estudio exploratorio sobre desvios en el modelado de software orientado a objetos en PROYECTOS FINALES

Luciana Gabriela Terreni1

1Instituto Sedes Sapientiae, Universidad Autónoma de Entre Ríos (Argentina).

Resumen

En el siguiente trabajo se presenta un estudio exploratorio sobre los desvíos detectados en los modelos de diseño de software orientados a objetos presentados como proyecto final por estudiantes de segundo de la tecnicatura en Análisis y Desarrollo de Software del Instituto Sedes Sapientiae. Siguiendo un enfoque cuali-cuantitativo se analizaron proyectos entregados en el periodo 2017-2024 detectándose desvíos generales en el nivel de abstracción, notaciones aplicadas y relaciones no válidas entre elementos del diseño y desvíos específicos correspondientes a los diagramas de clase, casos de uso y secuencia, especificados con Lenguaje de Modelado Unificado (UML). Los resultados obtenidos permiten reorientar estrategias y metodologías de enseñanza y aprendizaje de futuras cohortes con el fin de fortalecer las competencias de diseño y modelado del futuro egresado.

**Palabras clave:** Modelado de software. Orientación a objeto. Proyectos de software.

Abstract

This paper presents an exploratory study of the deviations detected in object-oriented software design models submitted as final projects by second-year students of the Software Analysis and Development program at the Sedes Sapientiae Institute. Using a mixed-methods approach (qualitative and quantitative), projects submitted between 2017 and 2024 were analyzed. General deviations were identified in the level of abstraction, the notations used, and invalid relationships between design elements. Specific deviations were also found in class diagrams, use cases, and sequence diagrams, specified using the Unified Modeling Language (UML). The results obtained allow for the reorientation of teaching and learning strategies and methodologies for future cohorts, with the aim of strengthening the design and modeling skills of future graduates.

**Keywords:** Software modeling. Object orientation. Software projects.

# INTRODUCción

Las prácticas profesionalizantes son actividades de formación y de aproximación progresiva al campo ocupacional dentro la propuesta curricular de las carreras técnicas, que pueden realizarse dentro o fuera de la institución, y que tienen como objetivo que los estudiantes construyan capacidades y conocimientos profesionales [1].

Las prácticas utilizan, en ocasiones, el proyecto como dispositivo articulador de contenidos y habilidades. En ellos, los estudiantes transfieren y ponen en juego conocimientos con el propósito de resolver un problema o generar una oportunidad de mejora desde su especialidad y consolidar, integrar y ampliar las competencias que se corresponden con el perfil [2]. Para ello es necesario el despliegue de estrategias didácticas que articulen capacidades prestando especial atención a la contextualización del proyecto [3].

En el Instituto Sedes Sapientiae, el espacio Pràctica Profesionalizante II ha optado por trabajar con entidades y organizaciones externas formulando proyectos de enfoque sistémico para problemas detectados en cada una y desarrollando el análisis, diseño y modelado y desarrollo de la solución informática. Algunas actividades se realizan en las instituciones como el proceso de ingeniería de requerimientos y otras se realizan en el aula con la tutoría del profesor como el modelado, desarrollo y documentación del proyecto.

A partir de la observación de errores recurrentes en los proyectos en la fase de modelado orientado a objetos, se procedió a realizar un análisis de los mismos, en proyectos producidos entre 2017 y 2024, buscando la mejora en los procesos de enseñanza de los futuros técnicos.

Los objetivos planteados para este estudio son:

1. Recopilar los proyectos de modelado orientado a objetos formulados en el espacio Práctica Profesionalizante II
2. Encontrar desvíos de los estándares del modelado orientado a objeto con UML en los proyectos recopilados
3. Describir los principales desvíos de modelado encontrados

Es importante remarcar que este trabajo toma como objeto de estudio la fase de modelado de estos proyectos.

## Producción de un modelo orientado a objetos como entregable evaluable

Las técnicas de modelado mejoran la productividad de los equipos de desarrollo pues propician la oportunidad de generar código automáticamente, reducen los defectos en el código al aplicarse sobre ellos herramientas de verificación, testeo y validación, permiten explorar alternativas sin costos de codificación, facilitan el reúso y el mantenimiento y documentan formalmente el sistema. Todas estas funciones son las que hacen del modelado un conocimiento especifico esencial en las carreras técnicas vinculadas a la informática.

Partiendo de estas ideas, en Práctica Profesionalizante II se les propone a los estudiantes seleccionar un caso de estudio que posea una problemática u oportunidad de mejora abordable desde un proyecto de software, el cual será formulado siguiendo el ciclo de vida de desarrollo de sistemas. Para la etapa de diseño se plantea que a partir de los requerimientos especificados realicen un modelo orientado a objetos con UML, siguiendo el esquema de vistas que se presenta a continuación:

Figura 1. Etapas previstas en el modelado orientado a objetos.

Este diseño o modelo se compone de la siguiente documentación:

* **Documento formal del proyecto:** contiene descripción general del caso, justificaciones, motivaciones, objetivos, etc.
* **Documento de requerimientos:** contiene los requisitos según el estándar IEEE 830 [4].
* **Documento de modelado orientado a objetos:** contiene los diagramas en UML y especificaciones asociadas.

