



Campeonato Português de Andebol



1.^a Divisão

Base de Dados 2022/2023 — Licenciatura em Engenharia Informática e Computação:

Docente:

Lázaro Costa

Grupo 606:

João Figueiredo up202108873@fe.up.pt

José Santos up202108729@fe.up.pt

Madalena Ye up202108795@fe.up.pt

Resumo

Este trabalho foi realizado no âmbito da unidade curricular Base de Dados, com a intenção de explorar, criar e interrogar uma base de dados. Esta permitirá gerir os resultados do campeonato português de andebol, da primeira divisão. Pressupõe então a criação de um modelo conceptual relativo ao tema, o mapeamento do mesmo para um esquema relacional e a implementação desse esquema numa base de dados SQLite.

Palavras-Chave

Campeonato de andebol; UML; Modelo Conceptual; Esquema relacional; Formas normais; Dependências funcionais; Interrogações; SQLite; Base de dados;

Agradecimentos

Ao professor Lázaro Costa pela orientação dada e acompanhamento constante durante o desenvolvimento do projeto.

Índice

1. Descrição	5
2. Modelo Conceptual	6
2.1 Diagrama UML	6
3. Esquema Relacional	7
4. Formas Normais e Análise de Dependências Funcionais	8
4.1 Dependências Funcionais	8
4.2 Violações à Terceira Forma Normal e à Forma Normal Boyce-Codd	9
4.3 Restrições	9
5. SQLite	11
5.1 criar.sql	11
5.2 povoar.sql	11
6. Interrogações	11
7. Gatilhos	12
8. Autoavaliação	12

1. Descrição

O **Campeonato Placard Andebol1** – de nome único e caracterizado por ter uma edição, data de início e fim – é a principal competição de Andebol em Portugal, da 1.^a **divisão**. É disputado ao longo de várias **jornadas** (numa só edição). Durante o campeonato, as **equipas** participantes jogam num formato de liga, todos contra todos duas vezes, num sistema de pontos corridos e na condição de **visitado** e **visitante**.

No início da edição é definido um calendário de jogos para cada jornada. Sobre um **jogo**, é relevante saber quando e onde se passou (em que **Pavilhão**), o número de golos marcados pela equipa que joga em casa e o número de golos marcados pela equipa visitante. Além disso, é necessário guardar informações sobre os diferentes tipos de **eventos** que ocorrem no decorrer do jogo. Esses eventos são tipificados (exemplo: golo, cartão azul/amarelo/vermelho, exclusão, etc.). Para todos os eventos, também é pertinente guardar o minuto do jogo em que ocorrem.

No que diz respeito às equipas, cada uma é caracterizada por um nome (do clube, neste caso). Além disso, ao longo de cada jornada, é relevante saber o número de pontos obtidos até à atualidade, o número de vitórias, de derrotas, de empates e o número de golos somados pela equipa. Uma equipa é constituída por 7 jogadores por jogo, sendo convocados 10 a 14 jogadores por jornada.

Sobre um **jogador** deve-se destacar os golos marcados por ele até à altura, o número de jogos em que participou e a sua posição (pivô, ponta, lateral, defesa/central ou guarda-redes). Durante um jogo, é importante armazenar os golos marcados por um determinado jogador e o número de cartões e de exclusões que recebeu.

Numa equipa, os jogadores têm um número de camisola associado e um deles é escolhido para ser o capitão. Há ainda um **treinador** por equipa e **dois árbitros** por jogo. Estes (jogador, treinador e árbitros) são **pessoas** descritas por nome, número de cartão de cidadão, idade e nacionalidade.

No final do campeonato, o primeiro classificado tem acesso à “EHF Champions League” e os seguintes (2.º, 3.º e 4.º) irão disputar a “EHF European League”. Por outro lado, as equipas que terminarem nos dois últimos lugares serão despromovidas à 2.^a Divisão.

