

INTERVENCION EN LA TRANSFORMACION DEL MODELO CIM AL MODELO PIM: UNA  
REVISIÓN DE LITERATURA

Cesar Alexander Zapata Jiménez<sup>1</sup>, José Alejandro Gutiérrez Blandón<sup>2</sup>

<sup>1</sup> Especialización ingeniería de software. Universidad de Medellín. [purpura0829@gmail.com](mailto:purpura0829@gmail.com)

<sup>2</sup> Especialización ingeniería de software. Universidad de Medellín. [jalegut@gmail.com](mailto:jalegut@gmail.com)

Asesor metodológico: Jesús Andrés Hincapié Londono

Programa: Ingeniería de Software

Facultad de Ingeniería

Medellín

2014

# INTERVENCION EN LA TRANSFORMACION DEL MODELO CIM AL MODELO PIM: UNA REVISIÓN DE LITERATURA

Cesar Alexander Zapata Jiménez<sup>1</sup>, José Alejandro Gutiérrez Blandón<sup>2</sup>

<sup>1</sup> Especialización ingeniería de software. Universidad de Medellín. [purpura0829@gmail.com](mailto:purpura0829@gmail.com)

<sup>2</sup> Especialización ingeniería de software. Universidad de Medellín. [jalegut@gmail.com](mailto:jalegut@gmail.com)

## Resumen

La orientación a la implementación de modelos en el desarrollo de software es esencial para potencializar y facilitar la aplicabilidad y el uso de los diferentes elementos de los sistemas de información. Dichos modelos proveen los elementos y herramientas necesarias para encausar la metodología dentro de la Arquitectura de Software Dirigida por Modelos (MDA), de la misma forma que los frameworks apoyan el desarrollo tradicional. MDA separa los modelos por capas, con el objetivo de poder llevar las representaciones más abstractas de los requerimientos de un sistema empresarial, hasta el código fuente de una aplicación digital por medio de transformaciones. La OMG (Object Management Group) explica estas capas como diferentes modelos. En este artículo se presenta una revisión de aquellos documentos publicados en los últimos años que tratan la intervención de la transformación del modelo CIM al modelo PIM, donde se tendrá como referencia los lenguajes BPMN y UML, para representar CIM y PIM respectivamente, utilizando como herramienta de transformación QVT.

**Palabras clave:** OMG, CIM, PIM, Transformación de modelos, QVT, BPMN, UML.

## **1. Introducción**

La creciente complejidad de las aplicaciones de software y la evolución tan rápida de las nuevas tecnologías, hacen que las inversiones en tecnologías concretas sean demasiado volátiles. El Desarrollo de Software Dirigido por Modelos (DSDM), también denominado (MDD) por su acrónimo en inglés, (Model Driven Development) es una propuesta que atribuye a los modelos el papel principal de todo el proceso de desarrollo. Con él a su vez nace la Arquitectura de Software Dirigida por Modelos (MDA), un paradigma enfocado en separar los requerimientos de una organización de los requerimientos tecnológicos. MDA separa los modelos por capas, con el objetivo de poder llevar las representaciones más abstractas de los requerimientos de un sistema empresarial hasta el código fuente de una aplicación.

Sin embargo, como lo es en un gran porcentaje de las veces, la complejidad de las estrategias de negocio no sigue un lineamiento estandarizado que permita con facilidad, identificar los artefactos tecnológicos que representen con mayor precisión los objetivos de negocio. Esta dificultad para abstraer los escenarios de una organización incurre en el aumento de los esfuerzos para estructurar los elementos que debe intervenir en las diferentes capas de MDA.

Dichos esfuerzos se ven focalizados en las transformaciones que son requeridas para llevar las abstracciones entre las diferentes capas, trayendo como consecuencia la realización de modificaciones o construcción de artefactos que apoyen o complementen las herramientas de transformación estándar para lograr que estos procesos sean ideales para el desarrollo de software dirigido por modelos.

