

Base de Dados

Campeonato Nacional de Basquetebol

2ª entrega



Bruno Drumond (up201202666)

Gonalo Nadais de Pinho (up202108672)

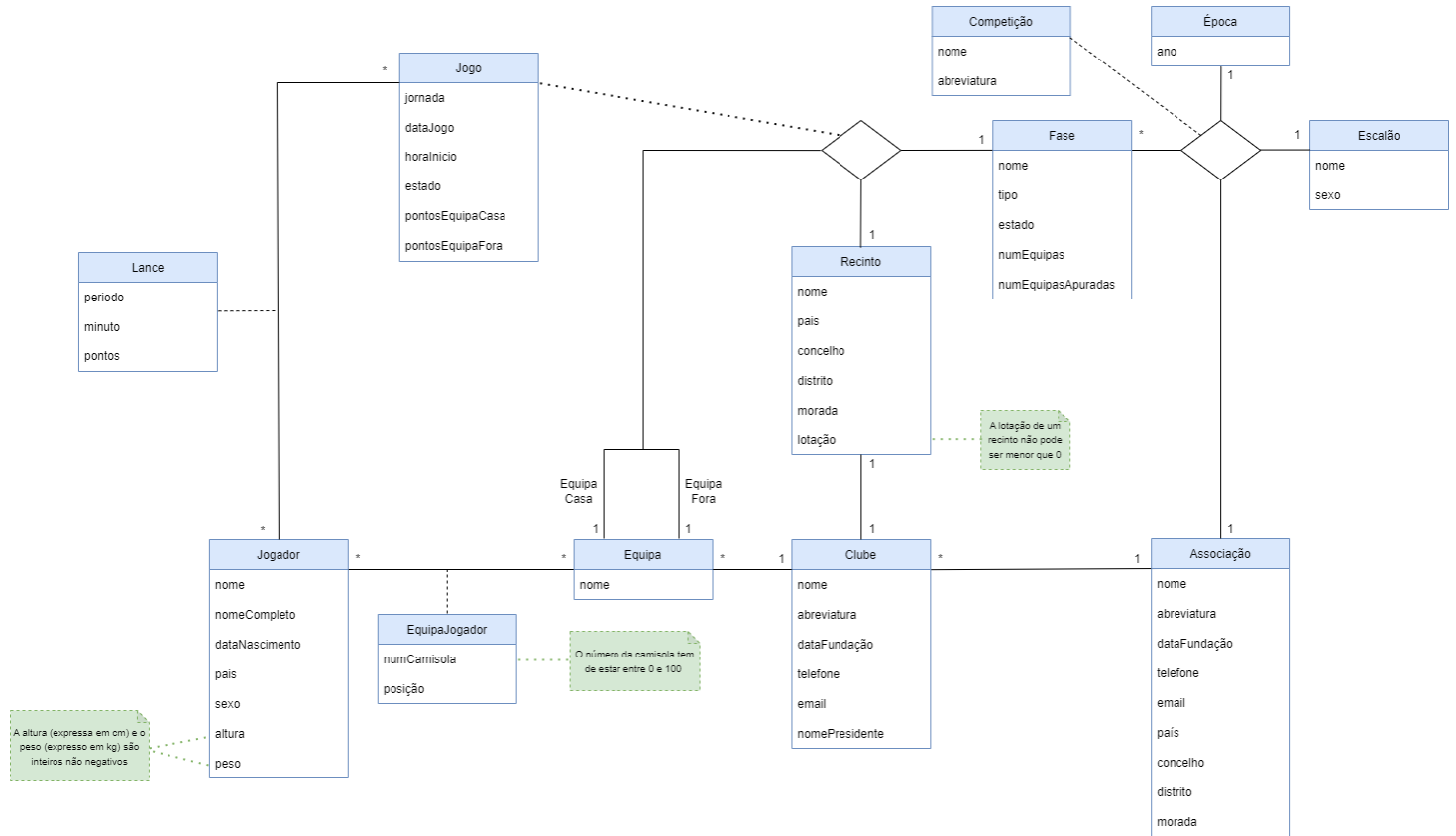
Tomás Sucena Lopes (up202108701)

Dezembro, 2022

Índice

Índice	2
Modelo UML	3
Modelo relacional	4
Análise das Dependências Funcionais e Formas Normais	5
Restrições	7
Interrogações	12
Gatilhos	13

