



**Programação Orientada a Objetos
(POO)**

IDENTIFICAÇÃO

RA	Nome
12115631	Arthur Ângelo de Souza Santos
1211622	Edson Ricardo Buzetti Gonçalves Junior
12116358	Eduardo Medeiros Martins
12117039	Felipe Guilherme de Jesus Lopes
12116208	Nicolas Kyan Souza Chi

TÍTULO: Trabalho Rede Social (NewtonGran)

LÍDER DO GRUPO: Edson Ricardo Buzetti Gonçalves Junior

ORIENTADOR(A): Michelle Hanne Soares De Andrade

DATA DA ENTREGA: 13/06/2023

Visto do(a) Orientador(a)

Programação Orientada a Objetos

Arthur Ângelo de Souza Santos

Edson Ricardo Buzetti Goncalves Junior

Eduardo Medeiros Martins

Felipe Guilherme de Jesus Lopes

Nicolas Kyan Souza Chi

Rede Social (NewtonGran)

**Belo Horizonte
2023**

1 Introdução

A solução implementada é uma Rede Social chamada NewtonGran, onde os usuários podem se cadastrar, fazer login, adicionar amigos, enviar mensagens e visualizar as mensagens recebidas.

A estrutura de dados utilizada é composta por três classes principais: Usuário, Mensagem e NewtonGran.

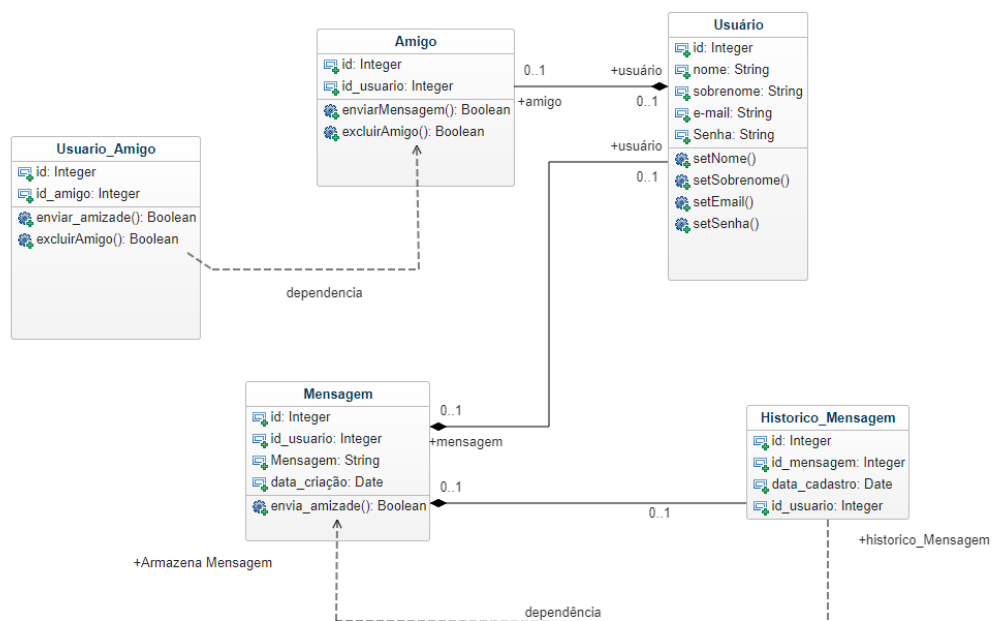
A classe Usuário representa um usuário da rede social e possui os atributos nome, e-mail, senha, amigos e mensagens. Os amigos são armazenados em uma lista do tipo Usuário, permitindo que cada usuário tenha uma lista de amigos. As mensagens também são armazenadas em uma lista do tipo Mensagem, que contém informações sobre o remetente e o conteúdo da mensagem.

A classe Mensagem representa uma mensagem enviada entre usuários da rede social. Ela possui os atributos remetente (do tipo Usuário) e conteúdo (do tipo String), que armazenam as informações relacionadas à mensagem.

A classe NewtonGran é a classe principal que contém o método main e controla o programa. Nela, é exibido um menu principal com opções para cadastrar usuário, fazer login ou sair do sistema. Dependendo da opção selecionada, o programa chama os métodos correspondentes para realizar as operações desejadas.

A estrutura de dados utilizada é adequada para armazenar um número indefinido de usuários, amigos e mensagens, pois utiliza listas para isso.

Quanto ao diagrama de classe UML, segue abaixo uma representação simplificada das classes e seus relacionamentos:



2 Desenvolvimento

A classe `Usuario` é responsável por representar um usuário na rede social e possui atributos que armazenam informações essenciais do usuário, como nome, e-mail, senha, lista de amigos e lista de mensagens recebidas.

A classe `Usuario` tem o papel crucial na estrutura da rede social, pois ela permite a criação e manipulação dos perfis dos usuários, incluindo o gerenciamento de amigos e mensagens, nela também é fornecido os métodos para adicionar e remover amigos enviar e receber mensagens.

A classe `Mensagem` é responsável por representar uma mensagem enviada por um remetente em uma rede social. Ela possui dois atributos principais: `remetente` e `conteudo`.

A classe `Mensagem` tem a finalidade de encapsular as informações relacionadas a uma mensagem específica, permitindo o acesso ao remetente e ao conteúdo da mensagem. A estrutura facilita o processo de envio, recebimento e visualização das mensagens na rede social.

A classe `NewtonGran` é a classe principal do programa, responsável por executar a lógica do sistema NewtonGran, uma espécie de rede social simplificada. Essa classe contém o método `main` que inicia a execução do programa.

A classe `NewtonGran` utiliza os recursos da classe `Usuario` para realizar as funcionalidades do sistema, para cadastrar usuários, fazer login, adicionar amigos, enviar mensagens e visualizar mensagens recebidas.

Esse programa demonstra o uso de estruturas de controle, como loops e condicionais, para criar um sistema básico de rede social. Ele utiliza a biblioteca `javax.swing.JOptionPane` para exibir mensagens e obter entradas do usuário por meio de caixas de diálogo.

3 Considerações finais

Implementar o NewtonGran em Java, apresentaram algumas dificuldades, no desenvolvimento da UML, tivemos alguma dificuldade de associar a estrutura de códigos a um banco de dados que não foi criado, também tivemos dificuldades em fazer o relacionamento de adicionar amigos.

Ao final do projeto, aprendemos com os erros encontrados ao longo dele e a solucioná-los de forma eficiente, através desse trabalho foi possível colocar em prática todos os conceitos e técnicas do paradigma de orientação a objetos, dessa forma ficou claro o funcionamento desse padrão utilizado.