Este artículo hará una revisión de documentos que traten el tema de la intervenciónen la transformación de modelos, ya que es un tema que permite

conocer cómo evolucionan las piezas del sistema a lo largo del proceso de transformación y como se relacionan entre sí. Este contenido de información puede utilizarse en diferentes actividades como el análisis de impacto del cambio, la toma de decisiones de diseño o el mantenimiento. Así, dada la importancia de la intervención en la transformación de los modelos, sería deseable que las metodologías de desarrollo dieran soporte para la gestión de la intervención con el fin de identificar una posible forma o estrategia para que las personas puedan ejecutar estos procesos, partiendo desde los modelos más abstractos de (MDA), como son el Modelo Independiente de la computación (CIM) y el Modelo Independiente de la Plataforma (PIM), ya que al poder intervenir en estas transformaciones, se aumenta la reutilización, el mantenimiento y la comprensión de los modelos, alcanzando la facilidad de analizar de manera sencilla que requisitos han sido cubiertos y que elementos se verán afectados por un cambio en un momento dado, dicho esto, es necesario hacer énfasis en la importancia que tiene la intervención en la transformación de modelos ya que por medio de esta será posible poder lograr cumplir todos los objetivos propuestos en el momento de desarrollar aplicaciones dirigidas por modelos.

De esta forma se llevará a cabo un acercamiento a las herramientas empleadas para cada modelo y la tecnología utilizada para su transformación. Este artículo a su vez se divide en la siguiente estructura: en la sección II se hará la pregunta de investigación, se llevara a cabo unos criterios de inclusión y exclusión de los artículos que se van a tener en cuenta para esta revisión y se seleccionaran los artículos, en la sección III se evaluaran los trabajos seleccionados, se propondrán unos criterios de evaluación, se obtendrán los resultados de la evaluación, en la sección IV se hará la discusión de este artículo, en la sección V se mostraran algunas conclusiones y en la sección VI se pondrán las referencias de todos los artículos utilizados para esta revisión de literatura (A, Mendez, Perelli, Pésico, & B, 2012).

## **2. Método**

### ***2.1 Preguntas de investigación***

¿Cuáles son las estrategias que permiten intervenir en la transformación del modelo (CIM) al modelo (PIM)?

### ***2.2 Criterios de inclusión y exclusión***

El criterio de inclusión para esta revisión parte del análisis realizado al resumen, introducción, conclusiones, palabras claves y el título, obtenido en los artículos, con el fin de identificar si el planteamiento del mismo está orientado a la intervención y transformación de los modelos CIM a PIM, además se van a incluir aquellos textos que se enfocan en las utilidades ofrecidas en la plataforma eclipse RCP que ayuden a llevar a cabo estos procesos tales como son las herramientas (BPMN), (UML), (QVT); de igual manera se hace énfasis en las palabras claves de dichos contextos para determinar la oportuna veracidad y profundidad de la información. Los artículos serán excluidos si surgen confusiones en el análisis del resumen, introducción y conclusiones, además se descartarán aquellos contextos que no tienen un enfoque claro y no hacen un aporte notable para esta revisión de literatura.

## 2.3 Trabajos seleccionados (Tabla1)

ID	Título	Autores	Fecha	Fuente
1	Integración de modelos BPMN en ambientes MDA	Martinez Ignacio, Mendez Lautaro, Nahuel Leopoldo, Perelli Julián, Pérsico Mauricio, Giandini Roxana	2012	<a href="http://sedici.unlp.edu.ar/bitstream/handle/10915/23737/Documento_completo.pdf?sequence=1">http://sedici.unlp.edu.ar/bitstream/handle/10915/23737/Documento_completo.pdf?sequence=1</a>
2	Una aproximación a la generación automática de código en un contexto MDD sobre modelos BPMN	Ignacio Martínez A., Lautaro Mendez, Julián Perelli, Mauricio Pérsico, Nahuel Santos B	2012	<a href="http://www.41jaiio.org.ar/sites/default/files/24_EST_2012.pdf">http://www.41jaiio.org.ar/sites/default/files/24_EST_2012.pdf</a>
3	Hacia una integración de MDA y el Proceso Unificado a través de reglas de transformación QVT	Ariel Arsautte, Marcelo Uva, Fabio Zorzan, Marcela Daniele, Paola Martellotto, Ariel Gonzalez, Mariana Frutos	2012	<a href="http://www.41jaiio.org.ar/sites/default/files/353_ASSE_2012.pdf">http://www.41jaiio.org.ar/sites/default/files/353_ASSE_2012.pdf</a>
4	UN MOTOR DE TRANSFORMACIÓN DE MODELOS CON SOPORTE PARA EL LENGUAJE QVT RELATIONS	Pascual Queralt, Luis Hoyos, ArturBoronat, José Á. Carsí e Isidro Ramos	2011	<a href="http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-227/paper14.pdf">http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-227/paper14.pdf</a>
5	Trazabilidad de Requisitos en Almacenes de Datos basada en MDA	Alejandro Mate Juan Trujillo	2011	<a href="http://hdl.handle.net/10045/21183">http://hdl.handle.net/10045/21183</a>
6	Análisis basado en variables para trazabilidad en transformación de modelos	Omar Martínez Grazzi, Claudia Pons	2012	<a href="http://www.41jaiio.org.ar/sites/default/files/145_ASSE_2012.pdf">http://www.41jaiio.org.ar/sites/default/files/145_ASSE_2012.pdf</a>

