erróneas a lo largo de su desarrollo. Es por todo esto que se hace necesario realizar un buen estudio de la empresa que se emprenderá y utilizar una correcta metodología de desarrollo, con el objetivo de logar bunas prácticas en el transcurso del proyecto.

Una de las decisiones más importantes que se tomarán, es qué métodos seguir para la conformación del ODS. A pesar de que no se encuentra formalizado este proceso, se han expuesto numerosos intentos por exponer algunos de los aspectos que pueden ser definidos. En la mayoría de los enfoques se repiten recetas o formulas que no pueden ausentarse, como por ejemplo la forma de cómo se modelarán los datos o pudiera ser también las frecuencias de actualización del sistema. Tomando como base, algunas de las sugerencias de cada variante, se propuso en el capítulo anterior la metodología que se iba a utilizar para el desarrollo del ODS en el presente capítulo, para no dar cabida a improvisaciones; que luego pudieran traducirse en fatales errores. Se está ya, listos para poner, entonces, manos a la obra.

Tipos de Fuentes de Datos

Las Fuentes u Orígenes de Datos como también comúnmente se le conoce, son el punto de partida para la construcción del cualquier sistema de bases de datos y son de especial importancia para el diseño y desarrollo de un almacén de datos. Las fuentes de datos se pueden agrupar en cuatro categorías o tipos diferentes importantes estas son: Datos Internos, Datos Externos, Datos de Producción y Datos Archivados.(Humphries and Hawkins 2002)

Datos Internos

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

Datos Externos

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

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 Archivados

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

La principal fuente identificada en el caso del negocio de CIMEX es: el sistema operacional Sentai, donde la Información referente a varios objetos del negocio como: proveedor, producto, localidad, cliente se encuentran ubicadas en múltiples tablas dentro de varios módulos del sistema. Esta fuente de datos entra dentro de la clasificación o de tipo datos de producción.

Sentai: este sistema trata la información de las transacciones de interés separada en varias tablas, unas con el encabezado y otras con los detalles.

El intercambio de datos se conoce que se realiza a través de ficheros, estos datos se generan por Sentai. Los ficheros solicitados se corresponden con cada una de las tablas que contienen datos necesarios para la carga, es decir, se reciben tantos ficheros como tablas hubiese sido necesario recorrer para obtener la información. Esto se hace con el objetivo de no sobrecargar a Sentai con este procesamiento.

El nombre de dichos ficheros está estructurado de la siguiente manera:

<aliasBaseDatos>.<fechaInicio>.<fechaFin>.<nombreTabla>.txt

Siendo fechaInicio y fechaFin los límites inferior y superior respectivamente del intervalo de tiempo a cargar. Las fechas tienen el formato AAAAMMDD.

Cada uno de estos ficheros de texto es compactado en ficheros .gz con una estructura similar en el nombre, y luego agrupados en un único fichero .tar correspondiente a cada base de datos de la que se extrae. Los compactados finales son colocados de manera automática en un ftp creado con ese propósito, del cual el proceso ETL obtiene los datos.

Definición de las Áreas de Análisis

La definición de las Áreas de Análisis (AA) es uno de los pasos más importantes dentro del desarrollo de DW. 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 procesos del negocio que realiza CIMEX. En este sentido se definieron 6 AA que están en concordancia con las necesidades de información del cliente que se identificaron en los siguientes ámbitos.

1. Ajuste
2. Compra
3. Inventario
4. Existencia
5. Transferencia
6. Venta

Pasos para el diseño del ODS

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

   El proceso generalmente está vinculado a sistemas con colecciones de fuentes a integrar en el almacén.

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 podrá describir una 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 la Tabla de Hecho. En este caso le pregunta es: ¿Cuáles son las medidas o hechos?

Para la implementación de los Almacenes de Datos, cualesquiera que estos sean, es necesario utilizar una matriz para representar la relación entre las dimensiones y los procesos del negocio, conocida de acuerdo a la metodología que se está utilizando y la generación de artefactos según establece esta metodología como Matriz de Bus. Destacando que los Mercados de Datos se realizan en base a las fuentes no a los departamentos existentes en la organización.

Diseño del Sistema

Para lograr un correcto diseño de la solución que se expondrá, este estará enfocado a dos niveles fundamentales, uno muy bien detallado con un alto nivel de atomicidad, que por lo general entorno al él giran las informaciones operacionales y el otro menos detallado que será más cercano a la información que comúnmente se consulta para análisis hitóricos. Para logar el correcto modelado y diseño, se utilizarán los mismos pasos que define la metodología descrita en el capítulo anterior.

Procesos del Negocio a modelar

La corporación CIMEX es una entidad cubana encargada de comprar disímiles productos a suministradores nacionales y extranjeros, con el fin de comercializarlos a través de una red de tiendas distribuidas por todo el país. El manejo de los datos correspondientes al Comercio Mayorista que ocurre diariamente, se realiza haciendo uso de Sentai, un sistema automatizado orientado a la gestión empresarial. Está implementado sobre el gestor de bases de datos Progres y soportado por Unix y Linux.

Las bases de datos de Sentai pueden residir en servidores centrales de la corporación o en servidores locales ubicados en Sucursales, Divisiones u otras entidades.

En la actividad comercial mayorista se pueden identificar distintos procesos, entre ellos las compras, tanto las efectuadas como las pendientes, las ventas, el inventario, entre otros. Estos procesos involucran objetos del negocio como son los productos, los clientes, los proveedores, las localidades, por citar algunos. El proceso mayorista se encuentra estructurado por módulos, cada uno encargado de manejar los distintos tipos de transacciones que tienen lugar. Una acción en el sistema puede generar transacciones en uno o varios módulos. Como son de interés los procesos relacionados con la actividad comercial, los módulos sobre los que se centrará la atención serán el de Compra, Venta y Administración de Inventario.

Desde el punto de vista comercial, toda negociación que implica adquisición de mercancías o productos cuyo destino sea generar inventarios en almacenes, para su posterior venta a clientes, distribución a otras sucursales o transferencia a otros almacenes propios; se materializa generalmente, mediante un contrato comercial, cuando el suministro de los proveedores de los mismos es estable con la entidad y cuando las compras son eventuales o puntuales con un suministrador, se requiere de una autorización de compra u orden de compra específica para materializar la negociación. Aunque las entregas de mercancías o productos previstas en los contratos comerciales pueden ejecutarse totalmente de una sola vez o en forma parcial, suele confeccionarse una autorización de compra u orden de compra por cada entrega de suministro. Luego, la orden de compra, es por excelencia el documento primario que se utiliza para formalizar las negociaciones por concepto de adquisiciones de mercancías o productos entre empresas, compañías y demás entidades. Por tanto, dicho documento, es una solicitud de las cantidades ordenadas a adquirir, expresadas en las unidades de medida acordadas, los precios unitarios pactados y el importe o monto de las operaciones que se generan por cada mercancía o producto a comprar. La orden de compra, por su naturaleza, no ejecuta contabilización alguna en los registros contables del sistema, hasta tanto sea recibida en una localidad o área de inventario prevista, por las cantidades e importes realmente recibidos. Por otra parte las ventas están dirigidas a satisfacer todas las necesidades de la gestión de ventas de una organización. Por medio del mismo se permite controlar las cotizaciones a los clientes, generar órdenes de ventas, documentos asociados a la disposición de los inventarios, facturación, ingreso de pagos, devoluciones y las transferencias de existencias entre localidades de una entidad.

En el proceso de negocio del CIMEX, los módulos de órdenes de compras, órdenes de ventas, inventarios, administración de la distribución y administración de almacenes, están orientados y relacionados directamente a las actividades de la gestión comercial de compras y ventas de mercancías, tanto mayoristas como minoristas, así como al almacenamiento, manipulación y entrega de las mismas, en aquellas empresas, compañías y otras entidades, que las desarrollan. Mientras que los servicios y el inventario asociado, están orientados y relacionados directamente con la gestión comercial de prestación de servicios de reparación, mantenimiento, garantía y otras actividades afines en talleres dedicados a reparar equipos técnicos de cualquier naturaleza, talleres dedicados a reparar transportes automotores, y otros talleres similares, que laboren por órdenes de trabajo, en aquellas empresas, compañías y otras entidades, que desarrollan estas actividades. Así las órdenes de producción y el de inventario asociado, están orientados y relacionados directamente con la gestión de centros de elaboración de alimentos con fines gastronómicos, gestión de producciones materiales de proceso continuo, así como la comercialización posterior de las mismas, en empresas, compañías y entidades dedicadas a actividades de naturaleza productiva. Teniendo como trasfondo las cuentas por cobrar, cuentas por pagar, activos fijos, el de inventario propiamente, el de contabilidad general y el de conciliación bancaria, están asociados por su naturaleza, a la actividad económica de las empresas, compañías y otras entidades que explotan el sistema, y mediante los cuales pueden controlar su registro contable y emisión de estados financieros, incluidos los reportes estadísticos correspondientes que se nutren de la contabilidad, pueden regular la actividad de planificación y presupuestos de gastos corrientes e inversiones y pueden implementar su actividad financiera y bancaria, entre otros aspectos. Siendo todos estos módulos parte de la gestión comercial, de prestación de servicios o producción, en las empresas y compañías, se enlazan con los módulos que reflejan la actividad económica, en aras de la integralidad de la administración empresarial en su conjunto.

Granos identificados

Según las buenas prácticas de desarrollo y siguiendo los pasos de la metodología de Kimball, se sugiere que se elaboren modelos dimensionales sobre la información más detallada que se captura como parte del proceso del negocio sobre el cual se está trabajando. La definición de los granos es lo que determina en gran medida el alcance dimensional que pudiera llegar a tener el modelado de las estructuras, influyendo esto significativamente en el tamaño del sistema que se está desarrollando.

En concordancia con las especificaciones del negocio quedan identificados 6 granos.

Ajuste, definido como, el acomodo que realiza un área, este es de un tipo específico, en un día, en una transacción sobre un producto establecido por un proveedor.

Compra, definido como, en un día, un área realiza la compra de determinado producto establecido por su proveedor en una transacción y mediante un documento que autorice.

Existencia, definido como, en un día se realiza un control de existencia, en una localidad, sobre un producto establecido por un proveedor.

Inventario, definido como, el control que se realiza de un producto establecido por un proveedor, en un día, de un área del negocio.

Transferencia, definido como, en un día se realiza una transacción desde un área, de un producto establecido por un proveedor, mediante un documento autorizador.

Venta, en un día se realiza una venta a un cliente de un producto establecido por un proveedor, el producto proviene de determinada fuente de venta, establecida en un área. Mediante un documento autorizador que apruebe la transacción.

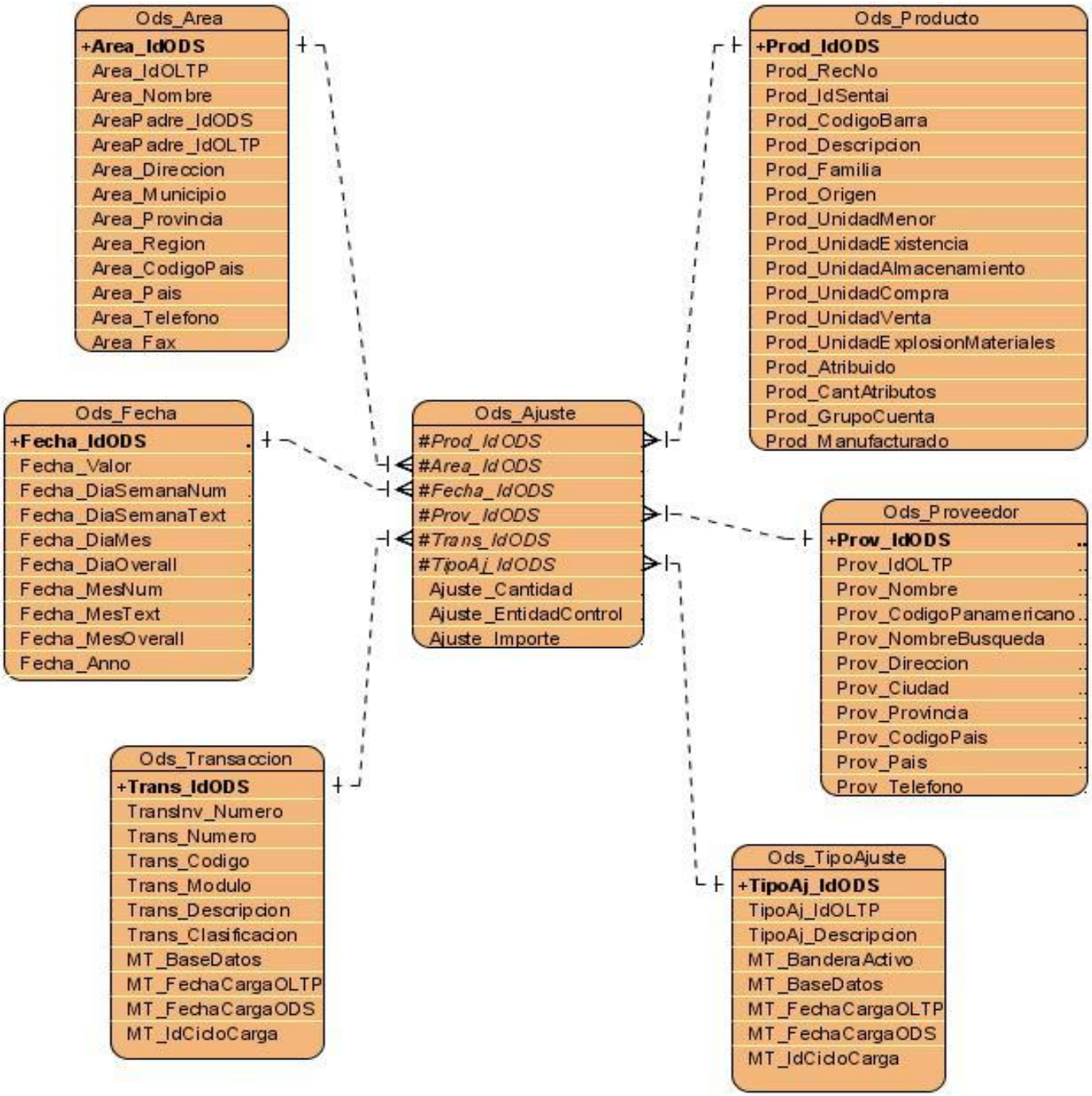

Figura 9: Estructura del Grano del proceso Ajuste en la Solución. Modelo Dimensional