Estos documentos formales o entregables son evaluados durante su desarrollo y en su versión final por docentes y expertos externos en la temática que realizan devoluciones documentadas que posibilitan la mejora y el avance en las siguientes fases del proyecto. También se realizan instancias áulicas de evaluación por pares.

## Pautas para la documentación del modelo en UML

El Lenguaje de Modelado Unificado (UML) [5] es un lenguaje gráfico para visualizar, especificar, producir y documentar un sistema. Básicamente permite describir los planos de un sistema desde diferentes vistas, las cuales son aportadas por los diagramas que constituyen el lenguaje.

Las pautas planteadas a los estudiantes para la documentación del modelo orientado a objeto consistieron en realizar los diagramas de UML que representen las diferentes vistas del sistema en un documento formal utilizando Lucichart como herramienta CASE.

Las recomendaciones generales para la tarea de modelado orientado a objeto consistieron en identificar niveles aceptables de abstracción, aplicar en forma consistente las notaciones y esquematizar los diagramas que sean relevantes al caso de estudio, pero estableciendo como obligatorios el diagrama de clase, diagrama de casos de uso y diagrama de secuencia.

# METODOLOGÍA

El enfoque del estudio es exploratorio, ya que tiene carácter limitado y preliminar. El diseño es cuali-cuantitativo, ya que el estudio se basa en técnicas de observación y análisis de proyectos en su fase de modelado y en la medición de variables para la búsqueda de patrones al procesar los datos numéricos relevados [6].

La muestra seleccionada es pequeña y representativa y se constituye de 45 proyectos finales de la cátedra Practica Profesionalizante II presentados 2017 y 2024. Los proyectos analizados se corresponden a una entrega pre-eliminar anterior a la entrega final que fue aprobada y específicamente se tomó la sección de modelado como objeto de estudio. A cada uno de ellos se le aplico un instrumento de análisis presentado en la Tabla 1. Las categorías de la tabla fueron establecidas por el docente de la cátedra según revisión bibliográfica y experiencia previa.

Tabla 1. Instrumento de análisis de proyectos.

|  |  |  |
| --- | --- | --- |
| **Desvíos detectados en el modelado UML** | **SE ENCUENTRA** | **NO SE ENCUENTRA** |
| Utilizar un nivel de abstracción incorrecto |  |  |
| Usar una notación inconsistente |  |  |
| Presentar relaciones no válidas |  |  |
| Presentar elementos faltantes o redundantes |  |  |
| Presentar elementos superpuestos o conflictivos |  |  |
| Utilizar demasiados o muy pocos diagramas |  |  |
| **Desvíos detectados en diagramas de casos de uso** | **SE ENCUENTRA** | **NO SE ENCUENTRA** |
| Definición del alcance |  |  |
| Uso inadecuado de Include y Extend |  |  |
| Incorporación de almacenes o archivos |  |  |
| Búsqueda de analogías con diagramas de flujo |  |  |
| Interacción entre actores |  |  |
| **Desvíos detectados en diagramas de clases** | **SE ENCUENTRA** | **NO SE ENCUENTRA** |
| Ausencia de clases relevantes |  |  |
| Excesivo uso de adornos (notas) |  |  |
| Uso inadecuado de herencias |  |  |
| Inclusión de actores en el diagrama |  |  |
| Presencia de tablas multivariadas |  |  |
| **Desvíos detectados en diagramas de secuencia** | **SE ENCUENTRA** | **NO SE ENCUENTRA** |
| Inclusión de actores en lugar de objetos |  |  |
| Inclusión de Archivos/Almacenes al final del diagrama |  |  |
| Uso del diagrama para casos de poca relevancia |  |  |
| Secuencias de mensajes confusas o mal establecidas |  |  |

Esta lista de chequeo se aplicó a cada uno de los proyectos y luego se contabilizaron los totales por categoría en una tabla de frecuencia construida en Excel.

# RESULTADOS

En el proceso de recopilación de proyectos pudieron obtenerse 51 proyectos, pero solo 45 cumplieron con los criterios de selección muestral. La distribución de estos proyectos entre los años 2017 y 2024 se evidencia en la Figura 2.

Figura 2. Proyectos de Modelado OO en el periodo 2017-2024

Del procesamiento de los datos relevados surgen los resultados que se muestran en la Figura 3. En este gráfico se evidencia la cantidad de proyectos que presentan un desvío determinado y puede observarse que son divisibles en desvíos relacionados a los niveles de abstracción y por otro lado a desvíos en las notaciones.

Figura 3. Desvíos generales detectados en el modelado con UML.

Al profundizar en el análisis y más específicamente en los diagramas que se habían establecido como obligatorios se pudieron obtener evidencias de los desvíos en cada uno de ellos, tal como se muestra en las Figuras 4, 5 y 6.

Figura 4. Desvíos detectados en diagrama de casos de uso.

Como puede apreciarse los desvíos en los diagramas de caso de uso se presentan principalmente en el uso inadecuado de las relaciones de "include" y "extend" y en la definición el alcance.

