

**II CONVENCION CIENTIFICA INTERNACIONAL
“II CCI UCLV 2019”**

**DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.**



**II CONFERENCIA INTERNACIONAL DE PROCESAMIENTO DE
LA INFORMACIÓN (CIPI 2019)**

Módulo CIM-PIM versión 2.0 de la herramienta jMDA

Module CIM-PIM version 2.0 for jMDA tool

Dr. Rosendo Moreno Rodríguez¹, Ing. Jennifer Ferrer Pacual²

1- Dr. Rosendo Moreno Rodríguez. Universidad Central “Marta Abreu” de Las Villas, Cuba. E-mail: rosendo@uclv.edu.cu

2- Ing. Jennifer Ferrer Pacual. UCLV, Cuba. E-mail: jfp@uclv.cu

Resumen:

Dentro de los diversos paradigmas y marcos de trabajo de la ingeniería de software se encuentra uno que ha ganado fuerza en los últimos años: la Arquitectura Dirigida por Modelos. Concibe el desarrollo a partir de la construcción y transformación de modelos en diferentes niveles de abstracción. El paradigma MDA global incluye 4 etapas con 3 transformaciones entre ellas; mundialmente es conocido el trabajo realizado sobre los módulos PIM (Modelo Independiente de la Plataforma), PSM (Modelo Específico de la Plataforma), y Código (el programa resultante en sí). Por supuesto las transformaciones entre estas etapas para generar automáticamente la siguiente se han implementado en variados CASE, comerciales o no, pero con resultados muy específicos. No obstante el verdadero paradigma incluye una etapa previa CIM (Modelo Independiente de la Computación) y por supuesto su transformación automatizada hacia el PIM. Esto no ha sido tan investigado, ni implementado en herramientas CASE, al menos que se conozcan. Es por supuesto un desafío científico y tecnológico por lo que esta investigación está dirigida a crear un ambiente lógico de preguntas y respuestas dirigidas, que pueda ser utilizado por un usuario o analista de sistemas, orientado a dar como resultado, los

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.



Actores, Casos de Uso, Clases y Actividades lógicas de un futuro sistema de información, y que posibilite a partir de esta interfaz, generar automáticamente los diagramas indicados en fases PIM. El resultado de este trabajo es un módulo perteneciente a la herramienta jMDA en desarrollo desde hace unos 5 años por parte del Centro de Investigaciones de Informática de la UCLV.

Abstract:

Inside the diverse paradigms and marks of work of the software engineering, Directed Architecture for Models (MDA) is one that has won force in the last years. MDA conceives the development starting from the construction and transformation of models in different levels of abstraction. The paradigm global MDA includes four stages with three transformations among them; worldwide the work carried out on the modules PIM is known (Platform Independent Model), PSM (Platform Specific Model), and Code (the resulting program in yes). Of course, the automatically transformations among these stages to generate the following one have been implemented in varied CASEs, commercial or not, but with very specific results. Nevertheless, the true paradigm includes a previous stage CIM (Computer Independent Model) and of course, the automated transformation toward the PIM. It has not been so investigated, neither implemented in CASEs tools; at least that they know each other. It is a scientific and technological challenge, for what, this investigation is directed to create a logical atmosphere of questions and directed answers that it can be used by any user or analyst of systems, guided to give as a result, the Actors, Use Cases, Classes, Interfaces, and logical Activities of a future system of information, and that it facilitates starting from this interface, to generate the diagrams indicated in phases PIM automatically.

Palabras Clave: Ingeniería del software; Arquitectura dirigida por modelos, Módulo Independiente de la Computación.

Keywords: *Software engineering; Model driven architecture, Computer Independent Model.*

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.



1. Introducción

La ingeniería de software es una disciplina que tiene como objetivo crear técnicas y métodos para desarrollar y mantener software de varios tipos con calidad, eficiencia y productividad. Los Sistemas de Información (SI) son un tipo específico de software caracterizados por ser un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes, que permitan desempeñar su actividad de acuerdo a su estrategia de negocio.

Uno de los Sistemas de Información más utilizados son los Sistemas de Gestión de Información que almacenan toda la información que se requiere para la toma de decisiones de forma organizada en registros específicos e identificables, generalmente utilizando Bases de datos. Según (Rafael Camps Paré, 2005) una base de datos de un SI es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del SI y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.