### 3. Evaluación de los trabajos seleccionados

#### 3.1 Criterios de evaluación

1. Estos documentos deben promover el uso de herramientas que apoyen el proceso de desarrollo de software, que permitan definir los modelos y a la vez faciliten la transformación entre ellos.
2. Los artículos deben proponer nuevas ideas para intervenir en la transformación de modelos.
3. Los modelos independientes de la computación (CIM) y los modelos independientes de la plataforma (PIM) deben ser las áreas tratadas en estos artículos.
4. La automatización de la transformación entre modelos que permita a los usuarios definir sus propias innovaciones y ejecutarlas posteriormente, debe de ser un tema propuesto por estos documentos.
5. Estos documentos deben definir Metamodelos específicos para cada escenario o Metamodelos genéricos para la intervención.

Escala: literal Creciente para calificar los artículos de acuerdo a los criterios de evaluación

Mínimo aprobatorio: 3 (Bueno)

Máximo aprobatorio: 5 (Excelente)

Excelente	5
Muy Bueno	4
Bueno	3
Regular	2
Insuficiente	1

***Escala: literal Creciente para calificar (Tabla 2)***

### 3.2. Resultado de la evaluación (Tabla 3)

		Criterios de evaluación					Resultado
		Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5	
Trabajos	1	4	3	4	4	5	<b>4</b>
	2	4	3	5	5	5	<b>4.4</b>
	3	4	5	4	4	4	<b>4.2</b>
	4	3	4	5	5	4	<b>4.2</b>
	5	4	5	4	4	4	<b>4.2</b>
	6	4	5	3	5	4	<b>4.2</b>

El Artículo 1 propone la integración de los lenguajes BPMN, UML, QVT con el objetivo de construir modelos más precisos y generar transformaciones entre los modelos CIM a PIM, a su vez propone la creación de plugins que definen meta modelos en la plataforma EMF (Eclipse Modeling Framework) para llevar a cabo dichas transformaciones, este escrito conjuntamente expone unos ejemplos prácticos de aplicación donde se prueban dichas transformaciones basadas en documentación existente.

El artículo 2 propone algunos framework como GMF y QVT de Eclipse, para llevar a cabo transformaciones entre modelos, pero a pesar de que en este texto se integran varias tecnologías que permiten crear herramientas de transformación de muy alto nivel sin mayores inconvenientes, el grado de originalidad es bajo, ya que se enfoca básicamente en obtener información de otras fuentes.

Un factor importante de este documento es que posee información relacionada con modelos, meta modelos, transformaciones entre otros, donde sugiere que se debe contar con información de los meta modelos ya que describen las representaciones de cada elemento del modelo y a la vez define las restricciones que debe cumplir cada uno de ellos.

El artículo 3 es un texto que se basa en información ya existente para aplicar una nueva propuesta a la transformación de modelos, este documento hace énfasis en la automatización de procesos unificados, desde la Captura de Requisitos, hasta su Implementación, define reglas QVT para la transformación entre los modelos fuente y objetivo, promueve el uso eficiente de sistemas en el proceso de desarrollo de software para definir los modelos y facilitar transformaciones entre ellos y propone además un conjunto de reglas QVT que establecen una transformación de forma automática entre los modelos producidos en las etapas de Captura de Requisitos y Análisis .

