

Programação Orientada ao Objeto em Java

Problemática do Sistema de jogadores de futebol

Paulo Vitor Quintanilha - 202202780952¹, Luiz Paulo da Cunha e Souza - 202202331171²

¹Ciências da Computação – Universidade Estácio de Sá (UNESA)
Caixa Postal 24.020-320 – Niterói – RJ – Brasil

²Paulo Vitor Quintanilha - Universidade Estácio de Sá (UNESA), RJ

³Luiz Paulo da Cunha e Souza - Universidade Estácio de Sá (UNESA), RJ

{Paulo Vitor, Luiz Paulo} 202202780952@alunos.estacio.br, 202202331171@alunos.estacio.br

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

1.Nome do Sistema.

O nome do sistema é FUTSYSTEM.

2.Objetivo do Sistema.

O objetivo do sistema é funcionar como uma espécie de scouting para clubes de futebol contratarem jogadores.

2.1.Qual problemática o sistema irá resolver.

O Sistema cadastra as informações dos jogadores de futebol com base em seus respectivos nomes, valores, idade, clubes que jogam, se estão livres no mercado, facilitando assim a procura dos clubes de futebol que desejam contratar algum atleta.

3. Requisitos do Sistema

Criação da tabela "jogador": O sistema cria uma tabela chamada "jogador" no banco de dados PostgreSQL. A tabela possui colunas como nome, idade, time, país e valor.

Inserção de dados: O sistema insere registros na tabela "jogador" com informações sobre jogadores de futebol, como nome, idade, time, país e valor.

Atualização de dados: O sistema realiza uma atualização na tabela "jogador", modificando o valor de um jogador específico.

Exclusão de dados: O sistema exclui um jogador específico da tabela "jogador".

Consultas: O sistema realiza consultas na tabela "jogador" para recuperar informações sobre os jogadores. Há exemplos de consultas para listar todos os jogadores, consultar jogadores com valores altos, médios e baixos.

4. Caso de Uso do Sistema.

Pode ser utilizado por qualquer clube que deseja ter informações para contratar atletas.

5. Explicação do que foi Implementado.

. Esse código Java é um exemplo de interação com um banco de dados PostgreSQL usando JDBC (Java Database Connectivity). Ele realiza várias operações relacionadas a uma tabela chamada "jogador" em um banco de dados chamado "FUTSYSTEM".

Métodos principais:

-“main(String[] args)”: É o método principal que inicia a execução do programa. Ele lança uma exceção SQLException, que pode ocorrer durante a conexão com o banco de dados. O método main é responsável por estabelecer a conexão com o banco de dados, realizar diversas operações como criação de tabelas, inserção, atualização e exclusão de dados, além de executar consultas e exibir os resultados.

-Além do método principal, o código também inclui comentários que descrevem operações que podem ser realizadas, como a criação da tabela, inserção de dados, atualização, exclusão e consultas.

-As classes importadas do pacote java.sql são usadas para trabalhar com o banco de dados, estabelecer conexões, executar consultas e atualizações, e obter resultados de consultas. Algumas dessas classes são “Connection”, “DriverManager”, “ResultSet” e “Statement”.

-Classes abstratas: Não há.

-Banco de Dados: Foi utilizado um banco de dados relacional, por meio da API JDBC. O banco de dados pode ser acessado por meio da conexão estabelecida com a classe java.sql.conection.

Escolha dos Elementos usados: A escolha dos elementos foi feita por eles fazerem parte da API JDBC, que é uma biblioteca padrão de Java para acessar Banco de Dados Relacionais. Essa API fornece uma maneira consistente de interagir com diferentes bancos de dados e executar as operações do CRUD.

Importações: `import java.sql.Connection;` Importa a classe `Connection` do pacote `java.sql` para estabelecer uma conexão com o banco de dados.

`import java.sql.DriverManager;` Importa a classe `DriverManager` do pacote `java.sql` para gerenciar os drivers de conexão com o banco de dados.

`import java.sql.ResultSet;` Importa a classe `ResultSet` do pacote `java.sql` para lidar com resultados de consultas ao banco de dados.

