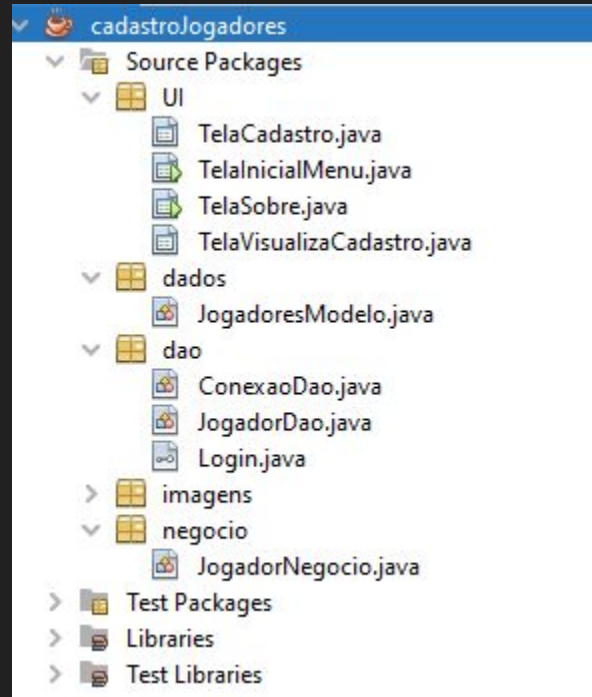


Sistema para cadastro de jogadores

Gustavo Manfio

Introdução

O presente sistema visa realizar o cadastro de jogadores em um banco de dados. A divisão do presente trabalho será organizada a fim de responder às questões propostas no documento.



Questão 1. Interface

A interface (Login) foi utilizada para gerenciar o acesso ao MYSQL.

```
package dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 *
 * @author Gustavo
 */
public interface Login {
    public Connection getConexao();
}
```

```
/**
 */
public class ConexaoDao implements Login{

    static final String DATA_BASE_URL = "jdbc:mysql://localhost/jogadores";

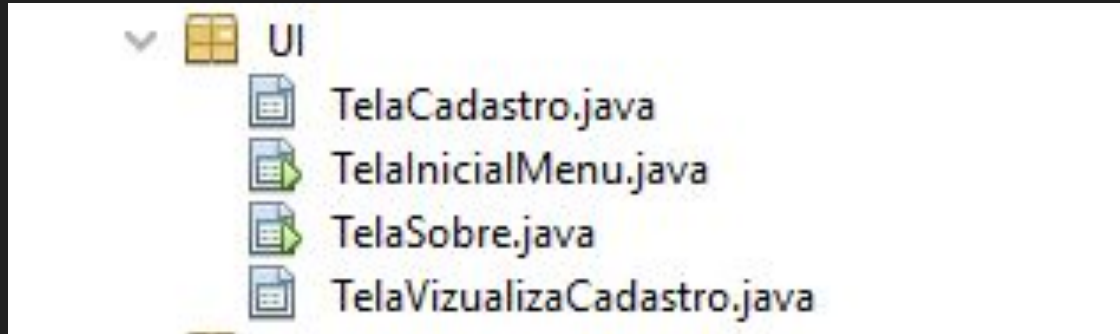
    static final String USERNAME = "root";

    static final String PASSWORD = "";
    Connection connection = null;
    //Statement statement = null;
    //ResultSet resultSet = null;


    @Override
    public Connection getConexao(){
        try {
            connection = DriverManager.getConnection(DATA_BASE_URL, USERNAME, PASSWORD);
        } catch (SQLException ex) {
            Logger.getLogger(ConexaoDao.class.getName()).log(Level.SEVERE, null, ex);
        }
        System.out.println("Conexão estabelecida");
        return connection;
    }
}
```

Questão 2. Ui




A camada UI foram implementados TelaCadastro, TelaInicialMenu, TelaSobre e TelaVizualizaCadastro.



