



Universidad de las Ciencias Informáticas
Facultad 3

Título:

Desarrollo de un DataMart para la obtención de las razones financieras de los subsistemas de Cobros y Pagos, Caja y Banco del proyecto ERP-Cuba.

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

Autor: Mauricio Guanche Cañizares.

Tutor (es): Ing. Ariel Torres Gálvez.
Ing. Joisel Pérez Pérez.

Ciudad de la Habana
Junio de 2011

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Mauricio Guanche Cañizares

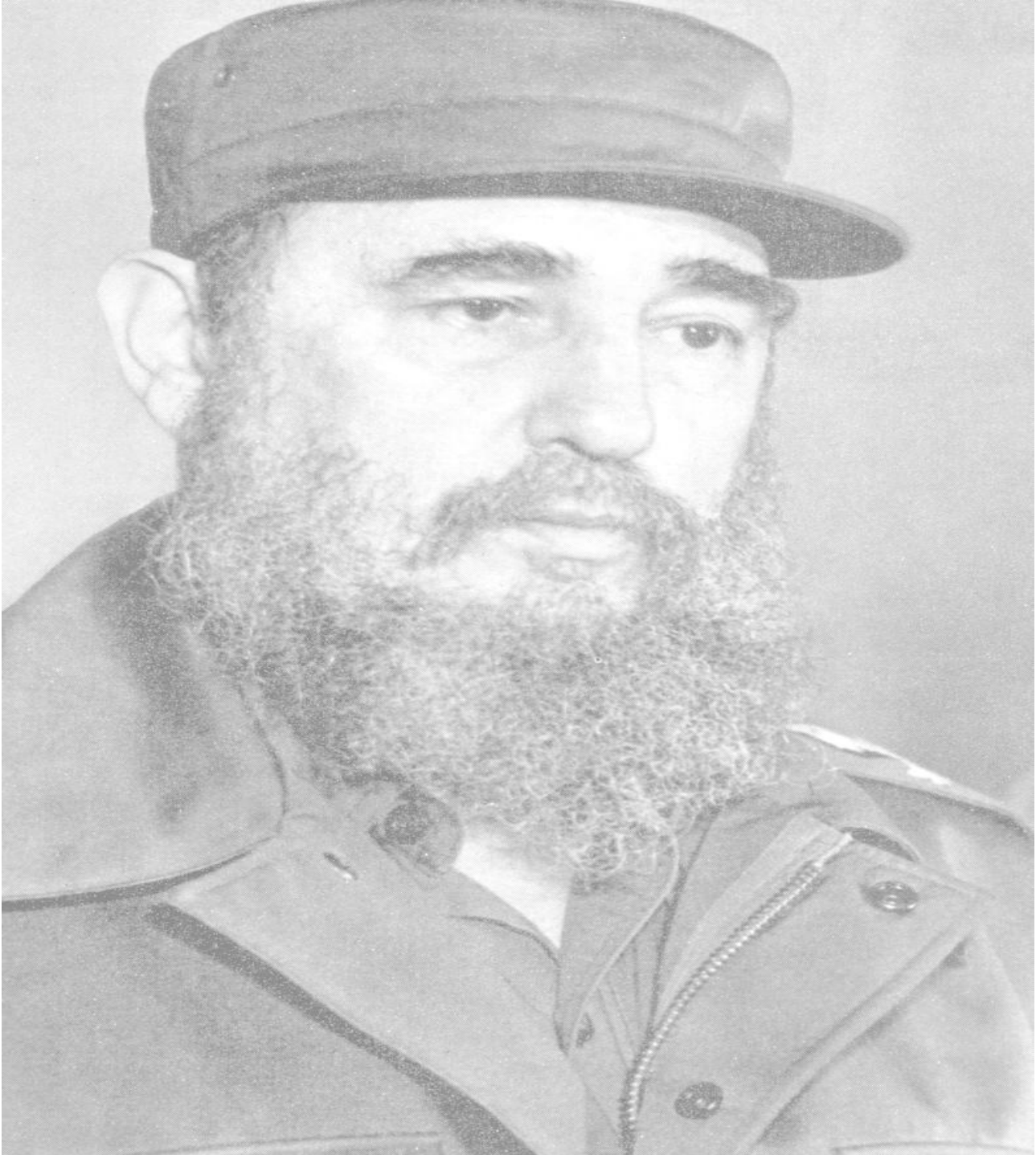
Firma del Autor

Ing. Joisel Pérez Pérez

Firma del Tutor

Ing. Ariel Torres Gálvez

Firma del Co-tutor



“El futuro de Cuba tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento” Fidel Castro

DATOS DE CONTACTO

Síntesis del Tutor: Ing. Joisel Pérez Pérez

Graduado de Ingeniero en Ciencias Informáticas en la UCI en el 2008. A partir de ese momento se incorpora como Arquitecto de Datos del proyecto ERP Cuba. Actualmente se desempeña como arquitecto de datos del módulo Costos y Procesos del proyecto ERP Cuba. En su trabajo de diploma investigó sobre las arquitecturas de software y su relación con los atributos de calidad haciendo una propuesta de un modelo para seleccionar los estilos y patrones más adecuados a la hora de potenciar o inhibir dichos atributos. Ha tutorado 3 tesis y ha sido oponente de 4, ha impartido curso de pregrado de Data Warehouse y en su haber se encuentra la publicación de varios artículos en eventos de contabilidad financiera.

A mi mamá y mi papá por haberme hecho el hombre que soy hoy, a mi hermanita por el cariño y el apoyo que siempre me ha dado. Nadie mejor que yo sabe el sacrificio que han hecho para que yo lograra mis metas.

A toda mi familia por su constante preocupación por mi desarrollo profesional y en la vida y sobre todo por tener plena confianza en mí.

A mis amigos de la infancia, A mis primos por hacerme pasar momentos felices.

A todos los profesores que me han formado desde la primaria hasta la universidad.

A mis profesores de los Camilitos de Arroyo Arenas por haberme dado una excelente preparación.

A mis amigos de la universidad Randy, Jorge, Yendri, Alexander, Ruby, Yandy, Yero, el Lacho y el resto del piquete por ser los mejores del mundo.

A mi familia de las FAR tanto oficiales como profesionales y cadetes que tanto apoyo me han dado durante estos años de universidad.

Al mi gente del ERP por permitirme ser parte de ustedes, no los nombro uno a uno porque son muchos.

Al equipo de de Costos y Contabilidad por ser mi otra familia en especial Carlos, Joisel, Yanay, Roberto y Juan.

A mi Livan por ser como un hermano para mí y apoyarme cada vez que lo necesito.

A mi tutor, por haberme enseñado que para lograr las metas hay que sacrificarse y por haber sido más que un tutor un amigo. Gracias Joisel.

A Marielena por haberme ayudado en todo para que este trabajo saliera a flote.

A todos muchas gracias.

Mauricio

Este trabajo se lo dedico a mis padres y a mi hermana que son lo más grande que tengo un la vida. A ustedes mi familia por el amor y el apoyo que siempre me han dado.

RESUMEN

Hoy en día las compañías quieren explotar y maximizar el valor de su información con la idea principal de tener mayor ventaja sobre sus competidores. Los negocios se están moviendo con una mayor rapidez, por lo que tener la información correcta en las manos adecuadas y en el momento preciso es esencial. De manera que el control de la gestión financiera y la aplicación de la inteligencia de negocios han ocupado un papel fundamental dentro de la gestión de las empresas, estas necesitan utilizar la inteligencia de negocios a través de los análisis financieros con el objetivo de tomar decisiones que las hagan más competentes y les permitan subsistir en el mercado. Dentro de este proceso las razones financieras juegan un papel fundamental. Actualmente en los subsistemas de Cobros y Pagos, Caja y Banco del producto Cedrux, la realización de este análisis no es posible ya que se registran solamente los datos provenientes de las transacciones que se realizan a diario en las entidades dificultando así a los ejecutivos de las empresas el proceso de toma de decisiones. En este trabajo se presenta un DataMart para realizar el análisis de estos datos con el objetivo de poder predecir futuros eventos contables. Para el desarrollo de la investigación se emplearon herramientas libres siguiendo las políticas de independencia tecnológica aplicadas en el país, y se tomó como referencia la metodología que se utiliza en la Unidad de Compatibilización, Integración y Desarrollo de Software (UCID) para este tipo de soluciones.

PALABRAS CLAVES: subsistemas de Cobros y Pagos, Caja y Banco, razones financieras, DataMart

TABLA DECONTENIDO

INTRODUCCIÓN.....1

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS SOBRE EL DESARROLLO DE UN DATA WAREHOUSE4

1.1 Introducción.....4

1.2 Introducción a los Data Warehouse.....4

1.3 DataMart4

1.3.2 Características principales6

1.3.3 DataMart VS Data Warehouse7

1.3.4 Ventajas de los DataMart7

1.3.5 Desventajas de los DataMart.....8

1.4 Componentes de un DataMart.....8

1.4.1 Fuentes de Origen9

1.4.2 Área de ETL.....9

1.4.3 Área de Consulta10

1.5 Modelo Relacional.....10

1.6 Sistemas OLTP y OLAP12

1.7 Modelo Multidimensional13

1.8 Modelos y Arquitecturas habitualmente utilizados en el diseño de un DataMart16

1.9 Herramientas para la gestión financiera19

1.10 Metodologías para el diseño de un DW25

1.11 Metodología Hefestos.....26

1.12 Ciclo de vida Kimball27

1.13 Metodología de Inmon28

1.14 Metodología de DATEC29

1.15 Metodología UCID (16).....29

1.16 Justificación de la metodología a utilizar31

1.17 Herramientas para la construcción del DataMart31

1.17.1 Sistemas Gestores de Bases de Datos31

1.17.2 Herramientas de integración de datos33

1.17.3 Herramientas para OLAP36

1.17.4 Herramientas CASE38

1.17.5 SERVIDOR DE APLICACIONES.....39

Conclusiones39

CAPÍTULO 2: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL DATAMART	41
Introducción	41
2.1 Procedimiento para desarrollo de la solución utilizando la metodología escogida.	41
2.1.1 Fase 1: Análisis de los requerimientos.....	41
2.1.2 Fase 2: Análisis de los OLTP.....	43
2.1.3 Fase3: Modelo lógico del DW	48
Definir Tipo de Modelo Lógico del DW	48
Modelo Físico del DataMart.....	49
2.1.4 Fase 4: Proceso ETL	50
Conclusiones del capítulo	55
CAPÍTULO 3: VALIDACIÓN Y DESPLIEGUE.....	56
3.1 Introducción.....	56
3.2 Validación	56
3.2.1 Validación del diseño mediante pruebas de rendimiento a la aplicación.....	62
3.3 Mecanismo de consolidación de la información.....	63
3.4 Despliegue del DataMart	64
CONCLUSIONES GENERALES.....	66
RECOMENDACIONES	67
BIBLIOGRAFÍA REFERENCIADA	68
BIBLIOGRAFÍA CONSULTADA.....	70
GLOSARIO DE TÉRMINOS.....	72

ÍNDICE DE FIGURAS

Figura. 1. Cubo OLAP..... 13

Figura. 2 Modelo en Estrella 17

Figura. 3 Modelo en Copo de Nieve..... 17

Figura. 4 Modelo Constelación..... 19

Figura. 5 Fases de la metodología HEFESTOS..... 27

Figura. 6 Ciclo de vida de Kimball..... 27

Figura. 7 Modelo Conceptual 42

Figura. 8 Modelo Conceptual del cubo finanzas detalle 43

Figura. 9 Modelo Conceptual Ampliado del cubo Finanzas..... 47

Figura. 10 Modelo Conceptual Ampliado del cubo Finanzas Detalle 48

Figura. 11 Modelo Lógico..... 49

Figura. 12 Modelo físico..... 50

Figura. 13 Selección de la base datos de origen y consulta para seleccionar los campos. 53

Figura. 14 Llenado de la dimensión estructura..... 53

Figura. 15 Llenado de la tabla de hechos cub_hecho_finanzas. 54

Figura. 16 Trabajo para llenar las dimensiones y la tabla de hecho cub_hecho_finanzasdetalle. 55

Figura. 17 Cubos multidimensionales con sus medidas y dimensiones asociadas..... 57

Figura. 18 Reporte Inversiones Financieras..... 58

Figura. 19 Reporte Inversiones Financieras..... 58

Figura. 20 Ciclo de Caja Global 59

Figura. 21 Reporte Inversiones Financieras..... 60

Figura. 22 Ciclo de Caja Global 61

Figura. 23 Diagrama de despliegue del DataMart 65

INTRODUCCIÓN

Hoy en día la informática se ha introducido de forma permanente en la sociedad y la economía. El mundo de hoy ha aumentado considerablemente la competencia entre las empresas motivado por los cambios constantes y acelerados adelantos en la Tecnológica y la alta complejidad, pasando a un primer nivel el manejo eficaz de las empresas haciendo imprescindible un correcto control sobre la gestión financiera en las mismas, de manera que estas sean rentables; dentro de este proceso, las finanzas juegan un papel fundamental, ellas registran la información real de todas las operaciones sobre los activos y el efectivo de la empresa, por lo que son utilizadas como instrumento de planificación, ejecución y control que repercute decididamente en la economía.

En los últimos 20 años las empresas han introducido los análisis financieros como parte de sus estrategias en la inteligencia de negocios, los cuales se utilizan para diagnosticar la situación financiera, precisar el grado de liquidez, de rentabilidad, y endeudamiento de la empresa, entre otros. Lo que hace posible a la dirección de la empresa hacer predicciones o ajustes para corregir los aspectos negativos que pudieran afectar su supervivencia.

Hace algunos años la mayoría de los procesos que intervenían en la actividad financiera se realizaban de manera manual, lo que provocaba que estos fueran lentos y engorrosos, conllevando en muchos casos a errores por el manejo de gran cantidad de información, influyendo directamente en la competitividad de la entidad.

Por la importancia de estos procesos para los negocios y aprovechando las facilidades que brindan las Tecnologías de la Información y las Comunicaciones (TIC). Muchas empresas han invertido en la automatización de los mismos, a través de los Sistemas de Planificación de Recursos Empresariales (ERPs por sus siglas en inglés). Estos facilitan la integración de la información de las diferentes áreas o departamentos de la entidad separándola en módulos, tienen la capacidad de adaptarse a las necesidades particulares de cada empresa optimizando los procesos empresariales y mejorando el aprovechamiento del tiempo y los costos, lo que puede transformarse en eficiencia y aumento de la productividad.

En Cuba, como parte de las transformaciones económicas que se vienen llevando a cabo, se han introducido algunos de estos sistemas en sus entidades sin que los mismos satisfagan con éxito sus necesidades, ni se adecúen a las características del sistema económico del país. Es por eso que en la Universidad de Ciencias Informáticas (UCI) actualmente se desarrolla un ERP cubano que se adapte a las

necesidades de las entidades del país, facilite la comunicación entre las distintas empresas, y cumpla con los principios de la independencia tecnológica.

Las empresas que usarán el Sistema de Gestión de Entidades (Cedrux), principal producto del ERP- Cuba necesitan utilizar la inteligencia de negocios a través de los análisis financieros con el objetivo de tomar decisiones que las hagan más competentes y subsistir en el mercado. Por lo que este software debe permitir realizar este tipo de análisis.

Actualmente en los subsistemas de Cobros y Pagos, Caja y Banco donde se controlan operaciones financieras, estos análisis no se pueden realizar ya que la gran cantidad de datos obtenidos como resultado de las mismas, reciben un procesamiento operacional de manera que muchos se vuelven irrelevantes y pasan inadvertidos para los directivos impidiendo ver el estado real de la situación financiera de la empresa, realizar comparaciones entre varios periodos e identificar con exactitud dónde está el problema, afectando todo esto la toma de decisiones oportunas y la capacidad de la empresa para enfrentarse de manera favorable a sucesos económicos futuros.

De la situación problemática anteriormente expuesta se deriva el siguiente **problema a resolver**: ¿Cómo mejorar la obtención de razones financieras en los subsistemas de Cobros y Pagos, Caja y Banco del proyecto ERP-Cuba para las entidades cubanas?

Se plantea como **objeto de estudio**: los procesos contables y como **campo de acción**: el análisis de los datos de los procesos financieros en las entidades cubanas.

Para dar solución al problema se plantea como **objetivo general**: Desarrollar un DataMart para realizar el análisis de los datos generados en los subsistemas de Cobros y Pagos, Caja y Banco del proyecto ERP-Cuba que permita obtener razones financieras.

Objetivos específicos:

1. Evaluar la obtención de las razones financieras en los subsistemas de Cobros y Pagos, Caja y Banco del proyecto ERP-Cuba.
2. Fundamentar la investigación mediante la elaboración del Marco Teórico.
3. Diseñar un DataMart para los subsistemas de Cobros y Pagos, Caja y Banco del proyecto ERP-Cuba.

4. Implementar un DataMart para los subsistemas de Cobros y Pagos, Caja y Banco del proyecto ERP-Cuba.
5. Validar los artefactos de salida mediante pruebas en cada una de las fases del proceso de desarrollo de software.

Las **tareas de la investigación** definidas son:

1. Realización del estudio del estado del arte para el análisis y definición de razones financieras.
2. Realización de un estudio sobre el estado del arte de las principales metodologías y herramientas utilizadas para la construcción de DataMarts.
3. Definición de dimensiones, medidas naturales y calculadas del DataMart.
4. Elaboración de estrategias para la estandarización de los datos de las fuentes de origen.
5. Realización del diseño del modelo de datos del DataMart.
6. Implementación del modelo de datos del DataMart.
7. Realización de las vistas de las razones financieras.
8. Realización de pruebas de concepto al modelo de datos.

El contenido de este trabajo se divide en 3 capítulos:

- **CAPÍTULO 1.** Se plantean los principales aspectos relacionados con los sistemas informatizados vinculados al campo de acción, se introducen conceptos claves referentes al problema, indispensables para comprender el negocio y la propuesta de solución. Se realiza un análisis de las razones financieras. Se fundamenta la utilización de un conjunto de técnicas, y herramientas usadas para el diseño e implementación de un DataMart, así como la relación de los artefactos.
- **CAPÍTULO 2.** Se realiza el diseño e implementación de la solución aplicando las metodologías, técnicas y herramientas escogidas en el capítulo anterior. Se describe paso a paso cada una de las fases de la metodología escogida desde el análisis de los requerimientos, pasando por la identificación de indicadores y perspectivas, dimensiones, medidas, hasta los procesos de ETL
- **CAPÍTULO 3.** Se plantean diferentes tipos de pruebas realizadas al DataMart para realizar la validación de la solución, así como su evaluación. Estas pruebas se basaron en dos aspectos fundamentales, las relacionadas con el desempeño el sistema a partir de su diseño y las correspondientes con el cumplimiento de las funcionalidades solicitadas por los clientes. Se explica además el funcionamiento de la consolidación de la información vinculada al desempeño del almacén de datos, así como su despliegue.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS SOBRE EL DESARROLLO DE UN DATA WAREHOUSE

1.1 Introducción

En este capítulo se abordan definiciones y características fundamentales de los Data Warehouse (DW por sus siglas en inglés) y DataMart (DM por sus siglas en inglés). Se realiza un estudio sobre la situación actual del uso de las razones financieras nacional e internacionalmente. Se hace un estudio bibliográfico de las metodologías existentes en el mundo y en nuestro país para el desarrollo de un DM. Se fundamenta el uso de las distintas herramientas y tecnologías utilizadas para la construcción de los mismos.

1.2 Introducción a los Data Warehouse

Desde su surgimiento, hace ya algunos años, muchos autores han estudiado a los Almacenes de Datos o Data Warehouse como también se les conoce, y han planteado sus consideraciones acerca de los mismos, a continuación se referencian algunas de las definiciones más conocidas.

En 1993, Susan Osterfeldt publicó una definición muy acertada de Data Warehouse: "Yo considero al Data Warehouse como algo que provee dos beneficios empresariales reales: Integración y Acceso de datos. Data Warehouse elimina una gran cantidad de datos inútiles y no deseados, como también el procesamiento desde el ambiente operacional clásico". (25)

Refiriéndose a los Data Warehouse Rall Kimball, reconocido mundialmente por sus muchos libros sobre el tema, planteó:

"Es una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis". Kimball también expresó que un Data Warehouse no era más que: "la unión de todos los DataMart de una entidad". (1)

Pero la definición más conocida y con la cual concuerda el autor es la que dio William Immon, conocido como el padre del Data Warehouse, en su libro "Using de Data Warehouse", en 1992 cuando planteó:

"El Data Warehouse es una colección de datos orientados al tema, integrados, no volátiles e historizados, organizados para el apoyo de un proceso de ayuda a la toma de decisiones. (1)

1.3 DataMart

Un DataMart no es más que un Almacén de Datos con función departamental o regional. Consta de las mismas características que un DW y brinda las mismas facilidades, pero está orientado a una sola actividad y no a satisfacer las necesidades de toda la empresa. Por tanto, no se puede pensar en

DataMart en los términos de un DW más pequeño, porque no es su tamaño lo que lo define sino su objetivo en la organización.

La creación de DataMart es particularmente apropiada cuando el DW central crece muy rápidamente y los distintos departamentos requieren solo una pequeña porción de los datos contenidos en él.

1.3.1 Objetivos de un DataMart:

El objetivo principal de un DM es satisfacer los requerimientos de la información interna de la empresa para una mejor gestión, con eficiencia y facilidad de accesos.

Para complementar este objetivo los DM deben construirse de manera que:

- Deben hacer fácilmente accesible la información.