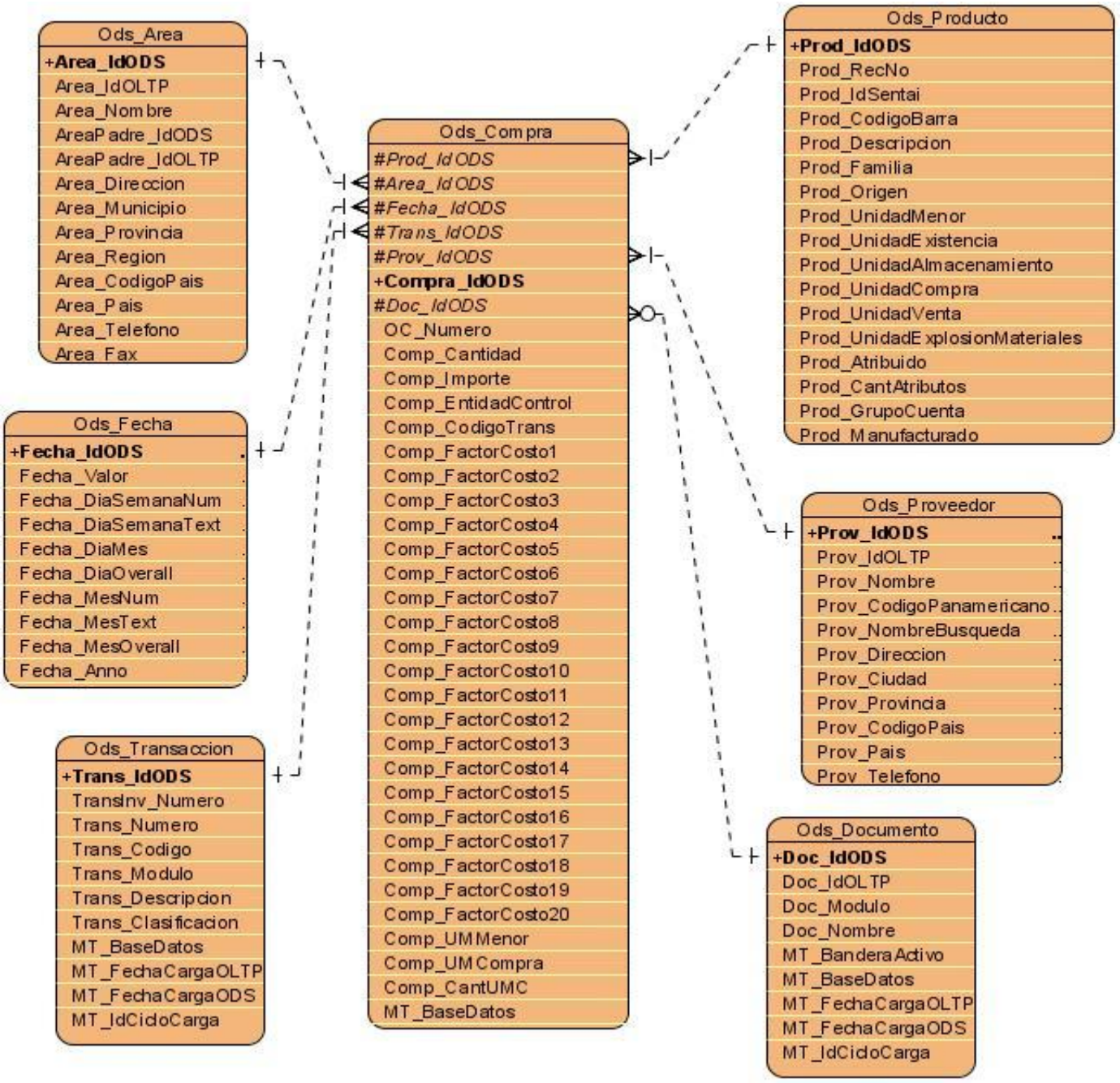

Figura 10: Estructura del Grano del proceso Compra en la Solución. Modelo Dimensional

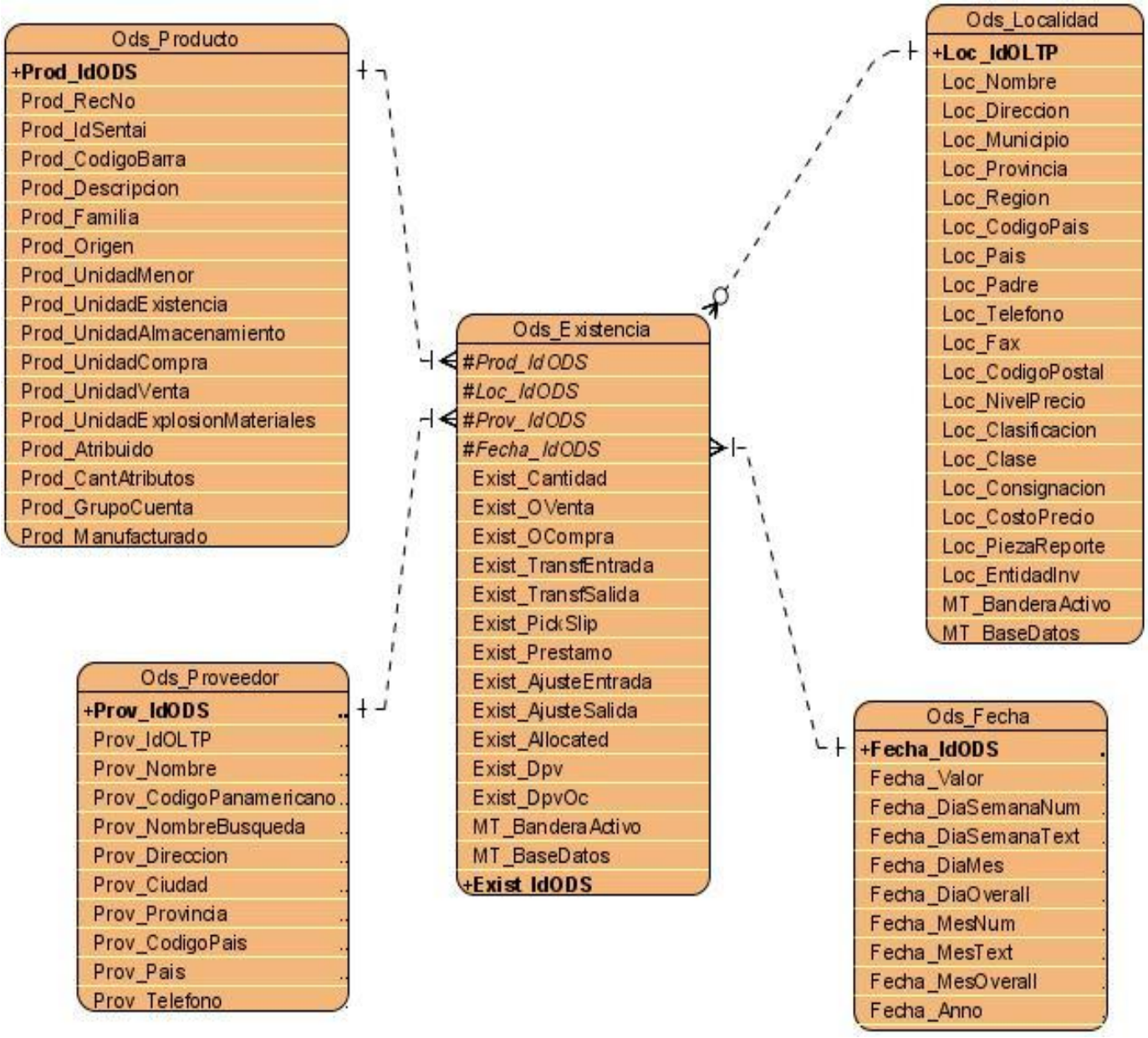

Figura 11: Estructura del Grano del proceso Existencia en la Solución. Modelo Dimensional

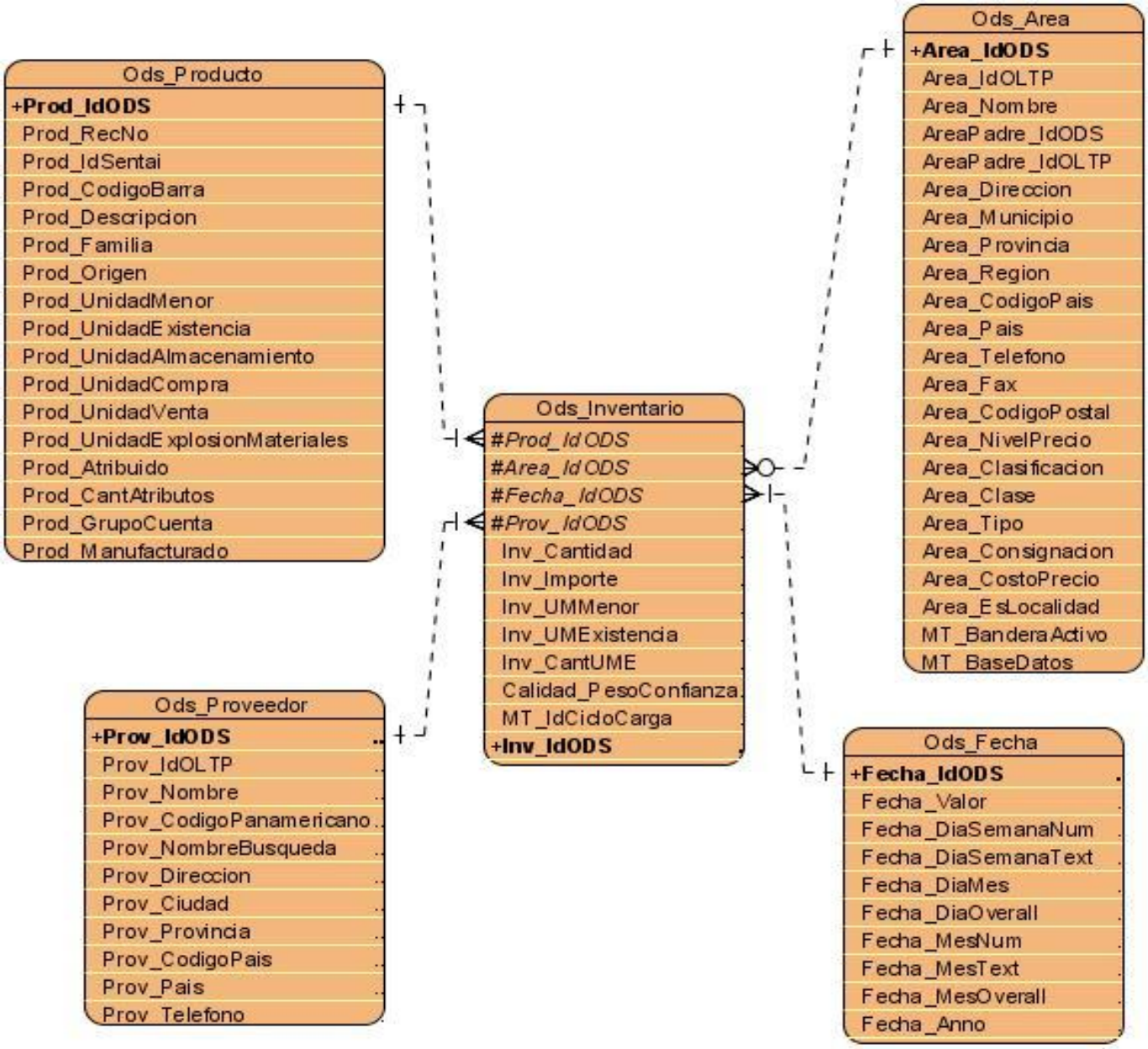

Figura 12: Estructura del Grano del proceso Inventario en la Solución. Modelo Dimensional

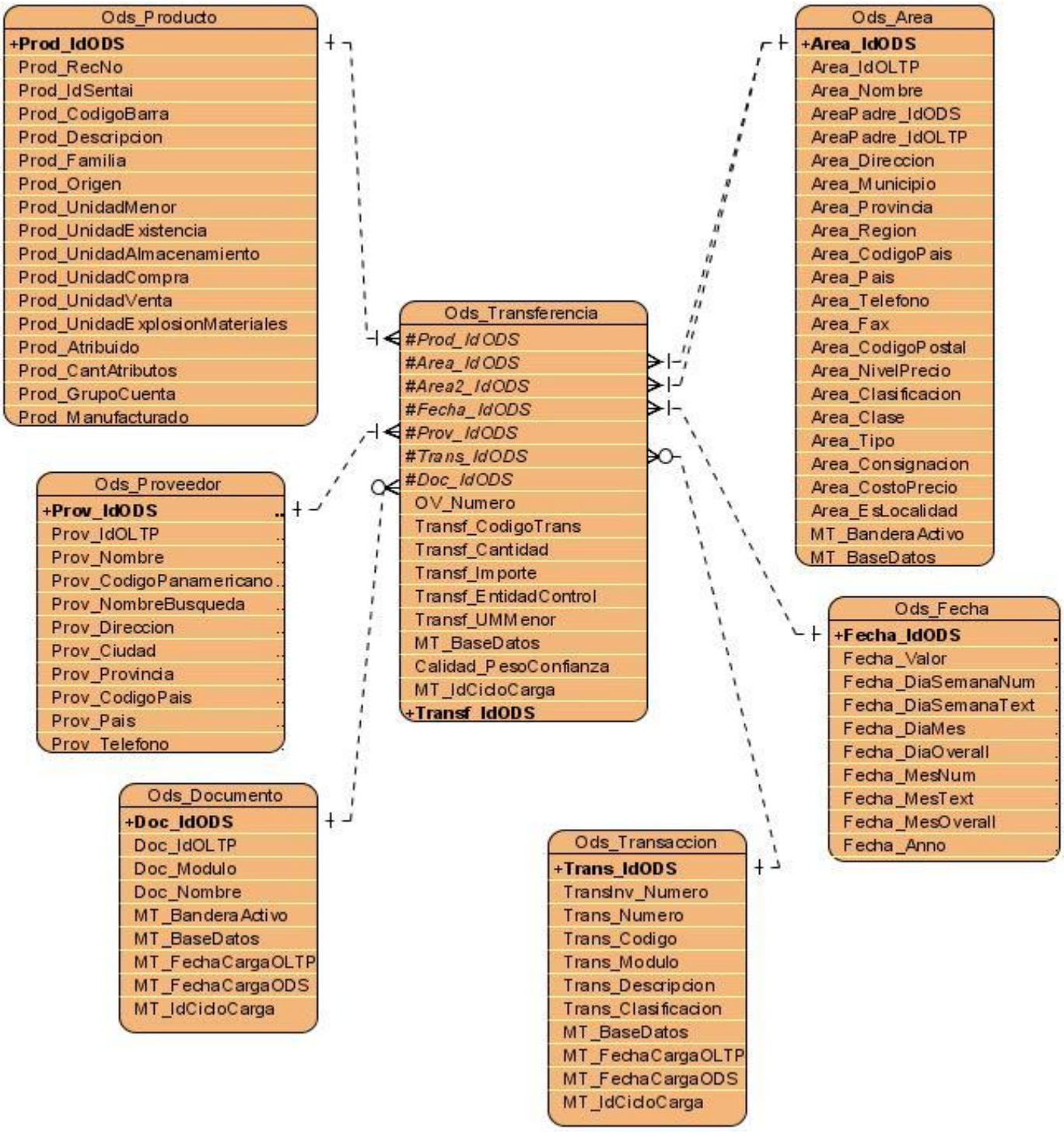

Figura 13: Estructura del Grano del proceso Transferencia en la Solución. Modelo Dimensional

