

# Relatório Projeto

## Base de Dados – L.EIC012



Trabalho realizado pelo grupo 208, constituído por:

- Guilherme Coutinho – up202108872
- Mário Branco – up202008219
- Luís Sousa – up202005832

## Tópico A

### Descrição do Contexto

O objetivo deste trabalho é desenvolver uma base de dados para gerir os resultados do campeonato brasileiro de futebol, da primeira divisão chamado Brasileirão. Esta tem como função armazenar os resultados dos jogos jornada a jornada, marcadores dos golos, equipas que jogam em casa e como visitante. Também é constituída com informação do estado do campeonato em qualquer jornada, as equipas em condições de ir às competições europeias e em risco de despromoção.

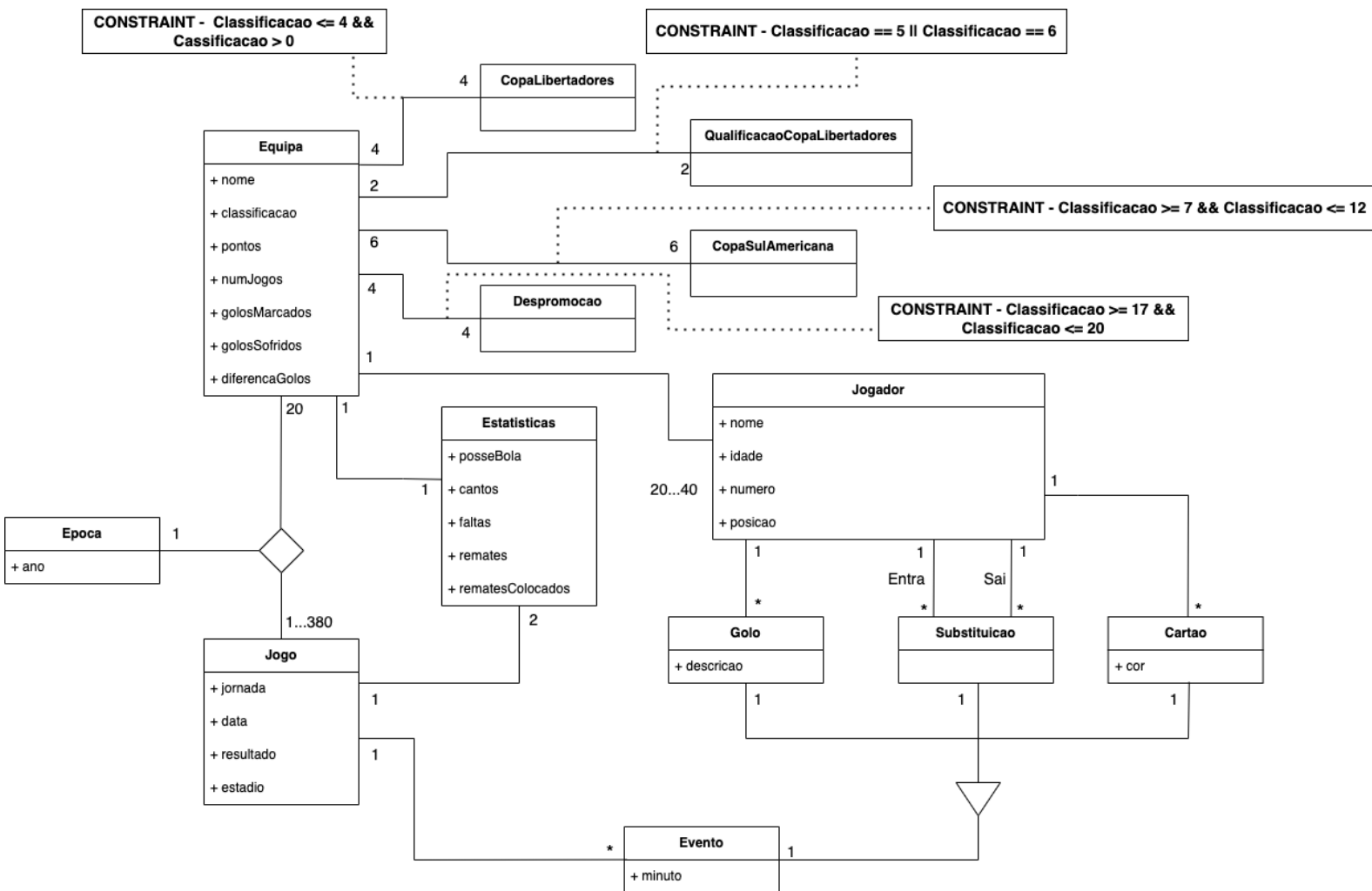
O Brasileirão é constituído por 20 equipas, sendo que cada uma tem cerca de 40 jogadores. Num jogo, jogam 11 jogadores por cada equipa, fazendo assim 22 jogadores dentro de campo, contamos também com eventuais substituições dos jogadores que estão no banco.

