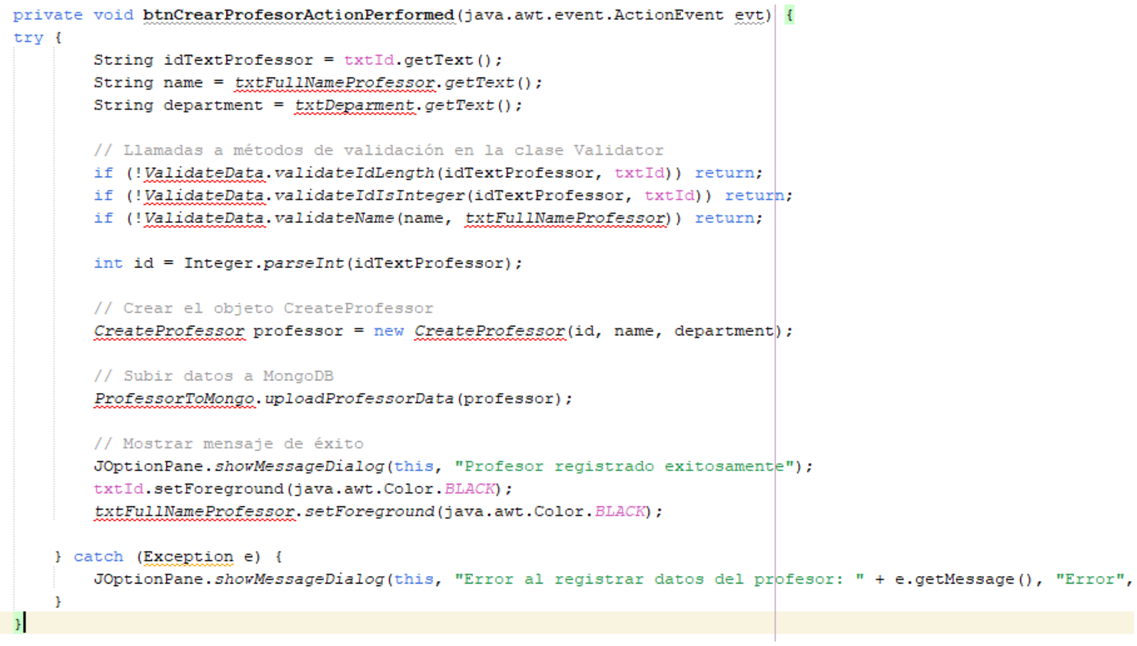
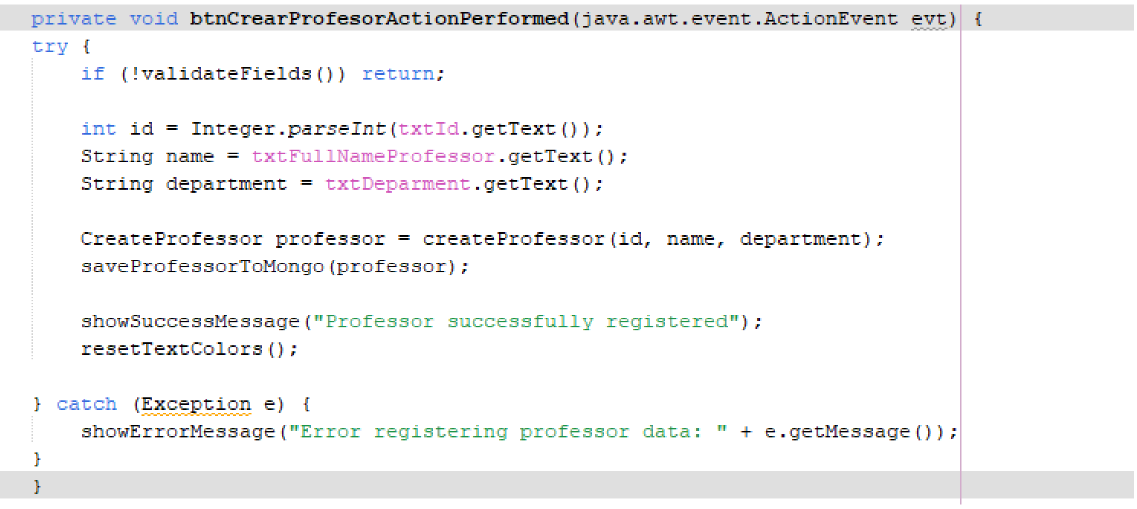
HW 22 FIX THE PITFALLS

in the code when creating a new teacher, it was fine, the only bad thing is that the button where it was placed made many functions, and it should do few or even 1, and that was what was improved.