2. Modelo Conceptual

2.1 Diagrama UML

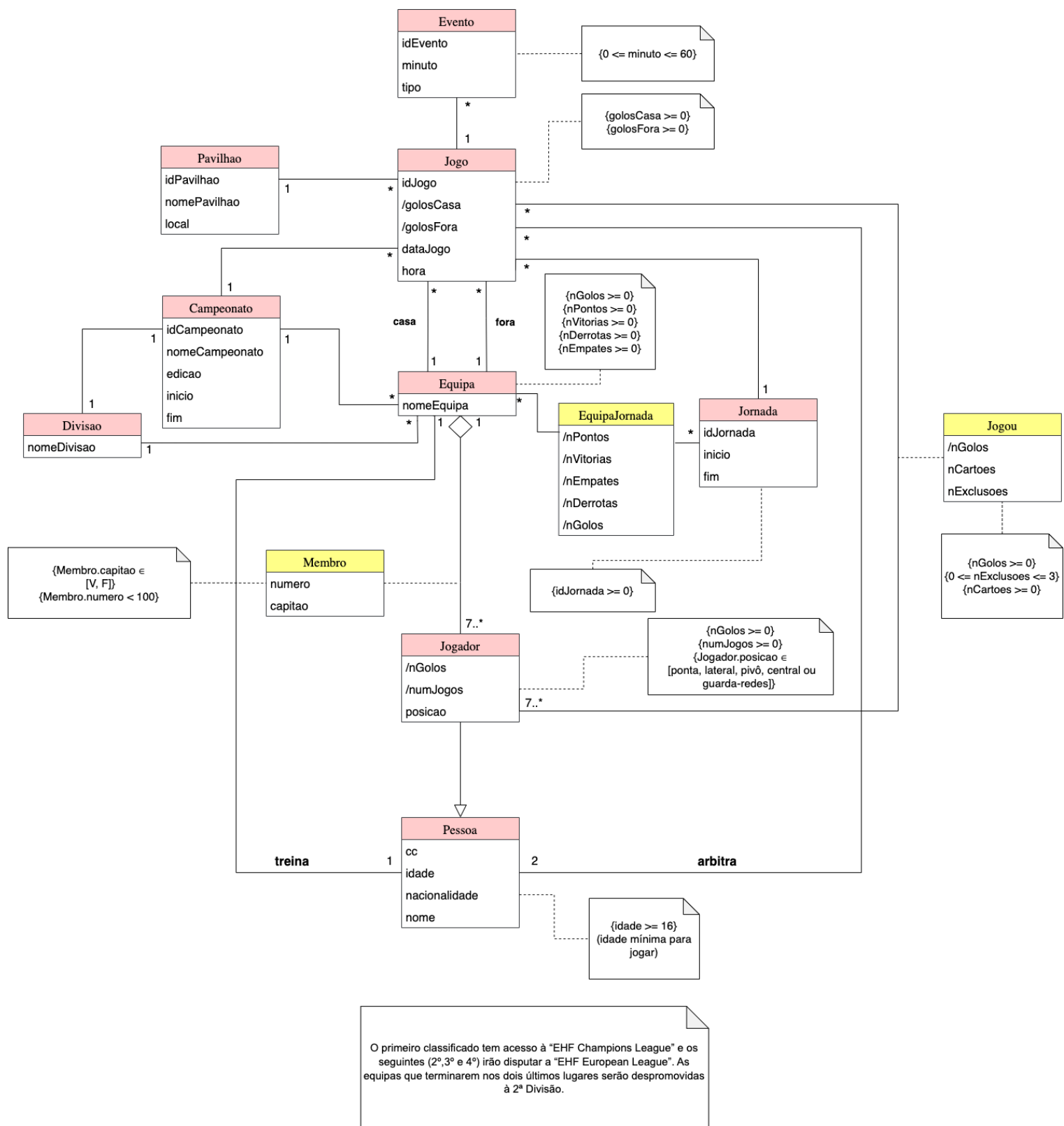


Figura 1. Diagrama UML, disponível em bit.ly/Equipa606_Andebol

3. Esquema Relacional

Evento (idEvento, minuto, tipo, idJogo -> Jogo)

Pavilhao (idPavilhao, nomePavilhao, local)

Divisao (nomeDivisao, idCampeonato -> Campeonato)

Campeonato (idCampeonato, nomeCampeonato, edicao, inicio, fim)

Jornada (idJornada, inicio, fim)

Pessoa (cc, idade, nacionalidade, nome)

Jogador (cc->Pessoa, /nGolos, /numJogos, posicao)

Equipa (nomeEquipa, nomeDivisao->Divisao, idCampeonato->Campeonato, treinador->Pessoa)

Membro (cc->Jogador, nomeEquipa->Equipa, numero, capitao)

Jogo (idJogo, arbitro->Pessoa, golosCasa, golosFora, dataJogo, hora, idPavilhao->Pavilhao, idJornada->Jornada, idCampeonato->Campeonato, equipaCasa->Equipa, equipaFora->Equipa)

Jogou (idJogo->Jogo, cc->Jogador, nGolos, nCartoes, nExclusoes)

EquipaJornada(nomeEquipa->Equipa, idJornada->Jornada, /nPontos, /nVitorias, /nDerrotas, /nEmpates, /nGolos)

Arbitrou(idJogo->Jogo, cc->Pessoa)

4. Formas Normais e Análise de Dependências Funcionais

4.1 Dependências Funcionais

Evento

{idEvento} -> {minuto, tipo, idJogo}

{minuto, tipo, idJogo} -> {idEvento}

Pavilhao

{idPavilhao} -> {nomePavilhao, local}

Divisao:

{nomeDivisao} -> {idCampeonato}

Campeonato

{idCampeonato} -> {nomeCampeonato, edicao, inicio, fim}

{nomeCampeonato, edicao} -> {idCampeonato, inicio, fim}

Jornada

{idJornada} -> {inicio, fim}

{inicio, fim} -> {idJornada}

Pessoa

{cc} -> {idade, nacionalidade, nome}

Jogador

{cc} -> {nGolos, numJogos, posicao, nomeEquipa}

Equipa

{nomeEquipa} -> {nomeDivisao, idCampeonato, nPontos, nVitorias, nDerrotas, nEmpates, nGolos, treinador}

Membro

{cc->Jogador, nomeEquipa} -> {numero, capitao}

Jogo

{idJogo} -> {arbitro, golosCasa, golosFora, dataJogo, hora, idPavilhao, idJornada, idCampeonato, equipaCasa, equipaFora}

{dataJogo, hora, idPavilhao, equipaCasa, equipaFora} -> {idJogo}

Jogou

{idJogo, cc} -> {nGolos, nCartoes, nExclusoes}

EquipaJornada

{nomeEquipa, idJornada} -> {nPontos, nVitorias, nDerrotas, nEmpates, nGolos}

4.2 Violações à Terceira Forma Normal e à Forma Normal Boyce-Codd

Pode-se observar que todas as relações da base de dados seguem a Terceira Forma Normal e a Forma Normal Boyce-Codd. Ao analisarmos a definição destas, é possível afirmar que:

- Uma relação está na Terceira Forma Normal se: para todo $P \rightarrow Q$ não trivial, P é uma “superkey”/”key” ou Q consiste apenas em atributos primos – atributos que são membros de pelo menos uma chave da relação.
- Uma relação está na Forma Normal Boyce-Codd se: para todo $P \rightarrow Q$ não trivial, P é uma “superkey”/”key”.

Assim sendo, todas as relações e FD's da base são satisfeitas, uma vez que o lado direito das relações dependem unicamente do lado esquerdo, que é chave primária. A partir da parte esquerda de cada FD, conseguimos conhecer todos os atributos da relação, implicando assim que P é uma “superkey”/”key”.

4.3 Restrições

- Restrição CHECK:
 - $\text{Jornada.idJornada} \geq 0$;
 - $\text{Pessoa.idade} \geq 16$, já que esta é a idade mínima para se poder jogar na primeira divisão da liga de Andebol;
 - $\text{Jogador.nGolos} \geq 0$; $\text{Jogador.mediaGolos} \geq 0$; $\text{Jogador.numJogos} \geq 0$;
 - $\text{Jogador.posicao} \in [\text{Ponta}, \text{Lateral}, \text{Pivô}, \text{Central}, \text{Guarda-Redes}]$;
 - $\text{EquipaJornada.nGolos} \geq 0$; $\text{EquipaJornada.nPontos} \geq 0$; $\text{EquipaJornada.nVitorias} \geq 0$; $\text{EquipaJornada.nDerrotas} \geq 0$; $\text{EquipaJornada.nEmpates} \geq 0$;
 - $\text{Jogo.golosCasa} \geq 0$; $\text{Jogo.golosFora} \geq 0$; $\text{Jogo.idJornada} \geq 0$;
 - $0 \leq \text{Evento.minuto} \leq 60$;
 - $\text{Jogou.nGolos} \geq 0$; $0 \leq \text{Jogou.nExclusoes} \leq 3$; $\text{Jogou.nCartoes} \geq 0$;
 - $\text{Membro.numero} < 100$;
 - $\text{Membro.capitão} \in [\text{V}, \text{F}]$

- Restrições NOT NULL:

- Na classe **Evento**, quer o tipo e o minuto não podem ser NULL, visto que: (1) sem se saber o tipo do evento, a classe perde o seu propósito; (2) sem o minuto, não seria possível distinguir uma sequência de eventos.

- Na classe **Jogo**, é importante que existam valores para golosCasa e golosFora, de modo a poder realizar-se o jogo de andebol. A data também é relevante, visto que permite distinguir, por exemplo, a jornada em que o jogo se realizou. Do mesmo modo, a hora também é necessária, pois auxilia o utilizador da base de dados a saber quando decorreu/irá decorrer um jogo.

- Na classe **Equipa**, todos os atributos necessitam de ter um valor NOT NULL para se obter informações corretas sobre as equipas e para se poder atribuir resultados a esta equipa na classe **EquipaJornada**.

- Na classe **EquipaJornada**, todos os atributos necessitam de ser NOT NULL, visto que só assim é possível atribuir os resultados dos jogos de maneira a obter a classificação das equipas ao longo das jornadas.