`import java.sql.SQLException;` Importa a classe `SQLException` do pacote `java.sql` para lidar com exceções relacionadas ao SQL.

`import java.sql.Statement;` Importa a classe `Statement` do pacote `java.sql` para executar instruções SQL no banco de dados.

Declaração da classe: `FUTSYSTEM`.

Método `main(String[] args)`: É o ponto de entrada do programa. Lança uma exceção `SQLException` caso ocorra um erro na execução das instruções SQL.

Comentários: Existem alguns trechos de código comentados que representam diferentes instruções SQL, como criação de tabela, inserção de dados, atualização de dados, exclusão de dados e consultas.

String `driver`: Define a URL de conexão com o banco de dados PostgreSQL, especificando o endereço IP (127.0.0.1), a porta (5432) e o nome do banco de dados (`FUTSYSTEM`).

Declaração da variável `st` do tipo `Statement` e inicialização como `null`.

Bloco `try` com recursos (`try-with-resources`): Estabelece a conexão com o banco de dados utilizando a classe `DriverManager` e o método `getConnection`. Passa a URL de conexão, nome de usuário (`postgres`) e senha (`Estacio@123`) para autenticação no banco de dados. Se a conexão for bem-sucedida, imprime "Connected to the database!". Caso contrário, imprime "Failed to make connection!".

Bloco `catch`: Captura exceções `SQLException` e imprime informações sobre o erro.

6. Conclusão.

6.1. Análise crítica do Sistema Criado:

Este código interage com um banco de dados PostgreSQL chamado "FUTSYSTEM". Aqui está uma análise crítica do código:

1- Acesso ao banco de dados: O código estabelece uma conexão com o banco de dados usando as credenciais de usuário "postgres" e senha "Estacio@123". É fundamental garantir que essas informações de autenticação sejam mantidas em segurança e não sejam expostas publicamente.

2- Tratamento de exceção: O código inclui um bloco try-catch para capturar exceções do tipo SQLException. No entanto, o tratamento atual apenas imprime a mensagem de exceção no console. Em uma aplicação real, é recomendável implementar um tratamento mais robusto de exceções, como registrar os erros em um arquivo de log ou notificar os usuários sobre problemas de conexão com o banco de dados.

3- Execução de consultas: O código demonstra exemplos de consultas ao banco de dados usando a classe Statement. No entanto, todas as consultas estão comentadas, então não é possível avaliar a funcionalidade e a precisão dessas consultas. Para garantir a integridade dos dados e a segurança, é importante validar e sanitizar as entradas do usuário ao construir consultas dinâmicas.

Em resumo, o código mostra uma estrutura básica para interagir com um banco de dados em Java.

6.2. Problemática Abordada foi Solucionada:

O código implementa uma parte do que realmente queria fazer, ele demonstra jogadores se eles estão caros, no preço ou se ele está barato, nós queríamos que ele fizesse outras coisas como separar por idade, para clubes que quisesse contratar jogadores mais experientes, ou jovens promessas, ou jogadores mais próximos no seu auge, faltou também colocar gols, assistências, números de partidas entre outros.

Figure 1. Início do Código.

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class FUTSYSTEM {
    public static void main(String[] args) throws SQLException {
        /*String SQLcriarTabela = "CREATE TABLE jogador (" +
            "nome VARCHAR(100) PRIMARY KEY, " +
            "idade VARCHAR(10), " +
            "time VARCHAR(50), " +
            "pais VARCHAR(30), " +
            "valor DECIMAL(12,2)" +
            ");*/

        /* String SQLInserirJogador = "INSERT INTO jogador (nome, idade, time, pais, valor) VALUES " +
            "('Neymar Junior', '31', 'PSG', 'Brasil', 70000000.00), " +
            "('Khvicha Kvaratskhelia', '22', 'NAPOLI', 'Geórgia', 85000000.00), " +
            "('Lionel Messi', '35', 'INTER MIAMI', 'Argentina', 45000000.00), " +
            "('Vinicius Junior', '22', 'REAL MADRID', 'Brasil', 150000000.00), " +
            "('Cristiano Ronaldo', '38', 'AL - NASSR', 'Portugal', 15000000.00), " +
            "('Yuri Alberto', '22', 'CORINTHIANS', 'Brasil', 18000000.00), " +
            "('Endrick', '16', 'PALMEIRAS', 'Brasil', 20000000.00), " +
            "('Rodrygo', '22', 'REAL MADRID', 'Brasil', 100000000.00), " +
            "('Mbappe', '24', 'PSG', 'França', 180000000.00), " +

```

Figure 2. Meio do Código 1.