La información que se almacena debe ser accesible en todo momento, diseñándose mediante datos que sean intuitivos y obvios para los usuarios del negocio y no para los desarrolladores. Se debe etiquetar significativamente el contenido que almacena. El diseño de las estructuras debe soportar que los usuarios combinen y separen los datos en un sinnúmero de combinaciones. La información que está almacenada debe ser recuperada con un tiempo mínimo de espera.

- La información de la organización debe ser presentada de forma consistente.

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

- Deben ser adaptables y resistentes al cambio.

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

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

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

- Debe servir como base para mejorar la toma de decisiones.

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

1.3.2 Características principales

Un DataMart es una colección de datos:

Orientado a Temas

El DataMart se construye orientado a la información más relevante de la organización. Los datos se ordenan por temas de manera que todos los elementos relativos al mismo objeto del mundo real quedan unidos entre sí. Esto permite facilitar su acceso y su entendimiento por parte de los usuarios finales. Los temas a analizar por cada organización son particulares de cada una de ellas.

Integrado

Los datos que se almacenan en el DataMart provienen de varios y diferentes sistemas operacionales de la organización y/o fuentes externas, en los cuales los datos son almacenados con diferentes formatos, lo que provoca problemas a la hora de referenciarlos para hacer análisis con ellos. Es necesario eliminar las inconsistencias entre estos sistemas e integrar los datos en una estructura consistente. La información puede estructurarse también en distintos niveles de detalle, para adecuarse a las necesidades de los usuarios.

No volátil

Para realizar análisis de la información y la toma de decisiones se necesita una base de datos estable, por eso en un DataMart la información es permanente e invariable, de manera que una vez introducidos los datos en el mismo, pasan a ser de solo lectura, es decir, los datos no pueden ser modificados ni eliminados. Las únicas dos acciones que se permiten realizar son la carga inicial de los datos y el acceso a la información.

De tiempo variante

Dentro de la información que contienen los DataMart el tiempo es una de las más relevantes. A diferencia de los sistemas operacionales donde los datos solo reflejan el estado actual del negocio, la información almacenada en el DataMart sirve, entre otras cosas, para realizar la identificación y análisis de tendencias. Es por eso que, el DataMart no actualiza los datos, sino que almacena el historial de ellos, es decir, el conjunto de valores que el dato ha tenido a lo largo de su historia. Los datos son almacenados como fotos

(en inglés conocidos como snapshots) correspondientes a periodos de tiempo, lo que permite hacer comparaciones.

Generalmente el horizonte de tiempo que se establece en este tipo de estructura es de años, de 5 a 10 años mínimo.

1.3.3 DATAMART VS DATA WAREHOUSE

La principal característica que diferencia a un DM de un DW es el objetivo de la organización al cual están dirigidos, mientras el primero está diseñado para suplir las necesidades a nivel de departamento o regional el otro lo hace a nivel global o general de la empresa; pero esta no es su única diferencia existen otras que se deben valorar a la hora de elegir cuál de ellos se ajusta a las necesidades de la organización que lo va a emplear. (Ver Tabla 1)

Aspectos	DM	DW
Costo	Es menos costoso de diseñar e implementar que un DW.	Más costoso que un DM
Objetivo	Optimiza la distribución de información útil para la toma de decisiones.	Permite la mejora en la toma de decisiones.
Integración	Puede integrarse eventualmente a un DW, pero no se integra con otros DM fácilmente	Se puede integrar con varios DM
Alcance	Se ajusta mucho a las necesidades que tienen un área específica.	Tiene un enfoque global, se ajusta a las necesidades de la organización en su conjunto.
Cantidad de usuarios	Soporta menos usuarios que un DW.	Soporta más usuarios que un DM.

Tabla 1. Comparación de DataMart vs Data Warehouse

1.3.4 Ventajas de los DataMart

- Transforma datos orientados a las aplicaciones en información orientada a la toma de decisiones.
- Integra y consolida diferentes fuentes de datos en una única plataforma sólida y centralizada.
- Provee la capacidad de analizar y explotar toda la información que posee.

- Permite reaccionar rápidamente a los cambios del mercado.
- Aumenta la competitividad en el mercado.
- Mejora la entrega de información, es decir, información completa, correcta, consistente, oportuna y accesible.
- Aprovecha el enorme valor potencial de los recursos de información y los transforma en valor verdadero.
- Permite al usuario adquirir mayor confianza acerca de sus propias decisiones y de las del resto, y lograr así, un mayor entendimiento de los impactos ocasionados.
- Los usuarios pueden acceder directamente a la información en línea, lo que contribuye a su capacidad para operar con mayor efectividad en las tareas rutinarias.
- Los usuarios pueden tener a su disposición una gran cantidad de información multidimensional, presentada coherentemente como fuente única, confiable y disponible en sus estaciones de trabajo (1).

1.3.5 Desventajas de los DataMart

- Una herramienta tan imprescindible para la empresa de hoy, necesita de un constante y costoso soporte técnico. Debido a su complejidad, el DM es muy “frágil” en cuanto funcionamiento se refiere, por lo que se hace necesaria su continua revisión.
- Los almacenes de datos se pueden quedar obsoletos relativamente pronto.
- Una vez implementado puede ser complicado añadir nuevas fuentes de datos.
- En un proceso de implantación pueden encontrarse dificultades ante los diferentes objetivos que pretende una organización.
- El diseño e implementación del DM resulta caro (aunque puede considerarse como una inversión) y toma mucho tiempo puesto que se necesitan obtener muchos datos, y estos se tienen que organizar de la mejor manera para que el DM pueda realmente cumplir con su tarea.(2)

1.4 Componentes de un DataMart

En el desarrollo del Almacén de datos intervienen varios procesos los cuales varían en dependencia de las preferencias del que implementa el DataMart, aunque todos coinciden en que los datos para construir un DM provienen de fuentes de datos operacionales para luego ser transformados y cargados en el DM en forma de cubos multidimensionales de forma que se facilite el acceso y análisis de los datos por parte de los usuarios finales a través de herramientas de consulta.

A continuación se muestran los procesos que según la bibliografía consultada son los más utilizados en el desarrollo de un DM. (2)

1.4.1 Fuentes de Origen

Las fuentes de origen para los datos que se cargan en el DataMart la integran los sistemas de Procesamiento de Transacciones en Línea (OLTP, por sus siglas en inglés)¹, y presentan toda la información transaccional que genera la empresa en su accionar diario, y otras fuentes externas. Debido a la diversidad de procedencia, los datos contenidos en estas fuentes presentan en la mayoría de los casos distintos formatos y estructuras, entre las fuentes de origen más comunes se encuentran: los archivos de textos, hipertextos, hojas de cálculos, informes, bases de datos transaccionales, entre otros.

1.4.2 Área de ETL

En esta área se realiza el proceso más importante dentro del ciclo de vida del DataMart, los procesos de Extracción, Transformación y Carga de datos, comúnmente conocidos por sus siglas en inglés como ETL. En estos procesos se extraen los datos provenientes de las fuentes de origen, se transforman e integran para después ser cargados en el DM.

Para una mejor explicación del funcionamiento de este proceso, se dividió en 3 subprocesos:

Extracción y limpieza

Este proceso consiste en la selección de los datos de los sistemas de origen para su posterior procesamiento y carga. Los datos deben ser procesados e integrados antes de poblar el DM. Para ello se deben eliminar los problemas ocasionados por la heterogeneidad de las fuentes de origen (datos con diferente nombre, tipo, asignación de valores y estructura interna).

En esta etapa se realiza también la limpieza de datos, es decir, se eliminan los datos inconsistentes, que no existen, no pueden leerse o son erróneos; informando los errores cometidos en la extracción o en los sistemas fuentes. (3)

Al terminar esta fase los datos tienen un formato preparado para iniciar el proceso de transformación.

Transformación

¹ El procesamiento transaccional en línea (OLTP – On Line Transaction Processing), conocido también como procesamiento operacional, sustenta las operaciones diarias de la empresa y describe los requerimientos operacionales del sistema. Este procesamiento se refiere a un tipo de cómputo en el cual el énfasis está en el procesamiento de las transacciones tal y como son recibidas por las aplicaciones. Los sistemas operacionales constituyen el pilar fundamental para el análisis de la información.

En esta etapa se preparan los datos extraídos y filtrados, para que se puedan realizar consultas sobre ellos. Para ello se realizan un grupo de funciones entre las que se encuentran: la estandarización de códigos, agregar nuevos atributos calculados, etc.

Carga

En esta etapa se cargan en el DataMart los datos que fueron extraídos, limpiados y transformados en los procesos anteriores, para luego ser analizados por los usuarios.

1.4.3 Área de Consulta

En esta área están localizados los componentes con los que interactuarán los usuarios, los cuales podrán visualizar la información contenida en los DataMart, de manera que los datos contenidos en los mismos podrán ser agrupados y explorados.

1.5 Modelo Relacional

El modelo relacional es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos, y tras su definición en 1970 por Edgar Frank Codd, ha constituido el modelo de bases de datos más utilizado por excelencia.

La idea fundamental de este modelo es el concepto de relaciones, donde una relación se define como un conjunto de N-tuplas, y una tupla como un conjunto no ordenado de valores de atributos, aunque estrictamente en matemáticas, una tupla tiene un orden y permite duplicados. Un atributo sería entonces una pareja ordenada, formada por un nombre de atributo y un tipo o dominio de valores, refiriéndose a todos los valores únicos que un elemento puede contener. El valor de un atributo es un valor específico y válido según el tipo del atributo que lo define. Señalar que una relación va a estar compuesta por una cabecera y un cuerpo (también llamado extensión). La cabecera se compone de un conjunto de atributos no ordenados, y el cuerpo está formado por un conjunto no ordenado de n-tuplas.

En este modelo, todos los datos se almacenan en relaciones, y como cada relación es un conjunto de datos, el lugar y el orden en el que estos se almacenen, no tiene mayor relevancia. Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. Para manipular y consultar los datos que se almacenan en diferentes relaciones, se utiliza lo que se denomina un lenguaje relacional, que ofrece una amplia flexibilidad y poder, para administrar la información. (4)

Dentro de este modelo existe una técnica de diseño lógico usada principalmente para diseñar sistemas de Procesamiento Transaccional en Línea (OLTP, por sus siglas en inglés), denominada modelo entidad relación E/R. Estos modelos expresan las entidades más relevantes para el sistema, sus interrelaciones y

propiedades. Trabajan dividiendo los datos en muchas entidades discretas donde cada una se convierte en una tabla física en la base de datos operacional. Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización, el cual es entendido como el proceso necesario para que una base de datos se utilice de manera óptima y no exista redundancia de datos.

Estas bases de datos se caracterizan por:

- Garantizar que no existe la duplicidad de registros.
- Garantizar la integridad referencial, de forma que si se elimina una fila, se eliminan todas las filas dependientes.(5)
- Favorecer la normalización, por ser más comprensible y aplicable.
- Crecen fácilmente, haciéndose más y más complejos.
- Se puede apreciar la existencia de muchos caminos para ir de una tabla a otra. Sería natural pensar que al tener diversos caminos para llegar desde una tabla a otra, cualquiera de ellos entregaría el mismo resultado, pero lamentablemente esto no siempre sucede así.
- El diagrama se visualiza simétrico, donde todas las tablas se parecen, sin distinguir a priori la importancia de unas respecto a otras. No es fácil de entender tanto para usuarios como para los diseñadores.

El modelo relacional tiene entre sus objetivos guardar la integridad de los datos obtenidos en los procesos transaccionales automatizados. Sin embargo, este modelo no se corresponde con la forma en la que el usuario percibe la gestión del conocimiento de un negocio en general, y la gestión del conocimiento comercial en particular. De hecho, Codd afirmó que aunque los sistemas de gestión de bases de datos relacionales, han sido muy beneficiosos para los usuarios, nunca han sido diseñados para proporcionar funciones potentes de síntesis, análisis y consolidación de los datos (6). (7)

¿Por qué no son útiles los modelos relacionales para la construcción de los almacenes de datos?

- Divide los datos en muchas entidades, cada una de las cuales se convierte en una tabla en la BD.
- Se torna en un diagrama asombrosamente complejo generalmente consistente de cientos de tablas.
- Es muy simétrico, todas las tablas parecen iguales. No existe manera para decir que tabla es más importante o la mayor.
- No existe forma de decir que tablas contienen medidas numéricas y que tablas incluyen descriptores estáticos de los objetos

- Son muy difíciles para las personas (usuarios finales o diseñadores) visualizar y conservar en sus cabezas.
- Si dos tablas en el diagrama son necesarias en una consulta dada, existen un número inmenso de posibles rutas de conexiones entre ambas.

1.6 Sistemas OLTP y OLAP

Sistemas OLTP

La finalidad de los sistemas de bases de datos operacionales consiste en la realización de transacciones y la resolución de consultas, por esa razón se denominan sistemas OLTP. Una transacción no es más que un proceso atómico que debe ser validado o abortado, y que puede involucrar operaciones de consulta, inserción, modificación y borrado de datos. Estos sistemas cubren la mayor parte de la operativa del día a día de una empresa.

Sistemas OLAP

Son sistemas de procesamiento analítico en línea (On-Line Analytical Processing). Es una solución cuyo objetivo es agilizar la consulta de grandes cantidades de datos. Para ello utiliza estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes bases de datos o sistemas transaccionales.

Dentro de las principales características que se pueden encontrar en los sistemas OLAP están:

- Rapidez: proporciona la información al usuario a una velocidad constante, permitiendo la interactividad con él y proporcionándole la información deseada.
- Análisis: posibilita realizar análisis estadísticos y numéricos a los datos, que pueden ser predefinidos por el desarrollador de la aplicación o definido ad-hoc² por el usuario. Con un alto grado de personalización de las herramientas de análisis para ajustarlas a las necesidades del usuario final.
- Compartido: implementa los requerimientos de seguridad necesarios para compartir datos potencialmente confidenciales a través de una gran población de usuarios, dependiendo del nivel de seguridad asignado al usuario que realiza la consulta.

²ad-hoc: El término se utiliza en informática para referirse a consultas en bases de datos *ad hoc querían* o *ad hoc reporting*; esto implica que el sistema permite al usuario personalizar una consulta en tiempo real, en vez de estar atado a las consultas prediseñadas para informes.

- Multidimensional: considerada la característica esencial de estos sistemas, que consiste en ver la información en determinadas vistas o dimensiones, permitiendo la visualización gráfica de la información. (4)
- Información: se accede a todos los datos y a la información necesaria y relevante para la aplicación, donde sea que ésta resida y sin estar limitada por el volumen.

1.7 Modelo Multidimensional

Los DataMart a diferencia de las base de datos tradicionales que utilizan modelos relacionales para su representación y tecnología OLTP utilizan el modelo multidimensional para representar sus datos, el cual tiene los mismos datos que el modelo operacional pero con otra estructura. El modelo multidimensional es una técnica de diseño lógico de bases de datos que permite tener acceso flexible a los datos, permitiendo explorar y analizar sus relaciones, y consiguientes resultados, garantizando la velocidad y eficiencia en la recuperación de la información.

En otras palabras el modelo multidimensional no es más que el modelo que utilizan los Sistemas OLAP, incluidos los DataMart, para representar la información y poder realizar las complejas consultas que se necesitan para el análisis de los datos de los mismos. Comúnmente se representa en forma de Cubo, de forma que los datos planos que se encuentran en las filas y columnas de las tablas de la base datos se convierten en una matriz de N dimensiones, dando la sensación de una figura geométrica en tres dimensiones. (Ver Figura 1)

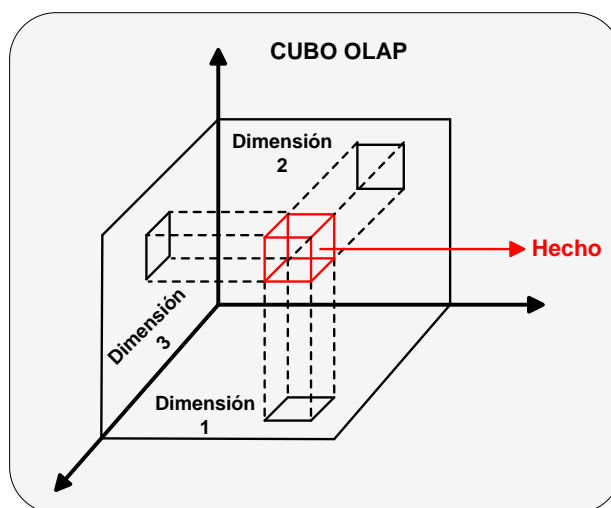


Figura. 1. Cubo OLAP

Para entender mejor como funciona este modelo es necesario conocer en qué consisten los hechos, y medidas y dimensiones, componentes fundamentales del mismo, aclarar que los cubos OLAP al ser la base del modelo multidimensional, también se encuentran constituidos por dichos componentes.

Hechos

Los hechos representan la actividad que se quiere analizar, por ejemplo, compras, ventas, préstamos, etc.
(8)

Un hecho es un concepto de interés primario para el proceso de toma de decisiones, corresponde a eventos o actividades que ocurren dinámicamente en el negocio de la empresa. Los hechos son almacenados en una tabla denominada tabla de hechos.

Medidas

Las medidas son los valores de los datos que se analizan en el almacén de datos. Una medida es una columna cuantitativa, en la tabla de hechos:

Las siguientes son algunas de las características de las medidas:

- Constituyen valores que permiten analizar los hechos
- Estos valores son las bases a partir de las cuales el usuario puede realizar cálculos.
- Deben ser numéricas.
- Cruzan todas las dimensiones en todos los niveles.
- Las medidas pueden clasificarse en:

Naturales

Son las columnas numéricas que provienen directamente de los sistemas OLTP, están estrechamente vinculadas con las agregaciones, resúmenes de datos precalculados (o medidas) que mejoran el tiempo de respuesta por el simple hecho de tener preparadas las respuestas antes de que se planteen las preguntas.

Calculadas

Son las medidas que se calculan manipulando las medidas naturales presentes en el cubo para facilitar el análisis de los hechos. Estas medidas no se almacenan en la base de datos.

Con las medidas calculadas se puede realizar:

- Cálculos Matemáticos
- Expresiones condicionales

- Alertas

Dimensiones

Las dimensiones de un cubo son catálogos de información complementaria necesaria para la presentación de los datos a los usuarios, como por ejemplo: descripciones, nombres, zonas, rangos de tiempo, etc. Es decir, la información general complementaria a cada uno de los registros de la tabla de hechos.

Las dimensiones pueden ser:

Locales: son las que se definen y se utilizan dentro de un mismo DataMart.

Compartidas: son aquellas dimensiones que se definen independientes de los DataMart y pueden ser utilizadas por varios de ellos.

Ventajas de las dimensiones compartidas:

- Evita duplicar dimensiones locales
- Asegura que los datos analizados estén organizados de la misma forma en todos los DataMart, lo que implica un menor costo de mantenimiento.

Desventajas de las dimensiones compartidas:

- Deben emplearse del mismo modo en los DataMart que las usen.
- Un cambio implica que la dimensión deberá ser modificada en todos los DataMart.

Las Dimensiones son categorías descriptivas por las cuales los datos numéricos en un cubo, son separados para su análisis. El cubo puede expandirse para incluir otra dimensión además de soportarla aritmética de matrices. Una dimensión puede ser creada para usarse en un cubo individual o en múltiples cubos.

Una dimensión creada para un cubo individual es llamada dimensión privada. Por el contrario, si esta puede ser usada por múltiples cubos se le llama dimensión compartida; estas podrán ser usadas dentro de todo cubo en la base de datos, así se optimiza el tiempo y se evita el andar duplicando dimensiones privadas.

Las dimensiones compartidas, también habilitan la estandarización de las métricas de negocios entre cubos. Por ejemplo, el estandarizar las dimensiones compartidas para el tiempo y localización geográfica, aseguran que los datos analizados, desde diferentes cubos, estén organizados similarmente. (1)

Jerarquías (9)

En un cubo multidimensional, los atributos de las dimensiones se suelen jerarquizar. Una jerarquía representa una relación lógica entre dos o más atributos pertenecientes a un cubo multidimensional, siempre y cuando posean su correspondiente relación “padre-hijo”.

Las jerarquías poseen las siguientes características:

- Pueden existir varias en un mismo cubo.
- Están compuestas por dos o más niveles.
- Se tiene una relación “1-n” o “padre-hijo” entre atributos consecutivos de un nivel superior y uno inferior.

Grano

La granularidad representa el nivel de detalle al que se desea almacenar la información sobre el negocio que se esté analizando. Mientras mayor sea el nivel de detalle de los datos, se tendrán mayores posibilidades analíticas, ya que los mismos podrán ser resumidos o sumariados. Es decir, los datos que posean granularidad fina (a nivel de detalle) podrán ser resumidos hasta obtener una granularidad media o gruesa. No sucede lo mismo en sentido contrario, ya que por ejemplo, los datos almacenados con granularidad media podrán resumirse, pero no tendrán la facultad de ser analizados a nivel de detalle, o sea, si la granularidad con que se guardan los registros es a nivel de día, estos datos podrán sumariarse por semana, mes, semestre y año, en cambio, si estos registros se almacenan a nivel de mes, podrán sumariarse por semestre y año, pero no lo podrán hacer por día y semana.

Tabla de Hechos (10)

La tabla de hechos es la tabla primaria en el modelo dimensional donde el rendimiento de las mediciones numéricas del negocio es almacenado. .

Cada tabla de hechos contiene, las claves externas, que se relacionan con sus respectivas tablas de dimensiones, y las columnas con los valores que serán analizados.

1.8 Modelos y Arquitecturas habitualmente utilizados en el diseño de un DataMart

Existen tres modelos fundamentales para el diseño de un DM, a continuación se explican cada uno de ellos.

