

Documento de reflexión

Proyecto 1



Fig. 1 - Diagrama UML Proyecto 1

Para el proyecto 1 creamos 5 clases: proyecto, actividad, participante, modelo y controller. Los atributos de cada clase eran: **proyecto** => nombre, descripción, fecha de inicio y lista de actividades; **actividad** => documentación, implementación, pruebas, investigación, diseño, título, descripción, tipo, fecha, hora inicio, hora fin y participantes; **participante** => nombre y correo; **modelo** => proyecto; **controller** => sin atributos.

La clase “Model” se agregó porque que ayuda al proceso de guardado y cargado del proyecto. A su vez, la clase “controller” es la encargada de gestionar todo el proceso de la interacción con el usuario por medio de la consola.

Al iniciar el proyecto teníamos dos clases llamadas “guardador” y “loader” pero al tratarse de un proyecto pequeño, se decidió ponerlas dentro de la clase proyecto y utilizar directamente sus atributos. De igual forma, se decidió utilizar una “Linked List” a cambio de una tabla de hash para guardar las actividades del proyecto.

¿Qué cosas salieron bien y que cosas salieron mal?

El diseño del programa salió bien porque leímos adecuadamente el enunciado, fuimos encontrando los requerimientos y restricciones que nos daban y logramos organizarlo en el uml. Sin embargo, al momento de pasar al código se presentaron algunos problemas con el manejo de las clases o temas teóricos.

¿Qué decisiones resultaron acertadas y qué decisiones fueron problemáticas?

La mayoría de las decisiones fueron acertadas dado que siempre estuvimos buscando un diseño eficiente que cumpliera con los requerimientos.

¿Qué tipo de problemas tuvieron durante el desarrollo y a que se debió?

Durante el desarrollo hubo algunos problemas debido a que no teníamos mucho conocimiento de la tecnología y el lenguaje implementado en el proyecto.

