

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Título: Desarrollo de un Data Mart para la obtención de los indicadores financieros de los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP.**

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

**Autor(es):** Liván Valdés Yero

**Tutor(es):** Ing. Yoan Arlet Carrascoso Puebla

Ing. Joisel Pérez Pérez

La Habana, Junio 2011



*"El conocimiento nos hace responsables"*

*Che*

**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 \_\_\_\_\_

Liván Valdés Yero

\_\_\_\_\_

Firma del Autor

Ing. Joisel Pérez Pérez

\_\_\_\_\_

Firma del Tutor

Ing. Yoan Arlet Carrascoso Puebla

\_\_\_\_\_

Firma del Co-tutor

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. Actualmente se desempeña como arquitecto de datos del módulo Costos y Procesos del proyecto ERP. 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.

**Síntesis del Co-Tutor: Ing. Yoan Arlet Carrascoso Puebla**

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el 2008. Ha trabajado como desarrollador en sistema de gestión hotelera para el ministerio del Turismo Cubano y como arquitecto principal en Sistemas de Gestión de Inventario y Almacenes para el Ministerio del Turismo basados sobre la plataforma J2EE. Ha formado parte del equipo central de arquitectura como arquitecto de integración del proyecto ERP. Posteriormente se he desempeñado como arquitecto de sistema y de datos de la línea contabilidad financiera del proyecto ERP Cubano. Actualmente pertenece al equipo central de arquitectos del proyecto ERP que se encuentra en desarrollo basado en tecnologías libres como PHP y PostgreSQL.

La realización de este trabajo ha sido posible gracias al apoyo y colaboración de muchas personas, que si en algún momento no menciono algunos de ellos por favor les pido disculpas.

Agradecer en primer lugar a mis padres, mi hermana y mi tío Albe por la formación que me han dado a lo largo de todos estos años, por haber depositado toda su confianza en que yo si podía lograr lo que me propusiera en la vida. Por haber sido tolerantes a lo largo de todos estos años cuando tenía que quedarme en la escuela algún que otro fin de semana.

A mi tío Beto por apoyarme siempre en lo que necesitara.

A toda mi familia en general porque de una forma u otra siempre han estado dándome el aliento que he necesitado en los momentos cruciales.

A mi tutor, por haber sido más que un tutor, por haberme enseñado lo que era sacrificarse por algo, no importa lo difícil que sea, por no haberse dado por vencido cuando las cosas estaban feas, por todos los conocimientos transmitidos a lo largo de todos estos años de trabajo en conjunto.

A Rainer, Rodolfo, Adolfo y a Ana que han sido mi pequeña familia aquí en la universidad, en todo momento.

A mis amigos, a todas esas personas que de una forma u otra han estado ahí en los momentos que he necesitado

A todas las personas que han contribuido al desarrollo de este trabajo, a todas esas personas que en muchas ocasiones no he dejado descansar.

Con admiración y respeto les agradecemos a todos.

Agradecer en primer lugar a mi papá y mi abuela mimas que de seguro estarían orgullosos de mí.

A mi mamá, mi hermana y mi tío Albe por haber soportado todas mis malcriadeces.

A mis familiares y a todos aquellos que de una forma u otra han estado presentes durante el desarrollo de mi vida.

A todas esas personas que pusieron todo su empeño en que yo lograría mis objetivos.

A todos mis amigos y demás personas que me brindaron su apoyo en los momentos difíciles.

**RESUMEN**

En la actualidad, a las organizaciones en sentido general les resulta de vital importancia contar con información actualizada en cuanto al estado de sus negocios. Esta constante necesidad ha provocado que muchas de ellas se encuentren inmersas en el proceso de informatización de sus procesos de gestión empresarial con el objetivo de agilizar los mismos. Más allá de contar con información actualizada sobre los procesos contables que reflejen lo sucedido o lo que está sucediendo en una empresa, a muchos directivos les resulta más objetivo el poder realizar proyecciones futuras en cuanto a materia de negocio. Para ello es de vital importancia la aplicación de la inteligencia de negocios mediante el análisis de estados financieros así como de sus indicadores, a partir de técnicas y mecanismos que posibiliten efectuar un estudio de tendencias, mostrar el estado de salud económica de una empresa, evitar el endeudamiento de la misma y con ello su insostenibilidad a largo plazo por solo citar algunos ejemplos. Actualmente dentro del producto CEDRUX, los Subsistemas de Contabilidad Costos y Procesos posibilitan solamente llevar el control de todo el flujo contable a través de la gestión de cuentas. De ahí que este trabajo presenta un Data Mart para realizar el análisis de estos datos permitiendo la obtención de indicadores financieros que favorezcan el proceso de toma de decisiones y la anticipación de acciones para dar una dirección bien informada a las empresas cubanas.

**PALABRAS CLAVES:** Data Mart, indicadores financieros, Subsistemas de Contabilidad Costos y Procesos.

ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1. FUNDAMENTACIÓN DEL MARCO TEÓRICO .....	4
1. DEFINICIONES FUNDAMENTALES DE DATA WAREHOUSE.....	4
1.1. Objetivos y Características de un Data Warehouse .....	4
1.2. Data Mart .....	5
1.2.1. Procesos que intervienen en la obtención de un Data Mart .....	6
1.2.1.1. Proceso de extracción, transformación y carga de datos.....	6
1.2.1.2. Proceso de explotación .....	7
1.2.1.3. Metadatos.....	7
1.3. DW vs DM.....	7
1.4. Modelo .....	9
1.5. Modelo de Datos .....	10
1.5.1. Modelo Relacional.....	11
1.5.1.1. Ventajas y desventajas.....	12
1.5.2. Modelo Multidimensional.....	13
1.5.2.1. Componentes del Modelo Multidimensional.....	14
1.5.2.1.1. Estilos de diseño de las Bases de Datos Multidimensionales .....	16
1.5.2.1. Ventajas del Modelo Multidimensional .....	19
1.6. Estado actual del cálculo de los indicadores financieros.....	19
1.6.1. En el mundo .....	20
1.6.2. En Cuba .....	23
1.6.3. Valoración del estudio del estado del arte .....	25
1.7. Metodologías estudiadas para la construcción del Data Mart .....	26
1.7.1. Metodología para el diseño o de arquitectura.....	26
1.7.2. Metodologías para el desarrollo.....	28



1.7.3. Justificación de las metodologías a utilizar .....	31
1.8. Herramientas para la construcción de un Data Mart.....	31
1.8.1. Sistemas gestores de Bases de Datos .....	32
1.8.2. Herramientas de integración de datos .....	34
1.8.3. Herramientas para diseñar cubos OLAP .....	37
1.8.4. Herramientas CASE .....	38
1.8.5. Servidor de aplicaciones .....	39
1.9. Conclusiones del capítulo .....	39
CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL DATA MART.....	41
2.1. Análisis de los requerimientos.....	41
2.1.1. Cronograma de entrevistas.....	41
2.1.2. Planificar entrevistas .....	42
2.1.3. Identificar perspectivas e indicadores .....	44
2.2. Análisis de los OLTP .....	47
2.2.1. Determinación de indicadores por cada hecho identificado.....	47
2.2.2. Nivel de granularidad .....	48
2.2.3. Modelo Conceptual ampliado de los hechos con sus respectivas perspectivas e indicadores identificados .....	49
2.3. Diccionario de datos del modelo conceptual ampliado de los hechos con sus respectivas perspectivas e indicadores identificados.....	50
2.3.1. Diccionario de datos de las perspectivas.....	50
2.3.2. Diccionario de datos de los indicadores.....	53
2.4. Modelo lógico del Data Mart .....	54
2.4.1. Definir tipo de Modelo lógico del Data Mart .....	55
2.4.2. Definir estándares para objetos físicos .....	55
2.4.3. Diseñar tablas y columnas físicas .....	58

2.4.3.1. Diseñar las tablas auxiliares físicas para la concepción estructuras en forma de árbol.....	59
2.5. Proceso ETL .....	60
2.5.1. Mapeo de datos fuente a destino.....	60
2.5.2. Establecer condiciones adicionales y restricciones .....	61
2.5.3. Diseño y construcción de la automatización del proceso de ETL.....	61
2.6. Conclusiones del capítulo .....	63
CAPÍTULO 3: VALIDACIÓN Y DESPLIEGUE .....	64
3.1. Validación .....	64
3.1.1. Validación funcional .....	64
3.1.2. Validación del diseño mediante pruebas de rendimiento a la aplicación .....	65
3.3. Despliegue del Data Mart.....	67
3.4. Conclusiones del capítulo.....	68
CONCLUSIONES GENERALES .....	69
RECOMENDACIONES.....	69
BIBLIOGRAFÍA REFERENCIADA.....	70
BIBLIOGRAFÍA CONSULTADA .....	72
GLOSARIO DE TÉRMINOS .....	73

## **Introducción**

El auge del desarrollo de las Ciencias Informáticas en el mundo ha provocado el surgimiento de una nueva forma de apreciar las soluciones a los problemas que se presentan a diario. Este nuevo modo de apreciar dichas soluciones ha estado relacionado con el aumento de la utilización de las Tecnologías de la Información y las Comunicaciones (TIC) en la sociedad, formando parte de su desarrollo sostenible. De igual modo, desde el surgimiento del comercio, la contabilidad ha proporcionado al hombre la posibilidad de obtener mejores resultados en cuanto a cómo controlar y organizar económicamente cualquier institución o entidad; garantizándole a la misma permanencia y estabilidad a largo plazo.

Diariamente las empresas generan gran cúmulo de datos producto de las transacciones realizadas por sus operaciones contables, en muchos casos realizadas manualmente. Una de las vías para agilizar este proceso ha sido la creación de sistemas denominados Enterprise Resource Planning (ERP por sus siglas en inglés), posibilitando además, integrar y automatizar los procesos de negocio de las empresas, agilizar sus tareas, mejorar la gestión y reducir los costos, aumentando la agilidad de los mismos y disminuyendo el tiempo de respuesta, apoyando así de manera eficiente el proceso de toma de decisiones.

Hoy en día, Cuba, y en particular la Universidad de las Ciencias Informáticas (UCI), se encuentra desarrollando un Sistema Integral de Gestión de Entidades (CEDRUX) correspondiente al proyecto ERP con el objetivo de automatizar todos los procesos contables que se realizan en las entidades cubanas. En este sistema resulta vital la aplicación de la inteligencia de negocios (BI por sus siglas en inglés) dentro de los subsistemas de Contabilidad y Costos y Procesos mediante el análisis de los estados financieros, así como de sus indicadores; constituyendo mecanismos que posibilitan realizar análisis de tendencias, mostrar el estado de salud económica de una empresa, evitar el endeudamiento de la misma y con ello su insostenibilidad a largo plazo por solo citar algunos ejemplos.

Lo que se busca es encontrar información que no solamente conteste preguntas de lo que pasó o lo que está sucediendo en el negocio, sino que también a través de esta información se construyan modelos e indicadores financieros que ayuden a predecir futuros eventos de negocio. Típicamente, esta información es obtenida por el sistema de forma transaccional almacenándose como resultado muchos datos que son transparentes para los directivos, siendo engorroso su procesamiento y transformación en conocimiento que enriquezca el proceso de toma de decisiones en tiempo real.

De ahí que el presente trabajo de diploma para brindar solución a la situación descrita con anterioridad, posee el siguiente **problema a resolver**. El estado actual del análisis de los datos generados en los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP afecta la obtención de los indicadores financieros en las entidades cubanas.

Se define por lo tanto como **objeto de estudio**: el análisis de los datos de la contabilidad y los costos en las entidades cubanas, tomando como **campo de acción**: el análisis de los datos en los subsistemas Contabilidad y Costos y Procesos del proyecto ERP.

Para dar respuesta al problema planteado se trazó el siguiente **objetivo general**: Desarrollar un Data Mart para realizar el análisis de los datos generados en los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP que permita obtener indicadores financieros.

**Objetivos específicos:**

1. Evaluar la obtención de los indicadores financieros en los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP.
2. Fundamentar la investigación, mediante la elaboración del Marco Teórico.
3. Desarrollar un Data Mart para los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP.
4. Validar artefactos de salida mediante pruebas en cada una de las fases del proceso de desarrollo de software.

**Idea a defender:** Si se implementa un Data Mart para obtener los indicadores financieros de los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP se favorecerá el proceso de toma de decisiones y la construcción de modelos para predecir futuros eventos contables en las entidades cubanas.

**Tareas para cumplir los objetivos:**

1. Realización del estudio del estado del arte.
  - 1.1. Análisis y definición de indicadores financieros.
  - 1.2. Investigación sobre el estado del arte de metodologías de construcción de Data Marts.
  - 1.3. Investigación de las buenas prácticas para la construcción de un Data Mart.
  - 1.4. Realización del estudio del estado del arte de herramientas para el proceso de extracción transformación y carga de datos.
  - 1.5. Elaboración del estado del arte de herramientas para la representación de la información.
2. Definición de dimensiones del Data Mart.
3. Definición de medidas naturales y calculadas.
4. Elaboración de estrategias para la estandarización de los datos de las fuentes de origen.
5. Realización del diseño del modelo de datos del Data Mart.
6. Implementación del modelo de datos del Data Mart.
7. Realización de las vistas de los indicadores financieros

8. Realización de pruebas de diseño al modelo de datos.

**ESTRUCTURA DEL DOCUMENTO:**

**Capítulo 1:** Se expone el estado del arte, donde se realiza la fundamentación teórica del tema. Al mismo tiempo se describe el objeto de estudio, se analizan y definen los indicadores financieros, se explica la metodología para la construcción de un Data Mart, se realiza el estudio y/o selección de las herramientas, así como de los artefactos para su elaboración.

**Capítulo 2:** Se realiza el diseño e implementación del Data Mart, se definen las dimensiones y las medidas, y se elaboran los artefactos seleccionados en el capítulo anterior para su construcción a partir de la metodología escogida.

**Capítulo 3:** Se plantean diferentes tipos de pruebas realizadas al Data Mart 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.

**Posibles resultados:**

Desarrollo de un Data Mart para la obtención de los indicadores financieros de los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP.

## **CAPÍTULO 1. FUNDAMENTACIÓN DEL MARCO TEÓRICO**

### **Introducción**

En el presente capítulo se abordan algunas de las definiciones fundamentales de Data Mart (DM) y Data Warehouse (DW) así como algunas de sus características fundamentales, se realiza un estudio de la situación actual del cálculo de los indicadores financieros así como de los principales sistemas que se enmarcan en el desarrollo del DM, ya sean sistemas nacionales o internacionales. Se realiza un estudio de las principales metodologías para realizar su desarrollo y conocer las diferentes arquitecturas propuestas, además de analizar las herramientas y tecnologías que pueden ser empleadas en su construcción.

### **1. Definiciones fundamentales de Data Warehouse**

A partir de mediados de los ochenta, en el entorno empresarial, ha cobrado importancia el concepto de Data Warehouse (DW por sus siglas en inglés) o almacén-factoría de datos, entendido como la plataforma que concentra toda la información de interés para la organización. Sus fuentes de información son tanto las bases de datos corporativas, como otras fuentes externas dígame hojas de cálculo Excel o algún otro fichero que contenga información.

En el mundo existen numerosas definiciones para el DW, la más conocida fue propuesta por Inmon (considerado el padre de los Data Warehouse) en 1992, el planteaba que: “Un DW es una colección de datos orientados a temas, integrados, no-volátiles y variante en el tiempo, organizados para soportar necesidades empresariales”.(8)

Otra definición acertada fue dada en 1993 por Susan Osterfeldt enfocándolo como algo que provee dos beneficios empresariales reales, en primer lugar la Integración y acceso de datos, eliminación de una gran cantidad de datos inútiles y no deseados; así como también el procesamiento desde el ambiente operacional clásico(27), por otro lado Ralph Kimball, considerado uno de los grandes exponentes en este campo planteo que: “El Data Warehouse es la unión de todos los Data Marts de una entidad”. (1) Asumiendo para esta investigación la definición dada por Inmon y la de Kimball que si bien no se contradicen manifiestan las características fundamentales que presentan estos sistemas. (2)

#### **1.1.Objetivos y Características de un Data Warehouse**

Los principales objetivos de un Data Warehouse son:

- Comprender las necesidades de los usuarios por áreas dentro del negocio.
- Determinar qué decisiones se pueden tomar con la ayuda del DW.
- Seleccionar un subconjunto del sistema de fuentes de datos que sea el más efectivo y procesable

para presentar el DW.

- Asegurar que los datos sean precisos, correctos y confiables y que mantengan la consistencia.
- Monitorear continuamente la precisión y exactitud de los datos y el contenido de los reportes generados.
- Publicar los datos.

Como ya se mencionó un DW debe tener cuatro características primarias. Es una colección de datos orientada a un tema, integrada, variable en el tiempo y no volátil que sea útil para la toma de decisiones.

- Orientada a un tema porque tiene en cuenta los procesos de negocio de la empresa que se deseen priorizar.
- Integrado porque agrupa a todos los sistemas operacionales en un sistema de información con formatos y códigos consistentes.
- Histórico o variante en el tiempo porque los datos se organizan y almacenan en jerarquías en el tiempo, lo que permiten realizar análisis comparativos de estados actuales y de períodos anteriores.
- No volátil ya que se usa principalmente para operaciones de recuperación de información y no para actualizaciones.

## **1.2. Data Mart**

Un Data Mart es una base de datos departamental, la cual se especializa en el almacenamiento de datos de un área específica del negocio. Los DMs poseen una estructura de datos óptima para analizar detalladamente la información que en ellos se almacenan desde todas las perspectivas que pueden afectar los procesos del departamento.

Un Data Mart puede obtener información desde los datos almacenados en un Data Warehouse o integrar por sí mismo un compendio de distintas fuentes de información. Es considerado un Data Warehouse con función departamental o regional contando con sus mismas características y brindando sus 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 un Data Mart 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 Data Marts 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. (3)

### **1.2.1. Procesos que intervienen en la obtención de un Data Mart**

La construcción de un Data Mart es el resultado de un proceso consciente y ordenado por parte de cada organización, en gran número de ocasiones puede implicar cambios en el entorno del negocio tanto por las mejoras que introduce como por la reorganización de los procesos que durante su construcción se llevan a cabo. La aplicación de un Data Mart está orientada a la toma de decisiones por lo que un buen diseño de su base de datos (BD) favorece el análisis y la recuperación de datos para obtener una ventaja estratégica; todo esto unido al crecimiento de la complejidad de las BD de los sistemas operacionales y el aumento de los requisitos por parte de los clientes que utilizarán los reportes generados a partir del mismo, torna complejo el procedimiento para su implementación, de ahí que sea necesaria la realización de procesos que contribuyan a la creación del mismo, dichos procesos son:

- Proceso de extracción, transformación y carga de datos.
- Proceso de explotación.
- Metadatos.

#### **1.2.1.1. Proceso de extracción, transformación y carga de datos**

Los datos para almacenarlos en un Data Mart necesitan agregarse, analizarse, computarse, procesarse matemáticamente, etc., y en muchos casos también se hace necesario realizar transformaciones específicas, los procesos de extracción, transformación y carga de datos conocidos como ETL (por sus siglas en inglés) recuperan los datos de todos los sistemas operacionales, permitiendo que la información de los mismos pueda transformarse y moverse desde el sistema operacional, u otros sistemas, al Data Mart. Este constituye uno de los procesos más importantes en el desarrollo del Data Mart, pudiéndose decir que la exactitud de la plataforma BI entera va a depender en gran medida de la calidad de los procesos ETL.