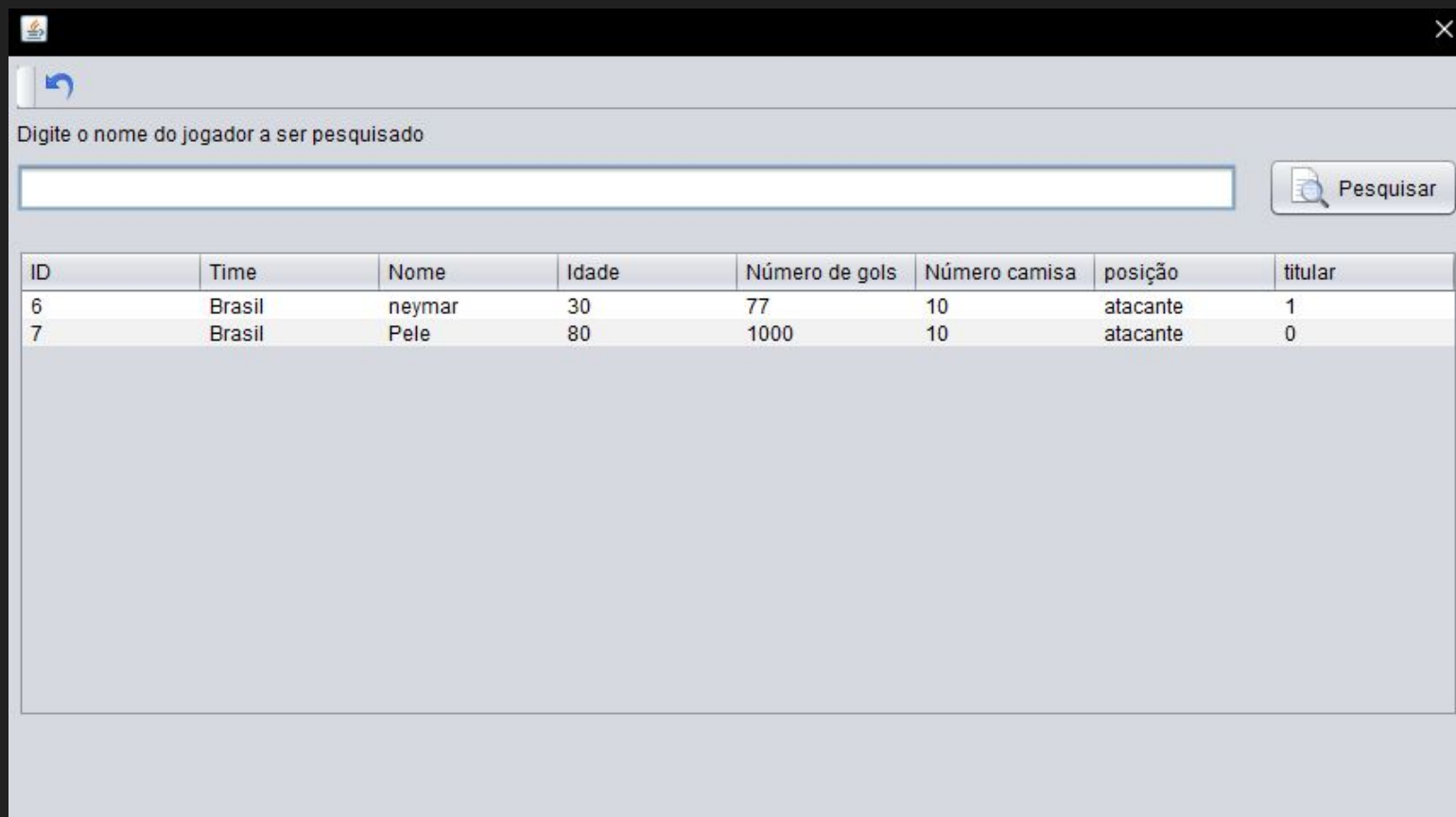
Questão 2. Ui - TelaCadastro

Pesquise o id do jogador:  pesquisar

| | | | | |
|---------|----------------------|------------------|----------------------------------|--|
| Índice | <input type="text"/> | Idade | <input type="text"/> | <input type="button" value="Novo"/> |
| Time | <input type="text"/> | Número de gols | <input type="text"/> | <input type="button" value="Alterar"/> |
| Nome | <input type="text"/> | Número da Camisa | <input type="text"/> | <input type="button" value="Excluir"/> |
| Posição | <input type="text"/> | Titular | <input type="text" value="Sim"/> | |

