**Isabella Alexandra Rojas Guzmàn**

**Isabella Santos Veras**

**Gustavo Jorge Batista de Oliveira**

**Lucas kayke Sousa Abreu**

**Projeto: Rede Social**

Projeto de conclusão do semestre apresentado ao Centro Universitário Newton Paiva, como exigência parcial para a disciplina de Programação Orientada à Objetos (POO).

Prof(a): Michelle Hanne

**Belo Horizonte**

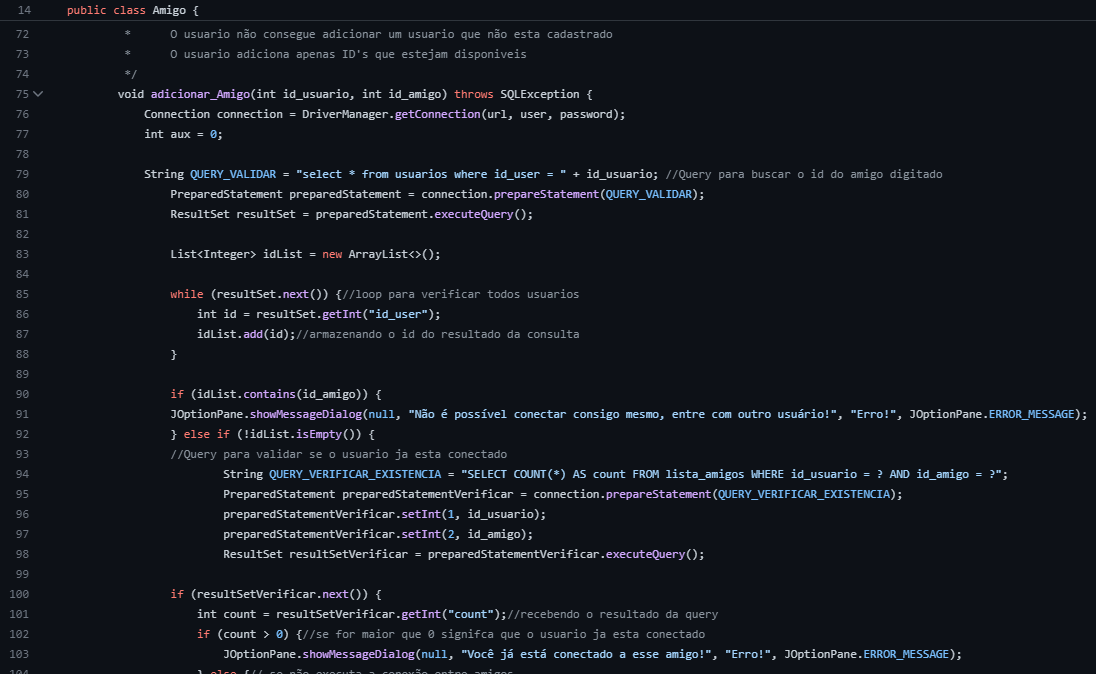
**2023**

**Introdução**

Mediante aos conhecimentos fornecidos durante as aulas de programação orientada a objetos. Os conteúdos transmitidos foram aplicados no desenvolvimento de uma rede social, que permite que o usuário realize “amizades”. O primeiro passo do projeto é criação do diagrama de classes, o UML, a partir dele iniciou-se o desenvolvimento do projeto.

**Desenvolvimento**

Um dos principais focos do projeto é permitir que o usuário consiga fazer um número indefinido de amizades, para isso usou o ArrayList.O método “adicionar\_Amigo” do tipo void,ele faz uma conexão no banco de dados, valida se o usuário logado possui uma conexão com o usuário que deseja adicionar e se esse usuário existe. Caso contrário realiza a conexão.





Aplicar o CRUD no desenvolvimento foi essencial, porque na rede social deve conter o cadastro, consultar, editar e inserir.

