Introdução

O código apresentado é uma solução para a simulação de uma rede social simples. A principal estrutura de dados utilizada é a lista (ArrayList), que permite armazenar um número indefinido de entradas. Essa estrutura é usada para armazenar usuários em nossa "RedeSocial" e para armazenar amigos e mensagens para cada usuário individual.

Diagrama de Classe

A estrutura do programa consiste em quatro classes: Pessoa, Usuário, Amigo e RedeSocial.

```
Pessoa
---------
- nome: String
- email: String
---------
+ Pessoa(nome: String, email: String)
+ getNome(): String
+ getEmail(): String

Usuario extends Pessoa
---------
- senha: String
- amigos: List<Amigo>
- mensagens: List<String>
---------
+ Usuario(nome: String, email: String, senha: String)
+ getSenha(): String
+ getAmigos(): List<Amigo>
+ getMensagens(): List<String>
+ adicionarAmigo(amigo: Amigo)
+ removerAmigo(amigo: Amigo)
+ enviarMensagem(amigo: Amigo, mensagem: String)
+ imprimirAmigos()
+ imprimirMensagens()

Amigo extends Pessoa
---------
+ Amigo(nome: String, email: String)

RedeSocial
---------
- usuarios: List<Usuario>
- usuarioLogado: Usuario
---------
+ RedeSocial()
+ cadastrarUsuario(nome: String, email: String, senha: String)
+ login(email: String, senha: String)
+ logout()
+ incluirAmigo(email: String)
+ consultarAmigos()
+ excluirAmigo(email: String)
+ enviarMensagem(email: String, mensagem: String)
+ imprimirMensagens()
```

Desenvolvimento

As classes ***Usuario*** e ***Amigo*** são subclasses da classe ***Pessoa***, o que é um exemplo de Herança. Ambas as subclasses herdam os campos ***nome*** e ***email*** da classe ***Pessoa***.

A classe ***Usuario*** representa um usuário registrado na rede social. Esta classe inclui métodos para adicionar e remover amigos, enviar mensagens e imprimir amigos e mensagens.

A classe ***Amigo*** é uma representação simplificada de um ***Usuario***, usada para manter a lista de amigos de um usuário. Esta classe é quase idêntica à classe ***Usuario***, exceto que ela não possui uma lista de amigos ou mensagens.

A classe RedeSocial é o ponto central do nosso programa, contendo uma lista de usuários e o usuário atualmente logado. Ela inclui métodos para cadastrar usuários, fazer login e logout, adicionar e remover amigos, enviar mensagens e imprimir mensagens.

Conclusão

A principal dificuldade ao criar esse programa foi gerenciar o estado do usuário atualmente logado. No entanto, este desafio foi superado ao adicionar um campo ***usuarioLogado*** à classe ***RedeSocial***, que mantém o controle do usuário logado atualmente.

Embora o programa funcione como esperado para a tarefa proposta, ele é bastante simplista e possui muitas limitações. Por exemplo, ele não possui recursos para lidar com situações em que o email de um usuário já está em uso, nem permite que um usuário mude sua senha. Além disso, o programa poderia ser melhorado para permitir a interação entre diferentes usuários, como o envio de mensagens entre eles.

No entanto, como uma simulação básica de uma rede social, acredito que o programa seja bem-sucedido. Ele ilustra claramente os conceitos de classes, herança, listas e loops, e fornece um ponto de partida sólido para expansões ou melhorias futuras.

Integrantes:

1. Alexander Max Dutra

2. Breno Italo gomes pereira

3. Igor César Alves Ribeiro

4. João Pedro Alves Antunes

5. Ramon Alves Gomes

6. Vinícius Andrade Nascimento