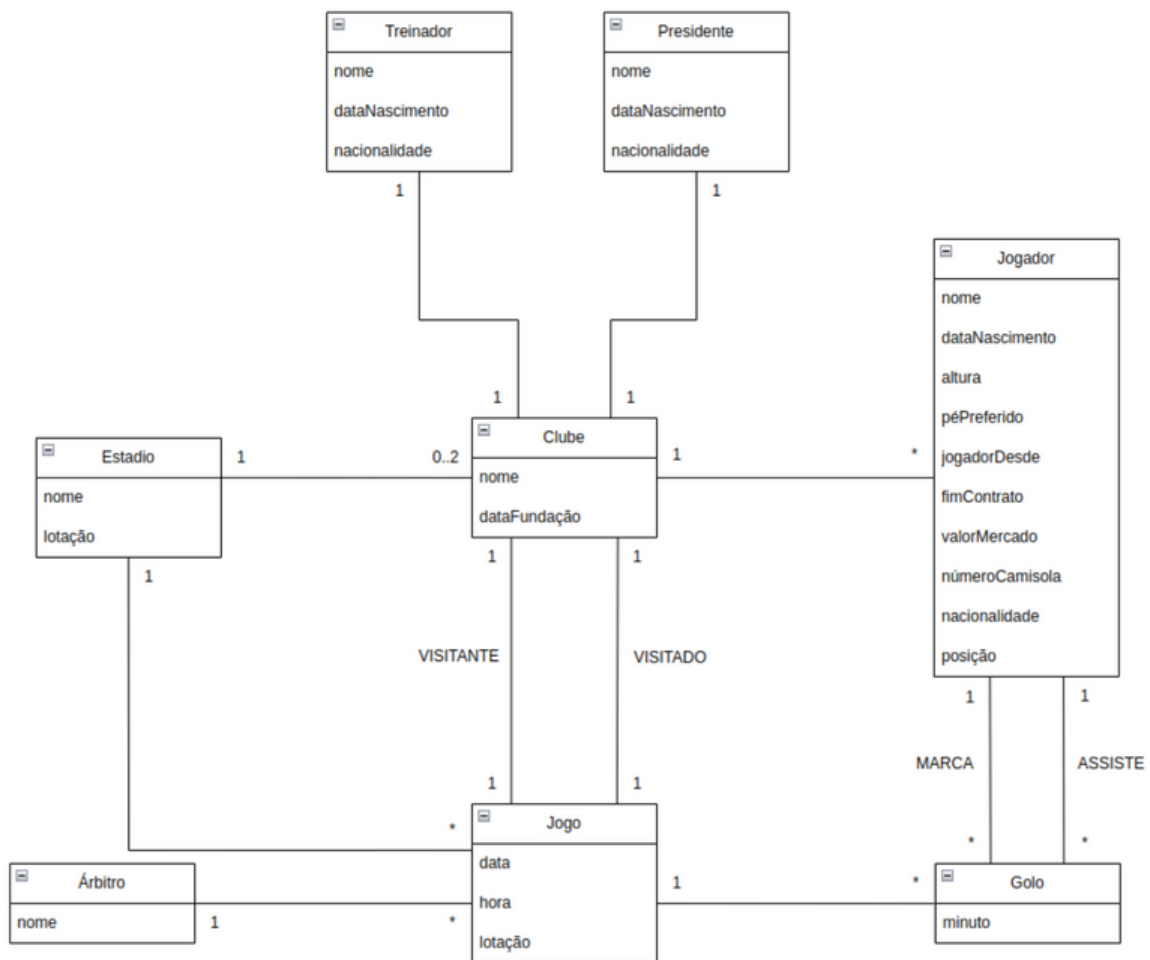


Faculdade de Engenharia da  
Universidade do Porto

# Projeto Base de Dados

Elaborado por:  
Carlos Daniel Lopes Rebelo  
Diogo Tomás Valente Fernandes  
Jaime Francisco Rodrigues Fonseca

# Diagrama UML



# Modelo Relacional

**Estadio** ( idEstadio, nome, lotacao )

**Clube** ( idClube, nome, dataFundacao, idEstadio → estadio )

**Presidente** ( idPresidente, nome, dataNascimento, nacionalidade, idClube → Clube )

**Treinador** ( idTreinador, nome, dataNascimento, nacionalidade, idClube → Clube )

**Jogador** ( idJogador, nome, dataNascimento, altura, pePreferido, jogadorDesde, fimContrato, valorMercado, numeroCamisola, nacionalidade, posicao, idClube → Clube )

