



# PROJETO DE BASE DE DADOS 2022-2023

Campeonato nacional de andebol

Ana Carolina Ribeirinho Soares Almeida | up202108757

Lara Inês Alves Cunha | up202108876

Sara Isabel Borges da Silva | up202007248

Faculdade de Engenharia da Universidade do Porto

# Índice

<b>A. Definição do Modelo Conceptual .....</b>	<b>2</b>
1. Tema do trabalho e contexto .....	2
2. UML.....	3
<b>B. Modelo Relacional .....</b>	<b>4</b>
<b>C. Análise de Dependências Funcionais e Formais Normais .....</b>	<b>5</b>
<b>D. Adição de restrições à base de dados .....</b>	<b>7</b>
<b>E. Avaliação da participação dos elementos do grupo na entrega 1 .....</b>	<b>9</b>
<b>F. Interrogação da Base de dados .....</b>	<b>10</b>
<b>G. Adição de gatilhos à base de dados .....</b>	<b>11</b>
<b>H. Avaliação da participação dos elementos do grupo na entrega 2 .....</b>	<b>12</b>

## A. Definição do Modelo Conceptual

### 1. Tema do trabalho e contexto

Pretende-se armazenar informação sobre o campeonato de andebol português da primeira divisão num determinado ano. Nomeadamente, o estado do campeonato em qualquer jornada, as equipas em condições de ir às competições europeias e em risco de despromoção e o ano em que está a ocorrer.

Este campeonato é constituído por dezasseis equipas, cada uma delas composta por doze jogadores, dos quais sete estão a jogar (um guarda-redes e seis em campo) e cinco suplentes, que acumulam pontos ao longo de trinta jornadas. Sobre cada jogador pretende-se armazenar o seu número.

A entidade designada por jornada tem de ter em vista o número da jornada em que o campeonato está a ocorrer, a data inicial e a data final da mesma.

Relativamente a cada equipa pretende-se guardar o número de pontos que vai ter ao longo do campeonato assim como o número de golos marcados e sofridos. Cada equipa pertence a um clube em que pretende-se saber o seu nome, localidade e a sua data de fundação.

As equipas jogam contra todas as outras duas vezes (uma delas como visitante e outra jogando em casa) e cada um dos jogos é deliberado por uma equipa de árbitros, em que cada elemento dessa equipa tem um estatuto atribuído.

No que diz respeito aos jogadores e aos árbitros pretende-se guardar o seu número de bi, o seu nome, morada e também a sua data de nascimento.

