

## HW 24 – SOLID Principles Fixed

## 1. Liskov Substitution Principle

```

12 13 14 15 16 17 18
    /**
     *
     * Author: Kerlly Chiriboga, ODS
     */
    public class AdUsuariosController extends MongoDBManager {

    /**
     *
     * @author Kerlly Chiriboga, ODS
     */

    public abstract class MongoDBManager {
        protected MongoClient mongoClient;

        public MongoDBManager(String connectionString) {
            this.mongoClient = MongoClient.create(connectionString);
        }

        public void close() {
            if (mongoClient != null) {
                mongoClient.close();
                System.out.println("Conexión a MongoDB cerrada.");
            }
        }
    }
}

```

## 2. Liskov Substitution Principle

```
*
 * @author Kerlly Chiriboga, ODS
 */
public abstract class MongoDBManager {

    protected MongoClient mongoClient;
    protected MongoDB database;

    public MongoDBManager(String connectionString, String databaseName) {
        this.mongoClient = MongoClient.create(connectionString);
        this.database = mongoClient.getDatabase(databaseName);
    }

    public boolean updateDocument(MongoCollection<Document> collection, Document query, Document update) {
        try {
            UpdateResult result = collection.updateOne(query, new Document("$set", update));
            return result.getModifiedCount() > 0;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }

    public boolean deleteDocument(MongoCollection<Document> collection, Document query) {
        try {
            DeleteResult result = collection.deleteOne(query);
            return result.getDeletedCount() > 0;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

### 3. Single Responsibility Principle

```
AdUsuariosController.java x CreateFlightController.java x FrmAdFlights.java x MongoDBManager.java x FieldValidator.java x
Source History
1.0
1.1 /**
1.2  * @author Kerlly Chiriboga, ODS
1.3  */
1.4 public class FieldValidator {
1.5
1.6     public static boolean validateTicketFields(JSpinner ticketsNumberSpn, JTextField userNameTxt, JTextField f
1.7         int ticketNumber = (int) ticketsNumberSpn.getValue();
1.8         String customerName = userNameTxt.getText().trim();
1.9         String flightId = flightIdTxt.getText().trim();
2.0         String ticketClass = cmbClass.getSelectedItem().toString().trim();
2.1
2.2         if (ticketNumber == 0 || customerName.isEmpty() || flightId.isEmpty() || ticketClass.isEmpty()) {
2.3             JOptionPane.showMessageDialog(parentComponent, "Todos los campos son obligatorios.", "Error", JOpt
2.4             return false;
2.5         }
2.6
2.7         if (!customerName.matches("[A-Za-z ]+")) {
2.8             JOptionPane.showMessageDialog(parentComponent, "El nombre solo puede contener letras y espacios.",
2.9             return false;
2.10        }
2.11
2.12        if (!flightId.matches("\\d+")) {
2.13            JOptionPane.showMessageDialog(parentComponent, "El ID del vuelo solo puede contener dígitos.", "Er
2.14            return false;
2.15        }
2.16
2.17        return true;
2.18    }
2.19
2.20    public static boolean validateFlightFields(String origin, String destination, String airline, Date departu
2.21        Date today = new Date();
2.22    }
```

### 4. Single Responsibility Principle

```
Start Page x ClearFields.java x
Source History
11
12 /**
13  * @author Kerlly Chiriboga, ODS
14  */
15 public class ClearFields {
16
17     public static void clearFieldsTicket(JSpinner ticketsNumberSpn, JTextField userNameTxt, JTextField flightI
18         ticketsNumberSpn.setValue(0);
19         userNameTxt.setText("");
20         flightIdTxt.setText("");
21         cmbClass.setSelectedIndex(0);
22     }
23
24     public static void clearFieldsFlight(JComboBox<String> cmbOrigin, JComboBox<String> cmbDestination, JTextF
25         cmbOrigin.setSelectedIndex(0);
26         cmbDestination.setSelectedIndex(0);
27         txtAirline.setText("");
28         jDateDeparture.setDate(null);
29         jDateArrival.setDate(null);
30     }
```