Modelo en Estrella

Es el más sencillo de los esquemas de almacenamiento de datos. Como su nombre lo indica este esquema se asemeja a una estrella donde en el centro de la misma se encuentra la tabla de hechos y las puntas de la estrella son representadas por las tablas de dimensiones que contienen los atributos de las

aperturas que interesan al negocio y que se utilizan como criterios de filtro, y son relativamente pequeñas. Cada tabla de dimensión se vincula con la tabla de hechos por un identificador. Generalmente es un esquema totalmente desnormalizado. (Ver Figura 2)

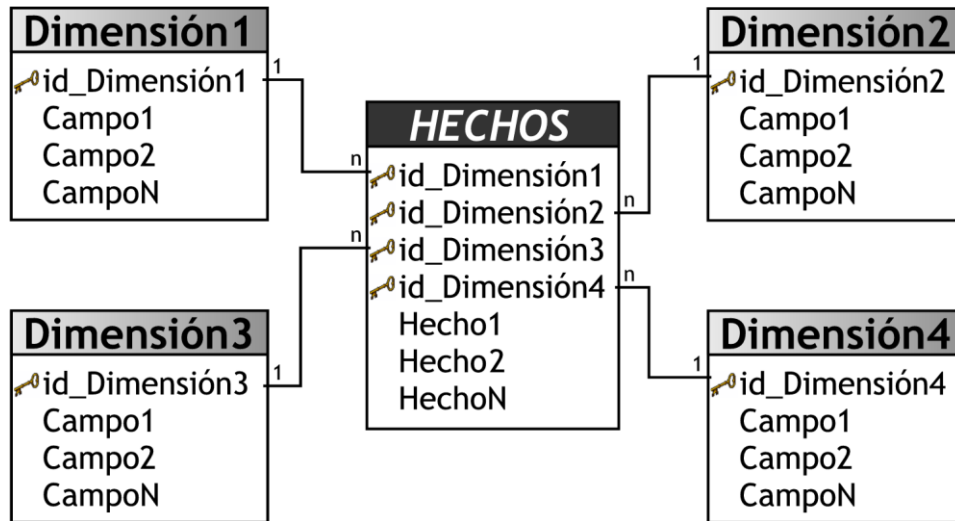


Figura. 2 Modelo en Estrella

Modelo en copo de nieve (Snowflake)

El esquema copo de nieve o snowflake es un esquema derivado del esquema estrella. En él, algunas tablas de dimensiones están normalizadas y se organizan en jerarquías de dimensiones. (Ver Figura 3)

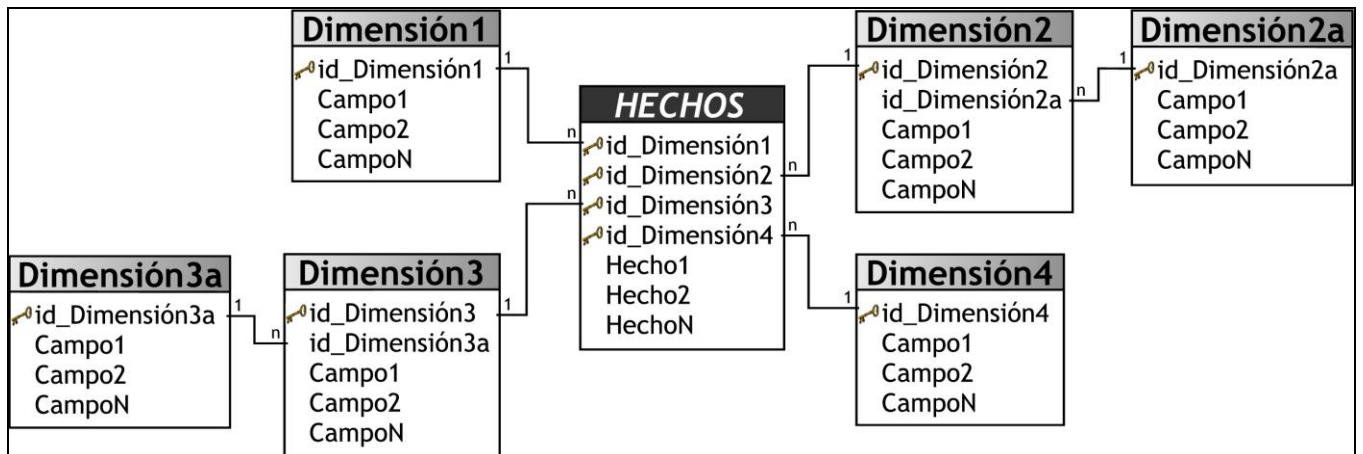


Figura. 3 Modelo en Copo de Nieve

Algunas de las principales características que presenta este modelo se muestran a continuación:

- Posee mayor complejidad en su estructura.
- Hace una mejor utilización del espacio debido a la normalización.

- Es muy útil en tablas de dimensiones de muchas tuplas.
- Las tablas de dimensiones están normalizadas, por lo que requiere menos esfuerzo de diseño.
- Puede desarrollar clases de jerarquías fuera de las tablas de dimensiones, que permiten realizar análisis de lo general a lo detallado y viceversa.

Aunque tiene ventajas presenta algunos inconvenientes tales como:

- Si se poseen múltiples tablas de dimensiones, cada una de ellas con varias jerarquías, se creará un número de tablas bastante considerable.
- Al existir muchas uniones y relaciones entre tablas, el desempeño puede verse afectado.
- La existencia de las diferentes jerarquías de dimensiones debe estar bien fundamentada, ya que de otro modo las consultas demorarán más tiempo en devolver los resultados, debido a que se deben realizar las uniones entre las tablas.

La principal diferencia con el esquema estrella está en la estructura de las tablas de dimensiones; mientras en el esquema estrella estas están totalmente desnormalizadas en el copo de nieve se dividen en otras tablas tras un proceso de normalización.

Uno de los motivos principales para utilizar este tipo de modelo, es la posibilidad de separar los datos de las tablas de dimensiones y proveer un esquema que sustente los requerimientos de diseño, otra razón es que es muy flexible.

Modelo en Constelación

Este modelo consiste en varios esquemas en estrella o copo de nieve que comparten dimensiones. Es decir, algunos atributos se agrupan formando una nueva dimensión que puede ser compartida con otros hechos. (Ver Figura 4)

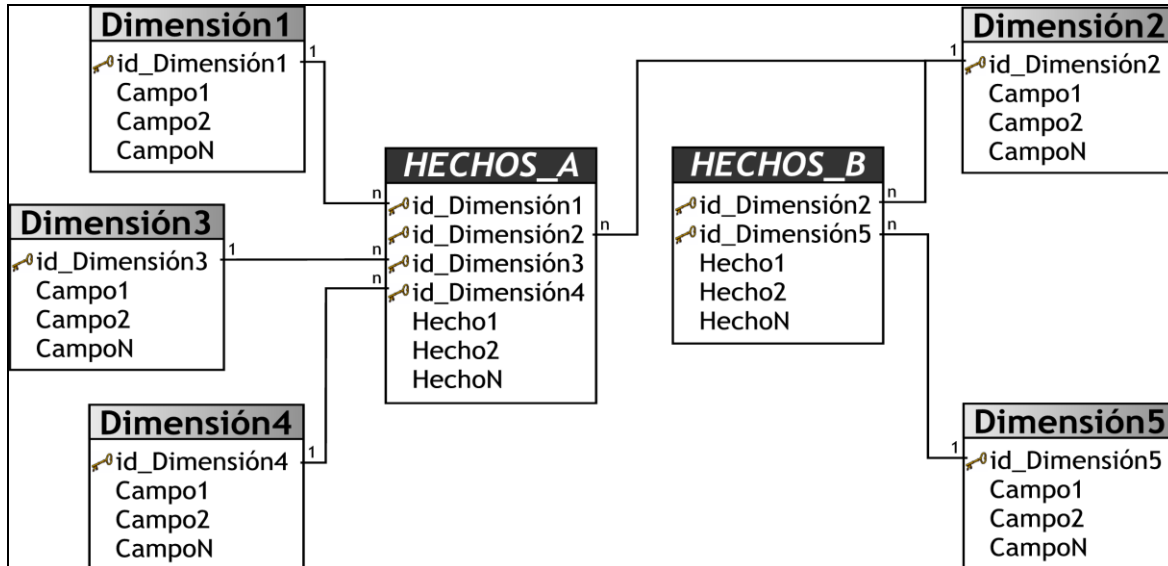


Figura. 4 Modelo Constelación

Las principales ventajas de este esquema son la posibilidad de navegar de un hecho hacia otro mediante la aplicación de la operación drill-across³ y la optimización del espacio gracias a la compartición de dimensiones, evitando así la redundancia de datos.

1.9 Herramientas para la gestión financiera

Para comprender el objetivo de la investigación es necesario conocer que es una razón financiera:

Una razón financiera no es más que método para determinar las relaciones existentes entre los diferentes rubros de los estados financieros, para que mediante una correcta interpretación, se pueda obtener información acerca del desempeño anterior de la empresa y su postura financiera para el futuro cercano.

Una razón expresa la relación matemática entre dos o más cantidades, de ahí que mediante éstas se pueda calcular la relación existente entre algunos conceptos de los estados financieros.

A continuación se hace un estudio del estado actual del cálculo de las razones financieras tanto en sistemas nacionales como internacionales.

En el mundo

eAnaliza

³drill-across: operación multidimensional que permite bajar a los niveles más atómicos de un esquema multidimensional, en sentido inverso al roll-up.

Es una utilidad del tipo Business Intelligence diseñada para analizar información financiera de una empresa o negocio de forma sencilla mediante la combinación de datos y su presentación con gráficos profesionales, tablas dinámicas detalladas, estadísticas y cualquier soporte necesario para mantener y analizar los factores de negocio en una empresa.

Esta herramienta posibilita trabajar fácilmente con tablas dinámicas o cubos de información, trabaja con conexiones a cualquier Base de Datos que soporte ODBC⁴. Es una aplicación de escritorio por lo que se instala en cada ordenador personal en el que se quiere utilizar. Es propietaria, y está construida con herramientas de Microsoft.Net utilizándose en el entorno del sistema operativo Windows siendo compatible con las versiones (Win2000/XP/2003/Vista/7).

Excel (11)

Esta es una de las herramientas más difundidas y utilizadas a la hora de realizar los cálculos de las razones financieras, por su alto nivel de usabilidad y personalización para los usuarios y las entidades donde se va a emplear. Es una herramienta ofimática perteneciente al paquete Microsoft Office producido por la corporación Microsoft. Esta aplicación no es de código abierto y se utiliza en entornos de escritorio, sobre el sistema operativo Windows bajo los derechos de licencia privativa.

IBM Cognos TM1

Es un software de planificación empresarial que ofrece un completo y dinámico entorno para la elaboración de previsiones y presupuestos, actuales, fiables y personalizados, y la realización de la gestión financiera. Posee un motor OLAP en memoria de 64 bits multidimensional que ofrece un funcionamiento rápido permitiendo analizar modelos complejos y sofisticados, así como grandes conjuntos de datos e incluso secuencias de datos.

Da soporte a una completa gama de requisitos de software de planificación empresarial: desde el análisis de rentabilidad que demanda alto rendimiento, el análisis financiero y el modelado flexible, hasta la contribución de información desde todas las unidades de negocio de la empresa. Esta herramienta permite que el departamento financiero y las líneas de negocio tengan control total sobre los procesos de planificación, la elaboración de presupuestos y previsión de resultados. El costo de la licencia establecido

⁴ODBC: Open Data Base Connectivity (ODBC) es un estándar de acceso a bases de datos desarrollado por SQL Access Group en 1992, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos

por la IBM para este sistema resulta alto, es una aplicación de escritorio, desarrollada sobre el sistema operativo Windows y no es de código abierto.

SAP (Sistemas, Aplicaciones y Productos para Procesamiento de Datos)

Dentro de las herramientas tecnológicas para soluciones de negocios integradas, se destaca SAP Business One como aplicación para la obtención de las razones financieras.

Esta herramienta integra las funciones de negocio a través de toda la compañía incluyendo la gestión financiera y otras operaciones contables. Dentro de sus módulos principales se encuentran los relacionados con:

- Contabilidad y Finanzas
- Gestión de relaciones con clientes (CRM)
- Operaciones y Distribución.
- Gestión de almacenes y producción
- Administración y Presentación de Informes (12)

La aplicación soporta múltiples estándares globales de información financiera, proporcionando informes que permiten gestionar eficazmente el flujo de caja y la rentabilidad para el negocio. Posibilita optimizar la liquidez y prever con exactitud el flujo monetario y las necesidades de efectivo dándole el poder a la empresa para manejar eficientemente las condiciones de pago, avisos de clientes y cobros en efectivo. Es considerada una de las herramientas más completas utilizadas actualmente y a su vez una de las más costosas. Es una aplicación de escritorio compatible con el sistema operativo Windows y con el gestor de base de datos SQL Server 2005.

ORACLE

Dentro de la gama de aplicaciones de ORACLE pertenecientes a la colección de planificación de recursos empresariales se destacan por la generación de las razones financieras las aplicaciones GL Wand e Hyperion Financial Reporting.

GL Wand: es una herramienta que posibilita integrar con Excel la información financiera del sistema Oracle e-Business Suite⁵, permite hacer uso de las funciones de Excel como fórmulas y gráficas. Dentro de sus funcionalidades se encuentra realizar el análisis de las razones financieras a partir de instalar una nueva

⁵Oracle e-Business Suite: Oracle E-Business Suite es una suite de aplicaciones empresariales globales totalmente integradas

barra de herramientas en Excel desde donde se accede de forma directa a toda la información financiera del sistema Oracle e-Business Suite.

Hyperion Financial Reporting: Es una solución Web que consolida datos financieros a través de diferentes sistemas mejorando su gestión. Es utilizada fundamentalmente por las agencias de gobierno las cuales deben mantener y cerrar sus libros y luego informar los resultados financieros con confianza, en ellas los organismos gestores de las finanzas para poder integrar los resultados financieros y operativos de múltiples sistemas en un mismo punto de vista, sin retrasos, necesitan de una única versión de la verdad. Permite realizar análisis multidimensionales, consolidar datos financieros y realizar la entrega de los mismos para apoyar las decisiones de gestión estratégicas y operativas de la empresa.

En ambos casos las soluciones al pertenecer al conjunto de herramientas de ORACLE Suite son altamente costosas y de código cerrado.

OPEN-ERP

Se describe a sí mismo como el ERP de Código abierto más destacado y sencillo que existe hasta el momento, es libre, liberado bajo la licencia GPL. Emplea a Postgresql como sistema manejador de bases de datos y ha sido programado con Python. Este multiplataforma, funciona sobre Linux y Windows.

La arquitectura del sistema es cliente – servidor, lo que permite que todos los usuarios trabajen sobre el mismo repositorio de datos. Esto tiene la ventaja de que toda la información está disponible y sincronizada en todo momento además descarga la mayor parte del trabajo de procesamiento de datos en las máquinas cliente (donde trabajan efectivamente los usuarios).

Dentro de sus módulos se encuentran los relacionados con:

- Contabilidad analítica
- Contabilidad financiera
- Gestión de almacenes/inventario
- Gestión de ventas y compras
- Automatización de tareas
- Campañas de marketing
- Punto de venta

Este sistema añade en la mayor parte de sus áreas herramientas de análisis y generación de informes, con lo que la gestión y visualización de la información se simplifica. Posibilita la creación de informes

multiplataforma, con HTML y Css3 haciendo que las posibilidades de esta combinación sean casi ilimitadas aunque dentro de sus funcionalidades no se encuentra la obtención de las razones financieras.

OPEN-BRAVO

Este sistema incluye una extensa cobertura funcional que proporciona un sólido fundamento para la agilidad en el negocio. Se basa en un modelo con una base de datos única e integrada que cubre todas las áreas principales de un sistema de gestión completo del negocio. Dentro de sus módulos se encuentran los relacionados con:

- Gestión de datos maestros
- Gestión de aprovisionamientos
- Gestión De almacenes
- Gestión De proyectos y servicios
- Gestión De producción
- Gestión comercial y CRM
- Finanzas y Contabilidad
- Inteligencia de negocios
- Punto de venta

Posee un componente de Inteligencia de negocios, integrado en el sistema de gestión, que proporciona la información relevante para la toma de decisiones, posibilitando realizar reportes predefinidos que permiten verificar el estado de la empresa aunque no posibilita obtener las razones financieras.

Es una aplicación de código abierto de gestión empresarial del tipo ERP, con arquitectura cliente/servidor web escrita en Java. Se ejecuta sobre Apache y Tomcat y con soporte para bases de datos PostgreSQL y Oracle. Openbravo Community Edition está licenciado bajo Openbravo Public License Version 1.1 ("OBPL"), que es una adaptación de la licencia libre Mozilla Public License. Es compatible sobre los siguientes sistemas operativos Windows, Linux, Unix, Solaris, FreeBSD (13)

En Cuba

Versat Sarasola: (14)

Versat Sarasola fue desarrollado en 1998 por TEICO Villa Clara, empresa del Ministerio del Azúcar encargada de la Informática y las Comunicaciones. Automatiza las actividades de planificación, control y análisis económico de cualquier tipo de entidad, abarcando la administración, contabilidad, los medios de

rotación, los activos fijos, las finanzas, cajas y costos. Aunque no permite el cálculo de razones financieras.

Está constituido por 12 módulos que incluyen configuración y seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial y control de activos fijos. Además, interviene finanzas y cajas, planificación y presupuestos, control de inventarios, de productos terminados, pago de salario, paquete de gestión, contratación y facturación.

El Versat Sarasola fue seleccionado por el Ministerio de Finanzas y Precios (MFP) como la herramienta Informática más adecuada para implantarse en varios Centros de Gestión Contables del País. Es compatible solamente con el sistema operativo Windows, utiliza el motor base datos SQL. Es una aplicación de escritorio implementado en Delphi y es un software propietario.

Siscont-5

Es un sistema desarrollado por la empresa de Tecnologías de la Información, Automática y las Comunicaciones (TECNOMATICA), que se aviene a las definiciones y conceptos del Ministerio de la Industria Básica por las acciones contables financieras que permite pueda ser utilizado en otras entidades nacionales.

Este sistema permite trabajar en forma monousuario, multiusuario y por Internet, dependiendo el tipo de hardware que el cliente adquiera bajo el sistema operativo Windows XP o superior. Se actualiza totalmente gratis desde la página Web de SISCONT y en cuanto a la seguridad posibilita configurar los permisos de cada usuario, de esta forma controla que el personal acceda únicamente a las funciones autorizadas. Fue hecho en la herramienta de desarrollo de software basada en conocimiento GeneXus⁶ y presenta como soporte para bases de datos SQL Server 2000.

Assets NS

Es un sistema comercializado por la firma panameña D'MARCO S.A. y distribuido en Cuba en el año 1997 por INFOMASTER, entidad informática perteneciente a la Empresa Nacional de Producción y Servicios a la Educación Superior del MES. Es un sistema flexible, amigable, y funciona en ambiente multiusuario incluidas estaciones remotas.

⁶GeneXus: Es una herramienta de desarrollo de software basada en conocimiento, orientada principalmente a aplicaciones de clase empresarial para la web, plataformas Windows y Smart Devices. Esta herramienta permite generar código en múltiples lenguajes.

Es un Sistema de Gestión Integral estándar y parametrizado capaz de adaptarse a las exigencias de cada entidad en particular; que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos.

Permite el control por Fondos de efectivo; pueden existir distintos tipos de fondos de acuerdo a los requerimientos de cada Empresa. Se procesan los documentos recibidos en los procesos de Cobros y Cobros Directos y los Pagos Menores realizados desde Caja. Controla los pagos de Anticipos para gastos de viajes y otros conceptos y la liquidación y justificación de los mismos. Genera, automáticamente, los asientos de diario a la contabilidad por cada una de las transacciones contempladas en el sistema.

Permite realizar Arqueo de Caja, que consiste en el análisis de las transacciones que afectan el fondo en un período, con el objetivo de verificar si el saldo según transacciones, corresponde con lo que se encuentra físicamente en Caja en dinero efectivo, cheques o vales. Se puede además realizar el Reembolso de los Fondos. (10)

Es una aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente Crystal Reports 7.0 para la generación de reportes de salidas. (15)

1.9.1 Comparación de las herramientas

Una vez realizado el estudio de un grupo de herramientas tanto nacionales como extranjeras viendo sus potencialidades y características técnicas, se puede afirmar que ninguna de las herramientas estudiadas satisface a cabalidad las necesidades del proyecto (cálculo de razones financieras e integración con el Cedrux). Ya que la mayoría de las herramientas estudiadas tanto las nacionales como las foráneas aunque posibilitan la gestión financiera, no realizan cálculo de las razones financieras lo que dificulta el análisis financiero, y las que lo realizan tienen licencias propietarias lo que conlleva al pago de altos precios por las mismas, por otro lado son aplicaciones de escritorio lo que dificulta la integración con el Cedrux que está desarrollado sobre una plataforma web. Otro problema de integración es que la mayoría utiliza como motor de base datos a SQL y no a Postgres como está establecido en el Cedrux. Por lo antes planteado se hace necesario entonces implementar un sistema que satisfaga las necesidades del proyecto y permita el cálculo de las razones financieras para la toma de decisiones.

1.10 Metodologías para el diseño de un DW

Con el desarrollo y avance en el nivel de conocimientos en los procesos de construcción de un DW han surgido un conjunto de metodologías que agilizan y rigen los procesos de desarrollo de dichos sistemas.

El vocablo metodología, es la ciencia que estudia los métodos del conocimiento. Se refiere a los métodos o procedimientos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia.

