

Implementación del sistema de información  
SIGEPI usando herramientas de desarrollo dirigido  
por modelos (MDD)  
Implementación de la plataforma web para la gestión de los proyectos  
presentados por los grupos y semilleros de investigación en la  
Universidad de la Amazonia

Julian David Mora Ramos <sup>1</sup>

<sup>1</sup>Universidad de la Amazonia, <sup>1</sup>Florencia, Caquetá

15 de junio de 2017

## UNIVERSIDAD DE LA AMAZONIA

Ingeniería de Sistemas

### PROYECTO FIN DE PREGRADO

# IMPLEMENTACIÓN DEL SISTEMA INFORMACIÓN PARA GESTIÓN DOCUMENTAL SIGEPI

**Autor:** Julian David Mora Ramos

**Tutores:** Heriberto Fernando Vargas

Diana Carolina Chico

Diana María Espinosa

Curso académico 2011/2017

*DEDICATORIA*

A la Facultad de Ingeniería y a la Universidad, por la formación que me han dado.

Es gracias a ustedes que es posible el presente trabajo.

En verdad, gracias.

Yo.



# Agradecimientos

¡Muchas gracias a todos!

NOTA DE ACEPTACIÓN

---

---

---

---

---

---

---

FIRMA DEL PRESIDENTE DEL JURADO

---

FIRMA DEL JURADO

---

FIRMA DEL JURADO

**Resumen**

**Abstract**





# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Descripción general del proyecto</b>	<b>3</b>
2.1. Planteamiento del problema . . . . .	3
2.1.1. Contexto . . . . .	3
2.1.2. Formulación del problema . . . . .	3
2.2. Justificación . . . . .	4
2.3. Objetivos . . . . .	4
2.3.1. Objetivo general . . . . .	4
2.3.2. Objetivos específicos . . . . .	4
<b>3. Marco referencial</b>	<b>5</b>
3.1. Marco teórico . . . . .	5
3.1.1. Desarrollo Dirigido por Modelos. Conceptos . . . . .	5
3.1.2. Lenguaje de dominio específico . . . . .	5
3.1.3. Arquitectura de desarrollo . . . . .	5
<b>4. Procesos</b>	<b>7</b>
<b>5. Conclusiones y trabajos futuros</b>	<b>9</b>
5.1. Anexos . . . . .	12



## Índice de figuras

3.1. Proceso de transformación de código . . . . .	5
3.2. Flujo de abstracción DSL . . . . .	5



# Capítulo 1

## Introducción

Actualmente, adaptarse a las necesidades del cliente es un problema que aumenta con el transcurso del tiempo, las condiciones de negocio cambian y consigo también la construcción de sistemas. Consecutivamente, el mundo moderno exige gran demanda tecnológica y adaptarse a esta línea de constantes transformaciones resulta en una tarea compleja. Por esto, gracias a las investigaciones realizadas en este campo, se han logrado avances en ciertas áreas fundamentales que influyen en la puesta en marcha de proyectos.

Muchos proyectos de desarrollo de software poseen gran demanda, y más cuando están soportados por herramientas de automatización de tareas. Los tiempos de ejecución y elaboración de proyectos se reducen cuando herramientas DSL (Lenguaje de dominio específico) son implementadas, ofreciendo características de generación de código reutilizable y componentes.

El lenguaje de dominio específico no es una tecnología emergente, las primeras apariciones fueron a mediados de los 80s. Actualmente se pueden encontrar ramificaciones DSL en diferentes aplicaciones, cada una con funcionalidades específicas, tales como MDD (Desarrollo Dirigido por Modelos), MDA (Arquitectura Dirigida por Modelos) y MDE (Ingeniería Orientada a Modelos), siendo estos un conjunto de recursos interoperables que permiten ser usados en el análisis, modelado y construcción de proyectos software. Por otro lado, han aumentado la cantidad de mecanismos para agilizar la planeación y ejecución de proyectos, generalmente todos estos siguen la misma filosofía de las DSL, por ejemplo, se pueden encontrar las ORM (Mapeo de Objetos Relacionales) para el control de múltiples gestores de bases de datos a partir de un solo lenguaje base. Del mismo modo, los ingenieros, programadores de computadoras, matemáticos, estadísticos y demás, hacen uso de herramientas de lenguaje de dominio específico a partir de programas de computador como R (Lenguaje estadístico), interpretes de expresiones regulares, SQL (lenguaje de consulta estructurada), LINQ (Lenguaje de consulta integrada), entre otros. Como se puede observar, un dominio específico tiene un vocabulario especializado para describir las cosas que son particulares a ese dominio.

