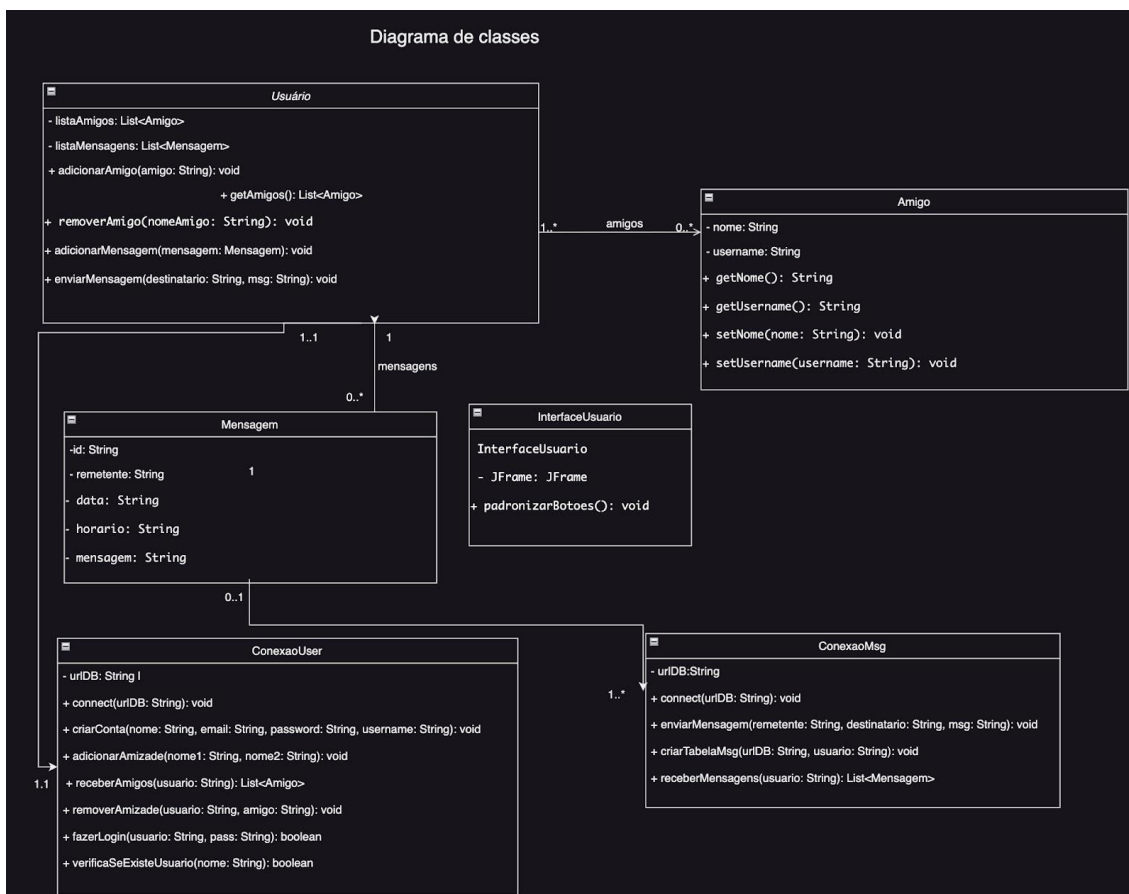


Simulador de Rede Social [ESN]

Introdução

O desenvolvimento do simulador de Rede Social baseia-se nos conceitos aprendidos ao longo do semestre sobre a criação de classes Java e seus métodos. Os dados são armazenados em um Banco de Dados, contemplando informações de usuários, login, cadastro, contatos adicionados e mensagens enviadas e recebidas.

A descrição detalhada do projeto é apresentada utilizando o diagrama UML (*Unified Modeling Language*).



DESCRIÇÃO DO CÓDIGO FONTE

Classe Amigo:

- ✓ Métodos Principais:
 - `getNome()`: Retorna o nome do amigo.
 - `getUsername()`: Retorna o username do amigo.
 - `setNome(String nome)`: Define o nome do amigo.
 - `setUsername(String username)`: Define o username do amigo.

Esta classe não utiliza herança ou polimorfismo, sendo composta principalmente por métodos de acesso e modificação.