- Na classe **Jogou**, cada atributo precisa de ter um valor superior ou igual a 0, de forma a auxiliar o armazenamento de estatísticas de um determinado jogador.

- Na classe **Campeonato**, todos os atributos requerem valores superiores ou iguais a zero, com a finalidade de permitir uma pesquisa eficiente e útil ao utilizador da base de dados.

- Na classe **Pavilhao**, o local não pode ser NULL. Um campeonato precisa de locais onde realizar os seus jogos.

- Na classe **Membro**, o número da camisola de cada jogador não pode ser NULL, uma vez que são estes que permitem distinguir jogadores entre si, durante o jogo. Além disso, o atributo capitao só retorna o char V ou F, simbolizando verdadeiro ou falso, respetivamente.

- Na classe **Jogador**, os atributos nGolos, mediaGolos e numJogos são importantes no cálculo das estatísticas de cada jogador.

- Na classe **Jornada**, é necessário ter datas de início e fim bem definidas.

- Na classe **Pessoa**, os atributos permitem a distinção entre jogador/treinador/árbitro e ter uma descrição dessa pessoa, portanto não podem ser NULL.

- Restrições UNIQUE:

- O treinador de cada equipa é único. Apenas existe um treinador por equipa, não podendo um treinador treinar várias equipas simultaneamente.

5. SQLite

5.1 [criar.sql](#)

5.2 [povoar.sql](#)

6. Interrogações

A seguinte lista de interrogações aborda questões comuns que alguém possa ter ao gerir a nossa base de dados, pertinentes também para a comprovação do correto funcionamento da mesma.

1. Em média, quantos golos marca um jogador de cada equipa? Indique o nome da equipa e a média, ordenando decrescentemente por nome da equipa. ([Int1.sql](#))
2. Indique o nome da equipa classificada em último ao longo de todas as jornadas que já ocorreram. ([Int2.sql](#))
3. Qual a probabilidade (em percentagem) do FC Porto ganhar o seu próximo jogo? ([Int3.sql](#))
4. Indique o nome e o número de golos marcados pelos 5 melhores marcadores até à 2.^a jornada. ([Int4.sql](#))
5. Entre os jogadores das equipas que estão no top 4 após a 4.^a jornada, indique o nome do jogador que fez mais golos num único jogo, especifique a sua equipa, o n.º de golos marcados e o 'id' desse jogo. Em caso de igualdade, selecione o jogador com mais golos feitos na totalidade. ([Int5.sql](#))
6. Liste todas as estatísticas do capitão de cada uma das equipas, isto é: nome, número da camisola, equipa onde joga, número de golos marcados, número de cartões e número de exclusões. Ordene por ordem crescente. ([Int6.sql](#))
7. Indique o nome do pavilhão, o local, a data, o id do jogo e a jornada em que se passou o jogo onde houve maior discrepância entre o número de cartões e o número de exclusões dado. ([Int7.sql](#))
8. Qual é a média de golos do 2.º melhor marcador da 5.^a melhor equipa do campeonato? Indique o seu nome, nome da equipa em que está e a média dos golos. ([Int8.sql](#))
9. Quantos jogadores estrangeiros tem cada equipa? ([Int9.sql](#))
10. Indique o número de golos dos 6 melhores marcadores de cada equipa cujo nome comece por 'A' ou acabe em 'o'. ([Int10.sql](#))

7. Gatilhos

Os gatilhos a seguir visam evitar diversos problemas de integridade que a base de dados possa ter e ajudam a mantê-la coerente e consistente.

Gatilho 1

O gatilho verifica se a equipa que joga em casa é diferente da equipa que joga como visitante. Se forem iguais, é lançado um erro. Caso contrário, procede-se a averiguar se os jogos a decorrer passam-se no mesmo pavilhão e à mesma data e hora. Novamente, se estiverem a decorrer no mesmo local e à mesma data e hora, é lançado um erro. No final, os dados inseridos são removidos.

[gatilho1_adiciona](#)

[gatilho1_remove](#)

[gatilho1_verifica](#)

Gatilho 2

Quando decorre um novo jogo, o gatilho calcula o número de pontos totais, o número de vitórias, empates e derrotas e o número de golos. Assim, mantém-se a base de dados coerente.

[gatilho2_adiciona](#)

[gatilho2_remove](#)

[gatilho2_verifica](#)

Gatilho 3

Atualiza o número de golos de um jogador, com base nos golos marcados por ele durante um determinado jogo. O número de jogos do jogador também aumento em 1.

[gatilho3_adiciona](#)

[gatilho3_remove](#)

[gatilho3_verifica](#)

8. Autoavaliação

- Consideramos que não houve problemas na realização deste projeto. Todos os elementos do grupo participaram de forma ativa na sua realização, portanto acreditamos que a avaliação entre os três é justa e equitativa.