

##### Linguagem de programação

**MINI SIMULADOR DE REDE SOCIAL**

##### 

| **RA** | **NOME** | **E-MAIL** |
| --- | --- | --- |
| **12118181** | **Edson De Salles Junior** | **junior111edson@outlook.com** |
| **12119065** | **Pedro Henrique** | **nunesp665@outlook.com.br** |
| **12118907** | **Fernanda Rocha da Silva** | **contato.fernandaroc@gmail.com** |
| **12119217** | **Rafael Marques** | **rafaelmarques1590@gmail.com** |

**PROFESSOR(A):**

João Paulo Carneiro Aramuni

Michelle Hanne



**1 INTRODUÇÃO**

O projeto consiste no desenvolvimento de um Mini Simulador de Rede Social, proporcionando o cadastro de usuários, gerenciamento de contatos e envio de mensagens. Iniciamos o desenvolvimento em 05/11/23, seguindo as diretrizes fornecidas pelos professores João Paulo Aramuni e Michelle Hanne.

**2 ESTRUTURA DE DADOS**

Optamos por utilizar estruturas de dados dinâmicas, para acomodar um número indefinido de entradas. A classe “usuario” representa cada usuário, contendo atributos como nome, e-mail, senha, e uma lista dinâmica de contatos.

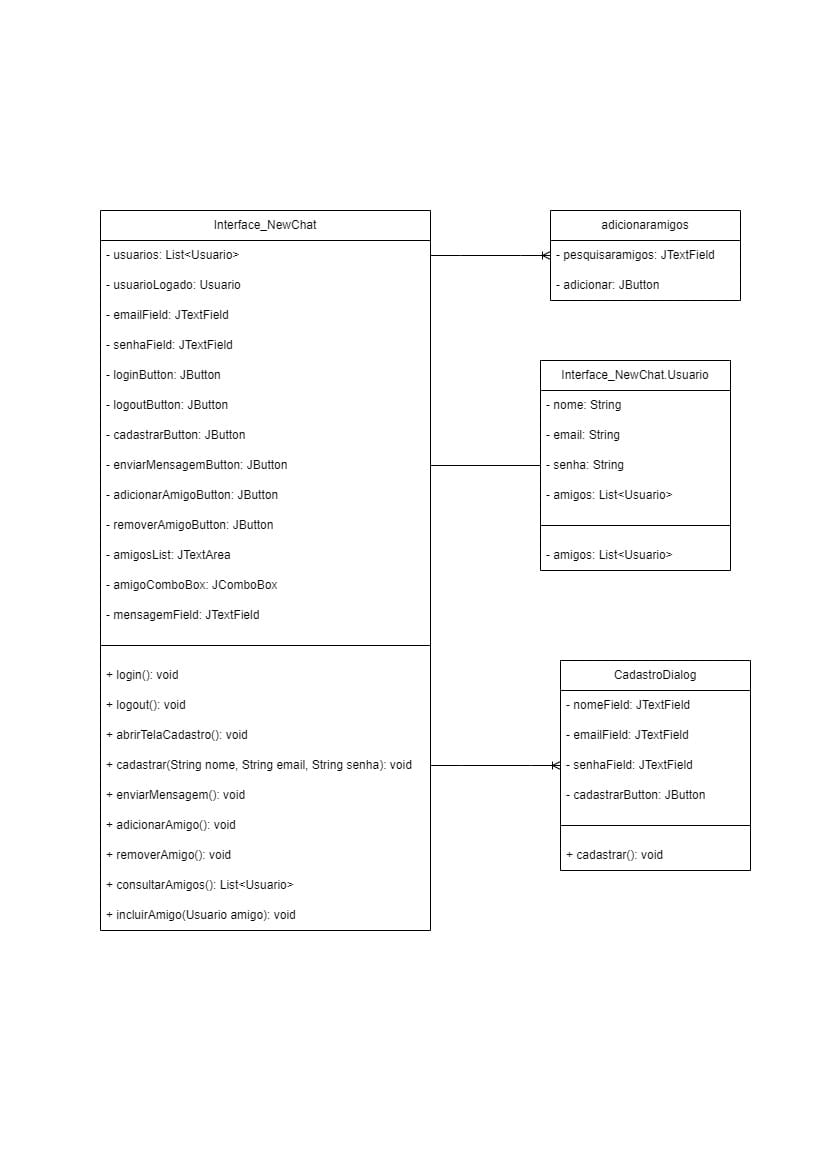
**3 DESENVOLVIMENTO**

Foram utilizadas a classe abstrata ”Pessoa”, que é herdada por usuário, garantindo a reutilização de código. A interface Autenticavel foi implementada para gerenciar a autenticação. O polimorfismo foi utilizado nas operações de adicionar e remover amigos, permitindo flexibilidade com diferentes tipos de usuários. Além disso, foi aplicado o padrão Singleton para a classe RedeSocial, com o objetivo de instanciar o sistema em execução apenas uma única vez.



**4 UML- DIAGRAMA DE CLASSE**

O diagrama de classe abaixo ilustra a estrutura do projeto, destacando as principais classes e relacionamentos.





**5 BANCO DE DADOS**

Foi utilizado o banco de dados PostgreSQLSQL para armazenar informações sobre usuários e suas relações de amizade. Configuramos a conexão utilizando JDBC.

**6 INTERFACE GRÁFICA**

Desenvolvemos uma interface simples, utilizando bibliotecas gráficas em Java, incluindo páginas para login, cadastro, feed de notícias, mensagens, e gerenciamento de amigos. Utilizamos CSS para estilizar e a linguagem JavaScript para interatividade.

**7 CONCLUSÃO**

Destacamos a importância do uso de padrões de projeto para manter o código modular e de fácil manutenção. Este relatório resume o trabalho realizado na implementação do Mini Simulador de Rede Social. O projeto proporcionou uma compreensão aprofundada de conceitos fundamentais de programação em Java, desenvolvimento web, e integração de frontend e backend.

