A2-Projeto de Gerenciamento de Time de Basquete

**Relacionamentos e Chaves Estrangeiras**

1. **Jogador → Time**: A tabela jogador tem a coluna time\_id como chave estrangeira para time.id.
   * **Chave estrangeira**: time\_id (em jogador)
   * Relacionamento: Muitos jogadores podem pertencer a um único time.
2. **Treinador → Time**: A tabela treinador tem a coluna time\_id como chave estrangeira para time.id.
   * **Chave estrangeira**: time\_id (em treinador)
   * Relacionamento: Um treinador é responsável por um único time.
3. **Time → Jogador**: A tabela time tem a associação com a tabela jogador através da chave estrangeira time\_id em jogador.
   * **Chave estrangeira**: time\_id (em jogador)
   * Relacionamento: Um time tem muitos jogadores.
4. **Time → Treinador**: A tabela time tem a associação com a tabela treinador através da chave estrangeira time\_id em treinador.
   * **Chave estrangeira**: time\_id (em treinador)
   * Relacionamento: Cada time tem um único treinador.
5. **Jogo → Time (Time da Casa e Time Visitante)**: A tabela jogo tem as colunas time\_casa\_id e time\_visitante\_id como chaves estrangeiras para time.id.
   * **Chaves estrangeiras**:
     + time\_casa\_id (em jogo, referenciando time.id)
     + time\_visitante\_id (em jogo, referenciando time.id)
   * Relacionamento: Um jogo envolve dois times, um da casa e outro visitante.
6. **Jogo → Arbitro**: A tabela jogo tem a coluna arbitro\_id como chave estrangeira para arbitro.id.
   * **Chave estrangeira**: arbitro\_id (em jogo, referenciando arbitro.id)
   * Relacionamento: Um árbitro apita vários jogos, mas cada jogo tem um único árbitro.
7. **Estatísticas → Jogador**: A tabela estatisticas tem a coluna jogador\_id como chave estrangeira para jogador.id.
   * **Chave estrangeira**: jogador\_id (em estatisticas, referenciando jogador.id)
   * Relacionamento: Muitas estatísticas podem ser associadas a um jogador.
8. **Estatísticas → Jogo**: A tabela estatisticas tem a coluna jogo\_id como chave estrangeira para jogo.id.
   * **Chave estrangeira**: jogo\_id (em estatisticas, referenciando jogo.id)
   * Relacionamento: Muitas estatísticas podem ser associadas a um jogo.
9. **Arbitro → Jogo**: A tabela arbitro tem uma associação de um para muitos com a tabela jogo (referenciado através da chave estrangeira arbitro\_id na tabela jogo).
   * **Chave estrangeira**: arbitro\_id (em jogo, referenciando arbitro.id)
   * Relacionamento: Um árbitro apita muitos jogos, mas cada jogo tem um único árbitro.

**Diagrama Relacional Resumido com Chaves Estrangeiras:**

* **Jogador**
  + id (int, chave primária)
  + nome (varchar)
  + idade (int)
  + posicao (varchar)
  + time\_id (int, chave estrangeira para time.id)
* **Treinador**
  + id (int, chave primária)
  + nome (varchar)
  + idade (int)
  + experiencia (int)
  + time\_id (int, chave estrangeira para time.id)
* **Arbitro**
  + id (int, chave primária)
  + nome (varchar)
  + idade (int)
  + certificado (varchar)
* **Time**
  + id (int, chave primária)
  + nome (varchar)
  + cidade (varchar)
* **Jogo**
  + id (int, chave primária)
  + data (date)
  + time\_casa\_id (int, chave estrangeira para time.id)
  + time\_visitante\_id (int, chave estrangeira para time.id)
  + arbitro\_id (int, chave estrangeira para arbitro.id)
* **Estatisticas**
  + id (int, chave primária)
  + pontos (int)
  + assistencias (int)
  + rebotes (int)
  + jogador\_id (int, chave estrangeira para jogador.id)
  + jogo\_id (int, chave estrangeira para jogo.id)

**Conclusão:**

Agora, com a tabela Pessoa desconsiderada, a herança foi implementada diretamente nas tabelas Jogador, Treinador e Arbitro. A chave estrangeira foi configurada para associar corretamente as tabelas entre si, permitindo que as relações entre jogadores, treinadores, árbitros, jogos, times e estatísticas sejam bem representadas no banco de dados.

PARTE DOIS

**Estrutura do Projeto com Herança e Associações**

**1. Classe Abstrata Pessoa**

* Atributos comuns para todos:
  + id (int)
  + nome (String)
  + idade (int)
* Esta classe será a superclasse abstrata, ou seja, não pode ser instanciada diretamente, mas serve como base para as classes que herdam dela.