1.11 Metodología Hefestos

Es una metodología propia que goza de gran popularidad, cuya propuesta está fundamentada en una muy amplia investigación. Ella plantea que la construcción e implementación de un DW puede adaptarse muy bien a cualquier ciclo de vida de desarrollo de software, con la salvedad de que para algunas fases en particular, las acciones que se han de realizar serán muy diferentes. Lo que se debe tener muy en cuenta, es no entrar en la utilización de metodologías que requieran fases extensas de reunión de requerimientos y análisis, fases de desarrollo monolítico que conlleve demasiado tiempo y fases de despliegue muy largas. Lo que se busca, es entregar una primera implementación que satisfaga una parte de las necesidades, para demostrar las ventajas del DW y motivar a los usuarios. (10)

Entre sus características están:

- Los objetivos y resultados esperados en cada fase se distinguen fácilmente y son sencillos de comprender.
- Utiliza modelos conceptuales y lógicos, los cuales son sencillos de interpretar y analizar.
- Es independiente de las estructuras físicas que contengan el Data Warehouse y de su respectiva distribución.
- Se basa en los requerimientos del usuario, por lo cual su estructura es capaz de adaptarse con facilidad y rapidez ante los cambios en el negocio.
- Reduce la resistencia al cambio, ya que involucra al usuario final en cada etapa para que tome decisiones respecto al comportamiento y funciones del Data Warehouse.

Esta metodología define 4 fases principales, la cuales su vez definen un conjunto de pasos para desarrollar el DM. Las fases son Análisis de los requerimientos, Análisis de los OLTP, elaboración del modelo lógico de la estructura del DW y procesos ETL, limpieza de datos y sentencias SQL. A continuación se detallan las fases y los procesos que las componen:

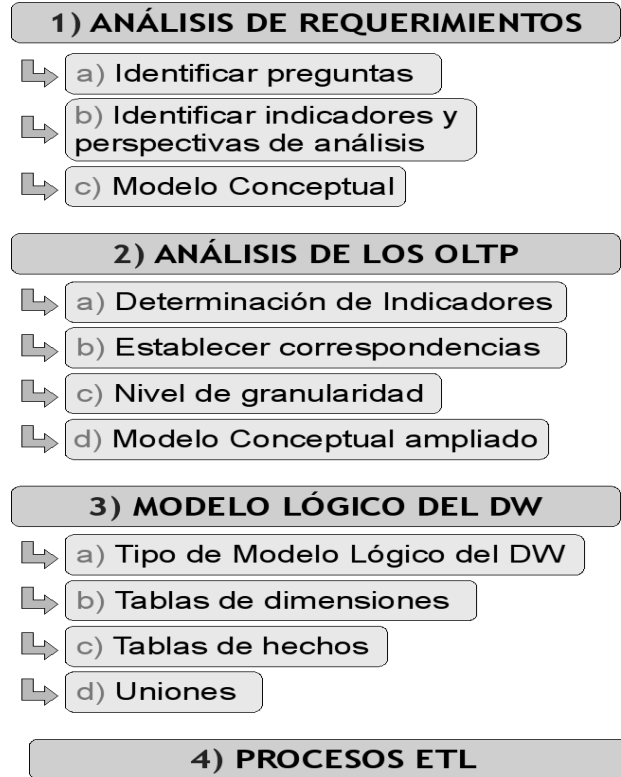


Figura. 5 Fases de la metodología HEFESTOS

Aunque define muy bien las fases y pasos a seguir no especifica los artefactos que se deben generar en cada fase.

1.12 Ciclo de vida Kimball

A continuación se presenta una figura donde está el ciclo de vida concebido para el desarrollo de esta metodología (ver figura 6):

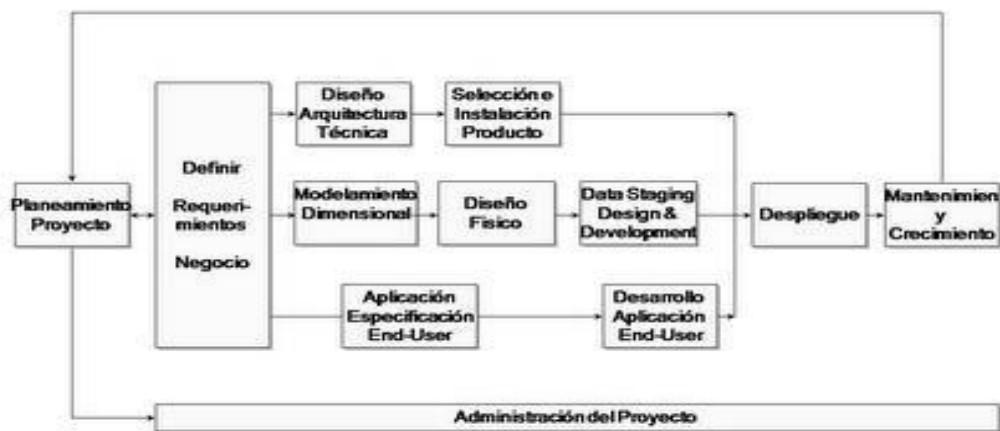


Figura. 6 Ciclo de vida de Kimball

El ciclo Kimball comienza con una planificación de proyecto, en la cual se define el alcance, se identifican y programan las tareas, se planifica el uso de los recursos, conformado con todo esto el plan de proyecto. En la segunda etapa de este ciclo se definen los requerimientos del negocio. Luego de definir los requerimientos del negocio, el proyecto se enfoca en tres líneas concurrentes: tecnología, datos y aplicaciones de la inteligencia de negocios (ver figura 6).

La etapa de diseño del DW, está enmarcada en la línea de datos, donde se realiza el modelo dimensional y se analizan los datos del negocio para identificar la granularidad de las tablas de hechos, dimensiones y atributos asociados. Las construcciones primarias que se hacen en esta etapa son las tablas de dimensiones y de hechos. Las primeras contienen las métricas derivadas de los procesos de negocio o eventos. Hay que tener en cuenta que la granularidad debe ser lo más atómica posible, lo que permite una mayor flexibilidad y extensibilidad. Por su parte, las tablas de dimensiones contienen la descripción de atributos y características asociadas con medidas de eventos tangibles y específicos.

Para terminar el ciclo de vida se realiza el despliegue con el objetivo de dejar sentadas las bases de crecimiento y mantenimiento del DW.

Kimball apunta a una solución completa que se puede implementar en poco tiempo. Esto cumple con lo que se necesita en una metodología de Business Intelligence (BI), dar resultados rápidos para demostrar el valor de la solución al negocio y no perder la confianza de los usuarios. Se enfoca principalmente en el diseño de la base de datos que almacenará la información para la toma de decisiones.

1.13 Metodología de Inmon

Esta metodología la definió su autor en el año 1992 en el libro "Building the Data Warehouse". En él proponía los mecanismos necesarios para llevar a cabo la correcta realización de un DW. Para Bill Inmon, el diseño de un DW comienza ya con la mera introducción de datos en el mismo. En su filosofía, un DM es sólo una de las capas del DW y los DM son dependientes del depósito central de datos o DW Corporativo y por lo tanto se construyen después de él. El enfoque de Inmon de desarrollar una estrategia de DW e identificar las áreas principales desde el inicio del proyecto es necesario para asegurar una solución integral ya que esto ayuda a evitar la aparición de situaciones inesperadas que puedan poner en peligro el proyecto, debido a que se conoce con antelación y bastante exactitud la estructura que presentarán los principales núcleos del desarrollo, lo que permite enfocar los esfuerzos del desarrollo actual para ser compatible con los subsiguientes.

De forma contraria a la de Kimball esta metodología puede tener una implementación mucho más tardada, y es recomendada cuando se hace demasiado difícil representar el modelo a través de dimensiones y la

complejidad de la solución se hace demasiado grande, pero no es muy recomendable para proyectos sencillos ya que va de lo más general el DW a lo más específico DM.

1.14 Metodología de DATEC

Fue creada en el año 2009 por el centro CENTALAD ⁷en la Universidad de las Ciencias Informáticas (devenido en DATEC en el año 2010). Está basada fundamentalmente en el enfoque Kimball, consta de 5 fases y un total de 43 artefactos. En esta se definen además los hitos para su desarrollo así como los roles y sus responsabilidades en el mismo definiendo también las herramientas a emplear en cada una de sus fases. Su aporte radica en la integración de algunas prácticas de RUP⁸ con este enfoque. Si bien esto ayuda a obtener una documentación amplia para el proyecto implica el empleo de mayores recursos como tiempo, esfuerzo y personas en la generación de estos artefactos. Las fases definidas en esta metodología son:

- Fase requerimientos
- Fase de Arquitectura y Diseño
- Fase de Implementación y Prueba
- Fase de despliegue
- Fase de soporte

1.15 Metodología UCID (16)

Esta metodología es un híbrido entre la metodología de Hefestos y la de Kimball adoptada por el centro de datos de la UCI adaptándola a las necesidades propias de la UCID (Centro para la compatibilización). Ella permite aprovechar la facilidad que brinda Hefestos de una metodología sencilla y entendible con la de Kimball de generar artefactos para la documentación a la hora de realizar el DW.

Sus principales características son:

- Los objetivos y resultados esperados en cada fase se distinguen fácilmente y son sencillos de comprender.
- Se basa en los requerimientos del usuario, por lo cual su estructura es capaz de adaptarse con facilidad y rapidez ante los cambios en el negocio.

⁷CENTALAB: Centro dedicado al trabajo con los almacenes de datos en la Universidad de las Ciencias Informáticas

⁸ RUP: El **Proceso Racional Unificado** (*Rational Unified Process* por sus siglas en inglés) es un proceso de desarrollo de software y constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- Reduce la resistencia al cambio, ya que involucra al usuario final en cada etapa para que tome decisiones respecto al comportamiento y funciones del DW.
- Utiliza modelos conceptuales y lógicos, los cuales son sencillos de interpretar y analizar.
- Es independiente del tipo de ciclo de vida que se emplee para contener la metodología.
- Es independiente de las herramientas que se utilicen para su implementación.
- Es independiente de las estructuras físicas que contengan el DW y de su respectiva distribución.
- Cuando se culmina con una fase, los resultados obtenidos se convierten en el punto de partida para llevar a cabo el paso siguiente.
- Se aplica tanto para Data Warehouse como para DataMart.

Como está basada en HEFESTOS presentas las mismas 4 fases de desarrollo, con la diferencia de que se le han incorporado algunos pasos en cada una de la fases y se han adoptado y creado artefactos para documentar lo que se realiza en cada fase, lo cual sirve para la reutilización del conocimiento en otros proyectos.

Las fases son:

Análisis de los requerimientos:

- Planificar Entrevista
- Identificar Preguntas
- Identificar Perspectivas e Indicadores
- Construir Modelo Conceptual
- Aprobación del Modelo Conceptual por el cliente

Análisis de los OLTP

- Definir Estado General de los sistemas fuentes
- Determinación de indicadores
- Establecer correspondencias
- Nivel de granularidad
- Construir Modelo Conceptual ampliado
- Definir reglas del negocio

Modelo físico del DW

- Definir Tipo de Modelo Lógico del DW
- Definición de estándares para objetos físicos
- Identificar dimensiones

- Identificar hechos
- Realizar Uniones entre dimensiones y hechos
- Diseñar tablas y columnas físicas

Proceso ETL

- Mapeo de datos fuente a destino.
- Establecer condiciones adicionales y restricciones.
- Carga incremental de datos.
- Diseño, construcción de la automatización del sistema de ETL.

1.16 Justificación de la metodología a utilizar

Las metodologías de Inmon y Kimball se contraponen entre sí, en la forma de enfrentar el problema mientras el primero plantea ir de lo más general a lo más específico, o sea construir un DW del cual se nutrirán posteriormente los DM que se crean en los departamentos. EL segundo plantea todo lo contrario: partir de pequeños DM por departamentos hasta consolidar la información en un Data Warehouse a partir de los DM creados. La teoría de Inmon no es buena para proyectos sencillos, la solución de Kimball es ideal para proyectos de poco tiempo, debido a que plantea que se debe crear por cada departamento un conjunto de DataMarts independientes orientados a los temas que estén relacionados con él. En este caso la metodología de Kimball se adapta mejor a las necesidades del proyecto.

Por otro lado la metodología DATEC se adapta a proyectos grandes ya que organiza muy bien sus fases pero se generan gran cantidad de artefactos como resultado de las mismas, lo que no se adecua a las necesidades de la implementación del DM. La metodología HEFESTOS a pesar de ser sencilla y entendible no define de forma explícita los artefactos que se generan en sus fases. Es por eso que se selecciona la metodología UCID para la implementación del DM, ya que esta integra los factores positivos de la metodología HEFESTOS y Kimball de manera que se satisfagan los requerimientos del proyecto.

1.17 Herramientas para la construcción del DataMart

1.17.1 Sistemas Gestores de Bases de Datos

ORACLE

Oracle es un Sistema Gestor de Bases de Datos (SGBD) con características objeto-relacionales, que pertenece al modelo evolutivo de SGBD. Es un sistema multiplataforma y está distribuido bajo licencia propietaria. Es un sistema muy extendido para las compañías que los pueden pagar ya que es muy costoso. No es una herramienta de código abierto.

Entre sus características se encuentran:

- Entorno cliente/servidor.
- Gestión de grandes bases de datos.
- Usuarios concurrentes.
- Alto rendimiento en transacciones.
- Sistemas de alta disponibilidad.
- Disponibilidad controlada de los datos de las aplicaciones.
- Adaptación a estándares de la industria, como SQL-92.
- Gestión de la seguridad.
- Autogestión de la integridad de los datos.
- Opción distribuida.
- Portabilidad.
- Compatibilidad.
- Conectividad.
- Replicación de entornos. (17)

MYSQL

MySQL Server es la base de datos de código fuente abierto.

Características principales:

- Escrito en C y C++.
- Trabaja bajo diferentes plataformas: AIX 4x 5x, Amiga, BSDI, Digital Unix 4x, FreeBSD 2x 3x 4x, HP-UX 10.20 11x, Linux 2x, Mac OS, NetBSD, Novell NetWare 6.0, Open BSD 2.5, OS/2, SCO Open Server, SCO UnixWare 7.1.x, SGI Irix 6.x, Solaris 2.5, SunOS 4.x, Tru64 Unix y Windows 9x, Me, NT, 2000, XP, 2003.
- Procesos Multihilo. Capacidad de trabajar servidores con varios procesadores. Velocidad en la utilización de joins y procesos de optimización, manejo de la memoria a través de manejo del buffer y cache.

Dentro de sus ventajas se encuentran:

- Velocidad al realizar operaciones.
- Es de software libre.
- Su conectividad y robustez.
- Escalabilidad.
- Acceso a las bases de datos de forma simultánea.

- Seguridad, en forma de permisos y privilegios.
- Bajo costo en requerimiento para la elaboración de bases de datos.
- Gratuito.

Desventajas:

- Lento con grandes bases de datos
- No tiene tantas capacidades como otros gestores profesionales
- Dificultad de manejo
- No es intuitivo como otros programas.
- Un gran porcentaje de utilidades de Mysql no están documentadas

POSTGRESQL

PostgreSQL es un servidor de base de datos relacional⁹, libre. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y joins de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Tiene entre otras ventajas las de contar con una gran comunidad de desarrollo en Internet, su código fuente es abierto, y el gestor es multiplataforma. Fue diseñado para ambientes de alto volumen. Tiene buen desempeño al aumentar el número de CPUs¹⁰ y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene buen soporte para vistas y procedimientos almacenados en el servidor, así como para las transacciones y el almacenamiento de objetos de gran tamaño. (18)

1.17.2 Herramientas de integración de datos

CLOVER ETL

Es un ambiente de transformación de datos de código abierto basado en Java, para datos estructurados, capaz de funcionar como aplicación independiente (standalone) o estar incluida en otra aplicación.

Características principales:

- Realiza transformaciones gráficas basadas en XML para la descripción de los metadatos de los registros.

⁹ Base de datos relacional: Modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente basados en el empleo de relaciones entre las tablas.

¹⁰ CPU: La unidad central de procesamiento o CPU (por el acrónimo en inglés de central processing unit), o simplemente el procesador o microprocesador.

- Se distribuye bajo la licencia LGPL¹¹.
- CloverETL soporta 4 diferentes tipos de datos: string, numeric, date, bytes.
- Arquitectónicamente, está conceptualizado en unidades lógicas separadas llamadas unidades de transformación que engloban funcionalidades de la transformación e inteligencia, cada una de las cuales puede ser utilizada como componente de una aplicación independiente en otras aplicaciones y servicios. Cada componente corre como un hilo de ejecución separado, creando un ambiente más tolerante a fallas.

ENHYDRAOCTOPUS

Es una herramienta de ETL para transformaciones de datos por conexión JDBC [JDBC, 2007]. Octopus solo soporta fuentes de datos que vengan con el manejador JDBC, incluye también drivers especiales que permiten la conectividad con archivos CSV, XML, MS-SQL y archivos de propietarios. Octopus utiliza archivos XML para cargar los trabajos, así como para definir los parámetros de las transformaciones dadas.

Características principales:

- La característica principal de Octopus es el requerimiento de que para acceder a las fuentes de datos que lo utilicen debe existir un manejador JDBC disponible.
- Herramienta poderosa capaz de: normalizar datos, crear llaves artificiales, tablas y llaves primarias.
- Todos los trabajos que corren en Octopus son independientes de los proveedores de bases de datos. (19)

KETTLE (20)

Es una herramienta belga, de tipo ETL Open Source conocido como Pentaho Data Integration (Herramienta de integración de datos de Pentaho). El mismo incluye un conjunto de herramientas para realizar los procesos de ETL.

Se compone de 4 herramientas:

- SPOON: permite diseñar de forma gráfica la transformación ETL.
- PAN: ejecuta un conjunto de transformaciones diseñadas con SPOON, conocidas como trabajos (jobs), creando dependencias entre dichas transformaciones.

¹¹ LGPL: Licencia Pública General Reducida de GNU, es una licencia de software creada por la Free Software Foundation. Los contratos de licencia de la mayor parte del software están diseñados para jugar con su libertad de compartir y modificar dicho software.

- CHEF: permite, mediante una interfaz gráfica, diseñar la carga de datos incluyendo un control de estado de los trabajos.
- KITCHEN: permite ejecutar los trabajos batch¹² diseñados con Chef.

Características principales:

- Tiene una interfaz visual con indicadores de las transformaciones.
- Es una aplicación escrita en Java con algunas características avanzadas escritas en Java Script.
- Basado en metadatos.
- Como soporte se encuentran los foros de Pentaho y la comunidad Pentaho.
- Con respecto a escalabilidad, soporta la arquitectura de procesamiento en paralelo para distribuir las tareas de ETL a través de múltiples servidores.
- Es compatible con los operativos como Windows, Unix y Linux.
- Está distribuido bajo la licencia pública GPL.
- Soporta Oracle, DB2, SQL Server y Sybase así como MySQL, PostgreSQL, Hypersonic, FireBird SQL e Ingres.
- Posee una interfaz gráfica con indicadores de transformación visual.
- Como software libre tiene una comunidad donde se le va haciendo mejoras.
- Está distribuido bajo la licencia Mozilla Public License.
- Es de código abierto.

Talend

Es una herramienta francesa, de tipo ETL Open Source. Entre sus funciones están la alimentación del DM, sincronización de las bases datos, transformación y verificación de la integridad de los datos, y administración de metadatos.

Características: (21)

- Plataforma: se ejecuta en Windows, Unix y Linux.
- Interfaz: Posee una interfaz que permite modificar sus componentes de forma sencilla a través de arrastrar y soltar.
- Código: Genera componentes como Perl, Java o código nativo de SQL para un ETL y / o enfoque ELT transformación.

¹²Proceso *batch*: es el proceso por lotes; se establece cuando las características de la información a procesar sean tales que no requieren un tiempo de respuesta inmediata.

- Fuente: Java / Eclipse código fuente está disponible para su descarga y personalización.
- Comunidad: Tiene una comunidad social con una enciclopedia en línea de Talend.
- Conectividad: Permite conectarse con a Oracle, DB2, MySQL, Sybase y PostgreSQL. Conectividad ODBC para bases de datos.
- Escalabilidad: Soporta el procesamiento de la red y una combinación de ETL y ELT para aprovechar la capacidad de procesamiento de la arquitectura.

Ventajas:

- Es una herramienta de código abierto distribuido bajo la licencia GPL, lo que hace que sea adecuado como herramienta de integración de datos integrados en un producto complementario, tales como inteligencia de negocios, calidad de los datos actualización del producto.
- Posee una comunidad que ofrece un lugar para obtener soporte técnico y el foro proporciona soporte técnico gratuito. Si bien este foro viene sin ningún tipo de servicio garantiza que es supervisado por personal de la empresa Talend para mantener la calidad del producto.

Desventajas:

Las transformaciones son prácticamente a base de código (de momento pearl y se supone que en un futuro java).

- Es menos intuitivo que kettle
- No hay perfiles complementarios, calidad de datos o gestión de metadatos de producto. Estas funciones se pueden concretar con terceros productos pero actualmente existen pocos sistemas de software libre disponibles para esto.
- Permite asignar recursos a través de la red pero no tiene la automatización subyacente del particionamiento y reparticionamiento de datos y esto podría dar lugar a cuellos de botella entre las tareas o trabajos del proceso ETL.
- Necesita el driver JDBC para acceder a las fuentes de información.

Luego de ver las características de cada una de las posibles herramientas para desarrollar un DataMart. Se seleccionó el Kettle ya que existe experiencia en su trabajo en el centro UCID. Es un sistema multiplataforma, se distribuye bajo una licencia libre, y soporta distintos sistemas de bases de datos, lo que posibilita la integración con el Cedrux.