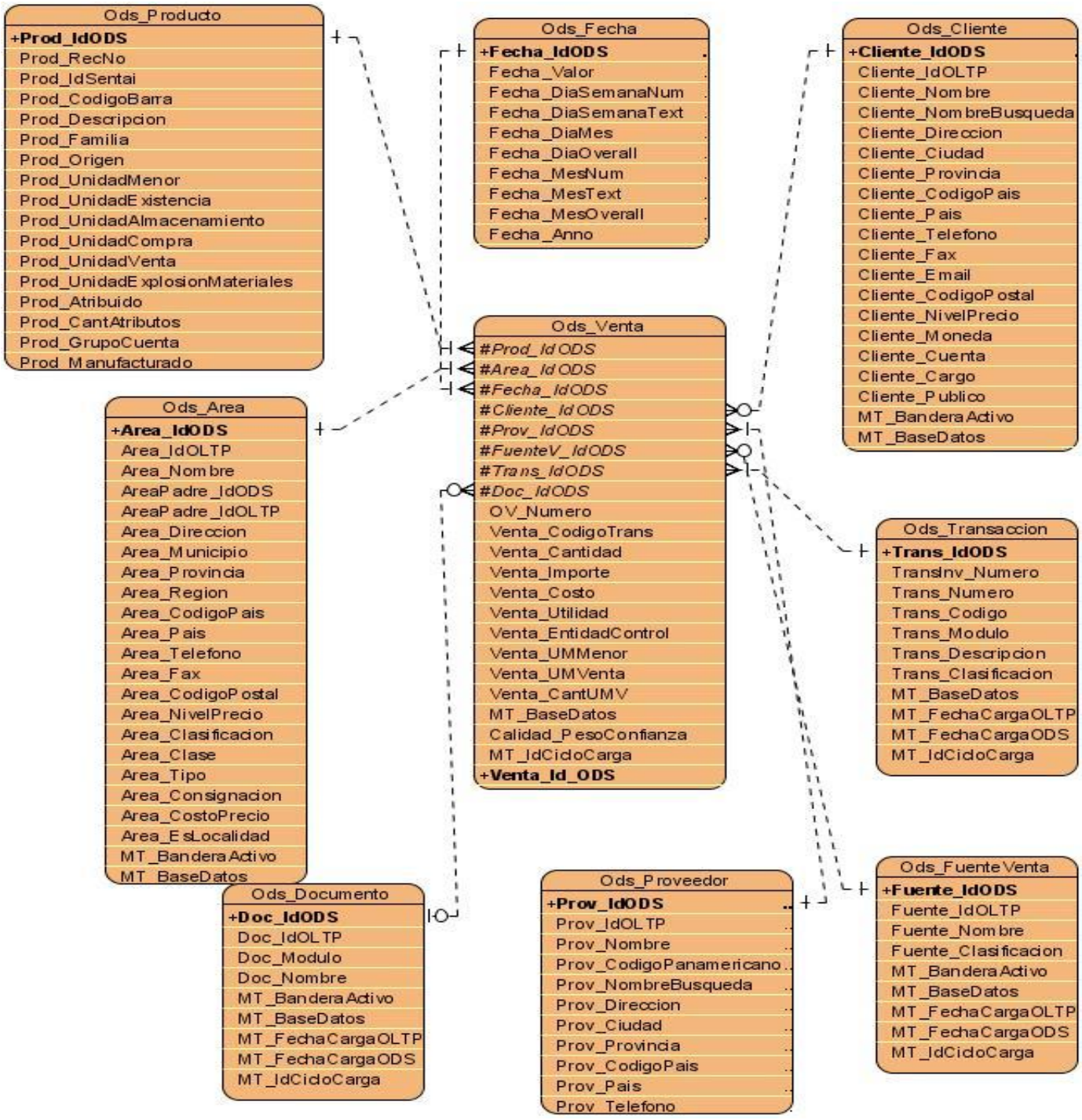

Figura 14: Estructura del Grano del proceso Venta en la Solución. Modelo Dimensional

Dimensiones identificadas

Luego de haber declarado el grano de los principales procesos en el desenvolvimiento del negocio de la Entidad, lo correcto sería seguir los pasos de las buenas prácticas, entonces debemos describir e identificar las principales dimensiones que entrono a ellas gira cada proceso del negocio. Las dimensiones pueden ser agrupadas según sus características, y desde el punto de vista conceptual existen dos grandes grupos: las dimensiones, propiamente dicho el término o también conocidas como *dimensiones unitarias* y la otra agrupación las *dimensiones compartidas*. En las figuras anteriores se pudo observar una propuesta completa de la estructura del modelo dimensional de cada proceso del negocio.

Dimensiones

Esta categoría agrupa las dimensiones que únicamente se relacionan con una sola tabla de hechos, que a su vez estas son interpretadas como los principales procesos dentro del negocio de CIMEX. De un total de 10 dimensiones que giran en torno a la agrupación de todos los principales procesos, solo 3 se consideran dimensiones unitarias.

Dimensión Tipo de Ajuste (Ods_TipoAjuste)

Esta dimensión describe que tipo de ajuste se va a realizar y de donde proviene el ajuste realizado.

Jerarquía:

1. descripción (TipoAj_Descripcion)

Dimensión Localidad (Ods_Localidad)

Esta dimensión describe las características de la localidad, hacia donde se enfoca uno de los procesos del negocio del sistema, en este caso el proceso Existencia.

Jerarquía:

1. país (Loc_Pais)
    1.1. provincia (Loc_Provincia)

1.1.1. municipio (Loc_Municipio)

1.1.1.1. nombre (Loc_Nombre)

Dimensión Cliente (Ods_Cliente)

Esta dimensión describe las características del cliente, quien gira en torno al proceso del negocio Venta.

Jerarquía:

1. país (Cliente_Pais)

1.1. provincia (Cliente_Provincia)

1.1.1. dirección (Cliente_Direccion)

1.1.1.1. nombre (Cliente_Nombre)

Dimensión Fuente de Venta (Ods_FuenteVenta)

Esta dimensión describe el origen de la venta de cada producto, dentro del propio proceso del negocio Venta.

Jerarquía:

1. clasificación (Fuente_Clasificacion)

1.1. nombre (Fuente_Nombre)

Dimensiones compartidas

Definir las siguientes dimensiones como compartidas o conformadas no responde a otro significado que no sea al de que poseen la misma interpretación para más de una tabla de hechos, esto quiere decir, físicamente, que están relacionadas con más de una tabla de hechos. El uso de este tipo de estructura ayuda a unir contenidos diferentes y permite el uso consistente de los atributos especificados en la dimensión.

Dimensión Área (Ods_Area)

Esta dimensión compartida describe las principales características que presenta un área, la misma está involucrada en los procesos del negocio: Ajuste, Compra, Inventario, Transferencia y Venta.

Jerarquía:

1. país (Area_Pais)
    1.1. provincia (Area_Provincia)
        1.1.1. municipio (Area_Municipio)
            1.1.1.1. nombre (Area_Nombre)

Dimensión Fecha (Ods_Fecha)

Esta es una de las dimensiones más complejas que existen ya que está presente en todos los procesos del negocio de CIMEX, describe el tiempo en que se realiza cada operación dentro del negocio.

Jerarquía:

1. año (Fecha_AnnoOverall)
    1.1. mes (Fecha_MesOverall)
        1.1.1. día (Fecha_DiaOverall)

Dimensión Producto (Ods_Producto)

Esta, al igual que la dimensión Fecha, está presente en todos y cada uno de los procesos del negocio que anteriormente se mencionaron. Describe cada una de las características que posee el producto en cuestión, un nivel de detalle bastante grande.

Jerarquía:

1. línea (Prod_Linea)

    1.1. sección (Prod_Seccion)

        1.1.1. departamento (Prod_Departamento)

            1.1.1.1. grupo (Prod_Grupo)

                1.1.1.1.1. división (Prod_Division)

                    1.1.1.1.1.1. origen (Prod_Origen)

                        1.1.1.1.1.1.1. familia (Prod_Familia)

                            1.1.1.1.1.1.1.1. código de barra (Prod_CodigoBarra)

Dimensión Proveedor (Ods_Proveedor)

Esta dimensión al igual que las otras dos anteriores, están involucradas en todos los procesos del negocio de la Entidad, describe de manera detallada las características de cada proveedor que fija determinado producto.

Jerarquía:

1. país (Prov_Pais)

    1.1. provincia (Prov_Provincia)

        1.1.1. ciudad (Prov_Ciudad)

            1.1.1.1. nombre (Prov_Nombre)

Dimensión Transacción (Ods_Transaccion)

La dimensión Transacción permite caracterizar los flujos cuales quiera que estos sean, girando en torno a las operaciones realizadas en los procesos Ajuste, Transferencia, Compra y Venta.

Jerarquía:

1. clasificación (Trans_Clasificacion)

    1.1. módulo (Trans_Modulo)

        1.1.1. número (Trans_Numero)

Dimensión Documento (Ods_Documento)

La dimensión Documento describe las características de cada uno de los documentos normativos que giran en torno a cada operación dentro de los procesos: Trasferencia, Compra y Venta.

Jerarquía:

1. módulo (Doc_Modulo)

    1.1. nombre (Doc_Nombre)

En muchas ocasiones interesa disponer de los datos a varios niveles de granularidad, es decir, es importante para el negocio poder consultar los datos por localidades, provincias, etc., en estos casos se crea una jerarquía con la dimensión, ya que se tienen varios niveles de asociación de los datos (con otras dimensiones como el *tiempo*, se podrían crear niveles jerárquicos del tipo 'días', 'semanas', 'meses'). Cuando las tablas de dimensión asociadas a una tabla de hechos no reflejan ninguna jerarquía (por ejemplo: las zonas siempre son 'provincias' y sólo *provincias*, el tiempo se mide en 'días' y sólo en *días*, etc.) el cubo resultante tendrá forma de estrella, es decir, una tabla de hechos central rodeada de tantas tablas como dimensiones, y sólo habrá, además de la tabla de hechos, una tabla por cada dimensión. Cuando una o varias de las dimensiones del cubo reflejan algún tipo de jerarquía existen dos planteamientos con respecto a la forma que deben ser diseñadas las tablas de dimensión. Uno de ellos consiste en reflejar todos los niveles jerárquicos de una dimensión dentro de una única tabla, en este caso también se tendrá un esquema en estrella como el que se ha descrito anteriormente.

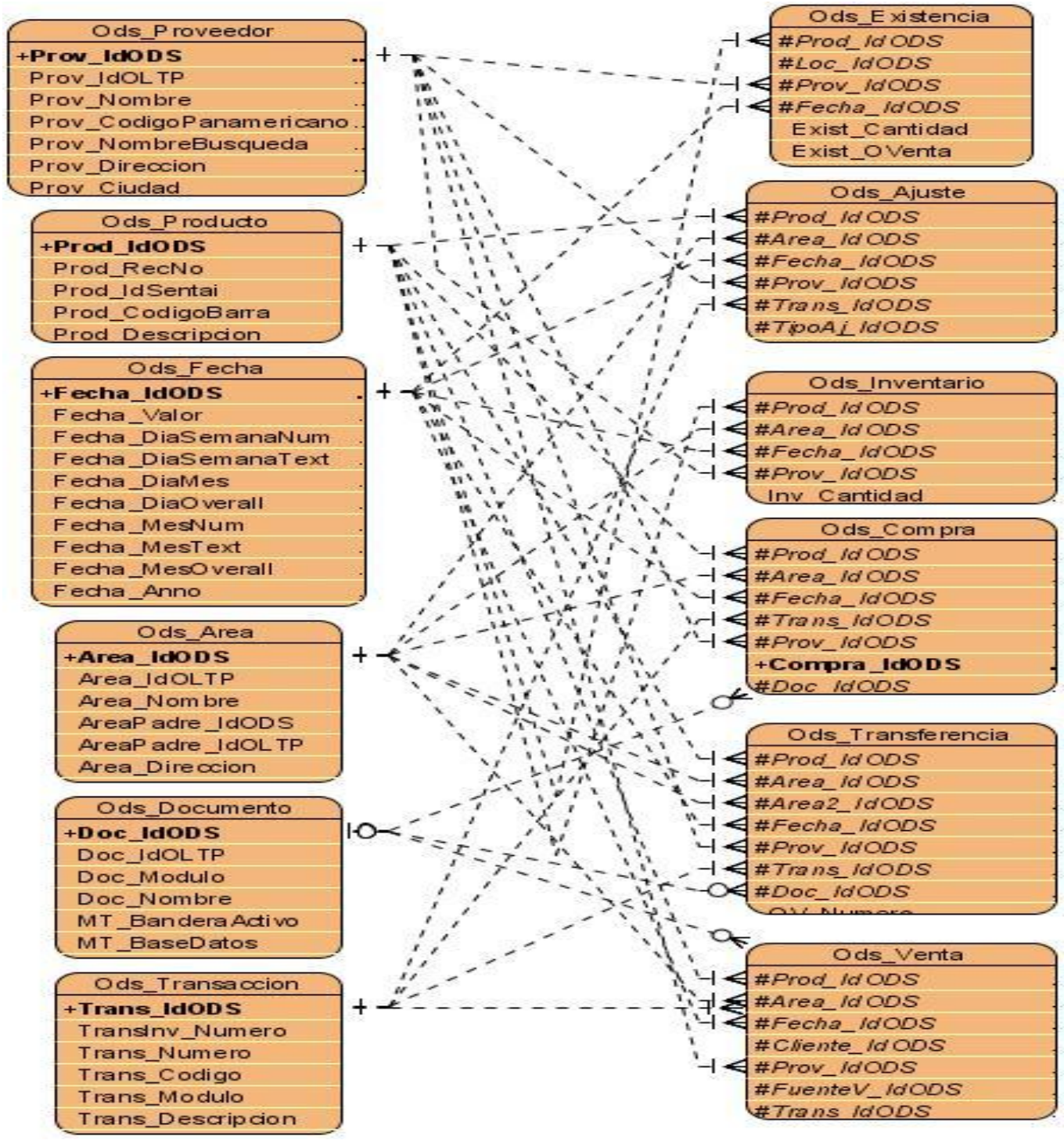

Figura 15: Modelo Dimensional. Constelación de las Dimensiones Compartidas

Hechos identificados

Las tablas de hechos, como se hizo referencia en el capítulo anterior, van a contener las medidas o hechos asociados a dicho proceso, así como una llave compuesta por llaves foráneas correspondientes a las llaves primarias de cada tabla de dimensión, representando de esa forma una relación muchos-muchos entre todas ellas. Por esto se recomienda que las llaves en las dimensiones deban ser simples, para disminuir el tamaño de la tabla de hechos y realizar el *join* por una sola columna.