Con el fin de ayudar a la ingeniería de software se han desarrollado una gran variedad de aplicaciones informáticas conocidas por herramientas CASE “*Computer Aided Software Engineering*” cuya función es la de aumentar la productividad en el desarrollo de software reduciendo los costos en términos de tiempo y dinero.(Franky, 2010)

Dichas herramientas se construyen para seguir determinados paradigmas y principios y nos sirven para abarcar todo el ciclo de vida de desarrollo del software brindándonos desde la facilidad de diseñar un proyecto, cálculo de costo hasta la generación de código fuente en algún lenguaje de programación, detección de errores, documentación, entre otras. Se espera que una herramienta CASE genérica, sea del tipo que sea, pueda ser capaz de cumplir con requisitos como: mejorar la productividad en el desarrollo y mantenimiento del software, aumentar la calidad del software, reducir el tiempo y costo

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

**DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.**



de desarrollo y mantenimiento de los sistemas informáticos, mejorar la planificación de un proyecto, aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos, automatizar el desarrollo de software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto, ayudar a la reutilización del software, a la portabilidad y estandarización de la documentación, gestión global en todas las fases de desarrollo de software con una misma herramienta y facilitar el uso de todas las metodologías propias de la ingeniería de software. (Anacleto, 2006)

Como instrumento para abordar la alta complejidad arbitraria en el desarrollo de software surge el paradigma de Desarrollo Dirigido por Modelos (MDD). En la literatura se reconoce la necesidad de utilizar propuestas basadas en MDD para el desarrollo de sistemas de alta complejidad, debido a la importancia de aumentar el nivel de abstracción para mejorar la comprensión, comunicación y análisis por los humanos.(Martínez, 2014) La Arquitectura Dirigida por Modelos (MDA) es una propuesta brindada por OMG para dar solución a los cambios de negocio y de tecnología. Dentro de los enfoques que siguen el MDD, MDA es la de mayor relevancia hasta el momento se desarrolla a partir de tres niveles de abstracción: modelos independientes de la computación (CIM), modelos independientes de la plataforma (PIM) y modelos específicos de la plataforma (PSM). La principal dificultad radica en la marcada diferencia entre los conceptos tratados en los CIM y los PIM. Los CIM ofrecen una vista desde una perspectiva de los involucrados en el negocio, mientras que los PIM constituyen una vista de los desarrolladores de software. De ahí que la mayoría de las propuestas MDA comienzan por los PIM.

Paralelamente y desde hace un poco más de tiempo se vienen desarrollando en el mundo algunas herramientas de apoyo a la ingeniería de software que se adhieren al paradigma MDA y que cuentan con licencias de software libre; sin embargo aún no se pueden considerar herramientas puras MDA pues no incluyen algunas fases y conversiones, o se apoyan en otras herramientas propietarias. La concepción real de MDA puede verse en la siguiente figura.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.

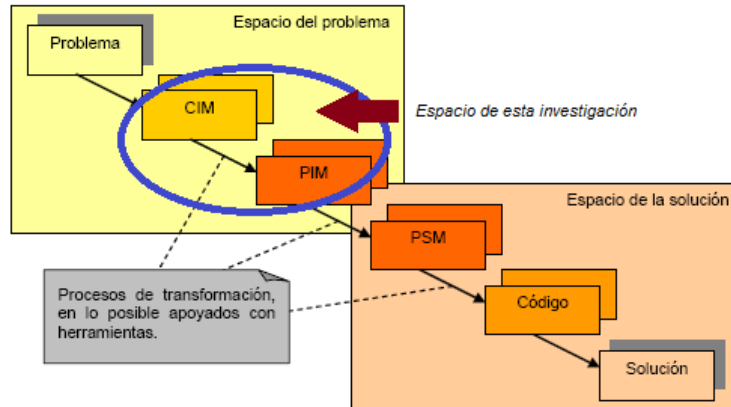


Figura1: Modelos y transformaciones en MDA.

