

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

**Centro de tecnologías y análisis de datos, Facultad 6
Laboratorio de Investigaciones de Gestión de Proyectos, Facultad 5**

**Propuesta de modelo de un almacén de datos para el control del tiempo y los
costos de proyectos**

Trabajo final presentado en opción al título de
Máster en Gestión de Proyectos Informáticos

Autor: Ing. Mabel Medina Rodríguez

Tutores:

Dr. Roberto Delgado Víctore

MSc. Julio César Díaz Vera

Ciudad de La Habana, Noviembre de 2012

DEDICATORIA

A mi abuelo querido, que Dios lo tenga en la gloria. Siempre te llevo conmigo.

AGRADECIMIENTOS

Le agradezco a Julio por guiarme durante todo el proceso de desarrollo de la tesis y por estar siempre presente para apoyarme e impulsarme a continuar.

Le agradezco a Pedro por toda su atención, consejos e interés en que todo saliera lo mejor posible.

Le agradezco al profesor Roberto por su apoyo y ayuda en función de lograr un resultado con calidad.

Le agradezco a mi familia, que aunque de lejos, me han apoyado e inspirado.

Le agradezco a mi novio Daniel por su amor, cariño y comprensión en todo este tránsito. Gracias por ser tan especial mi amor.

Le agradezco a esas amistades que han estado disponibles para mí en todo momento en que he necesitado apoyo y ayuda. Gracias Yoisita por ser una de ellas.

Le agradezco al centro DATEC por brindarme el espacio, el conocimiento y la oportunidad de llegar hasta aquí.

Una vez más agradezco a la UCI por constituir mi principal fuente de formación profesional y personal.

Gracias a todos aquellos que de una manera u otra han colaborado para lograr este resultado.

DECLARACIÓN JURADA DE AUTORÍA

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

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

Mabel Medina Rodríguez

Roberto Delgado Víctore

Julio César Díaz Vera

RESUMEN

La gestión de proyectos es una disciplina que requiere de técnicas y herramientas que permitan un adecuado control y seguimiento de la ejecución de los proyectos. Como parte del control es de vital importancia la presentación de informes y reportes que tributen a la toma de decisiones. Dichos reportes deben estar soportados por un almacén de datos que contenga las bases de cálculo de los indicadores a presentar. La mayor parte de los almacenes de datos para el control de proyectos se desarrollan desde cero y a la medida de las instituciones definiendo los indicadores según los objetivos estratégicos de las mismas, sin tener en cuenta elementos comunes asociados al control de los costos y del tiempo que pueden ser reutilizados en varios contextos para el control de los proyectos. La reutilización de estos elementos comunes contribuye a una disminución de los tiempos y costos de desarrollo de los almacenes de datos para el control de los proyectos. El presente trabajo propone un modelo de un almacén de datos para el control de proyectos en las áreas de tiempo y costos. El modelo se implementará siguiendo la línea de Arquitectura Guiada por Modelos, usando un sub-conjunto de los elementos definidos en la metodología *Data Warehouse Engineering Process* para la elaboración del Modelo Independiente de Plataforma, que posteriormente será transformado en un Modelo Dependiente de la Plataforma relacional utilizando el lenguaje *Atlas Transformation language* integrado al *Eclipse Modeling Framework* y usando los metadatos definidos por el *Common Warehouse Metamodel*.

PALABRAS CLAVE: almacén de datos, ATL, costos, CWM, gestión de proyectos, tiempo, toma de decisiones.

ABSTRACT

Project management is a discipline that requires techniques and tools for proper control and monitoring of the project implementation. As part of the control it is vital the presentation of report files that tax decisions making. Such reports must be supported by a data warehouse containing the bases of calculation of the indicators to be presented. Most data warehouses to control projects are developed from scratch and tailored to institutions, defining the indicators as strategic objectives of these institutions, regardless of common elements associated with the control of the cost and time that can be reused in various contexts to control projects. The reuse of these common elements contributes to a reduction in development time and costs of data warehouses for projects control. This paper proposes a model of a data warehouse for monitoring projects in the areas of cost and time. The model will be implemented

following the Model-Driven Architecture line, using a subset of the elements defined in Data Warehouse Engineering Process methodology for developing of the Platform Independent Model, which will then be transformed into a relational Platform Specific Model using the Atlas Transformation Language integrated into the Eclipse Modeling Framework and using metadata defined by the Common Warehouse Metamodel.

KEY WORDS: ATL, cost, CWM, data warehouse, decision making, project management, time.

ÍNDICE

INTRODUCCIÓN.....	9
Capítulo 1. Fundamentación teórica de los almacenes de datos para controlar la gestión de proyectos.....	16
Introducción.....	16
1.1 Análisis bibliométrico.....	16
1.2 Tendencias del análisis de información en la gestión de proyectos.....	17
1.3 Gestión de proyectos.....	18
1.4 Toma de decisiones en el control de proyectos.....	20
1.5 DWH.....	22
1.5.1 Enfoques de diseño de un DWH.....	22
1.5.2 Etapas de modelado de un DWH.....	23
1.5.3 Modelado de DWHs.....	23
1.5.4 Transformación a PSM.....	26
Conclusiones parciales.....	27
Capítulo 2. Construcción del modelo conceptual.....	29
2.1 Conceptualización de la propuesta.....	29
2.2 Consideraciones de diseño.....	30
2.2.1 Consideraciones para el diseño de metadatos de dimensiones comunes.....	31
2.2.2 Consideraciones para el diseño de metadatos de dimensiones estructurales.....	32
2.2.3 Análisis enfocados a diferentes niveles de una misma jerarquía.....	33
2.2.4 Manejo de jerarquía con diferentes niveles de profundidad.....	34
2.2.5 Distinción de instancias en cada nivel de una jerarquía.....	35
2.2.6 Consideraciones para el diseño de metadatos de hechos y dimensiones específicos por área.....	36
2.2.7 Medición de los costos y el financiamiento.....	36
2.2.8 Medición del tiempo.....	39
2.2.9 Medición de estadísticas del proyecto.....	40
2.3 Construcción del modelo conceptual.....	41
Conclusiones parciales.....	46
Capítulo 3. Construcción del modelo lógico.....	47
3.1 Pre-procesamiento del modelo multidimensional de origen.....	47
3.2 Construcción y montaje de meta-modelos origen y destino.....	49

3.3	Definición e implementación de reglas de transformación.	51
3.3.1	Implementación de reglas de transformación para modelo lógico relacional.	52
	Conclusiones parciales.....	58
	Capítulo 4. Análisis de los resultados de la investigación.....	59
4.1	Descripción del negocio del MPPCI.	59
4.2	Construcción del almacén de datos con métodos tradicionales en el caso de estudio del MPPCI.....	60
4.3	Construcción del almacén de datos aplicando el modelo basado en MDA en el caso de estudio del MPPCI.	61
4.3.1	Construcción del modelo conceptual del almacén de datos aplicando el modelo basado en MDA en el caso de estudio del MPPCI.	61
4.3.2	Construcción del modelo lógico del almacén de datos aplicando el modelo basado en MDA en el caso de estudio del MPPCI.	66
4.4	Análisis de los resultados de la aplicación del modelo al caso de estudio del MPPCI.	67
4.4.1	Análisis de la incidencia del modelo en las dimensiones de reutilización, multiplataforma y extensibilidad.	67
4.4.2	Análisis de la incidencia del modelo en los tiempos de desarrollo del almacén de datos con respecto a los tiempos de desarrollo del almacén de datos con métodos tradicionales.	68
4.4.3	Análisis de la incidencia del modelo en los costos de desarrollo del almacén de datos con respecto a los costos de desarrollo del almacén de datos con métodos tradicionales.	69
4.5	Análisis de la validez del modelo aplicado al caso de estudio, basado en la técnica de grupo focal.....	70
4.6	Impacto social del almacén de datos en el MPPCI.....	71
	Conclusiones parciales.....	71
	Conclusiones generales	72
	Trabajos Futuros.....	74
	Referencias bibliográficas	75

INTRODUCCIÓN

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado. Lo anterior se traduce a todas las acciones que deben realizarse para cumplir con una necesidad definida dentro de los plazos establecidos (PMI, 2009)¹.

Por tanto, un proyecto es una acción que tiene principio y fin, con metas que cumplir, que utiliza recursos identificados (humanos y materiales) durante su ejecución, que tiene un costo, que debe contar con la calidad requerida y que genera un resultado que responde a la necesidad por la que surgió (PMI, 2009).

Actualmente, los proyectos son cada vez más grandes y complejos debido a la cantidad de recursos que involucran y a las necesidades crecientes de informatización de la sociedad, razón por la cual se requiere una eficiente gestión que viabilice el correcto desarrollo del mismo en función de lograr los resultados que se propone. Independientemente de todas las áreas que componen la Gestión de Proyectos (PMI, 2009) como disciplina, se destacan la de costos y tiempo por su incidencia en los resultados del proyecto. El no controlar estas áreas en un proyecto puede provocar sobregiros en el presupuesto y excesos en los tiempos de ejecución establecidos.

Por sí sola puede tornarse engorrosa la responsabilidad de gestionar un proyecto de manera satisfactoria. Es por ello que en la mayoría de los casos se requiere de una herramienta o conjunto de herramientas que de manera automatizada, gestionen la información generada en el día a día del proyecto.

El objetivo de estas herramientas es gestionar proyectos, sea cual sea su clasificación y para ello brindan una serie de funcionalidades que aplican para cualquier ambiente de desarrollo de este tipo. Por la importancia que tienen para lograr resultados exitosos en un proyecto, las áreas de costos y tiempo se ven claramente reflejadas en las funcionalidades básicas de este tipo de herramientas. Las herramientas se pueden personalizar además en función del entorno en que van a ser usadas.

Cuando de proyectos se trata, las herramientas de gestión son necesarias para contribuir a obtener los resultados esperados brindando funcionalidades que permiten gestionar el proyecto de manera sistemática; aun así, solo manejan los datos actuales estructurados de manera relacional en función de su gestión. Carecen de los medios para la realización de análisis robustos y vistas flexibles que permitan al usuario analizar información para tomar decisiones estratégicas. No poseen además datos agregados para evaluar y monitorear resultados y tendencias a través de series de

¹ Para la mayoría de los conceptos relacionados con la gestión de proyectos se referencia el PMBOOK por constituir un documento rector en la elaboración de los materiales de la maestría de Gestión de proyectos de la Universidad de las Ciencias Informáticas (UCI).

tiempo, así como no proveen cuadros de mando personalizados con la última información a cualquier nivel que se requiera.

Esto dificulta notablemente el análisis en el tiempo de la información del proyecto gestionada día a día, lo cual incide negativamente a la hora de detectar fallas o errores que causaron deficiencias en la correcta ejecución del mismo y que afectaron sus resultados en un momento determinado de su ciclo de vida.

Al no poder detectar errores pasados en la ejecución de un proyecto, se entorpece el proceso de toma de decisiones lo que trae consigo la no corrección de los errores y cabe la posibilidad de cometer las mismas faltas en proyectos futuros, no contribuyendo con la mejora de la gestión propiamente.

En función de evitar los problemas anteriormente mencionados es necesario contar con un componente de análisis que permita controlar los proyectos a través de la presentación de indicadores que tributan directamente al control de las áreas de la gestión de proyectos. Este componente puede contribuir de manera significativa al mejoramiento del proceso de gestión, a través de la presentación al usuario final de información valiosa para la toma de decisiones en este contexto de la gestión de proyectos.

El referido debe estar soportado por una estructura de almacenamiento, preferentemente de diseño dimensional, que contenga las bases de cálculo requeridas para la presentación de los indicadores. Un almacén de datos es una de las estructuras adecuadas para este tipo de desarrollo al concebir un diseño dimensional y permitir el almacenamiento histórico de datos.

El problema básico del diseño de un almacén de datos consiste en obtener un conjunto de esquemas multidimensionales que permitan satisfacer los requisitos de análisis de los usuarios y que puedan ser mantenidos por las bases de datos operacionales existentes en la organización. En este caso los requisitos de los usuarios constituyen los principales indicadores a medir en función de costos y tiempo y los datos operacionales, aquellos relacionados con costos y tiempo manejados en las funcionalidades básicas de las herramientas de gestión de proyectos.

Los almacenes de datos para controlar proyectos actualmente se desarrollan partiendo de cero y modelando indicadores según los objetivos estratégicos de instituciones específicas, sin tener en cuenta un conjunto de indicadores básicos asociados a las áreas de tiempo y costos, comunes para diferentes contextos, lo que anula la facilidad de reutilización de estructuras ya modeladas para el desarrollo de almacenes de datos en el ámbito del control de proyectos. Lo anterior demuestra una falta de experiencia en este campo que extiende además los tiempos de desarrollo de

estos almacenes de datos para apoyar la toma de decisiones en el control de los proyectos, incidiendo negativamente además en los costos de este tipo de soluciones. La variedad de herramientas para gestionar proyectos incide negativamente también en la reutilización, pues en muchos casos manejan elementos asociados con costos y tiempo, sin embargo los metadatos asociados no son los mismos. Almacenes de datos con fines de análisis similares, que se deriven de diferentes herramientas de gestión de proyectos, resultan completamente diferentes aun cuando los requerimientos de análisis coinciden.

La reutilización de estructuras y el tiempo de desarrollo de almacenes de datos para el control de proyectos se ven afectados además por el uso de algunas tendencias donde el desarrollo se comienza partiendo del nivel lógico, atado a una plataforma en particular, lo que afecta los tiempos de desarrollo de los almacenes de datos para controlar los proyectos al no poder reutilizar las estructuras modeladas que contemplen requerimientos de análisis similares, sin implementar transformaciones entre plataformas en función de ahorrar tiempo y costos.

Es por ello que se plantea como **problema científico**:

¿Cómo disminuir los tiempos y costos de desarrollo de almacenes de datos para apoyar la toma de decisiones en el control de proyectos?

Objeto de investigación

Almacenes de datos para controlar proyectos.

Objetivo general

Desarrollar un modelo de un almacén de datos para controlar los proyectos, basado en la Arquitectura Guiada por Modelos (MDA) que permita disminuir los tiempos y costos de desarrollo de los almacenes de datos para apoyar la toma de decisiones en el control de proyectos.

Objetivos específicos

1. Realizar una investigación sobre almacenes de datos para controlar proyectos que permita identificar las principales tendencias y líneas de trabajo en este tipo de desarrollo.
2. Definir un modelo conceptual de un almacén de datos para controlar los indicadores de gestión de tiempo y gestión de costos que pueda ser reutilizado en varios contextos de toma de decisiones en el control de proyectos.
3. Obtener un modelo lógico relacional a partir del modelo conceptual definido que pueda ser desplegado en plataformas relacionales.

4. Validar el resultado obtenido para comprobar la utilidad del modelo.

Campo de acción

Desarrollo de almacenes de datos basados en MDA para el control de tiempos y costos de proyectos.

Tipo de investigación

- Correlacional: Se realizan comparaciones entre la realización de un caso de estudio sin aplicar el modelo y el mismo caso de estudio aplicando el modelo.

Métodos de investigación

- Método hipotético-deductivo para elaborar la hipótesis de investigación y proponer líneas de trabajo a partir de resultados parciales.
- Método histórico lógico para el estudio crítico de los trabajos anteriores.
- Método analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.
- Método de modelado para el diseño conceptual de la propuesta.
- Método cuasi-experimental para comprobar la utilidad de los resultados obtenidos.
- Grupo focal para corroborar la validez de los resultados del caso de estudio.

Hipótesis

Si se desarrolla un modelo de un almacén de datos para controlar los proyectos, basado en MDA, disminuirán los tiempos y costos de desarrollo de almacenes de datos para apoyar la toma de decisiones en el control de proyectos.

Operacionalización de las variables

Variable independiente	Dimensión	Indicadores
Modelo de un almacén de datos para controlar los proyectos, basado en MDA.	Reutilización.	Por ciento de paquetes estrellas del modelo aplicables al caso de estudio.
	Multiplataforma.	Aplicación del modelo en entornos con diferentes plataformas de bases de datos.

	Extensibilidad.	Cantidad de paquetes estrellas añadidos al modelo según especificidades del caso de estudio.
Variables dependientes	Dimensión	Indicadores
Tiempos de desarrollo de almacenes de datos para apoyar la toma de decisiones en el control de proyectos.	Tiempo.	Duración en meses.
Costos de desarrollo de almacenes de datos para apoyar la toma de decisiones en el control de proyectos.	Costos.	Monto monetario.

Aporte práctico de la investigación

El modelo de almacén de datos basado en MDA para controlar los proyectos en las áreas de tiempo y costos constituye un marco de referencia reutilizable y extensible en el diseño de almacenes de datos cuyo objetivo sea controlar proyectos, independientemente del área de trabajo donde se desarrolle. Al seguir la línea MDA, el modelo es además multiplataforma permitiendo su aplicación en entornos con diferentes plataformas de bases de datos. Las dimensiones de reutilización, multiplataforma y extensibilidad presentes en el modelo reducen considerablemente los tiempos y costos de desarrollo de los almacenes de datos para controlar proyectos viabilizando de esta manera el proceso de construcción de este tipo de soluciones.

Listado de publicaciones, eventos y avales de la investigación

1. Ing. Medina Rodríguez, Mabel. (2008) Tecnologías para un control de flujo web más efectivo. Serie Científica de la UCI, 2008.
2. Ing. Medina Rodríguez, Mabel. (2008) ¿Por qué usar Spring Web Flow?. Serie Científica de la UCI, 2008.
3. Ing. Medina Rodríguez, Mabel. (2009) Comunicación homogénea en sistemas heterogéneos a partir del uso de estándares de información. Memorias de la Novena semana tecnológica en FORDES, 2009.

4. Ing. Medina Rodríguez, Mabel. (2009) Experiencias de almacenes de datos con PostgreSQL, Primer Taller Nacional de PostgreSQL, 2009.
5. Ing. Medina Rodríguez, Mabel. (2010) Propuesta de arquitectura para almacenes de datos. V Conferencia Científica UCIENCIA, 2010.
6. Ing. Medina Rodríguez, Mabel. (2010) Metodología para el desarrollo de soluciones de almacenes de datos e inteligencia de negocio en CENTALAD. V Conferencia Científica UCIENCIA, 2010.
7. Ing. Medina Rodríguez, Mabel. (2011) Implementación de vistas materializadas en PostgreSQL. FORDES, 2011.
8. Ing. Medina Rodríguez, Mabel. (2011) La Dirección Integrada de Proyecto como Centro del Sistema de Control de Gestión en el Ministerio del Poder Popular para la Comunicación y la Información. Caracas, Venezuela, 2011.
9. Ing. Medina Rodríguez, Mabel. (2012) Almacén de datos para controlar alcance, costo y tiempo de proyectos. FORDES, 2012.

Estructura capitular

Capítulo 1. Fundamentación teórica de los almacenes de datos para controlar la gestión de proyectos: Se realiza un estudio de las principales tendencias en el análisis de información para controlar los proyectos y de los conceptos fundamentales de la gestión de proyectos. Se identifican además los indicadores básicos para controlar las áreas de tiempo y costos de los proyectos basados en los datos manejados por las funcionalidades básicas de las herramientas de gestión. Finalmente se definen los principales conceptos asociados a los almacenes de datos y su representación conceptual y lógica a través de modelos.

Capítulo 2. Construcción del modelo conceptual: Se conceptualizan las actividades a tener en cuenta para el modelado basado en MDA. Seguidamente se analizan un conjunto de buenas prácticas y consideraciones a tener en cuenta en el modelado de un almacén de datos para controlar la gestión de proyectos en las áreas de tiempo y costos. Se presenta y explica además la propuesta de modelo conceptual de un almacén de datos para controlar los tiempos y costos de proyectos. Se hace uso de estándares que permiten la representación de las principales particularidades de los almacenes de datos en el nivel de abstracción conceptual.

Capítulo 3. Construcción del modelo lógico: Se transforma el modelo conceptual del capítulo anterior al modelo lógico dependiente de la plataforma relacional. Se hace uso de estándares internacionales que proponen metadatos para representar las principales particularidades de los almacenes de datos, así como de reglas de transformación y lenguajes de programación desarrollados para este fin.