Siguiendo las particularidades del caso en cuestión se han identificados como hechos los principales procesos que lleva a cabo la Corporación, los mismo son: Ajuste, Compra, Existencia, Inventario, Trasferencia y Venta.[24]

Tabla de Hechos Ajuste (Ods_Ajuste)

En esta tabla se registran todos los ajustes que ocurren en las principales operaciones dentro de este específico proceso del negocio, está relacionada con seis dimensiones.

Dimensiones Asociadas:

1. Área (Ods_Area)
2. Fecha (Ods_Fecha)
3. Producto (Ods_Producto)
4. Proveedor (Ods_Proveedor)
5. Tipo de Ajuste (Ods_TipoAjuste)
6. Transacción (Ods_Transacción)

[24] Ver Anexo 5: Descripción de las Tablas de Hechos

Medidas:

1. Cantidad (Ajuste_Cantidad)
2. Importe (Ajuste_Importe)

Tabla de Hechos Compra (Ods_Compra)

En esta tabla se registran cada una de las operaciones que se pudieran realizar con la concreción del proceso compra, está relacionada con 6 dimensiones.

Dimensiones Asociadas:

1. Área (Ods_Area)
2. Fecha (Ods_Fecha)
3. Producto (Ods_Producto)
4. Proveedor (Ods_Proveedor)
5. Transacción (Ods_Transacción)
6. Documento (Ods_Documento)

Hechos:

1. Importe (Comp_Importe)
2. Cantidad (Comp_Cantidad)

Tabla de Hecho Existencia (Ods_Existencia)

Es una tabla registra una descripción más detallada que un inventario ordinario, pues esta contempla compromisos, productos que ya han salido pero que no han entrado y viceversa., entre otras características. Está relacionada con 4 dimensiones.

Dimensiones Asociadas:

1. Fecha (Ods_Fecha)
2. Producto (Ods_Producto)
3. Proveedor (Ods_Proveedor)
4. Localidad (Ods_Localidad)

Hechos:

1. Cantidad (Exist_Cantidad)
2. Órdenes de Venta (Exist_OVenta)
3. Órdenes de Compra (Exist_OCompra)
4. Transferencias de Entrada (Exist_TransfEntrada)
5. Transferencias de Salida (Exist_TransfSalida)
6. Préstamos (Exist_Prestamo)
7. Ajustes de Entrada (Exist_AjusteEntrada)
8. Ajustes de Salida (Exist_AjusteSalida)
9. Asignados (Exist_Allocated)

Tabla de Hechos Inventario (Ods_Inventario)

En esta tabla de hecho se registran los inventarios que se realizan producto a operaciones en cada uno de los procesos del negocio. Con esta tabla se relacionan 4 tablas dimensionales.

Dimensiones Asociadas:

1. Producto (Ods_Producto)
2. Fecha (Ods_Fecha)
3. Proveedor (Ods_Proveedor)
4. Área (Ods_Area)

Hechos:

1. Cantidad ()
2. Importe ()

Tabla de Hechos Transferencia (Ods_Transferencia)

Esta Tabla de hechos o proceso del negocio, tiene el control, registra cada movimiento en cada una de las operaciones que se vayan a realizar como lo indica su nombre.

Dimensiones Asociadas:

1. Producto (Ods_Producto)
2. Fecha (Ods_Fecha)
3. Proveedor (Ods_Proveedor)
4. Área (Ods_Area)
5. Documento (Ods_Documento)
6. Transacción (Ods_Transaccion)

Hechos:

1. Cantidad (Transf_Cantidad)

2. Importe (Transf_Importe)

Tabla de Hechos Venta (Ods_Venta)

En esta tabla se abarca uno de los procesos más importantes del negocio, las ventas, tipifica las órdenes de venta la cantidad de productos que se venden en cada operación y el importe que implica esto.

Dimensiones Asociadas:

1. Producto (Ods_Producto)
2. Fecha (Ods_Fecha)
3. Proveedor (Ods_Proveedor)
4. Área (Ods_Area)
5. Documento (Ods_Documento)
6. Transacción (Ods_Transaccion)
7. Cliente (Ods_Cliente)
8. Fuente de Venta (Ods_FuenteVenta)

Hechos:

1. Cantidad (Venta_Cantidad)
2. Importe (Venta_Importe)
3. Costo (Venta_Costo)

Para una mejor comprensión de la relación existente entre las diferentes dimensiones del negocio que se ha definido se ha construido la siguiente tabla que engloba esta correlación teniendo en cuenta que estos procesos del negocio coincides con las 6 Áreas de Análisis anteriormente identificadas y propuestas.

AA/Dimensión	Área	Fecha	Producto	Proveedor	Tipo de Ajuste	Transacción	Documento	Localidad	Cliente	Fuente de Venta
Ajuste	x	x	x	x	x	x				
Compra	x	x	x	x		x	x			
Inventario	x	x	x	x						
Existencia		x	x	x				x		
Transferencia	x	x	x	x		x	x			
Venta	x	x	x	x		x	x		x	x

Tabla 1: Correlación existente entre Áreas del Análisis y las Dimensiones

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.

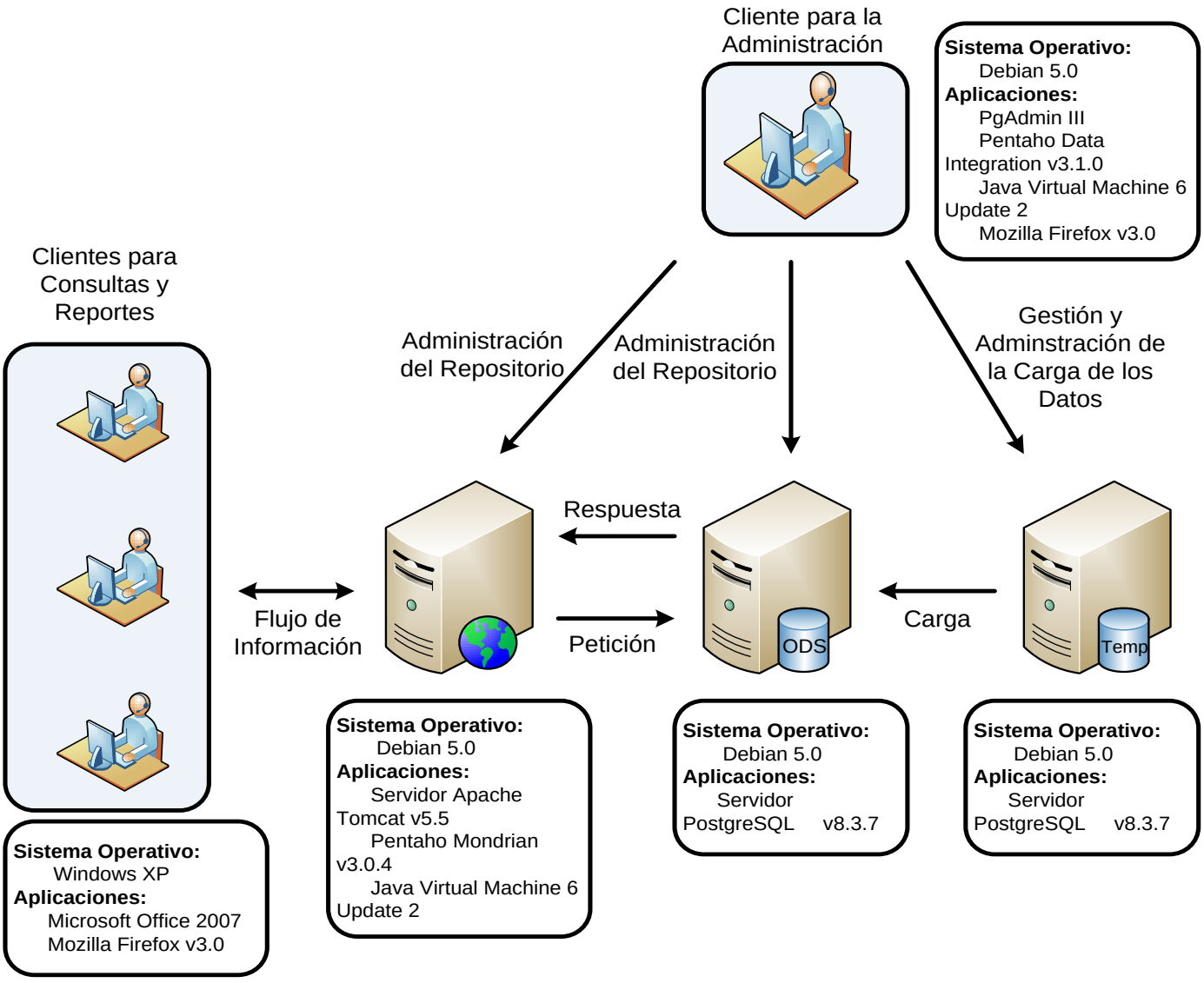

Figura 16: Arquitectura de la Solución propuesta

Arquitectura del ODS

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 o repositorio temporal. 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.

Las estructuras diseñadas están definidas en un principal nivel de agrupación y concentrada en una misma instancia de la base de datos. Estas son las estructuras con los datos detallados relacionados con todas las dimensiones propuestas donde estará almacenado además de los datos en explotación, los que se encuentran en proceso de validación por parte de los especialistas del negocio en CIMEX, traduciéndose estos en los principales procesos del negocio de la entidad.

El componente de presentación es mediante el cual los usuarios finales van a interactuar con la información. La comunicación se realizará mediante las estaciones de trabajo que actualmente poseen los especialistas del negocio y la utilización del navegador web Mozilla Firefox 3.0. La herramienta propuesta a utilizar en este caso es la interfaz de consulta del motor OLAP Mondrian en su versión 3.0.4 el cual facilitará la creación de consultas MDX y la incorporación de reportes dinámicos permitiendo además exportar a diferentes formatos y gráficos.

Por último y no por ello menos importante se encuentra el componente de administración y carga de datos, o también conocido como repositorio temporal. Para el cual se propone la utilización de la herramienta Pentaho Data Integration 3.1.0 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 dejará las rutinas de ETL diseñadas para ser utilizadas cada vez que se desee adicionar datos al repositorio central (ODS) siempre y cuando esos datos se encuentren en un formato similar al utilizado para la carga del primero.

Granularidad del proceso

En las tablas de hechos, la granularidad 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 las Tablas de Hechos. Por esta razón se puede inferir la estrecha relación existente entre las dimensiones y la granularidad.

La utilización y seguimiento de buenas prácticas sugieren 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, específicamente para negocios como el de CIMEX, almacenar la información en niveles de detalles altos, es decir, con bajos niveles de granularidad debido a que podría ser beneficioso para empresas como la en cuestión requieran altos niveles de detalles para el análisis de su información y llevar a cabo un correcto proceso de toma de decisiones.

Después del análisis anterior se puede concluir que la granularidad del repositorio central de la solución propuesta está dada por el registro de cada uno de los procesos principales del negocio que en anteriores epígrafes se han descrito, en un día.

Propuesta del Modelo Multidimensional del ODS

Una vez definido dentro del negocio las dimensiones, las medidas, los cubos OLAP de datos y la granularidad, se procede a la estructuración del modelo o los modelos multidimensionales 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.

La idea fundamental del modelo multidimensional es que los datos de negocio pueden ser representados como un tipo de cubo de datos. En los cubos cada celda contiene un valor y las aristas del cubo definen dimensiones naturales de análisis. Como se especificó en la fundamentación teórica existen diversos esquemas del modelo multidimensional pero los más reconocidos son 3 específicamente: Modelo Estrella, Modelo Copo de Nieve, Modelo Constelación de Estrellas.

En la solución que se propone se seleccionó el Modelo Tipo Estrella para el desarrollo de la misma, donde tendrán lugar varias tablas centrales o principales, estas son:

Ods_Ajuste; relacionada de manera directa con 6 tablas dimensionales.

Ods_Compra; relacionada de manera directa con 6 tablas dimensionales.

Ods_Inventario; relacionada de manera directa con 4 tablas dimensionales.

Ods_Existencia; relacionada de manera directa con 4 tablas dimensionales.

Ods_Transferencia; relacionada de manera directa con 6 tablas dimensionales.

Ods_Venta; relacionada de manera directa con 8 tablas dimensionales.

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 sentido dentro del negocio simplemente es un número que garantiza la conexión.

La estructura de las tablas del Sistema Transaccional Sentai es extremadamente compleja y no está enfocada a los temas que resultan de interés para el analista. Por esto se propone para el ODS un diseño conceptual basado en el Modelo Dimensional, como se sugiere en **1.C**, donde cada diagrama se corresponde con los procesos que fueron identificados a partir de las necesidades existentes[25].

El nivel de detalle que se pretende alcanzar es el de las transacciones que ocurren en cada módulo, separándolas de acuerdo a su tipo, ya sean compras, ventas, etc., que van a determinar el contenido de cada una de las tablas de hechos generadas a partir de los procesos identificados. Las dimensiones de cada proceso van a estar totalmente desnormalizadas.

Basándose en este modelo lógico, se elaboró un diseño físico de bases de datos que fuera consecuente con el primero. Al ser un proyecto tan ambicioso, se propuso comenzar desarrollando un ODS de Clase III como se recomienda en **1.B**, por lo que la frecuencia de actualización será diaria, y los procesos de carga se realizarán durante las madrugadas según lo sugerido en **1.A**.

[25] Ver en el Anexo 4 los diagramas dimensionales de los diferentes procesos del negocio identificados.

Desarrollar el modelo físico

El desarrollo exitoso de un modelo físico es necesariamente un aspecto invaluable dentro de la construcción de un Almacén de Datos, cualesquiera que este sea, un Almacén de Datos Operacionales o un Datawarehouse propiamente dicho. El punto de partida para llevar a cabo este modelo es definitivamente el modelo lógico, pues el primero debe ser un espejo irrefutable de este último. Sin embargo en ocasiones se realizan transformaciones en la estructura física de las tablas y las columnas, pues esto facilita el trabajo y la comodidad con el Sistema Gestor de Bases de Datos Relacionales y las herramientas que previamente fueron seleccionadas para realizar el acceso a los datos. Por varias razones en el Modelo Físico se recogen tablas de mantenimiento de las bases de datos que por lo general no son descritas en el modelo lógico de la solución.

La mayor diferencia entre ambos modelos es la especificación detallada y exhaustiva de todas las características físicas de la Base de Datos, entre varios, se pueden citar los tipos de datos de las tablas y sus parámetros de almacenamiento.

En la solución específicamente, para el desarrollo de este modelo se utilizó Visual Paradigm 3.1, entre las ventajas que facilita el uso de herramientas se muestran claramente, 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. Visual Paradigm 3.1 es una herramienta de modelado que le ayuda a diseñar, generar y mantener aplicaciones de base de datos de calidad y alto rendimiento. Desde un modelo lógico de sus requerimientos de información y reglas del negocio que definen su base de datos, hasta un modelo físico optimizado por las características específicas de su base de datos de destino, esta herramienta le permite visualizar la estructura adecuada, los elementos clave y un diseño optimizado de su base de datos.