El sistema de gestión documental, SIGEPI, es una plataforma web robusta, creada gracias a la implementación de herramientas de modelado y generación de código, DSL. En la primera sección de este documento se nombran todos aquellos procesos que se siguieron para la obtención del código fuente a partir de los esquemas dados al aplicar MDD (Desarrollo Dirigido por Modelos). Posteriormente, se explica en detalle la arquitectura de desarrollo que fue aplicada, incluyendo cada una de las librerías y marcos de trabajo que fueron usados.



## Capítulo 2

### Descripción general del proyecto

#### 2.1 Planteamiento del problema

##### 2.1.1 Contexto

En la búsqueda de mejores procesos que ayuden en la optimización de la productividad en el desarrollo de software, nuevas metodologías y herramientas han emergido, consigo vienen diferentes maneras de aplicar la ingeniería en la realización de tareas complejas, algunas veces para tiempos relativamente cortos. Un ejemplo común es el modelado de requisitos, siguiendo el estándar UML se pueden obtener esquemas que permiten la visualización de cada proceso por separado desde diferentes puntos de vista; diagramas de casos de uso, clases, bloques, secuencia, componentes, etc.

Hoy en día, es común encontrar variedades herramientas de modelado UML que permiten la generación de código fuente, pero existe una limitación que se puede plantear sobre la misma ideología que mantiene este estándar, dado que se deben seguir estrictamente una serie de normas. El tiempo gastado en el modelado UML produce el mismo valor en código, pero todavía necesita agregar o editar manualmente la mayoría del código.

Existen casos en donde es necesario diseñar y crear un sistema de información siguiendo un conjunto de normas no dadas por un estándar existente, propiamente personalizadas y generalmente creadas desde cero a partir de una base abstracta obtenida desde los requisitos, es decir, crear una serie de reglas de modelado a partir de un grupo de requisitos. Aquí es donde entra en acción el diseño dirigido por modelos, conocido por sus siglas MDD, un paradigma de ingeniería de software que permite la manipulación de grandes cantidades de requerimientos [NGC16], todo esto siguiendo un conjunto de reglas de modelado de esquemas personalizados con base fundamental en los requisitos. De esta manera, cada proceso que se propone para la construcción del sistema va a seguir estrictamente ese conjunto de reglas individualizadas.

##### 2.1.2 Formulación del problema

Desarrollar un sistema de información suele ser llevado a cabo de la manera convencional, siguiendo una serie de pasos imprescindibles para todo proyecto. Consiste en realizar el modelado del sistema a partir de un conjunto de requerimientos. El paso a seguir trata de la adecuación de un marco de trabajo, algunas veces esto se lleva a cabo manualmente, un ejemplo claro es cuando se usa la arquitectura de tres capas MVC (Modelo, Vista y Controlador); se construye individualmente cada elemento según sea la capa correspondiente.

## 2.2 Justificación

El desarrollo dirigido por modelos (MDD) es una disciplina que ha ido emergiendo con gran fuerza en los últimos años [PGP10]. Ha impuesto una nueva forma de desarrollar y mantener sistemas software. Cuando se desarrolla un SI con los métodos convencionales, requiere de bastante tiempo dependiendo de la complejidad del mismo, incluyendo el tiempo de modelado de esquemas, tales como diagramas UML (Lenguaje de Modelado Unificado). Lo anterior implica que el progreso del sistema debe mantener en constante supervisión, con el fin de velar por el cumplimiento de los requisitos que fueron regidos en la planificación inicial.

El ciclo de vida en el desarrollo de un SI está dado por seis fases: Planificación y determinación de requisitos, análisis, diseño, desarrollo, pruebas e instalación en producción [PGP10]. Con el uso de un lenguaje de dominio específico se reduce el tiempo de desarrollo de un SI, debido a la eficacia en el avance de las etapas que suelen tomar más tiempo. El elemento "diseño" y "desarrollo" tienden a trabajar conjuntamente, donde el primero está ligado de forma rigurosa al segundo gracias a un lenguaje intermediario, este último conocido como lenguaje de dominio específico, o DSL.