#### **Proceso de extracción y limpieza de datos:**

En este proceso se procede a extraer la información de las diferentes fuentes de origen de datos para su posterior limpieza posibilitando de esta manera eliminar redundancias, gestionar la corrección de errores, entre otros problemas que pueden presentar los mismos.

Estas fuentes de origen son las encargadas de alimentar el Data Mart, estando en muchos casos diseñadas para registrar grandes cantidades de transacciones. Una de las fuentes más comunes es la base de datos operacional (OLTP por sus siglas en inglés), estas son bases de datos que como requisito fundamental poseen el de soportar procesos transaccionales presentado algunas de las siguientes



características:

- Son pobladas por usuarios finales.
- Se optimizan en función de los procesos transaccionales.
- Se actualizan constantemente.
- Contienen mucha información de detalle.
- Generalmente se encuentran normalizadas hasta segunda o tercera forma normal.

### **Proceso de transformación:**

Luego de ser identificadas las diferentes fuentes de alimentación de datos es necesario realizar la unificación de los mismos, en este marco se define un único tipo de datos para cada uno de los campos al que se tendría que llevar la información de las distintas fuentes de alimentación externas.

### **Proceso de carga:**

Una vez realizado el proceso de extracción, limpieza y transformación de datos se procede a cargar los mismos en el almacén de datos, una vez concluido este proceso se puede decir que la información ya está lista para ser analizada por los usuarios.

#### **1.2.1.2. Proceso de explotación**

Estando la información en el almacén de datos se puede pasar al proceso de explotación del Data Mart, que mediante el empleo de diferentes herramientas de consulta realiza la extracción y análisis de la información en los distintos niveles de agrupación que se han definido o que el usuario final desee.

#### **1.2.1.3. Metadatos**

Estos describen los tipos de datos, las definiciones físicas y lógicas de los mismos, las consultas e informes predefinidos, las reglas de validación y negocio, las definiciones de las fuentes de datos, las rutinas de transformación y de proceso. En definitiva, se refieren a cualquier estructura que define un objeto del Data Mart.

### **1.3. DW vs DM**

Como ya se ha abordado previamente, el concepto Data Mart es una extensión natural del Data Warehouse estando enfocado a un departamento o área específica del negocio, permitiendo de esta manera un mejor control de la información que se está abarcando. Esta es la principal característica que diferencia a los mismos, aunque existen otros aspectos por los que se pueden realizar comparaciones

entre ellos, por ejemplo, costo de diseño e implementación, alcance, conexiones de usuarios, entre otros por solo citar algunos ejemplos.

Debido al alcance de los Data Marts en ocasiones se hace difícil coordinar el flujo de los datos a través de los múltiples departamentos que ellos representan, por otro lado cada departamento tendrá su propia visión de cómo un Data Mart debe lucir, siendo cada Data Mart específico para cada uno de ellos. Por el contrario, un Data Warehouse se encuentra diseñado en torno a toda la organización en su conjunto, en lugar de ser propiedad de un departamento, será propiedad de la compañía entera. También hay que considerar que la implementación de varios Data Mart para una organización puede acarrear conflictos si presentan diferentes fuentes o diferentes periodos de actualización. Algunas de las diferencias más relevantes entre ambos se presentan en la Tabla 1.

	<b>Data Warehouse</b>	<b>Data Mart</b>
<b>Costoso en diseño e implementación.</b>	Es más costoso de diseñar e implementar.	Es menos costoso de diseñar e implementar.
<b>Distribución de información útil para la toma de decisiones.</b>	No optimiza la distribución de información útil para la toma de decisiones.	Optimiza la distribución de información útil para la toma de decisiones.
<b>Radio de acción.</b>	Se ajusta a varias áreas de procesos.	Se ajusta mucho a las necesidades que tienen un área específica.
<b>Conexiones de usuarios.</b>	Soporta mayor cantidad de usuarios.	Soporta menos usuarios.
<b>Alcance.</b>	Posee un alcance menos limitado.	Posee un alcance más limitado.
<b>Consistencia.</b>	Consistente.	La proliferación de estos puede llevar a la inconsistencia.
<b>Integración a un DW.</b>	No se integra.	Puede integrarse eventualmente a un DW.

<b>Obtención de resultados.</b>	Los resultados no se muestran hasta el final del proceso de implementación.	Los resultados pueden observarse en la medida que se termine cada uno de los mismos.
---------------------------------	---	--

Tabla 1: Comparación entre un Data Warehouse y un Data Mart

No obstante, por lo costoso que resulta la implementación de un Data Warehouse en cuanto a tiempo y recursos, además de no poder contar con los resultados hasta el final del proceso; muchas organizaciones deciden la construcción paulatina de Data Marts departamentales garantizándole la recuperación de información de forma gradual resultando en ventajas para la organización.

#### 1.4. Modelo

Un modelo es una representación de la realidad que contiene las características generales de algo que se va a realizar. Es la representación en pequeña escala de alguna cosa, aunque también se le conoce como un esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, como la evolución económica de un país, que se elabora para facilitar su comprensión y el estudio de su comportamiento.(4) Puede involucrar tanto planos detallados como planos más generales que ofrecen una visión global del sistema en consideración.

El modelado es común en los proyectos de software exitosos, constituye una técnica de ingeniería probada y bien aceptada, que entre otros factores ayuda a:

- Visualizar a los usuarios el producto final.
- Comprender mejor el sistema.
- Comunicar las ideas a otros.

El modelado es útil tanto en pequeños como en grandes sistemas. Mientras más grande y complejo sea el sistema el modelado se hace más importante por una simple razón: se construyen modelos de sistemas porque no es posible comprender el sistema en su totalidad.

Los modelos se construyen para:

- Comunicar la estructura deseada y el comportamiento del sistema.
- Visualizar y controlar la arquitectura.
- Comprender qué se construye, muchas veces descubriendo oportunidades para la simplificación y reutilización.

- Controlar el riesgo.(5)

En sentido general, los modelos son construidos para comprender mejor el sistema que se está desarrollando.

A través del modelado se puede reducir el problema que se está estudiando, posibilitando centrarse en un solo aspecto a la vez. Se puede modelar formal e informalmente, pero este último no proporciona un lenguaje común que se pueda compartir fácilmente con otros. Mientras más complejo sea el sistema, más requiere modelaje. Si se construye un sistema simple y sencillo, al principio no se piensa que este necesite de modelaje, pero si este evoluciona y crece, se lamentará no haberlo realizado, sin perder de vista que aunque los mejores modelos se encuentran ligados a la realidad, simplificándola, hay que asegurarse que las simplificaciones no enmascaren ningún detalle importante.

En el desarrollo de software existen varias formas de enfocar un modelo, ocupando primordial importancia el modelado de la Base de Datos, esta representación se realiza de forma gráfica, pero un único modelo o vista no es suficiente. Cualquier sistema no trivial se aborda mejor a través de un pequeño conjunto de modelos casi independientes con múltiples puntos de vista, significa tener modelos que se puedan construir y estudiar separadamente, pero que aún así estén interrelacionados, de ahí que a la hora de diseñar una Base de Datos sean generados varios de estos modelos.

### **1.5. Modelo de Datos**

La descripción de la estructura de una base de datos, es el modelo de datos, es decir, una colección de herramientas conceptuales para describir datos, sus relaciones, semántica y restricciones. También se puede definir como una representación de la realidad, que conserva sólo los detalles relevantes de esa realidad que se está modelando.(5)

Para el propósito de esta investigación, el Modelo de Datos es un conjunto de conceptos que pueden servir para describir la estructura de una Base de Datos, esto se refiere a tipos de datos, sus vínculos y restricciones que deben cumplir los mismos.(5)

Un modelo de datos se define como la combinación de tres componentes:

- Una colección de tipos de objetos de información, los cuales son las unidades básicas para construir cualquier base de datos.
- Una colección de reglas generales de integridad, las cuales limitan el conjunto de los tipos de objetos que pueden aparecer en forma legal en cualquier base de datos.
- Una colección de operadores, aplicables a los objetos para obtener información y para otros

propósitos.(5)

Los principales objetivos del proceso de modelamiento de datos es saber identificar cual es el problema y encontrar la forma de representarlo en un sistema. Esto significa saber de los datos, quiénes van a usarlos y cómo van a usarlos. El modelado de datos constituye uno de los elementos más importantes a la hora de iniciar el desarrollo de cualquier proyecto, pues constituye la estructura sobre la que reside la verdadera esencia de la aplicación. El modelado de datos es además una técnica independiente de la implementación de la base de datos.

Cuando un Modelo de Datos describe un conjunto de conceptos de un problema real en términos comprensibles por los expertos del dominio de aplicación, se le llama Modelo de Datos conceptual. Los modelos conceptuales se usan en las primeras fases del diseño de Bases de Datos, existen varias categorías de Modelos de Datos:

- Modelos de Datos de Alto Nivel o Conceptual: dispone de conceptos muy cercanos al modo en que la mayoría de los usuarios perciben los datos.
- Modelo de Datos de Bajo Nivel o Físico: proporciona conceptos que describen los detalles de cómo se almacenan los datos en la computadora, dirigidos a especialistas en computación.
- Modelo de Datos de Representación o de Implementación: es un modelo intermedio, cuyos conceptos pueden ser entendidos por los usuarios finales aunque no está demasiado alejado de la forma en que los datos se almacenarán en la computadora.

Ejemplos:

Relacional (5)

### 1.5.1. Modelo Relacional

El Modelo de Datos relacional fue introducido por Edgar Frank Codd (1970). Se basa en una estructura de datos simple y uniforme, la relación.(6) El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Este Modelo representa la BD como una colección de relaciones. Cada relación representa una tabla o hasta cierto punto un archivo simple.

Dentro de una tabla las filas representan tuplas y las columnas los atributos. Cada fila es una colección de valores relacionados entre sí, no son más que hechos que identifican una entidad o vínculo entre entidades. (6)

Codd definió tres características del modelo relacional:

- Son estructuras de datos simples. Consiste en tablas de dos dimensiones donde los elementos son ítem de datos. Esto permite un alto grado de independencia de la representación física de los datos.
- El modelo relacional provee una sólida fundamentación para la consistencia de los datos. El diseño de la base de datos es asistido por los procesos de normalización que elimina las anomalías en los datos. Adicionalmente, el estado de consistencia de la base de datos puede ser uniformemente definido y mantenido a través de reglas de integridad.
- El modelo relacional permite la manipulación de las relaciones. Esta característica puede ser encargada a potentes lenguajes no procedimentales basados en la teoría (álgebra relacional) o en la lógica (cálculo relacional).(7)

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

#### **1.5.1.1. Ventajas y desventajas**

- Una de las principales ventajas del modelo relacional es que su objeto abstracto básico (es decir, la relación) tenga una representación (como tabla) muy sencilla. Esta representación sencilla posibilita que los sistemas relacionales sean fáciles de entender y usar.
- Las bases de datos relacionales luego de una serie de transformaciones son percibidas por el usuario como una colección de relaciones normalizadas de diversos grados evitando redundancia de datos disminuyendo el espacio en almacenamiento de información.
- Por las pocas implicaciones que presenta sobre el sistema algún cambio en el modelo relacional, se puede decir que este modelo es sencillo y fácil de mantener.
- Otra ventaja presentada por el modelo relacional es su compatibilidad y estandarización basándose en que muchas herramientas case utilizan el modelo relacional para el diseño de sus bases de datos. Con lo antes planteado se puede afirmar que es posible encontrar fácilmente asesoramiento en cuanto a cómo diseñar los mismos.

Dentro de las desventajas que pueden presentar los modelos relacionales podemos encontrar:

- Por presentar una visión casi normalizada totalmente, provoca un mayor tiempo de respuesta debido a la cantidad de uniones que se deben realizar entre las diferentes tablas para lograr reunir toda la información que el usuario desea obtener.
- Por la forma que se realiza su diseño resulta complejo establecer de forma clara alguna herencia

que pueda existir.(7)

- Divide los datos en muchas relaciones, cada una de las cuales se convierte en una tabla en la BD OLTP.
- Se torna en un modelo 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 cuales tablas del modelo contienen medidas numéricas y cuales tablas incluyen descriptores estáticos de los objetos.
- Es muy difíciles para las personas (usuarios finales o diseñadores) visualizar y conservar en sus mentes todas las tablas que componen el modelo.
- Si dos tablas en el diagrama son necesarias en una consulta dada, existen un número inmenso de posibles rutas de conexiones entre ambas.
- Para consultas, que alcanzan muchos registros y muchas tablas, los modelos relacionales son muy complejos de comprender por los usuarios y muy complejos de recorrer por el software.

### 1.5.2. Modelo Multidimensional

La modelación dimensional es un nuevo nombre para una técnica antigua que permite hacer simples y comprensibles bases de datos, la cual puede ser visualizada como un “cubo” de tres, cuatro, cinco o más dimensiones, donde cualquier punto interior es una intersección de las coordenadas definidas por los ejes del cubo.(8)

Para poder entender la definición presentada así como el modelo multidimensional, se deben comprender algunos conceptos fundamentales:

Un cubo es la unidad de representación de la información, equivalente a las tablas de las bases de datos relacionales, las que representan la información como matrices.

A los ejes de la matriz se les llama *dimensiones* representando los criterios de análisis, y a los datos almacenados en la matriz se denominan *medidas* y representan los indicadores o valores a analizar. Las dimensiones caracterizan a una actividad o hecho, permitiendo su análisis posterior en el proceso de toma de decisiones, brindando una perspectiva adicional al hecho dado. Son agrupaciones lógicas de atributos con un significado común y atómico. (9)

Se llama hecho a una operación que se realiza en el negocio la cual está estrechamente relacionada con

el tiempo y es objeto de análisis para la toma de decisiones. También puede verse como un valor numérico que representa una actividad específica casi siempre con cifras que se suman entre sí. La estructura que forman los hechos y las dimensiones puede verse como el plano o la visión desplegada de un cubo. Ver Figura 1.

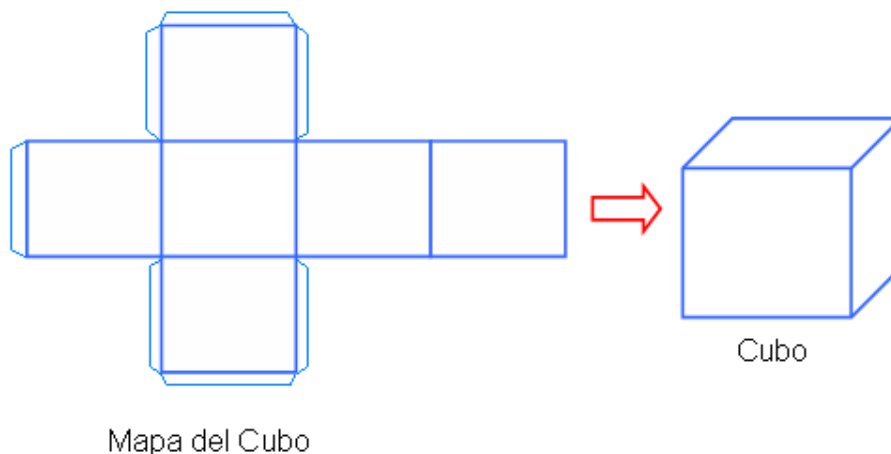


Figura 1: Cubo OLAP

La principal característica del modelo dimensional es su sencillez, permitiéndoles a los usuarios una fácil comprensión de las bases de datos, además de posibilitar al software un recorrido eficiente de sus estructuras.

#### 1.5.2.1. Componentes del Modelo Multidimensional

Para una correcta comprensión y construcción de un modelo multidimensional es necesario conocer cuáles son los elementos que lo componen y lo que significa cada uno de ellos.

##### Tablas de dimensiones

Las dimensiones organizan los datos en función de un área de interés para los usuarios. Cada dimensión describe un aspecto del negocio y proporciona el acceso intuitivo y simple a datos. Una dimensión provee al usuario de un gran número de combinaciones e intersecciones para analizar datos.

Las tablas de dimensiones son las compañeras de las tablas de hechos. Cada dimensión se define por su clave primaria que sirve para mantener la integridad referencial en la tabla de hechos con la que se relaciona. Un cubo requiere que se defina al menos una dimensión en su esquema.

Cada nivel de una dimensión debe corresponderse con una columna en la tabla de la dimensión. Los niveles se ordenan por grado de detalle y se organizan en una estructura jerárquica. Cada nivel contiene miembros, los miembros son los valores de la columna que define el nivel.



Todos los elementos que componen una dimensión están enmarcados en una determinada estructura jerárquica excepto los que aquellos que conformaran las propiedades de un determinado elemento.

### **Tablas de hechos**

Las tablas de hechos son las tablas primarias del modelo dimensional conteniendo los valores del negocio que se desea analizar así como cada una de las llaves de las dimensiones involucradas en el mismo. Todas estas columnas son valores numéricos calculables durante el proceso extracción, transformación y carga de datos.

### **Medidas o métricas**

El modelo dimensional divide el mundo de los datos en dos grandes tipos: las medidas y las dimensiones de estas medidas. Las medidas, siempre son numéricas, se almacenan en las tablas de hechos y las dimensiones que son textuales se almacenan en las tablas de dimensiones. Las medidas son los valores de datos que se analizan.

Una medida es una columna cuantitativa, numérica, en la tabla de hechos. Las medidas representan los valores que son analizados constituyendo:

- Valores que permiten analizar los hechos.
- Bases a partir de las cuales el usuario puede realizar cálculos.

Las medidas pueden clasificarse en:

- Naturales
- Calculadas

Medidas naturales: Son el resultado de la aplicación de operaciones matemáticas sencillas a un solo campo existente en la tabla de hechos. Cuando se define una medida se debe tener en cuenta cual será la forma de agregación (agrupación de la misma) al subir por la estructura dimensional.

Medidas calculadas: son el resultado de las diferentes operaciones que se pueden realizar con los valores de las medidas naturales. Debe tenerse en cuenta que estas medidas calculadas se pueden obtener durante el proceso ETL y después del mismo.

La decisión de cuando usar cada cual está en dependencia de los requisitos definidos por el cliente.

En sentido general la expresión medidas calculadas es muy amplio y engloba a cualquier manipulación de las medidas naturales que faciliten el análisis de los hechos, pero por lo general en una medida calculada puede haber:

- Cálculos Matemáticos
- Expresiones condicionales
- Alertas

Estos tres tipos (cálculos, condiciones y alertas) usualmente pueden existir juntos dentro de la misma medida calculada.(10)

#### **1.5.2.1.1. Estilos de diseño de las Bases de Datos Multidimensionales**

Para facilitar el análisis de la información y siendo consecuentes con los requerimientos de hardware y software existen tres estilos de diseño para los almacenes de datos. A continuación se presentan los mismos con sus principales características:

- Estilo de estrella
- Estilo copo de nieve
- Estilo constelación de hechos

##### **Estilo de estrella**

Esta estructura es considerada la más básica y sencilla de diseñar con respecto a las existentes. Está definida por una tabla central o tabla de hechos y un conjunto de tablas organizadas alrededor de ésta denominadas tablas de dimensiones, brindando la sensación de una estrella.