Capítulo 4. Análisis de los resultados de la investigación: Se aplica la propuesta a un caso de estudio relacionado con el Ministerio del Poder Popular para la Comunicación y la Información de la República Bolivariana de Venezuela (MPPCI) con el objetivo de validarla. Se analizan los resultados del caso de estudio desarrollado con métodos tradicionales y aplicando el modelo. Se realiza además la técnica de grupo focal para corroborar la validez de los resultados del caso de estudio. Se hace alusión también a la validación de la propuesta emitida por el Departamento de Almacenes del Centro de tratamiento y análisis de datos (DATEC).

Capítulo 1. Fundamentación teórica de los almacenes de datos para controlar la gestión de proyectos

Introducción

El presente capítulo realiza primeramente un estudio bibliométrico separado por categorías de nivel y tiempo de la bibliografía consultada. A continuación se analizan las principales tendencias del análisis de información en la gestión de proyectos, se abordan temas relacionados con la gestión de proyectos propiamente y se presentan indicadores que permiten el control de los costos y el tiempo de proyectos. Se estudian las principales características de los almacenes de datos y finalmente se realiza una investigación sobre los estándares aplicables para la construcción de almacenes basada en MDA.

1.1 Análisis bibliométrico.

A continuación se realiza un estudio de la bibliografía consultada teniendo en cuenta un conjunto de criterios y destacando aquellas de los últimos tres años.

Tabla 1. Análisis bibliométrico

	Últimos 3 años	Años anteriores
Libros y monografías	6	10
Tesis de doctorados	3	2
Tesis de maestrías	3	1
Proyectos de fin de carrera	2	
Artículos en Revistas referenciadas en Web of Science, SCOPUS	5	1
Artículos en otras revistas internacionales	5	4
Memorias de eventos		1
Artículos publicados en la web	1	1
Reportes técnicos y conferencias	10	1

A raíz del estudio anterior se concluye que el 63% de la bibliografía consultada es de los últimos tres años. Se evidencia el uso de revistas de primer nivel y la consulta de doctorados, maestrías y libros relacionados con los temas que se tratan. Se consultan además reportes técnicos para la familiarización con las tecnologías a usar.

1.2 Tendencias del análisis de información en la gestión de proyectos.

La tendencia a construir sistemas de información en el mundo va aumentando considerablemente a raíz del incremento de datos en las instituciones, los que deben ser presentados y analizados con el objetivo de tomar decisiones para mejorar el funcionamiento y la gestión interna de las propias empresas. El análisis de información en función de la toma de decisiones puede ser llevado a cabo a partir de la construcción de sistemas de información desarrollados con este fin, conocidos como Soluciones para la Toma de Decisiones (DSS) (Power, y otros, 2009).

Dichas soluciones pueden desarrollarse para el análisis de cualquier tipo de información. En el ámbito de la gestión de proyectos particularmente resaltan los resultados de la Solución para la Gestión de Proyectos y Recursos *Artemis Views* de España, que entre sus funcionalidades contempla la presentación de datos de costos y análisis de variaciones, perspectiva financiera de los calendarios de proyectos, actividades y recursos, la conservación de hasta 99 versiones diferentes de un proyecto, la realización de análisis exhaustivos del tipo “qué pasaría si”, gráficos de cantidades y toma de decisiones informada, en cualquier momento y lugar (Artemis International Solutions Corporation, 2001).

Se destacan en este sector además las soluciones de *Microsoft Enterprise Project Management* (EPM) que proveen servicios de Inteligencia de negocios y *reporting*, ayudando a las empresas a ganar visibilidad y a soportar decisiones para una gestión proactiva del trabajo (Microsoft, 2011).

Oracle también ha innovado en este campo de la gestión de proyectos con su *Primavera Contract Management, Business Intelligence Publisher Edition*, que mantiene los proyectos en construcción en tiempo y presupuesto a partir de un completo control de su ejecución (Oracle, 2011).

Las soluciones anteriores se enfocan en muchos de los elementos importantes a la hora de controlar la gestión de proyectos, pero constituyen soluciones privativas que no pueden ser extendidas con nuevos indicadores propios de la institución donde se implante, lo cual restringe el dominio de análisis a los indicadores contemplados en las mismas. Se centran además en tecnologías de bases de datos específicas que pudieran resultar privativas o simplemente no solicitadas por el cliente en el proyecto que se desarrolle.

A raíz de esto se evidencia la necesidad de realizar una investigación sobre los mecanismos de análisis en el ámbito del control de proyectos que permita la construcción de una estructura que soporte la presentación de indicadores claves para el control de los proyectos, que pueda ser extendida además con indicadores específicos de la empresa donde se implante y que provea facilidades de despliegue

en diferentes plataformas de base de datos según los requerimientos de las empresas.

Los indicadores a modelar en dicha estructura deben estar soportados por los datos generados en los procesos de gestión de proyectos.

1.3 Gestión de proyectos.

El desempeño y el sustento económico de instituciones nacionales como: DESOFT y Segurmática así como de otras internacionales como: Oracle, Google y Microsoft, se basan en la ejecución de proyectos productivos, sociales, culturales o de otra naturaleza. Esto determina que la gestión de proyectos esté cada día más presente en la mayoría de las instituciones, pues garantiza la subsistencia de la propia institución en un ambiente cambiante y dinámico. La comprensión de las características fundamentales asociadas a un proyecto resulta de vital importancia para los directivos de las instituciones de este tipo.

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único (PMI, 2009). Los proyectos deben ser gestionados a lo largo de todo su ciclo de vida con el objetivo de garantizar la obtención de los resultados propuestos cumpliendo con el tiempo definido, con la calidad requerida, con los costos establecidos y con el alcance definido. Para gestionar un proyecto con éxito, se debe comprender qué puede ir mal (para evitar esos problemas) y cómo hacerlo bien (Pressman, 2005).

La gestión de proyectos como concepto es la aplicación de conocimiento, habilidades, herramientas y técnicas a las actividades de un proyecto para cumplir con los requisitos del proyecto (PMI, 2009).

Los líderes de proyectos en su gestión deben balancear los requerimientos de tiempo, calidad, costos y alcance, al constituir éstas, áreas fundamentales para la obtención de resultados satisfactorios dentro de un proyecto (Ver Figura 1).



Figura 1. Áreas fundamentales en la gestión de proyectos

La gestión de un proyecto puede dificultarse si se implementa de forma manual, pues suele generarse un gran volumen de documentación que entorpece el correcto

funcionamiento del proceso en cuestión. Es por ello que se implementan sistemas de apoyo a la gestión de proyectos que automatizan todo el trabajo manual y permiten darle seguimiento continuo a la ejecución del proyecto.

Un sistema de gestión de proyectos es el conjunto de herramientas, técnicas, metodologías, recursos y procedimientos utilizados para gestionar un proyecto. Puede ser formal o informal y ayuda al director del proyecto a gestionar de forma eficaz un proyecto hasta su conclusión.

En las herramientas de gestión de proyectos más usadas (B. Tayntor , 2010) (A. Yinusa, et al., 2010), (K. Chemuturi , et al., 2010), (Maigua, et al., 2012), independientemente del nivel de detalle o complejidad implementado, existen mecanismos para darle seguimiento al progreso del proyecto.

El progreso del proyecto se puede controlar a partir de la gestión de las áreas reflejadas en la Figura 1, al representar éstas elementos claves dentro del proyecto. No obstante, las herramientas de gestión en muchos casos, presentan funcionalidades asociadas con otras áreas de la gestión de proyectos que contribuyen también al cumplimiento de los objetivos del proyecto.

En función de las cuatro áreas fundamentales de un proyecto, a continuación se presentan un conjunto de consideraciones asociadas con las herramientas de gestión:

- A pesar de la importancia que reporta la gestión de la calidad en la gestión de proyectos, no se incluyen funcionalidades asociadas con esta área en particular en las herramientas de gestión de proyectos en estado virgen. Sin embargo el control de la calidad se lleva a cabo aun sin registrar evaluaciones cualitativas en las herramientas de gestión de proyectos a partir del uso de listas de chequeo basadas en estándares tanto nacionales como internacionales, entre otros mecanismos existentes para este fin. En muchos casos se extienden las funcionalidades de la herramienta de gestión para incluir la gestión de la calidad.
- Por su parte la gestión de alcance se basa en los requisitos solicitados por el cliente, que tampoco son reflejados en la mayoría de las herramientas de gestión de proyectos. Gestionar el alcance basado en las tareas que materializan los requisitos no constituye un cálculo fiable pues no todas las tareas tienen la misma complejidad, prioridad e incidencia en la obtención de resultados dentro del proyecto en cuestión. No obstante, cálculos asociados a las tareas pueden brindar aproximaciones relacionadas con estadísticas agregadas del proyecto, que pudieran ser de interés tanto para el líder de proyecto como para los dirigentes de la empresa.

- En el caso del tiempo y los costos, las herramientas de gestión en su mayoría presentan funcionalidades asociadas con estas áreas, quedando registrados los datos correspondientes a dicha gestión en las propias herramientas.

De manera general, las herramientas de gestión apoyan los procesos de gestión contribuyendo a la obtención de resultados satisfactorios, los que están condicionados además por la posibilidad de tomar de decisiones dentro del proyecto. La toma de decisiones no se contempla entre los objetivos de las herramientas de gestión de proyectos, no obstante participan dentro de este proceso en calidad de fuentes de datos operacionales.

1.4 Toma de decisiones en el control de proyectos.

La generación de informes para apoyar la toma de decisiones constituye una funcionalidad de gran importancia en el ámbito del control de proyectos, incluyendo información del proyecto, del portafolio de proyectos y ofreciendo perspectivas de cuadro de mando.

Los informes deben permitir el control de los proyectos facilitando el proceso de toma de decisiones con el objetivo de mejorar la ejecución a lo largo de todo el ciclo de vida del proyecto. Esto implica la comparación entre diferentes momentos de un mismo proyecto para garantizar la corrección de errores pasados y adquirir experiencia para situaciones futuras.

El controlar el proyecto en el tiempo a partir de una evaluación de los resultados frente a sus objetivos se lleva a cabo mediante los denominados indicadores de gestión, factores que constituyen la clave de éxito en este tipo de frente (Franceschini , et al., 2007), (J. Thomson, 2005).

Los indicadores están directamente asociados a los resultados del proyecto, que responden directamente a los objetivos estratégicos de una organización. Todas las organizaciones tienen sus propias particularidades, no obstante, aquellas que ejecutan proyectos, entre sus objetivos contemplan, la ejecución de sus proyectos cumpliendo con el tiempo y los costos establecidos.

Así como las herramientas de gestión de proyectos presentan funcionalidades comunes basadas en las áreas de tiempo y costos, existen indicadores de gestión básicos que responden al control del tiempo y de los costos que pueden ser usados en el control de los proyectos independientemente de la organización que lo ejecute.

Algunos autores como (Delgado Víctore, 2006), (Earned value management in a data warehouse project, 2006) proponen indicadores como los siguientes:

- Costo planificado del trabajo planificado acumulado y en el período (CPTP): Indica los costos que deberían haberse ejecutado hasta el momento. $(CP * TP)$

- Costo planificado del trabajo realizado acumulado y en el período (CPTR): Indica el costo total presupuestado por el trabajo realizado en por ciento hasta el momento. $(CP * TR)$
- Costo real del trabajo real ejecutado acumulado y en el período (CRTR): Indica los costos reales. $(CR * TR)$
- Varianza de la programación (VP): Determina si el proyecto está en tiempo de acuerdo a su cronograma. $(CPTR - CPTP)$
- Varianza de los costos (VC): Determina si los gastos del proyecto están de acuerdo a su presupuesto. $(CPTR - CRTR)$
- Índice de rendimiento de la programación (IRP): Mide eficiencia en la utilización del tiempo. $(CPTR / CPTP)$
- Índice de rendimiento de los costos (IRC): Mide eficiencia en la utilización de los recursos. $(CPTR / CRTR)$
- Productividad (P): Capacidad del sistema para elaborar los productos que son requeridos, haciendo uso óptimo de los recursos disponibles. $(IRP * IRC)$
- Porcentaje planificado de ejecución del presupuesto (PIP): Porcentaje planificado de ejecución del presupuesto con relación al costo total planificado (CPF) de acuerdo con la línea base y en función del avance físico. $(CPTP * 100 / CPF)$
- Porcentaje de ejecución real (PIPR): Porcentaje de ejecución real con relación al CPF de acuerdo con la línea base y en función del avance físico. $(CPTR * 100 / CPF)$
- Porcentaje real de ejecución del presupuesto (PIR): Porcentaje real de ejecución del presupuesto con relación al costo total esperado (CEF). $(CRTR * 100 / CEF)$
- Flujo de caja real (FCR): Financiamiento real percibido menos los costos reales de producir el bien o servicio. $(FR - CR)$
- Flujo de caja planificado (FCP): Financiamiento planificado menos los costos planificados. $(FP - CP)$
- Ejecución financiera (EF): Determina una relación entre el financiamiento asignado y el costo real. (CR / FR)
- Ejecución presupuestaria (EP): Determina una relación entre el costo planificado y el costo real. (CR / CP)
- Porcentaje de financiamiento (PF): Determina una relación entre el financiamiento planificado y el financiamiento asignado. (FR / FP)
- Porcentaje de ejecución física (PEF): Determina una relación entre la ejecución física planificada y la ejecución física real. (TR / TP)

Que basan su cálculo en las siguientes variables:

- Costo planificado (CP): Costo planificado acumulado y en el período.
- Trabajo planificado (TP): Trabajo planificado acumulado y en el período.
- Costo real (CR): Costo real acumulado y en el período.
- Trabajo real (TR): Trabajo real acumulado y en el período.
- Financiamiento real (FR): Financiamiento real acumulado y en el período.
- Financiamiento planificado (FP): Financiamiento planificado acumulado y en el período.

A partir de una propuesta de indicadores de tiempo y costos soportados por las bases de cálculo que ofrecen las herramientas de gestión de proyectos para estas dos áreas, es posible describir el diseño tipo de una estructura que permita el almacenamiento histórico de los datos necesarios para el cálculo de dichos indicadores, teniendo como fuentes de datos operacionales las herramientas de gestión de proyectos.

La estructura de almacenamiento tipo más idónea para este fin es un almacén de datos o *data warehouse* (DWH).

1.5 DWH.

La definición de almacén de datos más extendida, es la propuesta por Bill Inmon: "Un DWH es una colección de datos orientados al dominio, integrados, no volátiles y variables en el tiempo, organizados para dar apoyo al proceso de toma de decisiones" (Inmon, 2005).

Un DWH se desarrolla enfocado a una entidad en su totalidad y puede tomar mucho tiempo desarrollar esta estructura teniendo en cuenta todas las áreas de una empresa.

Existen variantes de diseño que constituyen básicamente subconjuntos de un DWH enfocados a un área de interés particular de una entidad. Estos son los denominados mercados de datos.

1.5.1 Enfoques de diseño de un DWH.

Para diseñar un almacén de datos es necesario realizar un levantamiento de información que puede comprender varias perspectivas como son:

- Datos: Se tienen en cuenta los datos presentes en el sistema fuente para modelar el almacén de datos (*Top-down*).
- Usuarios: Se tienen en cuenta las necesidades de los usuarios finales de la solución (*Bottom-up*) (Luján Mora, 2005).

Por sí solas, ambas perspectivas pueden acometer la construcción de un almacén de datos, no obstante su unión resulta mucho más acertada al definir los requisitos de los usuarios finales de la solución condicionados por su disponibilidad en las fuentes operacionales.

En el caso particular de este trabajo, se tienen en cuenta ambas perspectivas al concebir el conjunto de indicadores básicos para la gestión de proyectos en sus áreas fundamentales que responden a los datos manejados por las funcionalidades básicas de las herramientas de gestión de proyectos más usadas.

1.5.2 Etapas de modelado de un DWH.

Dependiendo de las características y especificidades del negocio que se trate, el proceso de diseño del DWH debe organizarse en función de definirlo desde una perspectiva conceptual, lógica y física.

- Conceptual: Define el DWH desde un punto de vista conceptual, es decir, desde el mayor nivel de abstracción y contiene únicamente los objetos y relaciones más importantes.
- Lógico: Abarca aspectos lógicos del diseño del DWH, como la definición de las tablas y claves, etc. Se especifica el almacén de datos en función de la plataforma de base de datos correspondiente.
- Físico: Define los aspectos físicos del DWH, como el almacenamiento de las estructuras lógicas en diferentes discos o la configuración de los servidores de bases de datos que mantienen el almacén de datos (Luján Mora, 2005).

La meta del presente trabajo es definir un almacén de datos extensible que constituya un marco de referencia reutilizable para la construcción de almacenes de datos para controlar proyectos, que facilite el cambio de plataformas de base de datos y que disminuya los costos y tiempos de desarrollo de almacenes de datos para controlar proyectos.

En función de alcanzar la meta anterior es menester definir la variante de modelado más acertada en este campo.

1.5.3 Modelado de DWHs.

El modelado multidimensional (MD) es la base para los DWH, las bases de datos multidimensionales y las aplicaciones de Procesamiento Analítico en Línea (OLAP). Muchas propuestas han surgido para capturar las principales propiedades MD de un almacén de datos en su nivel conceptual mediante la abstracción de detalles de la plataforma donde el DWH será implementado. La abstracción de plataforma brinda la

posibilidad de desplegar el DWH en diferentes plataformas de bases de datos posteriormente.

Para abordar esta temática se defiende el uso del marco de trabajo MDA (*Model Driven Architecture*) (Object Management Group, 2003). MDA además de permitir el desarrollo integral y formal de modelos usando una notación estándar, también permite detallar transformaciones para obtener un DWH ajustado a los criterios de diseño, aportando así un entorno de desarrollo para DWHs integrado, bien estructurado y completo, permitiendo reducir el tiempo de su desarrollo.

Las principales ventajas del enfoque MDA son:

1. La complicada tarea del diseño de un Almacén de Datos es abordada de manera sistemática y bien estructurada. Al realizarse desde una perspectiva conceptual es totalmente independiente de los detalles de implementación.
2. El diseño sigue un sistema de modelado integrado, evitando de este modo problemas de integración.
3. La implementación del Almacén de Datos puede ser derivada del desarrollo previo de un PIM (*Platform Independent Model*) y la posterior aplicación de las correspondientes reglas de transformación para la obtención del PSM (*Platform Specific Model*).
4. Utiliza UML, un lenguaje de modelado orientado a objetos comúnmente aceptado y usado, lo cual evita el aprendizaje de nuevos modelos a los desarrolladores y sus notaciones correspondientes para el modelado Multidimensional (MD). Además, esta ventaja de utilizar UML para abordar el diseño de cada componente del Almacén de Datos de manera integrada, proporciona también una fácil interoperabilidad entre las diferentes capas (Sánchez Gutiérrez, 2010).

Ajustándose a los elementos anteriormente expuestos surge el método conocido como *el Data Warehouse Engineering Process (DWEPE)*, propuesto en la tesis doctoral de Sergio Luján Mora (Luján Mora, 2005). El DWEPE es un método orientado a objetos, independiente de cualquier implementación específica, ya sea relacional, multidimensional, etc y permite la representación de todas las etapas del diseño de un DWH. Está basado en UML que es el Lenguaje Unificado de Modelado (Object Management Group (OMG), 2009).

El presente trabajo hará uso del perfil UML referenciado anteriormente para modelar conceptualmente los metadatos del almacén de datos para controlar la gestión de proyectos. Dicho perfil no viene incorporado en las nuevas versiones de UML, por tanto se usa para el modelado un *add-in* creado para el montaje en la herramienta *Rational Rose* (International Business Machines, 2012), que contiene todos los

estereotipos, valores etiquetados y restricciones de esta extensión de UML para modelado multidimensional.

A partir del uso del perfil anterior se obtiene un modelo conceptual contenedor de objetos con sus respectivas asociaciones. Dicho modelo conceptual, representativo de un PIM, debe ser transformado a un tipo de PSM cuyas reglas de transformación y resultados correspondientes, dependen de la plataforma subyacente donde se despliegue el PIM.

De manera general, el PIM contiene un conjunto de metadatos que pueden ser compartidos entre diferentes implementaciones de PSM.

Precisamente para acometer el objetivo anterior surge el *Common Warehouse Metamodel* (CWM) como alternativa para resolver el manejo de metadatos y el problema de la integración (Poole, y otros, 2003).

