A2-Projeto de Gerenciamento de Time de Basquete

O projeto de gerenciamento de um **time de basquete** tem como objetivo organizar e controlar informações relacionadas a jogadores, times, treinadores, jogos e estatísticas de desempenho. Ele utiliza um banco de dados MySQL para persistir essas informações e permite realizar operações de **CRUD** (Criar, Ler, Atualizar e Excluir) através de uma aplicação Java que interage com o banco de dados.

Aqui está uma explicação detalhada da funcionalidade do projeto:

**1. Cadastro e Gerenciamento de Pessoas (Jogadores e Treinadores)**

* **Jogadores**: O sistema permite o cadastro de jogadores de basquete, armazenando informações como **nome**, **idade**, **posição** (ex: armador, ala, pivô) e o **time** ao qual pertencem.
* **Treinadores**: Além dos jogadores, o sistema também permite o cadastro de treinadores, armazenando informações como **nome** e **experiência** (ex: anos de experiência como treinador).

**Funcionalidade**:

* O usuário pode adicionar, visualizar, atualizar ou excluir jogadores e treinadores.
* Cada jogador pertence a um time específico e cada time tem um treinador associado.

**2. Cadastro e Gerenciamento de Times**

* **Times**: O sistema armazena os times de basquete, com informações como **nome** do time, **cidade** onde está localizado e o **treinador** responsável por ele.

**Funcionalidade**:

* O usuário pode cadastrar novos times, associando-os a um treinador específico.
* Um time pode ter vários jogadores, mas um jogador pertence a apenas um time.

**3. Cadastro e Gerenciamento de Jogos**

* **Jogos**: O sistema permite o registro de partidas entre dois times, armazenando informações como:
  + **Data do jogo**.
  + **Times envolvidos** (identificados pelas chaves estrangeiras time\_casa\_id e time\_fora\_id).
  + **Placar** de cada time (ex: 100-98).

**Funcionalidade**:

* O usuário pode registrar novos jogos, associando-os aos times participantes.
* O sistema registra o resultado do jogo, o que pode ser útil para análises posteriores.

**4. Cadastro e Gerenciamento de Estatísticas dos Jogadores**

* **Estatísticas**: Para cada jogo, o sistema registra as **estatísticas de desempenho** de cada jogador. Isso pode incluir:
  + **Pontos** marcados no jogo.
  + **Assistências** feitas.
  + **Rebotes** conquistados.

**Funcionalidade**:

* O sistema permite que o usuário registre as estatísticas de cada jogador em cada jogo.
* Essas informações ajudam a analisar o desempenho individual de cada jogador em cada partida, facilitando a avaliação de seu desempenho ao longo do tempo.

**5. Operações de CRUD**

As operações básicas de **CRUD** (Criar, Ler, Atualizar, Excluir) são implementadas para cada entidade (Jogador, Time, Treinador, Jogo, Estatísticas) e têm as seguintes funcionalidades:

* **Criar**: Adiciona novos registros (ex: um novo jogador, um novo jogo).
* **Ler**: Permite buscar registros no banco de dados, como visualizar todos os jogadores de um time ou obter as estatísticas de um jogador específico em um jogo.
* **Atualizar**: Altera informações já cadastradas, como modificar o nome de um jogador ou a pontuação de um jogo.
* **Excluir**: Remove registros do banco de dados, como excluir um jogador que saiu do time ou remover um jogo agendado.

**6. Interação com o Banco de Dados**

A aplicação Java interage com o banco de dados MySQL através de **JDBC (Java Database Connectivity)**. As operações de CRUD são realizadas utilizando **Prepared Statements** para garantir segurança contra SQL Injection e para facilitar a inserção de dados dinâmicos nas consultas SQL.

**7. Diagrama de Relacionamentos**

* **Jogador** está associado a um **Time** (muitos jogadores para um time).
* **Treinador** está associado a um **Time** (um treinador para um time).
* **Jogo** envolve dois **Times** (casa e fora), e a **Estatística** está relacionada tanto a **Jogador** quanto a **Jogo**.

**8. Aplicação e Interface (opcional)**

A aplicação pode ser desenvolvida com ou sem uma interface gráfica. Caso seja usada uma interface, o usuário pode interagir com o sistema visualmente, preenchendo formulários para cadastrar jogadores, times e jogos, além de visualizar as estatísticas e resultados dos jogos. Caso contrário, a interação pode ser feita via comandos no console.

**Exemplo de Fluxo de Funcionalidade:**

1. **Cadastrar um Novo Time**:
   * O usuário informa o nome do time, cidade e o treinador.
   * O time é registrado no banco de dados.
2. **Cadastrar Jogadores**:
   * O usuário informa os dados dos jogadores (nome, idade, posição) e seleciona a qual time ele pertence.
   * O jogador é registrado, associado ao time correspondente.
3. **Registrar um Jogo**:
   * O usuário registra os dados do jogo, como data, time da casa, time visitante e o placar.
   * O jogo é armazenado no banco de dados.
4. **Cadastrar Estatísticas dos Jogadores**:
   * Após um jogo, o usuário registra as estatísticas de cada jogador (pontos, assistências, rebotes).
   * As estatísticas são associadas ao jogador e ao jogo correspondente.

**Conclusão:**

Este sistema é útil para **gestão e análise de times de basquete**, ajudando a armazenar dados importantes sobre os jogadores, os times, os jogos e as estatísticas. Ele permite realizar análises do desempenho dos jogadores ao longo das temporadas e facilita o acompanhamento da evolução das equipes, além de melhorar a organização do time e o acompanhamento dos jogos disputados.

4o mini

Parte superior do formulário



Parte inferior do formulário