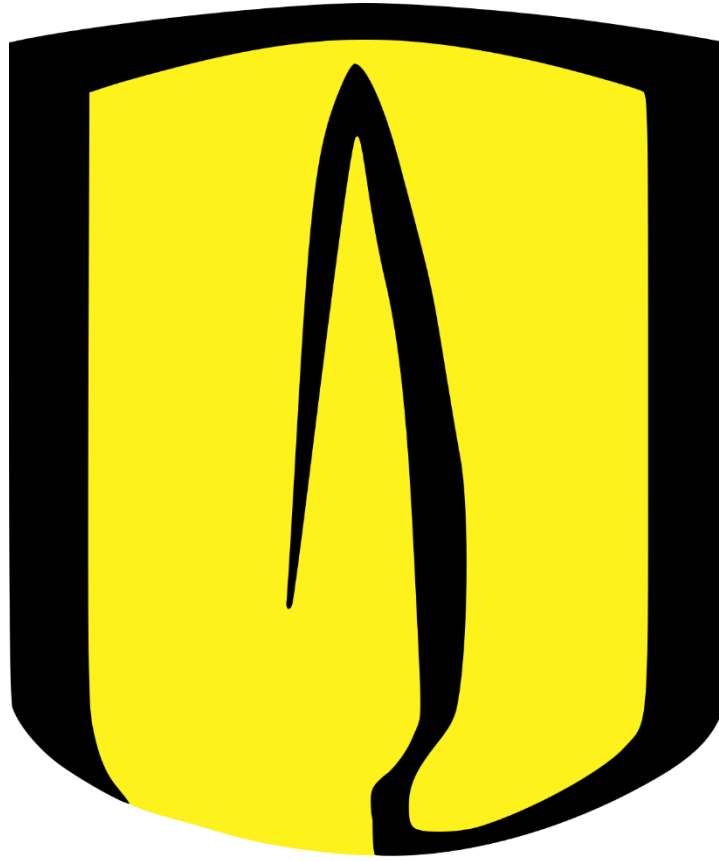


*Entrega #2 – Proyecto #3*

*Reflexión*



Universidad de los Andes

*Documento realizado por:*

Jose David Florez Ruiz – 202121507

Daniel Gustavo Clavijo Gutiérrez - 202122209

*Fecha:*

12 / 09 / 2022

Durante el semestre se fueron adquiriendo conocimientos sobre el área de programación orientada a objetos y diseño de software, así como el lenguaje de Java. El proceso en general no fue fácil ya que el aprendizaje se adquirió en aquellos momentos difíciles donde se encontraban errores que forzaban reconstruir diseños mal planteados dando a entender que para que un proyecto sea exitoso se debe tener la mejor planeación posible (diseño), esto se explica al tener en cuenta que con un buen diseño se logra estructurar la solución al problema de una forma fácil de implementar permitiendo evitar errores, bugs y a su vez, que el código sea extensible y mantenible. Una afirmación para resaltar sobre el proceso de diseño es que tener un buen diseño de la solución ayuda a planificar expansiones futuras. También es de resaltar la importancia de los patrones de diseño en la estructuración del código debido a que dan las bases para las soluciones de los problemas.

Fue durante el desarrollo de los distintos talleres y proyectos del curso que se descubrieron las grandes falencias que se generan por no aplicar los conceptos vistos en clase, un ejemplo de ello es tener un código con características como una cohesión baja y/o un acoplamiento muy alto, estas características conducen a que el código sea muy difícil de probar y que existan clases que solucionen una gran variedad de problemas y no uno en específico. La generación de bugs en clases donde no son bugs propios (son dados por otras clases), fueron de los problemas más difíciles de resolver. Esto también desencadenaba que lograr comprender cómo funcionaba una clase fuera complicado puesto que requería de métodos de distintas clases para funcionar. Adicionalmente, cabe mencionar el lado positivo de aquel proceso realizado durante el curso donde el controlador facilitó la comunicación de la lógica y el interfaz. También cabe destacar la existencia de la persistencia porque facilitó en gran medida la manera de consultar, escribir, modificar y eliminar archivos.

Comenzando con el proyecto 1, la mayoría de los problemas encontrados se deben a el desconocimiento de como diseñar apropiadamente una solución como también del uso del lenguaje, todo esto condujo a que existieran cosas importantes no tenidas en cuenta en el diseño inicial. Otro aspecto que tener en cuenta es que el haber venido de un curso (Estructura de datos y algoritmos), en el que se maneja una estructura Modelo – Vista – Controlador y programación estructural, se evidenció cierto caos al querer entrar en un ambiente completamente distinto donde se manejan e implementan clases en vez de funciones, por lo que es necesario adaptarse rápido al nuevo ambiente.

En la realización del proyecto 2 se generaron problemáticas por un mal diseño debido a que la información de los equipos por fechas no se guardaba como se deseaba. Otra gran dificultad encontrada se ubica en la implementación de las interfaces y de otros frameworks, lo que condujo a que el diseño del proyecto fuera plano y poco amigable con el usuario final.

Finalizando, en cuanto al proyecto 3, la extracción de la información para crear las gráficas fue de las partes más difíciles de implementar, puesto que en el diseño inicial no se tenía en cuenta almacenar aquellos datos necesarios de extraer, causando el forzamiento de cambios en clases que, a su vez, desencadenaba en cambios en otras clases. Además, se requirió cambiar la estructura del proyecto con el fin de tener más de un equipo por usuario, debido a que se encontraron requerimientos obsoletos por los nuevos cambios exigidos. Esto implicó solucionar nuevos bugs y errores causados por la implementación del nuevo método.

Para concluir, se puede afirmar que implementar un patrón de diseño y crear un nuevo diseño para el proyecto son requerimientos / tareas importantes que deben ser realizadas de manera prioritaria. De esta manera se evitan los errores generados por el mal diseño, como también se facilita la comunicación entre el equipo de trabajo, ayudando a su vez a la comprensión del código.