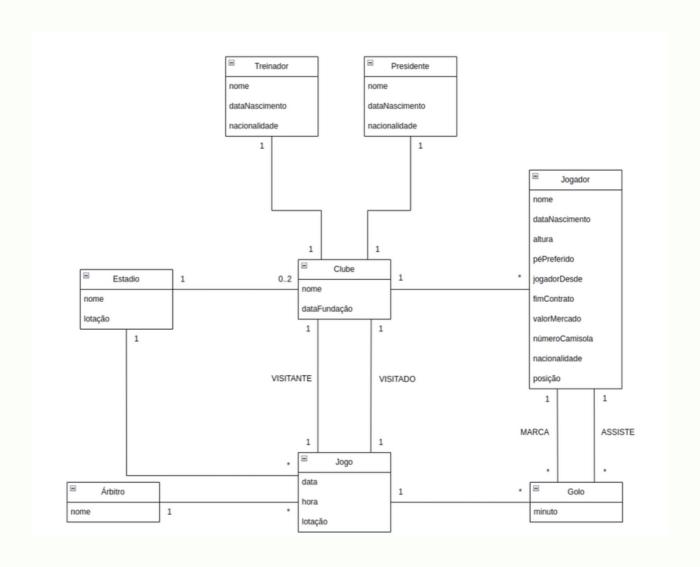
Faculdade de Engenharia da Universidade do Porto

Projeto Base de Dados



Diagrama UML



Modelo Relacional

Estadio (<u>idEstadio</u>, nome, lotacao)

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

Presidente (<u>idPresidente</u>, nome, dataNascimento, nacionalidade, idClube→Clube)

Treinador (<u>idTreinador</u>, nome, dataNascimento, nacionalidade, idClube→Clube)

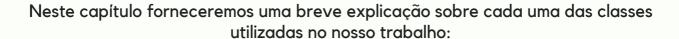
Jogador (<u>idJogador</u>, nome, dataNascimento, altura, pePreferido, jogadorDesde, fimContrato, valorMercado, numeroCamisola, nacionalidade, posicao, idClube→Clube)

Arbitro (<u>idArbitro</u>, nome)

Jogo (<u>idJogo</u>, data, hora, lotação, idArbitro→Arbitro, idEstadio → Estadio, idEquipaVisitada → Equipa, idEquipaVisitante → Equipa)

Golo (<u>idGolo</u>, minuto, idMarcador→Jogador, idAssistente→Jogador, idJogo→Jogo)

Contexto



- **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(<u>idEstadio</u>, nome, lotacao)

FDs:

idEstadio -> nome, lotacao

Formas: BCNF: Sim | 3NF: Sim

_Clube(<u>idClube</u>, nome, anoFundacao, idEstadio -> Estadio)

FDs:

idClube -> nome, anoFundação, idEstadio

<u>nomeClube</u> -> idClube, anoFundação, idEstadio

Formas: BCNF: Sim | 3NF: Sim

Presidente(<u>idPresidente</u>, nome, dataNascimento, nacionalidade, idClube -> Clube)

FDs:

<u>idPresidente</u> -> nome, dataNascimento, nacionalidade, idClube

Formas: BCNF: Sim | 3NF: Sim

• Treinador(<u>idTreinador</u>, nome ,dataNascimento, nacionalidade, idClube -> Clube)

FDs:

idTreinador -> nome, dataNascimento, nacionalidade, idClube

Formas: BCNF: Sim | 3NF: Sim

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

FDs:

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

Formas: BCNF: Sim | 3NF: Sim

Análise de dependências funcionais e formas normais

Arbitro(<u>idArbitro</u>, nome)

FDs:

idArbitro -> nomeArbitro

Formas: BCNF: Sim | 3NF: Sim

Jogo(<u>idJogo</u>, data, hora, lotacao, idArbitro -> Arbitro, idEstadio ->
 Estadio, IdEquipaVisitada -> Equipa, idEquipaVisitante -> Equipa)

 FDs:

<u>idJogo</u> -> dataJogo,horaJogo,lotacao,nomeArbitro,idEstadio,

id Equipa Visita da, id Equipa Visitante

<u>dataJogo,horaJogo,idEstadio</u> -> idJogo,lotacao,nomeArbitro,idEquipaVisitada,

idEquipaVisitante

idEquipaVisitada, idEquipaVisitante -> idJogo, dataJogo, horaJogo,

lotacao,idArbitro,idEstadio

Formas: BCNF: Sim | 3NF: Sim

 Golo(<u>idGolo</u>, minuto, idJogo -> Jogo, idMarcador, idAssistente -> Assistente)

FDs:

<u>idGolo</u> -> 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 -> B não trivial, A é superkey/key.

Uma relação está na "3rd Normal Form" (3NF) se para todo A -> 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 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 que 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.