

# Incorporación de Tecnologías de Desarrollo Dirigido por Modelos en una PYME

Alejandro Cortiñas<sup>1</sup>, Miguel R. Luaces<sup>1</sup>, Oscar Pedreira<sup>1</sup>, Javier Vilares<sup>2</sup>

<sup>1</sup>Universidade da Coruña, Facultade de Informática, A Coruña, Spain  
{alejandro.cortinas,luaces,oscar.pedreira}@udc.es

<sup>2</sup>Enxenio S.L., A Coruña  
jvilares@enxenio.es

## 1. Introducción

Este artículo presenta de forma breve un proyecto en el que se incorporaron tecnologías de desarrollo dirigido por modelos en una PYME dedicada al desarrollo de software y administración de sistemas. La solución desarrollada permite, a partir de un modelo de datos definido en un diagrama de clases UML, generar una aplicación base funcional, ahorrando trabajo a los desarrolladores en etapas iniciales de los proyectos. Además de reducir tanto el tiempo de desarrollo como el coste de implementación, la generación de arquetipos comunes a toda la empresa facilita la aplicación de buenas prácticas, mejorando la calidad de los productos finales, y el proceso de adaptación del personal a nuevos proyectos.

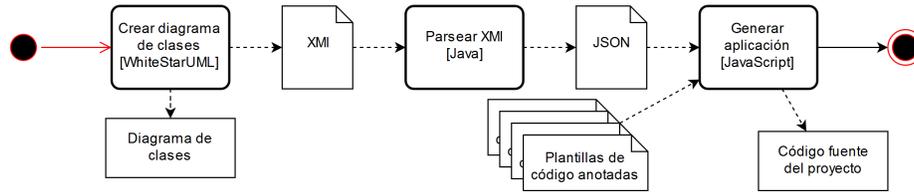
Enxenio<sup>1</sup> desarrolla aplicaciones web a medida, principalmente de gestión o en el ámbito del comercio electrónico. Anteriormente a la ejecución de este proyecto, Enxenio cuenta con varios arquetipos usados en función del dominio general del nuevo proyecto. Cada arquetipo va asociado a una serie de componentes que se pueden incluir utilizando un gestor de dependencias. Enxenio ha colaborado con el Laboratorio de Bases de Datos de la Universidad de Coruña en el diseño de una Línea Producto Software en el dominio de los Sistemas de Información Geográfica, siendo el primer contacto de la empresa con técnicas de generación de código actuales. Después de dicha colaboración, la empresa decide integrar la generación de código en su flujo de trabajo, de forma que el esqueleto base a utilizar en todos sus proyectos sea generado a partir de una especificación, unificando su conjunto de arquetipos. Además de proveer los componentes básicos necesarios para un nuevo proyecto, todo el código repetitivo dependiente del modelo de datos se debe generar automáticamente.

## 2. Solución desarrollada

En los proyectos de Enxenio se utiliza la herramienta de código abierto WhiteStarUML<sup>2</sup> para definir el diagrama de clases UML que representa el modelo de datos de la aplicación a desarrollar. Para no alterar el flujo habitual de la empresa, se ha decidido utilizar el archivo XMI que se puede exportar de WhiteStarUML como especificación de entrada en la herramienta generadora.

<sup>1</sup> <https://www.enxenio.es/>

<sup>2</sup> <http://whitestaruml.sourceforge.net/>



**Figura 1.** Flujo para la generación de un proyecto.

Por otro lado, se usará un motor de generación de código basado en *Scaffolding*<sup>3</sup>. Este motor recibe un JSON con la especificación del producto siguiendo un esquema preestablecido, que permita definir completamente la variabilidad de los productos a generar, así como plantillas de código anotadas, y genera código fuente final. La conversión del fichero XMI a la especificación en formato JSON se hace mediante una pequeña utilidad en Java. Se puede ver el proceso en la Figura 1. En ambos casos, los dos componentes de los que se compone la solución desarrollada son invocados desde el terminal.

A partir de un producto reciente de la empresa se ha diseñado una arquitectura base genérica con componentes nucleares que son necesarios en la mayor parte de los proyectos. En una primera versión sólo se ha implementado la generación de código a partir del modelo de datos en la parte servidor. Partiendo de dicho proyecto, el diseño del esquema de la especificación, la implementación del código anotado y de los *scripts* necesarios requirió 400 horas, 10 semanas a tiempo completo de una trabajadora. La persona que llevó a cabo el trabajo fue supervisada por un jefe de proyecto. Ninguno de los dos tenía conocimientos de MDD.

El resultado se validó precisamente generando la parte servidor del proyecto existente en el que se basó la arquitectura. Es decir, se consiguió reutilizar el código del cliente web existente conectándolo al API REST generado automáticamente a partir del diagrama de clases del proyecto. Desde que se terminó la primera versión del nuevo arquetipo generador, se ha utilizado en tres proyectos relativamente pequeños. Todos los proyectos comparten la mayor parte de su código fuente, aproximadamente 14.000 líneas de código muy levemente modificadas en función de la especificación, mientras que las líneas de código generadas específicamente a partir del modelo de datos se muestran en la Tabla 1. Las líneas de código están medidas sin tener en cuenta comentarios ni líneas en blanco. En versiones posteriores se espera incluir la interfaz web en el arquetipo, así como gestionar la evolución de la plataforma apoyándose en sistemas de control de versiones de código fuente.

<sup>3</sup> <https://github.com/AlexCortinas/spl-js-engine>

**Tabla 1.** Número de entidades y relaciones en los esqueletos generados.

Producto	# entidades	# propiedades	# relaciones	# LOC generadas
Proyecto base utilizado	18	139	23	3 212
<i>Dashboard</i>	8	37	10	1 151
Planificador	7	35	9	1 051
<i>Galicons</i>	12	50	11	2 111