São feitos ao todo 380 jogos numa época, sendo assim 38 jornadas com 10 jogos em cada jornada. Duas equipas jogam uma com a outra 2 vezes (como visitante e visitada nos estádios respetivos de quem joga em casa) e têm a chance de pontuar pontos dependendo do resultado do jogo, se uma equipa ganhar, a mesma ganha 3 pontos e a outra não ganha pontos e se o resultado ficar empatado as equipas ganham ambas 1 ponto.

No final da época, existem qualificações e despromoções a fazer, as equipas classificas em:

- 1º, 2º, 3º, 4º, 5º e 6º - estão automaticamente qualificadas para jogar na Copa Libertadores, que é uma competição entre as melhores equipas de futebol profissional da América do Sul;
- 7º e 8º - estão aptas a jogar uma série de jogos com outras equipas sul americanas para disputar um lugar na Copa Libertadores;
- 9º, 10º, 11º, 12º, 13º e 14º - estão automaticamente qualificadas para jogar na Copa Sul-Americana, que é outra competição secundária para as equipas de futebol profissional da América do Sul que não conseguiram entrar na Copa Libertadores;
- 17º, 18º, 19º e 20º - são despromovidas desta liga de futebol e dão lugar a outras 4 equipas de uma liga de futebol brasileira abaixo desta.

# UML



## Tópico B

### Modelo Relacional

Jogador ( idJogador, nome, idade, numero, posicao, idEquipa -> Equipa )

Equipa ( idEquipa, nome, classificacao, pontos, numJogos, golosMarcados, golosSofridos, diferencaGolos )

Epoca ( idEpoca, ano )

Jogo ( idJogo, jornada, dia, resultado, estadio)

JogoEquipaEpoca ( idJogo -> Jogo, idEpoca -> Epoca, idEquipaCasa -> Equipa, idEquipaVisitante -> Equipa )

Estatisticas ( idEstatisticas, posseBola, cantos, faltas, remates, rematesColocados, idEquipa -> Equipa)

Golo ( idGolo, minuto, descricao, idJogador -> Jogador, idJogo -> Jogo)

Outro ( idOutro, minuto, descricao, idJogador -> Jogador , idJogo -> Jogo)

Substituicao ( idSubstituicao, minuto, idJogadorEntra -> Jogador, idJogadorSai -> Jogador, idJogo -> Jogo)

Cartao ( idCartao, minuto, cor, idJogador -> Jogador, idJogo -> Jogo)

EntradaCopaLibertadores ( idEntradaCopaLibertadores, idEquipa -> Equipa )

QualificacaoCopaLibertadores ( idQualificacaoCopaLibertadores, idEquipa -> Equipa )

EntradaCopaSulAmericana ( idEntradaCopaSulAmericana, idEquipa -> Equipa )

Despromocao ( idDespromocao, idEquipa -> Equipa )

## Tópico C

**Jogador:**

IDJogador, Nome, Idade, Numero -> Posicao, IDEquipa

BCN: Sim

3NF: Sim

**Equipa:**

IDEquipa, Nome -> Classificacao, Pontos, NumJogos, GolosMarcados, GolosSofridos, DiferencaGolos