```

26      "('Mbappe', '24', 'PSG', 'França', 180000000.00), " +
27      "('Haaland', '22', 'MANCHESTER CITY', 'Noruega', 170000000.00), " +
28      "('Kevin De Bruyne', '31', 'MANCHESTER CITY', 'Bélgica', 80000000.00), " +
29      "('Ribamar', '26', 'PONTE PRETA', 'Brasil', 400000.00), " +
30      "('Andrey Santos', '19', 'VASCO', 'Brasil', 15000000.00), " +
31      "('De Arrascaeta', '29', 'FLAMENGO', 'Uruguai', 19000000.00), " +
32      "('Gabriel Barbosa', '26', 'FLAMENGO', 'Brasil', 21000000.00), " +
33      "('Luis Suárez', '36', 'GRÊMIO', 'Uruguai', 4000000.00), " +
34      "('Karim Benzema', '35', 'AL - ITIHAD', 'França', 25000000.00), " +
35      "('Dudu', '31', 'PALMEIRAS', 'Brasil', 11000000.00), " +
36      "('Rony', '28', 'PALMEIRAS', 'Brasil', 13000000.00), " +
37      "('Pedro', '25', 'FLAMENGO', 'Brasil', 22000000.00), " +
38      "('Pedri', '20', 'BARCELONA', 'Espanha', 100000000.00), " +
39      "('Gavi', '18', 'BARCELONA', 'Espanha', 90000000.00), " +
40      "('Robert Lewandowski', '34', 'BARCELONA', 'Polónia', 30000000.00), " +
41      "('Thiago Silva', '38', 'CHELSEA', 'Brasil', 2500000.00), " +
42      "('Casemiro', '31', 'MANCHESTER UNITED', 'Brasil', 50000000.00), " +
43      "('Antony', '23', 'MANCHESTER UNITED', 'Brasil', 70000000.00), " +
44      "('Rashford', '25', 'MANCHESTER UNITED', 'Inglaterra', 80000000.00), " +
45      "('Grealish', '27', 'MANCHESTER CITY', 'Inglaterra', 70000000.00), " +
46      "('Bruno Guimarães', '25', 'NEWCASTLE UNITED', 'Brasil', 60000000.00), " +
47      "('Joelinton', '26', 'NEWCASTLE UNITED', 'Brasil', 38000000.00), " +
48      "('Mohamed Salah', '31', 'LIVERPOOL', 'Egito', 70000000.00), " +
49      "('Alisson', '31', 'LIVERPOOL', 'Brasil', 45000000.00), " +
50      "('Van Dijk', '31', 'LIVERPOOL', 'Holanda', 45000000.00), " +
51      "('Alexander Arnold', '24', 'LIVERPOOL', 'Inglaterra', 65000000.00), " +

```

Figure 3. Meio do Código 2.