El CWM extiende el lenguaje UML, en el sentido que cada meta-clase de CWM se deriva directa o indirectamente de las meta-clases de UML. En otras palabras, el CWM define un lenguaje de dominio específico para especificar modelos de DWH. Todos los modelos de CWM son expresados en UML e implementados según la semántica de MOF.

El meta-modelo MOF define un conjunto de niveles de abstracción que constituye la arquitectura de metadatos de OMG (A Standard for Representing Multidimensional Properties: The Common Warehouse Metamodel (CWM), 2002) y define a qué nivel se describirán los metadatos de un almacén de datos.

El contemplar como objetivo fundamental del presente trabajo la construcción de un modelo para un almacén de datos que permita controlar la gestión de proyectos y al tener identificados los principales indicadores que tributan al citado control así como sus bases de cálculo, el modelo quedará definido en el nivel 1 de abstracción.

El CWM está organizado en 21 paquetes separados, agrupados en cinco capas escalables por medio de roles similares. El presente trabajo se centra en la capa de Recursos donde se definen un conjunto de estereotipos, valores etiquetados y restricciones que permiten representar los datos en diferentes formatos en función de la plataforma de base de datos a utilizar. Teniendo en cuenta el tipo de plataforma, cada PSM en particular requiere de un conjunto de reglas y herramientas para su obtención a partir del PIM multidimensional.

A continuación se hace referencia a los aspectos relacionados con las transformaciones del PIM al PSM.

1.5.4 Transformación a PSM.

Con el objetivo de obtener un PSM se hará uso de un conjunto de reglas aplicables a las clases del PIM como son los hechos, las dimensiones y las clases bases, así como también se aplicarán reglas a las relaciones del PIM del tipo: hecho – dimensión, dimensión – base y base – base. Para más información remitirse a (Sánchez Gutiérrez, 2010).

Las reglas a definir se basan en los metadatos contenidos en el meta-modelo relacional del CWM pues la relacional es una de las plataformas más usadas actualmente en este tipo de desarrollo debido a que muchos servidores de bases de datos libres y aptos para ser usados, son precisamente de tecnología relacional.

La especificación de las reglas se realiza por medio de un lenguaje de transformación que permite definir patrones de transformación entre los meta-modelos y en el que se establecen las correspondencias entre el meta-modelo fuente y el meta-modelo destino. Las reglas a aplicar formarán parte de transformaciones verticales pues la fuente y el destino en este caso son modelos con diferentes niveles de abstracción (PIM-PSM).

En función de acometer esta tarea se estudiaron un conjunto de enfoques de desarrollo y las herramientas asociadas con el objetivo de seleccionar la más adecuada para implementar las transformaciones pertinentes.

Entre los enfoques explicados en (López, y otros, 2009) se seleccionó el híbrido como el más adecuado para el presente desarrollo por combinar toda la gama de características del resto de los enfoques y nuevas facilidades como la de usar diferentes meta-modelos en el origen y en el destino.

En el enfoque híbrido se contemplan un conjunto de lenguajes, herramientas y estándares que con mayores y menores ventajas brindan mecanismos para las transformaciones entre modelos (López, y otros, 2009).

El lenguaje seleccionado fue el ATL (*ATLAS Transformation Language*) por las razones expuestas a continuación:

- Está basado en los lineamientos definidos por QVT (*Query/View/Transformation*), estándar creado por la OMG para este fin que constituye una referencia para el desarrollo de la mayoría de los lenguajes de transformación de modelos (López, y otros, 2009).
- La actual herramienta de desarrollo ATL está basada en el IDE Eclipse como un *plug-in* de esta plataforma (ADT (*ATL Development Tools*)). Al desarrollarse sobre esta plataforma con varios años de desarrollo y uso, se asegura cierta robustez para las herramientas (López, y otros, 2009).

- Permite definir pre y post-condiciones en *Object Constraint Language* (OCL) (OMG, 2012), resultando sumamente expresivo y de fácil escritura para sus usuarios (García Benítez, 2010).
- Crea enlaces de trazabilidad con la ejecución de cada regla (García Benítez, 2010).
- Posee abundante y actualizada documentación en (ATL, 2012).

En ATL un modelo fuente se transforma en un modelo destino mediante una definición de transformación escrita en ATL, que también es un modelo. Los modelos fuente, destino y la definición de la transformación, responden a sus meta-modelos respectivos y a su vez, todos los meta-modelos se ajustan a MOF (Usando ATL en la transformación de modelos multidimensionales temporales, 2007) (Ver Figura 2).

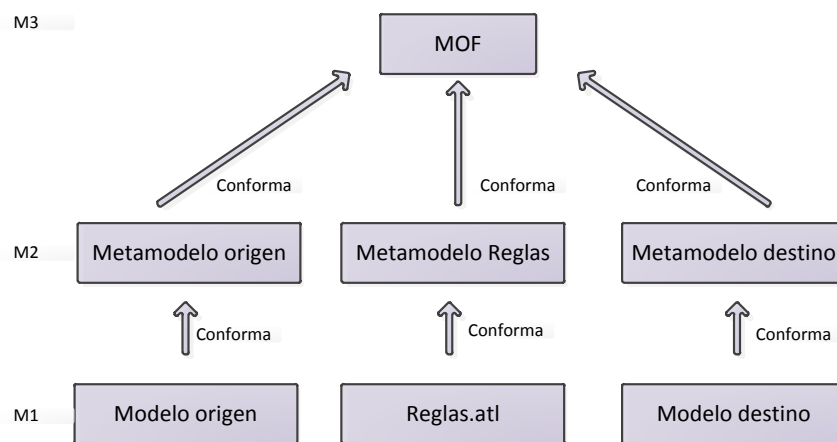


Figura 2. Lenguaje ATL

Las herramientas disponibles en ADT para la transformación de modelos son principalmente dos: el núcleo de funcionalidades ATL, que incluye el motor de transformación y las facilidades de administración de modelos.

La parte básica de ATL incluye todos los componentes requeridos para configurar y ejecutar transformaciones, en particular, el EMF (*Eclipse Modelling framework*) (Eclipse Modeling Framework Project (EMF), 2012) que permite manejar modelos definidos de acuerdo a la semántica *Ecore* (Eclipse Modeling Framework Project (EMF), 2012).

Conclusiones parciales

El estudio realizado partió de los conceptos fundamentales de la gestión de proyectos, identificando las áreas de gestión de tiempo y gestión de costos como fundamentales en la ejecución satisfactoria de un proyecto. Los datos asociados a estas áreas son recogidos en las herramientas de gestión de proyectos más usadas. Dichos datos constituyen las bases de cálculo de los indicadores básicos presentados para el control del tiempo y los costos de proyectos.

Para el almacenamiento de los indicadores presentados se estudiaron los DWHs, como la estructura más idónea para almacenar con un diseño dimensional las bases de cálculo de los indicadores a visualizar.

La aplicación de técnicas basadas en MDA al desarrollo de almacenes de datos ha reportado importantes resultados que hacen posible su utilización para el desarrollo de un almacén de datos para el control de proyectos. Para ello se seleccionó el método DWEP que propone un perfil UML para modelar a un nivel de abstracción conceptual (PIM) estructuras multidimensionales. Para obtener el modelo lógico (PSM) se determina el uso del *plug-in* ADT sobre el EMF, herramientas que en conjunto permiten la implementación de reglas de transformación verticales basadas en las diferentes extensiones que ofrece el meta-modelo CWM.

Capítulo 2. Construcción del modelo conceptual

Introducción

El presente capítulo primeramente define las actividades para la construcción del modelo siguiendo el enfoque MDA. Detalla en cada una de las actividades las herramientas a usar así como los elementos de salida correspondientes. Ofrece además una breve descripción de las actividades.

A continuación se desarrolla la primera de las actividades presentadas exponiendo un conjunto de experiencias y buenas prácticas para este tipo de modelado.

2.1 Conceptualización de la propuesta.

Para obtener la propuesta de modelo de un almacén de datos para controlar el tiempo y los costos de proyectos desarrollada con los principios de MDA, es menester identificar las principales actividades a desarrollar, las herramientas y los elementos de salida en cada caso. La siguiente figura diagrama los aspectos citados anteriormente.

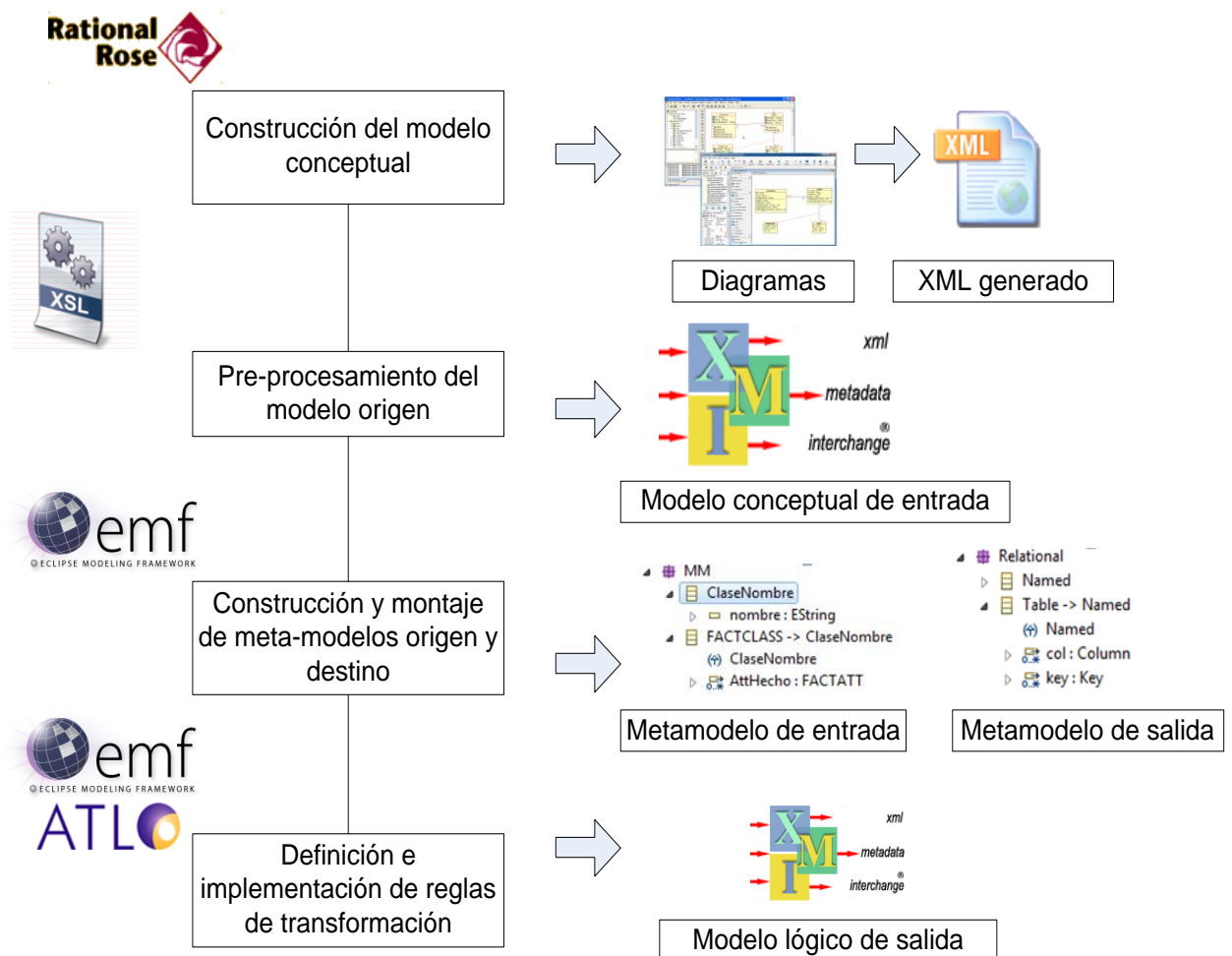


Figura 3. Flujo de actividades

A continuación se describen brevemente las actividades presentadas:

- Construcción del modelo conceptual: Se modelan los conceptos multidimensionales en el *Rational Rose* a partir de las necesidades del usuario, siguiendo los principios definidos en DWEP (teniendo en cuenta los tres niveles de profundidad) y llevados a la práctica con el *add-in* creado para integrar al *Rational Rose*. Una vez validado (opción *MD Validate*) y sin errores el diseño conceptual definido, se genera como resultado de este paso un fichero XML (opción *XML Generate*). El fichero generado contiene de manera estructurada los elementos modelados gráficamente en el *Rational Rose*.
- Pre-procesamiento del modelo origen: Se lleva a cabo un proceso de limpieza del XML generado anteriormente con el objetivo de facilitar su transformación y hacerlo conforme con el meta-modelo multidimensional. Lo anterior determina la realización de un grupo de operaciones de limpieza implementadas en un fichero con extensión *.xsl*.
- Construcción y montaje de meta-modelos origen y destino: Se define la plataforma para el modelo lógico en función de los requerimientos no funcionales de tecnología dispuestos por los clientes del proyecto. A partir de esta definición se crea o reutiliza el meta-modelo correspondiente a dicha plataforma (meta-modelo destino). La construcción se lleva a cabo a través del EMF, que provee como una de sus funcionalidades editores como el: *Sample Ecore Modelo Editor*. El mismo proceso se sigue con el meta-modelo multidimensional origen.
- Definición e implementación de reglas de transformación: Se definen las reglas de transformación basadas en el establecimiento de correspondencias entre las clases de ambos meta-modelos (origen y destino). Una vez definidas las reglas se implementan con el lenguaje ATL en los mecanismos que ofrece el EMF para este fin.

Seguidamente se definen un conjunto de consideraciones para llevar a cabo la primera actividad del flujo de actividades anterior.

2.2 Consideraciones de diseño.

Para la construcción del modelo conceptual del almacén de datos para controlar tiempo y costos de proyectos se plantean un conjunto de consideraciones asociadas con las estructuras que formarán parte del modelo. En cada sección se explican además buenas prácticas relacionadas con las estructuras que se diseñan.

2.2.1 Consideraciones para el diseño de metadatos de dimensiones comunes.

En todo proceso de diseño de un almacén de datos constituye parte importante la definición de dimensiones que perfilarán los futuros hechos a diseñar. Existen dimensiones, que en su generalidad son comunes a la mayoría de los hechos definidos. Su propósito es perfilar los análisis requeridos y pueden ser presentadas en los reportes como filas, columnas o filtros.

En el área de la gestión de proyectos, partiendo de las funcionalidades básicas de las herramientas de gestión más usadas y de los indicadores básicos para la gestión de proyectos, se pueden identificar las siguientes dimensiones como necesarias y comunes en un almacén de datos destinado a controlar la gestión de proyectos.

- Tiempo: Permite mantener un historial de la ejecución del proyecto.
- Estado: Estado de ejecución en que se encuentra una estructura del proyecto en un momento determinado.
- Prioridad: Prioridad asignada a una estructura del proyecto en función de su importancia y/o complejidad.

Existen otros perfiles de análisis, derivados de clasificaciones aplicables al proyecto en su proceso de gestión, pero asociadas a áreas del conocimiento específicas de la gestión de proyecto. Más adelante, al tratar específicamente cada área del conocimiento en particular a tener en cuenta, se definirán las dimensiones que apliquen en cada caso.

Por su importancia en la mayoría de los diseños de almacenes de datos, se explicarán las consideraciones para la definición de los metadatos de la dimensión tiempo.

2.2.1.1 Dimensión tiempo.

Para el modelado de la dimensión tiempo en un almacén de datos deben tenerse en cuenta dos factores:

- Nivel de detalle temporal con que se suceden y registran los eventos en la fuente de datos.
- Nivel de detalle temporal de interés de los clientes para la presentación de información.

El primer elemento determina el máximo nivel de detalle que tendrá la dimensión tiempo, mientras que el segundo define los niveles superiores que formarán parte de la o las jerarquías presentes en la dimensión tiempo.

De un día para otro no se considera significativo el avance en tiempo y costos en un proyecto en el proceso de control. Es por ello que suelen establecerse intervalos que comprenden un período en que el proyecto es evaluado a partir de su avance en las áreas priorizadas en ese

espacio de tiempo. Dichos intervalos pueden ser semanales, quincenales, mensuales, trimestrales, semestrales o incluso anuales.

Los metadatos representativos de la dimensión tiempo, a partir de una relación recursiva, permitirán la desagregación de niveles específica definida para cada proyecto en particular.

2.2.2 Consideraciones para el diseño de metadatos de dimensiones estructurales.

Toda organización está estructurada jerárquicamente en áreas o espacios de trabajos que cumplen con determinadas funciones y tributan a determinados objetivos. Dichas áreas de trabajo son estructuras cuyos resultados tributan a niveles superiores. El conjunto de estructuras que componen una organización desde el punto de vista organizativo puede resultar en una dimensión con jerarquías definidas asociadas a la estructura organizativa en cuestión.

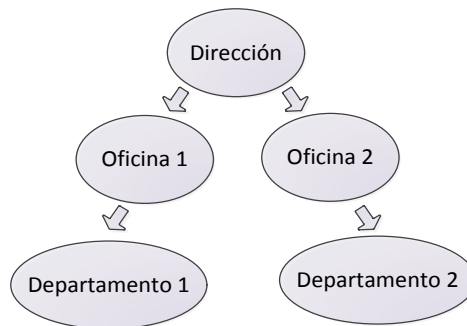


Figura 4. Organigrama ejemplo

Por otra parte, en el ámbito de la gestión de proyectos existe el concepto Estructura de Desglose del Trabajo (EDT), que no es más que una desagregación del proyecto en componentes más específicos y medibles, que son controlados y monitoreados a lo largo del ciclo de vida del proyecto. Dichos componentes pueden ser categorizados según el nivel donde se encuentren dentro de la EDT y cada categoría constituye precisamente la base de los metadatos a tener en cuenta para el modelado de las dimensiones estructurales del almacén de datos. La siguiente figura muestra un ejemplo de EDT.

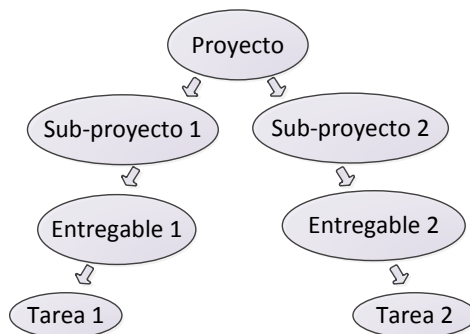


Figura 5. EDT ejemplo

Se debe tener en cuenta en el desglose que el resultado de cada nivel tributa al buen funcionamiento de las entidades de niveles superiores.

De ambas estructuras, el organigrama y el EDT, puede resultar una o varias dimensiones, todo depende del estilo de modelado que se use: estrella o copo de nieve, en función de los tipos de análisis asociados a cada nivel presente en la jerarquía o jerarquías definidas.

En el presente trabajo se propone que el organigrama constituya una dimensión independiente de la EDT, relacionada con la dimensión de la EDT según el estilo copo de nieve o de forma normalizada por las siguientes razones:

- Las estructuras que componen el organigrama de la organización pueden ejecutar otras estructuras que no sean necesariamente proyectos. Dichas estructuras pueden requerir de un control además, como sucede con los proyectos y todos sus hijos correspondientes.
- En ocasiones también los altos directivos de las organizaciones que ejecutan y gestionan proyectos, requieren la presentación de resúmenes de los propios proyectos al nivel de la estructura que los ejecuta, sin entrar necesariamente al detalle de cada proyecto en particular.

Lo anterior determina el modelado independiente de ambas dimensiones, mezclando de esta forma los estilos estrella y copo de nieve.

2.2.3 Análisis enfocados a diferentes niveles de una misma jerarquía.

En las dimensiones asociadas con la EDT y el organigrama, en muchas ocasiones se requieren análisis relacionados con un nivel superior al más detallado. Este tipo de situación suele presentarse además con la dimensión común de tiempo.

Con el objetivo de representar la variedad de medidas aplicables a las diferentes estructuras que componen una jerarquía, una buena práctica es repetir las instancias de los niveles superiores en los niveles inferiores, o sea, cada instancia de un nivel, se contendrá a ella misma dentro de sus hijos en el nivel inferior.