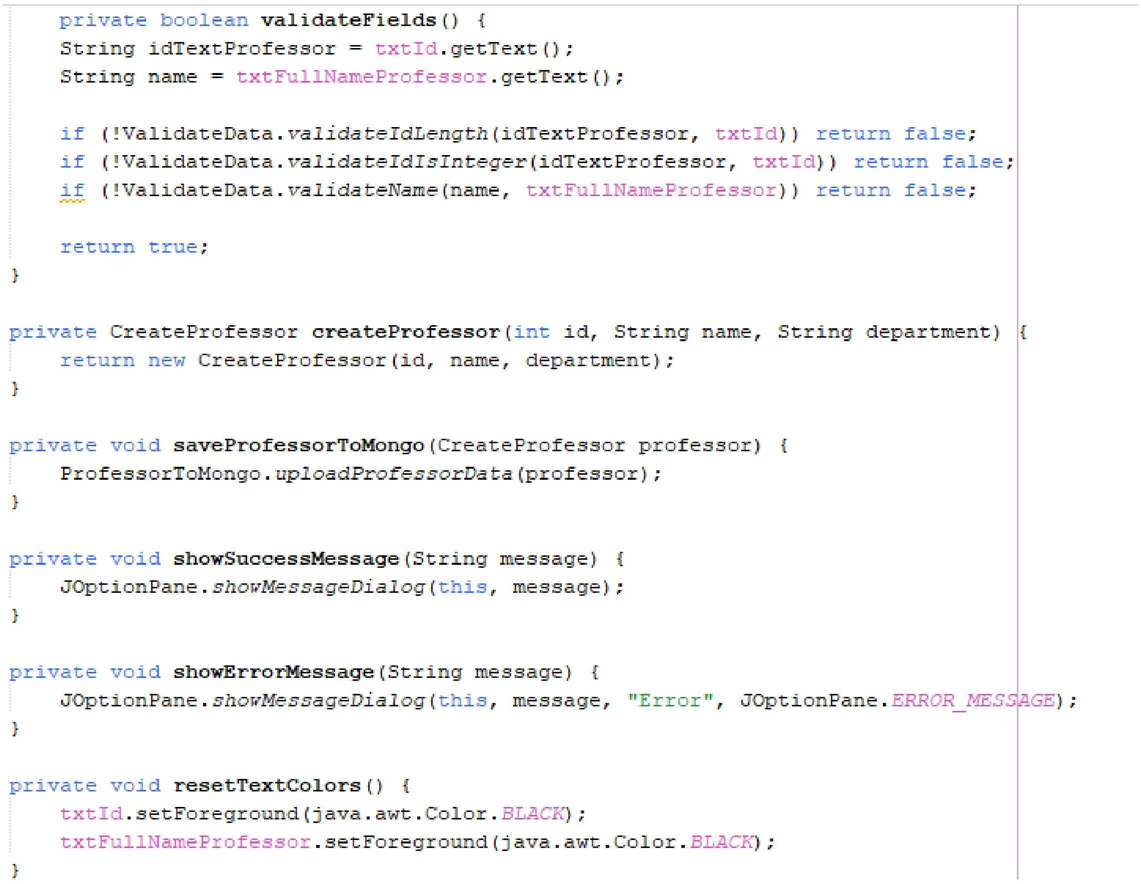


```java
private void btnCrearProfesorActionPerformed(java.awt.event.ActionEvent evt) {
try {
        String idTextProfessor = txtId.getText();
        String name = txtFullNameProfessor.getText();
        String department = txtDeparment.getText();

        // Llamadas a métodos de validación en la clase Validator
        if (!ValidateData.validateIdLength(idTextProfessor, txtId)) return;
        if (!ValidateData.validateIdIsInteger(idTextProfessor, txtId)) return;
        if (!ValidateData.validateName(name, txtFullNameProfessor)) return;

        int id = Integer.parseInt(idTextProfessor);

        // Crear el objeto CreateProfessor
        CreateProfessor professor = new CreateProfessor(id, name, department);

        // Subir datos a MongoDB
        ProfessorToMongo.uploadProfessorData(professor);

        // Mostrar mensaje de éxito
        JOptionPane.showMessageDialog(this, "Profesor registrado exitosamente");
        txtId.setForeground(java.awt.Color.BLACK);
        txtFullNameProfessor.setForeground(java.awt.Color.BLACK);

    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error al registrar datos del profesor: " + e.getMessage(), "Error",
    }
}
```

now the code of the button to create the teacher has few functions, and of which there are functions outside of this to not alter the code.



```java
private void btnCrearProfesorActionPerformed(java.awt.event.ActionEvent evt) {
try {
    if (!validateFields()) return;

    int id = Integer.parseInt(txtId.getText());
    String name = txtFullNameProfessor.getText();
    String department = txtDeparment.getText();

    CreateProfessor professor = createProfessor(id, name, department);
    saveProfessorToMongo(professor);

    showSuccessMessage("Professor successfully registered");
    resetTextColors();

} catch (Exception e) {
    showErrorMessage("Error registering professor data: " + e.getMessage());
}
}
```



```java
    private boolean validateFields() {
    String idTextProfessor = txtId.getText();
    String name = txtFullNameProfessor.getText();

    if (!ValidateData.validateIdLength(idTextProfessor, txtId)) return false;
    if (!ValidateData.validateIdIsInteger(idTextProfessor, txtId)) return false;
    if (!ValidateData.validateName(name, txtFullNameProfessor)) return false;

    return true;
}

private CreateProfessor createProfessor(int id, String name, String department) {
    return new CreateProfessor(id, name, department);
}

private void saveProfessorToMongo(CreateProfessor professor) {
    ProfessorToMongo.uploadProfessorData(professor);
}

private void showSuccessMessage(String message) {
    JOptionPane.showMessageDialog(this, message);
}

private void showErrorMessage(String message) {
    JOptionPane.showMessageDialog(this, message, "Error", JOptionPane.ERROR_MESSAGE);
}

private void resetTextColors() {
    txtId.setForeground(java.awt.Color.BLACK);
    txtFullNameProfessor.setForeground(java.awt.Color.BLACK);
}
```