El artículo 4 parte de los conceptos teóricos existentes y continúa con la presentación de un prototipo que ofrece soporte para el lenguaje QVT lo cual facilita la intervención y transformación de modelos CIM a PIM.

Este texto proporciona soporte para un lenguaje estándar dentro del framework de gestión de modelos MOMENT “Motor de transformación definido como plugins dentro de Eclipse”, además sugiere algunas herramientas que proporcionan la automatización de transformaciones de modelos que permite a los usuarios definir sus propias transformaciones y ejecutarlas posteriormente.

El artículo 5 parte de conceptos teóricos y presenta nuevas formas de poder ejecutar transformaciones de modelos dentro del marco (MDA), este presenta un meta modelo de trazas adaptado e integrado en el marco de la arquitectura dirigida por modelos además de la implementación de reglas QVT para automatizar la generación de las mismas.

El artículo 6 presenta una propuesta de soporte de trazabilidad en transformación de modelos mediante el análisis del código QVT, lo que permite la deducción de trazas entre los modelos origen y destino a partir de la especificación de la transformación, de forma sistemática.

#### 4. Discusión

El resultado de esta revisión de literatura da respuesta al interrogante ¿Cuáles son las estrategias que permiten intervenir en la transformación del modelo (CIM) al modelo (PIM)? este identifica varias de ellas y manifiesta que la mayoría se caracterizan por tener algo en común, como es el uso de las herramientas utilizadas y el enfoque especial en Operational Query View Transformation (QVT), un conjunto de estándares para la transformación de modelos que cuenta con tres lenguajes que son: dos declarativos llamados relation y core, y uno de naturaleza imperativa llamado operational mappings. Estas estrategias fueron identificadas como: El desarrollo de un marco metodológico que integra el uso de lenguajes BPMN UML y QVT, la creación de reglas QVT que establecen transformaciones de forma automática entre los modelos, el diseño de prototipos que ofrece soporte para el lenguaje QVT y la creación de Metamodelos de trazas adaptados a algún sistema en particular.

Lo anteriormente expuesto se puede corroborar con los resultados obtenidos en los diferentes documentos utilizados para esta revisión, donde los autores plantean diferentes tipos de propuestas, como es el caso del señor Ignacio Martínez en el que se identifica el desarrollo de una estructura que integra dos plugins para el soporte de diagramas BPMN y UML, El plugin STP BPMN provee las herramientas necesarias para la edición de diagramas BPMN, mientras que el plugin gvcase-uml2 con La utilización del lenguaje QVT, brinda el lenguaje y las dependencias necesarias para la traducción entre modelos (Ignacio et al., 2012).

De igual manera en el artículo del señor Ariel Arsaute se identifica que las transformaciones en QVT requieren al menos de dos modelos, uno fuente y otro objetivo. En esta propuesta se utiliza la herramienta MediniQVT, integrada en Eclipse, esta herramienta utiliza EMF para la representación de modelos, está diseñada para la transformación de los mismos, permitiendo un rápido

desarrollo, mantenimiento y particularización de reglas de transformación de procesos específicos. (Arsaute et al., 2012).

La propuesta del señor Pascual Queralt es un motor de transformaciones llamado MOMENT-QVT donde se identifica que el proceso de transformación consta de tres fases que son análisis, proyección y ejecución a Maude y proyección a EMF, estas tres fases se llevan a cabo, con un conjunto de componentes funcionales que forman la arquitectura de MOMENT-QVT, estos mecanismos son: QVT Parser, QVT Relations, MOMENT Registry, MOMENT Relations, OCL Parser ( Queralt, Hoyos, 2011).

