

### **Proyecto 3 – Documento de reflexión**

**Sergio Franco Pineda - Juan Pablo Rodríguez Briceño - Juan Pablo Martínez Pineda**

#### **¿Qué cosas salieron bien?**

- El sistema de archivo, guardado y relacionamiento entre archivos y clases quedo bien relacionado, por ende, la información que contiene la aplicación y un proyecto dentro de la aplicación es manipulable en el sentido de creación y supresión.
- El sistema de archivos está automatizado, por ende, cuando hay algún cambio en alguna parte de la aplicación todo lo que está relacionado con esa parte cambia como debe cambiar y se actualiza sin necesidad de que el usuario se tenga que preocupar por hacer cambios en otros archivos relacionados.
- El diseño general de la aplicación es bueno en cuanto a sus clases, atributos, métodos, y relaciones.
- La aplicación es usable por consola, incluso con el sistema de WBS.

#### **¿Qué cosas salieron mal?**

- El producto final de nuestro grupo no cumplió con el objetivo
- Las entregas anteriores no abarcaban en total lo requerido y, por ende, fue más difícil cumplir el objetivo de la siguiente entrega al tener que completar primero las funcionalidades esperadas de la entrega anterior.
- La interfaz gráfica no está conectada completamente con la lógica del proyecto.
- La parte gráfica de la aplicación como reportes y gráficas no se pudo completar.

#### **¿Qué decisiones resultaron acertadas?**

- Una decisión acertada fue hacer uso de una librería externa para poder manejar de una mejor manera los archivos csv del proyecto.
- 
- 
- Otra decisión acertada fueron trabajar en los requerimientos que se buscaban cumplir de las anteriores entregas. Esto se debe a que, sin estas funciones, sería más difícil poder visualizar si se estaban logrando los objetivos de las nuevas entregas.
- 
- De la misma manera, en la entrega 2 se implementó dos maneras de poder correr el proyecto para que, aunque la lógica del programa y la interfaz no estaban conectados, se pudiera comprobar el estado de la interfaz y del código por separado.
- Hicimos uso de ramas. Esto es importante ya que se crearon más de una versión de los requerimientos, pero solo se implementó la que más se adecuaba a la problemática y, de esta manera, no tuvimos que realizar un gran cambio al código base a comparación de haber hecho estas mismas versiones en simultaneo sobre la misma rama.
- Relacionar los diferentes archivos y estructuras de datos por medio de números identificadores.
- Estar abiertos a replantear nuestros documentos de diseño anteriores.
- La manera en cómo se cargan los datos.

- Se hizo uso del patrón MVC, lo que nos facilitó la organización del código del proyecto.

### ¿Qué decisiones fueron problemáticas?

- Precisamente por trabajar en los puntos de las anteriores entregas fuera de tiempo, no se dejó suficiente tiempo para poder acabar las siguientes entregas, y esto se puede evidenciar principalmente en la última versión del proyecto ya que presenta fallas de lo trabajado en la versión 2 y lo que se esperaba lograr en esta última versión.
- El uso de la librería externa fue beneficioso, pero también causó problemas en cada uno porque podía presentar errores de importación.

### Problemas que hubo y a qué se debieron

- **Problemas de librería:** Al momento de implementar la librería que trabaja con los .csv esta no se logró subir al repositorio, lo cual dio problemas al resto de los integrantes ya que no podían probar los cambios realizados en el código. Esto se solucionó para la entrega 3.
- **Problemas de comunicación:** Nuestro grupo tuvo un problema al no comunicar que era lo que estaba siendo desarrollado o que era lo que faltaba por hacer. Ya que se trabajó simultáneamente en tres ocasiones sobre distintos requerimientos, pero los integrantes no comunicamos que era lo que se estaba haciendo, lo que hizo que se malgastara el tiempo.
- **Problemas de repositorios:** Algo similar ocurrió con repositorios de GitHub. Para el proyecto 3 se crearon dos repositorios que partían de la segunda versión del proyecto. Esto significó un problema que pudo ser evitado fácilmente al tener que unir el código de ambos repositorios manualmente.
- **Hubo un problema con dos de nuestros computadores:** El computador de Juan Pablo Rodríguez se actualizó a Windows 11 pero se descargó durante la instalación, lo que hizo que el computador tuviera que ser mandado a reparación y quedo inutilizable por 2 días. El computador de Sergio Franco en algunas ocasiones no encendía por varios días, lo que imposibilitó el avance en varias entregas.

### En cuanto a la tecnología:

- Aunque eran varias clases de teoría sobre las tecnologías que debíamos usar, el tiempo para hacer uso de estas era corto debido a diferentes afanes y tener que aprender al tiempo que se implementa.
- El fuerte tipado de JAVA resultaba en problemas a veces difíciles de comprender y resolver.

### En cuanto al enunciado:

- Había requerimientos que se hacían complejos por el poco tiempo y el uso bajo-intermedio de la tecnología en su momento, como, por ejemplo, cronometrar actividades en tiempo real y visualizar una matriz de actividad en el proyecto como la de GitHub.

**Recomendaciones:**

- Planear un horario flexible en el que se establezcan metas entre los requisitos del proyecto. Esto servirá para saber qué es lo que falta por hacer y poder planear y cumplir todos o la mayoría de los puntos en cada versión del proyecto.
- Anunciar cuando se esté trabajando en alguna parte del proyecto y especificar cual.
- Crear un repositorio nuevo para cada entrega del proyecto y anunciar la creación del mismo.
- Realizar commits constantemente, de esta manera los otros integrantes podrán tener la más reciente versión del código y será menos posible trabajar en algo que ya está hecho.
- Documentar cada cambio en el diseño.