

Artigo de documentação do UrGrade

Fernando J. Tozato de Siqueira

Bloco A – Universidade Estácio de Sá (UNESA)
24020-340 – Niterói – RJ – Brasil

Tozato.fernando2004@gmail.com

Abstract. *This article documents UrGrade, a Java database program that stores classes and their grades.*

Resumo. *Este artigo documenta o UrGrade, um programa de banco de dados em Java que armazena matérias e suas notas.*

1. Informações Gerais

O nome UrGrade foi escolhido pois o programa lida com as notas do usuário. O objetivo do sistema é facilitar a gerência de notas de um aluno.

2. Funcionamento

O programa pode ser inicializado por meio de um arquivo jar executável. Para isso é necessário que o usuário tenha pelo menos o JDK 15 ou mais recente e o PostgreSQL 15 ou mais recente. A pasta necessária para utilizar o software está disponível no repositório git sob o link <https://github.com/Fernando-Tozato/TrabalhoFinalJava>.

3. Classes e métodos

No total, o Código Fonte apresenta 10 classes diferentes, sendo elas: Main, API, GUI, CR, Erro, Login, LoginErro, MainInterface, Materia e Nota. A maioria delas implementa interfaces gráficas em Swing.

3.1. Classe Main

Essa classe é a única que conta com o método main, consequentemente, é a única que pode ser executada. O único método presente nela é o próprio main, ele apenas instancia as classes API e GUI uma vez cada e chama o método GUI.init.

```

1
2 import my.app.API;
3 import my.app.GUI;
4
5 // Essa classe inicia todo o programa instanciando uma classe API que será usada
6 // até o final do mesmo e uma classe GUI que iniciará todo o Front-end
7 public class Main {
8
9     public static void main(String[] args) {
10         API api = new API();
11         GUI gui = new GUI(api);
12         gui.init();
13     }
14 }
15

```

Figura 1. Classe Main

3.2. Classe API

É a classe que faz toda a comunicação com o banco de dados. Ela tem vários métodos e atributos.

Tabela 1. Principais métodos da classe API

| Métodos | Função |
|------------------|---|
| validarLogin | Faz a validação do login com uma conexão ao PostgreSQL. |
| criarTabela | Cria a tabela "materias" caso ela não exista. |
| lerTabela | Pega os dados da tabela e envia para o Front-end. |
| adicionarMateria | Recebe um Object com as informações de uma matéria e adiciona à tabela. |
| adicionarNota | Atualiza uma nota por vez. |
| calcularCR | Calcula o Coeficiente de Rendimento. |
| deletarTabela | Deleta a tabela. |

3.3. Classe GUI

Essa classe instancia a classe MainInterface, que inicia todo o Front-End do programa. Ela também define o Look And Feel como o FlatLaf Dark, chamando o método FlatDarkLaf.setup(). O Look And Feel do programa é o que define toda a padronização visual em todas as interfaces criadas. O único método da classe é o init, que define a visibilidade de MainInterface como true.

```

1  package my.app;
2
3  import my.gui.MainInterface;
4  import com.formdev.flatlaf.FlatDarkLaf;
5
6  // Essa classe instancia a classe MainInterface, que inicia todo o Front-end
7  public class GUI {
8
9      private final API api;
10     private final MainInterface main;
11
12     public GUI(API api) {
13         FlatDarkLaf.setup();
14         this.api = api;
15         this.main = new MainInterface();
16         main.setApi(api);
17     }
18
19     public void init() {
20         main.setVisible(true);
21     }
22 }
23

```

Figura 2. Classe GUI

3.4. Classes Login e LoginErro

Essas são as classes que pedem os dados de login ao usuário e chama o método `API.validarLogin`, fazendo assim a verificação de login. Caso o login não seja válido, a classe `LoginErro` é instanciada e a janela anterior é fechada. As duas têm métodos iguais, a única diferença entre elas é um aviso em letras vermelhas que aparece na parte inferior da janela.