**2. Classes Herdeiras de Pessoa**

* **Jogador** (herda de Pessoa):
  + Atributos adicionais:
    - posicao (String)
    - **Associação com Time**: Um jogador pertence a um time, então há uma associação **muitos para um** entre Jogador e Time.
* **Treinador** (herda de Pessoa):
  + Atributos adicionais:
    - experiencia (int)
    - **Associação com Time**: Um treinador é responsável por um time, portanto, há uma associação **um para um** entre Treinador e Time.
* **Arbitro** (herda de Pessoa):
  + Atributos adicionais:
    - certificado (String)
    - **Associação com Jogo**: Um árbitro apita muitos jogos, então há uma associação **um para muitos** entre Arbitro e Jogo.

**3. Outras Classes de Associação (interagindo com a herança)**

* **Time**:
  + Atributos:
    - nome (String)
    - cidade (String)
  + **Associação com Jogador**: Um time tem vários jogadores, portanto, existe uma associação de **um para muitos** com Jogador.
  + **Associação com Treinador**: Um time tem um treinador, portanto, há uma associação de **um para um** com Treinador.
* **Jogo**:
  + Atributos:
    - id (int)
    - data (Date)
  + **Associação com Time**: Um jogo envolve dois times, um da casa e outro visitante, portanto, existem duas associações **muitos para um** entre Jogo e Time (representando o time da casa e o time visitante).
  + **Associação com Arbitro**: Um jogo é apitado por um árbitro, então há uma associação **muitos para um** entre Jogo e Arbitro.
* **Estatísticas**:
  + Atributos:
    - pontos (int)
    - assistencias (int)
    - rebotes (int)
  + **Associação com Jogador**: Muitas estatísticas podem ser associadas a um jogador, portanto, existe uma associação **muitos para um** entre Estatísticas e Jogador.
  + **Associação com Jogo**: Muitas estatísticas são registradas para um jogo, portanto, existe uma associação **muitos para um** entre Estatísticas e Jogo.

**Relacionamentos:**

1. **Pessoa** é a classe abstrata, e **Jogador**, **Treinador**, e **Arbitro** são as classes que a herdam.
   * **Jogador** e **Treinador** herdam de **Pessoa**, mantendo os atributos comuns de id, nome e idade.
   * **Arbitro** também herda de **Pessoa**, com atributos adicionais como certificado.
2. **Time** tem uma associação de **um para muitos** com **Jogador**.
   * Um time pode ter vários jogadores, mas um jogador pertence a um único time.
3. **Time** tem uma associação de **um para um** com **Treinador**.
   * Cada time tem um único treinador, e um treinador é responsável por um único time.
4. **Jogo** tem duas associações de **muitos para um** com **Time**.
   * Um jogo envolve dois times: um time da casa e um time visitante.
5. **Estatísticas** tem associações de **muitos para um** com **Jogador** e **Jogo**.
   * Cada estatística está associada a um jogador e a um jogo específico.
6. **Arbitro** tem uma associação de **um para muitos** com **Jogo**.
   * Um árbitro apita vários jogos, mas cada jogo tem apenas um árbitro.

**Exemplo de Como o Diagrama UML Ficaria**

**Classes e Heranças:**

* **Pessoa (classe abstrata)**
  + Atributos: id, nome, idade
  + (classe abstrata, não instanciável)
* **Jogador (herda de Pessoa)**
  + Atributos: posicao
  + Relacionamento com Time (muitos para um)
* **Treinador (herda de Pessoa)**
  + Atributos: experiencia
  + Relacionamento com Time (um para um)
* **Arbitro (herda de Pessoa)**
  + Atributos: certificado
  + Relacionamento com Jogo (um para muitos)

**Outras Classes e Associações:**

* **Time**
  + Atributos: nome, cidade
  + Relacionamentos:
    - **um para muitos** com Jogador
    - **um para um** com Treinador
* **Jogo**
  + Atributos: id, data
  + Relacionamentos:
    - **muitos para um** com Time (time da casa)
    - **muitos para um** com Time (time visitante)
    - **muitos para um** com Arbitro
* **Estatísticas**
  + Atributos: pontos, assistencias, rebotes
  + Relacionamentos:
    - **muitos para um** com Jogador
    - **muitos para um** com Jogo

**Relacionamentos e Multiplicidade:**

* **Pessoa** → **Jogador/Treinador/Arbitro**: Herança (uma classe abstrata que é estendida por outras).
* **Jogador** → **Time**: Muitos jogadores pertencem a um time.
* **Treinador** → **Time**: Um treinador é responsável por um time.
* **Time** → **Jogador**: Um time tem vários jogadores.
* **Time** → **Treinador**: Cada time tem um treinador.
* **Jogo** → **Time**: Um jogo envolve dois times (um para o time da casa, outro para o time visitante).
* **Estatísticas** → **Jogador**: Cada estatística pertence a um jogador.
* **Estatísticas** → **Jogo**: Cada estatística está associada a um jogo.
* **Arbitro** → **Jogo**: Um árbitro apita vários jogos, mas cada jogo tem um único árbitro.

Este diagrama pode ser criado usando ferramentas como **Lucidchart**, **Draw.io** ou **Visual Paradigm**. Com essa estrutura, você garante que o projeto tenha a herança e as associações necessárias para gerenciar um time de basquete de forma eficiente.

4o mini