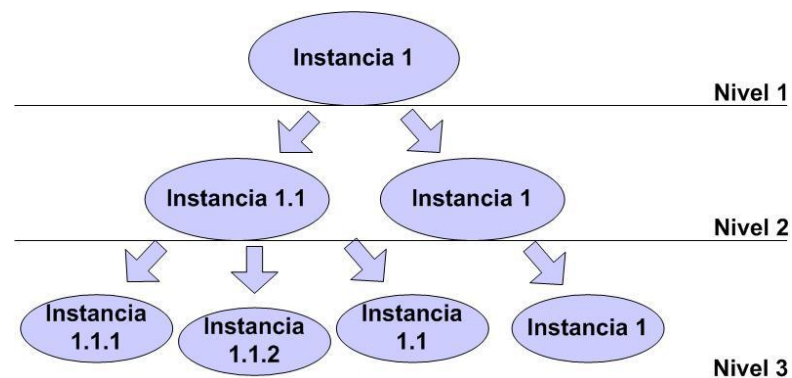


Figura 6. Representación de instancias por niveles en una jerarquía

Como se muestra en la imagen anterior, en el nivel 3 de mayor detalle, se ubicarán, tanto las instancias hijas que según la jerarquía, se sitúan en ese nivel, además del resto de las instancias de niveles superiores. Lo anterior permite la realización de diversos análisis con las diferentes instancias presentes en el mayor nivel de detalle, pues es con este nivel con el que se relaciona directamente el hecho que se cree.

De esta manera, se crea un solo hecho representativo del área de la gestión que aplica para los análisis de las diferentes estructuras, que contiene todas las medidas correspondientes para cada estructura. Como es evidente, en cada fila del hecho creado, habrá medidas que no aplican para la instancia referenciada, dichas medidas tendrán el valor nulo.

El valor nulo desde el punto de vista de presentación de datos no supone un problema pues simplemente se configura la herramienta de visualización para que no sea mostrado.

2.2.4 Manejo de jerarquía con diferentes niveles de profundidad.

Se puede dar el caso de que en una misma jerarquía, por un camino se llegue hasta el máximo nivel de detalle, mientras que por otro camino no. Tal pudiera ser el caso de una dimensión que modela proyectos que contienen sub-proyectos que a su vez contienen entregables. Para reflejar el caso anterior en el ejemplo citado, puede presentarse un sub-proyecto que no contenga entregables.

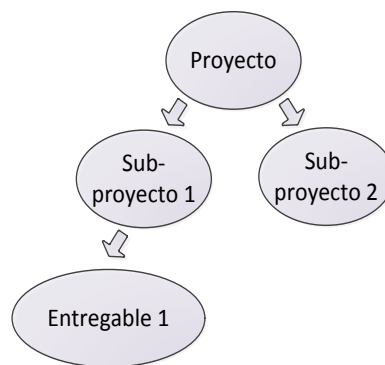


Figura 7. Ejemplo EDT sin entregable

En este caso la instancia del sub-proyecto sin entregables se repite en el nivel hijo correspondiente al entregable. De esta manera, todos los caminos de la jerarquía tienen el mismo nivel de profundidad.

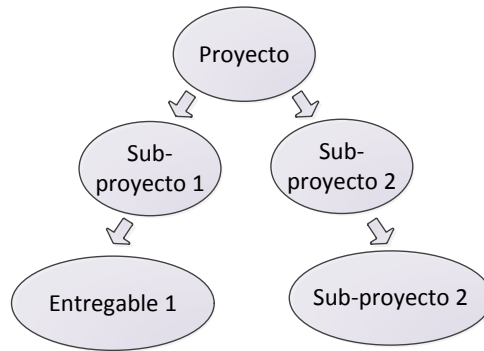


Figura 8. Ejemplo EDT repitiendo sub-proyecto como entregable

Los atributos correspondientes al nivel entregable que no apliquen para el sub-proyecto pueden dejarse con el valor nulo, simplemente basta con bajar al nivel inferior atributos asociados al identificador de negocio y al nombre.

2.2.5 Distinción de instancias en cada nivel de una jerarquía.

Al aplicar las soluciones anteriores es de vital importancia el uso de atributos bandera para identificar el tipo de instancia en cada nivel. Lo anterior se debe a que, siguiendo el ejemplo de la Figura 5, en el nivel de tareas también se tendrían los entregables, los sub-proyectos así como los propios proyectos.

Usando la misma figura como referencia, se tienen cuatro niveles en la jerarquía, en el de menor nivel de detalle no hay necesidad de añadir ningún atributo que identifique el tipo de instancia, pues solo se almacenarán instancias de tipo proyecto, no obstante se puede contemplar con el objetivo de lograr uniformidad en el diseño de los metadatos de los atributos por nivel. En el segundo nivel, se tendrán tanto sub-proyectos como proyectos, por tanto se requiere de un atributo que determine la naturaleza de la instancia. En el tercero se almacenarán entregables, sub-proyectos y el proyecto mientras que en el cuarto o nivel de mayor detalle se almacenarán todos los tipos de estructuras contempladas en la Figura 5 (Ver Tabla 2).

Tabla 2. Atributos banderas por nivel

Nivel	Atributo bandera	Tipo de dato	Valores
Proyecto	tipo_instancia	<i>varchar</i>	Proyecto
Sub-proyecto	tipo_instancia	<i>varchar</i>	Proyecto
			Sub-proyecto
Entregable	tipo_instancia	<i>varchar</i>	Proyecto
			Sub-proyecto
			Entregable
Tarea	tipo_instancia	<i>varchar</i>	Proyecto
			Sub-proyecto

			Entregable
			Tarea

De esta forma, en cada nivel de la jerarquía, es posible identificar qué tipo de estructura se almacena. Esto puede ser funcional a la hora de presentar informes que requieran de medidas correspondientes solo a los entregables y de esta forma es posible filtrar los resultados solo para los tipos de estructuras señalados.

2.2.6 Consideraciones para el diseño de metadatos de hechos y dimensiones específicos por área.

La gestión de proyectos abarca un conjunto de áreas del conocimiento, que integradas contribuyen a que el proyecto obtenga todos los resultados esperados. Sin embargo, existen áreas claves (tiempo y costos) que determinan el buen funcionamiento del proyecto y que al ser analizadas con indicadores medibles pueden conducir a la base de los problemas presentes en la ejecución del proyecto.

A continuación se presentan consideraciones por cada una de estas áreas a modelar así como se proponen un conjunto de indicadores de actividad que representan estadísticas para evaluar el estado del proyecto.

2.2.7 Medición de los costos y el financiamiento.

La Gestión de los Costos del Proyecto incluye los procesos involucrados en la planificación, estimación, preparación del presupuesto y control de costos de forma que el proyecto se pueda completar dentro del presupuesto aprobado (PMI, 2009).

La definición anterior determina la existencia del siguiente conjunto de indicadores a medir en el área de conocimiento de gestión de costos:

- Varianza de los costos.
- Índice de rendimiento de los costos.
- Porcentaje planificado de ejecución del presupuesto.
- Porcentaje de ejecución real.
- Porcentaje real de ejecución del presupuesto.
- Flujo de caja real.
- Flujo de caja planificado.
- Ejecución financiera.
- Ejecución presupuestaria.
- Porcentaje de financiamiento.

Los indicadores anteriores se sustentan en las siguientes métricas:

- Costo planificado.

- Costo real.
- Financiamiento planificado.
- Financiamiento real.
- Trabajo planificado.
- Trabajo real.

La combinación de las métricas asociadas con costos y financiamiento puede resultar además en la presentación de curvas de control que muestren el perfil de comportamiento acumulado de la comparativa entre los costos presupuestados y los costos reales, que puede combinarse además con la perspectiva acumulada de financiamiento real y planificado (Delgado Vítore, 2006). Las curvas de control permiten la visualización de la variación de los costos y el financiamiento ofreciendo una visión gráfica de la evolución de dichas métricas.

Los costos se gestionan y controlan por conceptos de lo gastado. Dichos conceptos se conocen como partidas presupuestarias, las que pueden ser específicas de una institución o genéricas, pero siempre deben constituir nomencladores flexibles a nuevas instancias.

Los costos se estiman por adelantado en las partidas presupuestarias requeridas y se desglosan por las actividades hasta el nivel donde radique la máxima unidad de detalle a la que se le asignan los costos planificados y por la que se controlan los costos reales. Esta máxima unidad puede estar al nivel de cualquiera de las estructuras que componen la EDT, la ubicación de la misma depende del nivel de detalle con que se manejen los elementos de costos en el negocio.

El objetivo del control de costos es colocar a disposición de la dirección del proyecto información actualizada, cierta y consistente de los desembolsos efectuados y proyectados teniendo presente el presupuesto asignado para su materialización. Para ello es de vital importancia la verificación de la exactitud de los valores que representan las bases de cálculo de los indicadores, en función de calcular indicadores que respondan a la situación real del proyecto. La siguiente figura muestra el proceso adecuado para llevar a cabo al cálculo de indicadores verificando con anterioridad los valores de costos reales.

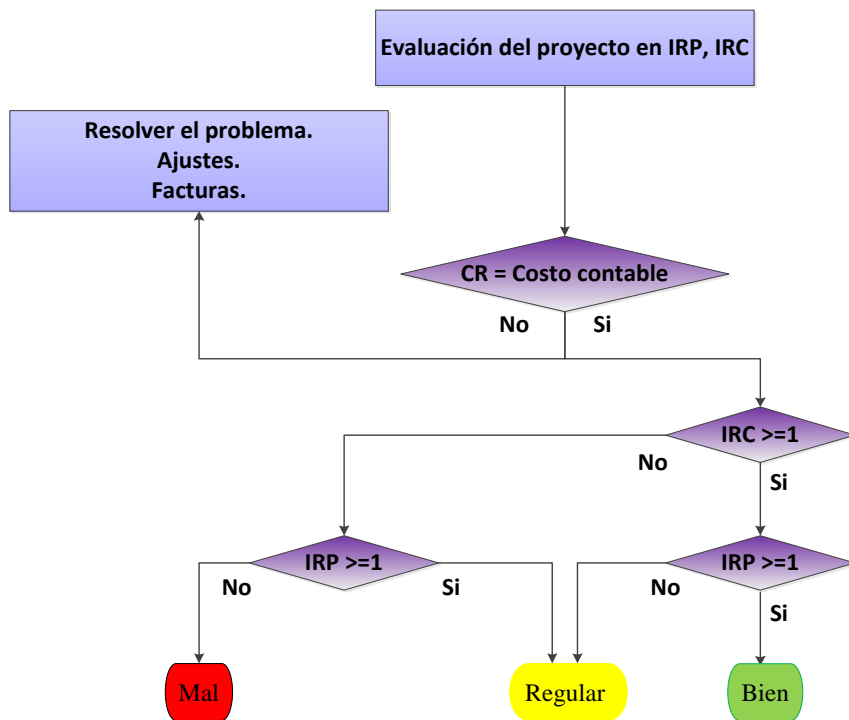


Figura 9. Comprobación de costos reales para cálculo de indicadores

En ocasiones, por razones externas o cálculos imprecisos, el costo real que se lleva internamente en el proyecto a través de las herramientas de gestión (CR), no coincide con los valores de costo real que emite el centro de contabilidad del proyecto (Costo contable). Esta diferencia en ambos costos reales puede afectar notablemente la precisión de los indicadores a presentar, resultando en la toma de decisiones no acertadas en el proyecto. Si los costos reales, tanto el interno del proyecto como el que emite contabilidad no coinciden, deben analizarse las razones de las desviaciones, corregirlas y posteriormente calcular los indicadores con el objetivo de presentar información exacta que permita conocer el estado real del proyecto.

Por su parte el financiamiento, tanto real como planificado, proviene de una fuente que provee los recursos monetarios para la ejecución del proyecto. Dichas fuentes deben estar pre-determinadas en el negocio con el objetivo de perfilar los indicadores de financiamiento por dicho concepto. En caso de que las fuentes de financiamiento no estén pre-definidas en el negocio, no se recomienda el diseño de una dimensión para las mismas, no obstante los valores asociados al financiamiento se siguen controlando como medidas del hecho correspondiente a financiamiento, sin especificar la fuente que lo provee.

Tanto los costos como el financiamiento requieren de un análisis que represente la ejecución real y la planificación, con el objetivo de comparar y evaluar el comportamiento en diferentes momentos del ciclo de vida del proyecto.

Para diseñar las estructuras que representan ejecución real y planificación en un almacén de datos para monitorear proyectos se deben tener en cuenta una serie de

aspectos que serán analizados en ambas perspectivas. Tomando como referencia la variable de costos se tienen los siguientes criterios:

- La frecuencia con la que se actualizan los valores en la herramienta que gestiona la ejecución real y planificada del proyecto no es la misma en la perspectiva real y en la planificada. El costo planificado de un proyecto forma parte de la línea base de dicho proyecto. Lo anterior implica que se define por adelantado y tiene en cuenta todo el tiempo de vida de un proyecto. Al formar parte de la línea base, está sujeto a cambios, que pueden llevarse a cabo en cualquier momento dadas las circunstancias en las que se encuentre el proyecto. Por tanto, el costo planificado puede cambiar en la herramienta de gestión con una frecuencia no estable. Esto implica que las estructuras donde se modelen cada una de estas medidas, deben manejarse de manera diferente en el proceso de ETL.
- El costo planificado es posible cargarlo por adelantado pues corresponde a una planificación a lo largo de todo el ciclo de vida de un proyecto, mientras que el costo real solo se obtiene en el momento actual, pues corresponde a la ejecución real del proyecto.
- Ambas perspectivas requieren de funciones de agregación que resuma los valores en los niveles superiores.

A pesar de que el criterio del último punto coincide para ambas perspectivas, el resto de los criterios determinan la modelación en hechos diferentes de las perspectivas real y planificada, tanto del costo como del financiamiento, con el objetivo de evitar problemas en la carga asociados con diferentes frecuencias de actualización en un mismo hecho y unión de líneas bases con conceptos de ejecución real en un mismo hecho.

2.2.8 Medición del tiempo.

La Gestión del Tiempo del Proyecto incluye los procesos necesarios para lograr la conclusión del proyecto a tiempo (PMI, 2009).

Los indicadores a tener en cuenta para controlar la variable tiempo son:

- Índice de rendimiento de la programación.
- Varianza de la programación.
- Productividad.
- Porcentaje de ejecución física.

Los indicadores anteriores se sustentan en las siguientes métricas:

- Trabajo planificado.
- Trabajo real.

- Costo planificado.
- Costo real.

De manera general los indicadores propuestos permiten el control de la gestión del cronograma del proyecto. Las bases de cálculo en todos los casos son las mismas, por tanto se modelan como variables derivadas con sus respectivas fórmulas asociadas.

2.2.9 Medición de estadísticas del proyecto.

Como valor agregado a los análisis de tiempo y costos, se proponen un conjunto de indicadores de actividad asociados a las estructuras que componen la EDT, que contribuyen a evaluar el estado del proyecto según sus estadísticas.

La razón de uso de indicadores de actividad se sustenta en que los reportes para el alto mando no siempre se presentan con un nivel de detalle elevado, pues no es de importancia para un directivo conocer el estado específico de un proyecto sino la unión de todos aquellos elementos que tributan al funcionamiento de la institución de manera general. A un nivel superior son más interesantes resúmenes de información como la cantidad de entregables por sub-proyecto, con su estado, prioridad, y otras dimensiones que perfilen dicha cantidad. Tal y como se requiere un resumen de cantidad de entregables por sub-proyecto, pudiera requerirse un resumen de cantidad de sub-proyectos por proyecto u otros análisis relacionados con cantidades de estructuras del propio negocio.

Dado que en la dimensión de la EDT, todas las instancias, de cualquier tipo que estas sean, se encuentran en el nivel de máximo detalle, esto puede resolverse categorizando las estructuras a cuantificar. En este caso para distinguir las estructuras que se cuente es menester establecer un listado de nomencladores que se refiere precisamente a las categorías definidas por nivel en la EDT. En este caso los nomencladores aplicables pueden ser 4: tarea, entregable, sub-proyecto y proyecto.

Los flujos de ETL deben ser capaces de identificar qué referencia insertar en el hecho desde la dimensión de la EDT y desde la dimensión de nomencladores citada.

Se debe resaltar que dichas referencias en cada fila del hecho son de vital importancia, pues al no relacionarlas debidamente, se obtiene un resultado erróneo.

En el caso puntual que se maneja, las relaciones de las referencias de las dos dimensiones mencionadas, quedaría de la siguiente manera:

Tabla 3. Relación entre dimensiones EDT y categorías de EDT para medir estadísticas

Dimensión de la EDT	Dimensión de categorías de la EDT
Proyecto X	Sub-proyecto

Sub-proyecto Y	Entregable
Entregable Z	Tarea
Tarea P	Estructuras de menor nivel (No conformidades, solicitudes de cambio, etc.)

En este caso solo se crea un hecho para el conteo de todas las estructuras requeridas, y solo basta con añadir un nomenclador más en la dimensión de categorías de la EDT en caso de solicitar la inclusión de otra estructura a cuantificar.

Se debe tener en cuenta, que todas las cantidades que se midan, deben ser perfiladas por las mismas dimensiones de análisis. Por otra parte, navegando en un mismo reporte analítico, se puede obtener diferentes vistas de lo anterior, realizando los filtros adecuados tal y como se presentan en la Tabla 3.

Todas las actividades a medir deben tener una prioridad, cambian en el tiempo, pasan por diferentes estados y tienen un por ciento de avance asociado así como los indicadores de cantidades se deben analizar desde dos escenarios: acumulado y en un período.

A continuación se presenta una propuesta de perfiles de análisis asociados a los indicadores de actividad que pudieran ser útiles a la hora de medir las estadísticas en un proyecto:

- **Atrasadas:** Es acumulada y representa la cantidad de actividades atrasadas.
- **Adelantadas:** Es acumulada y representa la cantidad de actividades adelantadas.
- **Culminadas:** Es acumulada y representa la cantidad de actividades terminadas del total.
- **Culminadas_adelantadas:** Es acumulada y representa la cantidad de actividades que debían estar a un por ciento determinado y ya están al 100%.
- **Culminadas_atrasadas:** Es acumulada y representa la cantidad de actividades que llegan al 100% en una fecha posterior a la fecha en que debían terminar.

2.3 Construcción del modelo conceptual.

Esta sección presenta el resultado correspondiente a la primera actividad del flujo de actividades presentado en la Figura 3. El diseño contempla los indicadores definidos para tiempo y costos así como los indicadores de actividad para medir las estadísticas

del proyecto. Se construyó siguiendo las consideraciones detalladas en secciones anteriores del presente capítulo y teniendo en cuenta los tres niveles de profundidad definidos en (Luján Mora, 2005).