Además, la industria de software nacional no cuenta con ninguna herramienta de este tipo que sea de su autoría netamente; por lo que contar con una aplicación de dichas características, que pueda irse perfeccionando gradualmente pudiera significar un paso más en el desarrollo de nuestra industria en el orden tanto económico como tecnológico. A partir de esta problemática es que en el Centro de Investigaciones de Informática de la UCLV se ha venido trabajando en este sentido.

La tarea “Desarrollo de Herramienta CASE para Arquitectura Dirigida por Modelos jMDA” es de gran importancia para el país, ya desde varios años anteriores se han desarrollado versiones de los módulos PIM-PSM (4 versiones), PSM-Código (3 versión) y CIM-PIM (1 versión); en esta última, que se corresponde con el actual proyecto, los Diagramas de Casos de Uso, Clases y Actividades al nivel de Modelo Independiente de Plataforma (PIM) no se lograron generar automáticamente, lo cual fue logrado con esta versión 2.

2. Metodología

El presente trabajo es una investigación aplicada. Se usaron métodos y técnicas de la Ingeniería del Software como el modelado UML y la programación en Java, para el desarrollo de la herramienta.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.



3. Resultados y discusión

3.1. Modelo Independiente de la Computación (CIM):

Surge en la fase inicial del proceso de desarrollo comprendiendo la modelación del negocio en su totalidad; donde se modelan los requisitos del sistema, se describe la situación en la que el sistema será utilizado y sirve de enlace entre los expertos en el dominio del problema y sus requisitos con los expertos en el diseño y construcción de software. Los requisitos pueden ser representados mediante diagramas de caso de uso, de actividad y de secuencia. Entre los tipos de requisitos más comunes obtenidos en CIM se encuentran los requisitos de usuario, los funcionales, los no funcionales y los organizacionales.

Es evidente que un módulo CIM es muy difícil y diverso de lograr. La estrategia seguida fue la de hacer un módulo CIM para un tipo específico de software, en este caso Sistemas de Gestión de Información. Independientemente de esto, en la literatura se encuentran varios enfoques de desarrollo de módulos CIM como:

1. Creación de un editor de texto en lenguaje natural (pseudocódigo realmente) donde el cliente o experto del problema, lo describa de manera semiestructurada, y a partir de esta descripción (historia de usuario), mediante programa de reconocimiento de ese lenguaje natural (con varias reglas y conocimiento incluido) se logre transformar esa descripción en los elementos necesarios de los diagramas UML de un módulo PIM.
2. Modelado del negocio del problema en cuestión en base a algún lenguaje gráfico que lo logre describir con exactitud (generalmente BPMN) y luego su transformación a los diagramas UML esenciales de un PIM. Esta variante tiene la dificultad del desconocimiento del lenguaje BPMN y las herramientas asociadas por parte de los expertos del problema.
3. Desarrollo de interfaces para específicos tipos de software, de preguntas-respuestas que posibiliten determinar automáticamente los elementos de los diagramas UML deseados. Esta fue la variante seleccionada para nuestro trabajo.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.



Como quiera que el Modelo Independiente de la Plataforma (PIM) representa modelos que describen una solución de software que no contiene detalles de la plataforma concreta de implementación de la solución, se determinó que las transformaciones deseadas en esta versión darían lugar a diagramas de Casas de Uso, de Clases y de Actividades.

3.2. Interfaz de preguntas-respuestas

Como quiera que decidió utilizar la variante de preguntas-respuestas se realizó un estudio sobre las cuestiones básicas de análisis de requisitos para sistemas de gestión de información, entre otras las siguientes:

- ¿Cuántas entidades usted desea controlar en su SI? Descríbalas.
- ¿Qué características tienen esas entidades? Descríbalas.
- ¿Cuáles son las funcionalidades deseadas para su SI? Nómbralas.
- ¿Quiénes serían los encargados de interactuar con esas funcionalidades? Defínalos.

En otra parte de la interfaz pueden preguntarse otros aspectos relativos a características propias del software, como si se desea una interfaz basada en menús y formularios o por Web, etc.