**Arbitro** ( idArbitro, nome )

**Jogo** ( idJogo, data, hora, lotação, idArbitro → Arbitro, idEstadio → Estadio, idEquipaVisitada → Equipa, idEquipaVisitante → Equipa )

**Golo** ( idGolo, minuto, idMarcador → Jogador, idAssistente → Jogador, idJogo → Jogo )

# Contexto

Neste capítulo forneceremos uma breve explicação sobre cada uma das classes utilizadas no nosso trabalho:

- **Estádio** - Possui um ID, o nome oficial do estádio e a sua lotação máxima. Em vez de adicionarmos, também, uma foreign key para a class Clube, optámos por adicionar na classe Clube uma foreign key para a classe Estádio uma vez que, entre os estádios, o número de Clubes aos quais estão associados varia entre zero e dois;
- **Clube** - Possui um ID, o nome do clube, data de fundação e uma foreign key o estádio associado a este clube (notar que, como referido anteriormente, poderão existir dois clubes com o mesmo estádio);
- **Presidente** - Possui um ID, o nome do presidente, a sua data de nascimento, a sua nacionalidade e, por fim, como foreign key, o ID do seu clube;
- **Treinador** - Possui um ID, o nome do presidente, a sua data de nascimento, a sua nacionalidade e, por fim, como foreign key, o ID do clube onde treina;
- **Jogador** - Possui um ID, o nome do jogador, a sua data de nascimento, a sua altura, o seu pé preferido, a data referente à sua entrada no clube atual, a data do final do seu contrato, o seu valor no mercado de transferências, o número da sua camisola, a sua nacionalidade, a posição em que joga e, por, fim, como foreign key, o ID do clube onde joga;
- **Árbitro** - Possui um ID e o nome do árbitro;
- **Jogo** - Possui um ID, a data em que ocorreu o jogo, a hora a que ocorreu o jogo, a lotação (número de adeptos que ao jogo no estádio) e quatro foreign keys com a informação relativa ao árbitro do jogo, ao estádio em que se fez o jogo, à equipa visitada e à equipa visitante;
- **Golo** - Possui informação relativa a um golo. ID e minuto a que foi marcado. Para além disso possui 3 *foreign keys* distintas : idJogo que permitirá aceder à informação de uma classe jogo, idMarcador que permitirá aceder à informação da classe do jogador que marcou o jogo e o idAssistente com a informação relativa ao jogador que assistiu o golo.

# Análise de dependências funcionais e formas normais

- **Estadio(idEstadio, nome, lotacao)**

**FDs:**

idEstadio -> nome, lotacao

Formas: BCNF: Sim | 3NF: Sim

- **Clube(idClube, nome, anoFundacao, idEstadio -> Estadio)**

**FDs:**

idClube -> nome, anoFundação, idEstadio

nomeClube -> idClube, anoFundação, idEstadio

Formas: BCNF: Sim | 3NF: Sim

- **Presidente(idPresidente, nome, dataNascimento, nacionalidade, idClube -> Clube)**

**FDs:**

idPresidente -> nome, dataNascimento, nacionalidade, idClube

Formas: BCNF: Sim | 3NF: Sim

- **Treinador(idTreinador, nome, dataNascimento, nacionalidade, idClube -> Clube)**

**FDs:**

idTreinador -> nome, dataNascimento, nacionalidade, idClube

Formas: BCNF: Sim | 3NF: Sim

- **Jogador(idJogador, nome, idade, altura, pePreferido, jogadorDesde, fimContrato, valorMercado, numeroCamisola, nacionalidade, posicao, idClube -> Clube)**

**FDs:**

idJogador -> nome, idade, altura, pePreferido, jogadorDesde, contrato, valorMercado, numeroCamisola, nacionalidade, posição, idClube

Formas: BCNF: Sim | 3NF: Sim

# Análise de dependências funcionais e formas normais

- **Arbitro(idArbitro, nome)**

FDs:

idArbitro -> nomeArbitro

Formas: BCNF: Sim | 3NF: Sim

- **Jogo(idJogo, data, hora, lotacao, idArbitro -> Arbitro, idEstadio -> Estadio, idEquipaVisitada -> Equipa, idEquipaVisitante -> Equipa)**