```

1  "{'Alexander Arnold', '24', 'LIVERPOOL', 'Inglaterra', 65000000.00), " +
2  "{'Robertson', '29', 'LIVERPOOL', 'Escócia', 48000000.00), " +
3  "{'Fabinho', '29', 'LIVERPOOL', 'Brasil', 45000000.00), " +
4  "{'Son', '30', 'TOTTENHAM', 'Coreia do Sul', 60000000.00), " +
5  "{'Harry Kane', '29', 'TOTTENHAM', 'Inglaterra', 90000000.00), " +
6  "{'Richarlison', '26', 'TOTTENHAM', 'Brasil', 55000000.00), " +
7  "{'Gabriel Martinelli', '21', 'ARSENAL', 'Brasil', 70000000.00), " +
8  "{'Odegaard', '24', 'ARSENAL', 'Noruega', 80000000.00), " +
9  "{'Bukayo Saka', '21', 'ARSENAL', 'Inglaterra', 100000000.00), " +
10 "{'Vitor Roque', '18', 'ATHLETICO PARANAENSE', 'Brasil', 18000000.00), " +
11 "{'Lucas Perri', '25', 'BOTAFOGO', 'Brasil', 2000000.00), " +
12 "{'Adryelson', '25', 'BOTAFOGO', 'Brasil', 3000000.00), " +
13 "{'Marçal', '34', 'BOTAFOGO', 'Brasil', 1200000.00), " +
14 "{'Luiz Henrique', '21', 'BOTAFOGO', 'Brasil', 6000000.00), " +
15 "{'Matheus Nascimento', '19', 'BOTAFOGO', 'Brasil', 7000000.00), " +
16 "{'Tiquinho Soares', '32', 'BOTAFOGO', 'Brasil', 4000000.00), " +
17 "{'Victor Sá', '29', 'BOTAFOGO', 'Brasil', 2000000.00), " +
18 "{'Jude Bellingham', '19', 'REAL MADRID', 'Inglaterra', 120000000.00), " +
19 "{'Luan', '30', 'CORINTHIANS', 'Brasil', 900000.00), " +
20 "{'Roberto Firmino', '31', 'SEM TIME', 'Brasil', 0), " +
21 "{' Hazard', '32', 'SEM TIME', 'Belgica', 0), " +
22 "{'Gundogan', '32', 'SEM TIME', 'Alemanha', 0), " +
23 "{'Lucas Moura', '30', 'SEM TIME', 'Brasil', 0), " +
24 "{'Philippe Coutinho', '31', 'ASTON VILLA', 'Brasil', 14000000.00), " +
25 "{'Di Maria', '35', 'SEM TIME', 'Argentina', 0), " +
26 "{'Roger Guedes', '26', 'CORINTHIANS', 'Brasil', 8500000.00)}";*/

```

Figure 4. Meio do Código 3.

```

// String SQLatualizarDados = "UPDATE jogador SET valor = 110000000.00 WHERE nome = 'Bukayo Saka'";

// String SQLexcluirJogador = "DELETE FROM jogador WHERE nome = 'Gavi'";

String driver = "jdbc:postgresql://127.0.0.1:5432/FUTSYSTEM";

Statement st = null;

try {Connection conn = DriverManager.getConnection(driver, "postgres", "Estacio@123"); {
    if (conn != null) {
        System.out.println("Connected to the database!");
    } else {
        System.out.println("Failed to make connection!");
    }
}

// Criação da tabela jogador
/* System.out.println("Criando Tabela, aguarde...");
st = conn.createStatement();
st.executeUpdate(SQLcriarTabela);
System.out.println("Tabela criada com sucesso!");
st.close();*/

```

Figure 5. Meio do Código 4.

```

// Inserção de dados
/*System.out.println("Inserindo dados na Tabela...");
st = conn.createStatement();
st.executeUpdate(SQLInserirJogador);
System.out.println("Dados inseridos!");
st.close();*/

// Consulta de dados
/*System.out.println("Consultando dados na Tabela...");
st = conn.createStatement();
String SQLconsultarDados = "SELECT nome, idade, pais, time, valor FROM jogador";
ResultSet result = st.executeQuery(SQLconsultarDados);
while (result.next()) {
    System.out.println("-----");
    System.out.println("Nome do jogador: " + result.getString("nome"));
    System.out.println("Idade do jogador: " + result.getString("idade"));
    System.out.println("Time do jogador: " + result.getString("time"));
    System.out.println("País do jogador: " + result.getString("pais"));
    System.out.println("Valor do jogador: " + result.getString("valor"));
}
result.close();
st.close();*/
//fim do bloco consulta

```

Figure 6. Meio do Código 5.