BCN: Sim

3NF: Sim

**Epoca:**

IDEpoca -> Ano

BCN: Sim

3NF: Sim

**Jogo:**

idJogo -> jornada, dia, resultado, estadio

BCN: Sim

3NF: Sim

**JogoEquipaEpoca:**

idJogoEquipaEpoca -> jogo, equipa, época

BCN: Sim

3NF: Sim

**Estatisticas:**

idEstatisticas -> posseBola, cantos, livres, remates, rematesColocados, idEquipa

BCN: Sim

3NF: Sim

**Golo:**

idGolo -> minuto, descricao, idJogador, idJogo

BCN: Sim

3NF: Sim

**Outro:**

idOutro -> minuto, descricao, idJogador, idJogo

BCN: Sim

3NF: Sim

**Substituicao:**

idSubstituicao -> minuto, descricao, idJogador, idJogo

BCN: Sim

3NF: Sim

**Cartao:**

idCartao -> minuto, cor, idJogador, idJogo

BCN: Sim

3NF: Sim

**EntradaCopaLibertadores:**

idEntradaCopaLibertadores -> idEquipa

BCN: Sim

3NF: Sim

**QualificacaoCopaLibertadores:**

idQualificacaoCopaLibertadores -> idEquipa

BCN: Sim

3NF: Sim

**EntradaCopaSulAmericana:**

idEntradaCopaSulAmericana -> idEquipa

BCN: Sim

3NF: Sim

**Despromoção:**

idDespromoção -> idEquipa

BCN: Sim

3NF: Sim

Após se analisar todas as relações, vemos que todas elas estão em 3NF e em BCN, porque todos os membros à esquerda são superkeys, está em 1NF e em 2NF.

## Tópico E

### Restrições

NOT NULL – Usamos esta restrição para garantir que os parâmetros nunca iam ser nulos em todas as tabelas;

PRIMARY KEY – Esta restrição foi utilizada em todas as tabelas no parâmetro do id dessa tabela.

```
CREATE TABLE EQUIPA
(
    IDEQUIPA INTEGER PRIMARY KEY NOT NULL,
```

Foi também utilizada na Relação entre as tabelas EPOCA, JOGO e EQUIPA pois pode haver um id de um jogo igual ao id de outro jogo, mas noutra época, portanto era necessário a chave primária conter os dois parâmetros.

```
CREATE TABLE JOGOEQUIPAEPOCA
(
    IDJOGO INT NOT NULL,
    IDEPOCA INT NOT NULL,
    IDEQUIPACASA INT NOT NULL,
    IDEQUIPAVISITANTE INT NOT NULL,
    PRIMARY KEY(IDJOGO,IDEPOCA), FOREIGN KEY (IDEQUIPACASA) REFERENCES EQUIPA(IDEQUIPA),
    FOREIGN KEY (IDEQUIPAVISITANTE) REFERENCES EQUIPA(IDEQUIPA));
```

FOREIGN KEY – Usada nas tabelas sempre que foi preciso fazer ligação entre tabelas. As chaves estrangeiras são atributos numa tabela que servem de referência para chaves primárias noutra tabela.

Exemplo: Um jogador pertence a uma equipa, logo tem de ter um atributo que será uma chave estrangeira para uma chave primária do id dessa equipa para haver essa ligação entre os dados.

```
CREATE TABLE JOGADOR
(
    IDJOGADOR INTEGER PRIMARY KEY NOT NULL,
    NOME TEXT NOT NULL,
    IDADE INT NOT NULL,
    NUMERO INT NOT NULL,
    POSICAO TEXT NOT NULL,
    IDEQUIPA TEXT NOT NULL,
    FOREIGN KEY (IDEQUIPA) REFERENCES EQUIPA(IDEQUIPA));
```

