

# 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

10 de mayo de 2017

TÍTULO DE LA TESIS

UNIVERSIDAD DE LA AMAZONIA



# **IMPLEMENTACIÓN DEL SISTEMA SIGEPI**

TRABAJO PRESENTADO COMO PARTICIPACIÓN RELEVANTE

GRUPO GIECOM

VICERRECTORÍA DE INVESTIGACIONES

PRESENTA

**JULIAN DAVID MORA RAMOS**



*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



# Índice general

|  |    |
|--|----|
| 0.1. INTRODUCCIÓN . . . . .                            | XI |
| 0.2. DESCRIPCIÓN GENERAL DEL PROYECTO . . . . .        | 1  |
| 0.2.1. Planteamiento del problema . . . . .            | 1  |
| 0.2.1.1. Contexto . . . . .                            | 1  |
| 0.2.1.2. Formulación del problema . . . . .            | 1  |
| 0.2.2. Justificación . . . . .                         | 1  |
| 0.2.3. Objetivos . . . . .                             | 2  |
| 0.2.3.1. Objetivo general . . . . .                    | 2  |
| 0.2.3.2. Objetivo específico . . . . .                 | 2  |
| 0.3. MARCO REFERENCIAL . . . . .                       | 3  |
| 0.3.1. Marco teórico . . . . .                         | 3  |
| 0.3.1.1. Lenguaje de dominio específico, DSL . . . . . | 3  |
| 0.3.1.2. DSL & Meta modelos . . . . .                  | 3  |
| 0.3.1.3. Arquitectura MVC . . . . .                    | 3  |
| 0.3.1.4. Framework de desarrollo Ext.NET . . . . .     | 3  |
| 0.4. CONCLUSIONES . . . . .                            | 4  |
| 0.5. TRABAJOS FUTUROS . . . . .                        | 5  |
| 0.6. Bibliografía . . . . .                            | 6  |
| 0.7. ANEXOS . . . . .                                  | 8  |



# Índice de figuras



# Índice de cuadros

**Resumen**

**Abstract**

## 0.1. INTRODUCCIÓN

Hoy en día, muchos proyectos de desarrollo de aplicaciones y sistemas informáticos poseen gran demanda, y más cuando están basados en 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, tales como componentes que pueden ser implementados en otros proyectos del mismo tipo. No se trata de una tecnología emergente, las primeras apariciones de los DSL fueron a mediados de los 80's.

Actualmente se pueden encontrar ramificaciones DSL en diferentes aplicaciones, cada una con funcionalidades específicas, tales como MDD, MDA y MDE, siendo estos un conjunto de recursos interoperables que permiten ser usados en el análisis, modelado y construcción de proyectos software. Por otro lado, las nuevas tecnologías han aumentado la cantidad de mecanismos y maneras de generar software, 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 programadores de software, matemáticos, estadísticos y demás, hacen uso de herramientas de lenguaje de dominio específico a partir de programas de computadora como R, interpretes de expresiones regulares, SQL (lenguaje de consulta estructurada), entre otros.

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.





## 0.2. DESCRIPCIÓN GENERAL DEL PROYECTO

### 0.2.1. Planteamiento del problema

#### 0.2.1.1. 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 mismo estándar, dado que se deben seguir estrictamente una serie de normas. 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 [2], 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.

#### 0.2.1.2. Formulación del problema

Actualmente, desarrollar un sistema de información suele ser llevado a cabo de la manera clásica, 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 (3) capas MVC (Modelo, Vista y Controlador); se construye individualmente cada elemento según sea la capa correspondiente.

### 0.2.2. Justificación

El diseño e implementación de sistemas informáticos generalmente requiere de bastante tiempo dependiendo de la complejidad del proyecto, todo esto sin incluir el tiempo que se requiere desde un inicio para modelar los esquemas necesarios, como los diagramas UML. Lo anterior implica que el desarrollo del sistema debe mantener en constante supervisión para velar por el cumplimiento de los requisitos que fueron regidos en la planificación inicial.

El ciclo de vida en el desarrollo de un sistema de información (SI) está dado por ocho etapas, a continuación se nombran en el orden de ejecución: Planifi-

cación, análisis, diseño, desarrollo, implementación, pruebas, instalación, y mantenimiento. Con el uso de herramientas DSL se reduce el tiempo de desarrollo de un SI, debido a la eficacia en el avance de las etapas que suelen tomar mas 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 ultimo 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.

### **0.2.3. Objetivos**

#### **0.2.3.1. Objetivo general**

Implementar el sistema de información 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 arquitectura de desarrollo dirigida por modelos, MDD.

#### **0.2.3.2. Objetivo específico**

- Procesar y generar resultados (código fuente) a partir de los meta-modelos definidos por los esquemas diseñados en la arquitectura MDD.
- 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 referentes al mismo.

### 0.3.1. Marco teórico

lalalalalalala

lalalalalalalalalalalalalalalalalalal

olaololaololaola olaololaololaolao laololaolao laololaola

[illegible]

## **0.4. CONCLUSIONES**

## **0.5. TRABAJOS FUTUROS**

## **0.6. Bibliografía**

# Bibliografía

- [1] José H Canós, Patricio Letelier, and M<sup>a</sup> Carmen Penadés. Metodologías ágiles en el desarrollo de software. *Metodologías Ágiles en el Desarrollo de Software*, 1(10):1–8, 2003.
- [2] Guido NUÑEZ, Magalí GONZÁLEZ, and D Sc Luca CERNUZZI. Un enfoque mdd para el desarrollo de ria. 2016.

## **0.7. ANEXOS**