3.3. Herramientas de implementación.

Para el desarrollo de la versión 2.0 del módulo CIM-PIM de la herramienta jMDA se utilizó el entorno de desarrollo integrado libre NetBeans, hecho principalmente para el lenguaje de programación Java, que fue el utilizado en la implementación. Para la codificación se utilizaron bibliotecas contenidas en el JDK versión 8 tales como: swing y awt, para manipular todos los componentes visuales de la aplicación, así como la biblioteca MxGraph, sobre la cual se desarrolló el ambiente gráfico que permite tanto crear, como modificar continuamente los diagramas UML tanto en el modelo PIM como en el PSM que se obtiene automáticamente. Además con esa biblioteca se logró la importante opción de guardar en un formato específico que posibilita la conectividad de este módulo con los otros dos desarrollados (módulo PIM-PSM y módulo PSM-Código). El diagrama de componentes se presenta a continuación:

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.

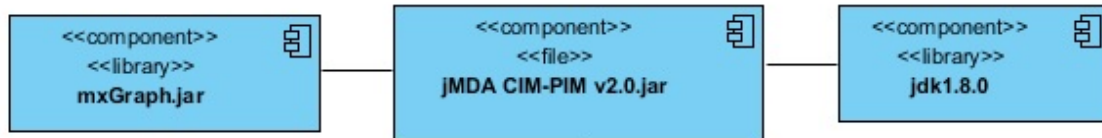


Figura 2: Diagrama de componentes de la herramienta CASE JMDA CIM-PIM v2.0.

3.4. Breve descripción de la herramienta.

Supongamos que se desea modelar un sistema para controlar la información asociada a los estudiantes y profesores pertenecientes a la Facultad de Matemática-Física-Computación; para ello se describirá paso a paso cómo se logra el resultado de la anterior problemática utilizando la herramienta jMDA módulo CIM-PIM.

Paso 1: Crear el nuevo formulario (este paso siempre se va a realizar cada vez que el analista desee generar un diagrama sin importar cuál sea). En la Figura 3 se muestra la creación de un nuevo formulario donde primeramente el analista debe seleccionar el nodo en el árbol del proyecto correspondiente al diagrama que desea modelar, luego hacer doble click sobre este e introducir el nombre del nuevo formulario que desea crear.

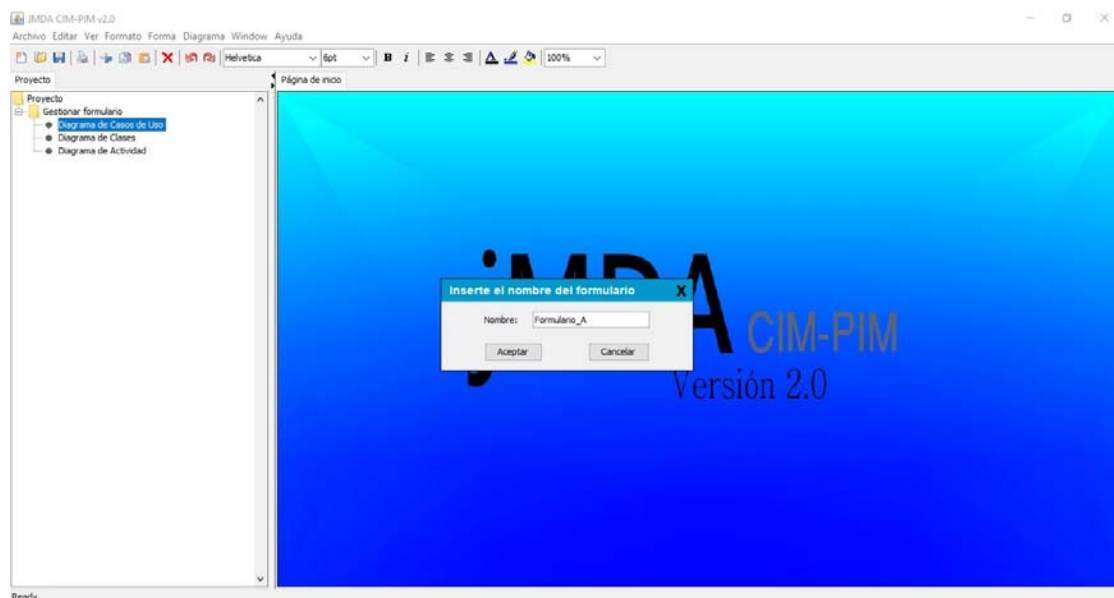


Figura 3: Inicio del módulo para la creación de un nuevo formulario.