En el modelo con UML, "include" y "extend" son relaciones que permiten modelar funcionalidades en los casos de uso [7]. La relación de "extend" indica que un caso de uso agrega funcionalidad opcional a otro mientras que la relación de "Include" implica que un caso de uso es parte integral de otro y es necesario para su ejecución. Se observó en los proyectos que estas relaciones eran aplicadas confundiéndose una con otra y con errores de notación en el sentido de las fechas en el diagrama.

En cuanto a los desvíos en la definición de alcances, los diagramas incluían casos de uso con escasa asociación entre sí.

Figura 5. Desvíos detectados en diagrama de clases.

Los principales desvíos en los diagramas de clases refieren a uso en exceso o erróneo de las relaciones de herencia, en la ausencia de clases relevantes que se incorporaban como atributo multivariado en otra clase y en el uso del adorno nota en exceso en los diagramas para realizar aclaraciones o enlazar documentos explicativos.

Figura 6. Desvíos detectados en diagrama de secuencia.

El diagrama de secuencia [8] es uno de los que reviste mayor complejidad en el modelado con UML. Una de las dificultades que se observaron en los proyectos es la relacionada a en qué casos aplicar este diagrama. Se encontró evidencia de diagramas de secuencia para casos poco relevantes.

Por otro lado, también se hallaron casos de inclusión de actores en la operatoria de pasaje de mensajes entre objetos y en la aparición de archivos o almacenes en la sección de observaciones del diagrama de secuencia, en la columna final derecha.

En resumen, en los desvíos en diagramas específicos pueden observarse que los mismos refieren a uso de elementos no correspondientes al diagrama en cuestión, mal uso de los elementos para explicitar relaciones y deficientes delimitaciones del alcance del diagrama.

# CONCLUSIONes

En el caso de la Tecnicatura en Análisis y Desarrollo de Software del IPSS se plantea el modelado como una competencia a construir previamente al cursado de los espacios de desarrollo o programación, ya que la concepción previa y la anticipación permiten explicitar modelos antes del desarrollo de software que disminuyen errores y problemas. Los lenguajes de modelado, como UML, permiten formalizar sin ambigüedades con un lenguaje único y comprensible para todos los integrantes de un proyecto.

Considerando la importancia de la construcción de estas habilidades para el futuro profesional se plantea este estudio que ahonda en los desvíos hallados en los proyectos a los fines de mejorar los procesos de enseñanza y aprendizaje. Específicamente, en lo que concierne a este trabajo exploratorio, es importante mencionar que se han cumplido los objetivos planteados en cuanto a la recopilación de proyectos y el análisis de los desvíos encontrados. Por un lado, se analizaron desvíos relacionados con el propio proceso de modelado que incluyeron dificultades de abstracción en el caso de estudio y de especificación con notaciones de UML y por otro lado se analizaron desvíos propios de las especificaciones con diagramas en el mismo lenguaje, más precisamente en los diagramas de casos de uso, secuencia y clases. La detección de estos desvíos permite delinear estrategias tendientes a mitigarlos en futuras cohortes del espacio curricular.

En otro orden, vale mencionar que este estudio exploratorio revela líneas de investigación futuras como las dificultades en el proceso de abstracción ante situaciones problemáticas y aquellas relacionadas a la apropiación de nuevos lenguajes y notaciones de modelado de software.

Para finalizar, desde las prácticas docentes es interesante mencionar dos recomendaciones a partir de los resultados obtenidos. La primera es el trabajo en talleres sobre el modelado para casos de estudio estratégicamente seleccionados y la segunda es la introducción de casos paradigmáticos o problemáticos sobre los desvíos de especificación en UML encontrados. Esto permitirá entrenar o fortalecer la toma de decisiones y aplicación criteriosa de notaciones a la hora de modelar software.

referencias

1. Res. 5083 CGE. Aprobación del Diseño Curricular de la carrera “Tecnicatura Superior en Análisis

Desarrollo de Software”.

1. Ley de Educación Técnico Profesional 26.058
2. Documento de Lineamientos y criterios con relación a las PP en la ETP – INET 2017.
3. D. Rojas Suabita, “Aplicación de la norma IEEE 830 para la Especificación de Requisitos de Software (ERS) del Sistema Integrado de Información de Prestaciones Sociales (SIIPS) para la Caja de Retiro de las Fuerzas Militares (CREMIL) - participación como ingeniero de requerimientos”, Universidad de los Llanos, 2016.
4. J. Rumbaugh, "Modelado y diseño orientados a objetos: Metodología OMT", Prentice Hall, España, 1996.
5. R. Hernández Sampieri, C. F Collado, P. Baptista Lucio, “Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta” (2ª ed.). McGraw-Hill, 2023.
6. C. Jiménez, “Conoce todo sobre UML: Aplicaciones en Java y C++”, Ra-Ma, 2021.
7. G. Booch, J. Rumbaugh, I. Jacobson, “El Lenguaje Unificado de Modelado (UML 2.0): Guía de usuario”, Addison Wesley / Pearson , 2006.