1.17.3 Herramientas para OLAP

JEDOX PALO

Jedox Palo es un servidor de bases de datos multidimensional capaz de centralizar y administrar casi un número infinito de hojas de cálculo. El sistema opera en tiempo real, soporta la consolidación de jerarquías así como numerosas funciones de inteligencia empresarial y es un servidor de código abierto. Palo es un servidor de datos multidimensional orientado a celdas, específicamente desarrollado para almacenamiento y análisis de datos en hojas de cálculo.

OLAP4J

OLAP4J (Online Analytical Processing for Java) es una interfaz de aplicación para el ambiente Java 2 Platform, Enterprise Edition, que soporta la creación, almacenamiento y administración de datos para una aplicación OLAP. Es compatible con Mondrian.

MONDRIAN

Mondrian es el motor OLAP integrado en la suite de Business Intelligence Open Source Pentaho. Es un proyecto Open Source y está licenciado bajo la Mozilla Public License (MPL). Permite analizar de forma interactiva grandes cantidades de información almacenada en cualquier Base de Datos que soporte JDBC, sin la necesidad de escribir SQL. (22)

Características de este sistema:

- Proporciona conexiones y ejecuta consultas SQL contra la base de datos relacionales que sirve los datos (ROLAP)
- Está orientado para la web (WOLAP)
- Trabaja con las bases de datos: Oracle, PostgreSQL, MySQL, Microsoft Access, Firebird, IBM, DB2, Microsoft.

Para acceder a las funcionalidades que brinda Mondrian se requiere de un cliente, es el caso de **JPivot** o **JRubik**. **JPivot**: Es una librería de Java Server Pages (JSP) ¹³personalizados que presentan tablas OLAP y permite realizar operaciones tales como filtrado hacia arriba y filtrado hacia abajo además de consultas OLAP por medio del lenguaje MDX. Está diseñado para trabajar con motores OLAP como el de Mondrian por lo cual no utiliza las interfaces de programación de aplicaciones o (APIs) de Mondrian directamente sino que implementa su propio modelo OLAP.

¹³ JSP: **JavaServer Pages (JSP)** es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas.

JRubik: Está basado en los componentes de JPivot y también se puede conectar a fuentes OLAP basadas en Mondrian. Las consultas son realizadas mediante el lenguaje MDX y cuenta con componentes similares a los de JPivot, solo que su interfaz no está basada en JSP sino en Swing, por lo que es una aplicación de escritorio.

Otra herramienta que utiliza el Mondrian es el **SCHEMA WORKBENCH (KETTLE)**.

SCHEMA WORKBENCH (KETTLE)

Es un entorno visual para el desarrollo y prueba de cubos OLAP Mondrian. Esta herramienta se utiliza para la creación de los archivos XML que se usan para la construcción de los cubos. Además permite la ejecución de consultas MDX¹⁴ contra el esquema y la base de datos.

Después de haber analizado se elige Mondrian como herramienta para OLAP por su alto desempeño, análisis interactivo de grandes o pequeños volúmenes de información y exploración dimensional de los datos.

1.17.4 Herramientas CASE

VISUAL PARADIGM

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Entre sus principales características podemos encontrar:

- Soporta aplicaciones Web.
- Es un producto de calidad.
- Fácil de instalar y actualizar.
- Posee versiones para múltiples plataformas.
- Compatibilidad entre ediciones. (18)

Una de sus principales ventajas es que la UCI posee la licencia para utilizar esta herramienta.

¹⁴ MDX: Las expresiones multidimensionales (MDX es el acrónimo de MultiDimensional eXpressions) son un lenguaje de consulta para bases de datos multidimensionales sobre cubos OLAP, se utiliza en Business Intelligence para generar reportes para la toma de decisiones basados en datos históricos, con la posibilidad de cambiar la estructura, o permitiendo rotar el cubo.

1.17.5 SERVIDOR DE APLICACIONES

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Combinando una arquitectura orientada a servicios, revolucionaria con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado y distribuido sin restricciones por la licencia. (23)

JBOSS ó JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible Java. Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., adquirida por Red Hat en abril del 2006, fundada por Marc Fleury, el creador de la primera versión de JBoss. El proyecto está apoyado por una red mundial de colaboradores. Los ingresos de la empresa están basados en un modelo de negocio de servicios.

JBoss implementa todo el paquete de servicios de J2EE.

TOMCAT

Tomcat es un servidor Web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache. Tomcat puede funcionar como servidor Web por sí mismo, que puede ser usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual. Se selecciona como servidor de aplicaciones para la construcción del DM el JBoss ya que es un servidor de aplicaciones de código abierto implementado en Java. Al estar basado en Java, puede ser utilizado en cualquier sistema operativo que lo soporte. Combina una arquitectura orientada a servicios revolucionaria con una licencia de código abierto permitiendo ser descargado, utilizado, incrustado y distribuido sin restricciones por la licencia. (23)

Conclusiones

Una vez finalizado el presente capítulo llegamos a las siguientes conclusiones:

- Los sistemas para el cálculo de las razones financieras en Cuba y el mundo no cumplen con los requisitos necesarios para utilizarlos en el proyecto ERP-Cuba.
- Se hace necesaria la construcción de un DataMart para calcular las razones financieras de módulo Cobros y Pagos, Caja y Banco del proyecto ERP-Cuba.
- De las metodologías analizadas para la construcción del DataMart la que más se ajusta a las características del Cedrux es la metodología UCID.

- Después del estudio de las diferentes herramientas utilizadas para la construcción de un DataMart se seleccionaron por las características del Cedrux: Como sistema gestor de base de datos al PostgreSQL, como herramienta para el proceso de extracción, transformación y carga de los datos a la herramienta Kettle (Pentaho), como herramienta para el diseño del cubo OLAP al WorkBench, el Visual Paradigm como herramienta CASE y como servidor web al Apache Tomcat.

CAPÍTULO 2: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL DATAMART

Introducción

En este capítulo se realiza el análisis, diseño e implementación del DataMart, en correspondencia con los pasos de la metodología escogida en el capítulo anterior para el desarrollo del mismo. Se describe paso a paso cada una de las fases de dicha metodología que incluye desde el análisis de los requerimientos, pasando por la identificación de indicadores y perspectivas, dimensiones, medidas, hasta los procesos de ETL.

2.1 Procedimiento para desarrollo de la solución utilizando la metodología escogida.

Como se hace referencia en el capítulo anterior la metodología escogida para la implementación del DataMart fue la metodología UCID la cual está compuesta por 4 fases. Seguidamente se explicará el procedimiento para el desarrollo de la solución en cada una de estas fases.

2.1.1 Fase 1: Análisis de los requerimientos

Para darle cumplimiento a esta fase se planificaron y realizaron 4 entrevistas con el objetivo de poder obtener los requerimientos para el desarrollo del almacén de datos, a continuación se expone una breve explicación de cada una de ellas.

Entrevista 1

Esta entrevista se realizó con el objetivo de conocer cómo funciona el negocio en los subsistemas Cobro y Pagos, Bancos. La misma se le realizó a los especialistas de dichos subsistemas. Al concluir la misma el equipo de desarrollo obtuvo la información del negocio que le serviría de base para el desarrollo del sistema por lo que se cumplieron los objetivos propuestos.

Entrevista 2

En esta entrevista estuvieron presentes el equipo de desarrollo y los especialistas de los subsistemas a los cuales va dirigido el DataMart. Su objetivo principal consistió en identificarlas perspectivas y los indicadores. En la misma se le realizaron una serie de preguntas a los especialistas en busca de sus necesidades de información, cuáles eran los procesos sobre los cuales les interesaba realizar análisis y que perspectivas e indicadores se utilizarían en los mismos. Los objetivos de la entrevista se cumplieron satisfactoriamente ya que se identificaron los indicadores y perspectivas a utilizar en la confección del modelo conceptual. (Ver **¡Error! No se encuentra el origen de la referencia.**)

Entrevista 3

Esta entrevista se desarrolló con el objetivo de definir el nivel de granularidad de las perspectivas identificadas anteriormente. En la misma participaron el equipo de desarrollo y los especialistas de los

subsistemas Cobro y Pagos, Caja y Banco. Una vez concluida la entrevista el quedó definido el nivel de granularidad para cada una de la perspectivas.

Entrevista 4

La entrevista fue realizada con el objetivo de precisar la forma de cálculo de los indicadores obtenidos en la primera entrevista a los especialistas de los subsistemas Cobro y Pagos, Caja y Banco. Para ello se entrevistó a un grupo de especialistas del Ministerio de Finanzas y Precios de Cuba. Como resultado de la misma se le dio cumplimiento al objetivo propuesto a través de la obtención de las fórmulas para la realización de sus correspondientes cálculos.

2.1.1.1 Construcción del modelo conceptual

Luego de haber identificados los indicadores y perpectivas se paso a confeccionar el modelo conceptual(¡Error! No se encuentra el origen de la referencia.7,¡Error! No se encuentra el origen de la referencia.) donde se representan las perspectivas del lado izquierdo y los indicadores a la derecha, permitiendo comprender cuáles serán los resultados que se obtendrán, cuáles serán las variables que se utilizarán para analizarlos y cuál es la relación que existe entre ellos por lo que se tiene una visión clara del alcance del proyecto.

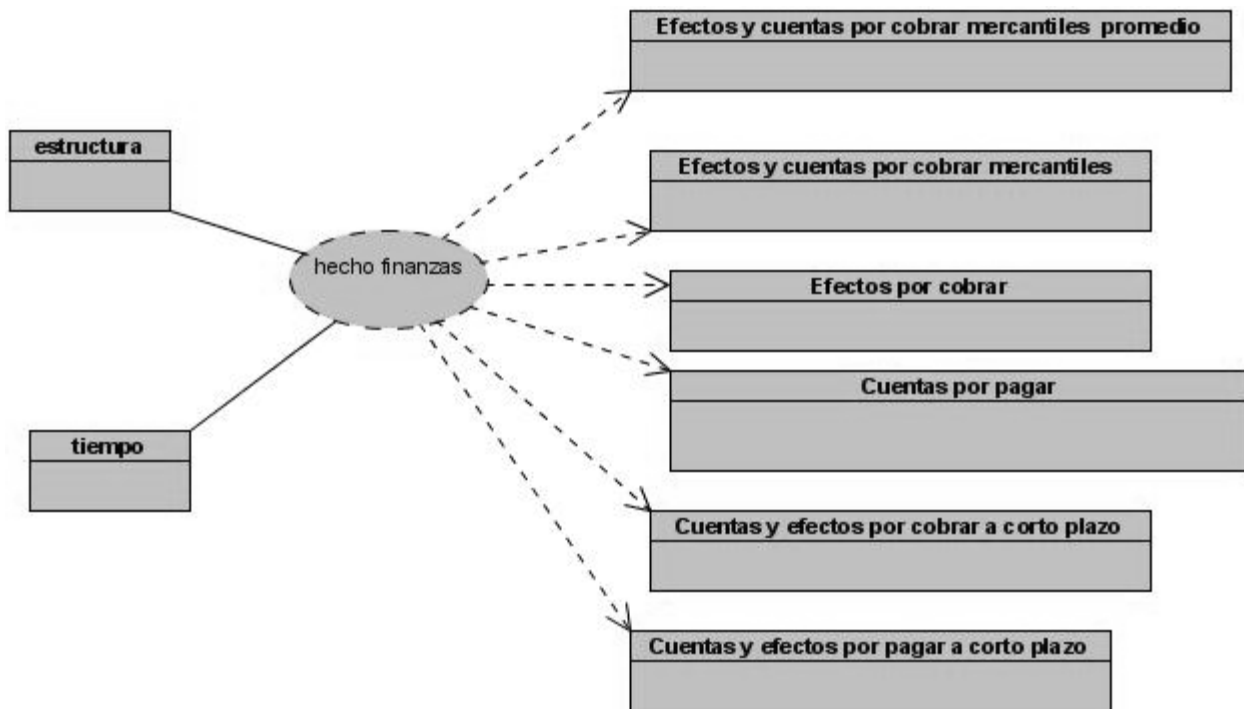


Figura. 7 Modelo Conceptual

En esta figura solo se muestran algunos indicadores del total, para mayor información ver el (¡Error! No se encuentra el origen de la referencia..

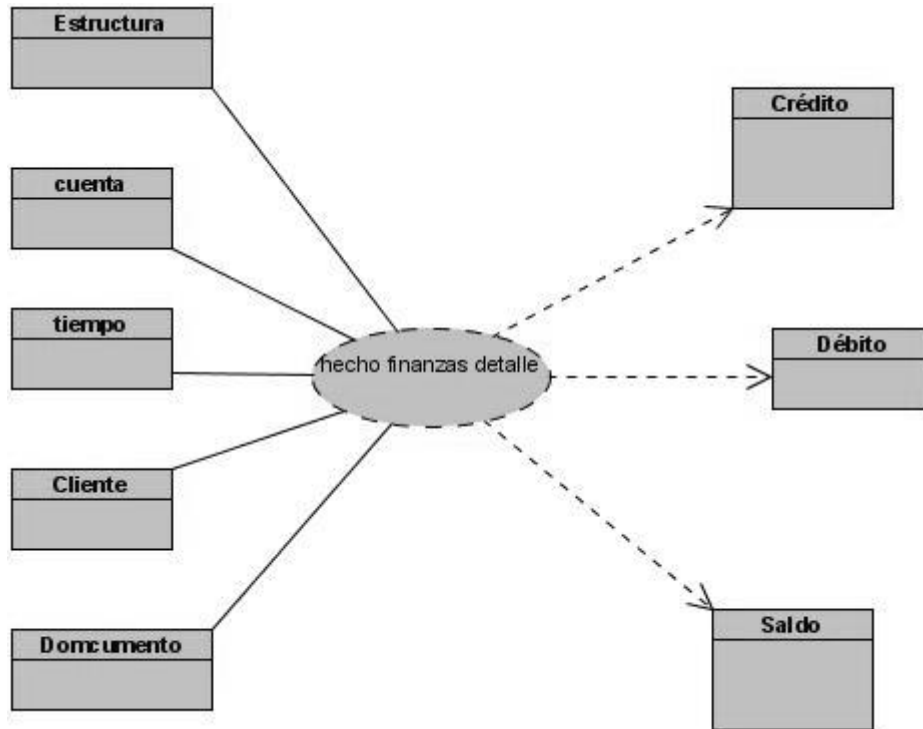


Figura. 8 Modelo Conceptual del cubo finanzas detalle

2.1.2 Fase 2: Análisis de los OLTP

Estado General de los sistemas Fuentes

Los datos de origen se obtendrán de los esquemas de Caja, Cobro y Pagos, Caja, Banco, Finanzas y Contabilidad, pertenecientes a la base datos del ERP. Las especificaciones sobre las características de los sistemas fuentes se especifican en el artefacto “Registro de Sistemas Fuente”.

Determinación de indicadores

1. Ciclo de Cobro (Plazo de Cobros)

Esta razón se expresa en días y significa los días de Venta pendientes de cobro que, como promedio, tuvo la empresa en el periodo analizado, o sea, los días promedio que tardaron los clientes en cancelar sus cuentas con la empresa.

Fórmula:
$$\frac{\text{Efectos y cuentas por cobrar mercantiles promedio}}{\text{Ventas al crédito}} \times \text{días del periodo}$$

2. Ciclo de Pago (Plazo de Pago).

Fórmula:
$$\frac{\text{Efectos y cuentas por pagar mercantiles promedio}}{\text{Compras al crédito}} \times \text{días del periodo}$$

3. Días de Venta pendientes de cobro.

Expresa los días de venta pendientes de cobro que tiene la empresa en este momento.

Mide la relación entre los saldos finales de las cuentas y efectos por cobrar mercantiles con Ventas efectuadas a crédito durante el periodo, refleja la rapidez con que se están cobrando las Ventas.

Fórmula: Efectos y cuentas por cobrar mercantiles x días del periodo

Ventas al crédito

4. Días de Compra pendientes de pago

Fórmula: Efectos y cuentas por pagar mercantiles x días del periodo

Compras al crédito

5. Financiación de la inversión en clientes por proveedores.

Fórmula: Cuentas y efectos por pagar a corto plazo

Cuentas y efectos por cobrar a corto plazo

6. Financiación de inventarios por proveedores.

Fórmula: Cuentas y efectos por pagar a corto plazo

Inventario promedio

Inmovilización de recursos financieros:

7. Ingresos pendientes de cobro

Fórmula: Efectos por Cobrar a Corto Plazo - Efectos por cobrar descontados + Cuentas por cobrar a corto plazo- Provisión para cuentas incobrables+ Ingresos acumulados por cobrar

Total: (1) Cobros pendientes a corto plazo

Fórmula: Ingresos diferidos+ Cobros anticipados Depósitos recibidos

8. Total: (2) Otros cobros

9. Inversiones financieras.

Análisis de los excesos de fuentes de financiamiento.

10. Gastos pendientes de liquidación.

11. Adeudos y obligaciones diversas.

12. Patrimonio.

Patrimonio o capital: Está compuesto por el aporte de los propietarios y las utilidades generadas por la empresa y retenidas por ésta. Por su naturaleza, representa el financiamiento propio, es decir, los derechos que tienen los propietarios sobre los activos de la empresa. No tienen fecha de vencimiento.

Ciclo de Conversión de Caja Global.

13. Ciclo de conversión de caja global.

Fórmula: CCCG = CCC + CCI – CCP

CCCG: Ciclo de Conversión de Caja Global

CCC: Ciclo de Conversión de Cuentas por Cobrar a corto plazo.

CCI: Ciclo de Conversión de Inventarios.

CCP: Ciclo de Conversión de Cuentas por Pagar

14. Ciclo de Conversión de Cuentas por Cobrar a corto plazo

Fórmula: $CCC = 360 / (VN/CC)$

VN: Ventas netas

CC: Cuentas por cobrar

15. Ciclo de Conversión de Inventarios

Fórmula: CCI: Ciclo de Conversión de Inventarios

$CCI = 360 / (CV/IV)$

CV: Costo de ventas

IV: Inventarios

16. Ciclo de Conversión de Cuentas por Pagar

Fórmula: CCP: Ciclo de Conversión de Cuentas por Pagar

$CCP = 360 / [(CV + GV) / (CP + S)]$

CV: Costo de venta

GV: Gastos de distribución y venta

CP: Cuentas por pagar

S: Nóminas por pagar

Análisis de las cuentas por cobrar por antigüedad

17. Edades por cuenta, clientes, proveedores y total en cada periodo y ejercicio (en un tiempo determinado).

18. Por ciento: representa el valor por edades con respecto al total, de las cuentas por cobrar, pagar, pagos y cobros anticipados a nivel de cliente, y total.

2.1.2.1 Correspondencia con los requerimientos

El objetivo de este paso, es el de examinar los OLTP disponibles que contengan la información requerida, para poder identificar las correspondencias entre el modelo conceptual y las fuentes de datos.

La idea es, que todos los elementos del modelo conceptual (perspectivas e indicadores) estén correspondidos en los OLTP, más específicamente con las tablas o campos de las tablas desde donde se vayan a extraer la información que se necesitará posteriormente para completar las perspectivas e indicadores. Se analizaron las correspondencias de manera que no quedaron fuentes de origen sin identificar.

Campos que integran las perspectivas. Nivel de granularidad

Estructura	Cuentas	Tiempo	Antigüedad	Cliente	Documento
Estructura	Grupo	Quinquenio	No vencidas	Cliente	Documento
	Contenido	Año	Vencida hasta 30 días		
	Cuenta	Semestre	Vencida hasta 60 días		
	Subcuenta	Trimestre	Vencida hasta 90 días		
		Mes	Vencida más de 90 días		

Tabla 2 Nivel de granularidad de las perspectivas

Campos de las perspectivas:

Estructura

País

Provincia

Municipio

Unidad de base

Cuentas

Grupo

Subgrupo

Contenido

Cuenta

Tiempo

Quinquenio

Año

Semestre

Trimestre

Mes

Antigüedad

Vencida 30 días

Vencida 60 días

Vencida 90 días

Más de 90 días

Clientes

Código

Nombre

Documento

Fecha de emisión

Denominación

Modelo Conceptual ampliado

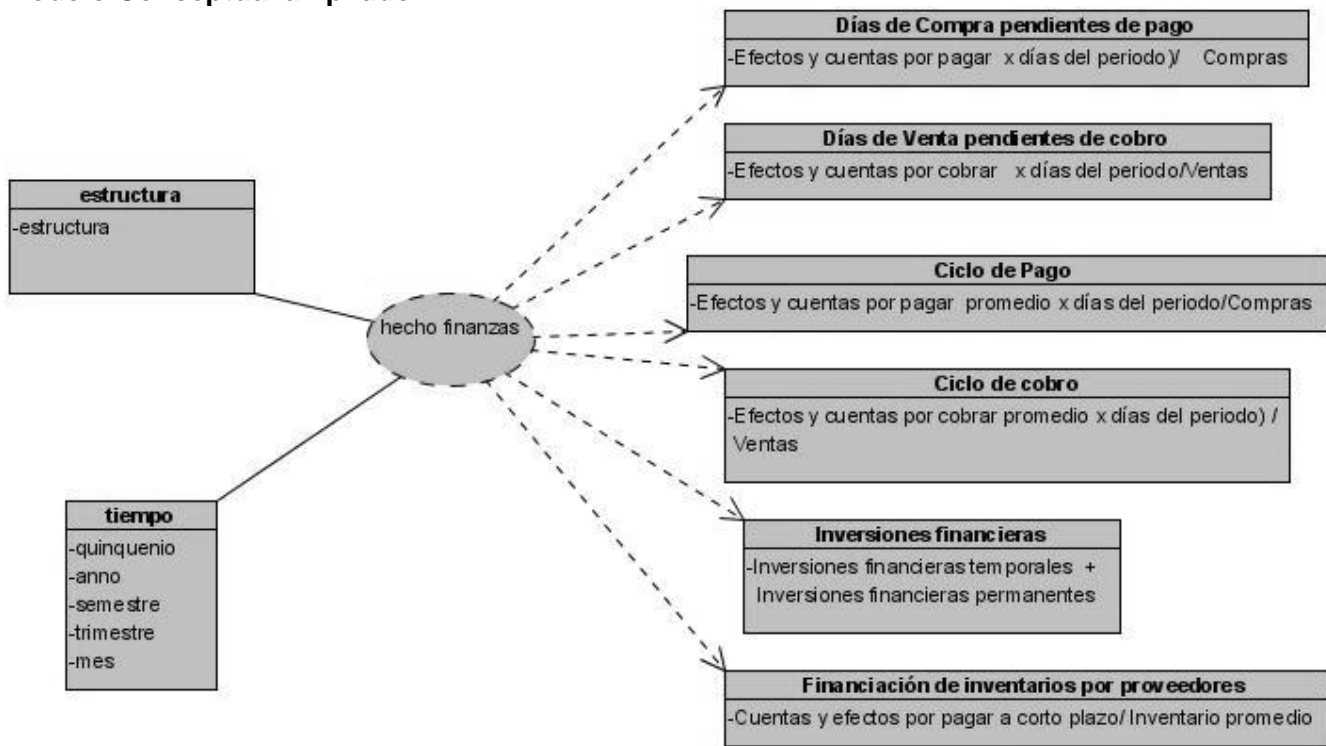


Figura. 9 Modelo Conceptual Ampliado del cubo Finanzas

En las figura 9 mostradas anteriormente no aparecen todos los indicadores para ver el modelo conceptual ampliado de Finanzas consultar el (¡Error! No se encuentra el origen de la referencia.

