

Base de Dados

Campeonato Português de Futebol

2LEIC08 – Grupo 801

2022/2023



João Alves	up202108670@fe.up.pt
Daniel Bernardo	up202108667@fe.up.pt
Tiago Oliveira	up202009302@fe.up.pt

Descrição

Este projeto baseia-se na criação de uma base de dados, e todo o processo adjacente, para guardar toda a informação necessária sobre uma época, neste caso da época 2021/2022, da primeira liga de futebol portuguesa. Nesta base de dados tem apenas uma amostra dos dados oficiais da época referida, tendo apenas dados até meio da época desportivo, ou seja, até à décima sétima jornada.

Primeiramente, consideremos a classe **Época**, que servirá apenas para guardar o ano, ou seja, neste caso, o conjunto dos dois anos que representam uma época futebolística. Esta classe apresenta apenas uma ligação com **Jogo**, de modo a associar um Jogo decorrido a uma certa Época.

Em segundo lugar, consideremos a classe **Clube** e as suas interligações. Cada **Clube** está ligado diretamente a todas as outras classes, exceto à classe **Época**. Todos os **Clubes** têm associados **Jogadores**, um **Estádio**, os **Golos** que vão marcando ao longo da época, os **Jogos** realizados e acesso a **Competições** internacionais ou à **Descida** de divisão, para além de terem como atributos o seu nome, a morada e o ano de fundação. A classe **Clube** também apresenta uma ligação à classe **Estádio**, de modo a definir que cada **Clube** terá um **Estádio** associado, onde ir jogar quando for o clube visitado.

Cada **Clube** tem associados diversos **Jogadores**, tal como referido anteriormente. A estes é-lhes atribuída uma quantidade considerável de informação, tal como, o nome, a nacionalidade, a posição no terreno de jogo, o seu pé preferencial e a sua altura. Para além disso, a classe Jogador apresenta interligações com a classe **Golo** de forma a atribuir a marcação de um **Golo** a um **Jogador**, e com **Clube**, pois cada Jogador tem de estar devidamente associado a um **Clube**.

De seguida examinemos a classe **Jogo**, que possui como atributos a jornada e a data de um jogo de futebol. **Jogo** apresenta enumeras interligações com outras classes sendo essas: uma interligação com **Época**, para associar um jogo a uma **Época** (neste trabalho todos os jogos serão referentes à época 2021/2022), uma interligação com **Golo**, para a base de dados associar os **Golos** marcados em cada **Jogo** às restantes classes, uma ligação dupla a **Clube**, pois cada **Jogo** é sempre disputado por dois clubes, sendo sempre um visitante e um visitado e por último uma ligação a **Estádio** para associar um **Jogo** sempre ao **Estádio** do clube visitado.