Modelo UML



Modelo relacional

Época(idÉpoca, ano)

Escalão(idEscalão, nome, sexo)

Associação(idAssociação, nome, abreviatura, dataFundação, telefone, email, país, concelho, distrito, morada)

Competição(idCompetição, nome, abreviatura, idÉpoca -> Época, idEscalão -> Escalão, idAssociação -> Associação)

Fase(idFase, nome, tipo, estado, numEquipas, numEquipasApuradas, idCompetição -> Competição)

Recinto(IdRecinto, nome, morada, idConcelho, idDistrito, lotação)

Clube(idClube, nome, abreviatura, dataFundação, telefone, email, nomePresidente, país, concelho, distrito, morada, idAssociação -> Associação, idRecinto -> Recinto)

Equipa(idEquipa, nome, idEscalão -> Escalão, idClube -> Clube)

FaseEquipa(idFase, idEquipa, pontuacao, numJogos, vitorias, derrotas, faltasComparencia, pontosMarcados, pontosSofridos)

Jogador(idJogador, nome, nomeCompleto, dataNascimento, país, sexo, altura, peso)

EquipaJogador(idEquipa -> Equipa, idJogador -> Jogador, numCamisola, posição)

Jogo(idJogo, IdFase -> Fase, jornada, dataJogo, horaInicio, estado, idRecinto -> Recinto, idEquipaCasa -> Equipa, idEquipaFora -> Equipa, pontosEquipaCasa, pontosEquipaFora)

Lance(idLance, idJogador, idJogo, periodo, minuto, pontos)

Análise das Dependências Funcionais e Formas Normais

- **Época**(idÉpoca, ano):
 - FDs: idÉpoca -> ano
- **Escalão**(idEscalão, nome, sexo):
 - FDs: idEscalão -> nome, sexo
- **Associação**(idAssociação, nome, abreviatura, dataFundação, telefone, email, país, concelho, distrito, morada):
 - FDs: idAssociação -> nome, abreviatura, dataFundação, telefone, email, país, concelho, distrito, morada
- **Competição**(idCompetição, nome, abreviatura, idÉpoca -> Época, idEscalão -> Escalão, idAssociação -> Associação):
 - FDs: idCompetição -> nome, abreviatura, idÉpoca, idEscalão, idAssociação
- **Fase**(idFase, nome, tipo, estado, numEquipas, numEquipasApuradas, idCompetição -> Competição):
 - FDs: idFase -> nome, tipo, estado, numEquipas, numEquipasApuradas, idCompetição
- **Recinto**(idRecinto, nome, morada, idConcelho, idDistrito, lotação):
 - FDs: idRecinto -> nome, morada, idConcelho, idDistrito, lotação
- **Clube**(idClube, nome, abreviatura, dataFundação, telefone, email, nomePresidente, país, concelho, distrito, morada, idAssociação -> Associação, idRecinto -> Recinto):
 - FDs: idClube -> nome, abreviatura, dataFundação, telefone, email, nomePresidente, país, concelho, distrito, morada, idAssociação, idRecinto
- **Equipa**(idEquipa, nome, idEscalão -> Escalão, idClube -> Clube):
 - FDs: idEquipa -> idEscalão, idClube

- **FaseEquipa**(idFase, idEquipa, pontuacao, numJogos, vitorias, derrotas, faltasComparencia, pontosMarcados, pontosSofridos):
 - FDs: idFase, idEquipa -> pontuacao, numJogos, vitorias, derrotas, faltasComparencia, pontosMarcados, pontosSofridos
- **Jogador**(idJogador, nome, nomeCompleto, dataNascimento, país, sexo, altura, peso):
 - FDs: idJogador -> nome, nomeCompleto, dataNascimento, país, sexo, altura, peso
- **EquipaJogador**(idEquipa -> Equipa, idJogador -> Jogador, numCamisola, posição):
 - FDs: idEquipa, idJogador -> numCamisola, posicao
- **Jogo**(idJogo, IdFase -> Fase, jornada, dataJogo, horaInicio, estado, idRecinto -> Recinto, idEquipaCasa -> Equipa, idEquipaFora -> Equipa, pontosEquipaCasa, pontosEquipaFora):
 - FDs: idJogo -> IdFase, jornada, dataJogo, horaInicio, estado, idRecinto, idEquipaCasa, idEquipaFora pontosEquipaCasa, pontosEquipaFora
- **Lance**(idLance, idJogador, idJogo, periodo, minuto, pontos):
 - FDs: idLance -> idJogador, idJogo, periodo, minuto, pontos