```

//bloco atualiza dados
/*System.out.println("Atualizando dados na Tabela...");
st = conn.createStatement();
st.executeUpdate(SQLatualizarDados);
System.out.println("Dados atualizados!");
st.close();
conn.close();*/

/*System.out.println("Excluindo dados na Tabela...");
st = conn.createStatement();
st.executeUpdate(SQLExcluirJogador);
System.out.println("Dados excluídos!");
st.close();
conn.close();*/

/*System.out.println("Consultando jogadores de alto valor...");
st = conn.createStatement();
String SQLconsultarValorAlto = "SELECT nome, idade, pais, time, valor FROM jogador " +
    "WHERE (valor >= 60000000.00)";
ResultSet result = st.executeQuery(SQLconsultarValorAlto);
while (result.next()) {
    System.out.println("-----");
    System.out.println("Nome do jogador: " + result.getString("nome"));
    System.out.println("Idade do jogador: " + result.getString("idade"));
}

```

Figure 7. Meio do Código 6.

```
result.close();
st.close();*/

// Consulta de jogadores com valor baixo
/*System.out.println("Consultando jogadores de valor baixo...");
st = conn.createStatement();
String SQLconsultarValorBaixo = "SELECT nome, idade, pais, time, valor FROM jogador " +
    "WHERE (valor < 10000000.00 AND valor > 0)";
ResultSet result = st.executeQuery(SQLconsultarValorBaixo);
while (result.next()) {
    System.out.println("-----");
    System.out.println("Nome do jogador: " + result.getString("nome"));
    System.out.println("Idade do jogador: " + result.getString("idade"));
    System.out.println("Time do jogador: " + result.getString("time"));
    System.out.println("País do jogador: " + result.getString("pais"));
    System.out.println("Valor do jogador: R$ " + result.getString("valor"));
}
result.close();
st.close();*/

// Consulta de jogadores sem time
/* System.out.println("Consultando jogadores sem time...");
st = conn.createStatement();
String SQLconsultarSemTime = "SELECT nome, idade, pais, time, valor FROM jogador " +
    "WHERE ( valor = 0)";
ResultSet result = st.executeQuery(SQLconsultarSemTime);
while (result.next()) {
    System.out.println("-----");
}
```

Figure 8. Fim do Código.

```
String SQLconsultarSemTime = "SELECT nome, idade, pais, time, valor FROM jogador " +
    "WHERE ( valor = 0)";
ResultSet result = st.executeQuery(SQLconsultarSemTime);
while (result.next()) {
    System.out.println("-----");
    System.out.println("Nome do jogador: " + result.getString("nome"));
    System.out.println("Idade do jogador: " + result.getString("idade"));
    System.out.println("Time do jogador: " + result.getString("time"));
    System.out.println("País do jogador: " + result.getString("pais"));
    System.out.println("Valor do jogador: R$ " + result.getString("valor"));
}
result.close();
st.close();*/