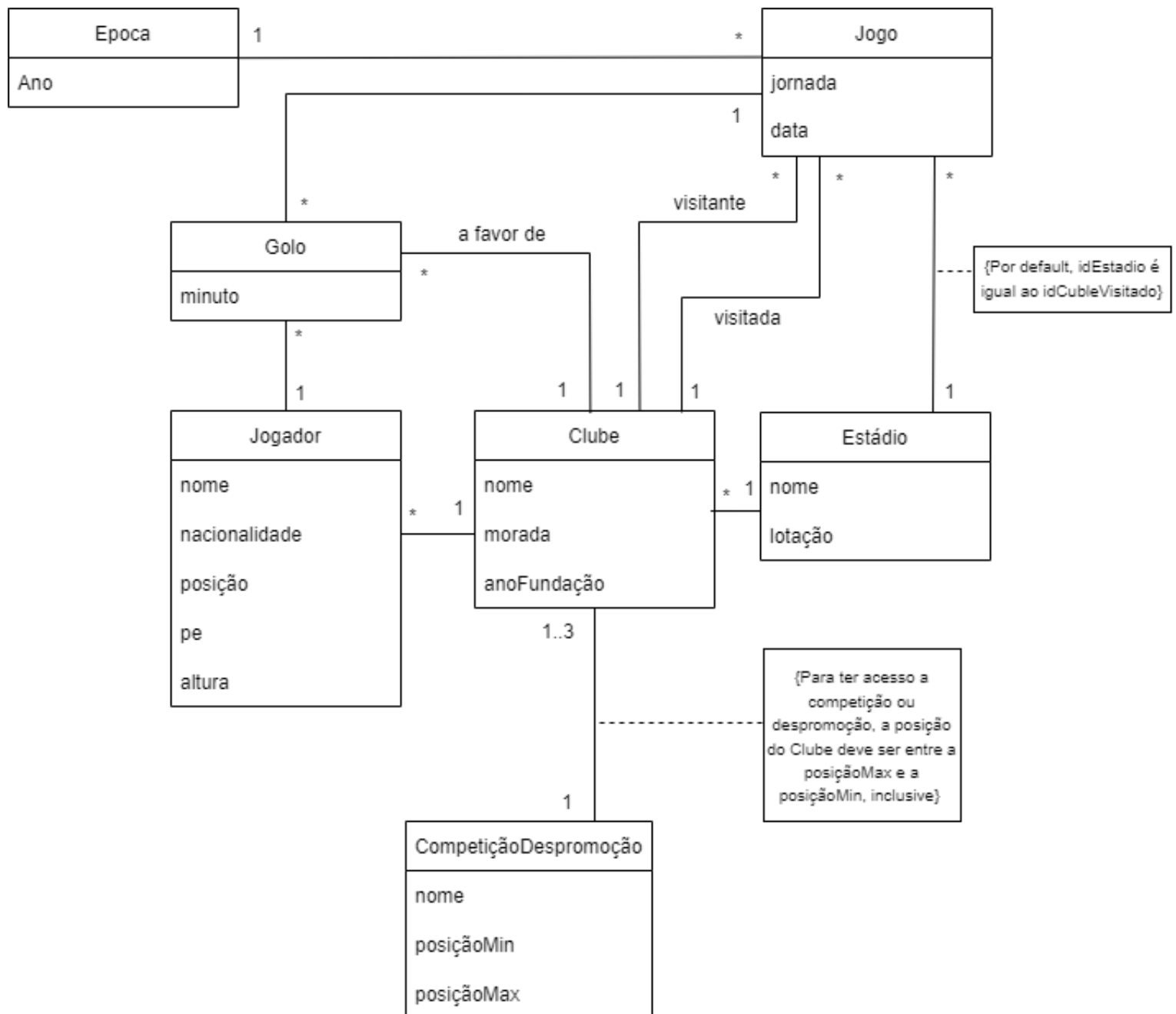
Um **Golo** é, evidentemente, sempre associado a um **Jogador**, a um **Clube** para qual o **Golo** seja a favor (para o caso de um golo for na própria baliza, este irá ser associado ao clube adversário) e também será obviamente associado a **Jogo**, de modo a ser posteriormente possível obter os resultados dos **Jogos**.

Descrição (continuação)

Olhando agora para a classe **Estádio**, vemos que esta possui como atributos, o nome e a sua lotação, que são as características principais de um estádio. Esta classe apresenta uma interligação com **Clube**, pois cada **Estádio** está sempre associado a um **Clube** e outra interligação com **Jogo**, pois cada jogo da liga tem de estar associado ao **Estádio** da equipa da visitada.

Por último, a classe **CompetiçãoDespromoção**, será a classe que contém o nome de uma competição internacional ou despromoção, a posição mínima e posição máxima na tabela classificativa para obter acesso a esta. Esta classe apresenta uma ligação com **Clube** de modo a associar os clubes às competições ou despromoção.

Diagrama de Classes - UML



Para visualização mais nítida, o diagrama UML também está disponível em https://drive.google.com/file/d/15ytO-sDzG05_60MPoCITqk6F-qtvR9i6/view?usp=share_link

Esquema Relacional

Época (idÉpoca, Ano)

Jogo (idJogo, Jornada, Data, idÉpoca -> Época, idEstádio -> Estádio, IdClubeVisitado -> Clube, IdClubeVisitante -> Clube)

Golo (idGolo, Minuto, idJogo -> Jogo, idJogador -> Jogador, idClube -> Clube)

Clube (idClube, Nome, Morada, AnoFundação, idEstádio -> Estádio)

Jogador (idJogador, Nome, Nacionalidade, Posição, Pé, Altura, idClube -> Clube)

Estádio (idEstádio, Nome, Lotação)

CompetiçãoDespromoção (idCompetiçãoDespromoção, Nome, PosiçãoMin, PosiçãoMax, idClube -> Clube)

Dependências Funcionais

Época (idÉpoca, Ano)

1. { idÉpoca } -> { Ano }

Jogo (idJogo, Jornada, Data, idÉpoca -> Época, idEstádio -> Estádio, IdClubeVisitado -> Clube, IdClubeVisitante -> Clube)