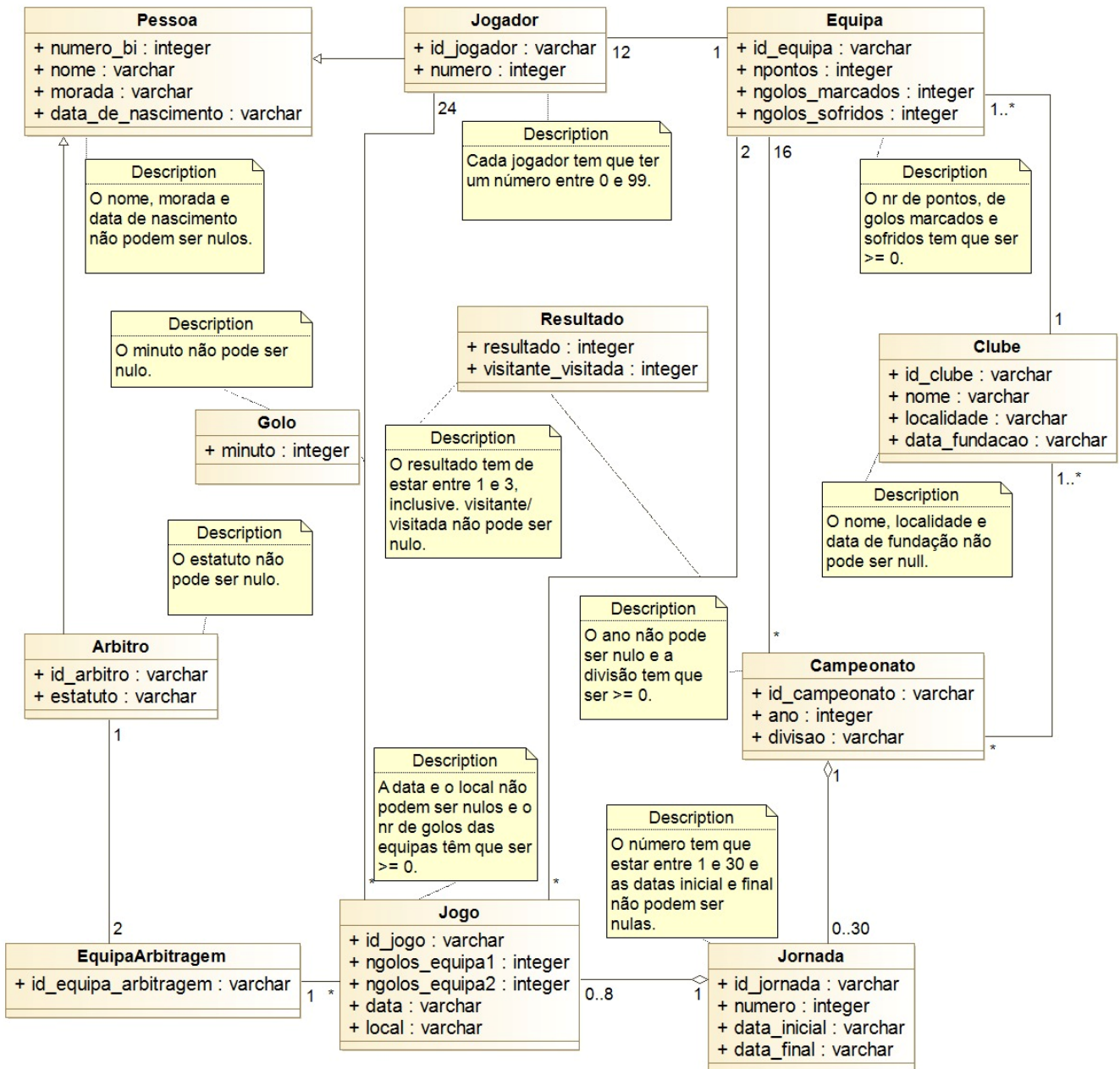
Em todos os jogos as equipas acumulam pontos, sendo que a vitória dá à equipa três pontos, o empate dois e a derrota somente um ponto. Acerca de cada jogo é necessário guardar o número de golos de cada uma das duas equipas, a data em que este ocorre assim como o local em que ocorre o jogo. Cada golo marcado num determinado jogo é necessário guardar o minuto em que ocorre.

O primeiro classificado ao fim deste campeonato, fica automaticamente inscrito para a fase de grupos da *EHF Champions League*. A segunda equipa com melhores resultados fica automaticamente apurada para a fase de grupos da *EHF European League*. O terceiro e o quarto classificados irão disputar um *play-off* com a finalidade de decidir quem integra a fase de grupos desse campeonato.

Por outro lado, a equipa que tiver acumulado menos pontos será despromovida à segunda divisão. Enquanto a equipa da segunda divisão com melhor classificação joga na próxima época desportiva na primeira divisão.

A equipa que fica em 15º lugar, disputa um *play-off* constituído por dois jogos (um em casa e outro na casa da equipa adversária), contra o segundo classificado da segunda divisão, com o objetivo de decidir qual das equipas passa para a primeira divisão na próxima época.

