

## HW22 - Fix the pitfalls

### Pitfall #1

```
...Va FrmCreateFlight.java x FrmTicket.java x ClearFields.java x FieldValidator.java x FrmRegisterCustomer.java x
Source Design History
183 private void btnCreateMouseExited(java.awt.event.MouseEvent evt) {
185     btnCreate.setBackground(new Color(157, 117, 185));
186 }
187
188 private void btnCreateActionPerformed(java.awt.event.ActionEvent evt) {
189     try {
190         String origin = cmbOrigin.getSelectedItem().toString();
191         String destination = cmbDestination.getSelectedItem().toString();
192         String airline = txtAirline.getText();
193         Date departureDate = jDateDeparture.getDate();
194         Date arrivalDate = jDateArrival.getDate();
195
196         CreateFlightController controller = new CreateFlightController();
197         String flightId = controller.createFlight(origin, destination, airline, departureDate, arrivalDate);
198
199         JOptionPane.showMessageDialog(this, "Vuelo creado exitosamente.", "Éxito", JOptionPane.INFORMATION_MESSAGE);
200
201         ClearFields.clearFieldsFlight(cmbOrigin, cmbDestination, txtAirline, jDateDeparture, jDateArrival);
202     } catch (IllegalArgumentException e) {
203         JOptionPane.showMessageDialog(this, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
204     } catch (Exception e) {
205         JOptionPane.showMessageDialog(this, "Ocurrió un error al crear el vuelo: " + e.getMessage(), "Error",
206                                     JOptionPane.ERROR_MESSAGE);
207     }
208 }
209
210 private void btnBackMouseClicked(java.awt.event.MouseEvent evt) {
211     // TODO add your handling code here:
212 }
213
```

### Pitfall #2

```
if (ticketId != null) {
    JOptionPane.showMessageDialog(this,
        "<html><body style='width: 300px;'>"
        + "<p style='font-size: 14px;'>Ha reservado con éxito su/s Ticket/s.</p>"
        + "<p style='font-size: 14px;'>El ID de su boleto es: <strong>" + ticketId + "</strong>"
        + "<p style='font-size: 14px;'>Por favor, proceda con el pago.</p>"
        + "</body></html>",
        "Éxito",
        JOptionPane.INFORMATION_MESSAGE);

    PayMethod payMethodDialog = new PayMethod();
    payMethodDialog.setLocationRelativeTo(null);
    payMethodDialog.setVisible(true);
} else {
    JOptionPane.showMessageDialog(this, "Error al guardar el boleto en la base de datos.", "Error",
        JOptionPane.ERROR_MESSAGE);
    ClearFields.clearFieldsTicket(ticketsNumberSpn, userNameTxt, flightIdTxt, cmbClass);
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Número de boleto o ID de vuelo no válido: " + e.getMessage(), "Error",
        JOptionPane.ERROR_MESSAGE);
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Error al registrar el ticket: " + e.getMessage(), "Error",
        JOptionPane.ERROR_MESSAGE);
}
```

### Pitfall #3

```
package ec.espe.edu.AirlineReservationSystem.utils;

import java.awt.Component;
import javax.swing.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.JSpinner;
import javax.swing.JTextField;

/**
 * @author Kerlly Chiriboga, ODS
 */
public class FieldValidator {

    public static boolean validateTicketFields(JSpinner ticketsNumberSpn, JTextField userNameTxt, JTextField flightIdTxt, JComboBox cmbClass) {
        int ticketNumber = (int) ticketsNumberSpn.getValue();
        String customerName = userNameTxt.getText().trim();
        String flightId = flightIdTxt.getText().trim();
        String ticketClass = cmbClass.getSelectedItem().toString().trim();

        if (ticketNumber == 0 || customerName.isEmpty() || flightId.isEmpty() || ticketClass.isEmpty()) {
            JOptionPane.showMessageDialog(parentComponent, "Todos los campos son obligatorios.", "Error", JOptionPane.ERROR_MESSAGE);
            return false;
        }

        if (!customerName.matches("[A-Za-z ]+")) {
            JOptionPane.showMessageDialog(parentComponent, "El nombre solo puede contener letras y espacios.", "Error", JOptionPane.ERROR_MESSAGE);
            return false;
        }
    }
}
```

### Pitfall #4

```
private void registerButtonActionPerformed(java.awt.event.ActionEvent evt) {
    registerButton.setBackground(new Color(157, 117, 185));
}

private void registerButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String idDocument = idDocumentTxt.getText().trim();
        String name = nameTxt.getText().trim();
        String lastName = lastNameTxt.getText().trim();
        String email = emailTxt.getText().trim();
        String phoneNumber = phoneTxt.getText().trim();
        String username = usernameTxt.getText().trim();
        String password = new String(passwordTxt.getPassword()).trim();
        String city = cityTxt.getText().trim();
        String state = stateTxt.getText().trim();
        String postalCode = zipTxt.getText().trim();
        Date dateOfBirth = jDateChooser1.getDate();
        String gender = gendercmb.getSelectedItem().toString();

        RegisterButtonController controller = new RegisterButtonController();
        controller.registerCustomer(idDocument, name, lastName, email, phoneNumber, username, password, city, state, postalCode, dateOfBirth, gender);

        JOptionPane.showMessageDialog(this, "Cliente registrado exitosamente!");

        ClearFields.clearFieldsUser(idDocumentTxt, nameTxt, lastNameTxt, emailTxt, phoneTxt, usernameTxt, passwordTxt, cityTxt, stateTxt, zipTxt, jDateChooser1, gendercmb);
    } catch (IllegalArgumentException e) {
        JOptionPane.showMessageDialog(this, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error al registrar los datos del cliente: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

## Pitfall #5

```
...va FrmCreateFlight.java x FrmTicket.java x ClearFields.java x FieldValidator.java x FrmRegisterCustomer.java x
Source History
1 package ec.espe.edu.AirlineReservationSystem.utils;
2
3 import com.toedter.calendar.JDateChooser;
4 import javax.swing.JComboBox;
5 import javax.swing.JPasswordField;
6 import javax.swing.JSpinner;
7 import javax.swing.JTextField;
8
9 /**
10  *
11  * @author Kerlly Chiriboga, ODS
12  */
13 public class ClearFields {
14
15     public static void clearFieldsTicket(JSpinner ticketsNumberSpn, JTextField userNameTxt, JTextField flightI
16         ticketsNumberSpn.setValue(0);
17         userNameTxt.setText("");
18         flightIdTxt.setText("");
19         cmbClass.setSelectedIndex(0);
20     }
21
22     public static void clearFieldsFlight(JComboBox<String> cmbOrigin, JComboBox<String> cmbDestination, JTextF
23         cmbOrigin.setSelectedIndex(0);
24         cmbDestination.setSelectedIndex(0);
25         txtAirline.setText("");
26         jDateDeparture.setDate(null);
27         jDateArrival.setDate(null);
28     }
29
30     public static void clearFieldsUser(JTextField idDocumentTxt, JTextField nameTxt, JTextField lastNameTxt, J
31         JTextField usernameTxt, JPasswordField passwordTxt, JTextField cityTxt, JTextField stateTxt, JText
```

## Pitfall #6

```

    *
    * @author Kerlly Chiriboga, ODS
    */
    public class FlightTableManager {

    *
    * @author Kerlly Chiriboga, ODS
    */
    public class UserTableManager {

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