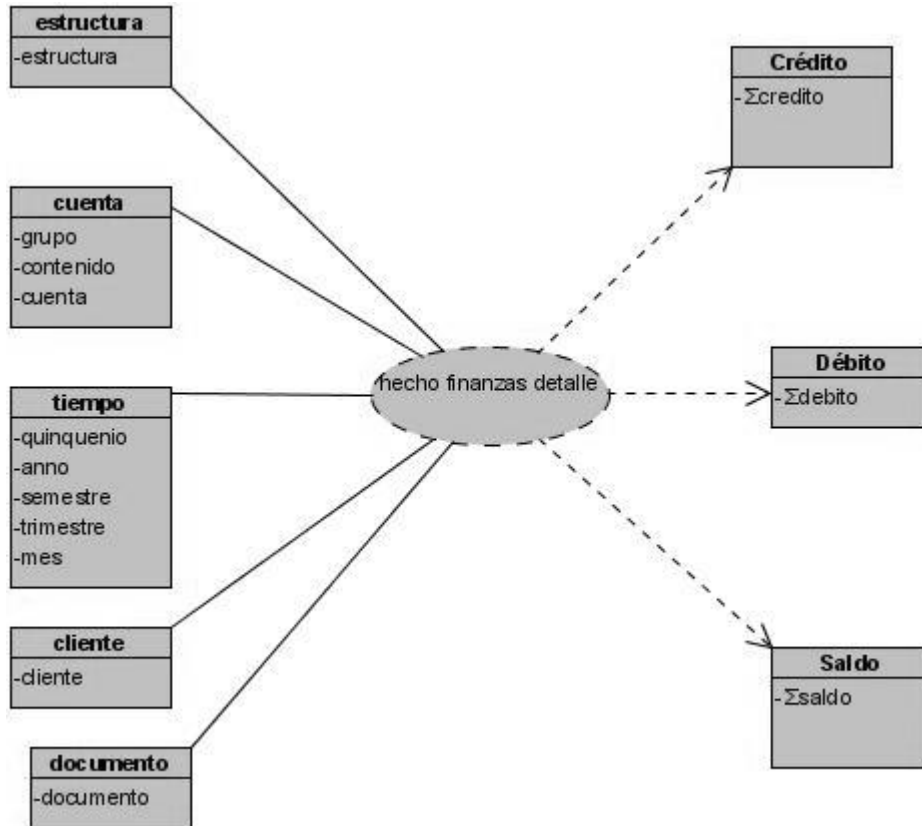


Figura. 10 Modelo Conceptual Ampliado del cubo Finanzas Detalle

2.1.3 Fase3: Modelo lógico del DW

Definir Tipo de Modelo Lógico del DW

Durante la construcción de modelo conceptual el equipo de desarrollo identificaron dos procesos fundamentales, uno para los casos donde se podían hacer análisis hasta el último nivel de detalle y otro donde estos sólo se podían en el nivel más alto, por lo que no era posible representarlos en una sola tabla de hechos de manera que fue necesario elaborar dos tablas de hechos independientes, una para cada procesos, las cuales iban a compartir dimensiones comunes ya que tanto la estructura como el tiempo se utilizan en los dos procesos. Teniendo en cuenta estas razones y por las ventajas que trae consigo su utilización, se decidió emplear como tipología de esquema **constelación de hechos** pues se tienen dos tablas de hechos que tienen relación con algunas tablas de dimensiones.

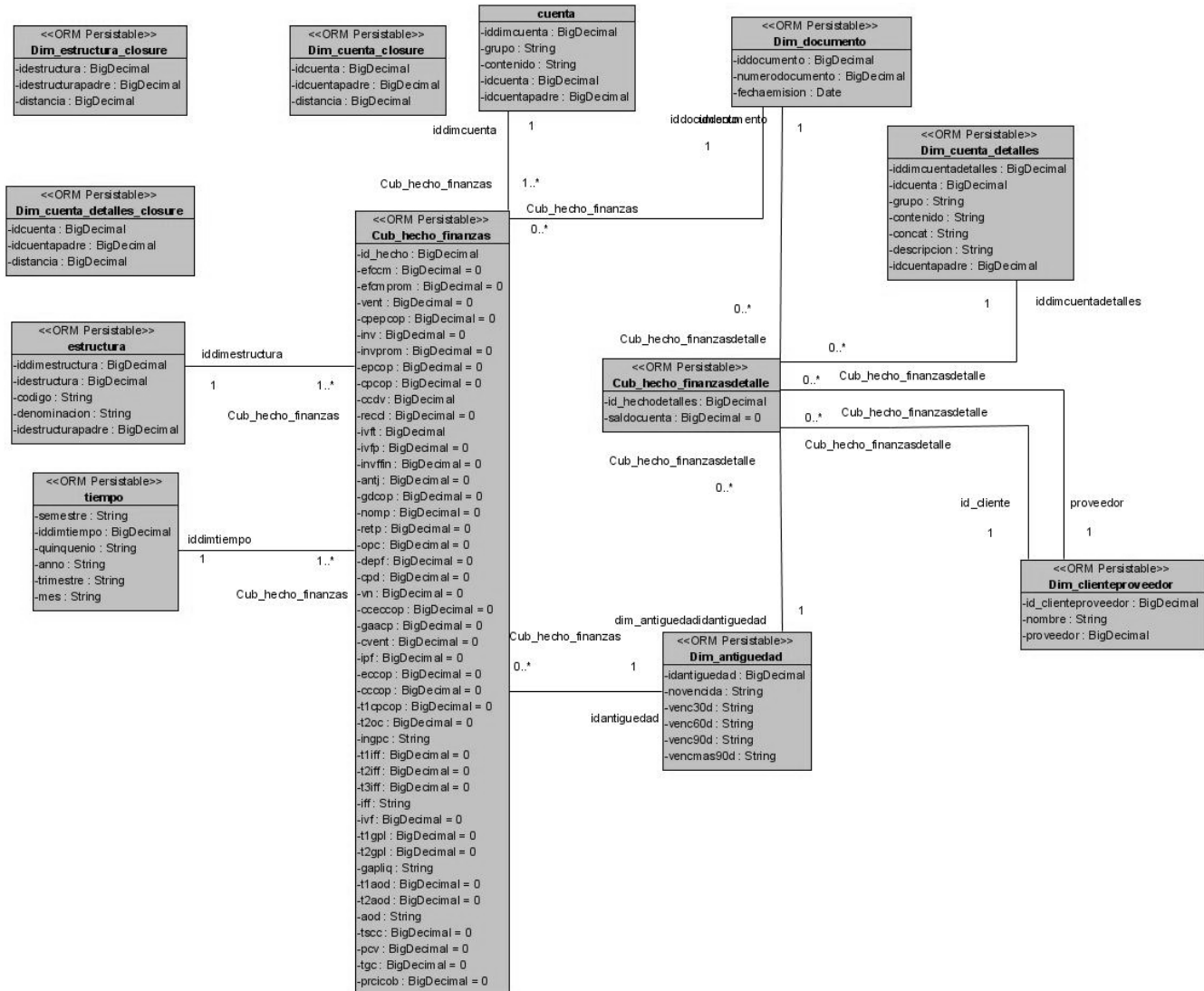


Figura. 11 Modelo Lógico

Definición de estándares para objetos físicos

Para nomenclatura de los objetos físicos se definió que los nombres de las dimensiones comenzarán con el prefijo “dim_” seguido por el nombre de la perspectiva que lo generó. Ejemplo: para el caso de la dimensión Cliente el nombre sería “dim_cliente”. Todos los nombres de las tablas de hechos deben comenzar con el prefijo “cub_hecho”.

Modelo Físico del DataMart

En este paso quedarían diseñadas e implementadas físicamente las tablas de hechos y dimensiones en el DataMart según el modelo lógico.

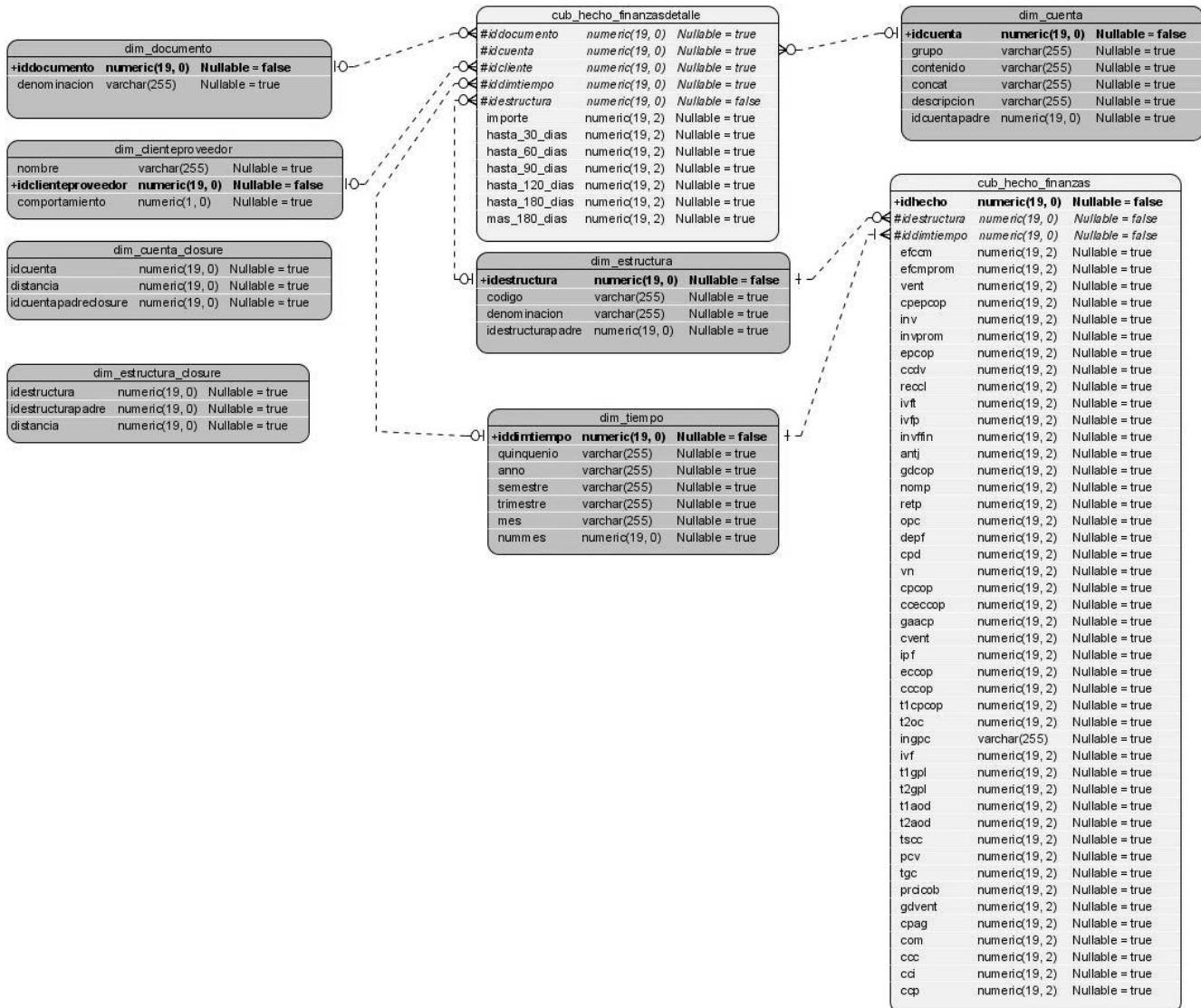


Figura. 12 Modelo físico

2.1.4 Fase 4: Proceso ETL

Una vez creado el almacén de datos es necesario realizar el proceso de Extracción, Transformación y Carga (ETL), este se realizará en herramienta Kettle perteneciente a suite Pentaho, explicada anteriormente.

Para desarrollar este proceso primeramente se realizará un mapeo de los datos fuentes hacia los destinos detallando claramente de donde salen y hacia donde se dirigen, luego se establecerán algunas restricciones y condiciones adicionales en caso que lo requiera. Una vez definidas estas transformaciones se procede a la carga incremental de los datos hacia las dimensiones y hechos según corresponda. Finalmente se diseñará y construirá la automatización de todo el proceso.

Mapeo de datos Fuente-A-Destino

Se estableció un mapeo de los datos desde la fuente hacia el destino de los mismos con el objetivo de Para evitar pérdidas de datos en el proceso ETL se realizó el mapeo de los datos desde las fuentes de origen hasta el destino, este proceso quedo documentado en la planilla “Mapa Lógico de Datos”.

Seguidamente se muestra un fragmento de este proceso.

Mapa Lógico de Datos					
Nombre BD Fuente: Cedrux			Nombre BD Destino: Almacen		
Nombre de la tabla	Nombre del campo	Tipo de dato	Nombre de la tabla	Nombre del campo	Tipo de dato
res_resumen_paseconsolidacion	idestructuracomun	NUMERIC(19,0)	dim_estructura	idestructura	Numeric(19,0)
dat_estructura	codigo	Varchar(20)	dim_estructura	codigo	Varchar(255)
dat_estructura	denominacion	Varchar(255)	dim_estructura	denominacion	Varchar(255)
res_resumen_paseconsolidacion	idestructurae	NUMERIC(19,0)	dim_estructura	idestructurapadre	Numeric(19,0)

Tabla 3 Mapa Lógico de Datos

Es importante destacar que para el caso de la tabla dim_tiempo no se establece mapeo ya que en esta se insertan los datos a partir de la ejecución de la función f_registrar_anos en la que al entrarle como parámetros una fecha inicio y fin esta genera todos los datos definidos para la dimensión, así mismo la generación de los iddimtiempo se realizan a través de la secuencia sec_dimtiempo_seq, ambas pertenecientes al esquema mod_comun.

Establecer condiciones adicionales y restricciones

En el momento de generar los ETL, se establecieron una serie de condiciones y restricciones con el objetivo de que la información a almacenar en el depósito de datos fuera seleccionada de forma

correcta. Estas condiciones fueron llevadas a cabo con mucha prudencia para evitar las pérdidas de datos importantes. Entre las condiciones y restricciones aplicadas se encuentran:

1. Para el llenado del cub_hecho_finanzasdetalle se hace necesario que las medidas sean calculadas por estructura, cuenta, cliente, documento y tiempo.
2. Para el llenado del cub_hecho_finanzas se hace necesario que las medidas sean calculadas por estructura y tiempo.
3. El saldo para las cuentas solo se almacenan en las cuentas hojas por lo que para obtener el saldo de la cuenta el primer nivel se hace necesario sumar todo el saldo de las cuentas hojas.
4. Para buscar el iddim tiempo en la tabla es necesario buscar los valores del mes y el año siendo necesario particionar el campo fecha que se guarda en la base de datos de origen.

Cargas incrementales de los datos

Para implementar el proceso de ETL la herramienta Kettle posee dos componentes principales:

Las Transformaciones y los Trabajos (jobs).

- Transformación: Este componente permite realizar todo el proceso de extracción, transformación y carga.
- Trabajo: Es el componente encargado de la automatización del proceso ETL mediante planificación y ejecución las Transformaciones de forma coordinada.

Para realizar la carga de los datos en la dimensión Estructura se utilizó el componente Entra de Tabla el cual permite conectarse a una tabla en una base de datos de origen y seleccionar los campos que se quieren almacenar en el DataMart a través de una consulta SQL. (**¡Error! No se encuentra el origen de la referencia.**)

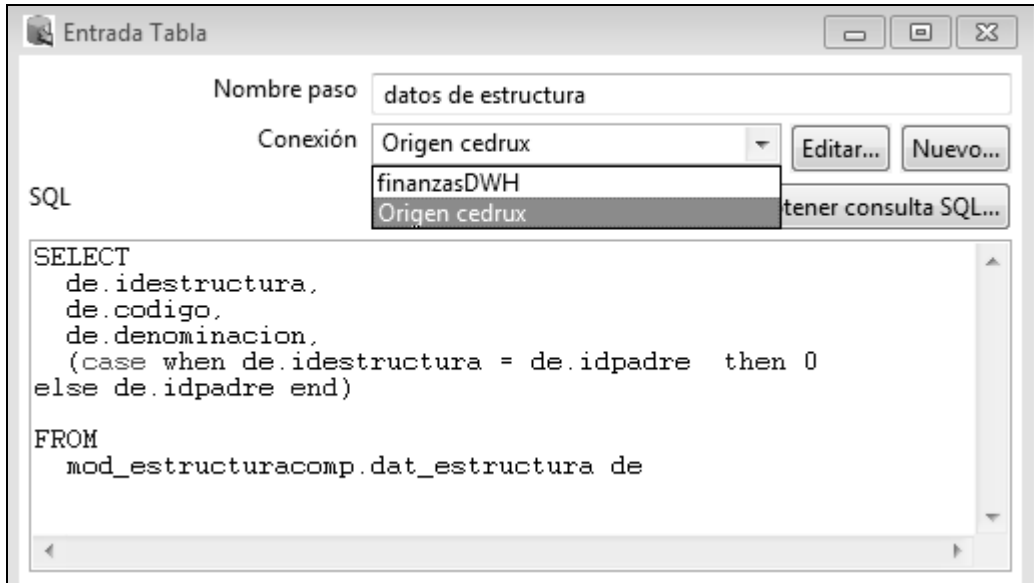


Figura. 13 Selección de la base datos de origen y consulta para seleccionar los campos.

Luego se llena la dimensión estructura utilizando el componente (Insertar/Actualizar), el cual permite insertar o actualizar en una tabla, así como filtrar los datos que se quiere insertar.

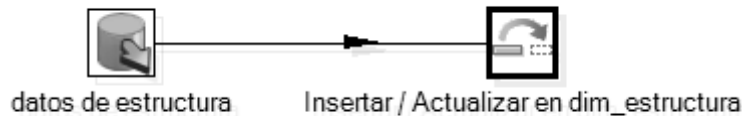


Figura. 14 Llenado de la dimensión estructura.

Este mismo procedimiento se realiza para el resto de las dimensiones (Cliente, Documento y Cuenta).

Llenado de la tabla de hechos:

Una vez llenas todas las tablas de las dimensiones se procedió a llenar las tablas de hechos.

Para llenar la tabla de hechos cub_hecho_finanzas se utilizó el componente (Entrada de Tabla) explicado anteriormente. En el cual se ejecutó una consulta a las base de datos de origen para obtener los información necesaria a almacenar en la tabla de hechos como son la estructura, el mes, el año y el cálculo de los saldos de los rangos de cuentas para cada estructura en un periodo determinado de tiempo.

Para ello fue necesario realizar transformaciones a los datos de forma que tuvieran el formato correcto para ser almacenados.

En este caso tenemos a que fue necesario extraer de la fecha de inicio del periodo el mes y el año.