Estas tablas de dimensión contienen los atributos de las aperturas que interesan al negocio y se pueden utilizar como criterios de filtro, siendo relativamente pequeñas. En el estilo estrella cada dimensión está compuesta por una sola tabla, esta tabla se encuentra desnormalizada en su totalidad. Cada tabla de dimensión se vincula con la tabla de hechos por un identificador. Dado que las tablas de dimensión se encuentran desnormalizadas se logrará en el modelo del Data Mart, una menor cantidad de tablas. Ver Figura 2.

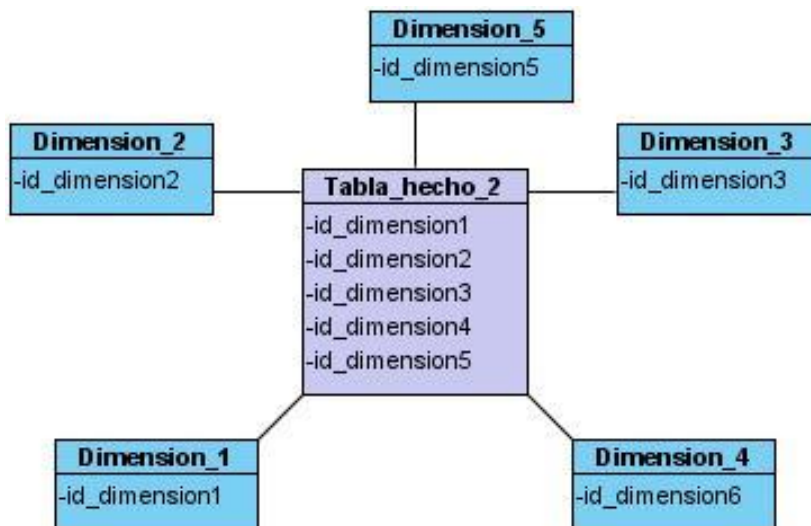


Figura 2: Estilo de estrella

Las principales características del estilo de estrella son:

- El centro de la estrella es la tabla de hechos.
- Los puntos de la estrella son las tablas de dimensiones.
- Cada esquema está compuesto por una sola tabla de hechos.
- Mayor facilidad de mantenimiento con relación a los otros estilos multidimensionales.

### Estilo copo de nieve

Este estilo es una variación del estilo de estrella en el que al menos una de sus puntas puede tener un cierto grado de normalización. El nombre del estilo está dado en que el mismo se asemeja a un copo de nieve.

En este estilo, algunas de las tablas de dimensión se encuentran normalizadas para eliminar redundancia de datos. A diferencia del estilo estrella, los datos de las dimensiones se reparten en múltiples tablas. Como ventaja principal de este estilo es el ahorro de espacio de almacenamiento en disco, pero en perjuicio de un aumento en la cantidad de tablas y disminución de rendimiento para la recuperación de información. Ver Figura 3.

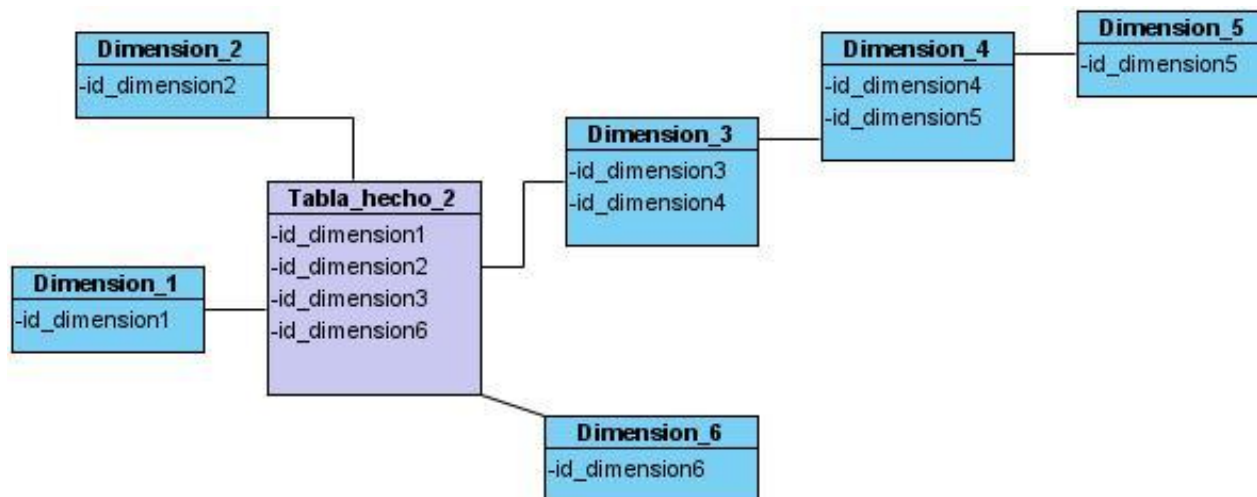


Figura 3: Estilo copo de nieve

Los siguientes son las características de un copo de nieve:

- Generalmente es un esquema parcialmente normalizado, aunque pudiera estarlo totalmente.
- Los distintos niveles se encuentran almacenados en tablas separadas.
- Se argumenta ahorro de espacio

### Estilo constelación de hechos

El concepto multiestrella, también conocido como estilo constelación de hechos, referencia las situaciones en que un único modelo multidimensional posee múltiples hechos, y por lo tanto, crea múltiples estilos estrellas, básicamente esta estructura es requerida cuando los hechos no comparten todas las dimensiones.(8)

Para cada estilo estrella o copo de nieve en almacén de datos es posible construir un esquema de constelación de hechos. Este esquema es más complejo que las otras arquitecturas debido a que contiene múltiples tablas de hechos. Con esta solución las tablas de dimensiones pueden estar compartidas para más de una tabla de hechos.

El estilo de constelación de hechos posee mucha flexibilidad y esta es su gran virtud; sin embargo, el problema es que cuando el número de las tablas vinculadas aumenta, la arquitectura puede llegar a ser muy compleja y difícil de mantener.

En un estilo de constelación de hechos, las distintas tablas de hechos están asignadas a las dimensiones relevantes para cada uno ellos. Esto puede ser útil cuando los hechos están asignados a un nivel de una dimensión y los otros hechos a otro nivel de detalle de otra dimensión. Ver Figura 4.

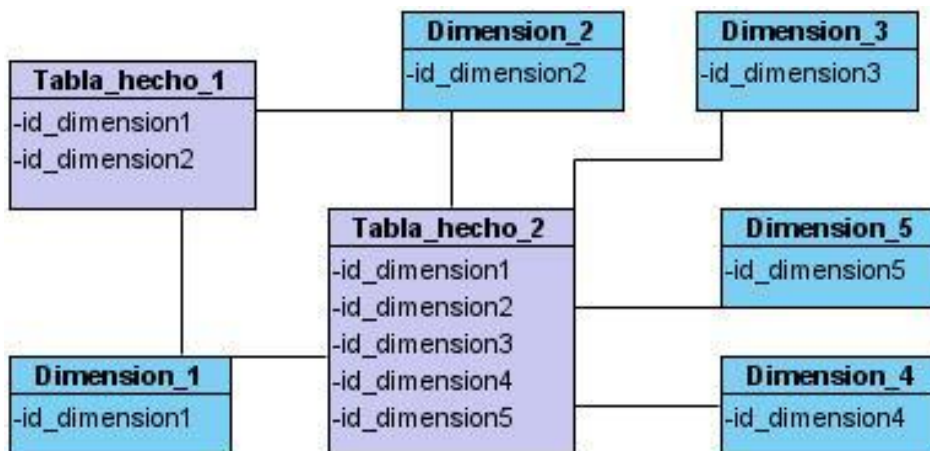


Figura 4: Estilo constelación de hechos

Independientemente del estilo, a nivel de dimensiones es posible definir jerarquías, las cuales son grupos de atributos que siguen un orden preestablecido. Una jerarquía implica una organización de niveles dentro de una dimensión, donde cada nivel representa el total agregado de los datos del nivel inferior. Una dimensión típica soporta una o más jerarquías naturales. (11)

#### 1.5.2.1. Ventajas del Modelo Multidimensional

- Por presentar una visión casi desnormalizada en su totalidad agiliza el proceso de recuperación de datos disminuyendo en gran medida el tiempo de respuesta del sistema.
- Agrupa los datos en pocas dimensiones, cada una de las cuales se convierte en una tabla en la BD.
- Se torna en un modelo simple consistente de pocas tablas.
- No es simétrico, pudiéndose determinar cual tabla es más importante.
- Se puede apreciar con facilidad qué/cuáles tablas del modelo contienen medidas numéricas.
- Son muy sencillas para las personas (usuarios finales o diseñadores) visualizar y conservar en sus mentes todas las tablas que componen el modelo.
- Para llegar a la información que el usuario desea obtener de una consulta donde se encuentren involucradas dos tablas solo existe un único camino de posibles conexiones entre ellas.
- Para consultas, que alcanzan muchos registros y muchas tablas, los modelos multidimensionales son muy fáciles de comprender por los usuarios y muy sencillos de recorrer por el software.

### 1.6. Estado actual del cálculo de los indicadores financieros

El análisis de los indicadores financieros es por lo general el primer paso de un análisis financiero. Se emplean para mostrar las relaciones que existen entre las cuentas de los estados financieros. Desde el punto de vista del inversionista sirve para la predicción del futuro, mientras que para la administración del negocio, es útil como una forma de anticipar las condiciones futuras y, como punto de partida para la planeación de aquellas operaciones que hayan de influir sobre el curso posterior de eventos (12).

El uso de estos indicadores proporciona al analista financiero una herramienta para obtener información de los estados financieros, señalando los puntos fuertes y débiles de un negocio.(13)

De forma general, no es posible establecer un conjunto estático de indicadores financieros, cada entidad emplea los que más considere de acuerdo a los análisis que desee realizar; por lo que se puede decir que si bien no hay un conjunto estático, existen indicadores básicos establecidos para conocer el estado de salud de la empresa y es en la interacción de estos donde se encuentra el aporte para las entidades.

Automatizar entonces el cálculo de estos indicadores ha sido el objetivo de muchas empresas creadoras de software para proveer a las organizaciones de herramientas potentes que se basen fundamentalmente en la posibilidad que posee el usuario de confeccionar diferentes estudios de las relaciones negocio-indicador financiero.

### **1.6.1. En el mundo**

En el mundo existen diferentes herramientas que tienen como objetivo brindar ayuda al proceso de toma de decisiones mediante el cálculo de indicadores financieros. Algunas de las más conocidas por su amplia utilización son:

#### **eAnaliza**

Es una herramienta del tipo BI diseñada para analizar información de una empresa o negocio de forma sencilla, mediante la combinación de datos y su presentación con gráficos profesionales, tablas dinámicas detalladas, estadísticas imprescindibles y cualquier soporte necesario para mantener y analizar los factores de negocio en una empresa, es totalmente personalizable, por lo que se adaptará a las características de cada empresa independientemente de su campo de acción.

Es desarrollada por la compañía Efebege, su licencia no es gratis, no es de código abierto, no posee opciones para las configuraciones de conexión a diferentes bases de datos; y es compatible solo con los sistemas operativos Win2000/XP/2003/Vista/7.

#### **Excel**

Es una de las herramientas más utilizadas actualmente para el cálculo de indicadores financieros.

Aplicación informática desarrollada por Microsoft, su licencia no es gratis, no es de código abierto. Su integración con otras aplicaciones está en relación con componentes específicos desarrollados para este propósito; y es compatible solo con los sistemas operativos Win2000/XP/2003/Vista/7.

### **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.

Permite analizar rápidamente datos y requisitos del modelo empresarial de toda la organización y utiliza los resultados para elaborar presupuestos y previsiones con la total confianza de obtener mejores resultados empresariales. Más que un software para la elaboración de presupuestos y previsiones, Cognos TM1 es una solución de planificación empresarial completa.

Es una aplicación web que posee un motor OLAP en memoria de 64 bits, multidimensional, ofrece un funcionamiento excepcionalmente rápido que permite analizar modelos complejos y sofisticados, grandes conjuntos de datos e incluso secuencias de datos.(14)

Posee además alcance para una cobertura de miles de usuarios, un enfoque potente, escalable y multiservidor en varios entornos como Windows, UNIX y Linux. Es necesario pagar su licencia para poder usarlo, no se permite el acceso a su código fuente en caso de requerir modificación alguna. Brinda soporte para bases de datos en Microsoft SQL 2005, SAP Oracle 10g/11g e IBM DB2 9.5 y 9.1.

Por otra parte algunos de los ERP más difundidos en el mundo implementan diversos mecanismos para realizar su gestión financiera, entre estos se encuentran:

### **SAP**

Software desarrollado en la Ciudad de Mannheim, Alemania, por antiguos empleados de IBM. Posibilita configurar cada uno de los procesos según las necesidades de las empresas con el fin de obtener una ventaja competitiva en su sector. SAP ERP está diseñado para incrementar la eficiencia de la planificación y la gestión de procesos a lo largo de las empresas.(15)

SAP dentro de su gama de herramientas utiliza SAP Business One para obtener los indicadores financieros de las entidades. Esta aplicación ofrece un completo e integrado conjunto de herramientas para dirigir y automatizar eficazmente todos los procesos contables y financieros en una compañía. Como núcleo de SAP Business One, la administración financiera incorpora todos los procesos financieros principales tales como el libro mayor, las entradas, las cuentas por pagar y cuentas por cobrar. El software automáticamente registra en tiempo real el estado de las cuentas cuando ocurren las transacciones.(16)

Posibilita además crear varios reportes financieros incluyendo pérdidas y ganancias, flujo de caja, balance, entre otros.

Se caracteriza por ser un sistema totalmente preparado para trabajar con él mediante la web debido a la nueva plataforma tecnológica denominada SAP NetWeaver<sup>1</sup>. Implementado en .NET, funciona sobre el sistema operativo Windows y solo presenta soporte para bases de datos Oracle. (16)

### **ORACLE**

Dentro del paquete de sistemas generales de las aplicaciones de Oracle (Oracle Applications), Oracle Corporation E-Business Suite (también conocida como Aplicaciones o EB-Suite/EBS), consiste en una colección de herramientas de planificación de recursos empresariales (ERP), gestión de relaciones con clientes (CRM), y gestión de la cadena de suministro (SCM), así como de aplicaciones informáticas desarrolladas y/o adquirida por Oracle. El E-Business Suite (la versión actual: 12.1) contiene varias líneas de productos, es considerada la suite de aplicaciones de negocio más completa del mercado. Ya ponga en marcha un solo módulo o el conjunto completo.

Una de las herramientas empleadas por e-Business Suite lo constituye GL Wand. Este software integra en Excel la información financiera del módulo GL con el sistema Oracle e-Business Suite. GL Wand instala una nueva barra de herramientas en Excel desde donde se accede en directo a toda la información financiera del sistema Oracle e-Business Suite, integrando sus reportes en tiempo real. (17)

Otro de los módulos incluidos en Oracle Business Intelligence Suite Enterprise Edition que posibilita también el análisis de los indicadores financieros es Hyperion Financial Reporting. Esta herramienta convierte los datos de la empresa en conocimiento operacional creando informes financieros en una amplia variedad de formatos. Para producir dichos informes Hyperion Financial Reporting, accede a una extensa gama de fuentes de datos multidimensionales. Esta forma de generar los informes cumple con todas las regulaciones y requisitos externos, posibilitando que puedan ser entregados de inmediato o programados para su posterior distribución, mejorando la planificación financiera de la organización, su gestión y control.(17)

Ambas herramientas se caracterizan por ser privativas siendo su código fuente cerrado, y utilizar el gestor de bases de datos Oracle. Por otro lado, el costo de su licencia es independiente para cada producto y de elevado precio.

### **OPEN-ERP**

---

<sup>1</sup> SAP NetWeaver es una plataforma conformada por varias herramientas enfocadas a optimizar y sincronizar los recursos informáticos con los requisitos particulares, a las estrategias y aplicaciones.



Software desarrollado en Argentina. OpenERP se describe a sí mismo como el ERP de código abierto más destacado y sencillo que existe hasta el momento. Es una de las aplicaciones que gestiona de manera eficiente los datos económicos basándose en los registros contables de las empresas, soporta múltiples monedas y múltiples entidades. El módulo contable que posee brinda las funcionalidades relacionadas con la contabilidad general, analítica y presupuestaria, permite además llevar los libros contables en forma rigurosa.

Se caracteriza por ser un sistema con componentes separados en un esquema Cliente-Servidor. Es multiplataforma; funciona sobre Linux y Windows. Implementado en el lenguaje Python, dispone de interfaces XML-RPC<sup>2</sup>, y SOAP<sup>3</sup>, presenta como soporte para bases de datos PostgreSQL, liberado bajo la GPL<sup>4</sup>. Puede ser utilizado como un programa independiente o integrado. (18) Aunque es válido señalar que no posee ninguna funcionalidad para el cálculo de los indicadores financieros.

### **OPEN-BRAVO**

Software desarrollado por la Universidad de Navarra en España en el año 2005, destinado principalmente para las pequeñas y medianas empresas. Se caracteriza por ser una aplicación completamente web que ha sido desarrollada siguiendo el modelo-vista-controlador (MVC en inglés Model-View-Controller) que facilita el desacoplamiento de las áreas de desarrollo. Implementada en el lenguaje Java, presenta soporte para bases de datos PostgreSQL y Oracle, se ejecuta sobre Apache y Tomcat. La estructura de datos de la aplicación está basada originalmente en una versión antigua de Compiere<sup>5</sup>. Una de las principales ventajas con las que cuenta este ERP es que sigue un licenciamiento de software libre asegurando el acceso público al código fuente y la posibilidad de modificar dicho código libremente, adaptándolo a las necesidades de cualquier empresa. (19) Este sistema aunque es uno de los más usados no posee funcionalidades para la obtención de indicadores financieros.

### **1.6.2. En Cuba**

#### **Versat Sarasola**

Software desarrollado por la entidad cubana TEICO de Villa Clara, constituye un sistema que automatiza prácticamente todas las actividades de Planificación, Control y Análisis Económico de cualquier tipo de

---

<sup>2</sup> XML-RPC es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.

<sup>3</sup> SOAP es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML

<sup>4</sup> GPL: Licencia Pública General de GNU.

<sup>5</sup> Compiere5 es una aplicación para negocios de código abierto, ERP y CRM destinada para las empresas de pequeño y mediano tamaño.

Entidad, ya que es configurable. Presenta dos variantes para su instalación: la variante típica que incluye todos los subsistemas del VERSAT-Sarasola y la variante personalizada en la que solo estarán los subsistemas seleccionados por el usuario.

Permite llevar el control y el registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades y obtener los Estados Financieros y Análisis Económicos y Financieros en estos niveles. Aparece el Subsistema de Contabilidad General como rector del sistema que maneja el resto de los subsistemas.

Se caracteriza por ser una aplicación de escritorio. Concebido sobre una plataforma de trabajo Cliente-Servidor lo que permite además su instalación en red por las posibilidades que esta tecnología facilita para el trabajo en un entorno multiusuario. Implementado en Delphi. Trabaja sobre el sistema operativo Windows y presenta como soporte para bases de datos SQL Server 2000.(20) No posee dentro de sus funcionalidades alguna destinada a la obtención de indicadores financieros.