FDs:

idJogo -> dataJogo, horaJogo, lotacao, nomeArbitro, idEstadio, idEquipaVisitada, idEquipaVisitante

dataJogo, horaJogo, idEstadio -> idJogo, lotacao, nomeArbitro, idEquipaVisitada, idEquipaVisitante

idEquipaVisitada, idEquipaVisitante -> idJogo, dataJogo, horaJogo, lotacao, idArbitro, idEstadio

Formas: BCNF: Sim | 3NF: Sim

- **Golo(idGolo, minuto, idJogo -> Jogo, idMarcador, idAssistente -> Assistente)**

FDs:

idGolo -> minuto, idJogo, idMarcador, idAssistente

Formas: BCNF: Sim | 3NF: Sim

## Conclusões sobre dependências funcionais e formas normais

A “Boyce-Codd Normal Form” (BCNF) diz que uma relação R obedece a esta forma se para todo  $A \rightarrow B$  não trivial, A é superkey/key.

Uma relação está na “3rd Normal Form” (3NF) se para todo  $A \rightarrow B$  não trivial, A é uma key ou superkey ou B é constituído apenas por atributos membros de pelo menos uma chave da relação (atributos primos).

Todas as relações acima expostas seguem tanto a “Boyce-Codd Normal Form” como a “3rd Normal Form” pois observando todas as FDs, observamos que obtemos sempre os atributos da relação partindo da parte esquerda de cada FD (seja um conjunto de atributos A, sendo A uma key).

## Análise de restrições

### Estadio:

- Não podem haver dois estádios com o mesmo idEstadio uma vez que idEstadio é PRIMARY KEY;

Todos os estádios têm de ter um idEstadio e um nome pois são NOT NULL. Estes atributos têm de ser diferentes para todos os estádios pois são UNIQUE;

### Clube:

- Não podem haver dois clubes com o mesmo idClube uma vez que idClube é PRIMARY KEY;
- Todos os clubes têm de ter um idClube, um nome e um idEstadio pois são NOT NULL. Para além disso, o idClube e o nome têm de ser diferentes para todos os clubes por serem UNIQUE;
- O idEstádio faz referência ao estádio onde joga o clube;

### Treinador:

- Não podem haver dois treinadores com o mesmo idTreinador uma vez que idTreinador é PRIMARY KEY;
- Todos os treinadores têm de ter um idTreinador, um nome e um idClube pois são NOT NULL. Para além disso, o idTreinador tem de ser diferente para todos por serem UNIQUE;
- O idClube faz referência ao clube que treina;

## Análise de restrições

### Presidente:

- Não podem haver dois presidentes com o mesmo idPresidente uma vez que idPresidente é PRIMARY KEY;
- Todos os presidentes têm de ter um idPresidente, um nome e um idClube pois são NOT NULL. Para além disso, o idPresidente tem de ser diferentes para todos os presidentes por ser UNIQUE;
- O idClube faz referência ao seu clube;

### Jogador:

- Não podem haver dois jogadores com o mesmo idJogador uma vez que idJogador é PRIMARY KEY;
- Todos os jogadores têm de ter um idJogador ,um nome e um idClube pois são NOT NULL. Este atributo idJogador deve ser diferente para todos por ser UNIQUE;
- idClube faz referência ao clube pelo qual o jogador joga;

### Arbitro:

- Não podem haver dois árbitros com o mesmo idArbitro uma vez que idArbitro é PRIMARY KEY;
- Todos os árbitros têm de ter um idArbitro e um nome pois são NOT NULL. Este atributo idArbitro tem de ser diferente para todos os árbitros por ser UNIQUE;



## Análise de restrições

### Jogo:

- Não podem haver dois jogos com o mesmo idJogo uma vez que o idJogo é PRIMARY KEY;
- Todos os jogos têm de ter um idJogo, um idArbitro, um idEstadio, um idEquipaVisitada e um idEquipaVisitante pois são NOT NULL. Para além disso, o idJogo tem de ser diferente para todos os jogos por ser UNIQUE; idEquipaVisitada e idEquipaVisitante fazem referência às equipas (visitada e visitante, respetivamente), idEstádio faz referência ao estádio onde se realiza o jogo e idArbitro faz referência ao árbitro da partida;