A diferencia de los casos expuestos por los autores anteriormente descritos existen otras posibilidades para intervenir en la transformación de modelos, como lo propone, el señor Omar Martínez quien presentó una técnica de análisis de trazabilidad basado en variables que permite identificar en la definición de una transformación escrita en lenguaje QVT ciertos patrones que posibilitan la deducción automática de trazas, estos patrones son identificados como: *Inferencia de trazas* mediante una variable auxiliar, *Inferencia de trazas* mediante una expresión en función de una variable auxiliar, *Inferencia de trazas* mediante el uso de una constante, *Inferencia de trazas* mediante una variable auxiliar definida como función en la cláusula *Where* , esta idea ha sido implementada en un prototipo llamado QVTrace, el cual está diseñado como plugin de Eclipse, y pensado para que interactúe con otras herramientas de desarrollo (Grassi, & Pons, 2012), este argumento es apoyado por el señor Alejandro Mate quien propone que al incluir la trazabilidad, se aumenta la reutilización, el mantenimiento y la comprensión de los modelos y se hace más fácil analizar de manera sencilla que requisitos han sido cubiertos y que elementos se verán afectados por un cambio en un requisito dado (Mat & Trujillo, 2011).

De acuerdo a los resultados obtenidos en las diferentes estrategias para intervenir en la transformación de modelos y a pesar del excelente aporte para esta forma de desarrollo, es evidente la importancia de estar familiarizado con

el Eclipse Modeling Framework (EMF) y los plugin necesarios para llevar a cabo dichas tareas, debido al impacto que estas herramientas generan en la ejecución de dichos procesos; siendo sensatos es necesario preguntar: ¿Se justifica aprender estas técnicas para tener la satisfacción de tener el conocimiento de una nueva forma de desarrollar sistemas de computación?, es el MDSD un aporte válido para el futuro? o se puede ver como un tema de moda?, estos interrogantes surgen de acuerdo al esfuerzo que se debe hacer en aprender y entender todas las técnicas y funcionalidades necesarias que exigen las herramientas utilizadas para llevar a cabo estas técnicas.

Dentro de los resultados de esta revisión de literatura se han identificado una serie de sucesos anómalos como es el caso del artículo “Integración de modelos BPMN en ambientes MDA” escrito por el señor Ignacio Martínez ya que no profundiza en el desarrollo de su propuesta planteada y no queda claro cuáles son las pautas a seguir para obtener el logro de estos objetivos lo cual genera muy pocas probabilidades en la adopción de estas herramientas, debido quizás a la falta de documentación .

En el caso del artículo “Una aproximación a la generación automática de código en un contexto MDD sobre modelos BPMN” queda una especie de incertidumbre, al afirmar, que aún falta camino que recorrer para lograr una transformación que nos brinde código en un determinado lenguaje, “modelo a texto “, quizás esto posiblemente se presente por el grado de dificultad en el aprendizaje de las herramientas ofrecidas en la plataforma de Eclipse. En el artículo “Trazabilidad de Requisitos en Almacenes de Datos basada en MDA” escrito por el señor Alejandro Mate se declara que la derivación automática de código se realiza utilizando transformaciones Modelo a Modelo, especificadas mediante reglas (QVT). Pero de acuerdo a la experiencia adquirida en proyectos reales esta persona manifiesta que a pesar de disponer de este tipo de transformaciones, la carencia de un soporte para trazabilidad imposibilita validar adecuadamente los requisitos, introduciendo un sobre coste cuando estos cambian, lo que induce a generar una serie preguntas tales como: Es

bueno, o no , la integración del estándar QVT en la intervención de la transformación de modelos? , Que tan válido es el aporte que nos brindan los autores incluidos en esta revisión, ya que su enfoque principal está basado en el estándar QVT?

De acuerdo a lo anteriormente escrito se considera que con el buen uso, la buena comprensión y profundidad en los conceptos técnicos de las herramientas existentes para llevar a cabo estos procesos se pueden obtener muy buenos resultados para intervenir en la transformación de modelos; solo basta tener presente las experiencias vividas por los autores incluidos en esta revisión donde se percibe que cumplieron con los objetivos propuestos sea cual sea las herramientas y estrategias utilizadas para llevar a cabo sus propuestas.

A pesar de que el estándar QVT es uno de los enfoques principales en los artículos obtenidos en esta revisión, existen personas quienes dicen que este tiene problemas para manejar los perfiles UML y la única ventaja con la que cuenta es que cumple con un enfoque declarativo al 100%, de acuerdo a esto es necesario recomendar que no es la única opción para intervenir en la transformación de modelos ya que también existe ATL (Atlas Transformation Language), un lenguaje de transformación de modelos donde algunas personas manifiestan que es más usado que QVT, tiene más soporte, la comunidad es mayor y su velocidad es más alta, lo que justifica que es mejor teóricamente hablando.