**8 DIFICULDADES E CONSIDERAÇÕES**

Enfrentamos desafios durante o desenvolvimento, principalmente na integração entre frontend e backend, na fase inicial da configuração do ambiente, e na implementação com o banco de dados, mas conseguimos superar os obstáculos com colaboração e pesquisa. As funcionalidades de adicionar e remover contatos foram implementadas com sucesso. O trabalho em grupo permitiu uma distribuição eficiente das tarefas, resultando em uma implementação coesa.



**9 INSTRUÇÕES DE USO**

Para executar a aplicação, é necessário seguir os seguintes passos:

Configurar o ambiente de desenvolvimento com JDK, IDE Java e servidor web (Apache Tomcat).

Criar um novo projeto Java no IDE.

Dividir o projeto em pacotes e configurar o sistema de construção (Maven ou Gradle).

Escolher e configurar o banco de dados (MySQL ou PostgreSQL) utilizando JDBC ou Hibernate.

Implementar as classes Java para representar usuários, amigos e mensagens.

Desenvolver as funcionalidades de registro, autenticação, gerenciamento de amigos e envio de mensagens.

Utilizar servlets ou frameworks web (como Spring MVC) para manipular solicitações HTTP.

Criar páginas HTML para a interface do usuário, estilizando-as com CSS e adicionando interatividade com JavaScript.

Realizar chamadas assíncronas entre o frontend e o backend para enviar dados e receber atualizações.

Realizar testes unitários para as principais funcionalidades e testar a integração entre frontend e backend.

Implantar a aplicação em um servidor web, como o Apache Tomcat.

Implementar medidas de segurança, como hashing de senhas e prevenção contra injeções SQL.

**10 REFERÊNCIAS**

Foram utilizadas as seguintes fontes e referências para realização do trabalho:

> OPENAI. ChatGPT. Disponível em: <https://chat.openai.com>. Acesso em 5 de novembro de 2023;

> ORACLE. Documentação Oracle. Disponível em: <https://docs.oracle.com>. Acesso em 5 de novembro de 2023;

> ARAMUNI, João Paulo. Repositório no GitHub. Disponível em: <https://github.com/joaopauloaramuni>. Acessado em 18 de novembro de 2023;

> MIHANNE. Repositório no GitHub. Disponível em: <https://github.com/mihanne>. Acesso em 18 de novembro de 2023.