### Golo:

- Não podem haver dois golos com o mesmo idGolo uma vez que idGolo é PRIMARY KEY;
- Todos os golos têm de ter um idGolo, um minuto, um idJogo e um idMarcador pois são NOT NULL. Este atributo idGolo deve ser diferente para todos por ser UNIQUE;
- O idJogo deve fazer referência ao jogo no qual foi marcado tal golo;
- O idMarcador faz referência ao jogador que marcou o golo;
- O idAssistente, se definido, faz referência ao jogador que assistiu para o golo.

# Interrogações à base de Dados

- 1) Obtenção de uma tabela com a classificação das equipas, vitórias, empates e derrotas. Foi usada a view 'RESULTADOS' que se pode encontrar em 'criar.sql'.
- 2) Listagem das equipas que se encontram simultaneamente nas 3 melhores equipas a atacar e nas 3 melhores a defender. No caso da database utilizada, apenas o palmeiras conseguiu estar entre as 3 melhores equipas tanto a defender como a atacar. O critério utilizado foi o número de golos marcados (na categoria de melhor ataque) e o número de golos sofridos (na categoria de melhor defesa).
- 3) Listagem de todos os jogos disputados em abril.
- 4) Obtenção de um histograma, onde serão representados quantos golos foram marcados ao longo de todo o campeonato em cada minuto.
- 5) Obtenção do valor de mercado total de cada uma das equipas do campeonato bem como a média do valor de mercado de um jogador de cada uma das equipas. Ordenação decrescente de acordo com o valor total do plantel e, em caso de empate, por o valor médio de um jogador do plantel.
- 6) Obtenção da percentagem de golos marcados nos minutos de prolongamento em relação aos golos totais marcados por essa mesma equipa. Ordenação feita por ordem decrescente dessa mesma percentagem e, em caso de empate, por ordem alfabética do nome da equipa.
- 7) Listagem dos jogadores estrangeiros por equipa com maior número de golos e assistências entre as jornadas 10 (exclusive) e 20 (inclusive). Este intervalo é manipulável, com uma pequena alteração na query. a última coluna indica se este jogador é - independentemente da nacionalidade- o mais influente da equipa. A ordenação é feita por número de golos e assistências e em case de empate por nacionalidade.
- 8) Listagem dos jogadores da equipa do campeonato cujo presidente é mais velho, e com mais de 1.75m de altura.
- 9) Obtenção da diferença entre as percentagens média de lotação do estádio, de cada uma das equipas do campeonato, dos jogos realizados em casa dessa mesma equipa, ou fora.
- 10) Listagem dos jogos decididos nos minutos de compensação. Ordenação feita pela data de realização do jogo. Foram usadas as views 'RESULTADOS' e 'RESULTADOS\_REG' que podem ser encontradas em 'criar.sql'.

# Triggers

1) Este trigger garante que a contagem de golos sofridos e golos marcados de cada uma das equipas do campeonato é atualizado, aquando da inserção de um novo golo na base de dados.

2) Este trigger verifica se, ao adicionarmos um novo jogo na nossa base de dados, não há um jogo nessa mesma jornada na qual uma das equipas participantes desse novo jogo já tenha jogado. Se houver, este trigger lança uma mensagem de erro, alertando o utilizador.

3) Este trigger verifica se, ao removermos um jogador de uma equipa da nossa base de dados, o número de jogadores na equipa não é inferior a 13. Caso seja, é lançado uma mensagem de erro, alertando o utilizador. Foi usado o 'ON DELETE CASCADE' ao remover um jogador da base de dados.

# Views

*Foram criadas duas views no ficheiro criar.sql:*

- *RESULTADOS: esta view permite analisar, por cada jogo, o nome das equipas visitada e visitante e os golos de cada uma, a data e a hora do jogo.*
- *RESULTADOS\_REG: esta view é semelhante à anterior diferindo apenas no facto de apenas mostrar os resultados do jogo antes da compensação. Decidimos criar esta view para tornar a leitura da query 10 mais fácil uma vez que é necessário recorrer aos resultados dos jogos antes e depois da compensação várias vezes.*