## 2. UML



## B. Modelo Relacional

Pessoa (numero\_bi, nome, morada, data\_de\_nascimento)

Jogador (id\_jogador, numero\_bi->Pessoa, numero)

Equipa (id\_equipa, npontos, ngolos\_marcados, ngolos\_sofridos)

Clube (id\_clube, nome, localidade, data\_fundacao)

Arbitro (id\_arbitro, numero\_bi->Pessoa, estatuto)

EquipaArbitragem (id\_equipa\_arbitragem)

Jogo (id\_jogo, ngolos\_equipa1, ngolos\_equipa2, data, local)

Jornada (id\_jornada, numero, data\_inicial, data\_final)

Campeonato (id\_campeonato, ano, divisao)

Golo (id\_jogador -> Jogador, id\_jogo -> Jogo, minuto)

Resultado (id\_equipa->Equipa, id\_jogo->Jogo, resultado, visitante\_visitada)

campeonatoClube (id\_campeonato -> Campeonato; id\_clube -> Clube)

equipaCampeonato (id\_equipa -> Equipa; id\_campeonato -> Campeonato)

## C. Análise de Dependências Funcionais e Formais Normais

Pessoa (numero\_bi, nome, morada, data\_de\_nascimento)

**Dependências (FD):**

numero\_bi -> nome, morada, data\_de\_nascimento

Jogador (id\_jogador, numero\_bi -> Pessoa, numero)

**Dependências (FD):**

id\_jogador -> numero

numero\_bi -> Pessoa

Equipa (id\_equipa, npontos, ngolos\_marcados, ngolos\_sofridos)

**Dependências (FD):**

id\_equipa -> npontos, ngolos\_marcados, ngolos\_sofridos

Clube (id\_clube, nome, localidade, data\_fundacao)

**Dependências (FD):**

id\_clube -> nome, localidade, data\_fundacao

Arbitro (id\_arbitro, numero\_bi -> Pessoa, estatuto)

**Dependências (FD):**

id\_arbitro -> estatuto

numero\_bi -> Pessoa

EquipaArbitragem (id\_equipa\_arbitragem)

**Dependências (FD):**

Sem dependências!

Jogo (id\_jogo, ngolos\_equipa1, ngolos\_equipa2, data, local)

**Dependências (FD):**

id\_jogo -> ngolos\_equipa1, ngolos\_equipa2, data, local

Jornada (id\_jornada, numero, data\_inicial, data\_final)

**Dependências (FD):**

id\_jornada -> numero, data\_inicial, data\_final

Campeonato (id\_campeonato, ano, divisao)

**Dependências (FD):**

id\_campeonato -> ano, divisao

Formas Normais:

A normalização de um banco de dados tem como principais objetivos a redução de dados, o aumento do seu desempenho e a sua integridade. Assim, examinamos os atributos de cada classe e as relações entre elas com o intuito de evitar anomalias observadas na inclusão, na exclusão e também na alteração de registros do nosso banco de dados.

Efetivamente, existem várias formas de normalizar dados, porém a primeira diz-nos que os atributos de uma dada relação devem ser atômicos. Isto quer dizer que as tabelas não podem ter valores repetidos nem atributos contendo mais de um valor. Para colocar as relações nesta forma primeiro identificamos a chave primária de cada relação assim como a coluna que possui dados repetidos e tivemos que a remover. De seguida, construímos uma tabela com o atributo em questão, não esquecendo de fazer uma relação entre as duas tabelas.

Para colocarmos os dados na segunda forma normal eles devem estar obrigatoriamente na primeira forma normal. Além disso, os atributos que não são chave devem depender unicamente da chave primária da tabela. Assim, os atributos da classe que não são dependentes dessa chave devem ser removidos da classe principal e cria-se uma nova classe para esses dados.

Passando agora à terceira forma normal, os dados devem estar de acordo com a segunda forma normal, à semelhança do que acontece entre a segunda e a primeira forma normal. Todos os atributos da relação também devem estar funcionalmente independentes uns dos outros e, ao mesmo tempo, dependentes exclusivamente da chave primária da tabela. Esta forma foi projetada para melhorar o desempenho do banco de dados e minimizar os custos de armazenamento.