En relación al modelo físico, una vez que este se tiene ya definido en la “herramienta case” 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. Respecto a este particular paso, en la solución ya desde la creación del modelo lógico se acotaron los aspectos necesarios para que cuando se generara en una ulterior fase el físico no se realizaran variadas transformaciones. 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 el DW 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.

Estrategia Inicial de Indexado

Las demoras del sistema ante operaciones que involucren un gran volumen 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. Muchas veces 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 ODS 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 e interesara leer sobre un tema en específico, se tendría que recorrer cada página hasta encontrar lo buscado, llegando necesariamente hasta el final, pues no se podrían determinar cuándo se ha encontrado la última referencia al tópico de interés. Este proceso, definitivamente, haría de 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[26], buscando las que cumplen la condición. Si se tuviera una estructura que, al igual que un índice en un libro, guía 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[27].(England and Powell 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

[26] Este proceso es denominado método de acceso secuencial (*sequential access method*).

[27] Para profundizar en el tema de los índices clusterizados y no clusterizados, se recomienda consultar (England and Powell 2007)

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 ODS 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 valuación 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 join 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 expresión43 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 ODS:

2.A

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 con las que se cuenta. Se recomienda, además, desactivar los índices durante los procesos de carga.

La solución de este ODS posee implementado un indexado, el que trae por defecto el gestor PostgreSQL para la búsqueda de datos utilizando las llaves primarias y foráneas. Todas las llaves primarias, que son llaves surrogadas, 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. El ejemplo más claro que posee el sistema para entender esto es la relación existente entre las tablas de hechos con las dimensiones ya que utiliza el método “B-Tree” para optimizar la recuperación de los datos cuando se le realicen cortes a la información almacenada. La utilización de más índices sobre la tabla de hechos u otras estructuras no es una tarea sencilla debido a que un consumo excesivo de índices lejos de mejorar el rendimiento del sistema atenta de forma directa a su ralentización.

Diseño y construcción de la Instancia de Base de Datos

Los Sistemas Gestores de Bases de Datos Relacionales, comprenden dentro de su arquitectura interna a los Almacenes de Datos, y estos sistemas a su vez se encuentran determinados dentro de un servidor físico, para que todo el sistema y la gama de parámetros, como la memoria, puedan servir de óptimo rendimiento en función de los requerimientos establecidos por el ODS. Aunque en numerosas ocasiones los ajustes varían debido a especificaciones dentro de los propios SGBDR, así como de los mismos Almacenes de Datos, independientemente de los ajustes que se puedan realizar, existen parámetros que son de vital importancia para el buen funcionamiento del Almacén y su óptimo rendimiento. Por lo general los parámetros que tienen amplias probabilidades de ser ajustados en la base de datos, estos evolucionan y crecen.

Este paso que se describe, tiene como objetivo principal garantizar la existencia de los requerimientos físicos mínimos necesarios para el buen funcionamiento del ODS. Uno de los parámetros más importantes y necesarios parámetros para el Almacén es la adecuada disponibilidad de memoria, ya que en él se realizan numerosas y complejas consultas; pues son múltiples y demasiado grandes las tablas sobre las cuales se realiza operación de *join,* para lograr una respuesta eficiente por parte del sistema. Esto implica que la solución propuesta necesite de al menos 1Gb 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, ya que solo se realizarán operaciones con los seis principales procesos del negocio que con anterioridad se han descrito en otros epígrafes.

Desarrollar la Estructura Física de Almacenamiento

Muchas veces se habitúa a utilizar, en el ambiente de las bases de datos, términos como tabla, fila o columna, sin pensar en la estructura física que subyace. Se asume generalmente, que si se añade filas, estas son almacenadas en tablas. Sin embargo, estos son conceptos que el Sistema Operativo no maneja. Entonces, ¿cómo se almacena realmente una fila o una tabla? La respuesta depende en gran medida de cómo funcione cada gestor de bases de datos: algunos, por ejemplo, crean un fichero independiente para cada tabla; en otros, un fichero puede ser compartido por varias de éstas. Pero en esencia, se almacenan en ficheros.

Cada fichero está dividido en páginas y en cada una de ellas se puede almacenar un número fijo de filas. Este número lo van a determinar dos factores: el tamaño de la página[28] y la longitud de las filas. Las páginas no tienen por qué estar completamente llenas, sino que pueden contener espacios en blanco, originados, posiblemente, por eliminaciones de filas previamente almacenadas. Para la inserción de nuevos datos son usadas diferentes técnicas. Algunos gestores asumen que siempre una fila nueva es añadida detrás de la última fila en la página final. En caso de que dicha página esté llena, una página

[28] El tamaño de la página depende del Sistema Operativo y del Gestor de Bases de Datos. Tamaños como 2K, 4K, 8K y 32K son muy comunes.

vacía es agregada al fichero y en ella se colocarán los nuevos datos. Otros gestores llenan de manera automática los espacios intermedios, haciendo uso de algoritmos de manejo de páginas más sofisticados.

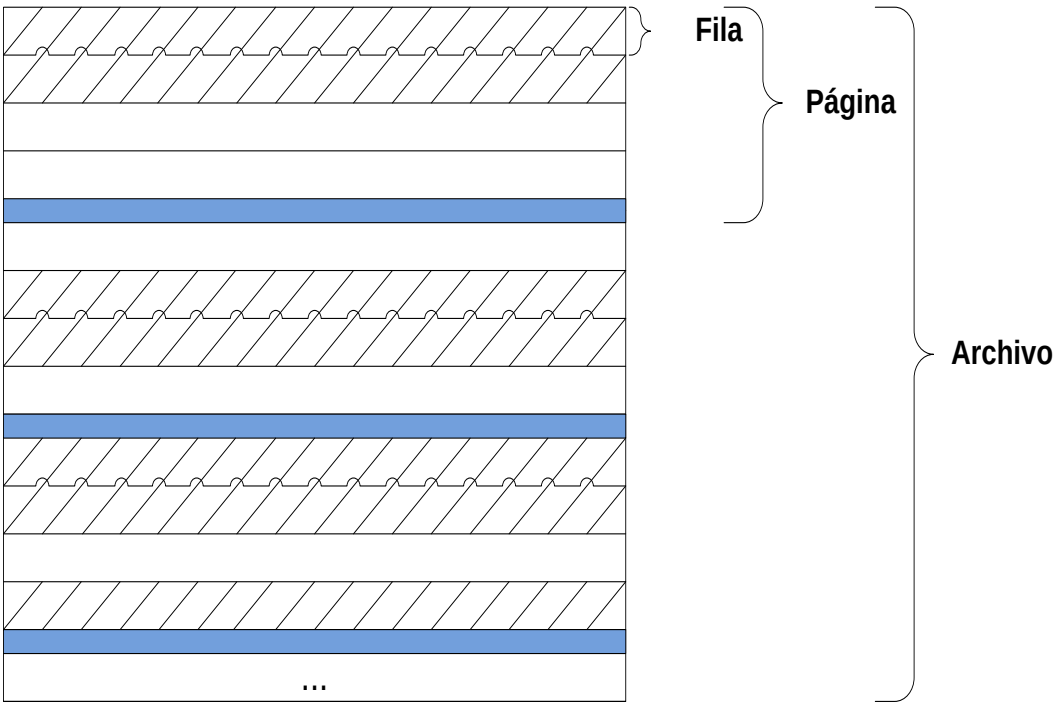

Figura 17: Posible estructura de un fichero donde se almacenan datos.[29]

Es importante aclarar que las páginas forman la unidad de Entrada/Salida: si el Sistema Operativo recupera datos en el disco duro, esto es hecho página por página. Por lo tanto, un gestor de bases de datos puede solicitar al Sistema Operativo una página dentro del fichero, pero sólo una fila, no. Para lograr recuperar una fila, son necesarios dos pasos: primero, deberá recuperarse del disco la página donde se encuentra dicha fila. La manera de identificar una página es usando el identificador del fichero donde se encuentra, más el número que tiene esa página dentro del fichero. Luego, se deberá buscar la fila dentro de la página. Este último paso puede hacerse de diferentes maneras. Una vía es examinar toda la página hasta encontrar la fila deseada. Como este proceso se realiza completamente en memoria, es llevado a

[29] Se representan las páginas separadas por una línea azul, y las filas sin datos en blanco.

cabo relativamente rápido. Otro método más directo es el siguiente: cada página puede contener una lista de entidades enumeradas en las que se puede encontrar la ubicación de todas las filas almacenadas en dicha página. Cada fila tendrá, entonces, un identificador único, formado por el identificador de la página y el número que tiene esa fila en la lista. Por lo tanto, el procedimiento para encontrar una fila sería: primero seleccionar la página correcta y luego acceder a la ubicación especificada en la lista para el identificador de dicha fila.

Como se ha visto, manejar la información almacenada implica efectuar distintas operaciones en el nivel físico, en las que se combinan las funciones a realizar en disco, con las capacidades en cuanto a memoria y CPU. Algunos de los factores que pueden provocar los llamados “cuellos de botella” en esta capa son los accesos o búsquedas en disco y las operaciones de lectura y escritura en el mismo.

Monitorización del Uso

Este paso llamado Monitorización del Uso, es el último que plantea Kimball dentro de su metodología, para el Diseño Físico del Almacén. La implantación de este paso dentro del Sistema ODS contribuye al seguimiento de las respuestas a consultas, reportes y la carga de los datos hacia el almacén. Esta información es particularmente de vital importancia durante el desarrollo del Almacén de Datos y los sistemas de mantenimiento. Este paso se encuentra regido por cuatro áreas fundamentales: rendimiento, soporte a usuarios, marketing y planificación.

Los datos obtenidos de la monitorización son usados para identificar cuáles son las tablas y columnas más expensas a realizar sobre ellas excesivas operaciones de join, seleccionadas, agregadas y filtradas. Esta información permite la inclusión de índices y esquemas al ODS con el fin de mejorar el tiempo de respuesta del sistema y que de esta forma cumpla la misión para la cual fue destinado.

El seguimiento al crecimiento, promedio de tiempo de respuestas de las consultas, 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. Esto conlleva a perfeccionar la estrategia que inicialmente se tiene sobre el indexado a utilizar. 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 más en profundidad

los resultados arrojados por las pruebas realizadas. Pero se pudiera vaticinar, pues producto a todo el modelado que se ha realizado del negocio resulta evidente que las tablas Fecha, Producto y Proveedor, serán las más concurrentes por los usuarios finales del negocio de Cimex, pues son actualmente tablas dimensionales que se encuentran relacionadas con todos y cada uno de los principales procesos del negocio de la entidad.

Presentación de la Información

De que valdría hacer todo aquello que se ha preparado, si no se puede mostrar las maravillas que puede ser capaz de hacer un ODS. La presentación de la información a los usuarios finales del negocio es una de las partes más importantes de todo este proceso que ya muestra sus primeros frutos, el Front End, como se suele denominar a esta parte, es el refinamiento de una obra esplendida en la que un artista ha trabajado, se le denomina así; pues es la interfaz que interactúa con los usuarios para mostrarle los datos de manera fácil, con el formato adecuado y en un ambiente amigable donde se facilite el análisis.

Sería entonces la interfaz de publicación con que cuenta en Mondrian la que se utilizará en este sentido. De tal manera que los usuarios que se encargarán de elaborar las consultas y reportes tienen que poseer conocimientos básicos de consultas MDX y de OLAP para construir las consultas, además del lenguaje XML debido a que la arquitectura Mondrian trabaja mediante este estándar[30]. Esta interfaz va a ser configurada inicialmente con los principales reportes que en Cimex se realizan para que se utilicen como base para la creación de todo el universo de reportes candidatos por sus especialistas.

A continuación, y para un mejor entendimiento de cómo realiza las funciones esta potente herramienta, se brinda una breve descripción de su arquitectura en tres capas.

La capa de presentación determina lo que el usuario final ve en su monitor, y cómo él puede interactuar para pedir nuevas consultas. Hay muchas formas de presentar datos multidimensionales, incluyendo tablas pivote, pastel, líneas y gráficas de barras, y avanzadas herramientas de visualización tales como

[30] Ver Anexo 6: Ejemplo de XML generado por el workbench, que luego será interpretado por el Mondrian

hacer clic en los mapas y gráficos dinámicos. Estos pueden ser escritos en Swing o JSP, gráficos prestados en el formato JPEG o GIF, o de su transmisión a una aplicación remota a través de XML. Lo que todas estas formas de presentación tienen en común es el carácter multidimensional "gramática" de las dimensiones, medidas, en esta capa las consultas que se realicen el servidor OLAP las devuelve.

La segunda capa es la capa lógica. La capa lógica analiza, valida y ejecuta las consultas MDX. Una consulta es evaluada en múltiples etapas. Los ejes se calculan en primer lugar, los valores de las tuplas dentro de los ejes. Por eficiencia, la capa lógica envía las solicitudes a la capa de datos en lotes. Un transformador de consulta permite que la aplicación manipule las consultas, en lugar de construir una declaración de MDX desde cero para cada solicitud. Esta capa describe el modelo de las dimensiones. Además es responsable de mantener un total de caché. Una agregación es un conjunto de valores de medida en la memoria, calificado por un conjunto de dimensión de los valores de las columnas. La capa lógica envía las solicitudes de grupos de tuplas. Si las tuplas no están en la caché, o implícita por la rodadura de un conjunto en la caché, el administrador lógico envía una solicitud a la capa de almacenamiento.

La capa de datos o de almacenamiento como también se conoce es la encargada de proporcionar los datos con los que se trabajará en las capas superiores de la arquitectura.

