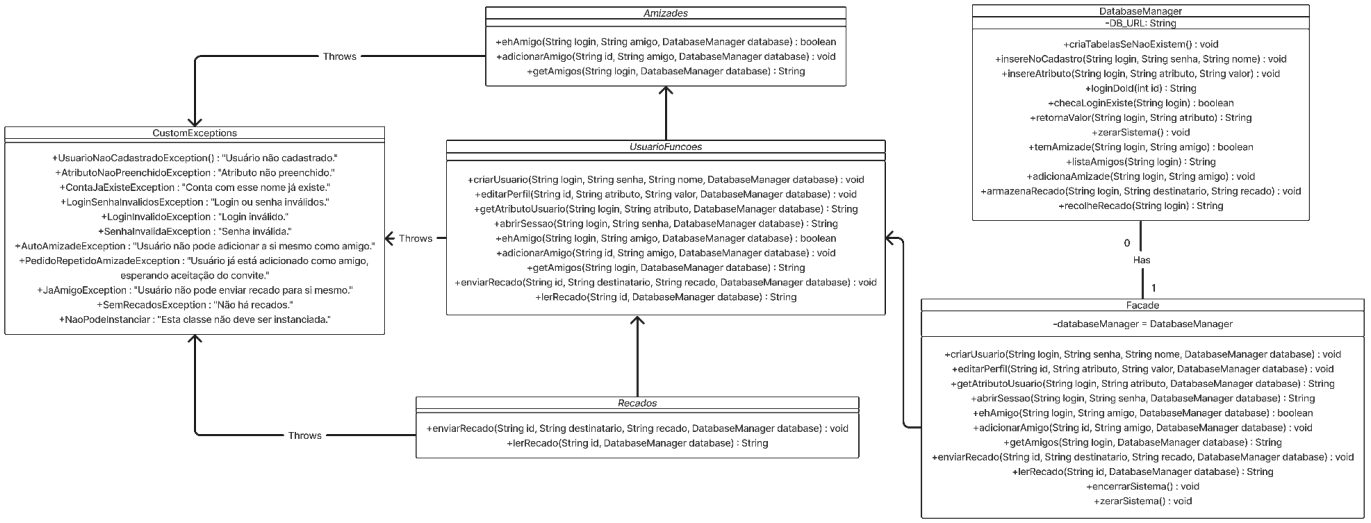
Relatório de P2

Aluno: Gustavo Domingos de Oliveira
Matrícula: 21211834

Diagrama de Classes



Descrição das Classes

Facade: Classe que tem a método de servir de “fachada” para os testes.
UsuarioFuncoes: Classe abstrata que contém todas as métodos vinculadas aos usuário.
Recados: Classe abstrata que contém os métodos de receber e enviar recados.
Amizades: Classe abstrata que contém os métodos de fazer e checar amizades.
DatabaseManager: Classe que contém todos os métodos que interagem com o banco de dados.
CustomExceptions: Classe que contém todas as exceções personalizadas do programa.

Métodos em Facade

Tdoso os métodos de Facade, com exceção de zerarSistema() e encerrarSistema(), apenas chamam os métodos de UsuarioFuncoes, que, para as funcionalidades de amizades e recados, chamam métodos de Amizades e Recados, respectivamente.

void criarUsuario(String login, String senha, String nome, DatabaseManager database)

Cria um registro no banco de dados com o login, senha e nome do usuário, como também um id único.

void editarPerfil(String id, String atributo, String valor, DatabaseManager database)

Este método permite que o usuário possa mudar algum atributo no banco de dados.

String getAtributoUsuario(String login, String atributo, DatabaseManager database)

Este método retorna algum atributo do usuário, decidi utilizar um id único por usuário que é gerado e armazenado automaticamente quando o usuário é criado.

String abrirSessao(String login, String senha, DatabaseManager database)

Este método retorna o id do usuário.

boolean ehAmigo(String login, String amigo, DatabaseManager database)

Este método retorna verdadeiro caso o usuário tenha relação de amigo com outro no banco de dados.

void adicionarAmigo(String id, String amigo, DatabaseManager database)

Adiciona um usuário com outro no banco de dados. Neste caso um pedido de amizade se refere a um registro no banco de dados do tipo:
login: login1 amigo: login2
Já uma amizade entre os dois é armazenada no banco de dados da seguinte forma:
login: login1 amigo: login2
login: login2 amigo: login1

String getAmigos(String login, DatabaseManager database)

Retorna a lista de amigos do banco de dados que o usuário tem.

void enviarRecado(String id, String destinatario, String recado, DatabaseManager database)

Envia um recado do um usuário para o usuário destinatário e o armazena no banco de dados.

String lerRecado(String id, DatabaseManager database)

Recolhe do banco de dados a primeira mensagem vinculada ao usuário no banco de dados.

void encerrarSistema()

Atualmente, esse método não faz nada, como o meu sistema armazena toda informação de forma dinâmica no banco de dados. Então, não existe nada que o sistema precise fazer no encerramento.

void zerarSistema()

Esse método deleta os registros em todas as tabelas do banco de dados.

Métodos de DatabaseManager

void criaTabelasSeNaoExistem()

Caso as tabelas não existam no banco de dados elas são criadas.

void insereNoCadastro(String login, String senha, String nome)

Insere um usuário no cadastro.

void insereAtributo(String login, String atributo, String valor)

Insere algum atributo no registro do usuário.

String loginDoId(int id)

Retorna o login a partir do id.

boolean checaLoginExiste(String login)

Retorna true se o login existe no registro do banco de dados.

String retornaValor(String login, String atributo)

Retorna um determinado valor vinculado a um login no banco de dados.

void zerarSistema()

Limpa os registros nas tabelas do banco de dados.

boolean temAmizade(String login, String amigo)

Checa na tabela de amizades se o usuário1 tem amizade com usuário 2.

String listaAmigos(String login)

Retorna uma string com todos os amigos de login.

void adicionaAmizade(String login, String amigo)

Adiciona uma relação de amizade no banco de dados.

void armazenaRecado(String login, String destinatario, String recado)

Armazena um recado no banco de dados.

String recolheRecado(String login)

Recolhe o primeiro recado em que o destinatário é o login.

Exceções

Além das exceções base do SQL Lite, temos as seguintes exceções personalizadas junto com a mensagem que será imprimida caso esse erro ocorra, todas definidas na classe “CustomException”.

UsuarioNaoCadastradoException()

"Usuário não cadastrado."

AtributoNaoPreenchidoException()

"Atributo não preenchido."

ContaJaExisteException()

"Conta com esse nome já existe."

LoginSenhaInvalidosException()

"Login ou senha inválidos."

LoginInvalidoException()

"Login inválido."

SenhaInvalidaException()

"Senha inválida."

AutoAmizadeException()

"Usuário não pode adicionar a si mesmo como amigo."

PedidoRepetidoAmizadeException()

"Usuário já está adicionado como amigo, esperando aceitação do convite."

JaAmigoException()

"Usuário não pode enviar recado para si mesmo."

SemRecadosException()

"Não há recados."

NaoPodeInstanciar()

"Esta classe não deve ser instanciada."