1. { idJogo } -> { Jornada, Data, idÉpoca -> Época, idEstádio -> Estádio, IdClubeVisitado -> Clube, IdClubeVisitante -> Clube }

Golo (idGolo, Minuto, idJogo -> Jogo, idJogador -> Jogador, idClube -> Clube)

1. { idGolo } -> { Minuto, idJogo -> Jogo, idJogador -> Jogador, idClube -> Clube }

Clube (idClube, Nome, Morada, AnoFundação, idEstádio -> Estádio)

1. { idClube } -> { Nome, Morada, AnoFundação, idEstádio -> Estádio }
2. { Nome } -> { idClube, Morada, AnoFundação, idEstádio -> Estádio }
3. { Morada } -> { idClube, Nome, AnoFundação, idEstádio -> Estádio }

Jogador (idJogador, Nome, Nacionalidade, Posição, Pé, Altura, idClube -> Clube)

1. { idJogador } -> { Nome, Nacionalidade, Posição, Pé, Altura, idClube -> Clube }

Estádio (idEstádio, Nome, Lotação)

1. { idEstádio } -> { Nome, Lotação }
2. { Nome } -> { idEstádio, Lotação }

CompetiçãoDespromoção (idCompetiçãoDespromoção, Nome, PosiçãoMin, PosiçãoMax, idClube -> Clube)

1. { idCompetiçãoDespromoção } -> { Nome, PosiçãoMin, PosiçãoMax, idClube -> Clube }

Formas Normais

Para concluirmos que uma relação se apresenta na Terceira Forma Normal, é necessário assegurar a regra de não existir transitividade. Se, por algum motivo, esta regra não se mostrar verdadeira para qualquer relação, então esta também não se irá encontrar na forma de Boyce-Codd, pois a forma BCNF é apenas uma versão ligeiramente mais restrita do que a Terceira Forma Normal.

Como se pode verificar, todas as relações presentes no modelo relacional seguem a forma de Boyce-Codd, tal como a Terceira Forma Normal. Analisando as definições das duas formas concluímos que:

Uma relação está na Terceira Forma Normal se para cada $S \rightarrow A$ não trivial, S é uma superkey ou uma key da relação ou A consiste em apenas em atributos que são membros, de pelo menos, uma chave da relação;

Uma relação está na forma de Boyce-Codd se para cada $S \rightarrow A$ não trivial, S é uma superkey ou key;

Como para todas as relações existentes, a partir da parte da esquerda de todas as dependências funcionais (o conjunto de atributos S explicitado nas definições das formas) se consegue obter todos os atributos presentes em cada relação, então com isto dá para concluir que em todos os casos, o atributo S do exemplo, é sempre ou uma superkey ou uma key da relação, e deste modo é possível justificar que todas as relações se apresentam tanto na forma de Boyce-Codd (ou BCNF) tanto como na Terceira Forma Normal.

Lista e Implementação das Restrições

Para haver um correto funcionamento da base de dados, é necessário fornecer a esta restrições para assegurar mais segurança durante a sua utilização. Deste modo, nesta base de dados recorreu-se à utilização de diversas restrições do tipo chave, do tipo chave estrangeira e do tipo NOT NULL, CHECK e UNIQUE.

Começando por explicitar o uso da restrição **CHECK**, esta foi sempre usada com a ideia de restringir os valores que certos atributos poderiam obter, de modo a obter um melhor controlo sobre os dados inseridos na base de dados. A lista inclui:

- Atributo lotação na classe Estádio tem um limite mínimo de 5000 (lotação ≥ 5000), que é a lotação mínima que um estádio de uma equipa da primeira liga portuguesa deve dispor de modo a poder receber jogos no seu estádio;
- Atributo minuto na classe Golo tem um limite mínimo de 0 (minuto > 0), pois os golos de jogos de futebol são sempre contabilizados no minuto imediatamente a seguir, mal a contagem dos segundos volte a chegar a zero. Ou seja, caso um golo fosse marcado aos dez segundos de jogo, o golo seria contabilizado no minuto um, e nunca no mínimo zero ou inferior.

A restrição **UNIQUE** tem como vista garantir que não existe nenhum atributo com igual valor ao mesmo atributo de outro objeto existente na base de dados. Esta foi a aplicada a:

- Atributo nome da classe Clube – não podem existir dois clubes com exatamente ao mesmo nome;
- Atributo morada da classe Clube – a base de dados não permite o registo de dois clubes com exatamente a mesma morada (neste caso, a morada fiscal apresentada no site dos clubes não pode ser igual);
- Atributo nome da classe Estádio – não existem dois estádios com precisamente o mesmo nome.