Luego concluido este paso se realizó una búsqueda en las tablas de las dimensiones a través del componente (Búsqueda en base de datos) con el objetivo de encontrar las correspondencias entre los datos obtenidos en paso anterior (estructura, mes, año) en las dimensiones Estructura y Tiempo. Después de haber eliminado los campos del flujo de datos que no eran necesarios para el llenado de la tabla de hechos utilizando el componente (Selecciona/Renombrar valores), se procedió a la inserción de los datos en la misma utilizando el componente (Insertar/Actualizar) cuyo funcionamiento se explicó anteriormente. (**¡Error! No se encuentra el origen de la referencia.**)

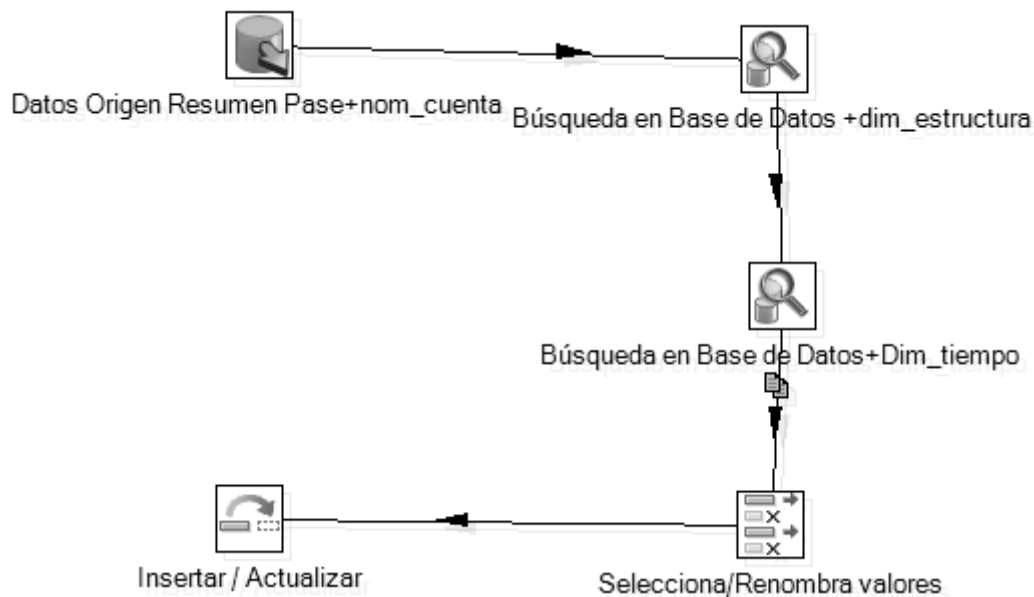


Figura. 15 Llenado de la tabla de hechos cub_hecho_finanzas.

Para llenar la tabla de hechos cub_hecho_finanzasdetalle se siguió un procedimiento similar.

Automatización del proceso ETL:

La configuración de la automatización del proceso ETL se realizó de la siguiente forma:

Se crearon dos Trabajos cada uno para llenar la tabla de hechos cub_hecho_finanza y sus dimensiones asociadas y otro para el llenado de la tabla de hechos cub_hecho_finanzadetalles y sus dimensiones asociadas (**¡Error! No se encuentra el origen de la referencia.**). Los trabajos se ejecutarán cada 30 días en correspondencia con la actualización de las tablas de origen.

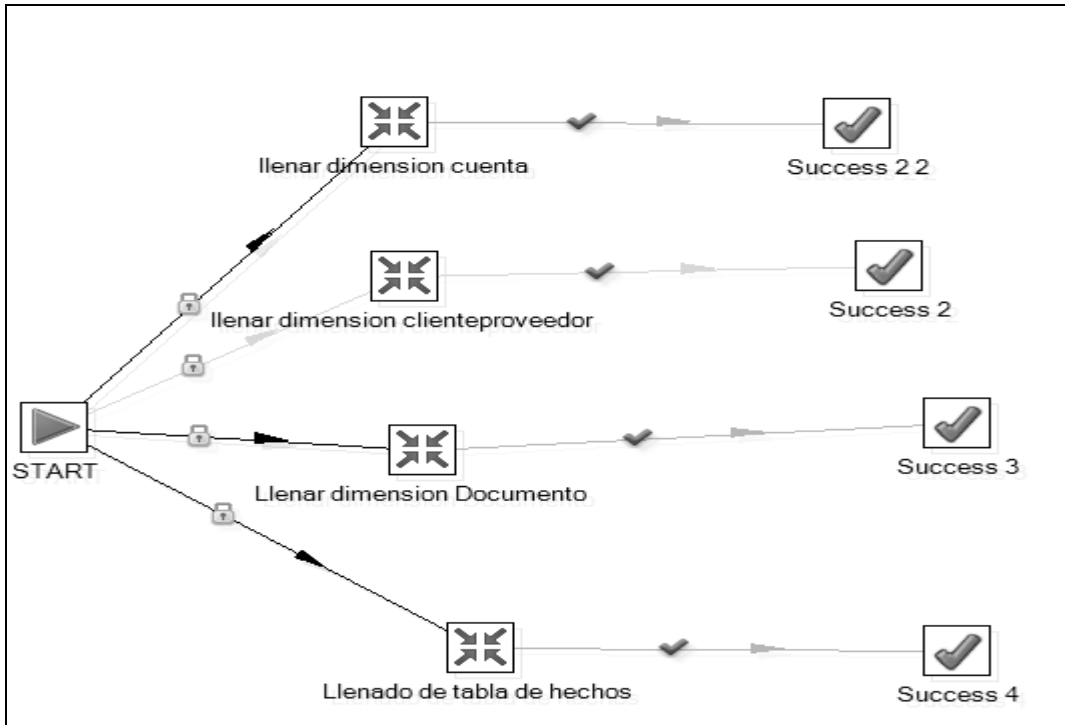


Figura. 16 Trabajo para llenar las dimensiones y la tabla de hecho cub_hecho_finanzasdetalle.

Conclusiones del capítulo

En el capítulo se desarrollaron cada uno de los pasos de la metodología UCID. Se obtuvieron aspectos claves del negocio, identificándose los indicadores y perspectivas del análisis. Se elaboró el modelo conceptual de datos proporcionando una idea clara del alcance del almacén. Además, se confeccionó el modelo lógico de la estructura del DataMart para luego realizar el proceso de extracción, transformación y carga de los datos en el almacén.

CAPÍTULO 3: VALIDACIÓN Y DESPLIEGUE

3.1 Introducción

En este capítulo se abordan los temas relacionados con la validación del DataMart través de dos variantes principales, las pruebas de concepto a la base de datos y las pruebas funcionales realizadas por los expertos. También se explica el funcionamiento de la estrategia de consolidación de la información y de despliegue del almacén de datos.

3.2 Validación

La validación del sistema está concebida sobre dos aspectos fundamentales, el cálculo y representación correcta de los indicadores, y la realización de dos tipos de pruebas, las de concepto que se realizan al diseño de la base de datos y las funcionales llevadas a cabo por los expertos funcionales. La primera consiste en que los indicadores estén representados correctamente de manera que se realicen los cálculos sin imprecisiones, para realizarlos se diseñan los cubos OLAP en dependencia de la cantidad de tablas de hechos que se emplearán en el sistema, las cuales van a depender de los diferentes análisis que se van a realizar.

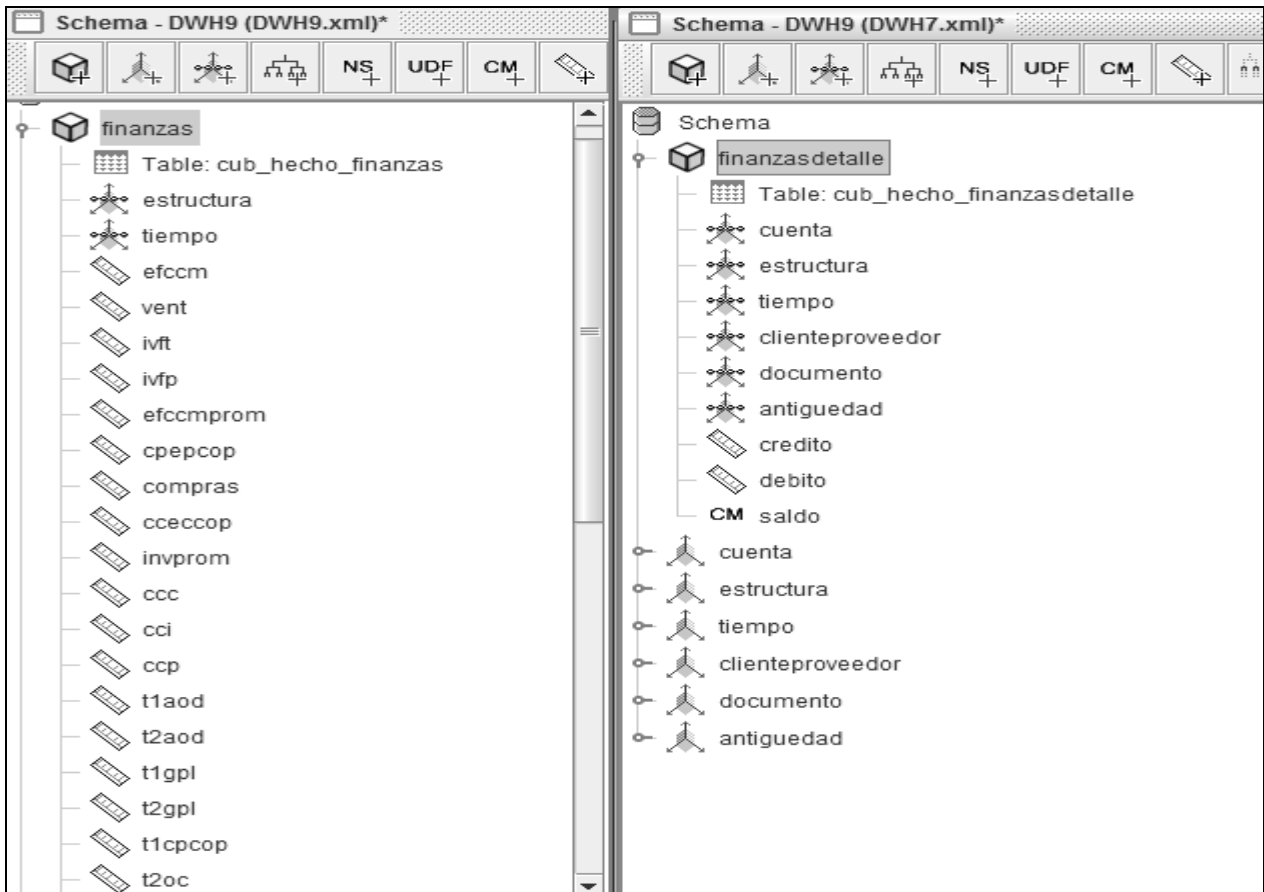


Figura. 17 Cubos multidimensionales con sus medidas y dimensiones asociadas.

En este caso el DataMart consta de dos cubos nombrados finanzas y finanzasdetalle. El primero debe posibilitar realizar el análisis de los indicadores financieros y mostrar la interrelación que se establece entre ellos para las dimensiones ya definidas. Está basado en la tabla de hechos cub_hecho_finanzas y está compuesto por 2 dimensiones, la dimensión estructura y la dimensión tiempo. Consta de un total de 31 medidas vinculadas a las medidas naturales definidas en el capítulo 2, necesarias para su utilización mediante operaciones matemáticas en la obtención de las razones financieras.

El cubo finanzasdetalle por otro lado se destinó a la realización de análisis más detallados a partir de las interrogantes generadas en el análisis de los indicadores, posibilitando conocer a nivel de saldos de cuentas, el comportamiento de cada uno de los indicadores previamente definidos, especificando a nivel de clientes y documentos el estado de salud de cada entidad.

Está basado en la tabla de hechos cub_hecho_finanzasdetalle y utiliza dimensiones tales como, estructura, tiempo, cuenta, cliente, documento y antigüedad, conteniendo también medidas naturales específicas correspondientes a los indicadores vinculados a clientes y documentos.

Durante la construcción de los cubos se tuvo en cuenta la jerarquía de las dimensiones de forma que estas tuvieran un orden lógico y permitieran realizar operaciones de un nivel general a uno específico o viceversa. Se realizó la comprobación de los cálculos de las medidas y el correcto funcionamiento de los reportes que serán objeto de análisis mediante la ejecución de consultas MDX utilizando la herramienta de representación y diseño de cubos: workbench.

A continuación se muestran algunos ejemplos de las consultas realizadas para la obtención de los reportes solicitados por los clientes:

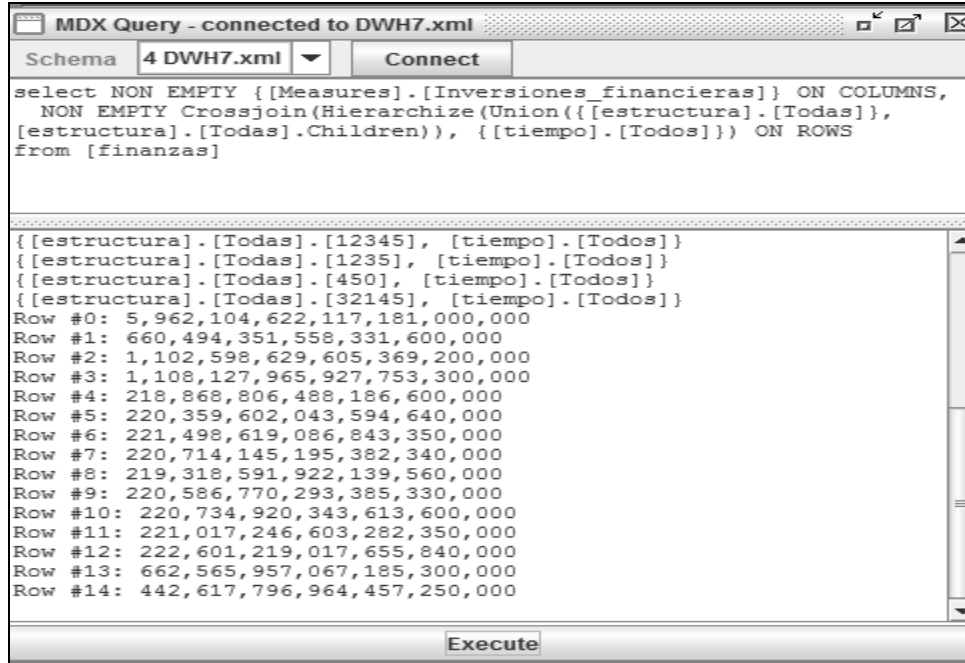


Figura. 18 Reporte Inversiones Financieras.

En este reporte nos muestra el comportamiento de las inversiones financieras de todas las estructuras en el tiempo.

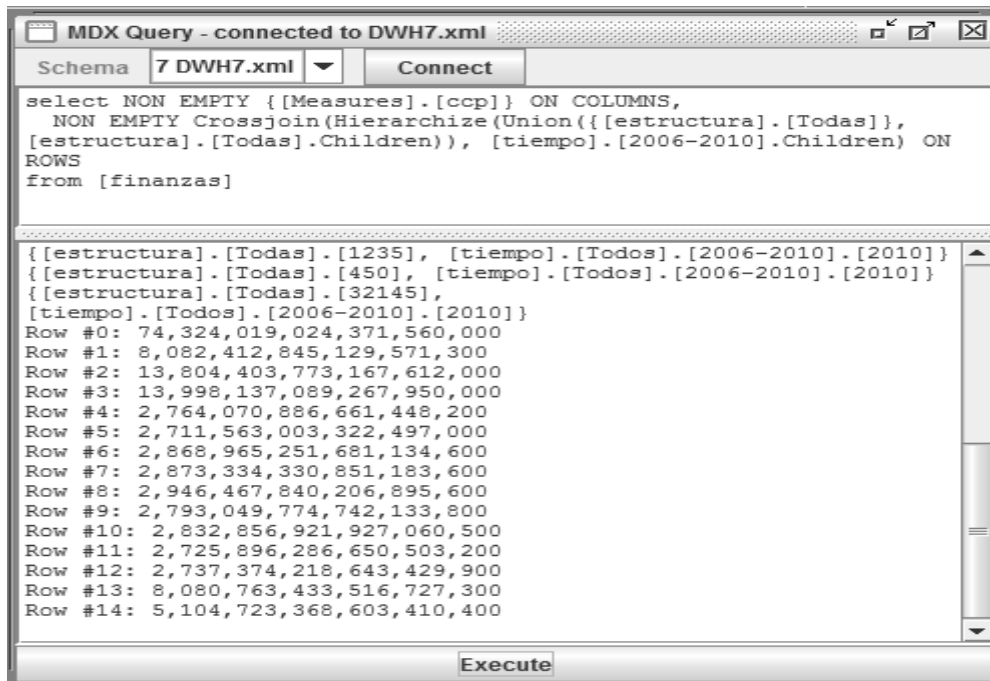


Figura. 19 Reporte Inversiones Financieras.

Como se muestra en la figura anterior mediante esta consulta se obtiene el reporte de inversiones financieras.

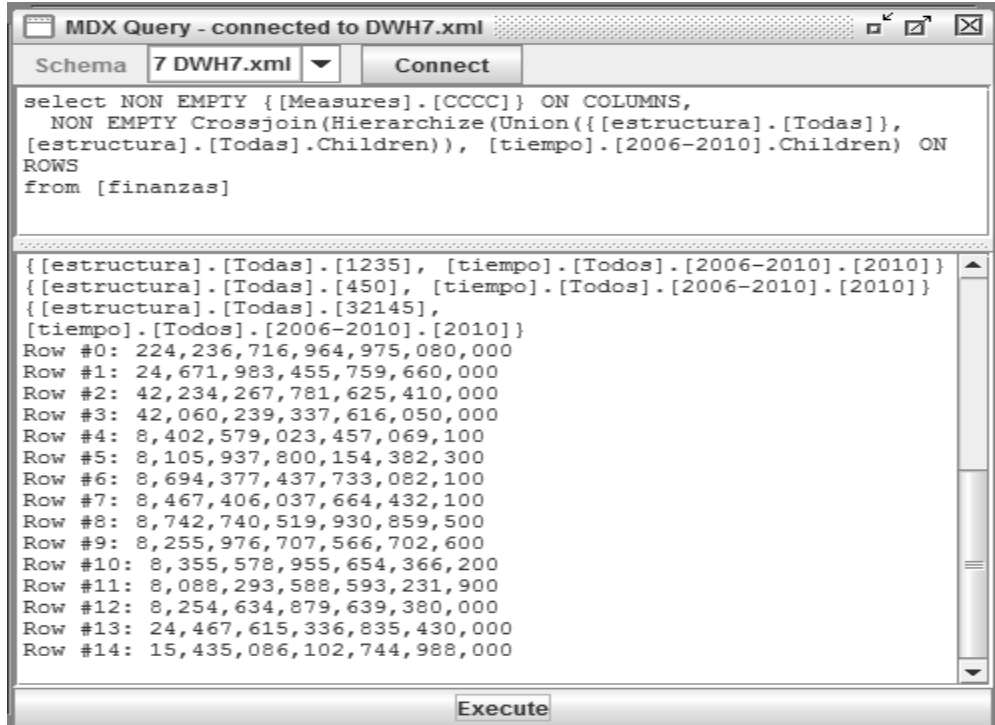


Figura. 20 Ciclo de Caja Global

A continuación se presentan algunos ejemplos de cómo quedarían los reportes en la herramienta Mondrian.

The screenshot shows the Mondrian reporting tool interface. On the left, there is a 'Navegar' (Navigate) pane with a tree view containing folders like 'Almacenes', 'Arbol de Prueba', 'Finanzas', 'Reportes', 'Sala Situacional UCI para toma de dec', and 'aki'. Below it is an 'Archivos' (Files) pane showing files like 'Reporte Inversiones Financieras por estructura', 'aod', 'cual', and 'cuenta saldo por documento grafico pastel'. The main area displays a data table titled 'Medidas' with the following data:

estructura	tiempo	Medidas
Inversiones financieras		
- Todas	+ Todos	5.962.104.622.117.181.000.000
+ 123	+ Todos	660.494.351.558.331.600.000
+ 234	+ Todos	1.102.598.629.605.369.200.000
+ 129	+ Todos	1.108.127.965.927.753.300.000
1	+ Todos	218.868.806.488.186.600.000
2	+ Todos	220.359.602.043.594.640.000
132	+ Todos	221.498.619.086.843.350.000
54456	+ Todos	220.714.145.195.382.340.000
101	+ Todos	219.318.591.922.139.560.000
575	+ Todos	220.586.770.293.385.330.000
22222	+ Todos	220.734.920.343.613.600.000
12345	+ Todos	221.017.246.603.282.350.000
1235	+ Todos	222.601.219.017.655.840.000
+ 450	+ Todos	662.565.957.067.185.300.000

Figura. 21 Reporte Inversiones Financieras

estructura	tiempo	Medidas
↑ Todas	↑ 2010	• Ciclo de Conversión de Cuentas por Pagar(CCP) 74.324.019.024.371.560.000
↓ 123	↓ 2010	8.082.412.845.129.571.300
↓ 234	↓ 2010	13.804.403.773.167.612.000
↓ 129	↓ 2010	13.998.137.089.267.950.000
1	↓ 2010	2.764.070.886.661.448.200
2	↓ 2010	2.711.563.003.322.497.000
132	↓ 2010	2.868.965.251.681.134.600
54456	↓ 2010	2.873.334.330.851.183.600
101	↓ 2010	2.946.467.840.206.895.600
575	↓ 2010	2.793.049.774.742.133.800
22222	↓ 2010	2.832.856.921.927.060.500
12345	↓ 2010	2.725.896.286.650.503.200
1235	↓ 2010	2.737.374.218.643.429.900
↓ 450	↓ 2010	8.080.763.433.516.727.300
↓ 32145	↓ 2010	5.104.723.368.603.410.400

Figura. 22 Reporte Ciclo de Conversión de Cuentas por Pagar (CCP)

estructura	tiempo	Medidas
↑ Todas	↑ 2010	• Ciclo de conversión de caja global 224.236.716.964.975.080.000
↓ 123	↓ 2010	24.671.983.455.759.660.000
↓ 234	↓ 2010	42.234.267.781.625.410.000
↓ 129	↓ 2010	42.060.239.337.616.050.000
1	↓ 2010	8.402.579.023.457.069.100
2	↓ 2010	8.105.937.800.154.382.300
132	↓ 2010	8.694.377.437.733.082.100
54456	↓ 2010	8.467.406.037.664.432.100
101	↓ 2010	8.742.740.519.930.859.500
575	↓ 2010	8.255.976.707.566.702.600
22222	↓ 2010	8.355.578.955.654.366.200
12345	↓ 2010	8.088.293.588.593.231.900
1235	↓ 2010	8.254.634.879.639.380.000
↓ 450	↓ 2010	24.467.615.336.835.430.000
↓ 32145	↓ 2010	15.435.086.102.744.988.000

Figura. 22 Ciclo de Caja Global

3.2.1 Validación del diseño mediante pruebas de rendimiento a la aplicación

Una de las estrategias utilizadas para la evaluación del diseño de la base de datos del DataMart fueron las pruebas de rendimiento. Las cuales dentro de sus objetivos principales tienen la verificar el comportamiento de las aplicaciones bajo ciertas condiciones. Estas fueron realizadas mediante el uso de la herramienta Jmeter, bajo las siguientes situaciones y escenarios.