CHECK – Utilizamos esta restrição para verificar se os valores introduzidos estão dentro do intervalo pretendido.

```
CREATE TABLE ENTRADACOPALIBERTADORES
(
  IDENTRADACOPALIBERTADORES INTEGER PRIMARY KEY NOT NULL,
  IDEQUIPA TEXT NOT NULL,
  CLASSIFICACAO INT NOT NULL CHECK(CLASSIFICACAO <= 4 AND CLASSIFICACAO > 0),
  FOREIGN KEY (IDEQUIPA) REFERENCES EQUIPA(IDEQUIPA),
  FOREIGN KEY (CLASSIFICACAO) REFERENCES EQUIPA(CLASSIFICACAO));
```

## Tópico F

### Carregamento de dados

Para carregar os dados para a Base de dados, utilizamos uma ferramenta do Google Chrome chamada “Web Scraper”. Visitámos o website [zerozero.pt](https://www.zerozero.pt) para conseguirmos retirar toda a informação sobre a época deste ano do Brasileirão.

1)



Criar Novo Sitemap para guardar os dados.



zerozero.pt

Pesquisar

ENTRAR

PORTUGAL COMPETIÇÕES ESTATÍSTICAS MERCADO NOTÍCIAS ENCICLOPÉDIA JOGOS APOSTAS MULTIMÉDIA MEMBROS FUN

AGENDA 26 RELATO 3 TV 14 TAÇA DA LIGA LEAGUE CUP LIGA DOS CAMPEÕES FEM. SCOTTISH PREMIERSHIP BEKER VAN BELGIË COUPE DE FRANCE

|  |   |                                   |   |   |   |   |                                   |
|--|---|-----------------------------------|---|---|---|---|-----------------------------------|
| <br><b>Aberdeen</b><br><b>Rangers</b><br>2<br>3<br>20/12 - 20:00 | <br><b>Académico de ...</b><br><b>Boavista</b><br>2<br>1<br>20/12 - 20:15 | <b>ODDS</b><br><b>PLACARD .pt</b> | <br><b>FC Zurich Fem.</b><br><b>Arsenal Fem.</b><br>0<br>1<br>21/12 - 17:45 | <br><b>Lyon Fem.</b><br><b>Juventus Fem.</b><br>0<br>0<br>21/12 - 17:45 | <br><b>PAS Giannina</b><br><b>Olympiacos</b><br>0<br>0<br>21/12 - 18:30 | <br><b>Besiktas</b><br><b>Saniurfaspor</b><br>0<br>0<br>21/12 - 18:00 | <b>ODDS</b><br><b>PLACARD .pt</b> |
|--|---|-----------------------------------|---|---|---|---|-----------------------------------|

## Campeonato Brasileiro 2022

Brasil | CONMEBOL

2022

Sitemaps Sitemap Create new sitemap

Sitemap name equipa

Start URL https://www.zerozero.pt/edicao.php?id\_edicao=162047

Create Sitemap

3)

Selecionar os dados que vão ser “scraped”.

Depois de ter os ficheiros csv das relações todas, foi necessário “limpar” as linhas, mas como alguns ficheiros tinham milhares de linhas, fabricamos um ficheiro de código em python para apagar os caracteres desnecessários.