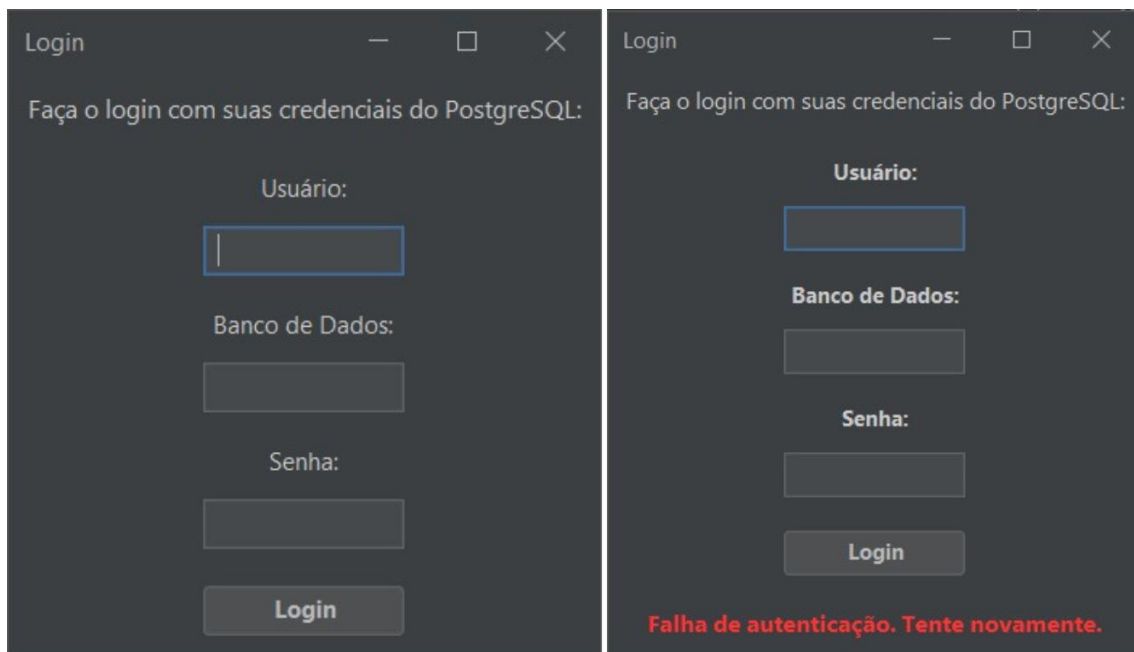


Figura 3. Janela Login à esquerda e LoginErro à direita

3.5. Classe MainInterface

É a classe mais importante da parte visual do programa. Ela que cria a janela principal, onde o usuário escolhe todas as ações. Tem vários métodos, os principais estão documentados na tabela abaixo.

Tabela 2. Principais métodos da classe MainInterface

| Método | Função |
|------------------|---|
| initComponents | Esse método define a janela a ser criada. |
| entrar | Esse método é chamado quando o usuário aciona o item "Entrar" no menu "Login" que fica na barra na parte superior da janela. Ele instancia a classe Login que faz a verificação do login. |
| sair | Esse método é chamado quando o usuário aciona o item "Sair" no menu com o nome do usuário que fica na barra na parte superior da janela. Ele troca o nome do menu para "Login", troca a visualização dos itens e define os dados de conexão como "". |
| adicionarMateria | Esse método é chamado quando o usuário aciona o botão "Adicionar Matéria" no lado direito da janela. Primeiro, ele verifica se o usuário não fez login, caso não tenha feito, ele instancia a classe Erro e define sua visualização como true. Caso tenha feito, ele instancia a classe Materia e define sua visualização como true. |
| adicionarNota | Esse método é chamado quando o usuário aciona o botão "Adicionar Nota" no lado direito da janela. Primeiro, ele verifica se o usuário não fez login, caso não tenha feito, ele instancia a classe Erro e define sua visualização como true. Caso tenha feito, ele instancia a classe Nota e define sua visualização como true. |
| calcularCR | Esse método é chamado quando o usuário aciona o botão "Calcular CR" no lado direito da janela. Primeiro, ele verifica se o usuário não fez login, caso não tenha feito, ele instancia a classe Erro e define sua visualização como true. Caso tenha feito, ele chama o método API.calcularCR e atribui seu retorno à variável nota, depois ele instancia a classe CR e define sua visualização como true. |
| deletarTabela | Esse método é chamado quando o usuário aciona o botão "Deletar Tabela" no lado direito da janela. Primeiro, ele verifica se o usuário não fez login, caso não tenha feito, ele instancia a classe Erro e define sua visualização como true. Caso tenha feito, ele chama os métodos API.deletarTabela e limparTabela. |
| loginDone | Esse método é chamado quando o login do usuário é validado e define o texto do menu na barra superior como o nome de usuário |

| | |
|----------------|--|
| | utilizado no login, troca a visualização dos itens, chama os métodos API.criarTabela e escreverTabela. |
| escreverTabela | Esse método define o modelo de jTable1 com os dados da tabela matérias. |
| limparTabela | Esse método limpa o modelo de jTable1 apagando os dados. |

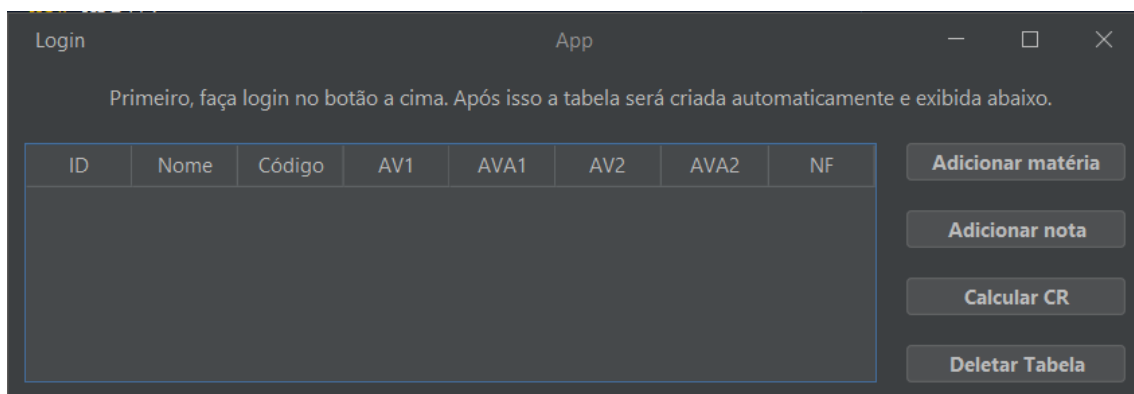


Figura 4. Janela MainInterface

3.6. Classes Materia e Nota

São classes que inserem dados na tabela. A classe Materia insere uma matéria inteira com o comando INSERT do SQL e a classe Nota atualiza uma nota utilizando o comando UPDATE. As duas têm vários métodos diferentes e são extremamente importantes para o funcionamento pleno do programa.

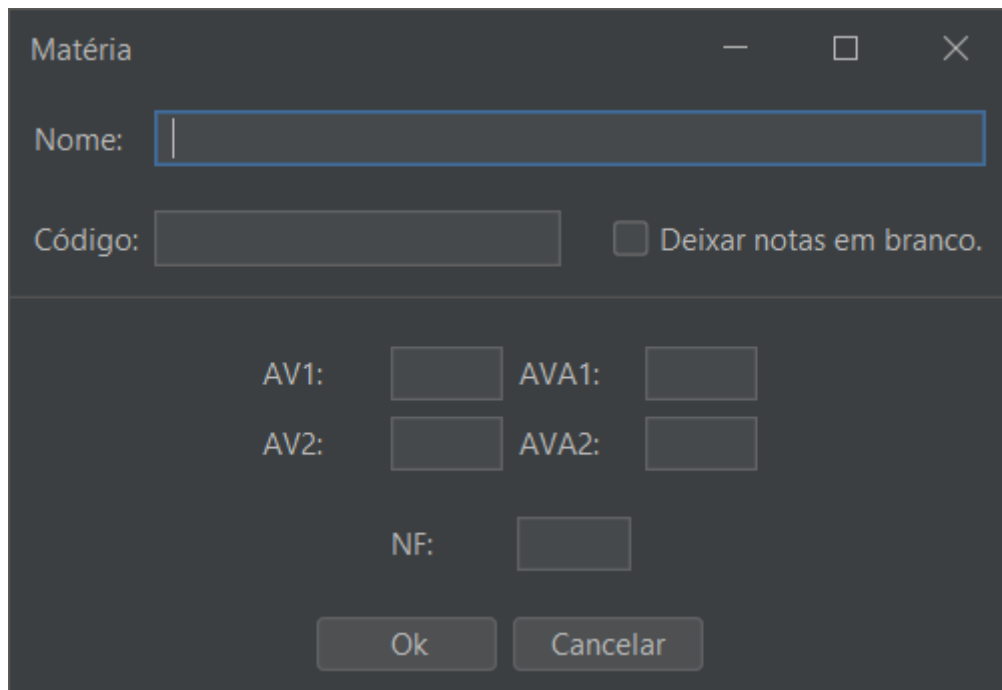


Figura 4. Janela Materia

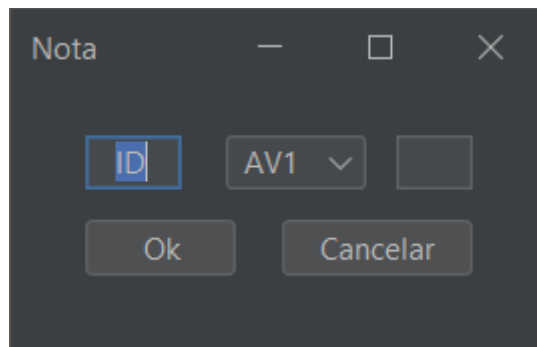


Figura 4. Janela Nota

4. Conclusão

Estou extremamente feliz com o resultado, ele reflete todas as minhas expectativas. O código é encapsulado, abstrato e seguro. Talvez ainda existam algumas melhorias a serem feitas, mas ele resolve o problema proposto e segue todas as exigências do trabalho.