

**Trabalho Final de Linguagens de Programação**

##### IDENTIFICAÇÃO

|  |  |  |  |
| --- | --- | --- | --- |
| **RA** | **NOME** | **e-mail** | **Telefone** |
| **12120786** | **Samuel Henrique C.O**  **Silva** | [**Samuelhc07@icloud.com**](mailto:Samuelhc07@icloud.com) | **(31)99721-5215** |
| **12121360** | **Aline Barbosa da Silva dos Reis** | [**abarbosa09012000@gmail.com**](mailto:abarbosa09012000@gmail.com) | **(31)98485-4631** |

**ORIENTADOR(A):**

João Paulo Aramuni

Data da Entrega: 23 / 11 / 2023

**Introdução :**

O projeto foi desenvolvido para oferecer uma interface simples para interações de usuários em uma rede social simulada. Utilizando Java e o framework Swing em uma estrutura de dados baseada em um mapa para armazenar usuários e suas informações, a aplicação proporciona funcionalidades como login, cadastro de usuários, adição de amigos, envio de mensagens e listagem de amigos. A estrutura de dados principal utilizada foi um Map<String, Usuario> chamado usuarios, onde a chave é o email do usuário e o valor é uma instância da classe Usuario. Isso permite um número indefinido de entradas para os usuários.

**Desenvolvimento :**

Os principais métodos utilizados no projeto foram :

1. **Cadastrar Usuário (cadastrarUsuario()):** Adiciona um novo usuário ao Map de usuários.
2. **Login (fazerLogin()):** Solicita email e senha, simulando um processo de autenticação.
3. **Adicionar Amigo (adicionarAmigo()):** Permite ao usuário atual adicionar outro usuário como amigo.
4. **Enviar Mensagem (enviarMensagem()):** Permite ao usuário enviar mensagens para um amigo, se estiver logado.
5. **Listar Amigos (listarAmigos()):** Mostra os amigos do usuário logado (Quando adicionado).
6. **Logout (logout()):** Realiza o logout do usuário atual.

**Diagrama de Classes (UML):**

RedeSocial

Usuario

- usuarios: Map<String, Usuario> |

- usuarioAtual: Usuario

- frame: JFrame

- panel: JPanel

+ RedeSocialGUI()

- initializeUI()

- createPanel()

+ main(args: String[])

- nome: String

- email: String

- senha: String

- amigos: List<Usuario>

+ Usuario(nome: String, email: String, senha: String)

+ getNome(): String

+ getEmail(): String

+ getSenha(): String

+ getAmigos(): List<Usuario>

+ adicionarAmigo(amigo: Usuario):void

TelaInicial

- usuarios: Map<String, Usuario>

- usuarioAtual: Usuario

- frame: JFrame

- panel: JPanel

- redeSocial: RedeSocialGUI

+ TelaInicialGUI()

- initializeUI()

- createWelcomePanel()

- createPanel()

+ main(args: String[])

**Código fonte :**

// Exemplo do método de login

public void fazerLogin() {  
 System.*out*.println("Digite o e-mail do usuário:");  
 String email = scanner.nextLine();  
 System.*out*.println("Digite a senha do usuário:");  
 String senha = scanner.nextLine();  
  
 Usuario usuario = usuarios.get(email);  
 if (usuario != null && usuario.getSenha().equals(senha)) {  
 usuarioAtual = usuario;  
 System.*out*.println("Login realizado com sucesso!");  
 } else {  
 System.*out*.println("E-mail ou senha incorretos!");  
 }  
}

// Exemplo do método de cadastro do usuário

public void cadastrarUsuario() {  
 System.*out*.println("Digite o nome do usuário:");  
 String nome = scanner.nextLine();  
 System.*out*.println("Digite o e-mail do usuário:");  
 String email = scanner.nextLine();  
 System.*out*.println("Digite a senha do usuário:");  
 String senha = scanner.nextLine();  
  
 Usuario usuario = new Usuario(nome, email, senha);  
 usuarios.put(email, usuario);  
  
 System.*out*.println("Usuário cadastrado com sucesso!");  
}

// os demais métodos seguem a lógica semelhante

**Interface Gráfica:**

A interface gráfica foi feita utilizando a biblioteca Swing, trazendo botões para as funcionalidades principais do projeto.

**Conclusão :**

Esse é um projeto básico que pode funcionar como base para uma Rede Social simples. Para uma aplicação que seja real, será necessário uma implementação de melhorias da interface e uma segurança maior para as informações do usuário.

A principal dificuldade do projeto foi a integração com o banco de dados.