Paso 2: Responder las preguntas del formulario. La Figura 4 muestra el ambiente de preguntas-respuestas que se diseñó para lograr el diagrama de casos de uso del modelo CIM.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.



Componente 1	Componente 2	Relación
secretaria	Gestionar_Profesor	Asociación
secretaria	Gestionar_Estudiante	Asociación
secretaria	Gestionar_Profesor	Asociación

Figura 4: Interfaz para determinar elementos de un futuro Diagrama de Casos de Uso.

En la Figura 5 se presenta el ambiente de preguntas-respuestas del modelo CIM que se diseñó para lograr el Diagrama de Clases. El mismo contiene además los formularios para “Adicionar Atributos” y “Adicionar Operaciones”.

Componente 1	Componente 2	Relación
estudiante	profesor	Asociación

Figura 5: Formulario para determinar elementos del Diagrama de Clases.

De la misma manera existe otro formulario para determinar elementos de un Diagrama de Actividades que describa algún caso de uso o funcionalidad.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.



Paso 3: Transformar al modelo PIM. Una vez que el formulario sea respondido el analista puede pasar a seleccionar el botón “Transformar” el cual realiza la transformación del modelo CIM al PIM. En la Figura 6 se muestra el diagrama de casos de Uso obtenido al realizar la transformación correspondiente.

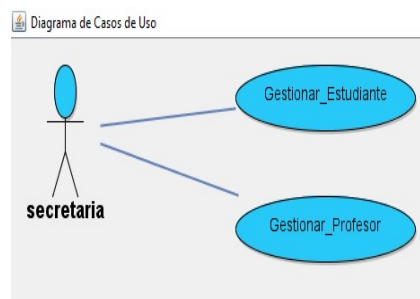


Figura 6: Diagrama de Casos de Uso generado en el modelo PIM.

En la Figura 7 se muestra el diagrama de Clases obtenido al realizar la transformación correspondiente.

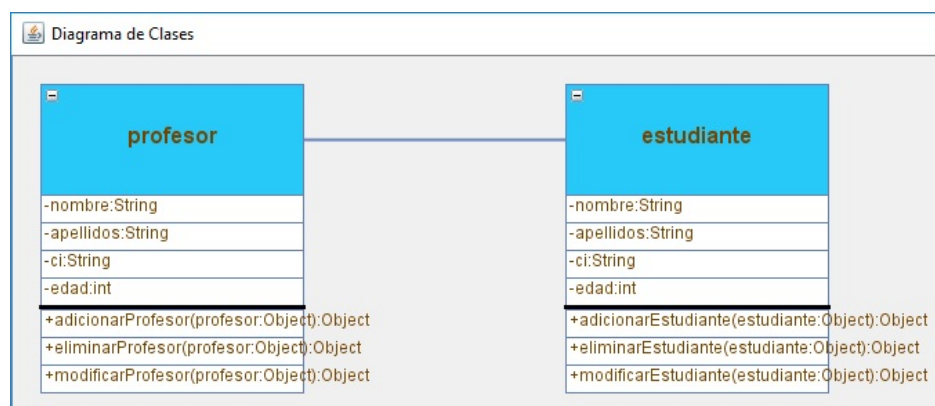


Figura 1: Diagrama de Clases del modelo PIM.

La herramienta incluye formas de tratamiento de errores y de validación de datos

4. Conclusiones

1. Se determinaron las preguntas-respuestas esenciales a tener en cuenta para el modelado de un Sistema de Gestión de Información deseado y en base al análisis crítico de la versión 1.0 se desarrolló la versión 2, que incluye una interfaz mejorada y la verdadera transformación de los elementos recogidos en los

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

II CONVENCION CIENTIFICA INTERNACIONAL “II CCI UCLV 2019”

**DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.**



diferentes formularios de la misma en los diagramas esenciales de UML en un modelo PIM: Casos de uso, Clases y Actividades.

