



Documento de Análisis

¿Qué cosas salieron bien y qué cosas salieron mal?

En el proyecto 1 nos enfrentamos por primera vez a actividades de este tipo y al desarrollo de proyectos utilizando el paradigma de programación y diseño orientado a objetos. Al principio, nos resultó difícil proyectar nuestras ideas y no se logró una buena abstracción completa del dominio. Sin embargo, gracias a lo que aprendimos en clase y a la retroalimentación del profesor y los monitores, pudimos realizar un buen análisis. Y eventualmente, tanto el diagrama de casos de uso como el diagrama simplificado de clases no requirieron muchas modificaciones en los proyectos siguientes.

En el desarrollo del proyecto 2, nos enfrentamos a dificultades adicionales en la construcción de las interfaces de usuario. Al ser nuestra primera vez abordando esta tarea, nos resultó difícil y no logramos obtener resultados óptimos en términos de eficiencia. A pesar de estas dificultades, pudimos implementar la aplicación de acuerdo con el diseño inicial y cumplir con todos los requisitos establecidos. Para abordar de manera efectiva las responsabilidades de cada funcionalidad, distribuimos tareas entre los miembros del grupo. Aunque encontramos obstáculos, logramos completar el proyecto de manera satisfactoria.

En el último proyecto, ya teníamos conocimiento de la mayoría de los contenidos del curso, lo que nos permitió aplicar las modificaciones necesarias en la implementación previa. Para nuestra sorpresa, estas modificaciones no afectaron todo el código, ya que habíamos mantenido un bajo acoplamiento en todas las implementaciones anteriores.

Por otro lado, al implementar las pasarelas de pago, pudimos diseñar una estructura eficiente y bien organizada siguiendo los principios de diseño. La implementación de la aplicación no representó un gran problema gracias al patrón Modelo-Vista-Controlador y la estructura que habíamos establecido en nuestro código.

¿Qué decisiones resultaron acertadas y qué decisiones fueron problemáticas?

Nos sorprendió gratamente descubrir que muchas de las decisiones que tomamos al crear las clases y establecer sus relaciones fueron acertadas, a pesar de ser nuestra primera vez abordando este ejercicio. A pesar de los cambios realizados en nuestra aplicación, las relaciones entre las clases y sus responsabilidades se mantuvieron intactas. Estas decisiones demostraron ser acertadas, lo cual fue afortunado, ya que la modificación de nuestra aplicación para cumplir con el patrón Modelo-Vista-Controlador resultó sencilla gracias a esas decisiones previas.

Sin embargo, las decisiones relacionadas con las interfaces de usuario fueron cuestionables. Aunque teníamos conocimiento de patrones como el observador o listener, nuestra falta de experiencia en el diseño de interfaces de usuario dificultó la estructura de nuestro código. Se repitió mucho código y se invirtió demasiado tiempo en escribir líneas que podrían haber sido métodos en una clase base que nos hubiera ahorrado ese trabajo. Al menos, el diseño de las interfaces fue intuitivo y estético.

¿Qué tipo de problemas tuvieron durante el desarrollo de los proyectos y a qué se debieron?

Problemas debido al desconocimiento de la tecnología	Problemas realizando el análisis del dominio	Dificultad de diseñar en un entorno incierto
<ul style="list-style-type: none">• Dificultad para comprender la sintaxis y convenciones de nomenclatura en Java. Esto se debió a que apenas teníamos experiencia previa en	<ul style="list-style-type: none">• Dificultad para proyectar ideas y abstraer completamente el dominio. Dado que se tenía poca experiencia previa en el desarrollo de	<ul style="list-style-type: none">• Dificultad para establecer requisitos claros y definidos. Dado que había falta de claridad en los objetivos y requisitos del proyecto, esto pudo deberse a la

programación en Java y falta de estudio adecuado de la documentación y recursos disponibles.

- Complicaciones para entender los componentes y la disposición en las interfaces gráficas utilizando Swing. Esto dado que teníamos poca familiaridad con la biblioteca Swing y falta de práctica en la creación de interfaces gráficas en Java.
- Estructura deficiente del código y falta de cohesión. Dado que a la par que aprendíamos los principios de diseño en clase, debíamos de realizar las entregas de los proyectos.

proyectos con programación orientada a objetos y diseño de software, lo que dificultó la comprensión de cómo aplicar estos conceptos en la práctica.

- Dificultad para identificar las responsabilidades y relaciones entre las entidades del dominio. Porque la identificación precisa de las entidades, atributos y relaciones relevantes para el sistema en desarrollo era confusa y poco desglosada en el enunciado. Esto puede deberse a una falta de conocimiento específico del dominio del problema.

naturaleza del enunciado o a la poco conocimiento del negocio.

- Riesgo de cambios frecuentes en los requisitos y especificaciones. Pues sabíamos que los requisitos y especificaciones cambiarían a medida que se adquiere mayor conocimiento y se obtiene retroalimentación.
- Falta de definición clara de los roles y responsabilidades. Esto se pudo deber a que habían confusiones en cuanto a quién es responsable de qué aspecto del diseño.