

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 ¹

¹Universidad de la Amazonia, ¹Florencia, Caquetá

15 de abril 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	3
0.3.1.3. Arquitectura MVC	3
0.3.1.4. Ext.NET, motor de renderizado	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 de los desarrollos de aplicaciones y sistemas informáticos con funcionalidades para automatización de tareas poseen gran demanda. Los tiempos de ejecución y elaboración de un proyecto se reducen cuando herramientas DSL (Lenguaje de dominio específico) y similares 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 lanzadas 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 (Object Relational Mapping) 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 mediante programas de computadora como R, interpretes de expresiones regulares, SQL, entre otros.

El sistema de gestión documental (SIGEPI) es una plataforma web robusta, que fue creada gracias a la implementación de herramientas de modelado y generación de código, DSL. En la primera parte de este documento se nombran todos aquellos procesos que se siguieron para la generación del código fuente a partir de los esquemas obtenidos después de aplicar el desarrollo dirigido por modelos, también conocido como MDD (en inglés, Model-Driven Development). Posteriormente, se explica la arquitectura de desarrollo que fue aplicada, incluyendo cada una de las librerías y entornos 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 a la optimización y mejoramiento 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 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 [1], 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, el ciclo de vida en el desarrollo de un sistema de información suele ser

0.2.2. Justificación

El diseño e implementación de sistemas informáticos generalmente se efectúa siguiendo la manera tradicional, que consiste en tomar como base inicial uno o más entornos de desarrollo (conocido en el inglés como: Framework) y desde ahí se inicia la creación y adecuación de cada elemento que va a conformar el sistema, muchas veces esto se lleva a cabo manualmente, un ejemplo claro es cuando se usa la arquitectura MVC (Modelo, Vista y Controlador) de cualquier framework; se construyen individualmente los controladores, modelos y vistas. Esta operación 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: Planificación, 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 intermedio, 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. MARCO REFERENCIAL

0.3.1. Marco teórico

0.3.1.1. Lenguaje de dominio específico

0.3.1.2. Meta modelos

0.3.1.3. Arquitectura MVC

0.3.1.4. Motor de vistas Ext.NET

0.4. Conclusiones

0.5. TRABAJOS FUTUROS

0.6. Bibliografía

Bibliografía

- [1] Guido NUÑEZ, Magalí GONZÁLEZ, and D Sc Luca CERNUZZI. Un enfoque mdd para el desarrollo de ria. 2016.

0.7. ANEXOS