Proyecto 2

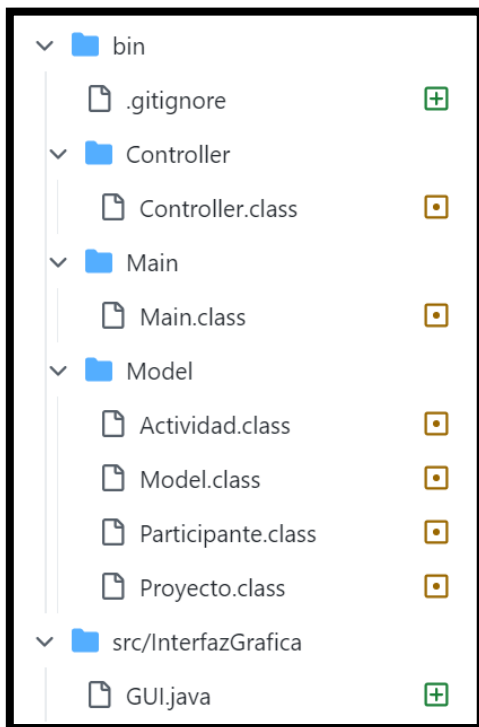


Fig. 2 – Creación clase GUI

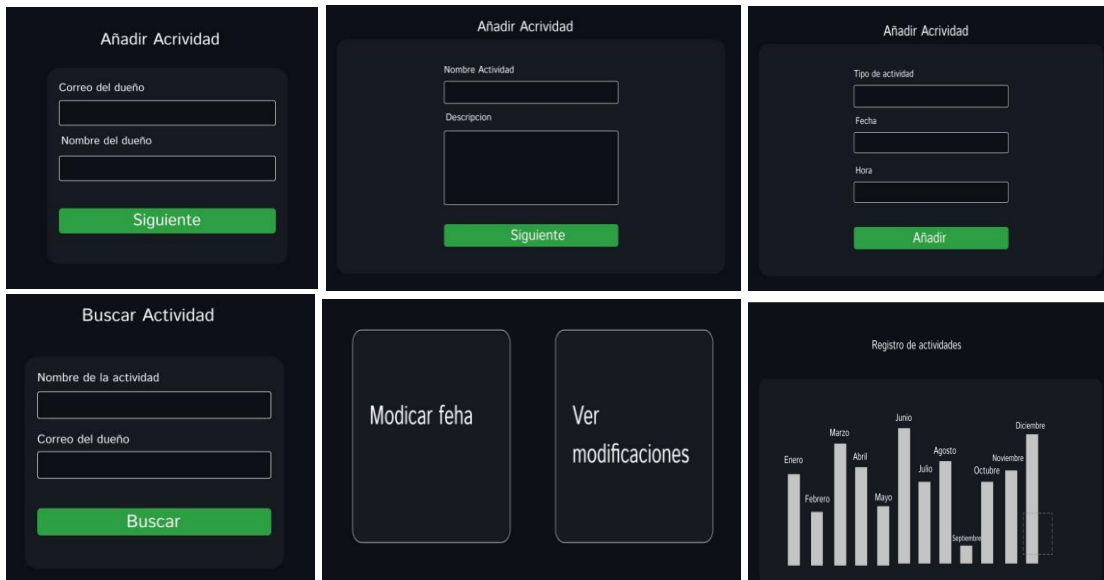
Para el proyecto 2 mantuvimos el mismo diseño del proyecto 1, pero se agregó un nuevo paquete InterfazGrafica con una nueva clase GUI. Esta clase implementaba ActionListener para poder tener interacción con el usuario y tenía atributos privados estáticos que se encargaban de dar forma a la interfaz. Entre ellos habían JLabel, JTextField y JButton.

```
+ package InterfazGrafica;
+ import java.awt.Color;
+ import java.awt.event.ActionEvent;
+ import java.awt.event.ActionListener;
+
+ import javax.swing.JButton;
+ import javax.swing.JFrame;
+ import javax.swing.JLabel;
+ import javax.swing.JPanel;
+ import javax.swing.JTextField;
+
+ public class GUI implements ActionListener{
```

Fig. 3 - Clase GUI

El siguiente es el diseño inicial planteado para la primera entrega del proyecto 2:





¿Qué cosas salieron bien y que cosas salieron mal?

El diseño del programa salió bien porque la interfaz principal estaba compuesta de otras ventanas que permitían cumplir con todas las necesidades del usuario al utilizar el programa. Sin embargo, al momento de pasar al código se presentaron algunos problemas con el uso del ActionListener y su función para que al presionar un botón específico se ejecutara un proceso definido.

¿Qué decisiones resultaron acertadas y qué decisiones fueron problemáticas?

La mayoría de las decisiones fueron acertadas porque siempre estuvimos buscando hacer una interfaz que fuera amigable con el usuario y al mismo tiempo eficiente. Sin embargo, una decisión algo problemática fue dividir el proceso de añadir actividad en tres secciones que pedían diferentes datos, pues el desarrollo de la interfaz fue mas largo al tener que crear tres ventanas en lugar de una.

¿Qué tipo de problemas tuvieron durante el desarrollo y a que se debió?

Durante el desarrollo tuvimos algunos problemas para conectar el código de la interfaz (parte de diseño) con el código de la lógica (parte funcional). También se dificultó la creación del diagrama de registro de actividades usando el framework swing.

Proyecto 3

¿Qué cosas salieron bien y que cosas salieron mal?

El diseño del programa salió bien porque se tomaron en cuenta los nuevos requerimientos y restricciones del enunciado. Sin embargo, hubo algunas complicaciones al pasar a desarrollarlo.

¿Qué decisiones resultaron acertadas y qué decisiones fueron problemáticas?

Tuvimos decisiones acertadas y decisiones problemáticas. El objetivo siempre fue hacer un programa eficiente y amigable con el usuario, pero el desarrollo se complicó por algunos problemas externos.

¿Qué tipo de problemas tuvieron durante el desarrollo y a qué se debió?

Durante el desarrollo tuvimos algunos problemas porque hubo un problema con el repositorio del proyecto 2. Debido a esto tuvimos que realizar bastantes cambios y no alcanzamos a tener una implementación completa de la aplicación para la entrega.

Link del repositorio del proyecto 2 que si cuenta con interfaz gráfica :
<https://github.com/Pabluchenkoo/entrega-final>

Trabajo general

Durante el desarrollo de los proyectos hubo cosas positivas y cosas negativas. En los tres proyectos la parte del diseño siempre fue mejor que la parte del desarrollo. Sin embargo, las dos partes fueron necesarias para lograr los objetivos.



Fig. 4 - Trabajo general