2. Se desarrolló la implementación del módulo CIM-PIM en Java utilizando la biblioteca Mxgraph con el NetBeans IDE 8.0.1 y el JDK 8. Dicha biblioteca permite la salva de los diagramas obtenidos en un formato que puede ser recuperado en los subsiguientes módulos de la herramienta jMDA.
3. Se desarrolló un ejemplo de uso del módulo implementado que incluye los formularios de inserción de los elementos requeridos (funcionalidades, entidades, características de estos, pasos de acciones de las funcionalidades) y los diagramas correspondientes en el modelo PIM.

5. Referencias bibliográficas

- ANACLETO, V. A. 2006. MDA Reusabilidad Orientada al Negocio.
- BATISTA, A. R. 2015. Trabajo de Diploma: Herramienta CASE para Arquitectura Dirigida por Modelos jMDA. Módulo CIM-PIM. Versión 1.0.
- BERNARDO, J. & DUITAMA, J. F. 2011. Reflexiones acerca de la adopción de enfoques centrados en modelos en el desarrollo de software.
- BERNARDO QUINTERO, J. & ANAYA DE PÁEZ, R. 2007. Marco de Referencia para la Evaluación de Herramientas Basadas en MDA.
- BOOCH, G., BROWN A., RUMBAUGH, J. 2004. An MDA Manifesto. .
- CARRILLO, C. E. 2006. Arquitectura dirigida por modelos (MDA) y su aplicación en un caso de estudio.
- CONSUELO, M. 2010. MDA: Arquitectura dirigida por modelos.
- CUESTA M. ALBEIRO, L. T. M. Y. J. A. L. 2009. Comparativo de herramientas MDA (AndroMDA , ArcStyler, Op timalJ).
- FRANKY, M. C. 2010. MDA: Arquitectura Dirigida por Modelos.
- LORENZO, L. D. 2017. Trabajo de Diploma: Desarrollo del módulo PIM-PSM versión 4.0 de la herramienta jMDA.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

**II CONVENCION CIENTIFICA INTERNACIONAL
“II CCI UCLV 2019”**

**DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.**



- LUIS MUÑOZ, G. S. Y. D. H. H. 2015. Una visión a la propuesta de Arquitectura Dirigida por Modelos.
- MARTÍNEZ S., L. L., MORENO R. Y JACHO N. 2015. Análisis de la Transformación de Modelo CIM a PIM en el Marco de Desarrollo de la Arquitectura Dirigida por Modelos (MDA).
- MORA, B. 2006. Definición de lenguajes de modelados MDA vs DSL.
- OMG 2003. Especificaciones sobre MDA.
- PÁEZ, J. B. Q. Y. R. A. D. 2007. Marco de Referencia para la Evaluación de Herramientas Basadas en MDA.
- PEÑA, C. A. G. 2017. Líneas de Productos Software: Generando Código a Partir de Modelos y Patrones.
- PÉREZ ARMAYOR, D. 2014. *Technology combinations decision model for supply chains information systems integration*, Oldenburg, Germany, Shaker Verlag.
- PÉREZ ARMAYOR, D., ABREU FONG, P. A., HERNÁNDEZ LANTIGUA, D., LEÓN ALEN, E. O. & DÍAZ BATISTA, J. A. 2016. El empleo de Temix para evaluación de tecnologías de la información en planes de Sistemas de Información.
- PLANTE, F. 2006. Introducing the GMF Runtime.
- PONS, C. 2010. Desarrollo de software dirigido por modelos.
- QUINTERO, R. 2003. Aplicación de MDA al desarrollo de aplicaciones web en OOWS.
- RAFAEL CAMPS PARÉ, L. A. C. S., DOLORS COSTAL COSTA, MARC GIBERT GINESTÀ, CARME MARTÍN ESCOFET Y OSCAR PÉREZ MORA 2005. Bases de datos.
- SÁEZ, P. A. F. 2009. *Un Análisis Crítico Sobre la Aproximación Model Driven Architecture* Universidad Complutense de Madrid.
- TUML-OMG 2012. OMG Unified Modeling Language.
- VILLA, D. L. O. Y. J. A. S. 2012. Estudio comparativo de las herramientas case: staruml, poseidon for uml y enterprise architect, para el modelamiento de diagramas uml.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu