

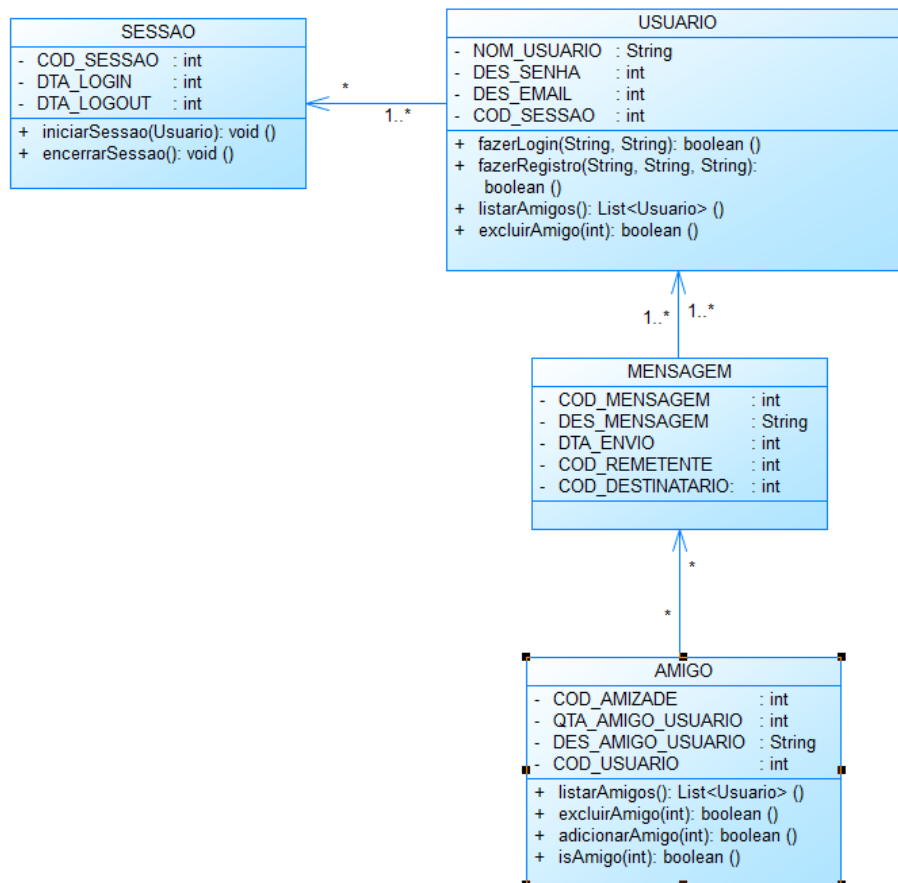
Introdução

Este projeto representa a implementação de uma aplicação de rede social em Java, utilizando a plataforma JavaFX para a construção da interface gráfica e o MySQL como sistema de gerenciamento de banco de dados. Sua proposta visa oferecer funcionalidades essenciais, como login de usuários, envio de mensagens entre amigos e gerenciamento da lista de contatos. Por meio dessa aplicação, os usuários podem interagir de forma eficaz, promovendo uma experiência básica, porém abrangente, no contexto de uma rede social.

Desenvolvimento

Nesta seção, apresentaremos uma visão geral do desenvolvimento do projeto, incluindo detalhes sobre a estrutura do código, as principais funcionalidades inovadoras e a abordagem adotada para a construção da interface gráfica. Vale ressaltar que as informações baseadas nas especificações fornecidas e nas estruturas comuns de projetos de redes sociais.

Diagrama de Classes



Estrutura do Projeto

O projeto é organizado em diversas classes, cada uma desempenhando papéis específicos no sistema. Abaixo estão algumas das principais classes:

- Usuario: Representa um usuário na rede social, armazenando informações como código de usuário, nome, senha e e-mail. Esta classe possui métodos para realizar operações como login, registro, listagem e interações com amigos.

```
public class Usuario {  
    // ... outros métodos e atributos  
  
    public boolean fazerLogin(String email, String senha) {  
        // Lógica de autenticação  
        // Retorna verdadeiro se o login for bem-sucedido, falso caso contrário  
    }  
  
    // ... outros métodos  
}
```

- Mensagem: Modelo de mensagem enviada entre usuários.

```
public class Mensagem {  
    // ... outros métodos e atributos  
  
    public void enviarMensagem() {  
        // Lógica para enviar a mensagem  
    }  
  
    // ... outros métodos  
}
```

- Amigo: Representa a relação de amizade entre dois usuários, mantendo detalhes como código de amizade, quantidade de amigos, entre outros.

```
public class Amigo {  
    // ... outros métodos e atributos  
  
    public List<Usuario> listarAmigos() {  
        // Lógica para listar amigos  
    }  
  
    // ... outros métodos  
}
```

- SessaoUsuario: Classe responsável por gerenciar a sessão do usuário, mantendo informações sobre o usuário logado.

```
1 package br.com.projeto.redesocial;
2
3 import br.com.projeto.redesocial.backend.Usuario;
4
5 public class SessaoUsuario {
6     private static Usuario usuarioLogado;
7
8     public static Usuario getUsuarioLogado() {
9         return usuarioLogado;
10    }
11
12    public static void setUsuarioLogado(Usuario usuario) {
13        SessaoUsuario.usuarioLogado = usuario;
14    }
15 }
```

- Interface Gráfica - Login: O controle da tela de login é feito pela aula LoginController. A seguir, um trecho que exemplifica o tratamento do clique no botão de login:

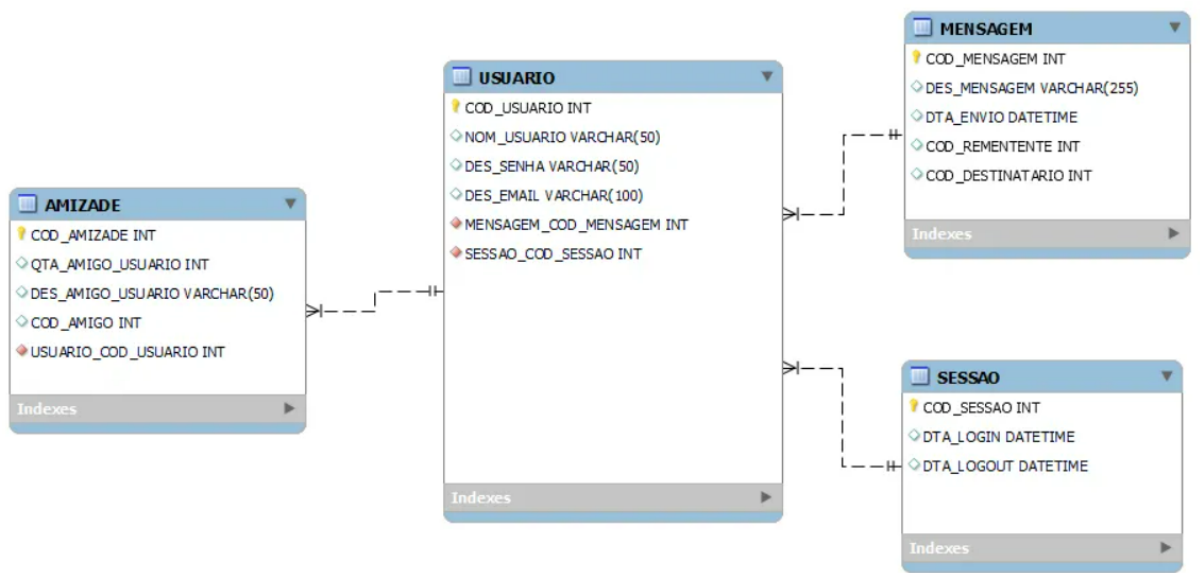
```
@FXML
private void handleLoginButtonAction(ActionEvent event) {
    // Obter email e senha dos campos de texto
    String email = emailField.getText();
    String senha = passwordField.getText();

    // Criar uma instância de Usuario
    Usuario usuario = new Usuario(email, senha);

    // Tentar fazer login
    boolean loginBemSucedido = usuario.fazerLogin(email, senha);
    if (loginBemSucedido) {
        // Lógica para redirecionar para a próxima tela
    } else {
        // Mostrar mensagem de erro
        showAlert("Erro de Login", "Email ou senha incorretos. Login falhou.", AlertType.ERROR);
    }
}
```

Banco de Dados

A estrutura do banco de dados é fundamental para o funcionamento da aplicação. Abaixo está o modelo de diagrama de banco de dados que ilustra a relação entre as entidades.



Parte do código SQL do modelo de banco de dados:

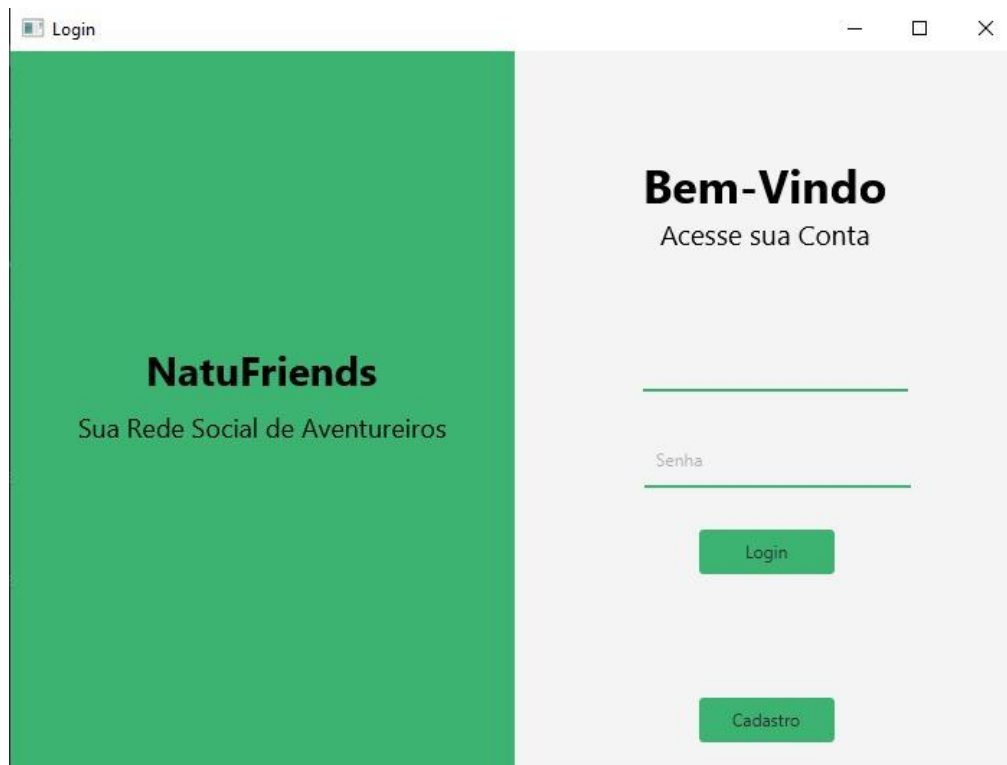
```
CREATE DATABASE RedeSocial;  
USE RedeSocial;
```

Criação da tabela USUARIO

```
CREATE TABLE USUARIO (  
    COD_USUARIO INT AUTO_INCREMENT PRIMARY KEY,  
    NOM_USUARIO VARCHAR(50),  
    DES_SENHA VARCHAR(50),  
    DES_EMAIL VARCHAR(100),  
    COD_MENSAGEM INT,  
    COD_SESSAO INT,  
    FOREIGN KEY (COD_MENSAGEM) REFERENCES  
    MENSAGEM(COD_MENSAGEM),  
    FOREIGN KEY (COD_SESSAO) REFERENCES SESSAO(COD_SESSAO)  
);  
...
```

Design da Interface

Página de Login: Uma página de login é uma porta de entrada para os usuários. Nela, é possível inserir as credenciais para acessar a rede social.



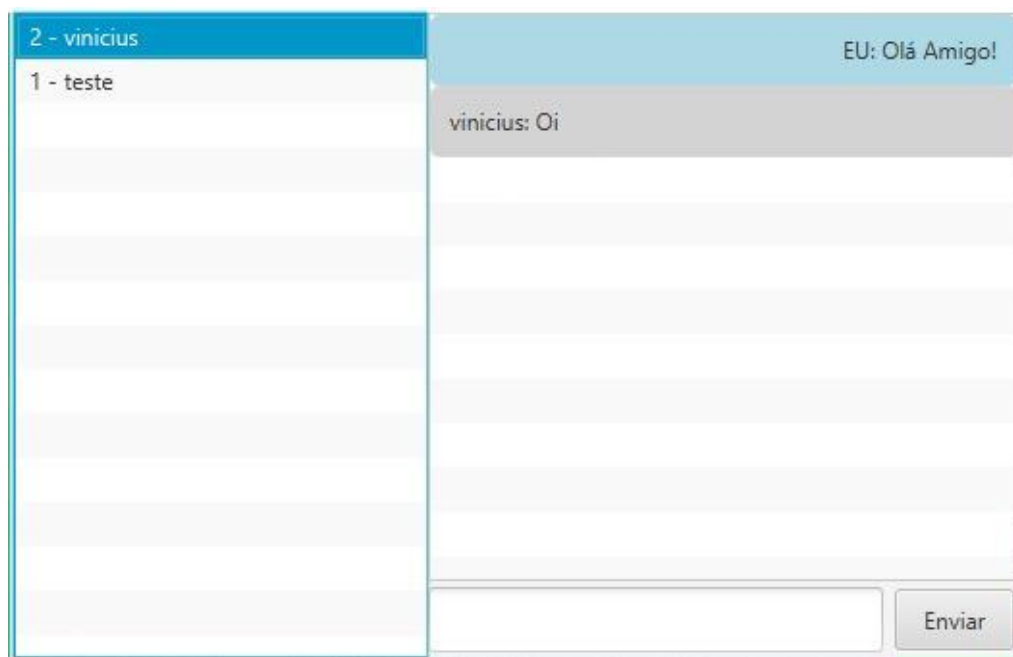
A imagem mostra a interface de login de uma aplicação web. O navegador exibe a aba "Login". A interface é dividida em duas seções principais: uma lateral esquerda com fundo verde e uma lateral direita com fundo cinza claro. Na seção verde, o texto "NatuFriends" é exibido em uma fonte grande e bold, com o subtítulo "Sua Rede Social de Aventureiros" abaixo dele. Na seção cinza, o título "Bem-Vindo" é seguido por "Acesse sua Conta". Abaixo disso, há dois campos de entrada: o primeiro para o nome de usuário (contendo o texto "Nome") e o segundo para a senha (contendo o texto "Senha"). Abaixo dos campos, há dois botões verdes: "Login" e "Cadastro".

Página de Cadastro: A página de cadastro permite que novos usuários se inscrevam na plataforma, fornecendo informações básicas.

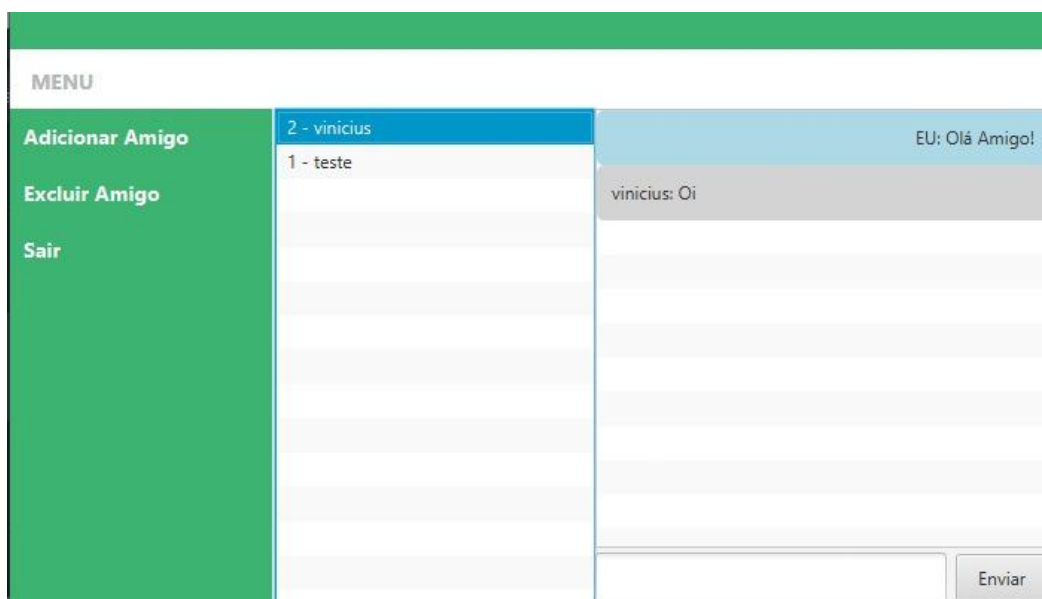


A imagem mostra a interface de cadastro de uma aplicação web. O navegador exibe a aba "Login". A interface possui uma barra de cabeçalho verde com o texto "NatuFriends" e "Rede Social para Aventureiros". Abaixo, o título "Cadastre um Novo Usuario" é exibido. Seguem-se três campos de entrada: o primeiro para o nome (contendo "Lucas"), o segundo para o e-mail (contendo "lucas@gmail.com") e o terceiro para a senha (contendo "•••"). Abaixo dos campos, há um botão verde com o texto "Cadastrar".

Página de Envio de Mensagem: A interface para envio de mensagens oferece uma experiência fácil e intuitiva para interação entre usuários.



Página do Menu: A página do menu é o ponto central da aplicação, proporcionando acesso à lista de amigos e à funcionalidade de envio de mensagens.



Conclusão

O desenvolvimento deste projeto proporcionou uma compreensão mais profunda de Java, JavaFX e integração com banco de dados MySQL. Enfrentamos desafios ao implementar funcionalidades como o histórico de mensagens, onde algumas dificuldades foram encontradas no processo de listar as mensagens trocadas entre os usuários.

Esses obstáculos destacam a importância contínua do aprendizado e aprimoramento das habilidades técnicas, especialmente em projetos práticos. Identificamos áreas de melhoria, como a expansão das funcionalidades para incluir o histórico de mensagens, grupos e suporte a multimídia.

O diagrama de classes e a estrutura do banco de dados fornecem uma base sólida para futuras iterações e expansão do projeto. Em resumo, este projeto serviu como um exercício prático, fornecendo insights sobre o desenvolvimento de aplicativos Java com interface gráfica e integração com banco de dados.