Uma relação encontra-se na forma normal Boyce-Codd se todos os seus atributos na chave dependerem funcionalmente diretamente da chave primária, ou seja, não podem haver dependências entre atributos não chave. Porém nem todas as relações que estão na terceira forma normal se encontram na forma normal Boyce-Codd.

Assim, através da análise às dependências funcionais na nossa base de dados e do conhecimento da normalização de uma base de dados, concluímos que não existem violações à Forma Normal Boyce-Codd nem à 3ª Forma Normal.

## **D. Adição de restrições à base de dados**

Cada pessoa tem um número de bi diferente - restrição chave (PRIMARY KEY). Cada pessoa tem que ter obrigatoriamente um nome, morada e data de nascimento - restrição NOT NULL.

A cada jogador corresponde um id de jogador específico - restrição chave (PRIMARY KEY). Cada jogador é uma pessoa logo tem um número de bi específico - restrição chave (FOREIGN KEY), restrição NOT NULL. Cada jogador pertence a uma equipa logo tem um id equipa associado - restrição chave (FOREIGN KEY), restrição NOT NULL. Cada jogador tem um número na sua equipa entre 0 e 99 - restrição NOT NULL, restrição CHECK.

Cada equipa tem um id de equipa associado - restrição chave (PRIMARY KEY). Cada equipa tem um número de pontos, número de golos sofridos e golos marcados maior ou igual a zero - restrição NOT NULL, restrição CHECK. Cada equipa pertence a um determinado clube com um id específico - restrição chave (FOREIGN KEY), restrição NOT NULL.

Cada clube tem um id de clube associado - restrição chave (PRIMARY KEY). Cada clube tem um nome diferente - restrição chave (UNIQUE). Cada clube tem obrigatoriamente um nome, uma localidade e uma data de fundação - restrição NOT NULL. Cada clube joga todos os anos num determinado campeonato - restrição FOREIGN KEY.

Cada árbitro tem um id de arbitro diferente - restrição chave (PRIMARY KEY). Cada árbitro tem um estatuto - restrição NOT NULL. Cada árbitro é uma pessoa logo tem um número de bi - restrição chave (FOREIGN KEY), restrição NOT NULL. Cada árbitro pertence a uma certa equipa de arbitragem logo tem um certo id da equipa de arbitragem - restrição FOREIGN KEY, restrição NOT NULL.

Cada equipa de arbitragem tem um id de equipa de arbitragem associado - restrição chave (PRIMARY KEY). Cada equipa de arbitragem é constituída por dois árbitros - restrição chave (FOREIGN KEY)

Cada jogo tem um id de jogo específico - restrição chave (PRIMARY KEY). Cada jogo tem que ter uma data e um local - restrição NOT NULL. Cada jogo tem um número de golos de cada equipa associada, que é maior ou igual a 0 - restrição NOT NULL, restrição CHECK. Cada jogo pertence a uma jornada logo tem um id de jornada - restrição chave (FOREIGN KEY), restrição NOT NULL. Cada jogo tem uma certa equipa de arbitragem logo tem um id de equipa de arbitragem - restrição chave (FOREIGN KEY), restrição NOT NULL.

Cada jornada tem um id de jornada associado - restrição chave (PRIMARY KEY). Cada jornada tem um número diferente, compreendido entre 1 e 30 - restrição chave (UNIQUE), restrição CHECK, restrição NOT NULL. Todas as jornadas têm de ter uma data inicial e uma data final- restrição NOT NULL. Cada jornada pertence a um determinado



campeonato logo tem um id campeonato - restrição chave (FOREIGN KEY), restrição NOT NULL.