Classe Mensagem:

- ✓ Métodos Principais:
 - `getId()`: Retorna o ID da mensagem.
 - Métodos de acesso e modificação para remetente, data, horário e mensagem.
 - Construtor que inicializa os atributos da mensagem.

Assim como a classe Amigo, a classe Mensagem não apresenta herança ou polimorfismo evidentes.

Classe Usuario:

- ✓ Métodos Principais:
 - `adicionarAmigo(String amigo)`: Adiciona um amigo à lista de amigos.
 - `getAmigos()`: Retorna a lista de amigos.
 - `removerAmigo(String nomeAmigo)`: Remove um amigo da lista de amigos.
 - `adicionarMensagem(Mensagem mensagem)`: Adiciona uma mensagem à lista de mensagens.
 - `enviarMensagem(String destinatario, String msg)`: Envia uma mensagem para um destinatário, verificando se são amigos antes de enviar.

Esta classe não faz uso de herança ou polimorfismo evidentes, sendo predominante composta por métodos de acesso e modificação.

Classe ConexaoMsg:

- ✓ Métodos Principais:
 - `connect (String urlDB)`: Estabelece uma conexão com o banco de dados PostgreSQL usando JDBC.
 - `enviarMensagem (String remetente, String destinatario, String msg)`: Conecta ao banco de dados e insere uma nova mensagem na tabela do destinatário.
 - `criarTabelaMsg (String urlDB, String usuario)`: Cria uma tabela para armazenar mensagens de um usuário no banco de dados.
 - `receberMensagens (String usuario)`: Conecta ao banco de dados e recupera todas as mensagens do usuário.

- Outros métodos específicos para operações relacionadas a mensagens.

Não há herança ou polimorfismo nesta classe.

Classe ConexaoUser:

- ✓ Métodos Principais:
 - connect(String urlDB): Estabelece uma conexão com o banco de dados PostgreSQL usando JDBC.
 - Outros métodos específicos para operações relacionadas a usuários, como criação de conta, adição/remoção de amizades, login, entre outros.

Não há herança ou polimorfismo evidentes nesta classe.

Observações Gerais:

Ambas as classes ConexaoMsg e ConexaoUser são responsáveis por estabelecer conexões com o banco de dados PostgreSQL e executar operações relacionadas a mensagens e usuários, utilizando JDBC para interagir com o banco de dados.

InterfaceLogin:

- ✓ Herança: A classe InterfaceLogin também herda da classe JFrame.
- ✓ Polimorfismo: Não há polimorfismo evidente.
- ✓ Métodos Principais:
 - realizarLogin(): Realiza a autenticação do usuário, verificando no banco de dados se as credenciais são válidas.
 - Outros métodos relacionados à interface de login.



ESN

Username: Senha:

InterfaceCriarConta:

- ✓ Herança: A classe InterfaceCriarConta não herda explicitamente de nenhuma outra classe, mas possui uma referência à classe JFrame.
- ✓ Polimorfismo: Não há polimorfismo visível.
- ✓ Métodos Principais:
 - criarConta(): Cria uma nova conta no banco de dados com base nas informações fornecidas.
 - Outros métodos relacionados à interface de criação de conta.

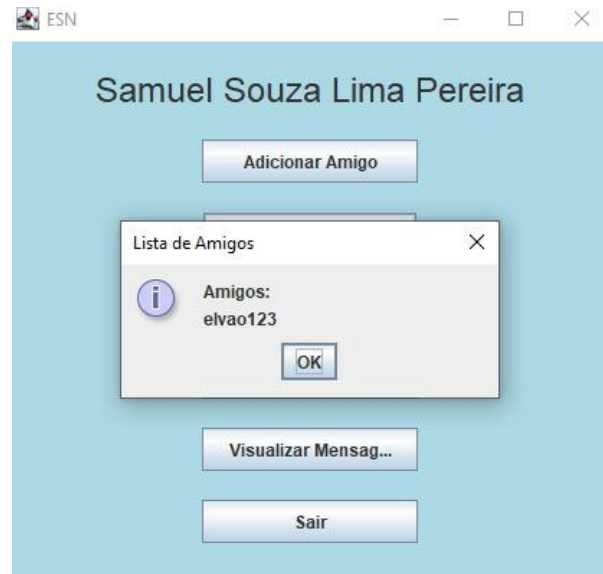


InterfaceUsuario:

- ✓ Herança: A classe InterfaceUsuario herda da classe JFrame, uma classe gráfica do Java Swing.
- ✓ Polimorfismo: Não há polimorfismo evidente na classe InterfaceUsuario.
- ✓ Métodos Principais:
 - padronizarBotoes(): Define o layout e as ações dos botões na interface.
 - Outros métodos relacionados à interface do usuário.