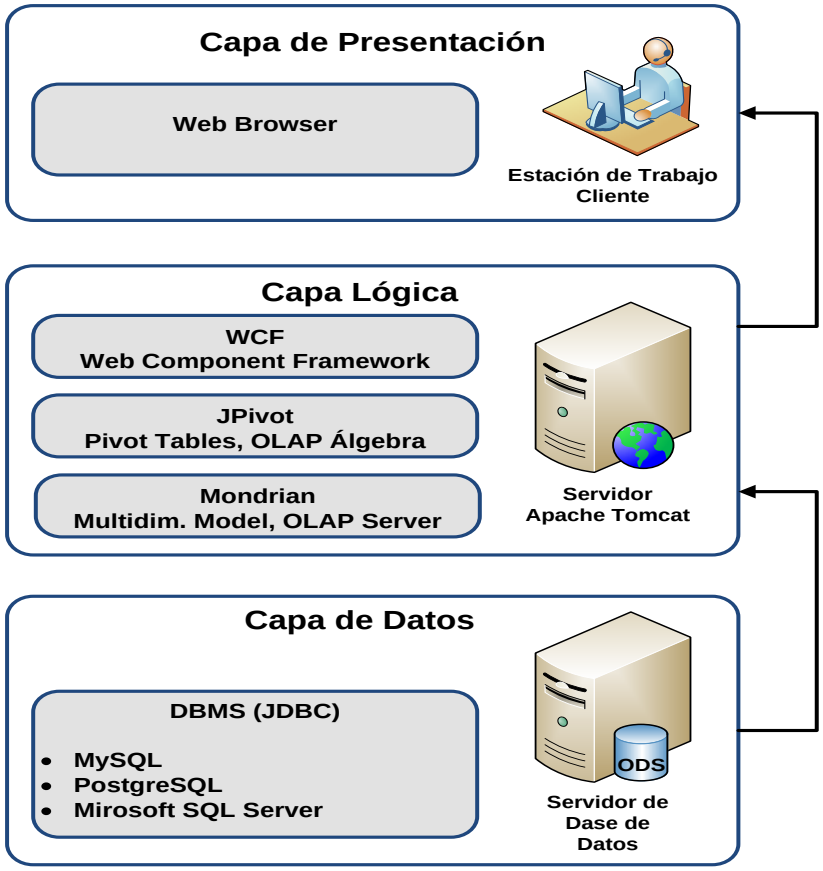

Figura 18: Arquitectura en 3 Capas del Mondrian

Conclusiones del Capítulo 2

En el presente capítulo se han descrito detalladamente cómo interactúan los componentes del Operational Data Store y la arquitectura que se ha propuesto para su desarrollo. En el mismo ha quedado de manifiesto la interacción de los distintos factores claves para el logro de un buen diseño así como los elementos relacionados con el almacenamiento de la información, los cubos de datos, la granularidad y el modelo multidimensional. Por último y muy importante se abordan los diferentes pasos para la implementación de este Sistema, estando estos muy bien adaptados a las condiciones y características del negocio en cuestión, por lo que se han cumplido las metas propuestas para este capítulo de diseño e implementación.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

Luego de haber recorrido algunas de las principales etapas del proceso de desarrollo de un ODS, nada mejor que poner en práctica las sugerencias brindadas. Con este objetivo, tomamos como base un trabajo que se está realizando en la Corporación CIMEX, en el cual se han ido aplicando las indicaciones de la metodología propuesta. En la concepción de dicho trabajo se han seguido las pautas del Ciclo de Vida del Software[31], el que se ha particularizado añadiendo los elementos ya discutidos, una vez que fue identificada la necesidad de crear un ODS. Para sustentar la decisión asumida, se ha tomado como base la metodología propuesta por Kimball del Ciclo de Vida Dimensional del Negocio durante el desarrollo de un Almacén de Datos[32], pero adaptado al entorno actual. Aunque a lo largo de este trabajo se han seguido los pasos del ciclo de desarrollo formulado, no es objetivo detallar cada uno de ellos, sino solamente resaltar aquellos aspectos en los que se avala la propuesta teórica - práctica realizada en capítulos anteriores.

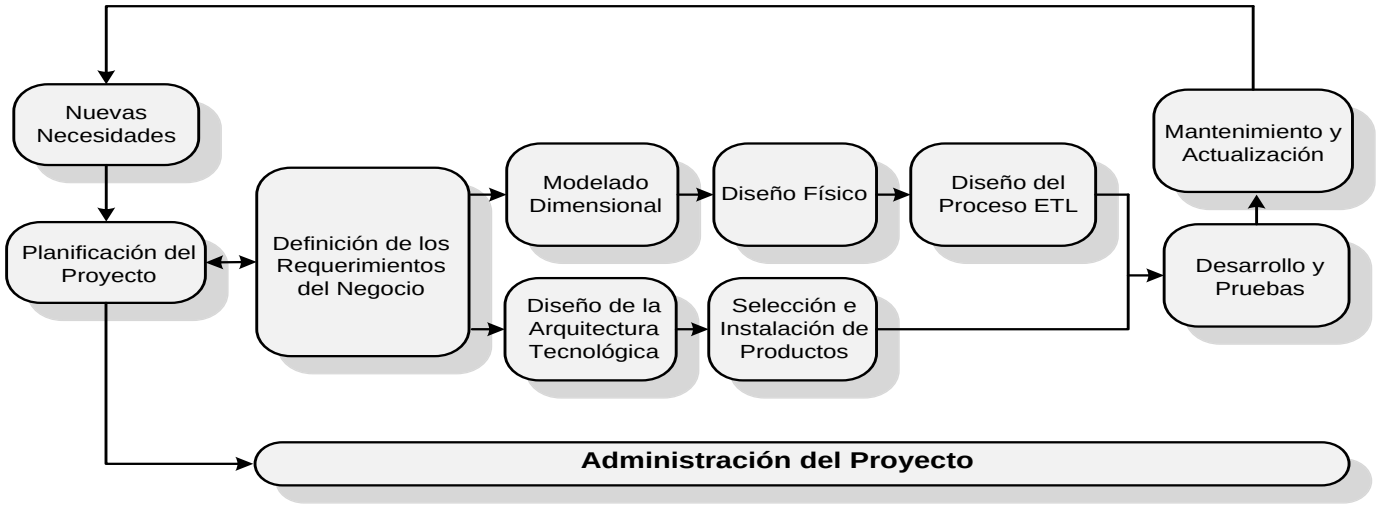

Figura 19: Diagrama del Ciclo de desarrollo del ODS

[31] El Ciclo de Vida del Software consta de varias etapas: análisis de requerimientos, diseño, implementación, pruebas y mantenimiento y actualización.

[32] Kimball brinda esta propuesta en la Segunda Edición de su libro “The Data Warehouse Toolkit: the complete guide to dimensional modeling”.

Validación del Sistema

Después de haber concluido una primera iteración de ciclo de desarrollo del futuro Almacén de Datos Operacionales, 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:

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

En el proyecto esta etapa duró aproximadamente 4 meses donde estuvo involucrada la principal cliente del mismo que es uno de los administradores de bases de datos del grupo de almacenes de datos del departamento DataCimex de la Corporación CIMEX, Osmara Ramos Vidot, mediante un sistema de chequeo quincenal donde se le presentaba los principales resultados y se definían las posibles funcionalidades a agregar, independientemente que se le mostraba el avanza en los trabajos de diploma referentes al proyecto.

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 medidas autocalculados como: importe, cantidad, etc.
- Posibilidad de obtener totales, promedios, porcientos, máximos, mínimos durante la realización de consultas OLAP.

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 familiarización de los analistas del negocio con él.

Análisis del tamaño, crecimiento y calibrado del ODS

Tomando como punto de partida, un estimado razonable, él cual se realizará teniendo en cuenta el tamaño de la base de datos, entonces se podrá tener una concepción más real de las dimensiones espaciales de la base de dato que formará parte indisoluble del ODS. En este sentido se profundizó en el análisis de las dimensiones del negocio, para calcular la cantidad de unidades existentes en cada una de ellas, número de filas implicadas, entre otras, esto hasta llegar al número de Bytes que serán ocupados por concepto de tamaño.

La actividad siguiente es estimar por cada tabla la cantidad de filas que podrá tener cuando el histórico esté cargado completamente en el sistema. En la tabla 2 se muestra la cantidad de filas estimada por cada tabla propuesta.

Dimensiones		**Hechos**	
Nombre de la Tabla	**Cantidad de Filas**	**Nombre de la Tabla**	**Cantidad de Filas**
Ods_TipoAjuste	500	Ods_Ajuste	2 281 250
Ods_Localidad	10000	Ods_Compra	4 730 400
Ods_Cliente	20000	Ods_Existencia	1 460 000
Ods_FuenteVenta	50000	Ods_Inventario	182 500
Ods_Area	10000	Ods_Transferencia	4 730 400
Ods_Fecha	365	Ods_Venta	368 073 300
Ods_Producto	200000		
Ods_Proveedor	100000		
Ods_Transaccion	1000000		
Ods_Documento	1000		

Tabla 2: Estimación de cantidad de filas por tablas del ODS

Filas aproximadas por cada Dimensión:

- *Dimensión Área:* consta de 10000 elementos de los cuales, 10 están relacionado con Ajustes, 10 con Compras, 5 con Inventario, 10 con Transferencias y 10 con Ventas.
- *Dimensión Fecha:* 1 año * 365 días = 365 días
- *Dimensión Producto:* consta de 200000 elementos de los cuales, 5 están relacionados con Ajustes, 13 con Compras, 10 con Inventario, 20 con Existencia, 13 con Transferencia y 15 con Ventas.
- *Dimensión Proveedor:* consta de 100000 elementos de los cuales, 5 están relacionados con Ajustes, 13 con Compras, 10 con Inventario, 20 con Existencia, 13 con Transferencia y 15 con Ventas.
- *Dimensión Tipo de Ajuste:* consta de 50 elementos, los mismos estarán relacionados única y exclusivamente con Ajuste en 5.
- *Dimensión Transacción:* consta de 1000000 elementos de los cuales, 5 estarán relacionado con Ajustes, 13 con Compra, 13 con Transferencia y 15 con Ventas.
- *Dimensión Documento:* consta de 100 elementos, puesto que se relacionan con Compra, Transferencia y Venta con cardinalidad repetida, entonces, será 13, 13 y 15 respectivamente.
- *Dimensión Localidad:* consta de 10000 elementos de los cuales, 10 se relacionan con Existencia.
- *Dimensión Cliente:* consta de 20000 elementos de los cuales, 15 están relacionado con Venta.
- *Dimensión Fuente de Venta:* consta de 50000 elementos de los cuales, 13 están relacionados con Venta.

Filas aproximadas por cada Tabla de Hechos:

- Ajuste: 10*365*5*5*5*5 = **2 281 250**
- Compra: 10*365*6*6*6*6 = **4 730 400**
- Existencia: 365*20*20*10 = **1 460 000**
- Inventario: 5*365*10*10 = **182 500**
- Transferencia: 10*365*6*6*6*6 = **4 730 400**
- Venta: 10*365*7*7*7*7*7*6 = **368 073 300**

Total de Campos Claves en las Tablas de Hechos:

- Ajuste: 7
- Compra: 7
- Existencia: 5
- Inventario: 5
- Transferencia: 8
- Venta: 9

Total de Campos Medidas en las Tablas de Hechos:

- Ajuste: 5
- Compra: 31
- Existencia: 14
- Inventario: 7

- Transferencia: 9
- Venta: 13

Total de Campos en las Tablas de Hechos:

- Ajuste: 12
- Compra: 38
- Existencia: 19
- Inventario: 12
- Transferencia: 17
- Venta: 22

En casi todo los DW el tamaño de las tablas de dimensiones es realmente insignificante comparado con el tamaño que alcanzarán las tablas de hechos y el segundo elemento más significativo son los índices utilizados para la optimización del rendimiento.

Para un mejor entendimiento, en lo que a estimación crecimiento se refiere, en la siguiente tabla se muestra el cálculo realizado por la propuesta presentada por el MSc Rosendo Moreno Rodríguez, de la Universidad central de las Villas “Marta Abreu” y colaborador del Centro de Tecnologías de Análisis y Almacenamiento de Datos, como resultado de la interpretación del libro “Data Warehouse Toolkit” de Ralph Kimball.

Nombre del Objeto	**Tipo**	**Tamaño Inicial**	**Tamaño Proyectado**	**Tamaño Máximo**
Ods_Ajuste	Table	0.1K	2.1M	0.6K
Ods_Area	Table	0.7K	0.7K	17.6K
Ods_Cliente	Table	1.0K	1.0K	21.9K

Ods_Compra	Table	0.2K	3.2G	6.9K
Ods_Documento	Table	0.2K	0.2K	2.1K
Ods_Existencia	Table	0.1K	42.5M	2.3K
Ods_Fecha	Table	0.5K	174.1K	11.9K
Ods_FuenteVenta	Table	0.2K	0.2K	2.0K
Ods_Inventario	Table	0.0K	17.4M	0.6K
Ods_Localidad	Table	0.7K	0.7K	17.9K
Ods_Producto	Table	1.9K	1.9K	94.8K
Ods_Proveedor	Table	0.8K	0.8K	20.4K
Ods_TipoAjuste	Table	0.2K	0.2K	1.3K
Ods_Transaccion	Table	0.3K	89.8M	2.8K
Ods_Transferencia	Table	0.1K	2.4G	2.3K
Ods_Venta	Table	0.1K	2.4G	2.6K
Total		**7.5K**	**8.1G**	**208.0K**

Tabla 3: Estimación de crecimiento de la BD en 1 Año.

Normalización

La normalización, dentro del universo de los diseños relacionales se justifica y obtiene 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 y millones de tuplas, no se recomienda la normalización, ya sea, a nivel de dimensiones como de tablas de hechos. En el sistema desarrollado las tablas de hechos propuesta se relacionan con cada una e indistintamente de las 10 dimensiones identificadas y, a su vez, son el único vínculo de 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.

Pruebas de Volumen y Carga

Se puede encontrar un conjunto de varias formas para realizar posibles a un sistema informático para validar su uso, ejemplo de ellas se pueden mencionar: integración, pruebas de unidad, funcionalidad, sistema, volumen, carga, stress, etc; las que más impactan en el desarrollo de un ODS son las pruebas que tengan relación con el rendimiento, capacidad y concurrencia de la base de datos. En este sentido las pruebas que se realizaron al Almacén de Datos Operacionales fueron 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 datos aleatorios generados por una herramienta llamada EMS Data Gerator, esta herramienta colma a la base de datos de una cantidad determinada de datos previamente establecida por quién esté realizando el llenado de datos en este caso, el propio desarrollador. Se configuró esta generación de datos con datos arbitrarios, pero coincidentes en cuanto a sus tipos y volúmenes con los datos reales que maneja la entidad. El uso de este generador se pudiera considerar una prueba más, ya que si existen inconsistencia en el diseño de la Base de Datos, este no comienza el poblado de datos hasta tanto no quede un correcto diseño.

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 ODS.

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 CIMEX nunca existirán más de 25 usuarios registrados en el servidor y, en general, la concurrencia será mínima. Las consultas se realizarán sobre las tablas de hechos o procesos del negocio, los mismos que fueron definidos con anterioridad. 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.