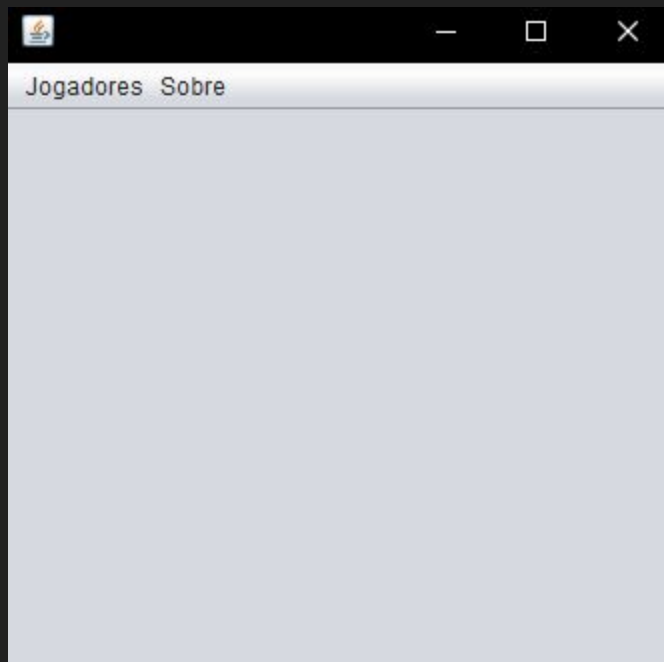
Questão 2. Ui - TelaVisualizaCadastro



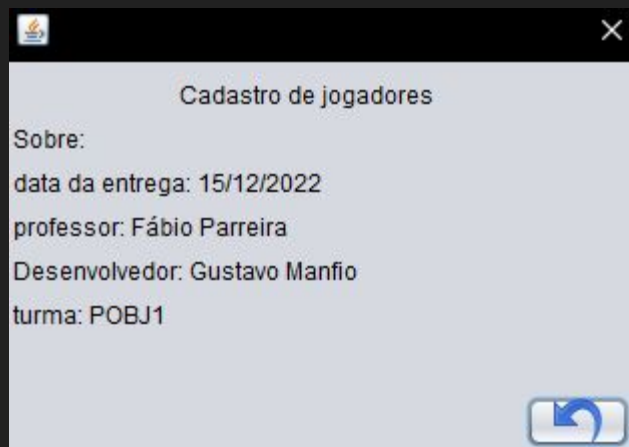
The screenshot shows a web application window titled "TelaVisualizaCadastro". It features a search bar with the placeholder text "Digite o nome do jogador a ser pesquisado" and a "Pesquisar" button. Below the search bar is a table displaying player information. The table has eight columns: ID, Time, Nome, Idade, Número de gols, Número camisa, posição, and titular. Two rows of data are visible, corresponding to Neymar and Pele.

| ID | Time | Nome | Idade | Número de gols | Número camisa | posição | titular |
|----|--------|--------|-------|----------------|---------------|----------|---------|
| 6 | Brasil | neymar | 30 | 77 | 10 | atacante | 1 |
| 7 | Brasil | Pele | 80 | 1000 | 10 | atacante | 0 |