Cada campeonato tem um id associado - restrição chave (PRIMARY KEY). Cada campeonato decorre num ano específico - restrição NOT NULL. Cada campeonato tem uma divisão, maior que zero - restrição NOT NULL, restrição CHECK.

Cada golo é marcado num minuto específico - restrição NOT NULL. Cada golo é marcado por um único jogador que tem um dado id específico - restrição FOREIGN KEY, restrição NOT NULL. Cada golo pertence a um certo jogo que corresponde a um id específico - restrição FOREIGN KEY, restrição NOT NULL.

Cada resultado pertence a um jogo - restrição chave (FOREIGN KEY). E cada resultado pertence a uma equipa - restrição chave (FOREIGN KEY). No final de cada jogo tem de haver um resultado - restrição NOT NULL. Em cada jogo tem de haver uma equipa visitante e uma equipa visitada - restrição NOT NULL.

## **E. Avaliação da participação dos elementos do grupo na entrega 1**

Todos os elementos deste grupo contribuíram de igual forma para a realização do projeto.

## **F. Interrogação da Base de dados**

1. Quais os jogadores que participaram no campeonato do ano de 2021? Indique o nome, o número.
2. Qual a equipa que foi despromovida para a segunda divisão (ficou em último lugar no campeonato)? Mostre o nome do clube a que pertence, número de golos marcados, número de golos sofridos e o número de pontos.
3. Qual o jogador que marcou mais golos no mesmo jogo? Indique o id do jogador, id do jogo e o número de golos.
4. Quantas derrotas teve cada clube? Ordenando desde o clube com menos derrotas para o que clube com mais. Indique o seu nome, localidade, data de fundação e número total de derrotas ao longo do campeonato chame DERROTAS a esta coluna.
5. Qual o jogo com mais golos? Mostre o local, data do jogo e o número de golos desse jogo.
6. Quais são os nomes dos clubes que não pertencem às localidades de Lisboa e Porto?
7. Quais os nomes dos jogadores do clube CF Os Belenenses?
8. Quantas vitórias teve o clube FC Porto, SL Benfica e Sporting CP?
9. Quais são os jogadores que nasceram antes de 2000? Mostre o seu nome e a sua data de nascimento e o clube a que pertence. Ordene-os por ordem alfabética.
10. Qual a classificação atual do campeonato? Mostre o nome da equipa com os pontos respetivos. Ordene-os por ordem decrescente.

## **G. Adição de gatilhos à base de dados**

### **1. adicionaPontos**

Este trigger tem como objetivo atualizar os resultados das equipas ao longo do campeonato, depois de cada jogo.

Após inserirmos um novo resultado de um jogo (AFTER INSERT), é automaticamente alterado o valor de pontos de cada equipa. O número de pontos será o valor anterior mais o novo resultado

### **2. adicionaJogadorPessoa e adicionaArbitroPessoa**

Quanto ao primeiro gatilho, depois de ser adicionado um novo elemento à classe jogador (AFTER INSERT), um novo número de bi na classe pessoa terá também que ser criado visto que todos os jogadores são pessoas.

Relativamente ao outro gatilho deste ficheiro, após inserirmos um novo árbitro à base de dados, teremos também que inserir um novo número de bi na tabela pessoa, porque os árbitros também são pessoas.

### **3. pessoaUpdateBi e pessoaInsert**

O último trigger verifica antes de adicionarmos uma nova pessoa (jogador ou árbitro) se já está na classe pessoa ou não. Caso esteja ignora-se, senão adiciona-se ao nosso banco de dados.

Antes de inserirmos (BEFORE INSERT) uma nova pessoa na base de dados ou modificarmos (BEFORE UPDADE) algum dos seus atributos, temos de confirmar se não existe nenhuma outra pessoa com o mesmo número de bi na tabela.

## **H. Avaliação da participação dos elementos do grupo na entrega 2**

Todos os elementos do grupo contribuíram de modo igual para a execução do projeto.