### **Siscont5**

Sistema cubano creado por la empresa Tecnomática en el año 2007, el cual se aviene a las definiciones y conceptos del Ministerio de la Industria Básica (MINBAS) aunque por las acciones contables financieras que permite puede ser utilizado en otras entidades nacionales.

SISCONT5 puede ser explotado en régimen mono usuario y multiusuario, mono entidad y multientidad; para esta última existe el control de su acceso para las entidades en un mismo equipo de cómputo como servidor.

Presenta solo soporte para bases de datos SQL Server 2000. Trabaja sobre el sistema operativo Windows. Fue implementado en la herramienta de desarrollo de software basada en conocimiento GeneXus<sup>6</sup>. Este sistema no posee funcionalidades para obtener indicadores financieros.

### **Assets NS**

ASSETS NS es un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos. Como Sistema Integral todos sus módulos trabajan en estrecha relación. Dispone, además, de métodos novedosos para administración y planificación de inventarios, así como una amplia gama de análisis y consultas que le permitirán no solo conocer exactamente la situación actual, sino proyectar decisiones futuras.

---

<sup>6</sup> GeneXus es una herramienta de desarrollo de software, orientada principalmente a aplicaciones de clase empresarial para la web y plataforma Windows.

Es un sistema flexible, amigable, con ayuda en línea, que puede ser instalado en una microcomputadora o sobre varias, funcionando en ambiente multiusuario incluidas estaciones remotas. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario y se encuentra diseñado para multientidad. (21)

Es una aplicación de escritorio, cliente-servidor, programada en Visual Basic 6.0, con soporte para Microsoft SQL Server 2000, (21) posibilita la obtención de los indicadores financieros, utilizando adicionalmente Crystal Reports 7.0 para la generación y visualización de reportes.(21)

### **1.6.3. Valoración del estudio del estado del arte**

Después de realizar un estudio de las herramientas antes mencionadas donde se tuvieron en cuenta aspectos relacionados no solo a los requisitos no funcionales del software sino además con el producto, y con la necesidad de estar acorde con lo ya desarrollado en Cedrux, se puede decir que las herramientas no cumplen a cabalidad con todos los requerimientos, algunas no son de código abierto, es necesario además pagar sus licencias, otras son aplicaciones de escritorio, y las que son libres no dan la posibilidad de calcular los indicadores financieros. Ver Tabla 2.

Por las razones antes planteadas se recomienda realizar una aplicación que cumpla con todos los parámetros analizados y posibilite una mejor toma de decisiones a partir del cálculo de los indicadores financieros.

	<b>Gratuito</b>	<b>Código fuente abierto</b>	<b>Multiplataforma</b>	<b>Web</b>	<b>Cálculo de indicadores financieras</b>
<b>eAnaliza</b>	0	0	0	0	1
<b>Excel</b>	0	0	0	0	1
<b>Sap BO</b>	0	0	0	1	1
<b>Siscont5</b>	0	0	0	0	1
<b>Asset NS</b>	0	0	0	0	1
<b>IBM</b>	0	0	1	1	1

<b>Cognos TM1</b>					
<b>ORACLE GL Wand</b>	0	0	0	0	1
<b>ORACLE Hyperion Financial Reporting</b>	0	0	0	1	1
<b>OPEN- ERP</b>	1	1	1	1	0
<b>OPEN- BRAVO</b>	1	1	1	1	0
<b>Versat Sarasola</b>	0	0	0	1	0

Tabla 2: Tabla comparativa entre herramientas y sistemas ERP que gestionan análisis financiero (1 afirmativo/ 0 negativo)

### 1.7. Metodologías estudiadas para la construcción del Data Mart

Según el diccionario online de la Real Academia Española, la palabra metodología viene de *método*, y *-logía* de ciencia: o sea, es la ciencia del método, la ciencia de decir o hacer con orden.

Para la construcción de un DW es necesario conocer las dos variantes de las metodologías, la primera para el diseño y la segunda para su implementación.

#### 1.7.1. Metodología para el diseño o de arquitectura

Esta metodología está basada fundamentalmente en cuál será la estructura o el diseño del DW. Dentro de sus principales exponentes podemos encontrar a Ralph Kimball y Bill Inmon los que plantean diferentes puntos de vistas para el diseño de los mismos. De estas metodologías no se puede determinar cuál es mejor o no, su selección para utilizarla está dada en las características del proyecto a desarrollar.

#### Metodología de Ralph Kimball

Según Kimball: “El Data Warehouse es la unión de todos los Data Marts de una entidad”, siendo además

una copia de los datos transaccionales, estructurados de una forma especial para realizar su análisis, de acuerdo al modelo dimensional no normalizado. Este enfoque también es conocido como Bottom-up. Esta característica le hace más flexible y sencillo de implementar, pues se puede construir un Data Mart como primer elemento del sistema de análisis, y luego ir añadiendo otros que compartan las dimensiones ya definidas o incluyan otras nuevas. En este sistema, los procesos ETL extraen la información de los sistemas operacionales y los procesan realizando posteriormente el llenado de cada uno de los Data Mart de una forma individual, aunque siempre respetando la estandarización de las dimensiones. Este enfoque es eficaz y conduce a una solución completa en un corto período de tiempo. Ver Figura 5.

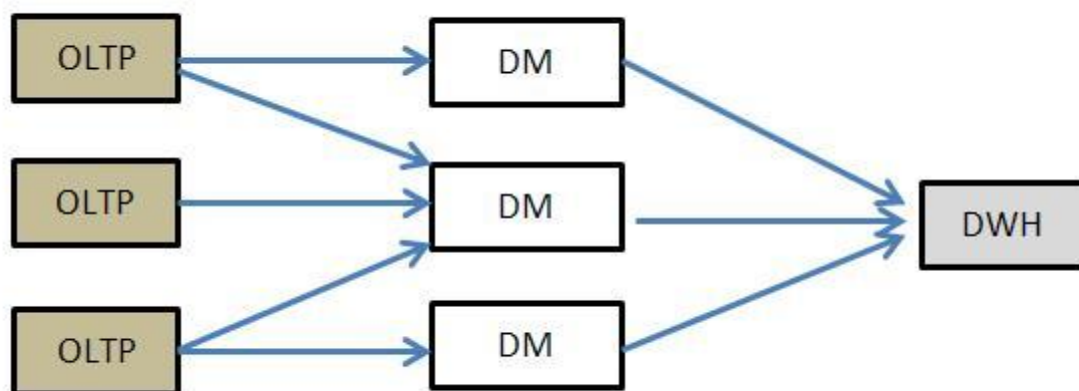


Figura 5: Enfoque Kimball

### Metodología de Bill Inmon

Inmon ve la necesidad de transferir la información de los diferentes OLTP de las organizaciones a un DW centralizado, donde los datos puedan ser utilizados para su posterior análisis. La información ha de estar a los máximos niveles de detalle. Los DW departamentales o Data Marts son tratados como subconjuntos de este DW corporativo, y son construidos para cubrir las necesidades individuales de análisis de cada departamento, siempre a partir de este DW central.

El enfoque Inmon también se referencia normalmente como **Top-down**. Los datos son extraídos de los sistemas operacionales por los procesos ETL donde son validados y consolidados para su posterior almacenamiento en el DW, donde además existen los llamados metadatos que documentan de una forma clara y precisa el contenido del DW. Una vez realizado este proceso, los procesos de actualización de los Data Mart departamentales obtienen la información de él, y con las consiguientes transformaciones, organizan los datos en las estructuras particulares requeridas por cada uno de ellos. Al tener este enfoque

global, es más difícil de aplicar en un proyecto sencillo (pues se intenta abordar el “todo”, a partir del cual luego se irá al “detalle”). Ver figura 6.

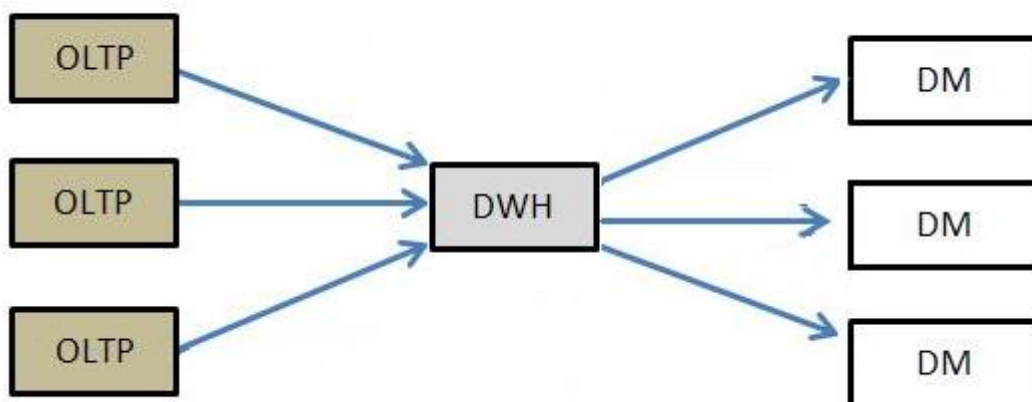


Figura 6: Enfoque Inmon

### 1.7.2. Metodologías para el desarrollo

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.(22)

La idea principal, es comprender cada paso que se realizará, para no caer en el tedio de tener que seguir un método al pie de la letra sin saber exactamente qué se está haciendo, ni por qué.

Existen varias metodologías empleadas en la construcción de un almacén de datos, no es posible decir cuál es la mejor o si alguna es mejor que otra, su selección está en dependencia de las características del negocio y de la organización para la cual se va a construir, entre todas, las seleccionadas para el estudio son las siguientes: Metodología de Hefesto, DATEC, UCID.

#### Metodología de Hefesto

Hefesto es una metodología propia, cuya propuesta está fundamentada en una muy amplia investigación, comparación de metodologías existentes, y experiencias propias en procesos de confección de almacenes

de datos. Es una de las más difundidas y utilizadas por su fácil implementación y aporte práctico, aunque no propone de forma explícita los artefactos y entregables a generar en cada fase.

Esta metodología cuenta con las siguientes características:

- 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.
- 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 el desarrollo de Data Marts como para DWs [12].

### **Metodología de DATEC**

Esta metodología fue creada en el año 2009 por el centro CENTALAD en 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 de 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 RUP 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 para cada una de sus fases, estas son las siguientes:

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

- Fase de despliegue
- Fase de soporte

### **Metodología UCID**

Esta metodología está basada fundamentalmente en los principios planteados en la metodología Hefesto, presentando las mismas 4 fases de desarrollo, definiendo nuevos artefactos y detallando de forma más específica los planteados por la propia metodología. Cumple con las mismas características fundamentales de la metodología Hefesto.

Las fases y artefactos generados para cada una de ellas son los siguientes.

#### **1. Análisis de los requerimientos:**

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

#### **2. 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

#### **3. 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

#### **4. Proceso ETL**

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

#### **1.7.3. Justificación de las metodologías a utilizar**

En cuanto a las metodologías de arquitectura se puede decir que el enfoque Inmon es más apropiado para sistemas complejos, en los que se quiera además asegurar su perdurabilidad y consistencia aunque cambien los procesos de negocio en la organización. Aunque, para pequeños proyectos, en los que además se haga necesario asegurar la usabilidad de los usuarios con un sistema fácil de entender y garantizar un rápido desarrollo en la solución, el enfoque Kimball es más apropiado, siendo el utilizado para el caso que ocupa esta investigación en la que se implementarán los procesos de negocios para el cálculo de indicadores financieros específicamente para el modulo de Contabilidad y Costos y Procesos del producto Cedrux.

En relación a las metodologías de desarrollo, si bien la metodología de DATEC y Hefesto posibilitan organizar de forma eficaz el proceso de creación de los DW; el empleo de la primera implica la creación de gran cantidad de artefactos para la documentación haciendo que el proceso de implementación del Data Mart se torne engorroso y tedioso, y la segunda, si bien por su aporte práctico es fácil de entender e implementar, no define de manera concisa los artefactos a generar en cada fase. Por todas las razones antes planteadas se selecciona la metodología del UCID para el desarrollo de este trabajo de diploma, partiendo de su basamento en la metodología Hefesto, y en la creación así como definición de artefactos y entregables necesarios, suficientes y con un gran sentido práctico para la implementación.

#### **1.8. Herramientas para la construcción de un Data Mart**

Las herramientas para la construcción del Data Mart se dividieron en 5 grupos principales, los relacionados con los sistemas de bases de datos, herramientas de integración de datos, herramientas para diseñar cubos OLAP, herramientas CASE y servidor de aplicaciones, realizando un estudio basados en soberanía tecnológica, usabilidad, así como documentación sobre el uso de las mismas.

### **1.8.1. Sistemas gestores de Bases de Datos**

#### **ORACLE**

El gestor de bases de datos Oracle fue creado a finales de la década de los 70 y principios de los 80 por la corporación Oracle. Es un Sistema Gestor de Bases de Datos (SGBD) con características objeto-relacionales. Dentro de sus características principales se destacan:

- 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.
- Replicación de entornos.
- Autogestión de la integridad de los datos.
- Opción distribuida.
- Portabilidad.
- Compatibilidad.(17)

Otras características presentadas por este potente SGBD garantizando su total fortaleza en el mercado internacional son el permitir trabajar sobre múltiples plataformas, dígame Linux, XP, UNIX, entre otras; aunque a su vez posee otra característica clasificada como una desventaja del mismo, su licencia privativa, siendo además elevado su costo de adquisición.

#### **MYSQL (23)**

El gestor de bases de datos MYSQL fue desarrollado por Sun Microsystems actualmente MySQL AB. MYSQL es un gestor de bases de datos libre desarrollado por la compañía sueca Sun Microsystems. Mantiene el copyright del código fuente del servidor así como también el de la marca. Este gestor de

bases de datos posee gran rapidez y facilidad de uso, además de ser multiplataforma. Algunas de sus características principales son:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

Este gestor presenta algunas desventajas dentro de las que se encuentran su bajo rendimiento con grandes volúmenes de datos, no permitir dar privilegios de acceso en grupos, ni realizar bloqueos a nivel de campo, así como no brinda la posibilidad de crear nuevos tipos de datos, no posee características de orientación a objetos, no permite realizar respaldos incrementales y no soporta identificadores de fila que son útiles para actualizaciones rápidas.(23)

### **PostgresSQL**

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto bajo licencia GPL. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad y corrección de datos. Se ejecuta en todos los principales sistemas operativos, como son Linux, UNIX, Mac OS X y Windows. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados. Incluye la mayoría de las sentencias SQL. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o videos. Cuenta con interfaces nativas de programación para C/C++, Java, .Net, Python, Ruby, entre otros y la documentación excepcional.(24) Algunas de sus características principales son:

- Instalación ilimitada.
- Mejor soporte que los proveedores comerciales.
- Ahorros considerables en costos de operación.
- Estabilidad y confiabilidad.
- Extensible.
- Multiplataforma.

- Herramientas gráficas de diseño y administración de bases de datos.(24)

Este gestor posee algunas ventajas con relación al gestor MYSQL entre las que se encuentran:

- Poseer un buen diseño para ambientes de alto volumen de datos usando una estrategia específica de almacenamiento de filas para conseguir una mejor respuesta en ambientes de grandes volúmenes.
- Brinda la posibilidad de gestionar permisos a grupos de usuarios.
- Controla la concurrencia y la realización de bloqueo a nivel de campo.
- Posee un sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario y rápido desarrollo de nuevos tipos.
- Posee características de orientación a objetos.
- Utiliza identificadores de filas (oid), para realizar actualizaciones de forma más rápida.

### 1.8.2. Herramientas de integración de datos

#### **CloverETL**

Es un ambiente de transformación de datos de código abierto basado en Java utilizado para entre otras funciones en el movimiento de datos entre cualquier número de diferentes sistemas y bases de datos. Es altamente configurable, muy rápido y ahorra mucho tiempo y esfuerzo no sólo en el desarrollo de las transformaciones de datos y las migraciones, sino también para el mantenimiento continuo. Este proceso se puede realizar mediante líneas de comandos o a través de una aplicación Java. Permite la combinación, transformación y circulación de los datos de cualquier origen.

CloverETL es una plataforma de integración de datos de alto rendimiento que le permite mover datos entre diferentes ubicaciones. Los datos pueden ser extraídos de cualquier número de fuentes, validados y modificados a lo largo de la migración, y luego ser escritos a uno o más destinos.

- Independiente de la plataforma - totalmente escrito en Java.
- Implementación rápida - sólo se tiene que descargar, instalar y trabajar.
- Flexible y escalable - Varias ediciones desde gratis hasta edición "Enterprise".
- Fácilmente extensible - API simple.

#### **EnhydraOctopus**

Esta herramienta es conocida también como Octopus, es una herramienta de ETL para transformaciones de datos por conexión JDBC. 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, entre otros. 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.

### **KETTLE**

Es desarrollado en Estados Unidos bajo la licencia GNU Lesser General Public License Version 2.1 (LGPLv2.1) desarrollado en Java. Utiliza como entorno gráfico la herramienta de diseño (Spoon) basada en SWT<sup>7</sup>, el entorno para su ejecución es desde la herramienta de diseño, o desde línea de comandos con las utilidades Pan y Kitchen. Posee un conjunto reducido de componentes, pero muy orientados a integración de datos. Las plataformas sobre las cuales se puede utilizar son Windows, Unix y Linux.

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

Puntos fuertes:

- Es un motor de transformación donde es más fácil gestionar los tipos de datos.
- La herramienta es muy intuitiva, y con unos conceptos básicos es posible realizar varias tareas de mediana complejidad.

---

<sup>7</sup> Conjunto de herramientas flash de código fuente abierto, para Java, diseñado para proporcionar un acceso eficiente y portátil para la interfaz de usuario de los sistemas operativos en los que se aplica.

- Conceptualmente muy sencilla y potente.
- El repositorio en base de datos brinda muchas posibilidades para el trabajo en equipo.
- El proceso en paralelo es muy sencillo realizar.
- Ejecución bien desde la herramienta (tiempos de respuesta bastante buenos) (25)

### Puntos débiles:

- El diseño de la interfaz puede resultar un poco pobre, y no hay una interfaz unificada en todos los componentes, siendo en ocasiones la interfaz de los componentes confusa.
- Evolución mucho más lenta de la herramienta e incierta, pues Pentaho no le da seguimiento completamente a la versión libre.
- Disponibilidad de componentes más limitada, aunque más que suficiente para la mayoría de procesos ETL o de integración de datos.
- Presenta una ayuda muy pobre, casi inexistente en la aplicación. La ayuda online en la web de Pentaho no es especialmente completa.
- Contiene una sencilla herramienta de debug la cual es muy básica.
- La generación de nuevas versiones no es muy frecuente.
- Algunos componentes no se han comportado de la forma esperada, al realizar transformaciones muy complejas o al enlazar llamadas entre diferentes transformaciones a Jobs.
- El código escrito en JavaScript dentro de los componentes no se puede reutilizar en otros componentes. Bastante limitado para añadir nuevas funcionalidades o modificar las existentes.
- Ejecución bien desde la herramienta (en ocasiones los tiempos de respuesta son lentos, sobre todo si se incluye estadísticas y trazas de ejecución).(25)

### Talend

Es desarrollado en Francia bajo la licencia GNU Lesser General Public License Version 2.1 (LGPLv2.1) desarrollado en Java. Utiliza como entorno gráfico la herramienta gráfica basada en Eclipse, el entorno para su ejecución es desde la herramienta de diseño, o a nivel de línea de comandos con Java o Perl (independiente de la herramienta). Posee una gran cantidad de componentes, el enfoque es tener un componente distinto según la acción a realizar. Las plataformas sobre las cuales se puede utilizar son Windows, Unix y Linux.

Puntos fuertes:

- El diseño de la interfaz de usuario es unificada en todos los componentes. Basada en Eclipse, lo que facilita el uso de la interfaz basado en el conocimiento acerca de la misma.
- Evolución muy rápida de la herramienta.
- Gran disponibilidad de componentes para conectarse a múltiples sistemas y orígenes de datos, y en continua evolución.
- Posee una gran ayuda contextual en la aplicación sumándosele una completa ayuda online de los componentes.
- Contiene una perspectiva Debug de Eclipse, en la que es posible seguir la traza de la ejecución (viendo el código fuente) tal y como si se estuviera programando en Eclipse.
- Continua generación de versiones, incorporando mejoras y arreglo de Bugs.
- Es posible realizar la reutilización de código al incluir sus propias librerías lo que permite diseñar sus propios componentes.(25)

Puntos débiles:

- Es un generador de código, y este aspecto hace que tenga una gran dependencia del lenguaje elegido en los proyectos.
- Es una herramienta poco intuitiva y difícil de entender.
- No se dispone de un repositorio en base de datos para el trabajo en equipo.
- El proceso en paralelo es muy reducido en las versiones libres.
- En ocasiones se nota en demasía la lentitud ocasionada por el uso de Java.(25)

### 1.8.3. Herramientas para diseñar cubos 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. (25)

#### **Mondrian**

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

Características de este sistema:

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

Schema workbench (kettle) es un entorno visual para el desarrollo y prueba de cubos OLAP. 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 MDX<sup>8</sup> contra el esquema y la base de datos.

Después de haber analizado las herramientas para el diseño de los cubos, 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 así como la amplia documentación que ofrece su comunidad de desarrollo.

### 1.8.4. 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. Pertenece a los software de modelado, ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, además de tener funcionalidades como generar código inverso, código desde diagramas y documentación. Entre sus principales características se pueden encontrar:

- Soporta aplicaciones Web.
- Es un producto de calidad.

---

<sup>8</sup>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.



- Fácil de instalar y actualizar.
- Posee versiones para múltiples plataformas.
- Compatibilidad entre ediciones.

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

#### **1.8.5. Servidor de aplicaciones**

##### **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, 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, seleccionándolo como servidor de aplicaciones para la construcción del DM permitiendo ser descargado, utilizado, y distribuido sin restricciones por la licencia.

#### **1.9. Conclusiones del capítulo**

La obtención de los indicadores financieros constituye un aspecto fundamental en la transformación de la información en conocimiento para enriquecer el proceso de toma de decisiones en las entidades cubanas, este capítulo fue esencial para valorar el estado actual de la obtención de lo mismos evidenciando que:

- Algunos de los sistemas analizados no realizan el cálculo de los indicadores financieros y los que lo efectúan no cumplen con todos los parámetros para su utilización en el sistema Cedrux.
- Se hace necesario construir un Data Mart para realizar el cálculo de los indicadores financieros en los módulos Contabilidad y Costos y Procesos del proyecto ERP.
- La metodología de arquitectura seleccionada es la correspondiente con el enfoque de Ralph Kimball por adecuarse a las características del negocio y las ventajas que presenta la misma.
- La metodología de desarrollo seleccionada es la creada por el UCID por su aplicación sencilla y la definición de los artefactos necesarios para la implementación del DM.
- Después del estudio de las diferentes herramientas utilizadas para la construcción del Data Mart se seleccionaron por las características del Cedrux: Como sistema gestor de base de datos al PostgreSQL, para el proceso de extracción, transformación y carga de los datos a la herramienta Kettle perteneciente a la suite del Pentaho, como herramienta para el diseño del

cubo OLAP al workbench, el Visual Paradigm como herramienta CASE y como servidor web al Apache Tomcat interno del Pentaho (BI).

## **CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL DATA MART**

### **Introducción**

En el presente capítulo se realizará el análisis, diseño e implementación del Data Mart, en correspondencia con las etapas del proceso de desarrollo de software planteado en la metodología seleccionada 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 y medidas, así como la definición de procesos de extracción, transformación y carga de datos, concluyendo con la visualización de la información almacenada en el mismo.

### **2.1. Análisis de los requerimientos**

En esta fase la idea central es que se formulen preguntas complejas sobre el negocio, que incluyan variables de análisis que se consideren relevantes, ya que son estas las que permitirán estudiar la información desde diferentes perspectivas.

Para esto se realizaron 4 entrevistas a fin de identificar cuáles serán los indicadores y perspectivas que serán tomadas en cuenta para la construcción del DM. Se confeccionará un modelo conceptual en donde se podrá visualizar el resultado obtenido de la entrevista concluyendo así esta primera fase.

#### **2.1.1. Cronograma de entrevistas**

El objetivo principal de este paso, es obtener e identificar las necesidades de información claves de alto nivel, se tuvo en cuenta que dicha información es la que proveerá el soporte para desarrollar los pasos sucesivos, prestando especial atención a los datos. Esta información se obtuvo mediante la realización de varias entrevistas al cliente. A continuación se muestra el cronograma de entrevistas llevado a cabo para la realización del Almacén de Datos y el análisis multidimensional de la información referente a los subsistemas de Contabilidad y Costos y Procesos.

<b>No.</b>	<b>Fecha</b>	<b>Hora</b>	<b>Lugar</b>	<b>Objetivo</b>	<b>Responsable</b>	<b>Aseguramientos</b>	<b>Participantes</b>
------------	--------------	-------------	--------------	-----------------	--------------------	-----------------------	----------------------

1	15/9/2010	8:30 am	Doc 1	Conocer el negocio del subsistema de Contabilidad y Costos y Procesos.	Liván Valdés Yero	Personal	Est. Liván Valdés Yero Esp. Funcional María Elena Ing. Annilié Manresa
2	1/10/2010	8:30 am	Doc 1	Identificar las perspectivas e indicadores para los subsistemas de Contabilidad y Costos y Procesos.	Liván Valdés Yero	Personal	Est. Liván Valdés Yero Esp. Funcional María Elena Ing. Annilié Manresa
3	29/10/2010	8:30 am	Doc 1	Definir el grado de granularidad de cada una de las perspectivas para los subsistemas de Contabilidad y Costos y Procesos.	Liván Valdés Yero	Personal	Est. Liván Valdés Yero Esp. Funcional María Elena Ing. Annilié Manresa
4	18/11/2010	8:30 am	Doc 1	Definir como se calculan los indicadores financieros para los subsistemas de Contabilidad y Costos y Procesos.	Liván Valdés Yero	Personal	Est. Liván Valdés Yero Esp. Funcional María Elena Ing. Annilié Manresa

Tabla 3: Cronograma de entrevistas realizadas

### 2.1.2. Planificar entrevistas

A continuación se mostrarán los resultados de las entrevistas realizadas a los clientes para obtener la información referente a los conceptos relacionados con el negocio de las entidades, así como la identificación de los indicadores y medidas a calcular, teniendo en cuenta la definición de las reglas establecidas en el negocio.

### **Entrevista 1**

La correspondiente entrevista fue realizada con el objetivo de conocer el negocio de los subsistemas de Contabilidad y Costos y Procesos. En la misma estuvieron presentes los especialistas de los respectivos subsistemas en conjunto con el equipo de desarrollo. Esta entrevista cumplió con el objetivo propuesto logrando que el equipo de desarrollo entendiera el negocio de los subsistemas de Contabilidad y Costos y Procesos garantizando el punto de partida para su posterior desarrollo.

### **Entrevista 2**

La entrevista fue realizada con el objetivo de identificar las perspectivas e indicadores para los subsistemas de Contabilidad y Costos y Procesos. En la misma estuvieron presentes los especialistas de los respectivos subsistemas en conjunto con el equipo de desarrollo. Esta entrevista cumplió con el objetivo propuesto en el que el equipo de trabajo y los especialistas llegaron a un acuerdo en cuanto a las principales funcionalidades para con la aplicación a desarrollar. Una vez cumplida esta entrevista el equipo de desarrollo ya puede comenzar a trabajar en la confección del modelo conceptual de los hechos “cub\_hecho\_contabilidad\_cuenta” y “cub\_hecho\_contabilidadind” con perspectivas e indicadores identificados.

### **Entrevista 3**

La entrevista fue realizada con el objetivo de definir el grado de granularidad de cada una de las perspectivas para los subsistemas de Contabilidad y Costos y Procesos. En ésta participaron los especialistas de los respectivos subsistemas en conjunto con el equipo de desarrollo. La entrevista culminó con la detallada determinación del grado de granularidad de cada una de las perspectivas. Finalizada esta entrevista el equipo de desarrollo podrá proceder a la realización del modelo conceptual ampliado de los hechos “cub\_hecho\_contabilidad\_cuenta” y “cub\_hecho\_contabilidadind” con perspectivas e indicadores identificados.

### **Entrevista 4**

La entrevista fue realizada con el objetivo de definir cómo se calculaban los indicadores financieros para los subsistemas de Contabilidad y Costos y Procesos. En ésta participaron los especialistas de los respectivos subsistemas en conjunto con el equipo de desarrollo. La entrevista dio cumplimiento al objetivo propuesto en la misma a través de la obtención de las fórmulas para la realización de sus correspondientes cálculos. Una vez culminada esta entrevista el equipo de desarrollo se encuentra en condición de comenzar la programación de los trabajos para realizar el cálculo de los mismos.

### **2.1.3. Identificar perspectivas e indicadores**

Para una correcta implementación del DM es necesario realizar un detallado estudio de las perspectivas e indicadores necesarios para el estudio. Se puede definir como una perspectiva a la entidad de negocios respecto de la cual se deben calcular las métricas, es decir los filtros de análisis de la información y como indicador o métrica como el valor cuantitativo que almacenan las métricas del negocio. Estas métricas se encuentran representadas por columnas numéricas en las diferentes tablas de hechos en dependencia del negocio planteado. Los datos de estas medidas provienen directamente de los sistemas OLTP.

#### **Perspectivas**

➤ **Estructura**

Esta perspectiva representa las estructuras físicas que en algún momento podrán ser objeto de análisis. Estas estructuras constituyen las divisiones físicas establecidas a nivel de país para agrupar a las distintas entidades contables ya sean particulares o presupuestadas.

➤ **Cuentas**

Esta perspectiva representa las cuentas económicas que en algún momento podrán ser objeto de análisis. Dígase cuenta al término establecido como depósito de saldo contable con un número único y perteneciente a un solo ciudadano legal.

➤ **Centros**

Esta perspectiva representa los centros de costos que en algún momento pueden ser objeto de análisis. Constituye una división ya sea lógica o física de un lugar en la cual se realizan gastos.

➤ **Elementos**

Esta perspectiva representa los elementos de gasto que en algún momento pueden ser objeto de análisis. Representan los objetos mediante los cuales se realizan los gastos de una determinada institución.

➤ **Tiempo**

Esta perspectiva representa el espacio de tiempo en el que se está analizando cada una de las perspectivas anteriores o alguna combinación de las mismas. Esta perspectiva es muy útil a la hora de conocer como ha sido el comportamiento de determinados indicadores financieros en diferentes periodos de tiempo.

#### **Identificar posibles estructuras en forma de árbol**

Estas estructuras constituyen objetos, o abstracciones de conceptos del negocio cuya representación posee una relación con otro u otros objetos definiendo jerarquías entre los mismos, sin niveles definidos, formando los que se ha dado a conocer como estructuras de árbol. Dentro de las posibles estructuras en forma de árbol en el sistema se identificaron:

- dim\_estructura
- dim\_cuentas
- dim\_centros
- dim\_elementos

### **Indicadores**

Para un mejor análisis, los indicadores se definieron en 9 grandes grupos, los cuales se muestran a continuación, para más detalle ver Anexo 3:

I. Razones de liquidez.

Se utilizan como medio de apreciar la capacidad de la empresa para garantizar sus obligaciones a corto plazo.

II. Razones de solvencia.

Mide la capacidad de la empresa para garantizar la totalidad de sus deudas con terceros.

III. Razones de Apalancamiento (Endeudamiento)

Miden el grado en el cual la empresa ha sido financiada mediante deudas (Endeudamiento) y Capital (Autonomía).

IV. Razones de Rentabilidad (Rentabilidad).

Mide la eficiencia con que está siendo manejada la empresa, la relación entre los beneficios y los capitales invertidos.

V. Razones de Crecimiento (Ventas)

Miden la habilidad de la empresa para mantener su posición económica a través del comportamiento de las ventas y de las utilidades.

VI. Razones de autofinanciamiento

Miden la capacidad de la empresa para cubrir con los fondos autogenerados diferentes recursos invertidos en la misma.

VII. Productividad

Mide el nivel de productividad para cada entidad expresado en la utilidad alcanzada.

VIII. Inmovilización de recursos financieros.

Mide los recursos financieros que no pueden ser utilizados por la empresa en un largo periodo de tiempo.

**2.1.4. Construcción del modelo conceptual de los hechos con sus respectivas perspectivas e indicadores identificados**

Se determinó realizar dos tipos de análisis a los indicadores, el primero de forma más detallada para analizar el saldo de cada una de las cuentas involucradas en la obtención de los mismos, y el segundo para realizar un análisis desde el punto de vista más general, o sea, de los indicadores y de las relaciones que se establecen entre ellos. Se definieron para esto, dos hechos, “cub\_hecho\_contabilidad\_cuenta” para el análisis más detallado y “cub\_hecho\_contabilidadind” para el análisis al nivel de los indicadores.

A continuación se muestra el modelo conceptual definido para el hecho “cub\_hecho\_contabilidad\_cuenta”, para el cual se definieron 5 perspectivas y 3 indicadores, ver Figura 7.

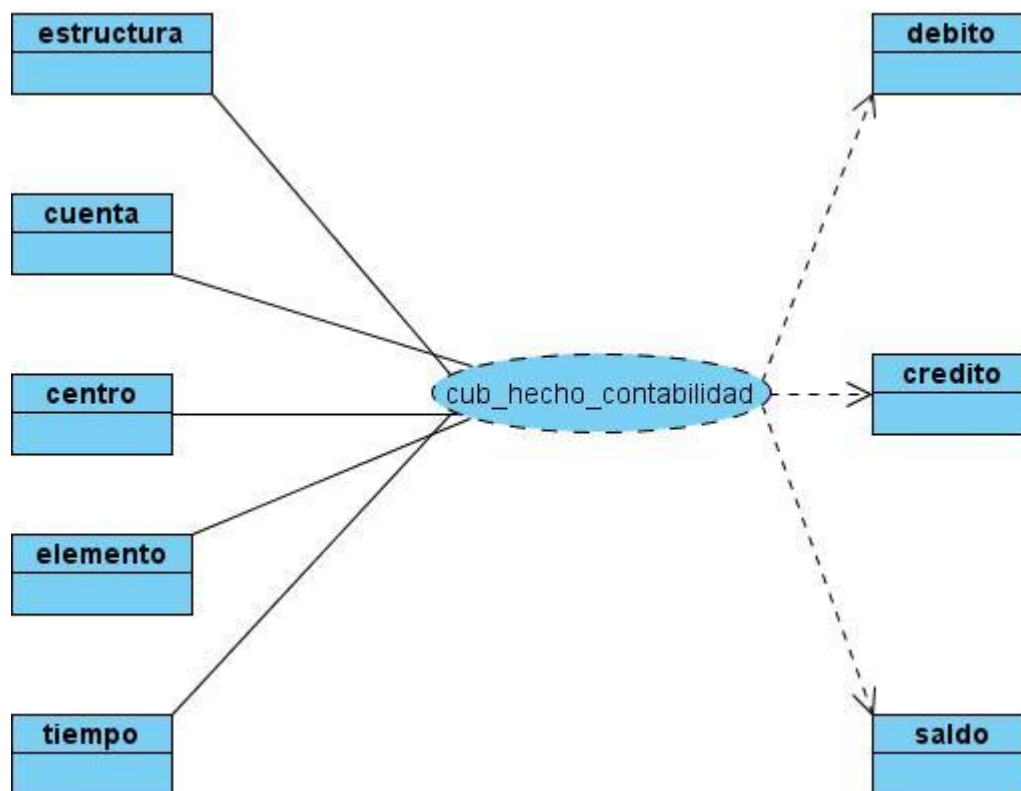


Figura 7: Modelo conceptual del hecho cub\_hecho\_contabilidad\_cuenta con perspectivas e indicadores identificados



Posteriormente se definió el hecho cub\_hecho\_contabilidadind, para el cual se identificaron 2 perspectivas y 49 indicadores, a continuación se muestra una porción de su modelo conceptual, ver Figura 8, para más detalle, ver Anexo 1.

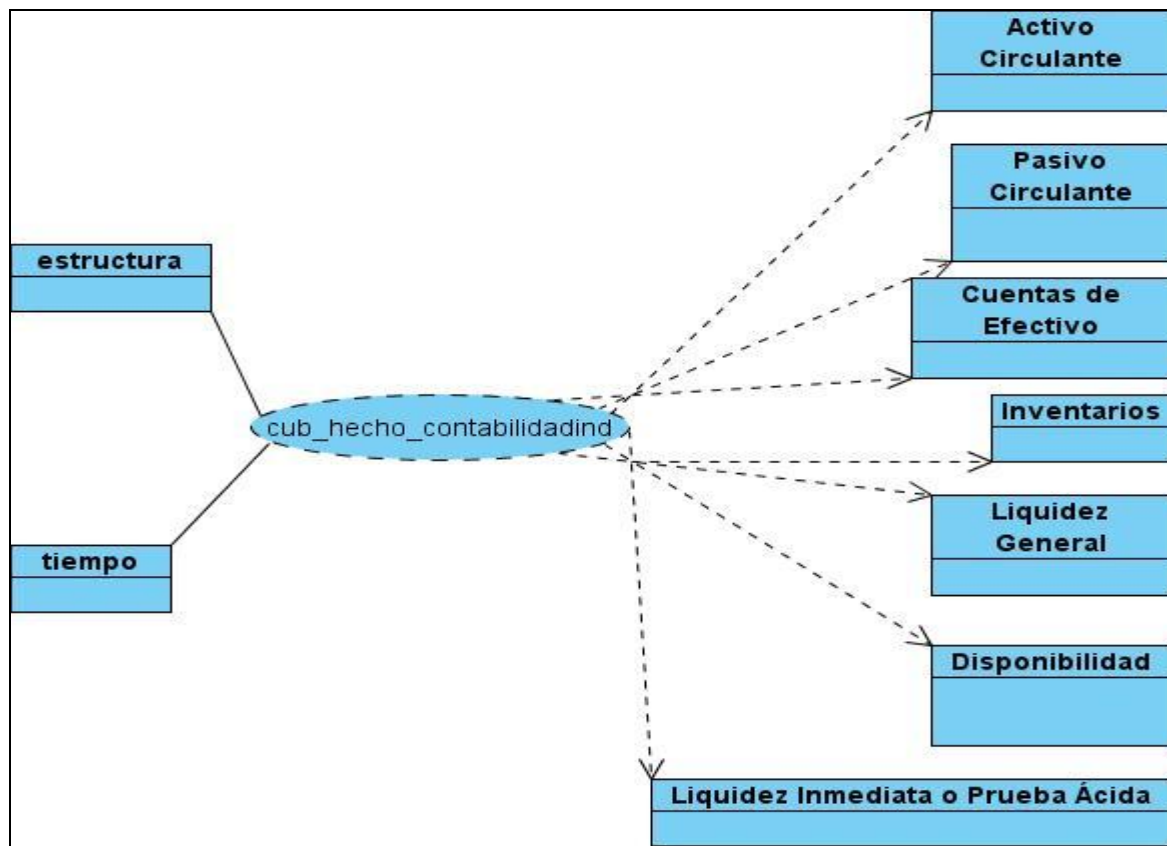


Figura 8: Modelo conceptual del hecho cub\_hecho\_contabilidadind con perspectivas e indicadores identificados

## 2.2. Análisis de los OLTP

Se analizaron las fuentes OLTP para definir el estado general de las mismas y determinar cómo serían calculados los indicadores, así como para establecer las respectivas correspondencias entre el modelo conceptual creado en el paso anterior y las fuentes de datos. Luego, se definieron cuáles campos de las fuentes de origen se incluirían en cada perspectiva. Se amplió el modelo conceptual con la información obtenida en este paso, y se finalizó definiendo las reglas del negocio.

### 2.2.1. Determinación de indicadores por cada hecho identificado

El valor de cada uno de los indicadores es el resultado de operaciones matemáticas, en algunos casos realizadas solamente con los saldos de las cuentas, mientras que en otros casos estos resultados se obtienen a partir de operaciones ejecutadas entre los propios indicadores, algunos de los cálculos

necesarios a realizar se presentan a continuación, para más detalle ver Anexo 3.

**cub\_hecho\_contabilidad\_cuenta**

- Crédito de la cuenta

$\Sigma$  credito

- Débito de la cuenta

$\Sigma$  debito

- Saldo de la cuenta

$\Sigma$  saldo

**cub\_hecho\_contabilidad**

I. Razones de liquidez.

- Liquidez General.

(Activo Circulante / Pasivo Circulante)

II. Razones de solvencia.

- Solvencia

(Activo Real / Financiamientos Ajenos)

III. Razones de Apalancamiento (Endeudamiento)

- Endeudamiento I.

(Financiamientos Ajenos / Financiamientos Propios)

IV. Razones de Rentabilidad (Rentabilidad).

- Rentabilidad de las ventas o margen sobre ventas.

(Utilidades o Pérdida del Período Antes del Impuesto / Ventas)

**2.2.2. Nivel de granularidad**

Uno de los elementos fundamentales para la construcción de un Data Mart lo es la definición del nivel de granularidad, este va a estar en dependencia del nivel de detalle especificado por el cliente a la hora de recuperar la información. En este caso se hizo necesario almacenar la información de la dimensión tiempo por mes y no diaria para obtener los saldos agrupados en un período de tiempo que posibilitara conocer la

variación monetaria de las mismas. A continuación se muestra el nivel de granularidad definido para las dimensiones de los dos hechos, ver Tabla 4.

Estructura	Cuentas	Centros	Elementos	Tiempo
Estructura	Grupo	Centro	Elemento	Quinquenio
	Contenido			Año
	Cuenta			Semestre
				Trimestre
				Mes

Tabla 4: Nivel de granularidad de las perspectivas

### 2.2.3. Modelo Conceptual ampliado de los hechos con sus respectivas perspectivas e indicadores identificados

Se definieron dos modelos conceptuales ampliados, uno para cada hecho definido. Cada modelo representa las relaciones entre las perspectivas y sus indicadores, así como la forma de calcularlos. La siguiente figura muestra el modelo conceptual del hecho “cub\_hecho\_contabilidad\_cuenta”, ver Figura 9

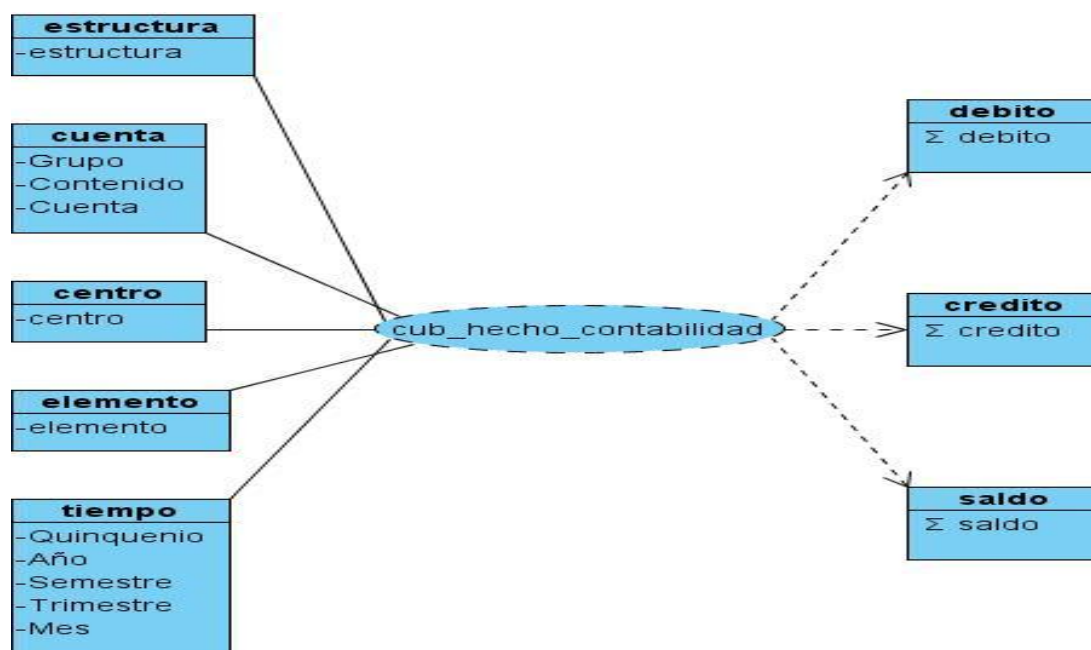


Figura 9: Modelo conceptual ampliado del hecho cub\_hecho\_contabilidad\_cuenta

En la siguiente Figura se muestra solamente una parte del modelo conceptual ampliado del hecho “cub\_hecho\_contabilidad”, ver Figura 10, para más detalle ver Anexo 2.

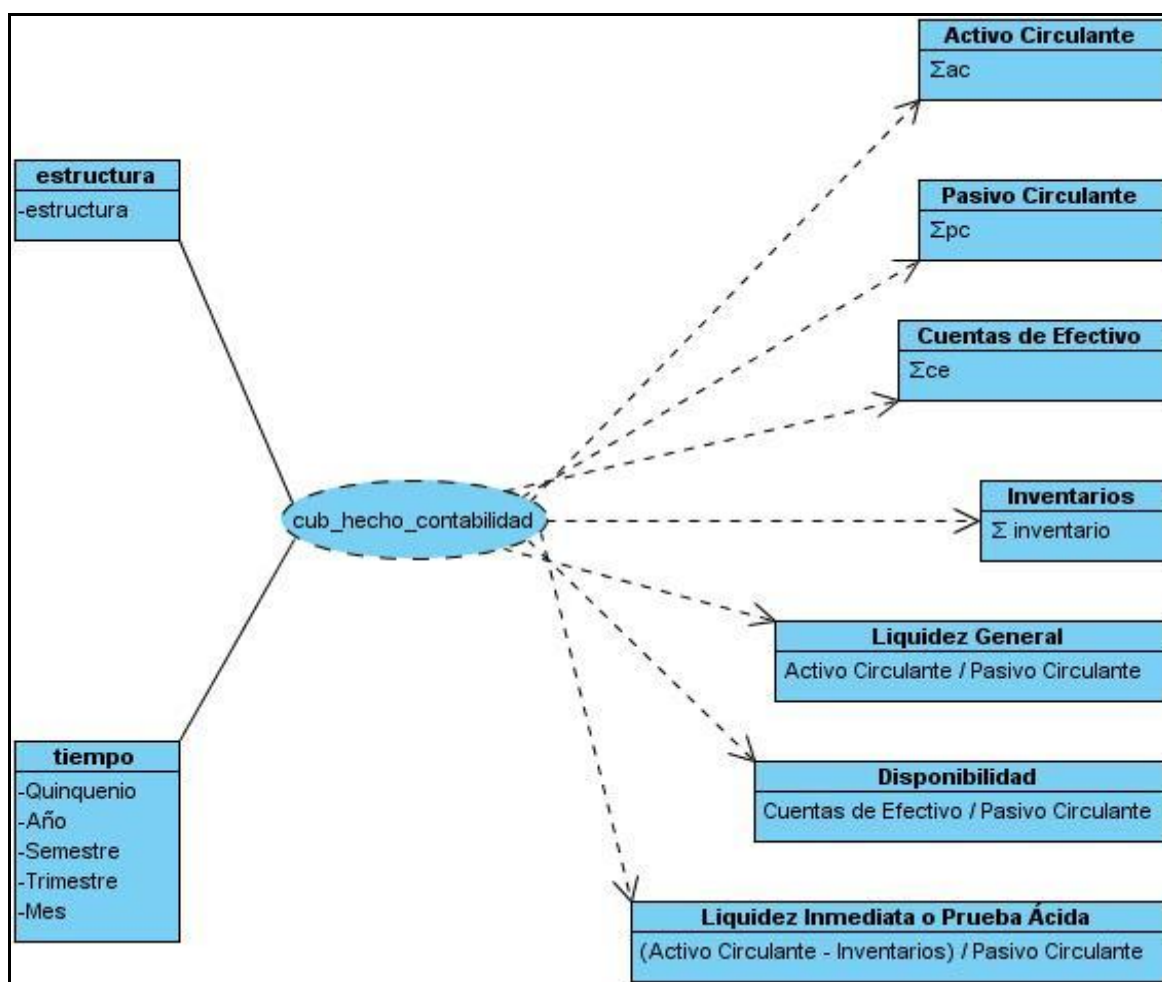


Figura 10: Modelo conceptual del hecho cub\_hecho\_contabilidad\_cuenta

### 2.3. Diccionario de datos del modelo conceptual ampliado de los hechos con sus respectivas perspectivas e indicadores identificados

En aras de llevar un mayor control en relación a las perspectivas e indicadores implicados en la construcción del Data Mart se hace necesario confeccionar un mecanismo que registre los detalles correspondientes a las mismas. Con este objetivo se muestran los diccionarios de datos de las perspectivas e indicadores respectivamente.

#### 2.3.1. Diccionario de datos de las perspectivas

A continuación se presentan los diccionarios de datos de las perspectivas por cada hecho identificado. Para ello se identificaron los hechos “cub\_hecho\_contabilidad\_cuenta” y “cub\_hecho\_contabilidad”. Con el

objetivo de evitar redundancia en las perspectivas que pueden ser comunes para ambos hechos se determinó registrarlas en un diccionario que queda descrito con el nombre de Diccionario de datos de las perspectivas comunes.

**Diccionario de datos de las perspectivas comunes**

En la siguiente tabla se encuentran representadas las perspectivas comunes definidas para la construcción del Data Mart.

No	Nombre de la perspectiva	Descripción de la perspectiva	Nombre del campo de la perspectiva.	Descripción del campo
1.	Estructura	Esta perspectiva representa las estructuras económicas que en algún momento podrán ser objeto de análisis.	idestructura	Este campo representa el identificador de la estructura física. El valor que toma este campo es el correspondiente a la propia estructura en el sistema operacional.
			codigo	Este campo representa el código de la estructura para diferenciarla de las demás.
			denominacion	Este campo representa la denominación de la estructura diferenciándola de las demás.
			idestructurapadre	Este campo representa el identificador de la estructura física padre. El valor que toma este campo es el correspondiente al padre de la propia estructura en el sistema operacional.

2.	Tiempo	Esta perspectiva representa el espacio de tiempo en el que se está analizando cada una de las perspectivas anteriores o alguna combinación de las mismas. Esta perspectiva es muy útil a la hora de conocer como ha sido el comportamiento de determinados indicadores financieros en diferentes periodos de tiempo.	Quinqueni	Este campo representa el Quinquenio.
			o	
			Año	Este campo representa el año.
			Semestre	Este campo representa el semestre.
			Trimestre	Este campo representa el trimestre.
Mes	Este campo representa el mes.			

Tabla 5: Diccionario de datos de las perspectivas comunes

En la siguiente tabla se encuentran representadas las perspectivas definidas para la construcción del hecho cub\_hecho\_contabilidad\_cuenta. De las 5 perspectivas definidas solo se muestran algunas, ver Tabla 6, para más detalle ver Anexo 4.

No	Nombre de la perspectiva	Descripción de la perspectiva	Nombre del campo de la perspectiva.	Descripción del campo
1.	Cuenta	Esta perspectiva representa las cuentas económicas que en algún momento podrán ser objeto de análisis.	idcuenta	Este campo representa el identificador de la cuenta contable. El valor que toma este campo es el correspondiente a la propia cuenta en el sistema operacional.
			grupo	Este campo representa el Grupo al que corresponde la cuenta.
			contenido	Este campo representa el

				Contenido económico al que corresponde la cuenta.
			concat	Este campo representa el concat de la cuenta para diferenciarlas unas las otras.
			descripcion	Este campo representa la descripción de la cuenta.
			idcuentapadre	Este campo representa el identificador de la cuenta padre inmediato.

Tabla 6: Diccionario de datos de las perspectivas

### 2.3.2. Diccionario de datos de los indicadores

En la siguiente tabla se encuentran representados los indicadores definidos para la construcción del hecho cub\_hecho\_contabilidad\_cuenta el que posee un total de 3 indicadores reflejados en la siguiente tabla. Ver tabla 7.

No	Nombre del indicador	Descripción del indicador	Fórmula de cálculo del indicador
1.	Débito	Determina el crédito correspondiente a una cuenta.	$\sum$ debito
2.	Crédito	Determina el crédito correspondiente a una cuenta.	$\sum$ credito
3.	Saldo	Determina el saldo correspondiente a una cuenta el cual se determina en dependencia de la naturaleza de la cuenta.	$\sum$ saldo

Tabla 7: Diccionario de datos de los indicadores del hecho cub\_hecho\_contabilidad\_cuenta

En la siguiente tabla se encuentran representados los indicadores definidos para la construcción del hecho

cub\_hecho\_contabilidad del Data Mart. De los 49 indicadores definidos solo se muestran algunos, para más detalles ver Tabla 8, para más detalle ver Anexo 3.

<b>No</b>	<b>Nombre del indicador</b>	<b>Descripción del indicador</b>	<b>Fórmula de cálculo del indicador</b>
4.	Liquidez General	Determina la posibilidad de la entidad de poder hacer frente a sus pagos a corto plazo.	(Activo Circulante / Pasivo Circulante)
5.	Liquidez Inmediata o Prueba Ácida	Determina cuántos pesos de Activos más líquidos posee la empresa por cada peso de deuda a corto plazo.	((Activo Circulante - Inventarios) / Pasivo Circulante )
6.	Disponibilidad	Determina cuántos pesos de efectivo dispone la empresa para pagar un peso de deuda a corto plazo.	(Cuentas de Efectivo / Pasivo Circulante)
7.	Capital de Trabajo	Determina los fondos o recursos con que opera una empresa a corto plazo, después de cubrir las deudas y obligaciones que vencen en ese corto plazo.	(Activo Circulante - Pasivo Circulante )
8.	Solvencia	Mide la capacidad de la empresa para garantizar la totalidad de sus deudas con terceros.	(Activo Real / Financiamientos Ajenos)
9.	Endeudamiento I.	Miden el grado en el cual la empresa ha sido financiada mediante deudas (Endeudamiento) y Capital (Autonomía).	(Financiamientos Ajenos / Financiamientos Propios)

Tabla 8: Diccionario de datos de los indicadores del hecho cub\_hecho\_contabilidad

#### **2.4. Modelo lógico del Data Mart**



Aunque si bien es cierto que los tres modelos lógicos poseen ventajas y desventajas, no existe una guía que diga cual es mejor que otro, lo que sí es cierto es que la aplicación de cada uno de ellos está en dependencia de lo que desea el cliente y del funcionamiento del negocio.

#### 2.4.1. Definir tipo de Modelo lógico del Data Mart

Para la implementación del almacén de datos se optó por la variante del modelo lógico de constelación de hecho; debido a que existen varios puntos del análisis que no pueden ser vistos en un solo cubo. Por ejemplo, para conocer el saldo de las cuentas por estructuras se hace necesario que intervengan en el análisis las dimensiones de cuenta y estructura, no siendo así para saber el comportamiento de la liquidez general de la propia estructura en el que la cuenta no llegará a formar parte de las perspectivas por las que se puede realizar este análisis. Ver Figura 11.

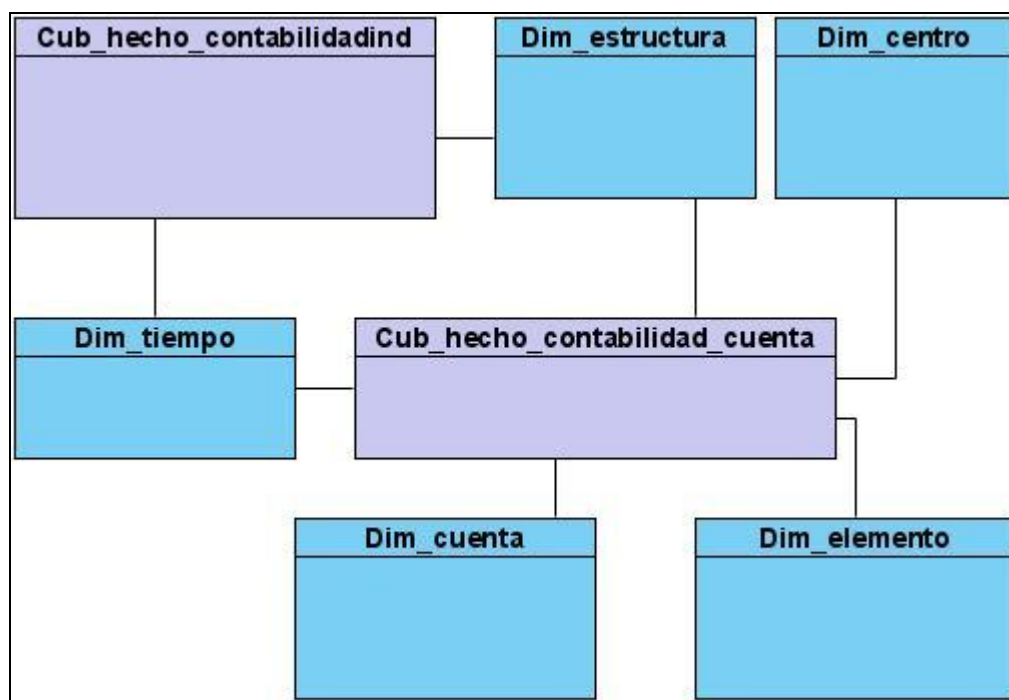


Figura 11: Modelo lógico del Data Mart

#### 2.4.2. Definir estándares para objetos físicos

Con el objetivo de lograr un mayor entendimiento de la base de datos y brindar una visión de conjunto lo más estructurada posible en relación con la estrategia arquitectónica del ERP y tomando en cuenta que actualmente no existe en el proyecto ningún documento que rijan esta nomenclatura para el desarrollo de almacenes de datos a continuación se definen algunas pautas para la nomenclatura en los mismos.

#### Nomenclatura de la base de datos

Los nombres de las Bases de Datos comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing*\*. Con sólo leerlo se reconoce el propósito de la misma.

Ejemplo

```
Almacen
```

### Apariencia de los esquemas

El nombre a emplear para los esquemas se escribe con todas las letras en minúscula, comenzando por el prefijo mod, a continuación el símbolo “\_”, y por último el nombre del módulo.

Ejemplo

```
create schema 'mod_contabilidad';
```

### Nombres de las tablas de dimensiones y hecho

El nombre empleado, debe escribirse con todas las letras en minúscula para evitar problemas con el *Case Sensitive* del gestor y con solo leerlo se reconoce el propósito de la misma.

Ejemplo

```
create table 'dim_producto';  
create table 'cub_hecho_contabilidad';
```

### Prefijos a utilizar en la creación de tablas de dimensiones y hecho

Los prefijos a utilizar en la creación de tablas serán los siguientes:

- **dim\_** Prefijo utilizado en tablas de dimensiones que almacenan los datos de la dimensión a la que se hace referencia.
- **cub\_hecho\_** Prefijo utilizado en tablas de hecho.

### Apariencia de los campos

El nombre a emplear para los campos se escribe con todas las letras en minúscula, con solo leerlo se reconoce el propósito del mismo y debe incluir un comentario con su descripción. Si el campo es un identificador debe empezar con id.

Ejemplo

```
add field 'idcuenta';
```

### **Nombre de las llaves primarias**

El nombre de las restricciones se escribe con minúscula. Comienza con el identificador id y si es la llave que proviene de la base de datos operacional le debe seguir el nombre de la tabla de la que proviene, todo junto y en minúscula. En caso que la llave sea autogenerada en la propia dimensión entonces comienza con el identificador id y seguido el nombre de la tabla, todo junto y en minúscula sin el guión bajo.

#### Ejemplo

idcuenta (Llave primaria de la tabla “dim\_cuenta” que proviene de la base de datos operacional).

iddimtiempo (Llave primaria de la tabla “dim\_tiempo”).

### **Nombre de las llaves foráneas**

El nombre de las llaves foráneas se escribe con minúscula y el nombre de la llave primaria de la tabla donde pertenece.

#### Ejemplo

idcuenta (Llave foránea de la tabla “dim\_cuenta”)

iddimtiempo (Llave primaria de la tabla “dim\_tiempo”)

### **Nombres de las funciones y triggers**

El nombre empleado permite con solo leerlo reconocer el propósito del mismo. Se utilizan los prefijos siguientes para la denominación de funciones y triggers.

**f\_** Funciones (Procedimientos Almacenados)

**ft\_** Funciones de triggers

**t\_** Triggers

### **Nombre de los tablespace**

El nombre se definirá todo en minúscula, empezando con la letra tbs, seguido un guión bajo y el nombre del módulo junto en minúscula.

#### Ejemplo

“Tbs\_modcontabilidad” (tablespace del módulo de contabilidad)

### **Nombre de los dominios**

El nombre se definirá todo en minúscula, empezando con la letra d, posteriormente el nombre del módulo al que pertenece y por último el nombre del dominio, separados por punto.

Ejemplo

“d\_contabilidad” (dominio que pertenece al módulo contabilidad)

### **Nombre de las secuencias**

El nombre de una secuencia se definirá todo en minúscula empezando con la letra sec posteriormente un guión bajo y nombre de la tabla donde pertenece.

Ejemplo

“sec\_hechocontabilidad\_seq”

“sec\_hechocontabilidad\_cuenta\_seq”

### **2.4.3. Diseñar tablas y columnas físicas**

El modelo físico está compuesto por 2 tablas de hechos, 5 tablas de dimensiones y 4 tablas closure. En el caso de las tablas dimensiones, existen algunas comunes para las 2 tablas de hechos, como son el caso de “dim\_estructura” y “dim\_tiempo” posibilitando acceder a la misma información desde cubos diferentes. Por otro lado se emplearon tablas closure para gestionar el funcionamiento de las estructuras en forma de árbol. Ver Figura 12.

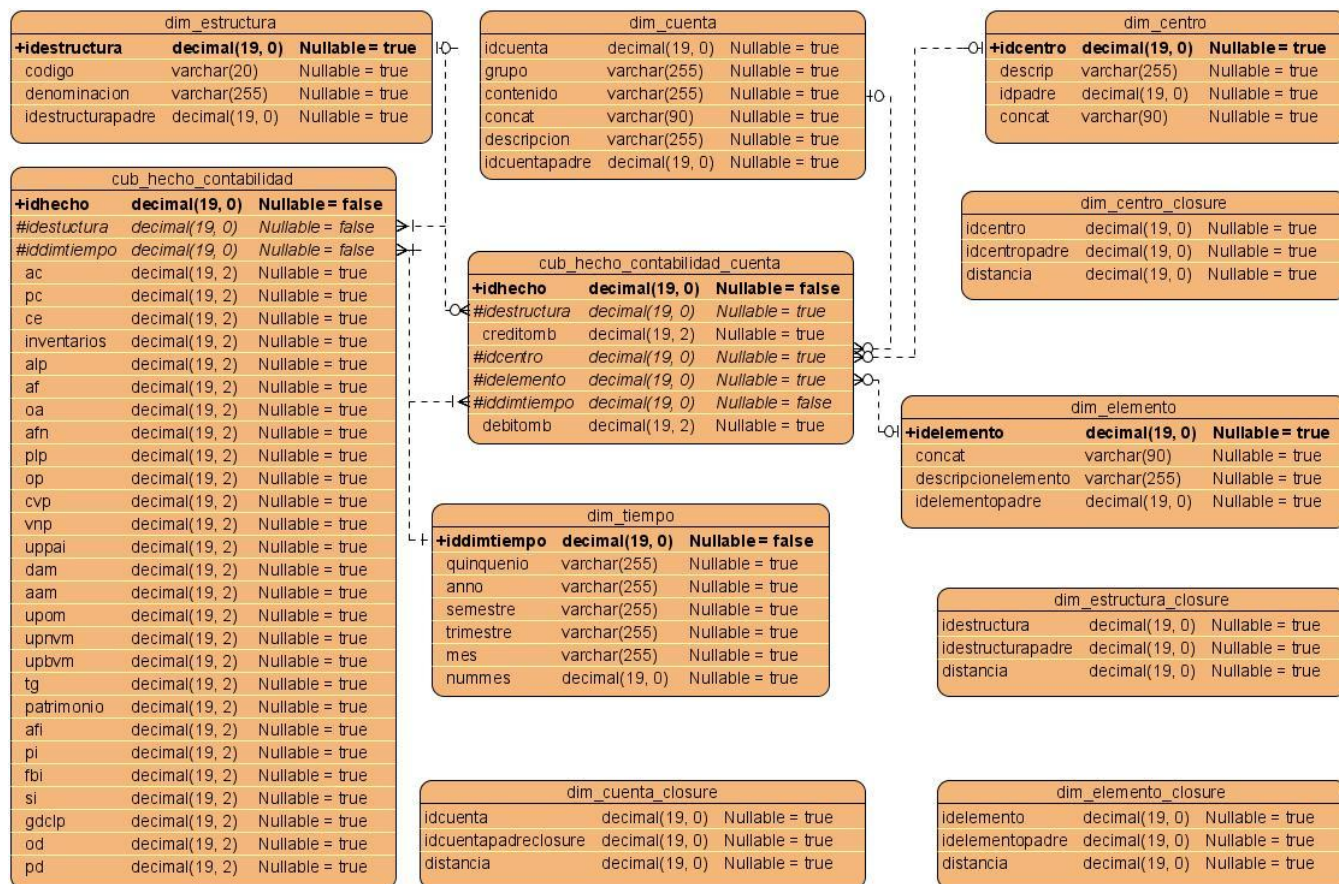


Figura 12: Modelo físico del Data Mart

### 2.4.3.1. Diseñar las tablas auxiliares físicas para la concepción estructuras en forma de árbol

Con el considerable incremento de los niveles de las diferentes dimensiones resultaría un inconveniente la estructura estática planteada en el modelo conceptual. Con el objetivo de resolver este problema se crearon algunas tablas auxiliares que ayudaron en la prolongación de estas estructuras. La representación de las mismas se presenta a continuación. Ver Figura 13.

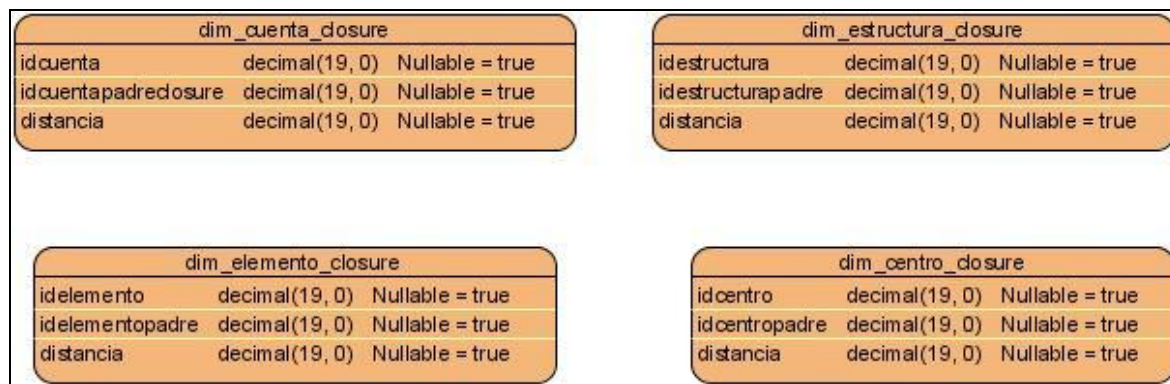


Figura 13: Modelo físico de las tablas auxiliares del Data Mart

Las tablas “dim\_estructura\_closure”, “dim\_cuenta\_closure”, “dim\_centro\_closure”, “dim\_elemento\_closure” se crearon específicamente para soportar la creación de dimensiones en forma de árbol.

## 2.5. Proceso ETL

Con el objetivo de centralizar los datos existentes en diferentes empresas para su posterior aplicación en la inteligencia de negocios surgió el término ETL, proceso de organizar el flujo de los datos entre diferentes sistemas en una organización y aportar los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos.

La idea es que una aplicación ETL lea los datos primarios de unas bases de datos de sistemas principales, realice transformación, validación, el proceso cualitativo, filtración y al final escriba datos en el almacén y a partir de ese momento los datos estén disponibles para ser analizados por los usuarios.(11)

### 2.5.1. Mapeo de datos fuente a destino

Con el objetivo de evitar la pérdida de datos en el proceso ETL se hace necesario llevar el control del mapeo de los mismos documentando la relación existente entre los campos de la base de datos origen y su correspondiente en la base de datos destino. A continuación solo se muestra una parte de este mapeo, ver Tabla 10. Para más detalle ver Anexo 5.

Para el caso puntual de la tabla “dim\_tiempo” no se establece mapeo, se creó la función “f\_registrar\_annos” mediante la cual se insertan los datos a partir de su ejecución. Al entrarle como parámetros una fecha de inicio y de fin a esta, se generan todos los datos definidos para la dimensión, así mismo la generación de los identificadores de la tabla “dim\_tiempo” se realizan a través de una secuencia denominada “sec\_dimtiempo\_seq”, ambos objetos pertenecen al esquema “mod\_comun”.

<b>Mapa Lógico de Datos</b>					
<b>Nombre BD Fuente: Cedrux</b>			<b>Nombre BD Destino: Almacen</b>		
<b>Nombre de la tabla</b>	<b>Nombre del campo</b>	<b>Tipo de dato</b>	<b>Nombre de la tabla</b>	<b>Nombre del campo</b>	<b>Tipo de dato</b>
dat_estructura	idestructura	numeric(19,0)	dim_estructura	idestructura	numeric(19,0)
dat_estructura	codigo	varchar(20)	dim_estructura	codigo	varchar(20)

dat_estructura	denominacion	varchar(255)	dim_estructura	denominacion	varchar(255)
dat_estructura	idpadre	numeric(19,0)	dim_estructura	idestructurapadre	numeric(19,0)

Tabla 9: Mapa Lógico de Datos

### 2.5.2. Establecer condiciones adicionales y restricciones

Para la realización del proceso ETL es necesario tener en cuenta una serie de condiciones y restricciones adicionales, mediante las que se garanticen que los datos en el almacén sean lo más confiable posibles, estas restricciones son las siguientes:

1. Para el llenado de la tabla “cub\_hecho\_contabilidad\_cuenta” se hace necesario que las medidas sean calculadas por estructura, centro, elemento y tiempo.
2. Para el llenado de la tabla “cub\_hecho\_contabilidadind” se hace necesario que las medidas sean calculadas por estructura y tiempo.
3. El valor del campo saldo en la tabla cubo “cub\_hecho\_contabilidad\_cuenta” está dado por la naturaleza de la cuenta.
4. El saldo para las cuentas solo se almacena en las cuentas hojas por lo que para obtener el saldo de una cuenta del primer nivel se hace necesario sumar todo el saldo de sus cuentas hojas.
5. Para la tabla “cub\_hecho\_contabilidadind” en caso de que para una estructura no existan cuentas asociadas el valor a poner en sus campos correspondientes será “0”.
6. Para buscar el “iddimtiempo” en la tabla de dimensión primero es necesario buscar los valores del mes y el año en la fuente de origen siendo necesario particionar en estos dos elementos el campo fecha que allí se almacena.
7. La división por 0 no está definida, por lo que en el caso de un denominador con valor 0, contablemente se divide por 1.

### 2.5.3. Diseño y construcción de la automatización del proceso de ETL

Para lograr una correcta carga de los datos es de vital importancia el orden en que se ejecuten los trabajos para la realización de la misma. Para una mayor comprensión se explica mediante dos momentos fundamentales, en el primero se expondrán los trabajos paralelos, es decir, los que no dependan de otros trabajos para su ejecución; y en un segundo momento, los trabajos dependientes o sea, aquellos que dependen de la culminación de otros trabajos para su ejecución.

Los trabajos paralelos en este caso son los que se encuentran relacionados con el llenado de las dimensiones, puesto que no dependen de ningún otro trabajo, no siendo así para el llenado de las tablas closure y la tabla de hechos que dependen de que hayan sido terminados previamente los trabajos de inserción de datos en las dimensiones. Ver Figura 14, 15 y 16.

**Trabajos paralelos:**

1. Llenado de la tabla “dim\_estructura”
2. Llenado de la tabla “dim\_cuenta”
3. Llenado de la tabla “dim\_centro”
4. Llenado de la tabla “dim\_elemento”



Figura 14: Trabajos paralelos

**Trabajos dependientes:**

1. Llenado de la tabla “dim\_cuenta\_closure”. En este caso es necesario que antes se haya llenado la tabla “dim\_cuenta”.
2. Llenado de la tabla “dim\_centro\_closure”. En este caso es necesario que antes se haya llenado la tabla dim\_centro.
3. Llenado de la tabla “dim\_centro\_closure”. En este caso es necesario que antes se haya llenado la tabla "dim\_centro".
4. Llenado de la tabla “dim\_ dim\_elemento\_closure”. En este caso es necesario que antes se haya



llenado la tabla “dim\_elemento”.

5. Llenado de la tabla “cub\_hecho\_contabilidad”. En este caso es necesario que antes se hayan llenado las tablas “dim\_estructura” y “dim\_tiempo”.

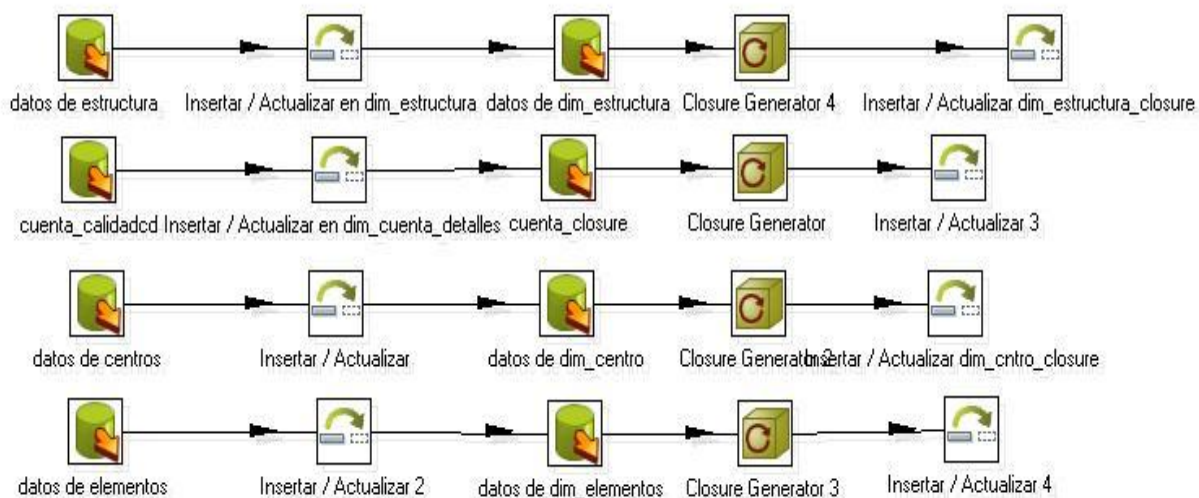


Figura 15: Trabajos dependientes

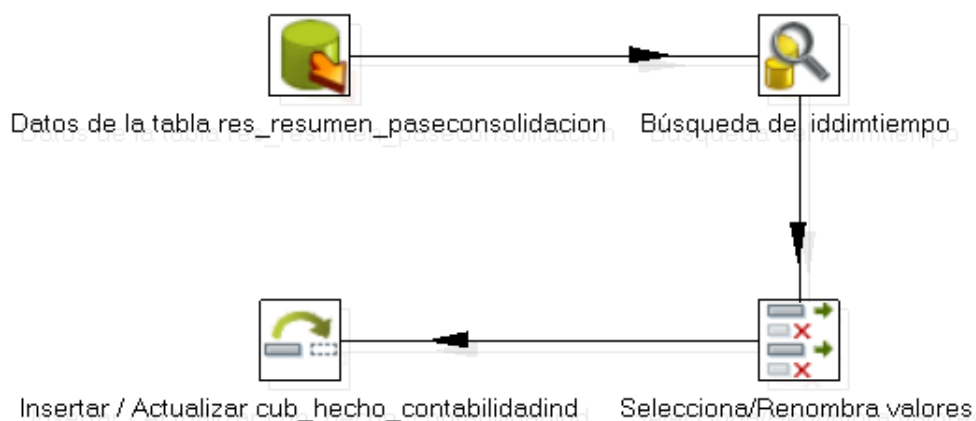


Figura 16: Llenado del cubo cub\_hecho\_contabilidadind

## 2.6. Conclusiones del capítulo

La identificación y definición de los indicadores, perspectivas y hechos constituyeron la base a partir de la cual fue posible realizar el diseño del Data Mart. De igual forma, el proceso ETL posibilitó organizar, limpiar y calcular los datos que van desde las fuentes de origen hacia la destino, logrando así una alta confiabilidad en los mismos, empleando además el modelo constelación para realizar varios tipos de análisis en la implementación del Data Mart con el fin de responder a las necesidades requeridas por el cliente.

## CAPÍTULO 3: VALIDACIÓN Y DESPLIEGUE

### Introducción

En el presente capítulo se tratan temas relacionados con las pruebas realizadas al Data Mart con el objetivo de realizar su validación; las mismas se basaron en dos aspectos fundamentales, las relacionadas con el desempeño del 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.

### 3.1. Validación

La validación de todo sistema constituye el mecanismo mediante el cual se garantiza el cumplimiento de las funcionalidades definidas por los clientes al inicio de su desarrollo y su correcto funcionamiento. En este caso como se mencionó anteriormente las pruebas se dividieron en dos vertientes, para comprobar el cumplimiento de las necesidades planteadas por el cliente y el correcto desempeño del almacén de datos. Para la primera variante se hizo necesario la creación de dos cubos OLAP definidos por dos tipos diferentes de análisis, uno más global en el cual son analizados los indicadores financieros y las interrelaciones que se establecen entre ellos a través de sus dimensiones; y un segundo cubo donde se realiza una observación del comportamiento de los saldos de las cuentas por sus distintas dimensiones para dar respuesta a las inquietudes generadas por el incorrecto funcionamiento de alguno de los indicadores definidos en el primer análisis. Para la segunda variante se realizaron pruebas de concepto para validar el diseño del almacén posibilitando determinar su tiempo de respuesta a partir de la complejidad de las consultas y el volumen de datos almacenados.

#### 3.1.1. Validación funcional

Para efectuar esta validación fue necesario el cálculo preciso de los indicadores financieros realizando el diseño del almacén de datos mediante el uso de una herramienta para la representación de los cubos, el workbench (ver epig.1.8.3), definiendo en cada cubo las dimensiones y medidas correspondientes. Estos cubos se denominaron “**contabilidad**” y “**contabilidadind**” siendo el primero el del análisis de los indicadores y el segundo más detallado. Cada uno de estos cubos tiene como base fundamental las tablas de hechos “**cub\_hecho\_contabilidad\_cuenta**” y “**cub\_hecho\_contabilidad**” respectivamente, y en el caso del primer cubo este permite el cálculo de 3 indicadores, y el segundo calcular 49. Ver Figura 17.

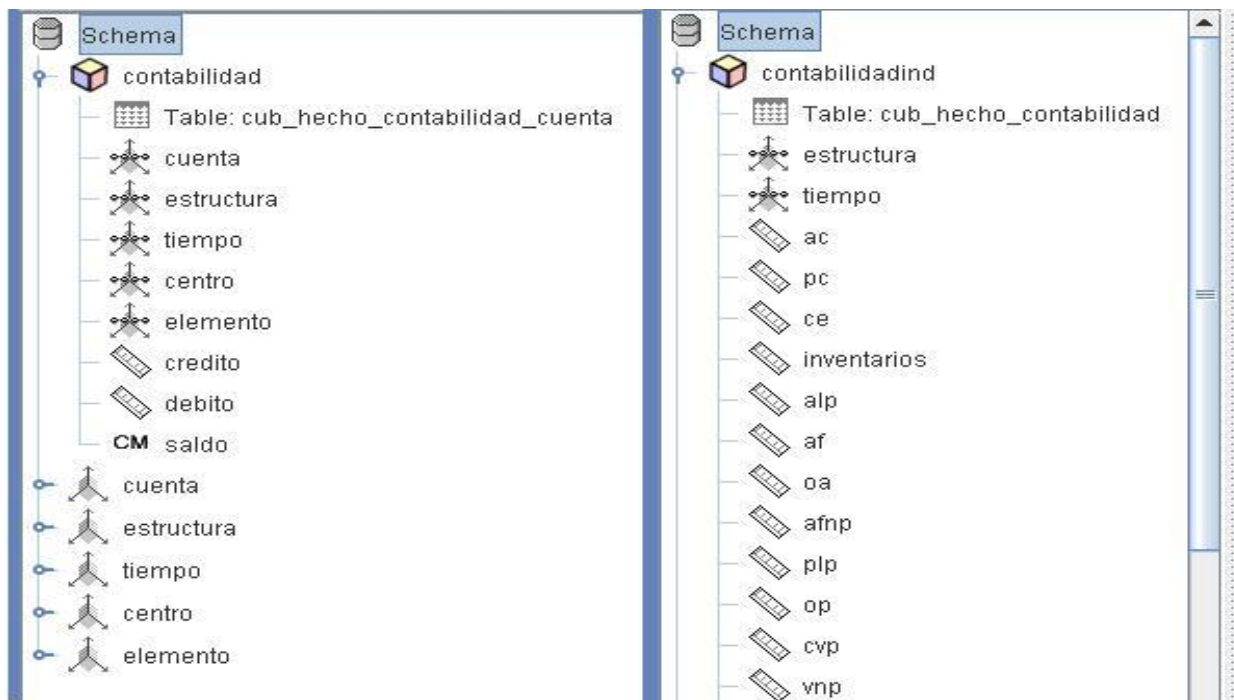


Figura 17: Cubos OLAP en el workbench

La salida del sistema se realiza mediante reportes que representan a escenarios que cumplen las necesidades del cliente, la comprobación de los cálculos de los indicadores para estos escenarios se realizó mediante consultas MDX a través de los reportes mostrados con la herramienta Mondrian, a continuación se presentan algunos de los escenarios más comunes.

Después de obtenidos los reportes con los valores de los indicadores se procedió a la validación del sistema por parte de los especialistas funcionales, pertenecientes al Ministerio de Finanzas y Precios con el objetivo de comprobar el cumplimiento de las necesidades planteadas por ellos. El resultado de los tiempos de respuesta de la aplicación para varios de estos reportes fueron comparados con sus homólogos en el sistema operacional siendo superiores para este último.

Estas pruebas concluyeron satisfactoriamente plasmando el resultado de la misma en un acta de validación a la solución desarrollada. Ver anexo 9.

### 3.1.2. Validación del diseño mediante pruebas de rendimiento a la aplicación

Las pruebas de rendimiento a las aplicaciones son muy utilizadas con el objetivo de evaluar la eficiencia de una solución desarrollada. Las mismas se realizan mediante el uso de la herramienta Jmeter, mediante la cual se pueden simular diferentes situaciones y escenarios. Para el caso en cuestión, las características del servidor de aplicaciones poseen las siguientes características:

- El servidor de aplicaciones debe estar corriendo.
- Las características del servidor de aplicaciones es de 1GB de RAM con 160 GB de disco duro.
- Las características de la red 100 Mbps.
- La tabla principal se encuentra poblada con más de 10000 tuplas.

Se realizaron las pruebas arrojando los siguientes resultados. Ver Tabla 10.

<b>ID del escenario</b>	<b>Escenario de la sección</b>	<b>Carga de trabajo</b>	<b>Descripción</b>	<b>Resultado de la prueba</b>
EC 1: Realizar la carga inicial de todos los indicadores financieros.	EC 1.1: Realizar la carga inicial de los 49 indicadores financieros con sus 5 dimensiones correspondientes para una concurrencia de 20 usuarios.	20	En este escenario se realiza la carga inicial de todos los indicadores financieros.	18.3 seg
	EC 1.2: Realizar la carga inicial de los 49 indicadores financieros con sus 5 dimensiones correspondientes para una concurrencia de 100 usuarios.	100	En este escenario se realiza la carga inicial de todos los indicadores financieros.	27.9 seg
	EC 1.3: Realizar la carga inicial de los 49 indicadores financieros con sus 5 dimensiones correspondientes para una concurrencia de 150 usuarios.	150	En este escenario se realiza la carga inicial de todos los indicadores financieros.	101 seg

	EC 1.4: Realizar la carga inicial de los 49 indicadores financieros con sus 5 dimensiones correspondientes para una concurrencia de 200 usuarios.	200	En este escenario se realiza la carga inicial de todos los indicadores financieros.	126.4 seg
--	---	-----	---	-----------

Tabla 10: Pruebas de rendimiento a la aplicación

Después de analizar los resultados correspondientes a las pruebas realizadas se puede llegar a la conclusión que los tiempos de respuestas para las mismas son satisfactorios. Esta valoración está fundamentada por la complejidad de la consulta realizada para este caso en particular. Resulta importante aclarar que los tiempos mostrados se determinan desde que el usuario realiza la acción de mostrar el reporte, que la aplicación transforme la consulta en lenguaje MDX a lenguaje plpgsql, esta busque la información en la base de datos para luego pintarla mediante el uso del componente jpivot para finalmente poder ser visualizada al usuario final, siendo el proceso de pintar el que más tiempo requiere.

Para apreciar algunas de las vistas de la configuración del Jmeter ver Anexo 6, 7 y 8.

### 3.3. Despliegue del Data Mart

La estrategia para el despliegue del Data Mart se basa en la existencia de 3 servidores:

- Un servidor de aplicaciones con 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: 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 recomienda que los servidores de origen y el almacén de datos se encuentren separados para brindarles una mayor seguridad a los mismos y más protección con respecto a fallas. Los servidores realizarán la comunicación mediante el protocolo TCP/IP, y la PC cliente se conectará mediante el

protocolo HTTP para realizar las consultas al Data Mart 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.

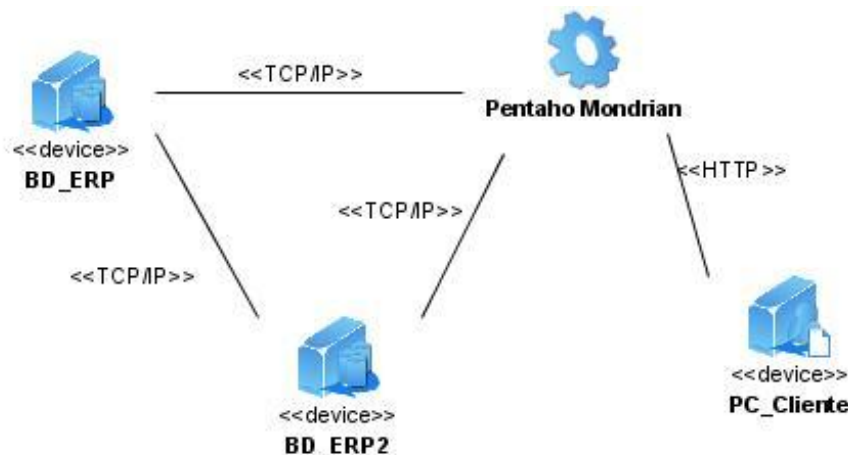


Figura 18: Diagrama de despliegue del Data Mart

### 3.4. Conclusiones del capítulo

La validación del Data Mart desarrollado fue el aspecto fundamental del capítulo que concluye. En este sentido se realizaron pruebas de concepto al Data Mart, teniendo en cuenta la complejidad de las consultas y el volumen de datos que en él se gestiona, con el objetivo de validar y evaluar el diseño, las cuales arrojaron valores satisfactorios para cada uno de los indicadores medidos. Se realizaron además pruebas funcionales al sistema cuyos resultados cumplen con las necesidades y expectativas establecidas por el cliente.

## CONCLUSIONES GENERALES

Una vez culminado el trabajo de diploma se puede concluir que se le dio cumplimiento al objetivo general trazado para el proceso de la investigación, puesto que:

- Se desarrolló un Data Mart para la obtención de los indicadores financieros de los subsistemas de Contabilidad y Costos y Procesos del proyecto ERP, con el objetivo de favorecer el proceso de toma de decisiones a partir de datos que posibiliten la creación de modelos para la anticipación de hechos contables futuros, mejorando los tiempos de respuesta para su obtención; validado a través de pruebas realizadas por los funcionales a las funcionalidades y el diseño, arrojaron resultados favorables, dando cumplimiento a las necesidades y exigencias previstas para su desarrollo.

## RECOMENDACIONES

Para un posterior desarrollo y perfeccionamiento del sistema desarrollado se recomienda:

- Continuar perfeccionando el sistema para el cálculo de otros indicadores financieros.
- Agregar nuevas vistas de análisis mediante la visualización de información a través de pizarras de gráficos.
- Realizar la integración del sistema con el marco de trabajo del CEDRUX.
- Gestionar la seguridad del sistema para el acceso a la información a nivel de entidades.

**BIBLIOGRAFÍA REFERENCIADA**

1. Dos enfoques clásicos en el diseño de la arquitectura de un Data Warehouse. Dos enfoques clásicos en el diseño de la arquitectura de un Data Warehouse. [En línea] 12 de 1 de 2009. [Citado el: 9 de 10 de 2010.] <http://bi-businessintelligence.blogspot.com/2009/01/cif-vs-md-dos-enfoques-clasicos-en-el.html>.
2. Sinnexus. Business Intelligence + Informatica estrategca. Business Intelligence + Informatica estrategca. [En línea] 2007. [Citado el: 9 de 10 de 2010.] [http://www.sinnexus.com/business\\_intelligence/datamart.aspx](http://www.sinnexus.com/business_intelligence/datamart.aspx).
3. Alonso, Eddy Manuel Infante. Diseño e Implenmentación de AkademosMart. Ciudad de la Habana : s.n., 2005.
4. Real Academia Española. Real Academia Española. Real Academia Española. [En línea] 2011. [Citado el: 12 de 1 de 2011.] <http://buscon.rae.es/drael/SrvltConsulta?LEMA=modelo>.
5. Universidad de las Ciencias Informáticas. Conferencia 2 Modelo de Datos. Ciudad de la Habana : s.n., 2011.
6. Universidad de las Ciencias Informática. Conferencia 3 Modelo Relacional. Ciudad de la Habana : s.n., 2011.
7. Universidad Nacional de Colombia. Modelo Relacional. Modelo Relacional. [En línea] [Citado el: 12 de 1 de 2011.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap4-1.html>.
8. Ralph Kimball, Margy Ross. The Data WareHouse Toolkit Second Edition. Canada. : John Wiley and Sons, Inc., 2002.
9. Llerena, Michel Diaz. DATA MART PARA LA GESTIÓN DEL CONOCIMIENTO. 2007.
- 10.. Unidad 3. Diseñando una solución OLAP. Unidad 3. Diseñando una solución OLAP.
11. Business Intelligence - Almacenes de Datos - ETL. ETL-Tools.Info. ETL-Tools.Info. [En línea] 2006. [Citado el: 14 de 1 de 2011.] [http://etl-tools.info/es/bi/almacenedatos\\_esquema-constelacion.htm](http://etl-tools.info/es/bi/almacenedatos_esquema-constelacion.htm).
12. Aula de Economía. Aula de Economía.com. Aula de Economía.com. [En línea] 2002. [Citado el: 18 de 1 de 2011.] <http://www.auladeeconomia.com/mercados2.htm>.
13. Mimi.hu. Razones financieras. Razones financieras. [En línea] [Citado el: 1 de 18 de 2011.] [http://es.mimi.hu/economia/razones\\_financieras.html](http://es.mimi.hu/economia/razones_financieras.html).
14. Corporation, IBM. IBM Cognos TM1 Real-time visibility into complex financials. Ottawa, Canadá : s.n.<http://www-142.ibm.com/software/products/es/es/cognostm1>.



15. —. Warehouse Management with SAP Business One Optimize Inventory and Production Processes. United States : s.n., 2010.
16. Sitio oficial de SAP . Sitio oficial de SAP . Sitio oficial de SAP . [En línea] [Citado el: 20 de 1 de 2011.] <http://www.sap.com/sme/whysap/index.epx>.
17. Oracle . GL wand para ORACLE. GL wand para ORACLE. [En línea] [Citado el: 23 de 1 de 2011.] [http://www.quadrantix.com/index.php?option=com\\_content&view=article&id=76&Itemid=86&lang=en](http://www.quadrantix.com/index.php?option=com_content&view=article&id=76&Itemid=86&lang=en)
18. OpenERP. El ERP de software libre. OpenERP. El ERP de software libre. OpenERP. El ERP de software libre. [En línea] [Citado el: 23 de 1 de 2011.] <http://www.openersite.com/erp-openerp-modulos>.
19. Sitio oficial de OpenBravo. Sitio oficial de OpenBravo. Sitio oficial de OpenBravo. [En línea] [Citado el: 24 de 1 de 2011.] [http://www.openbravo.com/errors/not\\_found.php](http://www.openbravo.com/errors/not_found.php).
20. El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. [En línea] 18 de 10 de 2008. [Citado el: 26 de 1 de 2011.] <http://elecodelcontador.blogspot.com/2008/10/el-versat-sarasola-sistema-cubano-de.html>.
21. Sitio oficial de Asset. Sitio oficial de Asset. Sitio oficial de Asset. [En línea] [Citado el: 4 de 2 de 2011.] <http://www.assets.co.cu/assets.asp>.
22. Dario, Ing. Bernabeu Ricardo. HEFESTO: Metodología para la Construcción de un Data Warehouse. Córdoba : s.n., 2010.
23. Sitio oficial de my SQL. Sitio oficial de my SQL. Sitio oficial de my SQL. [En línea] [Citado el: 3 de 2 de 2011.] <http://dev.mysql.com/>.
24. Sitio oficial de Postgres. Sitio oficial de Postgres. Sitio oficial de Postgres. [En línea] 2011. [Citado el: 4 de 2 de 2011.] <http://www.postgresql.org/>.
25. El Rincon del BI. El Rincon del BI. El Rincon del BI. [En línea] 1 de 6 de 2010. [Citado el: 13 de 2 de 2011.] <http://churriwifi.wordpress.com/2010/06/01/>.
26. Sitio oficial de Jdox. Sitio oficial de Jdox. Sitio oficial de Jdox. [En línea] 17 de 8 de 2010. [Citado el: 14 de 2 de 2011.] <http://anibalgoicochea.com/2010/08/17/%C2%BFpalo-de-jedox-vs-bpc-de-sap/>.
27. 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**

28. Inmon, William H. (1996).- Building the Data Warehouse, Wiley Computer, New York.
29. Guía de instalación IBM Cognos TM1 Versión 9.5. Ottawa, Canadá : s.n., 2009.
30. Corporation, Oracle. ORACLE HYPERION FINANCIAL REPORTING. 2010.
31. —. HYPERION FINANCIA LREPORTING, CONSOLIDATION, AND ANALYSIS SOLUTION FOR GOVERNMENT. 2008.
32. AG., SAP. Streamline Your Business, Accelerate Profitable Growth A Single, Affordable Application Exclusively for Small Businesses. United States : s.n., 2009.
33. AG, SAP. Streamline Your Business, Accelerate Profitable Growth Using an Affordable Application Built for Smal Businesses. United States : s.n., 2009.
34. Inmon, W. H. Building the data warehouse. Indianapolis : Wiley Publishing, Inc., 2005.
35. Putting Metadata To Work In The Warehouse. Putting Metadata To Work In The Warehouse. [En línea] <http://jobfunctions.bnet.com/>.
36. Delivering Data Warehousing Business Results. [Citado el: 21 de 3 de 2011.] <http://www.techguide.com/>
37. A Practical Guide To Getting Started With Data Warehousing. [Citado el: 21 de 3 de 2011.] <http://www.techguide.com/>

## 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.

**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 consulta 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.

**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.