Analisemos as definições da Forma Normal de Boyce-Codd (FNBC) e da 3ª Forma Normal (3FN):

- **FNBC:** Uma relação R está na Forma Normal de Boyce-Codd se e só se, para todas as dependências funcionais não triviais $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_n$ de R, se constata que $\{A_1, A_2, \dots, A_n\}$ é uma superchave de R
- **3FN:** Uma relação R está na 3ª Forma Normal se e só, para todas as dependências funcionais não triviais $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_n$ de R, se verifica que $\{A_1, A_2, \dots, A_n\}$ é uma superchave de R OU os atributos que não pertencem ao conjunto $\{A_1, A_2, \dots, A_n\}$ são primos, isto é, integram uma chave candidata (não necessariamente a mesma)

Uma vez que todas as relações apresentadas só possuem uma chave ou não têm dependências funcionais não triviais, conseguimos inferir que as mesmas obedecem tanto à FNBC como à 3FN.

Restrições

→ Época

- Não podem existir duas Épocas com o mesmo id

idEpoca `INTEGER PRIMARY KEY`

→ Escalão

- Não podem existir dois Escalões com o mesmo id

idEscalao `INTEGER PRIMARY KEY`

- Um Escalão tem de ter um nome

nome `TEXT NOT NULL`

→ Associação

- Não podem existir duas Associações com o mesmo id

idAssociacao `INTEGER PRIMARY KEY`

- Uma Associação tem de ter um nome e uma data de fundação

nome `TEXT NOT NULL`

dataFundacao `DATE NOT NULL`

- Uma Associação tem de ter um endereço de email

email `TEXT NOT NULL`

- Uma Associação tem de pertencer a um país

email `TEXT NOT NULL`

→ Competição

- Não podem existir duas Competições com o mesmo id

idCompeticao `INTEGER PRIMARY KEY`

- Uma Competição tem de ter um nome e abreviatura

nome `TEXT NOT NULL`

abreviatura `TEXT NOT NULL`

- Uma Competição ocorre numa Época, tem um dado Escalão e é financiada por uma Associação; caso os dados destes sejam modificados, serão atualizados nos respetivos Clubes

idEpoca `INTEGER NOT NULL REFERENCES Epoca(idEpoca) ON UPDATE CASCADE`

idEscalao `INTEGER NOT NULL REFERENCES Escalao(idEscalao) ON UPDATE CASCADE`

idAssociacao INTEGER NOT NULL REFERENCES Associacao(idAssociacao) ON UPDATE CASCADE

→ Fase

- Não podem existir duas Fases com o mesmo id

idFase INTEGER PRIMARY KEY

- Uma Fase tem de ter um nome, um tipo e um estado

nome TEXT NOT NULL
tipo TEXT NOT NULL
estado TEXT NOT NULL

- Uma Fase tem de ter um número de equipas e um número equipas apuradas

numEquipas INTEGER NOT NULL
numEquipasApuradas INTEGER NOT NULL

- Uma Fase ocorre numa competição

idCompeticao INTEGER NOT NULL REFERENCES Competicao(idCompeticao) ON UPDATE CASCADE

→ Recinto

- Não podem existir dois Recintos com o mesmo id

idRecinto INTEGER PRIMARY KEY

- Um Recinto tem de ter um nome e de pertencer a um país

nome TEXT NOT NULL
pais TEXT NOT NULL

- Um Recinto tem de ter lotação, sendo esta um inteiro positivo

lotacao INTEGER NOT NULL CONSTRAINT lotacaoPositiva CHECK (lotacao > 0)

→ Clube

- Não podem existir dois Clubes com o mesmo id

idClube INTEGER PRIMARY KEY AUTOINCREMENT

- Um Clube tem de ter um nome e uma data de fundação

nome TEXT NOT NULL
dataFundacao DATE NOT NULL

- Cada Clube pertence a uma e uma só Associação; caso os dados da Associação sejam modificados, serão atualizados nos respetivos Clubes

idAssociacao INTEGER NOT NULL REFERENCES Associacao(idAssociacao) ON UPDATE CASCADE

