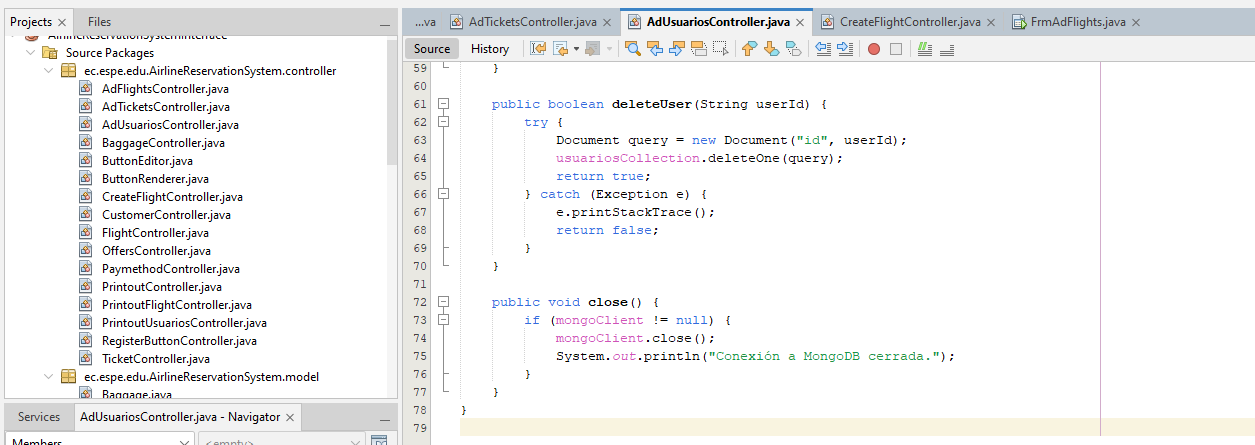
**HW23 – SOLID Principles**

1. **Liskov Substitution Principle**



**Explanation:**

In this class and in two others we repeat code to close the connection to the database which does not make it reusable.

**Possible solution:**

Create an abstract class with the method that is repeated and implement it in each of the classes adding the code that you needed applying the Liskov Substitution principle.

1. **Liskov Substitution Principle**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

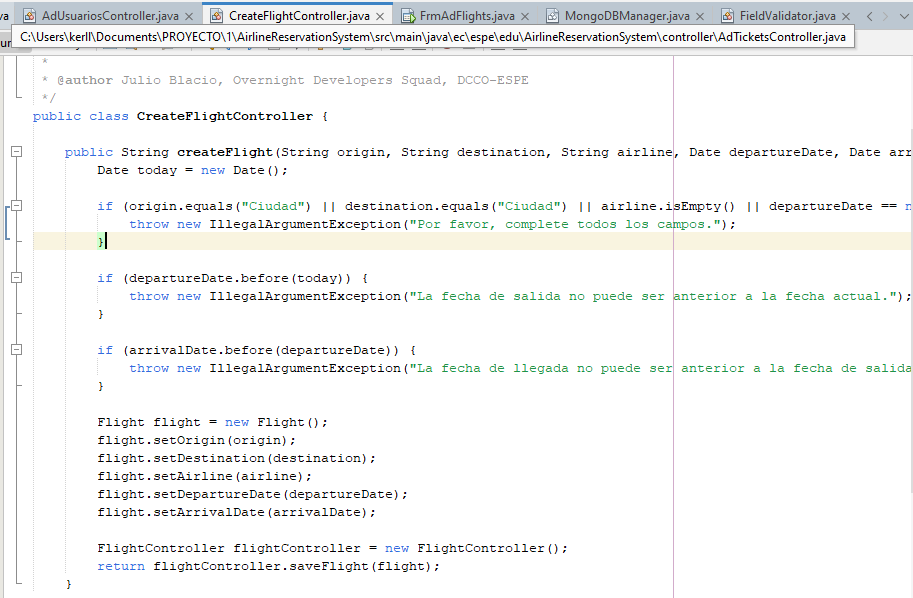
**Explanation:**

In this class, and like the other two previous ones, code was repeated that in the end could not be reused.

**Possible solution:**

In the class I created previously, the methods for updating and deleting documents could be increased, which would help to better structure the classes.

1. **Single Responsability Principle**



**Explanation:**

This code violates the principle of single responsibility since this function should only create the flight but not validate its fields.

**Possible solution:**

Move the field validation code to a class in utils to only call it in this class and thus keep this function clean.

1. **Single Responsability Principle**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Explanation:**

In this code we give the button more actions than it should have and we overload it.

**Possible solution:**

Create a class with a function that clears the fields and only makes a call to the button.