## **5. Conclusiones**

La importancia de poder intervenir en la transformación de los modelos CIM a PIM ha generado la necesidad de detectar estrategias que permitan llevar a cabo dichos procesos, para esto se diseñó un plan de estudio con el propósito de reconocer algunas ofertas que hagan un aporte al logro de este objetivo. Para

llevar a cabo este trabajo inicialmente surgió la pregunta ¿Cuáles son las estrategias que permiten intervenir en la transformación del modelo (CIM) al modelo (PIM)? de acuerdo a esto se tuvieron en cuenta unos criterios de inclusión y exclusión para los artículos a seleccionar, luego se adquirieron los documentos más adecuados para esta investigación y se propuso unos criterios de evaluación que nos mostraron el grado de importancia que tienen estos textos para esta revisión de literatura .

En esta investigación se ha logrado identificar la importancia de poder intervenir en la transformación de modelos, lo que hace necesario contar con metodologías de desarrollo que implementen posibles formas o estrategias para apoyar a los usuarios en el logro de sus objetivos propuestos y ayuden a aumentar la reutilización, el mantenimiento y la comprensión de los proyectos.

Con la publicación de herramientas que proporcionan la automatización y facilitan a los usuarios definir sus requerimientos para luego ejecutarlos, se logra que los desarrolladores y/o diseñadores sean los responsables de hacer realidad los requerimientos del sistema a desarrollar cuyos beneficios sean un menor número de líneas de código escritas, un diseño de una aplicación o artefacto de software que parte de lo más general a lo más concreto, unas especificaciones de requisitos de usuario a varios niveles, generación de código para plataformas previamente específicas, reducción de costos en el desarrollo de aplicaciones y la interoperabilidad entre los distintos niveles de modelaje.

Con la automatización de procesos unificados, la integración de algunos lenguajes de desarrollo, la definición de reglas QVT, la creación de algunos plugins que definen meta modelos, además de sus restricciones y con el uso de algunos Frameworks, surge la posibilidad de construir modelos más precisos, y generar transformaciones partiendo desde modelos abstractos hasta llegar al código fuente de una aplicación.

A pesar de que los modelos y las transformaciones pasen a ocupar un papel principal, las trazas se deben contemplar como enlaces entre los elementos de dichos modelos que podrán a su vez representarse en otros modelos y por tanto transformarse y generarse, de esta manera gestionar la trazabilidad permite conocer cómo evolucionan los elementos del sistema a lo largo del proceso de desarrollo y como se relacionan entre sí.

Dada la importancia de la trazabilidad en el proceso de los sistemas, sería deseable que las metodologías de desarrollo dieran soporte para la gestión de estas.

A pesar de las técnicas existentes para la intervención en la transformación de modelos, hay que ser conscientes de que se puede presentar algunas debilidades tales como: las reglas al igual que las restricciones asociadas con los metamodelos deben ser modificadas, cada vez que cambian los criterios de decisión debido a que existen medidas y juicios relevantes para ciertas aplicaciones que no son ajustables a otras.

Para concretar es muy importante dar solución a la pregunta de investigación y partiendo de los resultados obtenidos en este trabajo es necesario mencionar que quizás existan una gran cantidad de estrategias para intervenir en la transformación de los modelo modelos CIM a PIM pero de acuerdo a estos resultados se mencionan las estrategias identificadas, estas son: el desarrollo de un marco metodológico que integra el uso de lenguajes BPMN UML y QVT, la creación de reglas QVT que establecen transformaciones de forma automática entre los modelos, el diseño de prototipos que ofrece soporte para el lenguaje QVT y la creación de Metamodelos de trazas adaptados a algún sistema en particular.