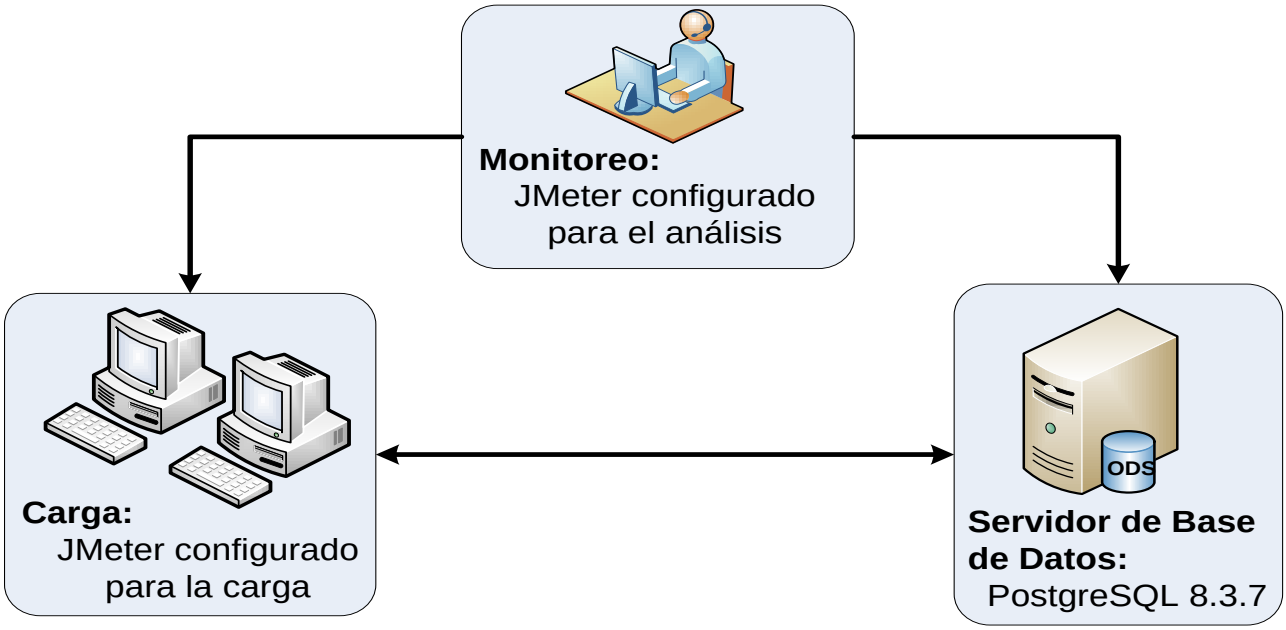

Figura 20: Configuración para las pruebas de Carga

Prueba No.1 Comportamiento de los indicadores en el proceso del negocio Compra

- ✓ Dimensiones involucradas: Producto, Producto, Fecha, Área, Transacción, Docuemnto
- ✓ Cantidad total de filas: 20000 filas
- ✓ Cantidad de filas recuperadas: 17973 filas

- ✓ Consulta: SELECT * FROM public."Ods_Compra" WHERE (public."Ods_Compra"."Area_IdODS" >= 1000)

- ✓ Cantidad de Usuarios:

  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	0,65	0,75	0,20	0,91

  - 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	1,52	1,73	0,82	1,80

  - gráfico de relación entre pruebas

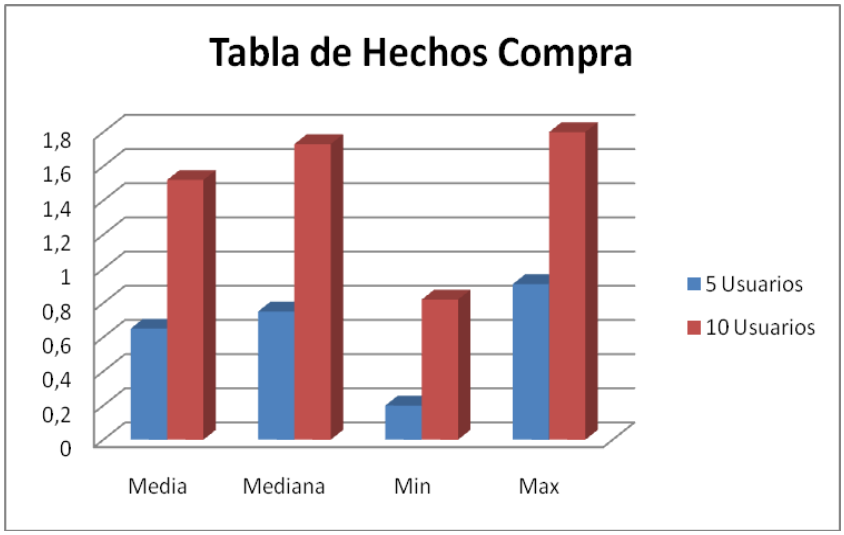

Gráfico 1: Representación de la prueba 1

Prueba No.2 Comportamiento de los indicadores en el proceso del negocio Venta

- ✓ Dimensiones involucradas: Producto, Producto, Fecha, Área, Transacción, Docuemnto, Fuente de Venta, Cliente

- ✓ Cantidad total de filas: 20000 filas
- ✓ Cantidad de filas recuperadas: 17991 filas
- ✓ Consulta: SELECT * FROM public."Ods_Venta" WHERE (public."Ods_Venta"."Trans_IdODS" >= 1000)
- ✓ Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	0,67	0,78	0,24	0,99

  - 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	1,57	1,78	0,88	1,91

  - gráfico de relación entre pruebas

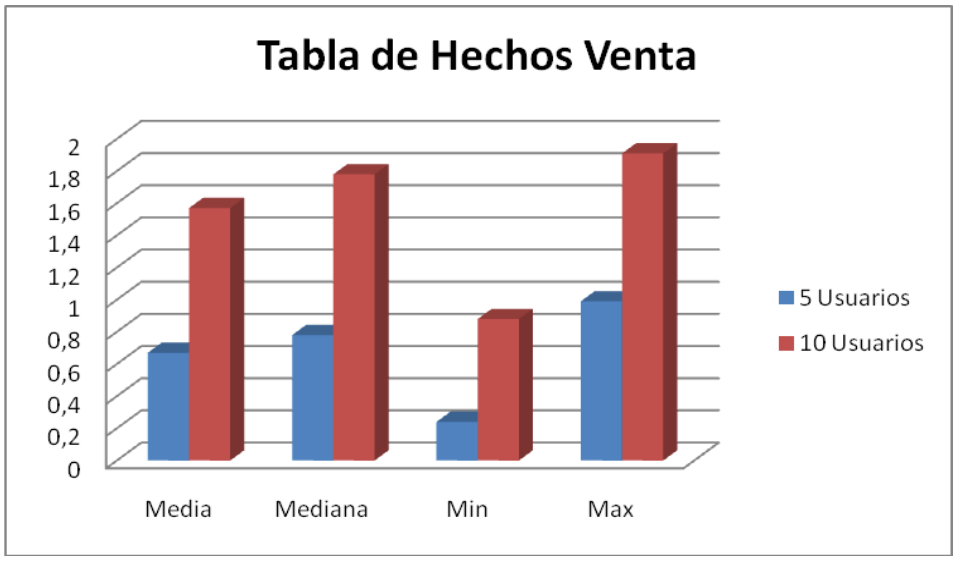

Gráfico 2: Representación de la prueba 2

Prueba No.3 Comportamiento de los indicadores en el proceso del negocio Ajuste

- ✓ Dimensiones involucradas: Producto, Producto, Fecha, Área, Transacción, Tipo de Ajuste
- ✓ Cantidad total de filas: 10000 filas
- ✓ Cantidad de filas recuperadas: 9495 filas
- ✓ Consulta: SELECT * FROM public."Ods_Ajuste" WHERE (public."Ods_Ajuste"."Area_IdODS" >= 500)
- ✓ Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	0,31	0,46	0,21	0,68

  - 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	1,02	1,09	0,69	1,53

  - gráfico de relación entre pruebas

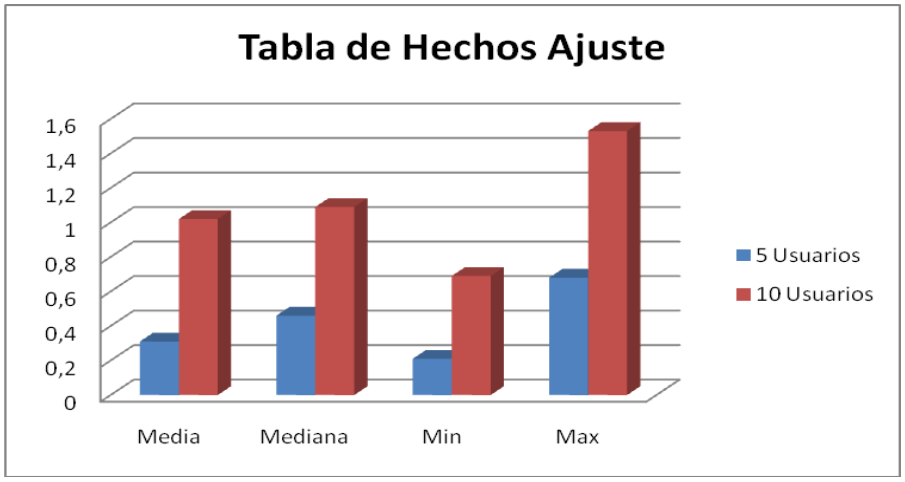

Gráfico 3: Representación de la prueba 3

Prueba No.4 Comportamiento de los indicadores en el proceso del negocio Inventario

- ✓ Dimensiones involucradas: Producto, Producto, Fecha, Área
- ✓ Cantidad total de filas: 10000 filas
- ✓ Cantidad de filas recuperadas: 9748 filas
- ✓ Consulta: SELECT * FROM public."Ods_Inventario" WHERE public."Ods_Inventario"."Prod_IdODS" >= 500)
- ✓ Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	0,34	0,47	0,23	0,69

  - 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	1,04	1,10	0,71	1,55

- gráfico de relación entre pruebas

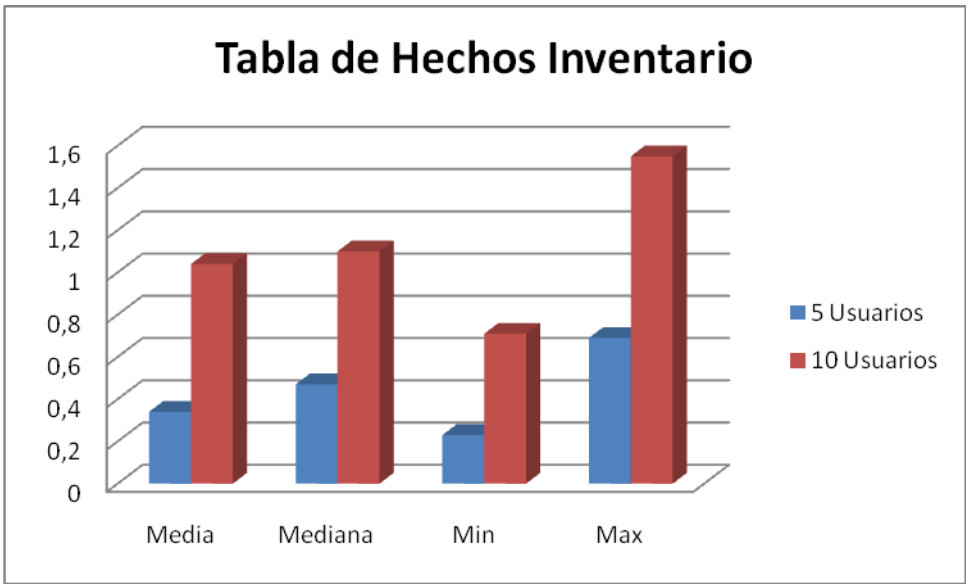

Gráfico 4: Representación de la prueba 4

Prueba No.4 Comportamiento de los indicadores en el proceso del negocio Inventario

- ✓ Dimensiones involucradas: Producto, Producto, Fecha, Área, Transacción , Documento
- ✓ Cantidad total de filas: 100000 filas
- ✓ Cantidad de filas recuperadas: 97169 filas
- ✓ Consulta: SELECT * FROM public."Ods_Transferencia" WHERE (public."Ods_Transferencia"."Fecha_IdODS" > 10) AND (public."Ods_Transferencia"."Fecha_IdODS" < 1000)
- ✓ Cantidad de Usuarios:
  - 5 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	0,78	0,96	0,51	1,08

- 10 usuarios concurrentes

	Media (seg)	Mediana (seg)	Mín (seg)	Max (seg)
Resultados	1,16	1,32	1,19	2,04

- gráfico de relación entre pruebas

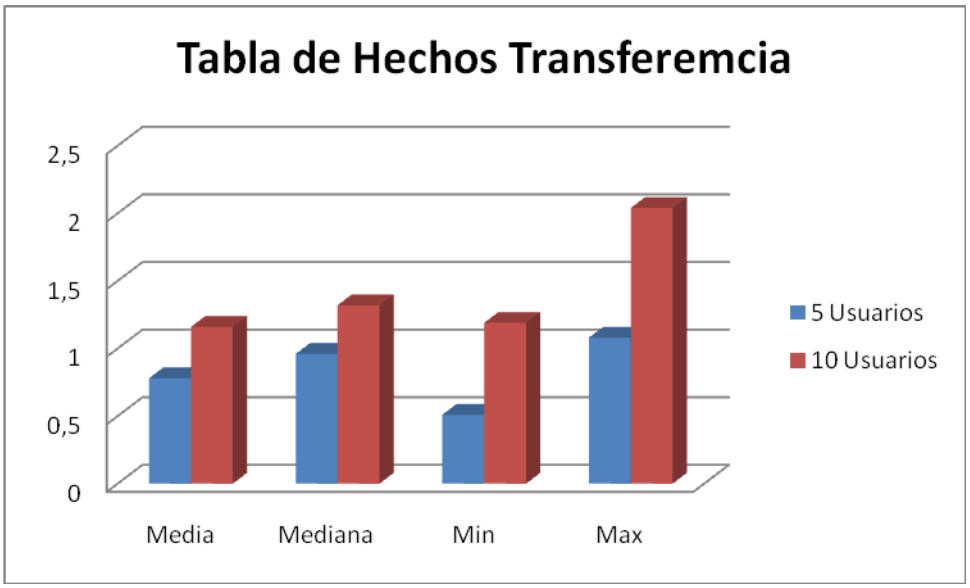

Gráfico 5: Representación de la prueba 5

Conclusiones del Capítulo 3

Al concluir este capítulo se puede plantear que se han cumplido los objetivos propuestos para el mismo, superando al mismo tiempo las expectativas esperadas. Se ha demostrado fehacientemente cuán optimo resulta ser el acceso a los datos en un Datawarehouse, el dinamismo y la rapidez de los reportes y consultas, y todo sobre la base de un estudio profundo del tamaño y crecimiento del Sistema, de su rendimiento, y de la validación del mismo, tanto por sus especialistas como por los usuarios finales.

CONCLUSIONES

Hoy en día el mundo se mueve vertiginosamente: el paso de una tecnología a otra y la rapidez requerida a la hora de dar respuesta a determinados problemas puede que se piense, en ocasiones, que no queda tiempo suficiente para ser formales durante el desarrollo de los proyectos. Por esto, seguramente en varias ocasiones se ha tenido que regresar al punto de partida, al no seguir una metodología que ayudara a definir cada detalle a tener en cuenta durante el largo camino a recorrer.

Precisamente ha sido objetivo de este trabajo evitar cometer tales errores durante el desarrollo de un Almacén de Datos Operacionales, intentando formalizar, desde sus primeras etapas, el arduo proceso que acompaña la creación de un sistema, alrededor del cual pueden surgir numerosas expectativas.

Se ha partido de una definición propia, elaborada tomando como base las ofrecidas por dos prestigiosos expertos, Inmon y Kimball, considerados como los “padres” de los Almacenes de Datos. Esta definición ayudó a precisar los objetivos reales de los Almacenes de Datos Operacionales dentro de la arquitectura de los sistemas de información, gracias a lo cual se pudo concretar determinadas normas a seguir.