A continuación se presenta el conjunto de paquetes estrellas del modelo conceptual resultado:

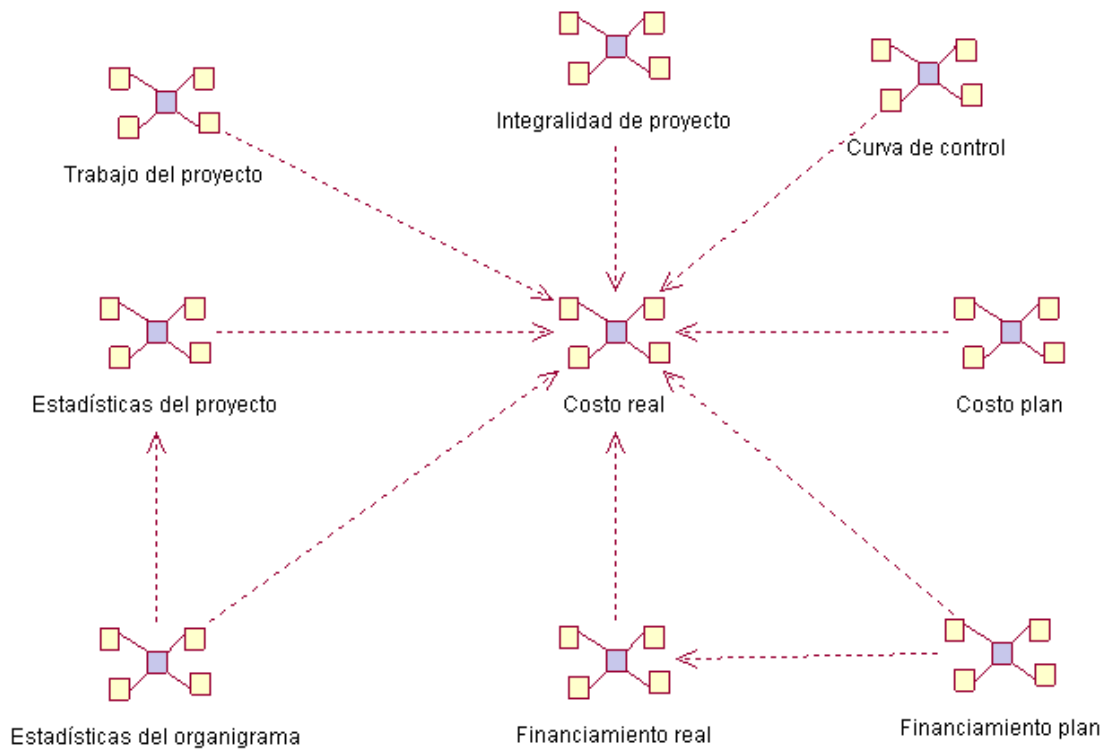


Figura 10. Paquetes estrellas del modelo conceptual

La figura anterior representa los paquetes estrellas relacionados entre sí, que forman parte de la propuesta conceptual definida. En conjunto abarcan todos los aspectos concernientes a las áreas de tiempo y costos de la gestión de proyectos, incluyendo además paquetes estrellas asociados a las estadísticas de proyectos y del organigrama. Los diagramas asociados a cada paquete en particular se encuentran en el Anexo 1 del presente trabajo. A continuación se explica de manera general cada paquete estrella:

- Trabajo del proyecto: Contiene el avance, tanto planificado como real así como acotado a un período y acumulado. Contiene además indicadores derivados de las medidas anteriores.
- Estadísticas del proyecto: Contiene las estadísticas medidas en cantidades acumuladas y en el período, de estructuras de proyecto clasificándolas por diferentes perspectivas de análisis.

- Estadísticas del organigrama: Contiene las estadísticas medidas en cantidades de proyectos por estructuras del organigrama de la institución y clasificadas por diferentes perspectivas de análisis.
- Costo real: Contiene el costo real, tanto acumulado como acotado al período y clasificado por diferentes perspectivas de análisis.
- Financiamiento real: Contiene el financiamiento real, tanto acumulado como acotado al período y clasificado por diferentes perspectivas de análisis.
- Costo plan: Contiene el costo planificado, tanto acumulado como acotado al período y clasificado por diferentes perspectivas de análisis.
- Financiamiento planificado: Contiene el financiamiento planificado, tanto acumulado como acotado al período y clasificado por diferentes perspectivas de análisis.
- Integralidad del proyecto: Representa a una vista materializada y contiene indicadores derivados de los paquetes de Costo real, Costo plan y Trabajo del proyecto.
- Curva de control: Representa a una vista materializada y contiene indicadores derivados de los paquetes de Costo real, Costo plan, Financiamiento real y Financiamiento plan.

Cada uno de los paquetes anteriores se encuentra perfilado por un conjunto de dimensiones, en varios casos, comunes a la mayoría de los paquetes. A continuación se explica brevemente la razón de ser de cada una de las dimensiones que compone la propuesta conceptual definida:

- Tiempo: Representa una estructura jerárquica de niveles de tiempo. Es una estructura genérica pues en lugar de especificar cada uno de los niveles existentes en la dimensión, contiene una relación recursiva que facilita la inclusión de tantos niveles como amerite el caso que se trate.
- Estado: Permite el almacenamiento de todos los estados por los que pueden pasar tanto las estructuras de los proyectos como los propios proyectos.
- Prioridad: Permite el almacenamiento de todas las prioridades que pueden ser asociadas a los proyectos y estructuras hijas.
- EDT: Representa una estructura jerárquica que permite el almacenamiento de las estructuras de desglose del trabajo del proyecto. Es una estructura genérica pues en lugar de especificar cada uno de los niveles existentes en la dimensión, contiene una relación recursiva que facilita la inclusión de tantos niveles como amerite el caso que se trate.
- Organigrama: Representa una estructura jerárquica que permite el almacenamiento de las estructuras departamentales o áreas que contiene la

institución. Es una estructura genérica pues en lugar de especificar cada uno de los niveles existentes en la dimensión, contiene una relación recursiva que facilita la inclusión de tantos niveles como amerite el caso que se trate.

- Partida presupuestaria: Permite el almacenamiento de las partidas presupuestarias que perfilan tanto el costo planificado como el real.
- Fuente de financiamiento: Permite el almacenamiento de las fuentes de financiamiento que perfilan tanto el financiamiento planificado como el real.
- Indicador actividad: Permite el almacenamiento de indicadores que perfilan las estadísticas medidas en cantidades asociadas a las estructuras de proyectos.
- Categoría actividad: Permite el almacenamiento de las categorías asociadas a cada nivel de la EDT.
- Intervalo por ciento: Permite el almacenamiento de un listado de por cientos por los que pasan las estructuras de los proyectos así como los propios proyectos.
- Clasificación proyecto: Representa una estructura jerárquica de dos niveles. El primer nivel o de menor detalle representa los tipos de clasificaciones que puede tener el proyecto. El segundo nivel o de mayor detalle permite el almacenamiento de las clasificaciones puntuales en función de los elementos padres correspondientes del nivel superior.

Para representar la relación existente, en cuanto a dimensiones comunes se refiere, y facilitar la integración de los diferentes paquetes estrellas, se usa la herramienta de la Matriz bus, ubicando los paquetes estrellas en las columnas y las dimensiones en las filas (Ver Tabla 4).

Tabla 4. Matriz Bus

Dimensiones/ Paquetes estrellas	Trabajo del proyecto	Estadísticas del proyecto	Estadísticas del organigrama	Costo real	Financiamiento real	Costo plan	Financiamiento plan	Integralidad del proyecto	Curva de control
Tiempo	X	X	X	X	X	X	X	X	X
Estado		X	X						
Prioridad		X	X						
EDT	X	X		X	X	X	X	X	X
Organigrama			X						
Partida presupuestaria				X		X			
Fuente de financiamiento					X		X		
Indicador actividad		X							
Categoría actividad		X							
Intervalo por ciento		X	X						
Clasificación proyecto			X						

Como se evidencia en la tabla anterior, las dimensiones Tiempo y EDT son comunes a la mayoría de los paquetes estrellas. Las dimensiones de estado y prioridad son referenciadas desde los paquetes que miden estadísticas, tanto del proyecto como del organigrama. La dimensión Organigrama y Clasificación de proyecto son propias del paquete de Estadísticas del organigrama. La Partida presupuestaria se asocia a los paquetes Costo real y Costo plan así como la Fuente de financiamiento se asocia a Financiamiento real y Financiamiento plan. Por su parte, el Indicador actividad y Categoría actividad son dimensiones propias del paquete Estadísticas proyecto y la dimensión Intervalo por ciento es compartida por el paquete anteriormente citado y el paquete Estadísticas organigrama.

Una vez obtenida la propuesta conceptual de un almacén que permita el control de la gestión de proyectos en las áreas tiempo y costos, es menester llevar a cabo las demás actividades que componen el flujo de la Figura 3 en función de obtener un conjunto de metadatos asociados al nivel de abstracción lógico. En el capítulo 3 se ejecutan el resto de estas actividades.

Conclusiones parciales

Siguiendo el enfoque MDA se especificaron un conjunto de actividades con sus herramientas y elementos de salida que guiarán todo el proceso de construcción de la propuesta del presente trabajo.

A partir de las experiencias de un grupo de proyectos similares se identificaron y explicaron un conjunto de buenas prácticas para el modelado de las aristas de tiempo y costos. Con los indicadores asociados a estas áreas y con indicadores de actividad propuestos para analizar estadísticas del proyecto, se construyó el modelo conceptual que constituye el resultado de la primera actividad presentada en el flujo de actividades basado en MDA. El modelo obtenido constituye el punto de partida para realizar el resto de las actividades basadas en los principios de MDA. Constituye además una propuesta que puede ser usada en marcos de trabajo que requieran un control de proyectos.

El modelo resultado del presente capítulo, al ser desarrollado a un nivel de abstracción conceptual, puede ser transformado a tantos modelos lógicos como se desee o requiera el negocio donde se aplique, siguiendo la línea de MDA.

Capítulo 3. Construcción del modelo lógico

Introducción

En el capítulo anterior se obtuvo un PIM multidimensional, a partir del cual en el presente capítulo se pretende obtener un PSM lógico dependiente de la plataforma relacional. Para ello en el presente capítulo se llevan a cabo el resto de las actividades de la Figura 3.

3.1 Pre-procesamiento del modelo multidimensional de origen.

El modelo origen es el presentado en el capítulo 2 del presente trabajo. Dicho modelo se construyó basado en las extensiones de UML definidas para modelado multidimensional detalladas en (Luján Mora, 2005) y está ubicado en la capa M1 de las definidas por MOF.

Una vez culminado el diseño conceptual anterior en la herramienta *Rational Rose* con la extensión de UML para modelado multidimensional, se genera un fichero con extensión .xml.

El formato de los ficheros generados en el *Rational Rose* no facilita su transformación de forma directa y es necesario realizar un grupo de operaciones previas que fueron agrupadas en un proceso denominado de limpieza. Los elementos que motivaron esa decisión son presentados a continuación:

- La generación adicional de un conjunto de etiquetas propias de la herramienta que no reportan contenido de interés para las transformaciones.
- Se detectó además que el fichero por cada clase de hechos de las modeladas repite la especificación de todas las estructuras modeladas. Lo anterior se traduce a que en caso de haber modelado 10 clases de hechos, el fichero contiene 10 veces repetida la misma información que debería estar solo una vez.
- Se requiere de una especificación de los tipos de datos como clases de primer nivel en la estructura XML (Clase *Type*). Los tipos de datos de los atributos tanto del hecho como de las dimensiones, tienen un atributo *type* cuyo valor es un objeto de tipo cadena que puede ser *Integer*, *String* o *Double*. La meta es evitar la especificación del objeto por cada atributo de tipo de dato y desde cada uno de ellos referenciar a la clase *Type* correspondiente que se debe crear por cada tipo de dato.
- El fichero a montar en el eclipse como modelo de entrada para las transformaciones debe estar en formato XML para que el *plug-in* ADT lo reconozca como elemento de entrada. Esto implica la modificación de la

cabecera del documento y cambiar la extensión del fichero de XML a XMI.

Este último paso debe hacerse de manera manual.

Para la corrección de los elementos anteriores se hace uso de transformaciones *Extensible Style Language* (XSL) donde se obtiene un fichero con extensión .xsl que contiene las transformaciones de limpieza ejecutadas (Ver Anexo 2).

Una vez culminado el pre-procesamiento se obtiene una especificación en formato XMI que constituye el modelo de entrada para el *plug-in* de eclipse encargado de las transformaciones.

La Figura 11 muestra un sub-conjunto de este modelo ya montado en la herramienta eclipse en formato XMI.

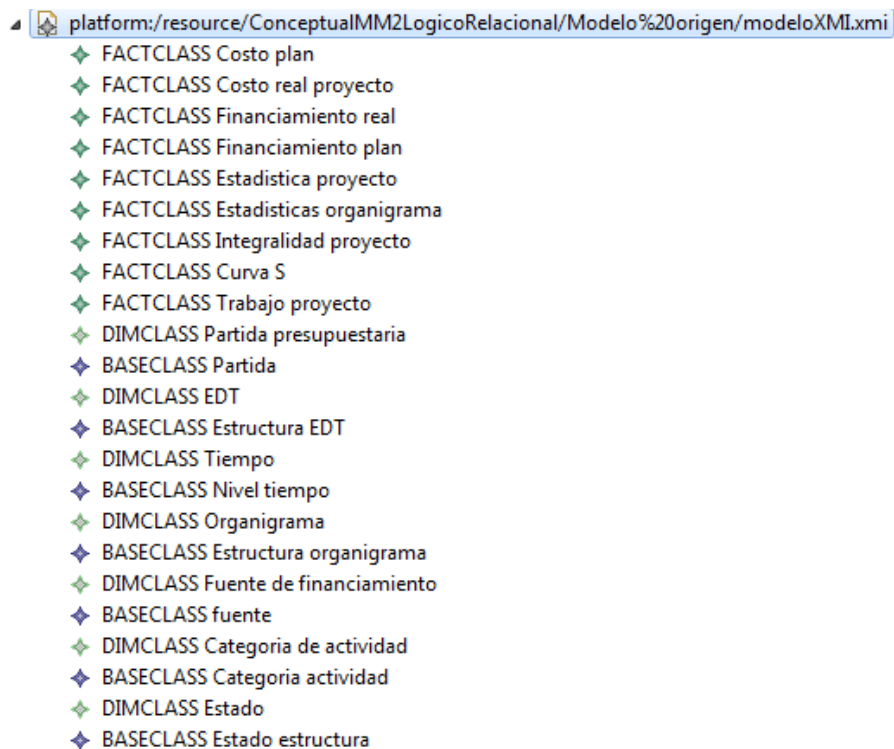


Figura 11. Modelo conceptual multidimensional de entrada

El modelo origen debe estar acorde o debe conformar a una especificación que define cuáles son los estereotipos correspondientes para el modelado multidimensional. Esta conformación se especifica a través de la siguiente línea código ubicada en la cabecera del fichero XMI del modelo:

```
xmlns="platform:/resource/ConceptualMM2LogicoRelacional/MMD.ecore".
```

El segmento platform:/resource determina que el recurso solicitado se encuentra en el espacio de trabajo (*workspace*) de la solución creada para implementar las transformaciones. Seguidamente se especifica el nombre de dicha solución y por último el nombre del meta-modelo al que conforma.

A continuación se construyen los meta-modelos requeridos para las transformaciones.

3.2 Construcción y montaje de meta-modelos origen y destino.

El meta-modelo de entrada en este caso representa la extensión de UML para modelado multidimensional que define los estereotipos a tener en cuenta para este tipo particular de modelado conceptual y está ubicado en la capa M2 de las definidas por MOF. Este meta-modelo se encuentra diagramado en (Luján Mora, 2005).

Seguidamente, el meta-modelo multidimensional origen es especificado en formato *ecore* (Eclipse Modeling Framework Project (EMF), 2012), estándar de codificación creado para describir meta-modelos en EMF. Dicha descripción se lleva a cabo a través del propio *framework* EMF, que provee como una de sus funcionalidades la creación de ficheros *ecore* en editores como el: *Sample Ecore Modelo Editor*. Una vez descrito el meta-modelo en formato *ecore* (Ver Figura 12) ya está listo para ser usado como entrada (*IN*) a las transformaciones que convierten elementos de su contenido en elementos del contenido del meta-modelo destino (*OUT*).

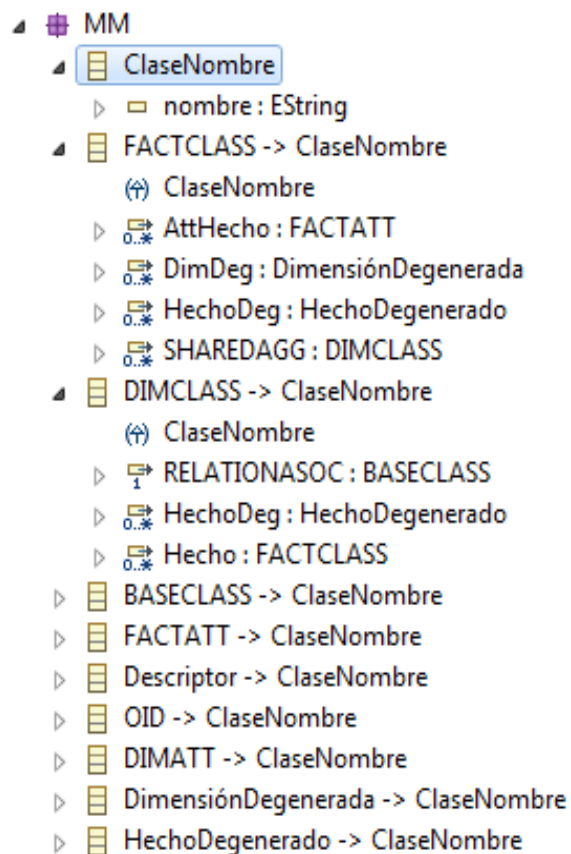


Figura 12. Meta-modelo multidimensional origen montado en eclipse

Se hizo necesario añadirle algunos elementos (que no fueron eliminados en el proceso de limpieza, pues eran de utilidad para las transformaciones) que no se contemplan en estos diagramas debido a que el modelo origen fue generado como fichero XML en la herramienta *Rational Rose* y en este proceso, además de las clases, atributos, relaciones y referencias requeridas por el meta-modelo multidimensional origen, se crearon en el modelo atributos adicionales como son:

identificadores de cada clase (independientes del nombre), atributos de cardinalidad en las relaciones, reglas de derivación para atributos de hechos derivados, condición de derivación y atributos que determinan anidación.

Una vez hechas las modificaciones anteriores se corroboró la conformidad del modelo multidimensional origen contra el meta-modelo multidimensional creado y se procedió a construir y montar el meta-modelo relacional destino.

El meta-modelo relacional del CWM es un estándar para representar la estructura de una base de datos relacional, permitiendo especificar tablas, columnas, claves primarias, claves foráneas, etc. Se define por un conjunto de clases derivadas del meta-modelo de UML que se encuentran diagramadas en (A Standard for Representing Multidimensional Properties: The Common Warehouse Metamodel (CWM), 2002).

El modelo relacional destino a obtener conformará exactamente el meta-modelo relacional.

Dicho modelo fue reutilizado de ejemplos ya implementados en el sitio del lenguaje ATL (ATL, 2012) y fue montado en el EMF en formato *ecore*. Quedó estructurado de la siguiente manera:

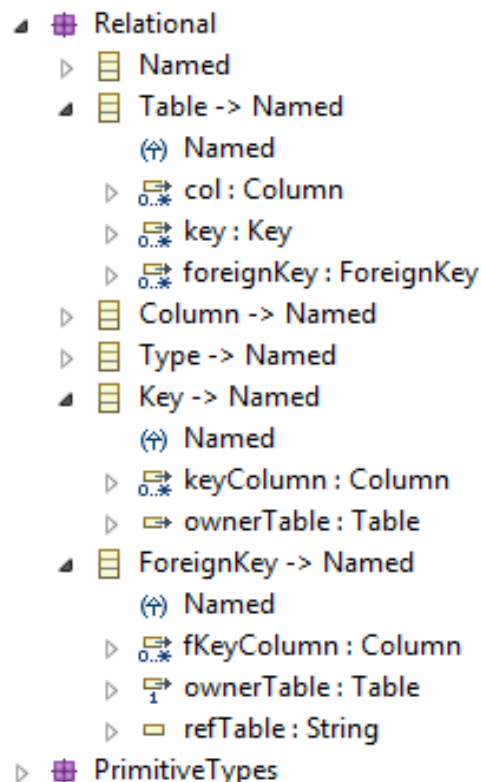


Figura 13. Meta-modelo relacional destino montado en eclipse

Las clases de tipo llave: Key y ForeignKey, contienen cada una, una referencia de la clase Column. De esta manera es posible la creación de llaves primarias compuestas (que es precisamente el tipo de llave que se suele crear en los hechos al unir todas o

un sub-conjunto de las referencias de las dimensiones) y de llaves foráneas que referencien llaves compuestas.

3.3 Definición e implementación de reglas de transformación.

A continuación se definen las reglas de transformación entre el meta-modelo multidimensional origen y el meta-modelo relacional destino:

1. Cada clase de tipo hecho se transformará en una tabla.
2. Los atributos de la clase de tipo hecho se transformarán en columnas de la tabla.
3. Las referencias de las dimensiones en una clase de tipo hecho conformarán la llave primaria de la tabla.
4. Los atributos que conforman la llave primaria serán además llaves foráneas referenciando a cada una de las tablas resultantes de las transformaciones de las dimensiones del hecho.
5. Cada clase de tipo dimensión se transformará en una tabla.
6. Los atributos, de cualquier tipo que estos sean, de todos los objetos de tipo clase base involucrados con la clase de tipo dimensión anterior, se convertirán en columnas de la tabla.
7. La tabla anterior tendrá además una llave primaria que la identificará.

