

BASES DE DADOS

**GESTÃO DOS RESULTADOS
DO CAMPEONATO DE FUTSAL**

2LEIC06 - GRUPO 604

Docente:

Lázaro Costa

Alunos:

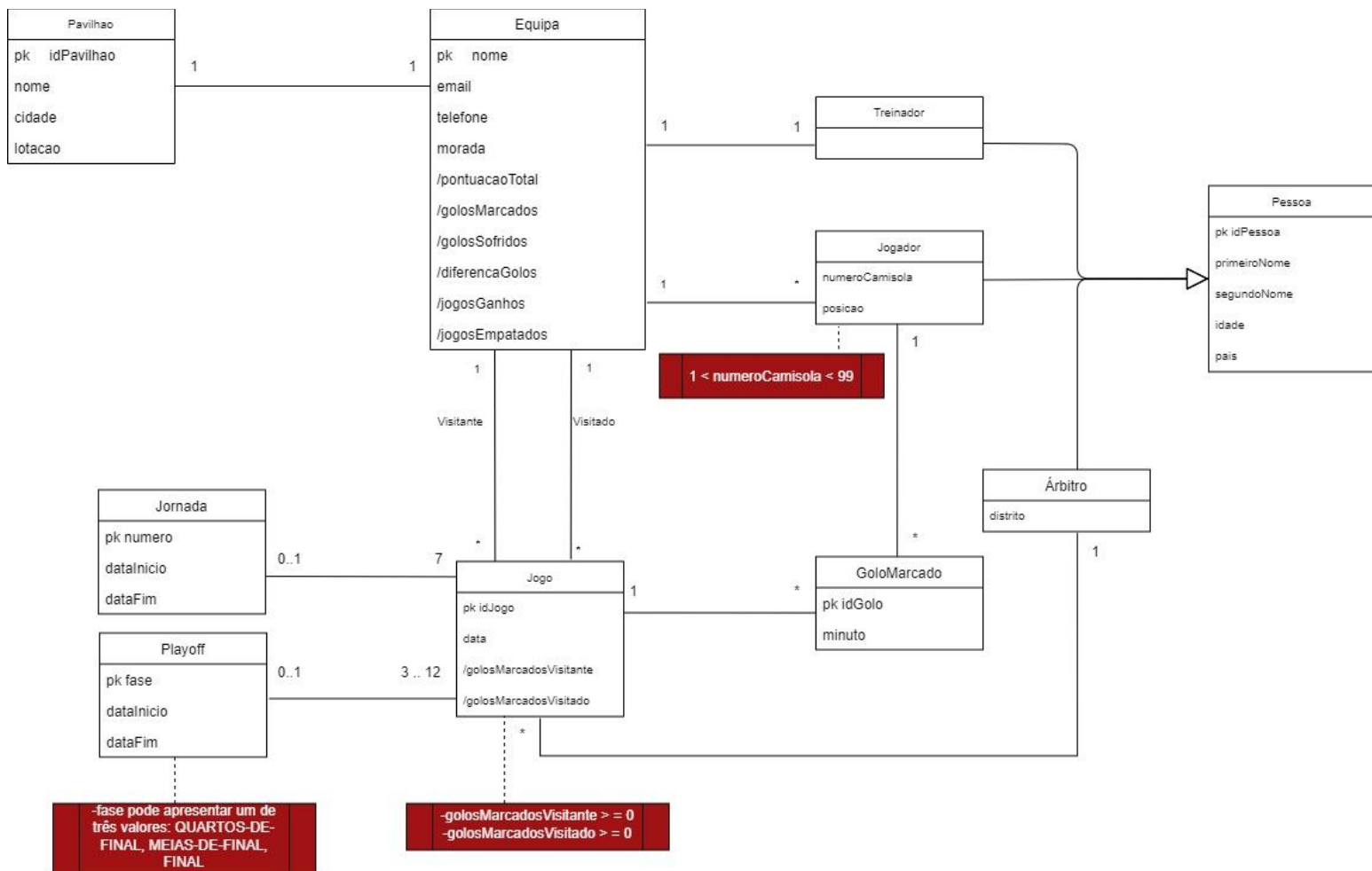
Adriano Machado	up202105352@fe.up.pt
Pedro Evans	up202108818@fe.up.pt

Introdução

O nosso projeto tem como objetivo a construção de uma base de dados que comporte as informações associadas ao campeonato nacional de Futsal, contemplando tanto a fase de grupos como os Playoffs, assim como as principais informações e estatísticas associadas a cada equipa, jogador e jogo. O principal desafio é a construção das tabelas adequadas para comportar todos os dados, de forma a evitar redundância de informação na base de dados, assim como implementar funcionalidades que permitam extrair estatísticas e informações úteis dos dados armazenados.