- Cada Clube pode ter até um Recinto

`idRecinto INTEGER REFERENCES Recinto(idRecinto) ON UPDATE CASCADE`

→ Equipa

- Não podem existir duas Equipas com o mesmo id

`idEquipa INTEGER PRIMARY KEY AUTOINCREMENT`

- Uma Equipa tem de ter um nome

`nome TEXT NOT NULL`

- Uma Equipa tem de ter um e um só escalão

`idEscalao INTEGER REFERENCES Escalao(idEscalao) ON UPDATE CASCADE`

- Cada Equipa pertence, necessariamente, a um Clube; caso os dados do mesmo sejam modificados, serão atualizados nas respetivas Equipas

`idClube INTEGER NOT NULL REFERENCES Clube(idClube) ON UPDATE CASCADE`

→ Jogador

- Não podem existir dois Jogadores com o mesmo id

`idJogador INTEGER PRIMARY KEY AUTOINCREMENT`

- O nome, nome completo e a data de nascimento de um Jogador têm de estar definidos

`nome TEXT NOT NULL
nomeCompleto TEXT NOT NULL
dataNascimento DATE NOT NULL`

- Cada Jogador tem um país de origem

`pais TEXT NOT NULL`

- O sexo de um Jogador tem de ser definido e pode ser um de dois: masculino (genero = 'M') ou feminino (genero = 'F')

`sexo CHARACTER(1) NOT NULL`

- A altura de um Jogador é expressa em centímetros, arredondada às unidades, e tem de ser superior a 0

`altura INTEGER NOT NULL CONSTRAINT alturaPositiva CHECK (altura > 0)`

- O peso de um Jogador é expresso em quilogramas, arredondado às unidades (ex: 90 kg), e tem de ser superior a 0

`peso INTEGER CONSTRAINT pesoPositivo CHECK (peso > 0)`

→ FaseEquipa

- Cada instância desta classe pertence a uma Fase e a uma Equipa

```
idFase INTEGER NOT NULL
idEquipa INTEGER NOT NULL
```

- A pontuação de uma Equipa, numa Fase, é um inteiro não negativo

```
pontuacao INTEGER NOT NULL CONSTRAINT puntuacaoValida CHECK (pontuacao ≥ 0)
```

- O número de vitórias e derrotas são representados por inteiros não negativos

```
vitorias INTEGER NOT NULL CONSTRAINT vitoriasValidas CHECK (vitorias ≥ 0)
derrotas INTEGER NOT NULL CONSTRAINT derrotasValidas CHECK (derrotas ≥ 0)
```

- O número de faltas de comparecimento de uma Equipa é expresso por um inteiro não negativo

```
faltasComparencia INTEGER NOT NULL CONSTRAINT faltasComparenciaValidas CHECK (faltasComparencia ≥ 0)
```

- Os pontos marcados e sofridos são representados por inteiros não negativos

```
pontosMarcados INTEGER NOT NULL CONSTRAINT pontosMarcadosValidos CHECK (pontosMarcados ≥ 0)
pontosSofridos INTEGER NOT NULL CONSTRAINT pontosSofridosValidos CHECK (pontosSofridos ≥ 0)
```

→ EquipaJogador

- Numa Equipa, cada Jogador tem um número de camisola que lhe é exclusivo, o qual pode ter um ou dois dígitos

```
numCamisola INTEGER CONSTRAINT numCamisolaValido CHECK (numCamisola ≥ 0 AND numCamisola ≤ 99)
```

→ Jogo

- Não podem existir dois Jogos com o mesmo id

```
idJogo INTEGER PRIMARY KEY AUTOINCREMENT
```

- Cada Jogo tem de pertencer a uma Fase e a uma jornada

```
idFase INTEGER NOT NULL REFERENCES Fase(idFase) ON UPDATE CASCADE
jornada INTEGER NOT NULL
```

- Um Jogo decorre numa data e hora bem definidas e num único Recinto; dois Jogos distintos não podem ter, simultaneamente, a mesma data e ocorrer no mesmo Recinto

```
dataJogo DATE NOT NULL
horaInicio TIME NOT NULL
idRecinto INTEGER NOT NULL REFERENCES Recinto(idRecinto) ON UPDATE CASCADE
```

- Um Jogo está, necessariamente, relacionado com duas Equipas: a visitada, que joga em casa, e a visitante

```
idEquipaVisitante INTEGER NOT NULL REFERENCES Equipa(idEquipa) ON UPDATE CASCADE
idEquipaVisitada  INTEGER NOT NULL REFERENCES Equipa(idEquipa) ON UPDATE CASCADE
```