## **6. Referencias**

A, I. M., Mendez, L., Perelli, J., Pérsico, M., & B, N. S. (2012). Una aproximación a la generación automática de código en un contexto MDD sobre modelos

- BPMN, 9. Retrieved from [http://www.41jaiio.org.ar/sites/default/files/24\\_EST\\_2012.pdf](http://www.41jaiio.org.ar/sites/default/files/24_EST_2012.pdf)
- Arsaute, A., Uva, M., Zorzan, F., Daniele, M., Martellotto, P., & Frutos, M. (2012). Hacia una integración de MDA y el Proceso Unificado a través de reglas de transformación QVT, 14. Retrieved from [http://www.41jaiio.org.ar/sites/default/files/353\\_ASSE\\_2012.pdf](http://www.41jaiio.org.ar/sites/default/files/353_ASSE_2012.pdf)
- Ignacio, M., Lautaro, M., Leopoldo, N., Julián, P., Mauricio, P., & Roxana, G. (2012). Integración de modelos BPMN en ambientes MDA, 10. Retrieved from [http://sedici.unlp.edu.ar/bitstream/handle/10915/23737/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/23737/Documento_completo.pdf?sequence=1)
- Mat, A., & Trujillo, J. (2011). Trazabilidad de Requisitos en Almacenes de Datos basada en MDA, 1, 14. Retrieved from <http://hdl.handle.net/10045/21183>
- Queralt, P., Hoyos, L., Boronat, A., & Carsí, J. Á. (2011). Un motor de transformación de modelos con soporte para el lenguaje qvt relations, 1–10. Retrieved from <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-227/paper14.pdf>
- Vara, J. M., Bollati, A., & Marcos, E. (2011). Gestión de la trazabilidad en el desarrollo dirigido por modelos de Transformaciones de Modelos : una revisión de la literatura, 1, 14. Retrieved from [http://lbd.udc.es/jornadas2011/actas/JISBD/JISBD/S5/Regulares/jisbd2011\\_submission\\_71.pdf](http://lbd.udc.es/jornadas2011/actas/JISBD/JISBD/S5/Regulares/jisbd2011_submission_71.pdf)
- Grassi, O. M., Pons, C., Informática, F. De, & Abierta, C. U. (2012). Análisis basado en variables para trazabilidad en transformación de modelos, 15. Retrieved from [http://www.41jaiio.org.ar/sites/default/files/145\\_ASSE\\_2012.pdf](http://www.41jaiio.org.ar/sites/default/files/145_ASSE_2012.pdf)
- Chicote, C. V., & Cáceres, D. A. (2007). Herramientas Eclipse para Desarrollo de Software Dirigido por Modelos Tabla de contenidos, 1, 1–49. Retrieved from [http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=4&ved=0CEgQFjAD&url=http%3A%2F%2Frepositorio.bib.upct.es%2Fdspace%2Fbitstream%2F10317%2F1216%2F1%2Fhed.pdf&ei=xs2bUr\\_aHYfmkAfunIC4Ag&usg=AFQjCNE-CcJZiO\\_ujyOaQ-FkLnhN7J6b4w&sig2=\\_ic8RdVFRKR\\_F49lZ9igoA](http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=4&ved=0CEgQFjAD&url=http%3A%2F%2Frepositorio.bib.upct.es%2Fdspace%2Fbitstream%2F10317%2F1216%2F1%2Fhed.pdf&ei=xs2bUr_aHYfmkAfunIC4Ag&usg=AFQjCNE-CcJZiO_ujyOaQ-FkLnhN7J6b4w&sig2=_ic8RdVFRKR_F49lZ9igoA)
- Correa, N., & Giandini, R. (2012). Integrando Trazabilidad a la Generación de Casos de Prueba del Sistema : una propuesta MDD / MBT, 14. Retrieved from [http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CCsQFjAA&url=http%3A%2F%2Fcibse2013.ort.edu.uy%2Fpresentaciones%2FSE%2FCorrea\\_Giandini.pdf&ei=2M-bUsvyI5OpkAfju4HICQ&usg=AFQjCNEwuyWB2\\_PK94OBexbn\\_mRpYPBJqw&sig2=Vf5o1UfM8kwd8jgrNksMOQ](http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CCsQFjAA&url=http%3A%2F%2Fcibse2013.ort.edu.uy%2Fpresentaciones%2FSE%2FCorrea_Giandini.pdf&ei=2M-bUsvyI5OpkAfju4HICQ&usg=AFQjCNEwuyWB2_PK94OBexbn_mRpYPBJqw&sig2=Vf5o1UfM8kwd8jgrNksMOQ)