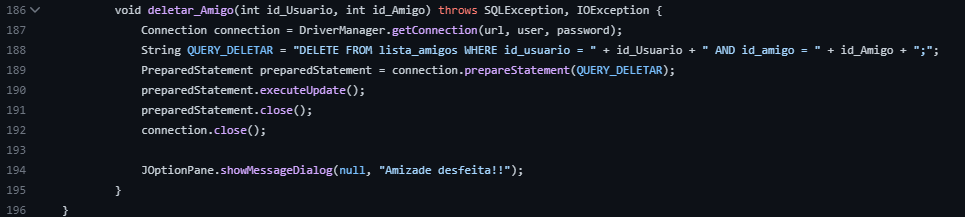
O método cadastrar\_Usuário consiste em obter os dados do usuário e inseri-los no banco de dados.



O método adicionar\_Amigo do tipo void, esse método irá buscar todos os usuários que começam com os caracteres digitados na barra de pesquisa.



O método deletar\_Amigo do tipo void, é um método para desfazer a amizade, ele recebe o ID do usuário logado e do amigo que deseja deletar. Para desfazer a amizade é necessário informar o ID.

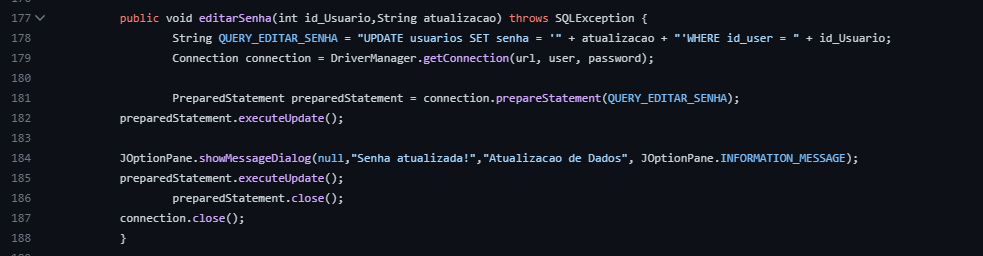


No código existem 3 métodos para editar o cadastro do usuário, dentre eles estão:

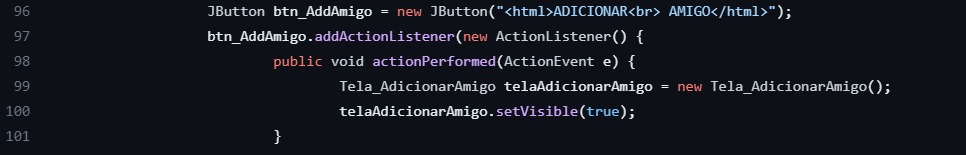
* método void editarNome
* método void editarEmail
* método void editarSenha

Ambos possuem a função de alterar os dados cadastrais do usuário e atualizar no banco de dados.





No projeto temos a classe Usuário que herda os atributos da classe Perfil. Da mesma forma a classe Tela\_Amigos herda da classe Tela\_AdicionarAmigo, pois Tela\_AdicionarAmigo é instanciado em um método void actionPerformed(ActionEvent e) o que também indica polimorfismo.



Outro exemplo de polimorfismo é com o método obterPerfilUsuario que se encontra na classe Perfil, que obtêm os dados do usuário. Ele vem sendo utilizado também na classe Usuário e Tela\_EditarPerfil.

Classe Perfil



Classe Usuário



Classe Tela\_EditarPerfil



**Conclusão**

O projeto teve algumas dificuldades no desenvolvimento. Como, por exemplo, na etapa de obter o ID no usuário que se encontra logado para realizar as ações dentro do perfil dele.

Além disso, teve desafios no tratamento dos erros que permitem adicionar uma amizade e seguir as condições necessárias para realizar essa ação. Pois foi importante validar e verificar as condições exigidas antes de adicionar uma amizade, assim como verificar se os usuários já são amigos.

Também havia surgido um bug na lista de amigos, pois só estava aparecendo os amigos que o usuário que está logado adicionou, mas quando outro usuário adicionava o usuário anterior ele não era apresentado. Esse bug foi solucionado alterando o script.

Mas mediante a resolução dessas dificuldades, o projeto consegue executar suas funcionalidades básicas e armazenar os dados no Postgre.