Questão 2. Ui - TelaInicialMenu



Questão 2. Ui - TelaSobre

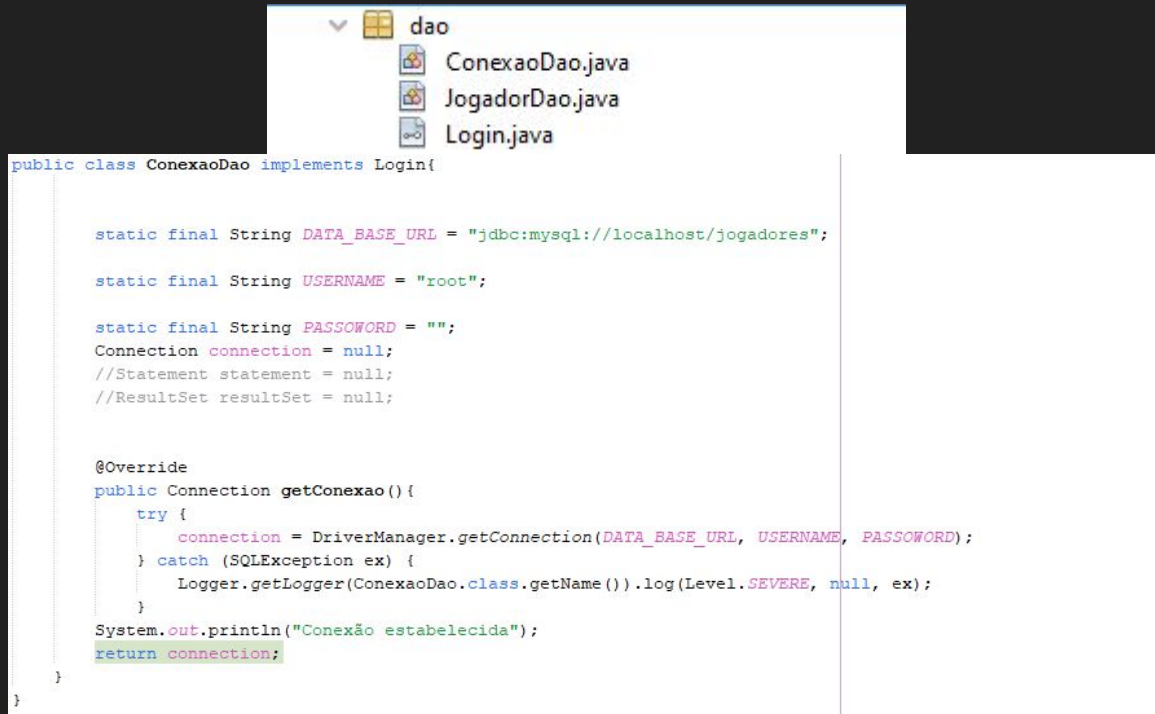


Questão 3 - JogadoresModelo

```
public class JogadoresModelo {  
    int Id;  
    String time;  
    String nome;  
    int idade;  
    int numerodegols;  
    int numerocamisa;  
    String posicao;  
    int titular;
```

Questão - 4 Camada Dao

Na camada DAO foi implementado o CRUD



```
public class ConexaoDao implements Login{

    static final String DATA_BASE_URL = "jdbc:mysql://localhost/jogadores";

    static final String USERNAME = "root";

    static final String PASSWORD = "";
    Connection connection = null;
    //Statement statement = null;
    //ResultSet resultSet = null;

    @Override
    public Connection getConexao(){
        try {
            connection = DriverManager.getConnection(DATA_BASE_URL, USERNAME, PASSWORD);
        } catch (SQLException ex) {
            Logger.getLogger(ConexaoDao.class.getName()).log(Level.SEVERE, null, ex);
        }
        System.out.println("Conexão estabelecida");
        return connection;
    }
}
```

Questão - 4 Camada Dao

```
public class JogadorDao {  
  
    private Connection conexao = null;  
  
    public JogadorDao() throws SQLException {...3 lines }  
  
    public ArrayList<JogadoresModelo> consultar(String nomePesquisar) throws SQLException {...29 lines }  
  
    public void inserir(JogadoresModelo jogadores) throws SQLException {...14 lines }  
    public void alterar(JogadoresModelo jogadores) throws SQLException {...18 lines }  
    public void remover(JogadoresModelo jogadores) throws SQLException {...12 lines }  
  
    public void localizarPrimeiroAtleta(JogadoresModelo jogador) throws SQLException {...18 lines }  
  
    public void localizarProximoAtleta(JogadoresModelo jogador, String id) throws SQLException {...24 lines }  
  
        public void localizarAtletaConsulta(JogadoresModelo jogador, String id) throws SQLException {...20 lines }  
  
    public void localizaUltimoAtleta(JogadoresModelo jogador) throws SQLException {...19 lines }  
        public void localizaAtletaAnterior(JogadoresModelo jogador, String id) throws SQLException {...26 lines }  
}
```

Questão 5 - Imagens

As imagens utilizados no desenvolvimento são as mesmas utilizadas no exemplos da aula



Questão 6 - Camada Negócio

Aqui estão os elementos responsáveis pela conexão da interface com o DAO.

```
public class JogadorNegocio {  
    public ArrayList consultarJogadores(String nomePesquisar) throws SQLException {...7 lines }  
  
    public void inserirJogadores(String time, String nome, int idade, int numerodegols,int numerocamisa, String posicao,int titular) throws SQLException {...17 lines }  
  
    public void alterarJogadores(String id,String time,String nome, int idade, int numerodegols,int numerocamisa, String posicao,int titular) throws SQLException {...20 lines }  
  
    public void removerJogadores(String id) throws SQLException {...13 lines }  
  
    public void localizaPrimeiroJogador(JogadoresModelo referenciaJogador) throws SQLException {...7 lines }  
  
    public void localizaProximoJogador(JogadoresModelo referenciaJogador, String id) throws SQLException {...7 lines }  
  
    public void localizaUltimoJogador (JogadoresModelo referenciaJogador) throws SQLException {...7 lines }  
  
    public void localizaJogadorAnterior(JogadoresModelo referenciaJogadores, String id) throws SQLException {...7 lines }  
  
    public void localizaJogadorCadastro(JogadoresModelo referenciaJogadores, String id) throws SQLException {...7 lines }  
}
```

Questão 7 - Comandos Banco de Dados

```
create database if not exists jogadores;
```

```
use jogadores;
```

```
create table atleta(
```

```
    id int not null auto_increment,
```

```
    time varchar(30),
```

```
    nome varchar(30),
```

```
    idade int,
```

```
    numerodegols int,
```

```
    numerocamisa int,
```

```
    posicao varchar(30),
```

```
    titular int,
```

```
    primary key (id)
```

```
);
```

Obrigado pela atenção :)