- Os pontos que cada Equipa marca num Jogo são representados por um inteiro não negativo

```
pontosEquipaCasa      INTEGER          CONSTRAINT      pontosEquipaCasaValidos    CHECK
(pontosEquipaCasaValidos ≥ 0)
pontosEquipaFora       INTEGER          CONSTRAINT      pontosEquipaForaValidos    CHECK
(pontosEquipaForaValidos ≥ 0)
```

→ Lance

- Não podem existir dois Lances com o mesmo id

```
idLance INTEGER PRIMARY KEY
```

- Cada Lance é efetuado por um e um só Jogador

```
idJogador INTEGER NOT NULL REFERENCES Jogador(idJogador) ON UPDATE CASCADE
```

- Cada Lance pertence a um e um só Jogo

```
idJogo INTEGER NOT NULL REFERENCES Jogo(idJogo) ON UPDATE CASCADE
```

- Um Lance tem de ocorrer num período, representado por um inteiro não negativo

```
periodo INTEGER NOT NULL CONSTRAINT periodoValido CHECK (periodo > 0)
```

- Um Lance ocorre num minuto, compreendido entre 0 e 12

```
minuto INTEGER NOT NULL CONSTRAINT minutoValido CHECK (minuto > '00:00:00' AND
minuto ≤ '00:12:00')
```

- Um Lance tem de ter pontos, sendo que estes podem ser: 0 (caso o Jogador falhe), 1, 2 ou 3.

```
pontos INTEGER NOT NULL CONSTRAINT pontosValidos CHECK (pontos ≥ 0 AND pontos ≤ 3)
```

Interrogações

1. Qual a altura e peso médios, arredondados às décimas, dos jogadores de um país?
2. Quais foram as 5 fases onde foram marcados mais pontos? Para cada uma, quais foram o número máximo e mínimo de pontos que foram marcados num só jogo?
3. Qual a terceira associação que possui o menor número de clubes que têm a palavra "Clube" ou "Club" no nome?
4. Quais as equipas que, em todos os jogos da fase regular do campeonato, nunca conseguiram marcar mais de 100 pontos?
5. Quais os 10 números de camisola de jogador mais comuns?
6. Qual o pior marcador com altura entre 1.85-1.95 m, que não tenha nacionalidade americana e que jogue como extremo?
7. Qual foi a equipa que venceu o campeonato?
8. Tendo em conta o desempenho individual de cada jogador no campeonato, forme a melhor equipa possível.
9. Quais foram as 3 equipas cujos jogadores tiveram, na fase regular, melhor FG?
10. Haverá, realmente, alguma vantagem em jogar em casa?

Nota: Optámos por reservar esta parte do relatório para as interrogações, propriamente, ditas. Assim sendo, as instruções que dizem respeito à visualização do resultado, nomeadamente, o nome das colunas e como as ordenar, encontram-se presentes nos respetivos ficheiros *intN.sql*.

Gatilhos

1. Verifica, antes de se marcar um jogo, se o recinto no qual se pretendia realizar o mesmo se encontra disponível para o efeito.
2. Quando ocorre um jogo numa dada fase do campeonato, atualiza a estatística de fase (pontuação, vitórias, derrotas, pontos marcados e pontos sofridos) das equipas que participaram nesse jogo.
3. Antes de se adicionar um jogador à base de dados, verifica se este apresenta valores para a altura e/ou peso. Caso o jogador não apresente, é adicionada uma entrada numa tabela auxiliar, denominada "log", que notifica o utilizador acerca do facto desses dados necessitarem de ser atualizados.

Nota: No que toca ao terceiro gatilho, preferimos, ao invés de, meramente, impedir a inserção do jogador na base de dados, guardar uma mensagem de erro numa tabela auxiliar, uma vez que, numa situação real, seria mais lógico adicionar os dados e, posteriormente, corrigi-los.

Esta ideia adveio do facto de, aquando lemos os ficheiros relativos aos jogadores, diversas entradas não possuírem qualquer informação relativa à altura e ao peso. Deste modo, com esta abordagem, torna-se, extremamente, simples modificar a base de dados, caso, no futuro, se saiba a altura e o peso dos jogadores onde estes faltam.