Se pudo comprobar mediante las pruebas realizadas que el Gestor de Base de Datos utilizado, en este caso PostgreSQL 8.3.7 soporta los volúmenes de datos previsto desde un principio en el modelamiento de la base de datos.

Se analizaron los principales aspectos a tener en cuenta durante su diseño y desarrollo, y a la vez, brindando recomendaciones que pueden servir de guía a los desarrolladores, lo cual, desde una perspectiva modesta, se considera, es el principal mérito que tiene el trabajo realizado. Se tuvo la oportunidad, además, de aplicar nuestra propuesta a un sistema que se prevé brindará cuantiosos frutos a la corporación CIMEX a la hora de sustentar la toma de decisiones.

Se piensa, entonces que se han cumplido los objetivos que se trazaron desde un inicio. No obstante, el tema tratado es tan amplio que siempre quedan sugerencias por realizar.

RECOMENDACIONES

Con el propósito de enriquecer la propuesta realizada en este trabajo, se sugiere, desde el punto de vista teórico:

- ✓ Ampliar la metodología propuesta tomando en consideración algunos de los procesos que no fueron tratados a profundidad, como las purgas y las salvas en el Almacén de Datos Operacionales, así como la formalización de un modelo de completitud general.
- ✓ Realizar un profundo estudio de las técnicas de optimización, ya sean las descritas u otras, aplicadas sobre los Almacenes de Datos Operacionales.

Para validar la propuesta realizada en la práctica, se recomienda:

- ✓ Realizar un estudio comparativo de las necesidades de los analistas y el nivel de satisfacción que se alcanza en cada una de las capas planteadas, de manera que pueda comprobarse la validez de esta arquitectura de los sistemas de información en la corporación CIMEX.

BIBLIOGRAFÍA

1. Alarcón José Manuel, H. (2006) "Administración de SGBD PostgreSQL." **Volume**, DOI:
2. England, K. and G. Powell (2007). Performance Optimization and Tuning Handbook. Estados Unidos de América.
3. Hoobs, L. (2005). Oracle Data Base. Datawarehousing. United States.
4. Humphries, M. and M. Hawkins (2002). Data Warehousing: Architecture and Implementation. United States of America.
5. Imhoff, C. (2000, Julio). "A New Class of Operational Data Store." Revista DM Review.
6. Inmon, W. H. (1995, Febrero). "The Operational Data Store." InfoDB.
7. Inmon, W. H. (1998). "Designing the Operational Data Store." Information Management Magazine**2009,**(10 de Febrero).
8. Inmon, W. H. (2002). Building the Datawarehouse. Estados Unidos de América.
9. Kimball, R. (1996, Abril). "Slowly Changing Dimensions." Revista DBMS Online.
10. Kimball, R. (1997). "A Dimensional Modeling Manifesto." Revista DBMS Online **2009,**(10 de febrero).
11. Kimball, R. and M. Ross (2002). The Data Warehouse Toolkit. New York.
12. Pence, N. and R. Creeth (2002) "An Introduction to OLAP." **Volume**, DOI:
13. PentahoCorporation. (2005). "Pentaho BI Suite Enterprise Edition." Retrieved 10 de Marzo, 2009, from www.pentaho.com.

14. The PostgreSQL Global Development, G. (2009). PostgreSQL Conference, East 09, Philadelphia, Pennsylvania, United States.

15. Yunko Nakamura, O. (2007). "Sistemas Gestores de Bases de Datos." Revista de Posgrado. Universidad Atónoma de México.

ANEXOS

Anexo 1: Tabla comparativa ODS - DW

Criterio	ODS	DW
Orientación	Temas	Temas
Contenido	Información	Información
Actualización	Frecuente	No Frecuente
Intereses del Usuario	Operacionales	Gerenciales
Tipo de Análisis	Operacional	De Tendencias
Modelamiento	Multidimensional	Multidimensional
Historia de los Datos	Limitada, enfocada a períodos vigentes	Larga
Volatilidad de los Datos	Volátil	No Volátil
Nivel de Agregación	Bajo. Muy detallado	Alto

Anexo 2: Principales características de las Clases de ODS

Un ODS se puede clasificar en ODS de Clase I, II, III, o IV. A continuación se muestran las principales características que distinguen cada categoría.

	Clase I	**Clase II**	**Clase III**	**Clase IV**
Frecuencia de actualización	Segundos después de los cambios en los OLTP	De varios minutos a varias horas	Diaria	Cuando se desee
Método de intercambio y tecnología	Segundos después de los cambios en los OLTP	Método store and forward usando herramientas ETL	Método store and forward usando herramientas ETL	Herramientas ETL
Nivel de integración y transformación	Poco o ninguno	Moderado	Alto	Mínimo
Sumarizaciones	Ninguna[33]	Muy pocas[34]	Sí	Sí
Costo	Muy alto	De moderado a Alto	Moderado	Mínimo

[33] En un ODS de Clase I las sumarizaciones instantáneas pueden ser difíciles, por lo que se podrían realizar en varios intervalos a lo largo del día.

[34] En un ODS de Clase II se aplica lo mismo que para la Clase I, sin embargo, pueden haber sumarizaciones diarias, por ejemplo, realizadas durante la madrugada.

Anexo 3: 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.

Anexo 4: Descripción de las Tablas de Hechos

Especificación de Tablas de Hechos

Data Cimex

Data Cimex

1.0

Control de versiones

Fecha	Versión	Descripción	Autor
3/3/2009	1.0	Construcción del Documento	Darián González
20/4/2009	1.1	Revisión Final	Darián González

Introducción

El presente documento almacena una descripción detallada de las tablas de hechos incluidas en el proyecto. Además describe la granularidad y relación de las dimensiones con las mismas.

Definiciones, Acrónimos y Abreviaturas

Factor de Costo: Es una tabla contiene la configuración de todos los posibles factores de costos arancelarios que usted puede usar cuando está comprando mercancía.

Existencia: Es una descripción más detallada que un inventario ordinario, pues esta contempla compromisos, productos que ya han salido pero que no han entrado y viceversa., entre otras características

Tablas de Hechos: Son las tablas primarias en el modelo dimensional, donde es almacenado el rendimiento de las dimensiones numéricas del negocio

Referencias

Diseño de las tablas de hechos

Tabla de Hechos Ajuste.

Ajuste (ODS_Ajuste)

Declarar el gránulo.

Un área realiza en un día, un tipo de ajuste en una transacción sobre un producto establecido por un proveedor.

Seleccionar las dimensiones asociadas a Ajuste.

Dimensiones:

- Tipo de Ajuste (ODS_TipoAjuste)
- Transacción (ODS_Transaccion)
- Area (ODS_Area)
- Fecha (ODS_Fecha)
- Producto (ODS_Producto)
- Proveedor (ODS_Provedor)

Seleccionar los hechos de Ajuste.

Hechos:

- Cantidad (Ajuste_Cantidad)
- Importe (Ajuste_Importe)

Seleccionar Tabla de Hechos Compra.

Compra (ODS_Compra)

Declarar el gránulo.

En un día, un área se realiza la compra de determinado producto establecido por su proveedor en una transacción y mediante un documento que autorice.

Seleccionar las dimensiones asociadas a Compra.

Dimensiones:

- Documento (ODS_Documento)
- Fecha (ODS_Fecha)
- Transacción (ODS_Transaccion)
- Proveedor (ODS_Proveedor)
- Producto (ODS_Producto)
- Área (ODS_Area)

Seleccionar los hechos de Compra.

Hechos:

- Cantidad (Comp_Cantidad)
- Importe (Comp_Importe)

Seleccionar Tabla de Hechos Existencia.

Existencia (ODS_Existencia)

Declarar el gránulo.

En un día se realiza un control de existencia, en una localidad, sobre un producto establecido por un proveedor.

Seleccionar las dimensiones asociadas a Existencia.

Dimensiones:

- Fecha (ODS_Fecha)
- Localidad (ODS_Localidad)
- Producto (ODS_Producto)
- Proveedor (ODS_Proveedor)

Seleccionar los hechos de Existencia.

Hechos:

- Cantidad que hay en existencia (Exist_Cantidad)
- Órdenes de venta en existencia (Exist_OVenta)
- Órdenes de compra en existencia (Exist_OCompra)
- Transferencias de entrada en existencia (Exist_TransfEntrada)
- Transferencias de salida en existencia (Exist_TransfSalida)
- Préstamos en existencia (Exist_Prestamo)
- Ajustes de entrada en existencia (Exist_AjusteEntrada)
- Ajustes de salidas en existencia (Exist_AjusteSalida)
- Existencia de Asignación (Exist_Allocated)

Seleccionar Tabla de Hechos Inventario.

Inventario (ODS_Inventario)

Declarar el gránulo.

Se realiza el inventario de un producto establecido por un proveedor, en un día, de un área del negocio

Seleccionar las dimensiones asociadas a Inventario.

Dimensiones:

- Fecha (ODS_Fecha)
- Producto (ODS_Producto)
- Proveedor (ODS_Proveedor)
- Área (ODS_Area)

Seleccionar los hechos de Inventario.

Hechos:

- Cantidad en inventario (Inv_Cantidad)
- Importe de ese producto en inventario (Inv_Importe)

Seleccionar Tabla de Hechos Transferencia.

Transferencia (ODS_Transferencia)

Declarar el gránulo.

En un día se realiza una transacción desde un área, de un producto establecido por un proveedor, mediante un documento autorizador.

Seleccionar las dimensiones asociadas a Transferencia.

Dimensiones:

- Documento (ODS_Documento)
- Fecha (ODS_Fecha)
- Producto (ODS_Producto)
- Proveedor (ODS_Proveedor)
- Área (ODS_Area)
- Transacción (ODS_Transacción)

Seleccionar los hechos de Transferencia.

Hechos:

- Cantidad de la Transferencia (Transf_Cantidad)
- Importe de la Transferencia (Transf_Importe)

Seleccionar Tabla de Hechos Venta.

Venta (Ods_Venta)

Declarar el gránulo.

En un día se realiza una venta a un cliente de un producto establecido por un proveedor, el producto proviene de determinada fuente de venta, establecida en un área. Mediante un documento autorizador que apruebe la transacción.

Seleccionar las dimensiones asociadas a Venta.

Dimensiones:

- Proveedor (ODS_Proveedor)
- Área (ODS_Area)
- Fecha (ODS_Fecha)
- Documento (ODS_Documento)
- Cliente (ODS_Cliente)
- Producto (ODS_Producto)
- Transacción (ODS_Transacción)
- Fuente de Venta (ODS_FuenteVenta)

Seleccionar los hechos de Venta.

Hechos:

- Cantidad de la Venta (Venta_Cantidad)
- Importe de la Venta (Venta_Importe)
- Costo de la Venta (Venta_Costo)

Anexo 5: Fragmento de XML generado por Pentaho Schema Workbench para el proceso Ajuste.

```
- <Schema name="New Schema1">
  - <Cube name="ajuste" caption="Ajuste" cache="true" enabled="true">
      <Table name="Ods_Ajuste" schema="public" />
    - <Dimension type="StandardDimension" foreignKey="Prov_IdODS" name="proveedor" caption="Proveedor">
      - <Hierarchy name="jerarquías_de_proveedor" hasAll="true" allMemberCaption="jerarquías de proveedor"
          primaryKey="Prov_IdODS" primaryKeyTable="Ods_Proveedor" caption="Jerarquías de Producto">
          <Table name="Ods_Proveedor" schema="public" />
          <Level name="pais" table="Ods_Proveedor" column="Prov_Pais" nameColumn="Prov_Pais" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Pais" />
          <Level name="provincia" table="Ods_Proveedor" column="Prov_Provincia" nameColumn="Prov_Provincia" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Provincia" />
          <Level name="ciudad" table="Ods_Proveedor" column="Prov_Ciudad" nameColumn="Prov_Ciudad" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Ciudad" />
          <Level name="nombre" table="Ods_Proveedor" column="Prov_Nombre" nameColumn="Prov_Nombre" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Nombre" />
        </Hierarchy>
      </Dimension>
    - <Dimension type="StandardDimension" foreignKey="Prod_IdODS" name="producto" caption="Producto">
      - <Hierarchy name="jerarquías_de_producto" hasAll="true" allMemberCaption="jerarquías de producto"
          primaryKey="Prod_IdODS" primaryKeyTable="Ods_Producto" caption="Jerarquías de Producto">
          <Table name="Ods_Producto" schema="public" />
          <Level name="linea" table="Ods_Producto" column="Prod_Linea" nameColumn="Prod_Linea" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Linea" />
          <Level name="seccion" table="Ods_Producto" column="Prod_Seccion" nameColumn="Prod_Seccion" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Seccion" />
          <Level name="departamento" table="Ods_Producto" column="Prod_Departamento" nameColumn="Prod_Departamento"
            type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Departamento" />
          <Level name="grupo" table="Ods_Producto" column="Prod_Grupo" nameColumn="Prod_Grupo" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Grupo" />
          <Level name="division" table="Ods_Producto" column="Prod_Division" nameColumn="Prod_Division" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="División" />
          <Level name="origen" table="Ods_Producto" column="Prod_Origen" nameColumn="Prod_Origen" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Origen" />
          <Level name="familia" table="Ods_Producto" column="Prod_Familia" nameColumn="Prod_Familia" type="String"
            uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Familia" />
          <Level name="codigo_barra" table="Ods_Producto" column="Prod_CodigoBarra" nameColumn="Prod_CodigoBarra"
            type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never" caption="Código de Barra" />
        </Hierarchy>
      </Dimension>
    - <Dimension type="StandardDimension" foreignKey="TipoAj_IdODS" name="tipo_ajuste" caption="Tipo de Ajuste">
      - <Hierarchy name="jerarquias_de_tipo_ajuste" hasAll="true" allMemberCaption="jerarquias de tipo ajuste"
          primaryKey="TipoAj_IdODS" primaryKeyTable="Ods_TipoAjuste" caption="Jerarquías de Tipo de Ajuste">
          <Table name="Ods_TipoAjuste" schema="public" />
```

GLOSARIO DE TÉRMINOS

CIMEX: Corporación Importadora y Exportadora de Divisas.

DW: Almacén de Datos; Data Warehouse, del inglés.

ODS: Almacén de Datos Operacionales; Operational Data Store, del inglés.

OLAP: Procesamiento Analítico en Línea; On Line Analytical Processing, del inglés.

ROLAP: Procesamiento Analítico Relacional en Línea; Relational On Line Analytical Processing, del inglés.

MOLAP: Procesamiento Analítico Multidimensional en Línea; Multidimentional On Line Analytical Processing, del inglés.

OLTP: Procesamiento Transaccional en Línea; On Line Transaction Processing, del inglés.

ETL: Extracción Transformación y Carga; Extraction Transformation and Load, del inglés.

BI: Inteligencia del Negocio; Business Intelligence, del inglés.