DEMAIS INTERFACES:

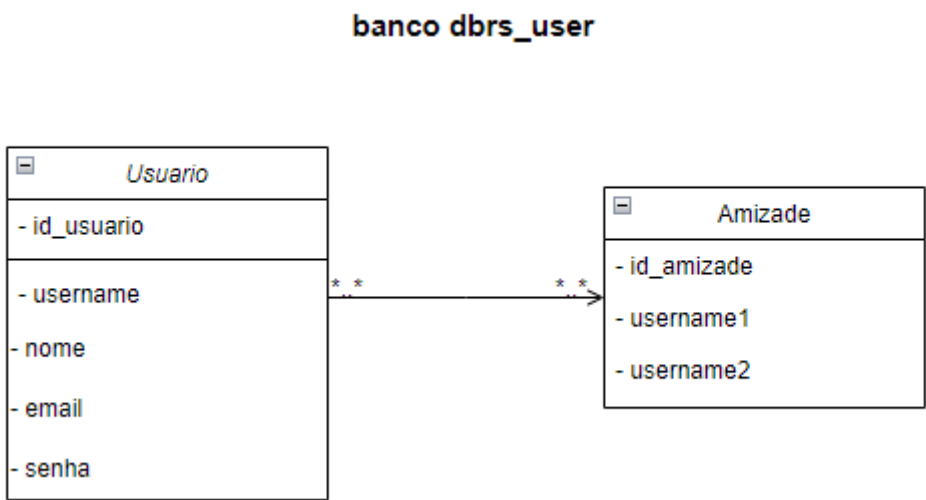


BANCO DE DADOS:

Banco de Dados DBRS_USER:

O DBRS_USER é fundamental para o armazenamento e gerenciamento dos dados relacionados aos usuários da nossa rede social. Este banco de dados contém duas tabelas principais:

- a. Tabela "usuario": Responsável por armazenar informações cruciais dos usuários, como nome, ID, e outros dados pessoais. Essa tabela é estruturada para garantir a integridade dos dados e otimizar o acesso às informações do usuário.
- b. Tabela "amizade": Estabelece as relações entre os usuários, indicando quem são amigos na rede social.




Banco de Dados DBRS_MSG:

O DBRS_MSG é dedicado ao armazenamento de mensagens recebidas pelos usuários. A abordagem adotada para esse banco de dados visa otimizar o acesso e a recuperação de mensagens específicas para cada usuário. A estrutura envolve:

- a. Criação Dinâmica de Tabelas: Logo após o usuário criar uma conta na rede social, o sistema automaticamente cria uma tabela dedicada a esse usuário dentro do DBRS_MSG. Isso permite a segmentação eficiente das mensagens, garantindo um acesso rápido e direto às informações pertinentes a cada usuário.

DBRS_USER

	<i>Username</i>
id_msg: int PK	
mensagem text	
dia: VARCHAR(20)	
hora: VARCHAR(12)	

HERANÇA E POLIMORFISMO:

O conceito de herança é aplicado nas classes de interface (InterfaceCriarConta, InterfaceLogin, InterfaceUsuario), proporcionando o reuso de código e contribuindo para a organização e manutenção do projeto.

O polimorfismo, que se refere à capacidade de um objeto assumir várias formas, não é claramente evidenciado nas classes de interface.

CONCLUSÃO:

A implementação das classes enfrentou desafios durante a modelagem, conexão e implementação da lógica, especialmente em relação ao banco de dados, uma novidade para o grupo. Tivemos algumas dificuldades na interface no quesito de padronização do layout envolvendo formatos e simetria dos campos.

As dificuldades foram superadas, e a resolução dos problemas de conexão facilitou a implementação da lógica, da interface e do banco. O simulador de Rede Social representa um projeto bem estruturado, aplicando conceitos aprendidos ao longo do semestre.