Tal es el caso de la plataforma web SIGEPI (Sistema de información para la gestión de proyectos de investigación), donde se optó por la implementación de la herramienta DSL Tools, un entorno de desarrollo creado por la multinacional Microsoft, conocido como ".NET". Esta herramienta es una notación del lenguaje de dominio específico, ofrece un complejo Kit de desarrollo de software (SDK) que se integra totalmente con el entorno de desarrollo integrado, Visual Studio. Todo esto se adapta completamente con los lineamientos establecidos por Universidad de la Amazonia, ya que el sistema puede ser integrado simultáneamente con el sistema misional de la misma, Chairá.

## 2.3 Objetivos

### 2.3.1 Objetivo general

Implementar el sistema de información SIGEPI para la gestión de los procesos de presentación, evaluación y seguimiento de los proyectos de investigación en la Universidad de la Amazonia, mediante el tratamiento y aplicación de los resultados obtenidos en el uso de la metodología de desarrollo dirigida por modelos.

### 2.3.2 Objetivos específicos

- Preparar los esquemas modelados por el proceso de DSL y generar el código fuente a partir de los metamodelos diseñados.
- Aplicar el código fuente obtenido por el procesamiento de los meta-modelos en un entorno de desarrollo basado en la web, específicamente la arquitectura de tres capas de .NET MVC.
- Realizar pruebas de rendimiento, usabilidad y seguridad para verificar el cumplimiento de los requisitos no funcionales.



## Capítulo 3

### Marco referencial

#### 3.1 Marco teórico

##### 3.1.1 Desarrollo Dirigido por Modelos. Conceptos

El desarrollo dirigido por modelos (MDD) es un paradigma que resuelve inconvenientes relevantes en el desarrollo de software, la causa surgió desde los inicios de la década de los 60s, cuando se introdujo el concepto de "crisis del software", originado por la complejidad y el costo requerido por las necesidades del cliente [PGP10]. MDD enriquece el proceso de desarrollo a partir de herramientas especializadas, destaca una relación entre lo abstracto y el código fuente. La abstracción está representada por modelos, por tanto estos requieren de una serie de fases culminar la transformación hasta obtener el código fuente [ver figura 3.1].

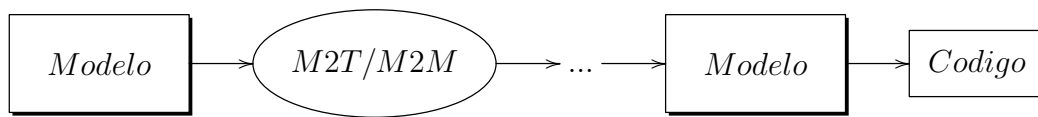


Figura 3.1: Proceso de transformación de código

##### 3.1.2 Lenguaje de dominio específico

El lenguaje de dominio específico (DSL) eleva el nivel de abstracción más allá que los lenguajes de programación para especificar una solución al problema usando conceptos de dominio [KT08]. Generalmente se le encuentra como una notación gráfica, cuyos modelos resultan en un conjunto de elementos y relaciones entre si.

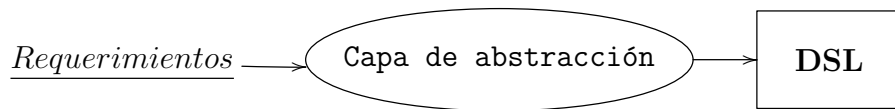


Figura 3.2: Flujo de abstracción DSL

##### 3.1.3 Arquitectura de desarrollo



## Capítulo 4

### Procesos



## Capítulo 5

### Conclusiones y trabajos futuros



## Bibliografía

- [KT08] Steven Kelly and Juha-Pekka Tolvanen. *Domain-specific modeling: enabling full code generation*. John Wiley & Sons, 2008.
- [NGC16] Guido NUÑEZ, Magalí GONZÁLEZ, and D Sc Luca CERNUZZI. Un enfoque mdd para el desarrollo de ria. 2016.
- [PGP10] Claudia Pons, Roxana Silvia Giandini, and Gabriela Pérez. Desarrollo de software dirigido por modelos. 2010.

## Anexos