En (Sánchez Gutiérrez, 2010) y en (TRANSFORMACIÓN DE ESQUEMAS MULTIDIMENSIONALES DIFUSOS DESDE EL NIVEL CONCEPTUAL AL NIVEL LÓGICO, 2010) se propone la conversión de cada clase de tipo base en una tabla diferente mientras que el presente trabajo propone construir una tabla a partir de una dimensión, cuyas columnas serían los atributos de las clases bases relacionadas con esta dimensión.

Se toma esta decisión debido a que el modelo conceptual multidimensional será transformado a un modelo lógico relacional, que por ende será almacenado en una base de datos relacional. Al almacenar los datos en una base de datos relacional, las estructuras serían tablas y relaciones, lo cual concuerda con los principios del modo de almacenamiento ROLAP. Esto implica que el proceso de recuperación de datos, cuando estos no se encuentra en la memoria caché de las herramientas de visualización, pase por las tres capas que componen la arquitectura ROLAP (Ver Figura 14).

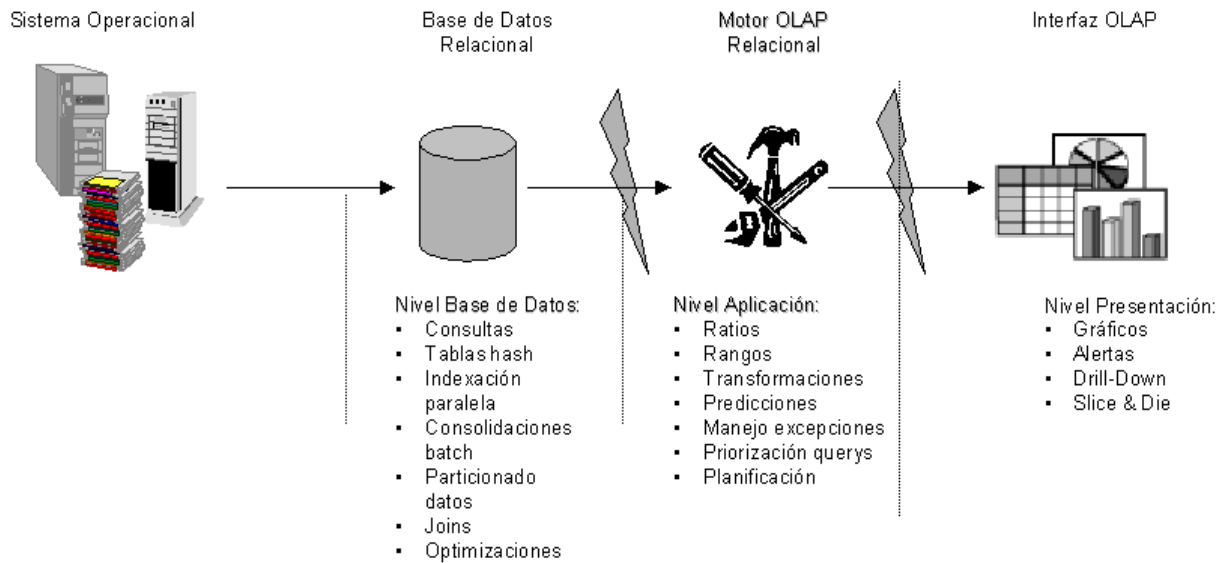


Figura 14. Arquitectura ROLAP

Si por cada clase de tipo base se creara una tabla, en la capa de la base de datos se requeriría el uso de *joins* para agregar las medidas de los hechos en los niveles (clase de tipo base) de la dimensión. Como es sabido, el uso de *joins* puede retardar el proceso de recuperación de datos, constituyendo esto una pérdida de rendimiento en la presentación de los reportes analíticos.

El transformar la dimensión y los atributos de todas las clases bases relacionadas, en una tabla con sus columnas respectivamente, es análogo a des-normalizar el conjunto de tablas en las que se convertirían todas las clases bases relacionadas con una misma dimensión. De esta forma se evita el uso de *joins* en la recuperación de los datos en la capa de la base de datos, compensando de cierta manera la pérdida en rendimiento por almacenar los datos en una base de datos relacional en lugar de una multidimensional. La variante propuesta permite una mejora considerable en comparación a la anterior y aporta una ganancia en flexibilidad para el almacenamiento.

3.3.1 Implementación de reglas de transformación para modelo lógico relacional.

Por cada transformación de las planteadas se implementan un conjunto de reglas y métodos de apoyo que tributan a la obtención del modelo lógico relacional.

Las reglas 5, 6 y 7 se implementan a partir de la regla central **DIMCLASS2Table**.

```

rule DIMCLASS2Table {
  from
    c : MMD!DIMCLASS
  to
    out : Relational!Table (
      name <- c.name,
      key <- colkey,
      col <- c.child_dimension(c.id)
      ->collect(s | thisModule.DIMATT2Column(s))
      ->union(Sequence{key})
    ),
    key: Relational!Column (
      name <- 'pk_' + c.name,
      type <- thisModule.objectIdType
    ),
    colkey: Relational!Key (
      keyColumn <- thisModule.DIMATT2Column(key)
    )
}

```

Figura 15. Regla para convertir dimensión en tabla

Primeramente se le asigna a la tabla destino el nombre de la clase dimensión correspondiente. A continuación se define la llave primaria de la tabla dimensión, identificada con la variable **colkey** de tipo clase **Key** del meta-modelo relacional, construyendo una columna (que constituirá la llave primaria) a partir de la regla **DIMATT2Column**, pasando como parámetro la variable **key**, de tipo clase **Columna** del meta-modelo relacional, cuyo nombre es la cadena **pk** concatenada con el nombre de la propia dimensión y el tipo de dato es el devuelto por el método **objectIdType**, en este caso *Integer* para las columnas que representan las llaves primarias de las tablas obtenidas a partir de clases dimensión. A continuación se presenta la regla **DIMATT2Column**.

```

lazy rule DIMATT2Column {
  from
    d : MMD!DIMATT
  to
    c : Relational!Column (
      name <- d.name,
      type <- d.type
    )
}

```

Figura 16. Regla para convertir atributos de dimensión en columnas

La regla anterior convierte atributos de dimensión a columnas de una tabla, asignando nombre y tipo de dato de atributos de una dimensión a columnas de una tabla.

Continuando con la regla principal de conversión de dimensión a tabla, las columnas de la tabla destino son los atributos, de cualquier tipo que estos sean, de todas las

clases bases involucradas con la clase dimensión fuente. A través del método **child_dimension** se obtienen este conjunto de atributos, los cuales posteriormente son transformados a columnas usando la regla **DIMATT2Column** y se le une además la variable **key** de tipo clase **Columna** del meta-modelo relacional, que constituye una columna también en el modelo destino.

A continuación se presentan los métodos usados para recopilar los atributos de las clases bases involucradas con una dimensión. El método **child_dimension** obtiene los atributos de dimensión de la clase base relacionada directamente con la clase dimensión y el método **child_base_dimension** implementa un algoritmo recursivo que obtiene los atributos del resto de las clases bases relacionadas.

```

helper context MMD!DIMCLASS def: child_dimension(id: String) :

Sequence(MMD!DIMATT) =
  MMD!BASECLASS.allInstances()
  -> select(i | i.id = MMD!DIMCLASS.allInstances())
  -> select(e | e.id = id).first().RELATIONASOC.child).first().DIMATT
  ->union(thisModule.child_base_dimension(id ,id, MMD!BASECLASS.allInstances())
  -> select(i | i.id = MMD!DIMCLASS.allInstances())
  -> select(e | e.id = id).first().RELATIONASOC.child).first());

```

Figura 17. Método para obtener atributos de la primera clase base de la dimensión

La llamada a **child_base_dimension** desde **child_dimension** pasa un conjunto de parámetros en el siguiente orden:

1. Identificador de la clase padre de la clase dimensión.
2. Identificador de la dimensión.
3. Clase base que se analiza.

Esta primera llamada pasa como identificador del padre el identificador de la propia dimensión coincidiendo en este caso con el segundo parámetro requerido y como tercer parámetro se tiene la clase base relacionada directamente con la dimensión.

```

helper def: child_base_dimension(id: String, id_dim: String, base: MMD!BASECLASS) :

Sequence(MMD!DIMATT) =
  base.RELATIONASOC
  -> select(z | z.child <> id_dim)
  -> select(z | z.child <> id)
  -> iterate(relacion; elements: Sequence(MMD!DIMATT) = Sequence{ } |
  elements->union(MMD!BASECLASS.allInstances())
  -> select(y | y.id = relacion.child)->first().DIMATT
  -> union(thisModule.child_base_dimension(base.id, id_dim, MMD!BASECLASS.allInstances())
  -> select(y | y.id = relacion.child)->first())));

```

Figura 18. Método para obtener atributos del resto de las clases bases de la dimensión

En la recursividad del método anterior, por cada clase base se obtienen todas sus relaciones, garantizando que estas no sean las relaciones con la dimensión ni con la

clase padre (cuyos atributos ya fueron recopilados). Posteriormente se itera por cada relación de las identificadas, determinando las clases bases correspondientes a cada relación y a su vez, llamando nuevamente al propio método **child_base_dimension**, pasando el identificador del padre, que en este caso es el identificador de la clase base pasada como el parámetro base en el paso anterior, el identificador de la dimensión y la propia clase base actual.

Por su parte, para las reglas 1, 2, 3 y 4 fue implementada la siguiente regla central o principal que evoca un conjunto de reglas y métodos de apoyo que transforman todas las estructuras relacionadas con el hecho como son: sus atributos y sus relaciones. Se define la siguiente como regla central porque una vez que se convierte un hecho a una tabla es que se le agregan a la misma las columnas, las llaves foráneas y primarias. Por tanto la raíz de este proceso es precisamente la conversión de hecho a tabla.

```
rule FACTCLASS2Table {
  from
    a : MMD!FACTCLASS
  to
    b : Relational!Table(
      name <- a.name,
      col<-a.FACTATT->collect(e | thisModule.FACTATT2Column(e))
      ->union(Sequence{a.SHAREDAGG->collect(f | thisModule.SHAREDAGG2Column(f))}),
      foreignKey<-a.SHAREDAGG->collect(f | thisModule.SHAREDAGG2ForeignKey(f)),
      key<- thisModule.SHAREDAGG2Key(a.SHAREDAGG)
    )
}
```

Figura 19. Regla para convertir un hecho en tabla

La regla anterior convierte elementos de clase de tipo hecho a tabla. Primeramente asigna el mismo nombre de la clase de tipo hecho a la tabla destino. Seguidamente procede a la creación de las columnas de la tabla que se generan a partir de los atributos de hechos contenidos por la clase de tipo hecho. En este segmento se evoca la regla **FACTATT2Column**, que por cada atributo de hecho obtiene una columna de la tabla. A continuación se presenta dicha regla.

```
lazy rule FACTATT2Column {
  from
    d : MMD!FACTATT
  to
    c : Relational!Column (
      name <- d.name,
      type <- d.type
    )
}
```

Figura 20. Regla para convertir atributos de hecho en columnas

En el caso anterior, se le asigna a la columna del modelo destino el nombre y el tipo de dato del atributo del hecho del modelo conceptual multidimensional.

Continuando con el flujo principal de la regla central, a dicho conjunto de columnas derivadas de los atributos del hecho, se le une el conjunto de referencias de clase dimensión asociadas a la clase hecho (referencias definidas por la etiqueta **SHAREDAGG**), evocando, por cada referencia, la regla **SHAREDAGG2Column** que al igual que **FACTATT2Column** convierte elementos (en este caso **SHAREDAGG**) a columnas de la tabla destino. A continuación se presenta la regla **SHAREDAGG2Column**.

```
lazy rule SHAREDAGG2Column {
  from
    d : MMD!SHAREDAGG
  to
    c : Relational!Column (
      name <- 'pk_' + d.name_dimension(d.dimclass),
      type <- thisModule.objectIdType
    )
}
```

Figura 21. Regla para convertir referencias de dimensiones en columnas

La regla anterior convierte cada referencia en columna, asignándole como nombre la cadena **pk** concatenada con el nombre de la dimensión obtenido a partir del método **name_dimension(id: String)** que dado un identificador de clase dimensión devuelve su nombre.

Luego de obtener el conjunto de columnas que conforman la tabla destino se procede a crear las llaves foráneas de la misma, que no son más que las referencias de las clases dimensión en el hecho. Dicha transformación se lleva a cabo mediante la regla **SHAREDAGG2ForeignKey** que a continuación se presenta.

```
lazy rule SHAREDAGG2ForeignKey {
  from
    d : MMD!SHAREDAGG
  to
    c : Relational!ForeignKey (
      name <- 'pk_' + d.name_dimension(d.dimclass),
      fKeyColumn <- thisModule.SHAREDAGG2Column(d),
      refTable<-d.name_dimension(d.dimclass)
    )
}
```

Figura 22. Regla para convertir referencias de dimensión en llaves foráneas

La regla anterior primeramente asigna el nombre de la llave foránea correspondiente (mismo principio de construcción que la columna a partir de la referencia). Posteriormente determina la columna representativa de esta llave foránea y finalmente define la clase dimensión a la que pertenece dicha referencia.

Finalmente, la regla central construye la llave primaria compuesta de la tabla destino, a partir de la llamada a la regla **SHAREDAGG2Key** pasándole el conjunto de referencias de clase dimensión presentes en la clase hecho.

```
lazy rule SHAREDAGG2Key {
  from
    d : MMD!SHAREDAGG
  to
    c : Relational!Key (
      keyColumn <- d->collect(f | thisModule.SHAREDAGG2Column(f))
    )
}
```

Figura 23. Regla para convertir referencias de dimensión en llave primaria compuesta

Por cada referencia de las pasadas a la regla anterior se construye una columna evocando a la regla ya presentada **SHAREDAGG2Column**.

La cabecera del fichero generado queda de la forma:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
        xmlns="Relational">
```

Como se evidencia, el espacio de nombres que referencia el meta-modelo relacional se genera solo como el nombre de dicho meta-modelo. Esto no permite abrir el fichero con el editor de modelos *ecore* del EMF, pues con el nombre no es posible localizar el fichero. Sin embargo, si se sustituye el nombre por: **platform:/resource/ConceptualMM2LogicoRelacional/Relational.xmi**, que representa la ubicación física del meta-modelo en el *workspace* de la solución, es posible abrirlo con el editor anteriormente citado y corroborar de esta forma su conformidad contra el meta-modelo relacional destino.

La obtención de código en *Structured Query Language* (SQL) a partir del modelo lógico relacional resultado se puede llevar a cabo haciendo uso de la gama de herramientas *case*² que entre sus funcionalidades contemplan esta actividad (MOMENT CASE: Un prototipo de herramienta CASE*, 2007), (Arantzazu Cámara, 2005), (Un estudio comparativo de herramientas para el modelado con UML, 2005). Se debe corroborar en cada caso la conformidad del meta-modelo relacional usado en el presente trabajo y el manejo por la herramienta en cuestión para evitar diferencias estructurales que puedan obstruir el montaje del modelo lógico relacional en la herramienta seleccionada para compilarlo y obtener el código SQL correspondiente.

² Aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

Conclusiones parciales

El pre-procesamiento del modelo conceptual resultó efectivo al corregir los elementos que constituían obstáculos para la obtención del modelo lógico relacional.

Una vez realizada esta actividad, se construyeron y montaron los meta-modelos origen y destino, comprobando la conformidad del modelo conceptual contra el meta-modelo multidimensional desarrollado a partir de los diagramas especificados en (Luján Mora, 2005).

La posterior definición e implementación de las reglas de transformación para convertir de un modelo conceptual multidimensional a un modelo lógico relacional permitió la obtención de un modelo relacional resultado a partir de un modelo conceptual fuente, facilitando la adaptabilidad de la propuesta a un entorno relacional. El modelo relacional obtenido conformó exactamente el meta-modelo relacional destino.

A través de las reglas implementadas es viable la gestión de cambios en los modelos resultados a partir de elementos específicos de negocio añadidos al modelo conceptual.

Capítulo 4. Análisis de los resultados de la investigación

Introducción

Con el objetivo de validar la solución propuesta se aplica la misma a un caso de estudio relacionado con uno de los proyectos productivos del Centro de tratamiento y análisis de datos (DATEC) con el Ministerio del Poder Popular para la Comunicación y la Información (MPPCI) de la República Bolivariana de Venezuela, donde uno de los componentes es precisamente un almacén de datos para el control de los proyectos del MPPCI. Una versión de la solución se encuentra actualmente desplegada y en uso en Venezuela.

Primeramente se presenta una breve descripción del negocio del MPPCI. Seguidamente se presenta el análisis de resultados asociado a la construcción del almacén de datos con métodos tradicionales. A continuación se aplica el modelo al caso de estudio siguiendo las actividades definidas en la Figura 3. Luego de aplicado el modelo se analizan los resultados teniendo en cuenta:

- Impacto del modelo en las dimensiones de reutilización, multiplataforma y extensibilidad.
- Incidencia del modelo en los tiempos de desarrollo del almacén de datos con respecto a los tiempos de desarrollo de la construcción del almacén de datos con métodos tradicionales.
- Incidencia del modelo en los costos de desarrollo del almacén de datos con respecto a los costos de desarrollo de la construcción del almacén de datos con métodos tradicionales.

Los resultados del caso de estudio se validan además aplicando la técnica del grupo focal (Grau Abalo, y otros, 1999) tomando como grupo un conjunto de especialistas en la materia y enfocando las preguntas a los resultados obtenidos luego de la aplicación del modelo al caso de estudio. Los resultados del caso de estudio fueron avalados también por el Departamento de Almacenes de DATEC. El documento de validación impreso, firmado y acuñado, forma parte de los anexos del presente trabajo.

Se presenta además un resumen del impacto desde el punto de vista social en el MPPCI, del almacén de datos para el control de los proyectos desarrollado.

4.1 Descripción del negocio del MPPCI.

Para lograr un adecuado diseño del Almacén de datos para el MPPCI y después de realizado el levantamiento de requisitos, fueron identificados un grupo de conceptos

de negocio asociados con las estructuras a tener en cuenta para el diseño del almacén de datos.

Las necesidades de los usuarios finales del MPPCI están enfocadas al análisis de los diferentes niveles de decisión definidos en (Delgado Víctore, y otros, 2010). Los análisis correspondientes al nivel de decisión físico se vinculan directamente con los resultados parciales, acciones específicas, tanto de proyectos como de direcciones y los proyectos en general. Los elementos citados anteriormente conforman la EDT de los proyectos del Ministerio.

El nivel de decisión intermedio concentra sus análisis en las direcciones, tanto generales como de línea, mientras que el mando superior agrupa todos los análisis concernientes al Ministerio propiamente y a sus entes adscritos. Los elementos anteriores conforman la estructura jerárquica de la organización o el organigrama.

De cada estructura reflejada en los diferentes niveles de decisión se requiere la medición de un conjunto de indicadores de resultados, ejecución física y financiera, así como se emiten evaluaciones integrales en función del resto de los indicadores. La especificación de los indicadores anteriormente referenciados se encuentra en el documento Sistema de Control de Gestión (Delgado Víctore, et al., 2010).

4.2 Construcción del almacén de datos con métodos tradicionales en el caso de estudio del MPPCI.

Para la construcción del almacén de datos con métodos tradicionales se destinaron 5 meses en el Cronograma del proyecto. Una de las pre-condiciones de este hito lo constituía la definición del Sistema de Control de Gestión (Delgado Víctore, y otros, 2010), documento rector que contiene la especificación de todos los indicadores a concebir en el diseño del almacén. Los requerimientos informativos para la definición de estos indicadores se obtuvieron a partir de entrevistas con los directivos del MPPCI y con especialistas en el tema de gestión de proyectos. No se recomendaba comenzar el diseño del almacén hasta tanto los indicadores no se hubiesen definido y aprobado por parte de los clientes. A pesar de lo anterior, se trabajó de forma paralela para obtener resultados a menor plazo y durante todo el proceso, el diseño del almacén sufrió un gran número de cambios, lo que incidió negativamente en la presentación inmediata y fiable a los clientes, de información que respondiera a las áreas de mayor impacto en los proyectos, disminuyendo así la motivación y expectativas de los mismos con la solución propiamente.

El costo aproximado asociado a la construcción del almacén de datos como componente de la Solución Tecnológica Integral del MPPCI fue de aproximadamente 304 705. 50 dólares (Albet, 2009). Teniendo en cuenta que la estimación anterior está

relacionada con todos los gastos implicados en el desarrollo del almacén de datos, el gasto requerido para el salario de los cuatro trabajadores que desarrollaron el almacén fue de 10 956.52 dólares aproximadamente, asignándole a cada desarrollador 2.85 dólares por cada hora de trabajo en los 5 meses con 24 días hábiles de trabajo en cada uno.

El almacén fue desarrollado atado a la plataforma relacional, debido a que es la que actualmente se usa en el Ministerio. Con esta forma de trabajo, en caso de un cambio de plataforma de base de datos, se tornaría complejo y extenso el proceso de remodelar las estructuras para el nuevo entorno, pues fue inicialmente concebido para un ámbito relacional sin tener en cuenta el modelado independiente de plataforma que facilita esta gestión de cambios a partir de la implementación de reglas de transformación para cada plataforma de base de datos específica. Este principio de diseño complejiza la implantación del almacén de datos en un entorno con otra plataforma de base de datos lo que incide negativamente en la facilidad de reutilización en proyectos similares, de estructuras previamente modeladas cuando las plataformas de base de datos no son compatibles.

4.3 Construcción del almacén de datos aplicando el modelo basado en MDA en el caso de estudio del MPPCI.

Para la construcción del almacén de datos aplicando el modelo, primeramente se ejecutan las actividades de la Figura 3 que obtienen como resultado el modelo conceptual, para posteriormente llevar a cabo el resto de las actividades cuyo resultado es el modelo lógico.

4.3.1 Construcción del modelo conceptual del almacén de datos aplicando el modelo basado en MDA en el caso de estudio del MPPCI.

Para obtener el modelo conceptual se llevan a cabo las dos primeras actividades contenidas en la Figura 3.

Partiendo de los indicadores detallados en (Delgado Víctore, y otros, 2010), se evidencia que los indicadores definidos para controlar la gestión de proyectos en el MPPCI incluyen los contenidos en la propuesta conceptual del modelo del presente trabajo. Adicionalmente se incluyeron otros indicadores debido a las particularidades del negocio del Ministerio propiamente.

A continuación se presentan los nuevos indicadores y las extensiones que suponen en el diseño conceptual del modelo (Ver Tabla 5).

Tabla 5. Indicadores propios del MPPCI

Nuevo indicador	Aditividad	Estructura a la que aplica	Perspectivas de análisis	Hecho existente	Nuevo hecho
Calidad	Aditiva	Resultado parcial	Tiempo	Trabajo del proyecto	
			EDT		
			Organigrama		
Meta real	Aditiva	Resultado parcial	Tiempo	Trabajo del proyecto	
Meta plan			EDT		
			Organigrama		
Cantidad de beneficiarios	Aditiva	Acción específica	Tiempo		Beneficiario
			EDT		
			Organigrama		
			Clasificación		
Cantidad de riesgos	Aditiva	Proyecto	Tiempo		Riesgo
			EDT		
			Organigrama		
			Categoría de riesgo		
			Probabilidad de riesgo		
			Impacto del riesgo		
			Estado del riesgo		
Cumplimiento de las adquisiciones	Aditiva	Proyecto	Tiempo	Trabajo del proyecto	
			EDT		
			Organigrama		
Evaluación integral	No aditiva	Todos los niveles.	Tiempo	Trabajo del proyecto	
			EDT		
			Organigrama		

Algunos de los indicadores anteriores pueden ubicarse como medidas de hechos ya existentes en el modelo conceptual, pues sus perspectivas de análisis coinciden con las de dicho hecho particular. En este caso, muchos indicadores aplican en el paquete de hecho Trabajo del proyecto. Los indicadores a incluir en el paquete de hecho

anterior, evidentemente no son de la misma temática o el mismo concepto de negocio que los ya contenidos en el hecho. A partir de esta decisión de unir todos los indicadores en un mismo hecho por compartir las mismas dimensiones, se deriva otra decisión para la capa de visualización de crear tantos cubos como conceptos de negocio estén almacenados en el hecho. Pudiera cambiarse el nombre de dicho paquete y por ende del hecho contenido en él, con el objetivo de reflejar la inclusión de estos indicadores pues actualmente el nombre supone solo la medición de indicadores de trabajo o ejecución física. El nombre Indicadores generales aplica para dicha tarea.

A la hora de extender el modelo conceptual con un conjunto de análisis cuyas dimensiones coinciden con las de algún paquete estrella ya existente, se propone adicionarlas en dicho paquete. Esto une un conjunto de conceptos que quizás no tengan relación alguna, pero evita la creación de nuevos paquetes estrellas cuyos paquetes de dimensión sean los mismos.

El resto de los indicadores, al implicar nuevas dimensiones, requieren de paquetes estrellas distintos a los ya existentes para incluirse en el modelo conceptual.

A continuación se muestra la matriz bus resultado de la extensión del modelo conceptual del almacén de datos para controlar la gestión de los proyectos del MPPCI.

Tabla 6. Matriz bus del modelo conceptual para el MPPCI

Dimensiones/ Paquetes estrellas	Trabajo del proyecto	Estadísticas del proyecto	Estadísticas del organigrama	Costo real	Financiamiento real	Costo plan	Financiamiento plan	Integralidad del proyecto	Curva de control	Beneficiario	Riesgo
Tiempo	X	X	X	X	X	X	X	X	X	X	X
Estado		X	X								
Prioridad		X	X								X
EDT	X	X		X	X	X	X	X	X	X	X
Organigrama	X	X	X	X	X	X	X	X	X	X	X
Partida presupuestaria				X		X					
Fuente de financiamiento					X		X				
Indicador actividad		X									
Categoría actividad		X									
Intervalo por ciento		X	X								
Clasificación proyecto			X								
Clasificación beneficiario										X	
Categoría riesgo											X
Probabilidad riesgo											X
Impacto riesgo											X

Como se evidencia en la tabla anterior, solo se añaden dos nuevos paquetes estrellas, el de Beneficiario y el de Riesgo. Ambos se relacionan con las dimensiones comunes (Tiempo, EDT y Organigrama) para el resto de los paquetes estrellas. La Figura 24 muestra la estructura de paquetes resultado del modelo conceptual extendido para el MPPCI.

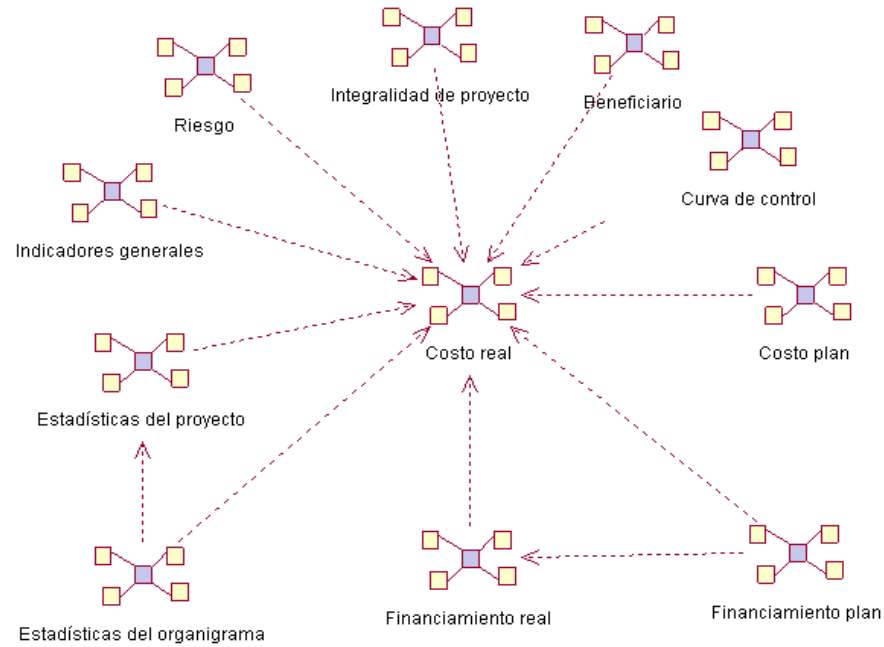


Figura 24. Paquetes estrellas del modelo conceptual para el MPPCI

El contenido de los paquetes estrellas extendidos al modelo se encuentra en el Anexo 3.

4.3.2 Construcción del modelo lógico del almacén de datos aplicando el modelo basado en MDA en el caso de estudio del MPPCI.

Para obtener el modelo lógico se llevan a cabo las dos últimas actividades contenidas en la Figura 3.

En este caso particular, el servidor para el almacenamiento de los datos es el *PostgreSQL*, por tanto el modelo lógico a obtener es del tipo relacional de manera que pueda ser soportado por la tecnología disponible.

Seguidamente se montan los meta-modelos correspondientes en el EMF y se ejecuta el fichero MM2Relational.atl que contiene la implementación de las reglas de transformación. Al fichero obtenido se le sustituye el nombre del meta-modelo del **xmllns** por la ubicación física del mismo como se explica al final del epígrafe Implementación de reglas de transformación para modelo lógico relacional.

A continuación se muestra un fragmento del modelo resultado de las transformaciones anteriores:

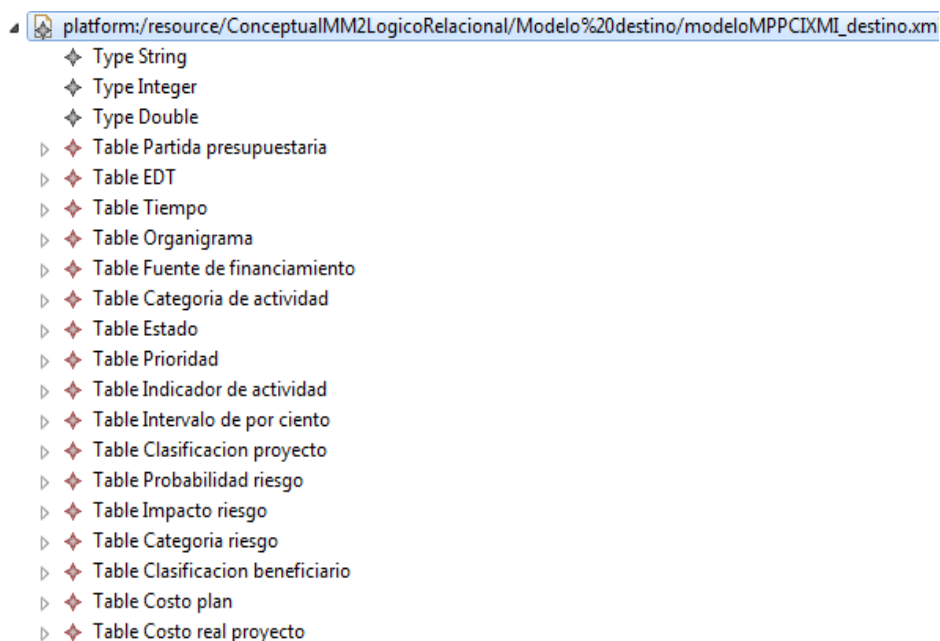


Figura 25. Fragmento del modelo resultado

El modelo anterior conforma exactamente el meta-modelo lógico relacional usado en la solución.

4.4 Análisis de los resultados de la aplicación del modelo al caso de estudio del MPPCI.

Luego de aplicado el modelo al caso de estudio del MPPCI, se analizan los resultados teniendo en cuenta la validación del modelo según criterios de reutilización, gestión de cambios y extensibilidad. Se analiza además la incidencia del modelo aplicado al caso de estudio del MPPCI en la variación de los tiempos y costos con respecto a los valores de dichas variables en la construcción del almacén de datos con métodos tradicionales.

4.4.1 Análisis de la incidencia del modelo en las dimensiones de reutilización, multiplataforma y extensibilidad.

Análisis respecto a dimensión reutilización

Al aplicar el modelo del presente trabajo al caso de estudio, se corroboró que para el desarrollo del almacén del MPPCI se reutilizaron todas las estructuras contenidas en el modelo debido a que aplicaban en su totalidad en los análisis requeridos por los clientes del MPPCI. Por tanto, se reutilizaron un 100% de los paquetes estrellas del modelo en el caso de estudio. El modelado no comenzó desde cero pues al aplicar el modelo, se contó con una base sobre la cual continuar el modelado.

Análisis respecto a dimensión multiplataforma

Se comprobó además el impacto positivo del modelo en la dimensión multiplataforma pues a partir de la construcción de un PIM siguiendo la línea MDA se pudo evidenciar la independencia del modelo con respecto a la plataforma de base de datos a usar y la posibilidad de generar uno o varios PSMs a través de la implementación de reglas de transformación entre el PIM y un PSM.

En el caso particular del MPPCI, el entorno es relacional pues el servidor de base de datos es *PostgreSQL*, por tanto, el PSM generado conforma el meta-modelo relacional. En caso de ocurrir un cambio de plataforma en el MPPCI, sólo basta con implementar las reglas de transformación correspondientes al meta-modelo asociado a la nueva plataforma de base de datos.

Lo anterior demuestra que independientemente de la plataforma de base de datos subyacente, el modelo puede ser aplicado en diferentes entornos sólo implementando las reglas de

transformación al PIM para obtener el PSM soportado por la plataforma de base de datos disponible en el entorno que se aplique.

Análisis respecto a dimensión extensibilidad

Adicionalmente a los indicadores de tiempo, costos y estadísticas presentes en la propuesta, se incluyeron otros particulares del negocio del Ministerio, relacionados con la gestión de beneficiarios y la gestión de riesgos.

La Figura 24 muestra un aumento de la cantidad de paquetes estrellas de nueve a once con el objetivo de modelar indicadores referentes a beneficiarios, riesgos y otros específicos del MPPCI (Ver Anexo 3).

Lo anterior evidencia que el modelo es extensible en el caso de estudio del MPPCI al permitir la inclusión de dos nuevos paquetes estrellas y de nuevos indicadores en paquetes estrellas ya existentes.

4.4.2 Análisis de la incidencia del modelo en los tiempos de desarrollo del almacén de datos con respecto a los tiempos de desarrollo del almacén de datos con métodos tradicionales.

Aplicando el modelo al caso de estudio se comprobó una reducción del tiempo empleado para el desarrollo del almacén de datos para controlar la gestión de proyectos en el MPPCI. La disminución del tiempo se debe precisamente a la base que facilita el modelo, a partir de la cual continuar el diseño del almacén. A continuación se presenta una imagen que muestra una aproximación de la diferencia de tiempos entre la variante de desarrollo con métodos tradicionales y la variante de desarrollo con el modelo basado en MDA.

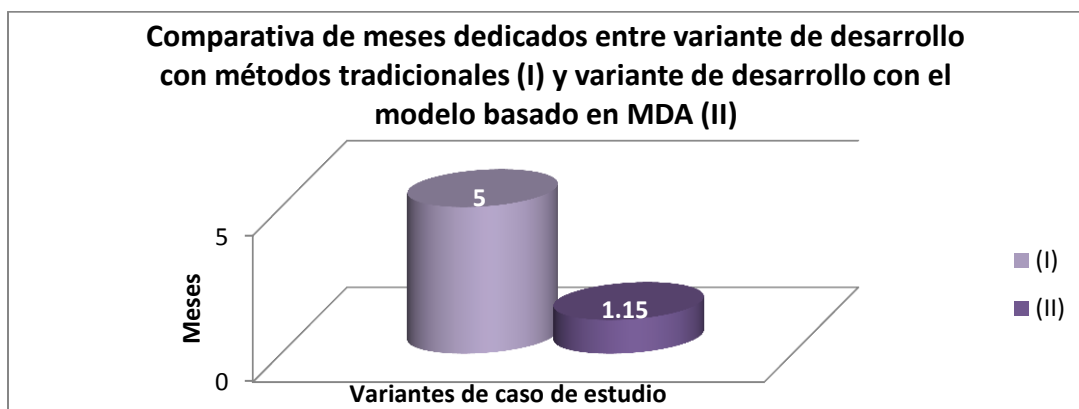


Figura 26. Comparativa de meses dedicados entre variante de desarrollo con métodos tradicionales y variante de desarrollo con el modelo basado en MDA

Teniendo en cuenta que las estructuras del modelo en el almacén de MPPCI representan un 77% del total de estructuras, se puede asumir que el tiempo se reduciría a 0.23 por cinco, que es la cantidad de meses que tomó la construcción del almacén. Por tanto el tiempo a dedicar para modelar las extensiones propias del negocio del Ministerio añadidas al modelo hubiese sido menos de un mes y medio. No obstante, por precaución en función de evitar atrasos por razones externas, se recomienda planificar un intervalo de 2 a 3 meses para construir el almacén de datos sobre la propuesta de modelo del presente trabajo.

4.4.3 Análisis de la incidencia del modelo en los costos de desarrollo del almacén de datos con respecto a los costos de desarrollo del almacén de datos con métodos tradicionales.

Aplicando el modelo al caso de estudio se comprobó una reducción de los costos para el desarrollo del almacén de datos para controlar la gestión de proyectos en el MPPCI. La disminución de costos se debe a que para construir el almacén de datos con el modelo se requieren 1.15 meses aproximadamente mientras que construyendo el almacén de datos sin el modelo el proceso se planificó por 5 meses.

A continuación se presenta una imagen que muestra una aproximación de la diferencia de costos entre la variante de desarrollo con métodos tradicionales y la variante de desarrollo con el modelo basado en MDA.

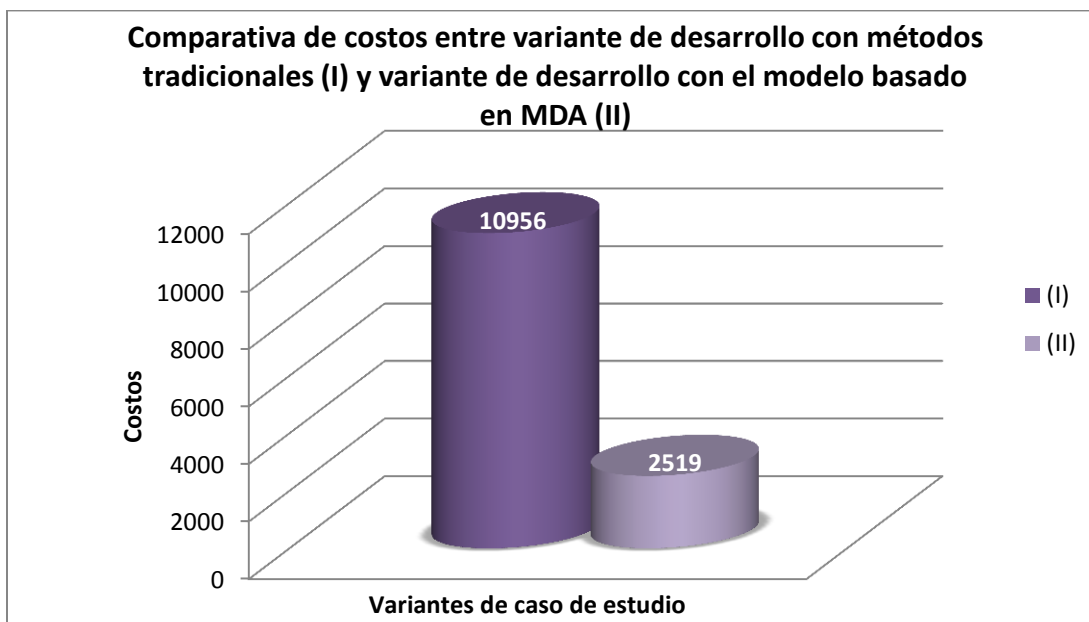


Figura 27. Comparativa de costos entre variante de desarrollo con métodos tradicionales y variante de desarrollo con el modelo basado en MDA

Teniendo en cuenta que la tarifa de pago de cada trabajador por hora es de 2.85 dólares, que la cantidad de trabajadores se mantiene constante en cuatro, que el tiempo de desarrollo se redujo aproximadamente a 1.15 meses según el análisis del epígrafe anterior y que la cantidad de horas de trabajo en los 24 días hábiles de un mes son 192 horas, el costo de construcción del almacén de datos se reduce aproximadamente de 10 956 a 2519 dólares.

La única variable modificada en este caso, que incide en la reducción de los costos es la de tiempo, no obstante teniendo en cuenta el avance en tiempo y desarrollo del almacén de datos que supone el uso del modelo, la cantidad de especialistas encargados del diseño pudiera disminuir también, lo que igualmente incidiría en la reducción de los costos de desarrollo.

4.5 Análisis de la validez del modelo aplicado al caso de estudio, basado en la técnica de grupo focal.

Para la aplicación de esta técnica se realizó una selección de 10 especialistas en temas relacionados con los almacenes de datos, organizados en un grupo focal. Estos especialistas pertenecen a DATEC, específicamente al departamento de Almacenes. El promedio de grado de especialización en el tema del conjunto seleccionado es de un 60% aproximadamente.

Para la realización de la actividad fueron diseñadas un conjunto de preguntas que conducirían el debate (Anexo 4), fundamentadas en las dimensiones analizadas en acápite anteriores. La actividad se llevó a cabo de manera abierta, propiciando el debate y la exposición de experiencias y conocimientos de la temática.

A continuación se muestra una tabla que cuantifica los resultados de la técnica de grupo focal aplicada por cada dimensión a validar en el modelo. La respuesta Sí indica la incidencia positiva del modelo en la dimensión, la respuesta No indica la no incidencia del modelo en la dimensión y la respuesta Parcialmente indica la incidencia parcial del modelo en la dimensión.

Tabla 7. Resultados de la aplicación de la técnica de grupo focal

Variable	Sí	No	Parcialmente
Reutilización	100%		
Multiplataforma	90%		10%
Extensibilidad	100%		
Tiempo	100%		
Costos	100%		

La tabla anterior muestra que un 100% de los participantes respondieron que sí a las preguntas relacionadas con la reutilización, extensibilidad, tiempo y costos mientras que un 90%

respondió que sí en la pregunta relacionada con la dimensión multiplataforma y el 10% respondió que parcialmente en la misma pregunta.

De manera general, la aplicación de la técnica de grupo focal corroboró la incidencia positiva del modelo en las dimensiones de reutilización, multiplataforma, extensibilidad, tiempo y costos.

4.6 Impacto social del almacén de datos en el MPPCI.

El principal objetivo del almacén de datos para el MPPCI es la optimización de la gestión pública. A partir de los indicadores reflejados en el almacén de datos se contribuye al logro del objetivo anterior mejorando en el proceso, el desempeño de la gestión comunicacional y de información del MPPCI.

El almacén de datos contiene lo referente a los proyectos del MPPCI. En función de apoyar la toma de decisiones, estos datos responden a indicadores que permiten mostrar en el sistema de información la situación real de la ejecución contra la planificación de cada proyecto, mostrando el cumplimiento de los objetivos trazados por el Ministerio y cada uno de sus entes adscritos. Lo anterior contribuye de manera significativa a mejorar los procesos de gestión de proyectos del MPPCI, tributando a una mejor organización, distribución de recursos y obtención de resultados satisfactorios en los proyectos que se ejecutan.

El cumplimiento de los objetivos del Ministerio a partir de la obtención de resultados satisfactorios en los proyectos que ejecuta, favorece significativamente todos los procesos de comunicación e información que se llevan a cabo en la República Bolivariana de Venezuela asociados con el aumento del nivel cultural e informacional de la población venezolana en general.

Conclusiones parciales

En la aplicación del modelo al caso de estudio del MPPCI se reutilizaron un 100% de los paquetes estrellas del modelo en el caso de estudio evidenciando que el modelo propuesto constituye un marco de referencia reutilizable para la construcción de almacenes de datos para controlar la gestión de proyectos.

Teniendo en cuenta las especificidades del MPPCI, se adicionaron al modelo dos nuevos paquetes estrellas con indicadores de beneficiarios, riesgos y otros específicos del entorno del MPPCI demostrando que el modelo es extensible al aumentar la cantidad de paquetes estrellas de nueve a once.

Se evidenció la dimensión multiplataforma del modelo al aplicarlo al caso de estudio del MPPCI y obtener a partir del PIM, un PSM dependiente de la plataforma de base de datos relacional.

La existencia del PIM ofrece la posibilidad de implementar reglas de transformación para obtener tantos PSMs como se requiera.

La construcción del almacén de datos del MPPCI con métodos tradicionales tomó una cantidad de 5 meses y se gastó un total de 10 956.52 dólares mientras que aplicando el modelo basado en MDA la cantidad de meses se redujo a aproximadamente 1.5 meses y el gasto empleado fue de alrededor 2519 dólares. Lo anterior evidencia la incidencia positiva del modelo basado en MDA en la reducción de los tiempos y los costos de desarrollo de almacenes de datos para controlar proyectos.

Más de un 90% de los participantes respondió afirmativamente a las preguntas realizadas en la técnica de grupo focal. Las respuestas afirmativas corroboraban la incidencia positiva del modelo en las dimensiones: reutilización, multiplataforma, extensibilidad, tiempo y costos. Lo anterior permite reforzar la validación del caso de estudio según las experiencias y conocimientos de especialistas en el tema.

El aval emitido por el Departamento de Almacenes de DATEC corrobora además la validez de los resultados obtenidos al aplicar el modelo al caso de estudio del MPPCI.

Conclusiones generales

Al término del presente trabajo es posible arribar a las siguientes conclusiones generales:

- El estudio de los elementos asociados con la gestión de proyectos realizado permitió la identificación de las áreas de tiempo y costos como las más relevantes y consideradas en la gestión de proyectos teniendo en cuenta incidencia en la obtención de resultados satisfactorios en los proyectos y presencia de los datos asociados en las herramientas de gestión de proyectos respectivamente. Por su parte los aspectos analizados sobre toma de decisiones en el control de proyectos permitieron la selección de un conjunto básico de indicadores de tiempo y costos a tener en cuenta en el modelo del almacén de datos propuesto.
- Para la construcción del modelo se decidió el uso del enfoque MDA por las ventajas en interoperabilidad e integración entre capas que provee, la facilidad de modelado independiente de plataforma y el uso de UML para el modelado de almacenes de datos.
- Siguiendo la línea MDA se obtuvo primeramente un PIM compuesto por nueve paquetes estrellas con indicadores referentes a tiempo, costos y estadísticas generales. La presencia del PIM en el modelo propuesto basado en MDA permite la conversión a PSMs dependientes de diferentes plataformas a partir de la implementación de reglas de transformación correspondientes a la plataforma requerida. Lo anterior demuestra

que el modelo propuesto es multiplataforma pues puede ser usado en entornos con diferentes plataformas de bases de datos.

- A partir del PIM obtenido se generó un PSM relacional implementando un conjunto de reglas de transformación enfocadas a la plataforma de base de datos relacional por constituir la relacional una de las plataformas más usadas hasta el momento en esta línea de trabajo. Las reglas de transformación implementadas pueden ser usadas en la construcción de almacenes de datos cuya plataforma sea relacional, lo que se evidencia en el caso de estudio del MPPCI donde se generó un PSM relacional debido al uso de *PostgreSQL* en la institución a partir del uso de las reglas de transformación implementadas para el entorno relacional.
- Los resultados del caso de estudio del MPPCI evidenciaron el perfil reutilizable del modelo propuesto basado en MDA al usar un 100% de los paquetes estrellas del modelo en el caso de estudio; así como se demostró su extensibilidad al adicionarle dos nuevos paquetes estrellas y otros indicadores según las especificidades del caso de estudio del MPPCI.
- Aplicando el modelo propuesto basado en MDA al caso de estudio del MPPCI se comprobó una reducción del tiempo de desarrollo del almacén de datos del MPPCI de 5 a 1.5 meses incidiendo esto en una reducción de los costos de desarrollo de 10 956.52 a 2519 dólares.
- Finalmente se ratificó la validez de la propuesta con los resultados obtenidos luego de la aplicación de la técnica de grupo focal y con el aval emitido por el Departamento de Almacenes de DATEC.
- De manera general, el almacén de datos para el MPPCI contribuye a mejorar los proyectos ejecutados por la institución, incidiendo positivamente en los procesos comunicacionales e informacionales que constituyen la razón de ser de los proyectos. La mejora de estos procesos tributa a una elevación del nivel de conocimiento de la población venezolana sobre aspectos políticos, económicos, culturales y de otra índole.

Trabajos Futuros

- Aplicar el modelo a otro caso de estudio.
- Transformar el modelo conceptual a otros modelos lógicos dependientes de las plataformas: NoSQL, XML y multidimensional.
- Implementar los flujos de ETL correspondientes a las herramientas de gestión de proyectos más usadas.
- Extender el modelo a cubos de datos difusos.
- Implementar la capa de visualización que contribuya a tomar decisiones en función de los procesos de dirección.
- Extender el modelo teniendo en cuenta las áreas de calidad, adquisiciones, recursos humanos y otras que contribuyan a la integración en la gestión de proyectos.

Referencias bibliográficas

B. Tayntor , Christine. 2010. *Project Management Tools and Techniques for Success*. Florida, USA : CRC Press, Inc., 2010. 1439816301 9781439816301.

López Sanz, Marcos. 2011. *ArchiMeDeS: A Service-Oriented Framework for Model-Driven Development of Software Architectures*. Madrid : Universidad Rey Juan Carlos, 2011.

Sánchez Gutiérrez, Emilio José . 2010. *Una propuesta para el desarrollo de Almacenes de Datos XML*. Móstoles, España : Universidad Rey Juan Carlos, 2010.

A Standard for Representing Multidimensional Properties: The Common Warehouse Metamodel (CWM).
Medina, Enrique y Trujillo, Juan . 2002. 2002, New York, USA : Springer-Verlag Berlin Heidelberg, 2002, Vol. 2435. 10.1007/3-540-45710-0_19.

A Survey of Multidimensional Modeling Methodologies. **Romero, Oscar y Abelló, Alberto . 2009.** Hershey, USA : International Journal of Data Warehousing & Mining, 2009.

A UML profile for the conceptual modelling of data-mining with time-series in data warehouses. **Zubcoff, Jose , Pardillo, Jesús y Trujillo, Juan . 2009.** 6, Butterworth-Heinemann Newton, MA, USA : Web of Science, 2009, Vol. 51.

A. Yinusa, Sheriffdeen y O. Omotosho, I. 2010. *AN OVERVIEW OF PROJECT MANAGEMENT TOOLS: Development of Automated Project Management and Scheduling Systems*. Alemania : VDM Verlag Saarbrücken, 2010. 3639243420 9783639243420.

Adaptation of transformations to metamodel changes. **Garcia, Jokin y Díaz, Oscar . 2010.** 2, Ciudad Real, España : Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, 2010, Vol. 4.

Albet. 2009. *Ficha de proyecto Solución Integral para el Sistema de Control de Gestión del MPPCI*. Caracas : s.n., 2009.

Arantzazu Cámara, D. . 2005. *SELECCIÓN DE HERRAMIENTAS CASE*. Madrid : Universidad Alfonso X El Sabio, 2005.

Artemis International Solutions Corporation. 2001. *Artemis Views*. *Artemis International Solutions Corporation*. [En línea] 2001.
http://www.aiscsp.com/index.php?option=com_content&view=article&id=2&Itemid=6&lang=es.

ATL. 2012. *ATL. Eclipse*. [En línea] 2012. <http://www.eclipse.org/atl/>.

Ballard, Chuck , y otros. 2006. *Dimensional Modeling:In a Business Intelligence Environment*. Utah : Vervante, 2006. 0738496448.

Blaha, Michael. 2010. *Patterns of data modeling*. Florida, USA : CRC Press, 2010.

Bollati, Verónica Andrea . 2011. *MeTAGeM: Entorno de Desarrollo de Transformaciones de Modelos Dirigido por Modelos*. Móstoles : Universidad Rey Juan Carlos, 2011.

COMPARATIVE STUDY OF DATA WAREHOUSE DESIGN APPROACHES: A SURVEY. **Jindal, Rajni y Taneja, Shweta . 2012.** 1, India : International Journal of Database Management Systems, 2012, Vol. 4.

Delgado Víctore, Roberto. 2006. La dirección integrada por proyectos apoyada por la contabilidad en el control de ejecución de los proyectos. *GestioPolis*. [En línea] 2006.
<http://www.gestiopolis1.com/recursos7/Docs/fin/tecnologias-informaticas-aplicadas-a-la-contabilidad.htm>.

Delgado Víctore, Roberto, Delfino Rodríguez, Anisley y García Vejerano, Juniedi. 2010. *Sistema de control de gestión para el MPPCI*. Caracas, Venezuela : MPPCI, 2010.

Desarrollo de almacenes de datos dirigido por modelos. **Trujillo, Juan y Pardillo, Jesús. 2008.** 7, España : JISBD, 2008, Vol. 1.

Designing Secure Data Warehouses by Using MDA and QVT. **Soler, Emilio, y otros. 2009.** 8, New York, USA : Journal of Universal Computer Science, 2009, Vol. 15. 10.3217.

Earned value management in a data warehouse project. **Gowan, J. Art , G. Mathieu, Richard y Hey, Mark B. . 2006.** 1, Cambridge, USA : Emerald Group Publishing Limited, 2006, Vol. 14.

Eclipse Modeling Framework Project (EMF). 2012. Eclipse Modeling Framework Project (EMF). *Eclipse*. [En línea] 2012. <http://www.eclipse.org/modeling/emf/>.

Fernández Sáez, Pedro Antonio . 2009. *Un Análisis Crítico de la Aproximación Model-Driven Architecture*. Madrid : Universidad Complutense de Madrid, 2009.

Franceschini , Fiorenzo , Galetto, Maurizio y Maisano, Domenico . 2007. *Management by Measurement: Designing Key Indicators and Performance Measurement Systems*. New York, USA : Springer Publishing Company, Incorporated, 2007. 3642092276 9783642092275.

García Benítez, Juan Pablo. 2010. *Marco para la transformación de modelos basado en gramáticas de atributos*. Madrid, España : Universidad Complutense de Madrid, 2010.

García Pau, Víctor . 2011. *Hacia la modernización de aplicaciones corporativas dirigidas por modelos de datos*. Valencia : Universidad Politécnica de Valencia, 2011.

Giandini, Roxana Silvia . 2008. *Un Marco Formal para Transformaciones en la Ingeniería de Software Conducida por Modelos*. La Plata : SeDiCI, 2008.

GONZÁLEZ MACHO, LAURA . 2009. *GESTIÓN DEL TIEMPO EN SISTEMAS DATA WAREHOUSE SOBRE BASES DE DATOS*. España : Archivo Digital UPM, 2009.

Grau Abalo, Ricardo , Correa Valdés, Cecilia y Rojas Betancur, Mauricio . 1999. *METODOLOGÍA DE LA INVESTIGACIÓN*. IBAGUÉ : CORPORACIÓN UNIVERSITARIA DE IBAGUÉ, CENTRO DE INVESTIGACIONES, CONSULTORÍA Y EXTENSIÓN A LA COMUNIDAD , 1999.

Hybrid methodology for data warehouse conceptual design by UML schemas. **Di Tria, Francesco , Lefons, Ezio y Tangorra, Filippo . 2012.** 4, USA : Journal Information and Software Technology, 2012, Vol. 54.

Inmon, Bill. 2005. *Building the Data Warehouse*. New York, USA : John Wiley & Sons, 2005.

International Business Machines. 2012. Rational Rose. *IBM Software*. [En línea] 2012. <http://www-01.ibm.com/software/awdtools/developer/rose/>.

Introducción a los almacenes de datos. **Moral Rubia, Jorge . 2009.** 2, s.l. : ICAI, 2009, Vol. 86.

J. Thomson, Alan. 2005. *Indicator-based knowledge management for participatory decision-making*. Amsterdam : Elsevier Science Publishers , 2005. 0168-1699.

K. Chemuturi , Murali y M. Cagley , Thomas . 2010. *Mastering Software Project Management: Best Practices, Tools and Techniques*. Florida, USA : J. Ross Publishing, Inc., 2010. 1604270349 9781604270341.

López, Horacio , y otros. 2009. *Estado del Arte de Lenguajes y Herramientas de Transformación de Modelos*. Montevideo, Uruguay : Pedeciba, 2009.

Luján Mora, Sergio. 2005. *Diseño de almacenes de datos con UML*. Alicante, España : Universidad de Alicante, 2005.

Maigua, Gustavo Gabriel y Fernando López, Emmanuel . 2012. *Buenas Prácticas en la Dirección y Gestión de Proyectos Informáticos*. Argentina : Editorial de la Universidad Tecnológica Nacional [edUTecNe], 2012. 978-987-1896-01-1.

Microsoft. 2011. Microsoft Project 2010. *Microsoft Project 2010*. [En línea] 2011. <http://www.microsoft.com/project/en-us/solutions.aspx>.

*MOMENT CASE: Un prototipo de herramienta CASE**. **Gómez, Abel , y otros. 2007.** Zaragoza, España : Actas de las XII Jornadas de Ingeniería del Software y Bases de Datos, 2007. 978-84-9732-595-0.

Neil, Carlos Gerardo . 2010. *Diseño de un Almacén de Datos Histórico en el Marco del Desarrollo de Histórico en el Marco del Desarrollo de*. La Plata, Argentina : Universidad Nacional de La Plata, 2010.

Object Management Group (OMG). 2009. *OMG Unified Modeling Language™ (OMG UML), Superstructure*. U.S.A : Object Management Group, Inc, 2009.

Object Management Group. 2003. *Model-Driven Architecture (MDA)*. New York, USA : John Wiley & Sons, Inc., 2003.

Object Oriented Data Modeling for Data Warehousing. **Alaskar, Kamal y Shaikh, Akhtar . 2009.** 2, India : International Arab Journal of e-Technology, 2009, Vol. 1.

OMG. 2012. Catalog of OMG Modeling and Metadata Specifications. *OMG*. [En línea] 2012. http://www.omg.org/technology/documents/modeling_spec_catalog.htm#OCL.

Oracle. 2011. Primavera Contract Management, Business Intelligence Publisher Edition. *Oracle*. [En línea] 2011. <http://www.oracle.com/us/products/applications/primavera/contract-management/overview/index.html>.

PMI. 2009. *Guía de los Fundamentos de la Dirección de Proyectos*. Pennsylvania : Project Management Institute, Inc., 2009.

Poole, John, y otros. 2003. *Common Warehouse Metamodel (CWM)*. New York, USA : John Wiley & Sons, Inc., 2003.

Power, Daniel J. y Sharda Dr, Ramesh. 2009. *Springer Handbook of Automation*. USA : Springer Berlin Heidelberg, 2009. 978-3-540-78831-7.

Pressman, Roger S. 2005. *Ingeniería de software, un enfoque práctico, Sexta Edición*. España : McGraw-Hill, 2005. 9701054733.

TRANSFORMACIÓN DE ESQUEMAS MULTIDIMENSIONALES DIFUSOS DESDE EL NIVEL CONCEPTUAL AL NIVEL LÓGICO. **Carrera Sepúlveda, Sabina, Varas Contreras, Marcela y Urrutia Sepúlveda, Angélica. 2010.** 2, Chile : Revista chilena de ingeniería, 2010, Vol. 18.

Un estudio comparativo de herramientas para el modelado con UML. **Bernardo Quintero, Juan , y otros. 2005.** 137, Medellín, Colombia : REVISTA Universidad EAFIT, 2005, Vol. 41.

Usando ATL en la transformación de modelos multidimensionales temporales. **Neil, Carlos, Baez, Martin y Pons, Claudia. 2007.** Buenos aires, Argentina : XIII Congreso Argentino de Ciencias de la Computación, 2007.

V. Rosson , Constance . 1988. *Management Indicators: Assessing Product Reliability and Maintainability*. Virginia, USA : Virginia Polytechnic Institute & State University Blacksburg, 1988.

Zepeda Sánchez, Leopoldo Zenaído . 2008. *Metodología para el Diseño Conceptual de Almacenes de Datos*. Valencia, España : Universidad Politécnica de Valencia, 2008.