Exemplo de código para um dos ficheiros:

```
# script to remove unwanted stuff from a certain file that has been
scraped from
# the web and put in a csv file

def cleanFile(readFile, writeFile, integers):
    read = open(readFile, 'r')
    write = open(writeFile, 'w')
    lines = read.readlines()
    aid = 0

    for line in lines:
        i = 0
        count = 0
        outString = ""
        aid += 1
        fields = line.split(',')

        equipaCasa = fields[1]
        equipaVisitante = fields[2]
        equipasFile = open("EQUIPA.csv", 'r')
        equipas = equipasFile.readlines()
        idEquipaCasa, idEquipaVisitante = 0,0

        for e in equipas:
            if equipas.index(e) == 0:
                continue

            fields_equipas = e.split(',')
            if fields_equipas[1] == equipaVisitante:
                idEquipaVisitante = fields_equipas[0]
            elif fields_equipas[1] == equipaCasa:
                idEquipaCasa = fields_equipas[0]

            outString = fields[0] + ",1," + str(idEquipaCasa) + ',' +
str(idEquipaVisitante) + '\n'
            # step ahead first line
            if lines.index(line) == 0:
                outString =
"idJogo,idEpoca,idEquipaCasa,idEquipaVisitante\n"
            write.write(outString)
            continue

        # count = 0
        # outString = ""
        # for i in range(len(fields)):
        #     if i in integers:
        #         #remove "" from integer fields
        #         fields[i] = fields[i][2:len(fields[i])-1]

        # # remove last " in line
```

```
# fields[len(fields)-1] = fields[len(fields)-1][:len(fields[len(fields)-1])-1]
# rejoin fields in a string
```

```
# add a newline
# outString += '\n'
# write out to the file
write.write(outString)
```

```
# it is necessary to change the file names and the
# indexes of the fields that are integers
# from each file that is going to be cleaned
cleanFile("JOGOEquipaEpoas.csv", "JOGOEQUIPAEPOCA.csv", [4])
```

Assim que acabámos a limpeza dos ficheiros, utilizamos o “DB Browser” uma app que cria um ficheiro de base de dados e guarda toda a informação das tabelas dessa base. Este software permitiu-nos importar os ficheiros csv e colocar a sua informação para as tabelas criadas. Quando a base obteve toda a informação necessária, exportamos a informação toda para um ficheiro sql, neste caso, o “povoar.sql”.

## Tópico G

### Interrogação da base de dados

Interrogação 1 – Listar todos os jogadores da Base de dados (mostrar nome e equipa);

Interrogação 2 – Selecionar as equipas que ficaram no top 5 da classificação final (mostrar nome da equipa e respetiva qualificação);

Interrogação 3 – Selecionar os jogos que ficaram empatados na primeira rodada do campeonato (mostrar resultado, equipas e número da jornada);

Interrogação 4 – Selecionar todos os jogadores que tenham “Silva” no nome (mostrar nome e equipa);

Interrogação 5 – Selecionar todas as equipas com diferença de golos negativa (mostrar equipa, classificação e a diferença de golos);

Interrogação 6 – Selecionar todos os jogos que foram jogados no estádio “Estádio Jornalista Mário Filho (Maracanã) (BRA)” (Mostrar dia do jogo, resultado e equipas que realizaram o jogo);

Interrogação 7 – Selecionar todos os avançados com idade superior a 30 anos (Mostrar nome do jogador, idade e número);

Interrogação 8 – Selecionar as equipas que se qualificaram para a copa Libertadores (Mostrar nome da equipa, classificação e número de pontos);  
Interrogação 9 – Selecionar as 5 equipas com maior média de golos por jogo (Mostrar o nome da equipa, média de golos por jogo, classificação);  
Interrogação 10 – Selecionar todos os jogos realizados no dia 2022-05-12. (Mostrar resultado, equipas que jogaram e estádio em que o jogo foi realizado).

## Tópico H

### Adição de gatilhos à base de dados

Na realização deste trabalho foram usados três gatilhos. Estes têm como principal objetivo introduzir restrições na base de dados. A lista de gatilhos utilizados é neste trabalho é a seguinte:

1. Quando se adiciona um jogador, temos de garantir que este tem um número válido (entre 1 e 99). Este gatilho garante que essa condição é cumprida uma vez que ao tentar adicionar um jogador com um número inválido é apresentada uma mensagem de erro e o jogador não é adicionado à tabela.
2. Quando é inserido um novo golo, o gatilho verifica se o jogador que marcou o golo já existe na tabela de jogadores. Se este não existir é levantado um erro a dizer que o jogador não existe e não é inserido o golo.
3. Quando se adiciona um jogador, é verificada a sua idade. Se este possuir uma idade igual ou superior a 50 anos é levantado um erro e não é possível adicionar o jogador em questão.

## Participação

Guilherme Coutinho (up202108872) – 37,5%

Mário Branco (up202008219) – 37,5%

Luís Sousa (up202005832) – 25%