

# Documento de reflexión

## ISIS-1226

David Valderrama Herrera  
Martín Daniel Rincón Molina



En el presente documento expondremos nuestra experiencia con el trabajo realizado a lo largo del semestre 2021-2 en los proyectos de clase. Así pues, si entendemos este proceso como uno dividido en tres etapas, podemos decir que, en general y a pesar de los problemas que tuvimos, estamos satisfechos con el diseño planteado desde la primera fase, porque nos permitió crear una implementación escalable a lo largo de los subsecuentes ciclos de trabajo, y consolidar un diseño que cumple con las buenas prácticas de la programación orientada a objetos. Estos dos puntos anteriormente mencionados contribuyeron a que, en aquellas dos ocasiones en las que se nos pidió implementar funcionalidades extra, la tarea se lograra sin tener que modificar drásticamente el código escrito antes, pues solo hubo necesidad de extender algunas clases y crear otras nuevas. A continuación ahondaremos en los aciertos y las áreas de oportunidad de los dos grandes bloques de nuestro proyecto: POS e inventario.

En cuanto al sistema POS, en la segunda parte del proyecto solo hubo que diseñar la interfaz y conectarle la lógica del sistema, lo cual demuestra que el diseño fue en general bueno. Un error rescatable del primer diseño, y por el cual no se implementó el requerimiento del histórico de compras, es porque en el primer diseño no se consideró que se fuese a necesitar el dato de las compras realizadas, de modo que esa información no se registraba en el sistema. Adicional a las modificaciones que había que hacer al proyecto anterior, no se encontró una solución al requerimiento del histórico que fuese lo suficientemente sencilla como para que valiese la pena implementarla.

En cuanto al sistema inventario como acierto vale mencionar que, al igual que con sistema POS, diseñar e implementar una GUI para el segundo proyecto fue bastante fácil; y como error, resalta la no utilización de la clase `Date` de la librería `java.util` en lugar de nuestra clase `Fecha` desde el comienzo, no obstante, nunca se substituyó completamente y, en su lugar, solo se integraron algunos métodos de `Date` para cumplir con el requerimiento funcional de poder eliminar productos vencidos.

Nuestro diseño fue el resultado de la corrección de múltiples errores, pero finalmente conseguimos una solución que goza de alta cohesión entre clases y bajo acoplamiento, y cumple con buenas prácticas. En las revisiones nunca fue extraño que, durante la implementación de las dos últimas partes del , nos diésemos cuenta de que habían métodos con más de una responsabilidad, lo que dificultaba la reutilización de estos métodos en los requerimientos nuevos. Estas situaciones nos hacían comprender la importancia de la regla de no asignar más de una responsabilidad a un elemento en particular. Este hecho se tradujo en la buena designación de estereotipos para las clases de nuestro diseño que garantizan que cada clase

se dedique a una cosa. A manera de ejemplo, queremos destacar que la inclusión de una clase Entrada fue una buena decisión de diseño, pues fue muy útil a la hora de implementar los nuevos requerimientos que se pedían en los proyectos 1 y 2. Además, varias de nuestras clases están construidas sobre el supuesto de una sola instancia: los cargadores, controladores, y coordinadores. La implementación de estas clases se hizo en el proyecto 1, cuando aún no teníamos conocimiento de patrones. Consideramos que habría sido una buena idea implementar Singletons sobre estas clases, lo cual nos hubiese ahorrado bastante tiempo de solución de errores provocados por descoordinación de instancias que no deberían haberse creado.

En cuanto al lenguaje y las tecnologías usadas, si bien al inicio del proyecto 1 representaban un reto adicional, con el tiempo nos pudimos adaptar usar de manera fluida tanto Java como Eclipse y GitHub. Aprender a usar nuevas librerías y frameworks fue todo un reto que sin duda nos ayudará a ser mejores ingenieros de sistemas, capaces de aprender nuevas tecnologías y afrontar nuevos desafíos.

En suma, pensamos que todo el trabajo hecho en los proyectos fue muy formativo. Si bien hubo varios momentos en los que nos sentimos frustrados con los retos que nacían a medida que crecía nuestro diseño y por los errores que descubríamos con cada revisión y retroalimentación de los monitores del curso, esto nos hizo entender la trascendencia de tomar buenas decisiones de diseño desde el principio y a seguir las recomendaciones que provistas durante las clases sobre cómo diseñar buen software, el cual es un objetivo del curso.