De seguida, a restrição **NOT NULL** que quando é associada a um atributo, torna-se obrigatória a existência desse atributo para a base de dados conseguir criar um objeto do tipo da classe à qual o atributo com NOT NULL está associado. A lista de uso desta restrição é a seguinte:

- Os atributos idJogo, idJogador e idClube da classe Golo são declarados com NOT NULL, pois sem eles seriam impossível associar um golo ao seu marcador, ao jogo em que aconteceu ou ao clube que favoreceu;
- Os atributos básicos de um Jogador também são todos declarados como NOT NULL, devido ao facto de que não faz sentido esses atributos serem nulos quando se refere a um jogador de futebol profissional em Portugal. Esses atributos são: o nome, a nacionalidade, a posição, o pé preferencial, altura e o idClube do Clube que representam;
- Os atributos jornada, idÉpoca, idEstádio, idClubeVisitante e idClubeVisitado da classe Jogo também são todos classificados como NOT NULL, pois se estes fossem nulos era impossível definir com precisão quando os jogos tinham acontecido, onde tinham acontecido e quais os intervenientes do jogo;

- Os atributos anoFundação e idEstádio da classe Clube também são restringidos como NOT NULL, pois todos os clubes têm quer ter associados a si um estádio no qual irão jogar os jogos como estatuto de visitado e todos os clubes têm de possuir uma data de fundação;

- O atributo nome, posiçãoMin e posiçãoMax da classe CompetiçãoDespromoção possuem a restrição NOT NULL, pois todas as competições internacionais têm um nome, e também, como é evidente, a despromoção e o playoff de despromoção têm nomes associados, que são estes que referi agora. Para além disso, as competições têm sempre de ter associados os lugares necessários a ser ocupados por um clube na tabela classificativa, de modo a terem o acesso;

- Por último uma Época tem o atributo ano com restrição NOT NULL, pois todas as épocas têm de ter associadas a si um ano (neste caso como as épocas de futebol em Portugal são jogadas em dois anos civis, as épocas serão no formato exemplo: "21/22") .

Agora veremos as **PRIMARY KEYS**, que foram aplicadas a cada classe, de modo a garantir mais segurança aos utilizadores da base de dados, impedindo a criação de dois objetos como a mesma PRIMARY KEY. A lista é muito simples e trivial, pois para cada classe foi criado um atributo do tipo id<NomeDaClasse>, para servir de chave primária:

- Para as classes Época, Estádio, CompetiçãoDespromoção, Clube, Jogo, Jogador e Golo foi criado um atributo do tipo INT idÉpoca, idEstádio, idCompetiçãoDespromoção, idClube, idJogo, idJogador e idGolo, associados respetivamente às classes, para servir como chave primária.

Por último, relativamente às restrições do tipo **FOREIGN KEY**, aplicamos chaves estrangeiras a classes diretamente relacionadas com outras. Todas estas relações possuem também uma restrição do tipo ON UPDATE CASCADE, para o caso de que se algo mudar numa classe um, por exemplo, a classe dois que possui a chave estrangeira da classe um, também será atualizada com os novos valores da classe um. A lista de utilização é a seguinte:

- Na classe Clube foi usada a FOREIGN KEY idEstádio, de modo, a ser possível associar um Clube ao Estádio no qual joga os jogos disputados no estatuto de visitado.
- Em Jogo foram usadas quatro FOREIGN KEYS: idEstádio, idÉpoca, idClubeVisitado e idClubeVisitante. Desta maneira, conseguimos associar em que Época é que um jogo é disputado, qual o Clube Visitante e qual o Clube Visitado e em que Estádio é que o jogo decorreu.
- Na classe Jogador, foi usada uma FOREIGN KEY, idClube. Assim conseguimos implementar a relação de um jogador tem um contrato com um Clube.
- De seguida, na classe Golo, foram utilizadas três FOREIGN KEYS, sendo estas: idJogo, idJogador e idClube. Assim, é possível ligar um golo sempre ao Jogador que o marcou, em que Jogo é que aconteceu e qual o Clube que beneficiou (pois pode ser um golo normal ou um autogolo).

Autoavaliação

Opinião Daniel Bernardo:

- Daniel: 33%
- Tiago: 32%
- João: 35%

Opinião Tiago Oliveira:

- Daniel: 35%
- Tiago: 25%
- João: 40%

Opinião João Alves:

- Daniel: 37,5%
- Tiago: 25%
- João: 37,5%