} catch (SQLException e) {
    System.err.format("SQL State: %s\n%s", e.getSQLState(), e.getMessage());
}
}
```

Figure 9. Rodando o código: Criação da tabela jogador com nome, idade,time,páís e valor.

<input type="checkbox"/> Name	Comment
<input type="checkbox"/> nome	
<input type="checkbox"/> idade	
<input type="checkbox"/> time	
<input type="checkbox"/> páís	
<input type="checkbox"/> valor	

Figure 10. Rodando o código: Inserindo dados na tabela.

1	Neymar Junior	31	PSG	Brasil	70000000.00
2	Khvicha Kvaratskhelia	22	NAPOLI	Geórgia	85000000.00
3	Lionel Messi	35	INTER MIAMI	Argentina	45000000.00
4	Vinicius Junior	22	REAL MADRID	Brasil	150000000.00
5	Cristiano Ronaldo	38	AL - NASSR	Portugal	15000000.00
6	Yuri Alberto	22	CORINTHIANS	Brasil	18000000.00
7	Endrick	16	PALMEIRAS	Brasil	20000000.00
8	Rodrygo	22	REAL MADRID	Brasil	100000000.00
9	Mbappe	24	PSG	França	180000000.00
10	Haaland	22	MANCHESTER CITY	Noruega	170000000.00
11	Kevin De Bruyne	31	MANCHESTER CITY	Bélgica	80000000.00
12	Ribamar	26	PONTE PETRA	Brasil	400000.00
13	Andrey Santos	19	VASCO	Brasil	15000000.00
14	De Arrascaeta	29	FLAMENGO	Uruguai	19000000.00

Figure 11. Rodando o código: Tabela com jogador Gavi.

14	De Arrascaeta	29	FLAMENGO	Uruguai	19000000.00
15	Gabriel Barbosa	26	FLAMENGO	Brasil	21000000.00
16	Luis Suárez	36	GRÊMIO	Uruguai	4000000.00
17	Karim Benzema	35	AL - ITTIHAD	França	25000000.00
18	DUDU	31	PALMEIRAS	Brasil	11000000.00
19	Rony	28	PALMEIRAS	Brasil	13000000.00
20	Pedro	25	FLAMENGO	Brasil	22000000.00
21	Pedri	20	BARCELONA	Espanha	100000000.00
22	Gavi	18	BARCELONA	Espanha	90000000.00
23	Robert Lewandowski	34	BARCELONA	Polônia	30000000.00

Figure 12. Rodando o código: Bukayo Saka com o valor desatualizado.

35	Robertson	29	LIVERPOOL	Escócia	48000000.00
36	Fabinho	29	LIVERPOOL	Brasil	45000000.00
37	Son	30	TOTTENHAM	Coreia do Sul	60000000.00
38	Harry Kane	29	TOTTENHAM	Inglaterra	90000000.00
39	Richarlison	26	TOTTENHAM	Brasil	55000000.00
40	Gabriel Martinelli	21	ARSENAL	Brasil	70000000.00
41	Odegaard	24	ARSENAL	Noruega	80000000.00
42	Bukayo Saka	21	ARSENAL	Inglaterra	100000000.00
43	Vitor Roque	18	ATHLETICO PARANAENSE	Brasil	18000000.00
44	Lucas Perri	25	BOTAFOGO	Brasil	2000000.00

Figure 13. Rodando o código: Bukayo Saka com o valor atualizado.

53	Hazard	32	SEM TIME	Belgica	0.00
54	Gundogan	32	SEM TIME	Alemanha	0.00
55	Lucas Moura	30	SEM TIME	Brasil	0.00
56	Philippe Coutinho	31	ASTON VILLA	Brasil	14000000.00
57	Di Maria	35	SEM TIME	Argentina	0.00
58	Roger Guedes	26	CORINTHIANS	Brasil	8500000.00
59	Bukayo Saka	21	ARSENAL	Inglaterra	110000000.00

Figure 13. Rodando o código: Tabela sem o jogador Gavi.

16	Luis Suárez	36	GRÊMIO	Uruguai	4000000.00
17	Karim Benzema	35	AL - ITTIHAD	França	25000000.00
18	DUDU	31	PALMEIRAS	Brasil	11000000.00
19	Rony	28	PALMEIRAS	Brasil	13000000.00
20	Pedro	25	FLAMENGO	Brasil	22000000.00
21	Pedri	20	BARCELONA	Espanha	100000000.00
22	Robert Lewandowski	34	BARCELONA	Polônia	30000000.00
23	Thiago Silva	38	CHELSEA	Brasil	2500000.00
24	Casemiro	31	MANCHESTER UNITED	Brasil	50000000.00
25	Antony	23	MANCHESTER UNITED	Brasil	70000000.00
26	Rashford	25	MANCHESTER UNITED	Inglaterra	80000000.00
27	Grealish	27	MANCHESTER CITY	Inglaterra	70000000.00

Figure 14. Rodando o código: Consulta de jogadores sem clube.

Nome do jogador: Hazard
Idade do jogador: 32
Time do jogador: SEM TIME
País do jogador: Belgica
Valor do jogador: R\$ 0.00

Nome do jogador: Gundogan
Idade do jogador: 32
Time do jogador: SEM TIME
País do jogador: Alemanha
Valor do jogador: R\$ 0.00

Nome do jogador: Lucas Moura
Idade do jogador: 30
Time do jogador: SEM TIME
País do jogador: Brasil
Valor do jogador: R\$ 0.00

Nome do jogador: Di Maria
Idade do jogador: 35
Time do jogador: SEM TIME
País do jogador: Argentina
Valor do jogador: R\$ 0.00