- El servidor de aplicaciones estaba corriendo.
- El servidor de aplicaciones era de 1GB de RAM con 160 GB de disco duro.
- La velocidad de la red era de 100 Mbps.
- La tabla principal se encontraba poblada con más de 1000 tuplas.

Se realizaron las pruebas arrojando los siguientes resultados. Ver la **(¡Error! No se encuentra el origen de la referencia..)**

ID del escenario	Escenario de la sección	Carga de trabajo	Descripción	Resultado de la prueba
EC 1: Realizar la carga inicial de todos los indicadores financieros.	EC 1.1: Realizar la carga inicial de los indicadores financieros con sus dimensiones correspondientes para una concurrencia de 20 usuarios.	20	En este escenario se realiza la carga inicial de todos los indicadores financieros.	64.9 seg
	EC 1.2: Realizar la carga inicial de los indicadores financieros con sus dimensiones correspondientes para una concurrencia de 50 usuarios.	50	En este escenario se realiza la carga inicial de todos los indicadores financieros.	64.3 seg
	EC 1.3: Realizar la carga inicial de los indicadores financieros con sus dimensiones correspondientes para una concurrencia de 100 usuarios.	100	En este escenario se realiza la carga inicial de todos los indicadores financieros.	54.7 seg

Tabla 4: Pruebas de rendimiento a la aplicación

Luego de analizar los resultados correspondientes a las pruebas realizadas se llegó a la conclusión que los tiempos de respuestas para las mismas fueron satisfactorios. Esta valoración está fundamentada por la complejidad de la consulta realizada para este caso en particular. Es de gran

importancia aclarar que los tiempos obtenidos se determinan desde que el usuario realiza la acción de mostrar el reporte, incluyendo el tiempo para que la aplicación transforme la consulta en lenguaje MDX a lenguaje plpgsql, busque la información en la base de datos para luego pintarla mediante el uso del componente jpivot y que finalmente pueda ser visualizada por el usuario, siendo este último proceso el que más tiempo requiere.

3.2.2 Validación funcional

El sistema está actualmente sometido a un conjunto de pruebas en el ministerio de finanzas y precios para su evaluación por un grupo de expertos en la materia, los cuales probarán y chequearán si cumple con las especificaciones y normas establecidas en las resoluciones emitidas por ese ministerio, validando si el sistema responde a los requisitos definidos inicialmente.

Luego de la evaluación se concluyó que el sistema cumple con las normas establecidas en las resoluciones y que es un sistema amigable que contribuirá a facilitar el trabajo de los compañeros que lo utilizarán. El resultado de esta evaluación está recogido en el documento Acta de Validación del Ministerio de Finanzas.

3.3 Mecanismo de consolidación de la información

En esta investigación el sistema origen está constituido por varios esquemas donde se almacena la información perteneciente a las transacciones que se realizan con el Cedrux. Los módulos de Cobro y Pagos, Caja y Banco presentan la información en 4 esquemas fundamentales (mod_caja, mod_banco, mod_cobroypago y mod_comunfinanzas) en este último se guarda la información común de estos subsistemas pero no la totalidad necesaria para el negocio por lo que se busca información en otros esquemas como datos_maestros para obtener la información relacionada con los documentos y clientes y con mod_contabilidad de donde se obtienen los datos sobre la cuentas. Para llevar a cabo la estrategia de consolidación se creó un esquema determinado mod_consolidacion cuyo objetivo es almacenar toda la información de los esquemas relacionados con la consolidación de la información. En el subsistema toda la información es gestionada por entidades, cuando una entidad realiza el cierre de un periodo contable automáticamente la información es guardada en las tablas del esquema consolidación. Existen entidades que llevan una doble contabilización, además de controlar su gestión financiera y contable poseen a información de entidades inferiores a ellas, o sea, el ministerio de finanzas y precios consolida los datos de todas sus entidades a nivel provincial.

Para esta versión del producto hay dos aspectos que se deben tener en cuenta, el primero es que esta consolidación se va a realizar en modo offline a través de un componente que se encarga de

importar y exportar comprobantes que son entendidos por el sistema, el otro aspecto consiste en que de las tablas relacionadas con los subsistemas Cobro y Pagos, Caja y Banco solo van a estar representadas en ese esquema de consolidación `res_centrocuenta_elemento_periodoconsolidacion` y `res_resumen_paseconsolidacion`, lo que implica que el proceso de consolidación pueda ser realizado parcialmente.

Cada entidad tendrá además de su fuente de datos origen una base de datos donde radicarán todos los datos referentes al almacén posibilitando que las entidades consolidadoras posean la información de ellas y las consiguiente a las entidades a las cuales van a realizarle el proceso de consolidación.

Una de las ventajas de la estrategia de consolidación es que el algoritmo para consolidar se basa en los códigos de las cuentas y no en los identificadores, esto significa que si una entidad consolidadora posee una cuenta con un id determinado y sus entidades inferiores (hijas) poseen las mismas cuentas pero con identificadores diferentes al poseer el mismo código de cuenta la consolidación se realiza a partir de la suma de los saldos de las cuentas con el mismo código abstrayendo el proceso de los cambios que puedan existir en los identificadores puesto que estos vienen determinados por el número del servidor donde va a radicar la base de datos. Agregar además que el clasificador de cuentas está establecido a nivel de país, sin muchas variaciones, y en el caso de modificar un código se realizaría para todo el país.

3.4 Despliegue del DataMart

Para realizar el despliegue del DataMart se necesitarán un total de 3 servidores:

- Un servidor de aplicaciones donde se ejecuta el Pentaho Mondrian sobre el Apache Tomcat con las siguientes características.
 - Ubuntu Server, Apache 2.0, PHP 2.5.6
 - Procesador 3.00 GHz, RAM 1GB, HDD 160 GB
- Dos servidores de base de datos Servidor de base de datos: uno para la base de datos del Cedrux y otro para el DataMart ambos cumpliendo con las siguientes características:
 - Ubuntu Server, PostgreSQL 8.3.8 o superior
 - Procesador 3.00 GHz, RAM 1GB, HDD 160 GB

Se realiza de esta forma ya que se recomienda que los servidores de origen y el almacén de datos estén separados para una mayor seguridad de los mismos y mayor protección con respecto a fallas.

Los servidores se comunicarán mediante el protocolo TCP/IP y la PC cliente se conectará mediante el protocolo HTTP para realizar las consultas al DataMart a través de la herramienta Mondrian, la PC

cliente debe tener un procesador 1.40 GHz, y 256 MB de RAM y tener instalado el navegador web Mozilla Firefox 2.0.17 o superior.

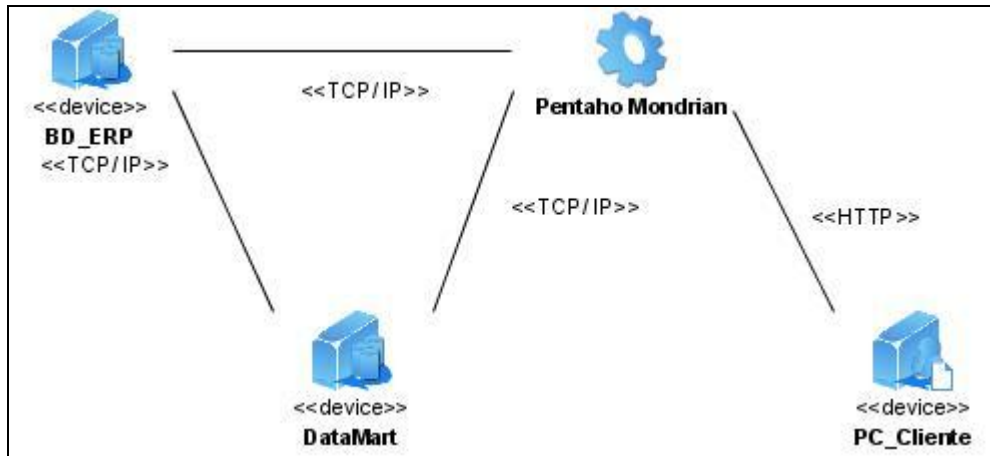


Figura. 23 Diagrama de despliegue del DataMart

3.5 Conclusiones del capítulo

En este capítulo se realizaron las pruebas de rendimiento y de validación con el cliente para evaluar el diseño del DataMart y su correcto funcionamiento acorde a los requisitos para el cual fue construido. Las pruebas aplicadas arrojaron ver los bajos tiempos de respuesta del sistema.

En este capítulo se realizó la validación de la aplicación mediante las pruebas de rendimiento y aceptación por el cliente para evaluar el diseño del DataMart y su correcto funcionamiento. Las pruebas aplicadas arrojaron resultados satisfactorios respaldados por los bajos tiempos de respuesta del sistema y la correspondencia del mismo con las expectativas y necesidades del cliente.

CONCLUSIONES GENERALES

Una vez finalizado el presente trabajo de diploma se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos, para esto:

- Se realizó un estudio para comprobar en qué estado se encontraba la obtención de razones financiera actualmente, para ello se analizaron un conjunto de sistemas nacionales y extranjeros relacionados con la gestión financiera; evidenciándose de esta manera la no existencia de una solución informática capaz de realizar el cálculo de las mismas y que cumpliera con los requisitos establecidos a nivel nacional.
- Se realizó el diseño y la implementación de un DataMart para los subsistemas Cobro y Pagos, Caja y Banco, con el objetivo de organizar la información de manera existente de manera que se pudieran hacer análisis con ella a través del cálculo de las razones financieras.
- Se realizaron pruebas de conceptos y de validación con el cliente lo que permitió evaluar la viabilidad del DataMart, las cuales arrojaron resultados favorables posibilitando dar cumplimiento a las funcionalidades previstas para el mismo.

RECOMENDACIONES

- Se recomienda actualizar el sistema paulatinamente a medida que se creen las condiciones para realizar la consolidación totalmente.
- Se recomienda integrar el sistema al marco de trabajo del ERP.
- Se recomienda buscar una estrategia que posibilite al sistema gestionar la seguridad a nivel de entidades.

BIBLIOGRAFÍA REFERENCIADA

1. **Díaz Morales, Themis Patricia y Salvador Bermúdez, José.** *Diseño de un Datawarehouse para los Ensayos Clínicos que se gestionan en el Centro de Inmunología Molecular.* FAC 6, Universidad de las Ciencias Infomáticas. Ciudad de la Habana : s.n., Junio de 2010. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.
2. **Miguel, Rodriguez Zanz.** *Análisis y diseño de un datamart para el seguimiento académico de alumnos en un entorno universitario.* Informática, Universidad Carlos III de Madrid. Madrid : s.n., 2010. Proyecto Fin de Carrera.
3. **Everis.** *Arquitectura de un DataWarehouse.* enero,2007.
4. **Bartomeus, Pol Rojas.** *Estudio comparativo de las bases de datos analíticas.* Barcelona : Facultad de Informatica de Barcelona, septiembre,2009.
5. **Peña Rivera, Juan David and Armando Suarez, Jesús.** *Utilización de la información histórica para decisiones empresariales.* Santa Fe de Bogota : Facultad de Ingeniería, junio,2005.
6. **Hurtado Torres, M. Visitación.** *BASES DE DATOS Y DATA WAREHOUSE: HERRAMIENTAS ESTRATÉGICAS PARA LA EFICACIA COMERCIAL.* Granada : Facultad de Ciencias Económicas y Empresariales. Universidad de Granada, 2007.
7. **Addison-Wesley.** *The relational Model for Database Management: version 2.* 1990.
8. **Fernández, Jorge.** *Business Intelligence, Apuntes de Sistemas de Información.* Barcelona : Facultad de Informática, 2008.
9. **Simón Mir, Yailin y Iñiguez Bermúdez, Yoander.** *Análisis de datos a un Datawarehouse para los Ensayos Clínicos del Centro de Inmunología Molecular, aplicando técnicas OLAP.* FAC 6, Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2010. Trabajo Diploma para optar por el título de Ingeniero en Ciencias Informáticas.
10. **Dario, Ing. Bernabeu Ricardo.** *HEFESTO: Metodología propia para la construcción de un Data Warehouse.* Cordoba, Argentina : s.n., 2009.
11. Recursos en la web. [En línea] [Citado el: 7 de 04 de 2011.] <http://recursosenweb.com/eanaliza-programa-de-analisis-de-negocios/>.

12. Informatica hoy. [En línea] [Citado el: 25 de 2 de 2011.] <http://www.informatica-hoy.com.ar/sap/SAP-Business-ONE-Administracion-Presentacion-Informes.php>.
13. Sitio Oficial Open Bravo. [En línea] [Citado el: 17 de 4 de 2011.] <http://www.openbravo.com/es/product/erp/>.
14. [En línea] [Citado el: 22 de 03 de 2011.]
15. ASSET. [En línea] [Citado el: 25 de 03 de 2011.] <http://www.assets.co.cu/ventajas.asp>.
16. Metodología UCID para el Desarrollo de Data Warehouse. s.l. : Sistema de Apoyo a la Toma de Decisiones.
17. **Ibarra, María de los Ángeles.** . IBM DB2 OLAP SERVER. [En línea]
18. **González, Mairelys Fernández y Rivera, Osley Zorrilla.** *Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Cedrux.* Ciudad de la Habana : s.n., junio,2010.
19. **Morales, González.** Inteligencia empresarial para la toma de decisiones en la PYME enfocada en la administración de la relación con el cliente (CRM) utilizando análisis de la canasta de compra (MBA). Escuela de Ingeniería y Ciencias, Universidad de las Américas. [En línea] 2007. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_m_ac/capitulo_3.html..
20. Acis.org. [En línea] [Citado el: 09 de 03 de 2011.] <http://www.acis.org.co/index.php?id=623>.
21. mike2. [En línea] [Citado el: 18 de 05 de 2011.] <http://mike2.openmethodology.org/wiki/Talend>.
22. *Análisis del estado de Mondrian (OLAP Open Source).* **Giménez, Javier.** 2007.
23. *Implementación de un DataMart para la Unidad Central de Cooperación Médica.* **López, Leydis Hidalgo y Cartaya, Alier Caballero.** Ciudad de La Habana : s.n., Junio 2010.
24. **Rojas Bartomeus, Pol.** *Estudio comparativo de bases de datos analíticas.* Ingeniería del Software y Sistemas de Información, Everis – Facultat de Informàtica de Barcelona – UPC. 2009. Proyecto Final de Máster.
25. Wolff, Carmen Gloria. La Tecnología Datawarehousing. La Tecnología Datawarehousing. [En línea] 28 de 8 de 2002. [Citado el: 21 de 3 de 2011.] <http://www.inf.udec.cl/revista/edicion3/cwolff.htm>.

BIBLIOGRAFÍA CONSULTADA

1. **Díaz Morales, Themis Patricia y Salvador Bermúdez, José.** *Diseño de un Datawarehouse para los Ensayos Clínicos que se gestionan en el Centro de Inmunología Molecular.* FAC 6, Universidad de las Ciencias Infomáticas. Ciudad de la Habana : s.n., Junio de 2010. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.
2. **Miguel, Rodriguez Zanz.** *Análisis y diseño de un datamart para el seguimiento académico de alumnos en un entorno universitario.* Informática, Universidad Carlos III de Madrid. Madrid : s.n., 2010. Proyecto Fin de Carrera.
3. **Everis.** *Arquitectura de un DataWarehouse.* enero,2007.
4. **Bartomeus, Pol Rojas.** *Estudio comparativo de las bases de datos analíticas.* Barcelona : Facultad de Informatica de Barcelona, septiembre,2009.
5. **Peña Rivera, Juan David and Armando Suarez, Jesús.** *Utilización de la información histórica para decisiones empresariales.* Santa Fe de Bogota : Facultad de Ingeniería, junio,2005.
6. **Hurtado Torres, M. Visitación.** *BASES DE DATOS Y DATA WAREHOUSE: HERRAMIENTAS ESTRATÉGICAS PARA LA EFICACIA COMERCIAL.* Granada : Facultad de Ciencias Económicas y Empresariales. Universidad de Granada, 2007.
7. **Addison-Wesley.** *The relational Model for Database Management: version 2.* 1990.
8. **Fernández, Jorge.** *Business Intelligence, Apuntes de Sistemas de Información.* Barcelona : Facultad de Informática, 2008.
9. **Simón Mir, Yailin y Iñiguez Bermúdez, Yoander.** *Análisis de datos a un Datawarehouse para los Ensayos Clínicos del Centro de Inmunología Molecular, aplicando técnicas OLAP.* FAC 6, Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2010. Trabajo Diploma para optar por el título de Ingeniero en Ciencias Informáticas.
10. **Dario, Ing. Bernabeu Ricardo.** *HEFESTO: Metodología propia para la construcción de un Data Warehouse.* Cordoba, Argentina : s.n., 2009.
11. Recursos en la web. [En línea] [Citado el: 7 de 04 de 2011.] <http://recursosenweb.com/eanaliza-programa-de-analisis-de-negocios/>.

12. Informatica hoy. [En línea] [Citado el: 25 de 2 de 2011.] <http://www.informatica-hoy.com.ar/sap/SAP-Business-ONE-Administracion-Presentacion-Informes.php>.
13. Sitio Oficial Open Bravo. [En línea] [Citado el: 17 de 4 de 2011.] <http://www.openbravo.com/es/product/erp/>.
14. [En línea] [Citado el: 22 de 03 de 2011.]
15. ASSET. [En línea] [Citado el: 25 de 03 de 2011.] <http://www.assets.co.cu/ventajas.asp>.
16. Metodología UCID para el Desarrollo de Data Warehouse. s.l. : Sistema de Apoyo a la Toma de Decisiones.
17. **Ibarra, María de los Ángeles.** . IBM DB2 OLAP SERVER. [En línea]
18. **González, Mairelys Fernández y Rivera, Osley Zorrilla.** *Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Cedrux.* Ciudad de la Habana : s.n., junio,2010.
19. **Morales, González.** Inteligencia empresarial para la toma de decisiones en la PYME enfocada en la administración de la relación con el cliente (CRM) utilizando análisis de la canasta de compra (MBA). Escuela de Ingeniería y Ciencias, Universidad de las Américas. [En línea] 2007. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_m_ac/capitulo_3.html..
20. Acis.org. [En línea] [Citado el: 09 de 03 de 2011.] <http://www.acis.org.co/index.php?id=623>.
21. mike2. [En línea] [Citado el: 18 de 05 de 2011.] <http://mike2.openmethodology.org/wiki/Talend>.
22. *Análisis del estado de Mondrian (OLAP Open Source).* **Giménez, Javier.** 2007.
23. *Implementación de un DataMart para la Unidad Central de Cooperación Médica.* **López, Leydis Hidalgo y Cartaya, Alier Caballero.** Ciudad de La Habana : s.n., Junio 2010.
24. **Rojas Bartomeus, Pol.** *Estudio comparativo de bases de datos analíticas.* Ingeniería del Software y Sistemas de Información, Everis – Facultat de Informàtica de Barcelona – UPC. 2009. Proyecto Final de Máster.
25. Wolff, Carmen Gloria. La Tecnología Datawarehousing. La Tecnología Datawarehousing. [En línea] 28 de 8 de 2002. [Citado el: 21 de 3 de 2011.] <http://www.inf.udec.cl/revista/edicion3/cwolff.htm>.

GLOSARIO DE TÉRMINOS

A continuación se presentan los términos que podrían resultar de difícil comprensión, nuevos al lector o de diversos significados dependiendo del contexto que se analice. Esta sección tiene como objetivo facilitar la comprensión del contenido expuesto en el documento.

SQL: Lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella.

Apache Tomcat (también llamado Jakarta Tomcat o Apache Tomcat): Es un servidor web con soporte de servlets y JSPs. fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

BD: Una Base de Datos es un conjunto de datos relacionados entre sí, entendiéndose por dato los hechos conocidos, que pueden registrarse y que tienen significado implícito.

Inteligencia de negocios o BI (del inglés Business Intelligence): Conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

Visual Basic: Es un lenguaje de programación dirigido por eventos, desarrollado por el alemán Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y, en cierta medida, también la programación misma.

Microsoft SQL Server: Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server.

Crystal Reports: Es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos (bases de datos). Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión de Crystal Reports como una herramienta de propósito general del informes/reportes.

SGBD (Sistema de Gestión de Base de Datos, en inglés Database Management System o DBMS): Conjunto de aplicaciones de propósito general que permiten crear y mantener una base de datos, con una arquitectura específica, asegurando su integridad, confidencialidad y seguridad. Permite definir, construir, manipular y consultar una base de datos.

Modelo conceptual: Descripción de alto nivel de la estructura de la base de datos, en la cual la información se representa a través de indicadores y perspectivas.

Modelo lógico: Representación de una estructura de datos, que puede procesarse y almacenarse en algún Sistema Gestor de Bases de Datos (SGBD). A diferencia de los modelos lógicos comunes, el modelo lógico de un DW tiene en cuenta tres elementos fundamentales para su modelación: tipología de esquema que será utilizada, tabla de dimensiones y tabla de hechos. Otros elementos que distinguen este modelo son las uniones y las jerarquías, que se tienen en cuenta en el proceso de diseño.

Diccionario de datos: Artefacto que se utiliza para definir la correspondencia que existe entre las variables de las tablas, bases de datos o modelos determinados con sus perspectivas.