É apresentado o esquema UML, o esquema relacional e a análise das diferentes dependências funcionais, a análise da forma normal, as restrições a ter em conta nos diferentes elementos, as interrogações a fazer à base de dados e os gatilhos implementados.

Modelo Conceptual



Esquema Relacional

Pessoa(idPessoa, primeiroNome, segundoNome, idade, pais)

Treinador(idTreinador -> Pessoa, equipa -> Equipa)

Arbitro(idArbitro -> Pessoa, distrito)

Jogador(idJogador -> Pessoa, numeroCamisola, posicao, equipa -> Equipa)

Equipa (nome,email,telefone,morada, pontuacaoTotal, golosMarcados, golosSofridos, diferencaGolos, jogosGanhos, jogosEmpatados)

Pavilhao(idPavilhao, nome, cidade, lotacao, equipa -> Equipa)

Jogo(idJogo, data, golosMarcadosVisitante, golosMarcadosVisitado, nomeEquipaVisitante -> Equipa, nomeEquipaVisitada -> Equipa, idArbitro -> Arbitro, numero -> Jornada, fase->Playoff)

GoloMarcado(idGolo, minuto, idJogador->Jogador, idJogo -> Jogo)

Jornada(numero, dataInicio, dataFim)

Playoff(fase, dataInicio, dataFim)

Formas Normais e Análise de Dependências Funcionais

3.1 - Dependências funcionais

Pessoa:

idPessoa -> primeiroNome
segundoNome, idade, pais

Equipa:

nome -> email, telefone, morada
pontuacaoTotal, golosMarcados,
golosSofridos, diferencaGolos,
jogosGanhos, jogosEmpatados

GoloMarcado:

idGolo -> minuto, idJogador, idJogo

PlayOff:

fase -> dataInicio, dataFim

Arbitro:

idArbitro -> distrito

Jogo:

idJogo -> data, golosMarcadosVisitante, golosMarcadosVisitado, nomeEquipaVisitante,
nomeEquipaVisitada, arbitro, numeroJornada, fasePlayOff

data, nomeEquipaVisitante, nomeEquipaVisitada -> idJogo, golosMarcadosVisitante,
golosMarcadosVisitado, arbitro, numeroJornada, fasePlayOff

numeroJornada -> fasePlayOff (se numero jornada != NULL, então fasePlayOff = NULL)

fasePlayOff -> numeroJornada (se fasePlayOff != NULL, então jornada = NULL)

nomeEquipaVisitante, nomeEquipaVisitada, numeroJornada (não nulo) -> idJogo,
data, golosMarcadosVisitante, golosMarcadosVisitado, arbitro.

Treinador:

idTreinador -> equipa
equipa -> idTreinador

Jogador:

idJogador -> numeroCamisola, equipa,
posicao
numeroCamisola, equipa -> idJogador,
posicao

Pavilhao:

idPavilhao -> nome, cidade, lotacao,
equipa
equipa -> idPavilhao, nome, cidade,
lotacao

Jornada:

numeroJornada -> dataInicio, dataFim

3.2 - Violações da Terceira Forma Normal

De acordo com a anterior análise de dependências funcionais, conclui-se que todas as relações estão de acordo com a forma de normal de Boyce-Codd, ou na 3ª Forma normal, exceto as relações que envolvem o numeroJornada e a fasePlayoff. De acordo com as definições destas fórmulas:

Uma relação está em BCNF se, para todo $A \rightarrow B$ não trivial, A é uma superkey/key;

Uma relação está em 3NF se, para todo $A \rightarrow B$ não trivial, A é uma superkey/key

OU B consiste apenas em atributos primos (atributos que são membros de pelo menos uma chave da relação).

Para todas as relações apresentadas, exceto as já indicadas, a partir do lado esquerdo conseguimos saber todos os restantes atributos, logo A é superkey/key da relação, e as relações encontram-se na forma normal de Boyce-Codd.

- **numeroJornada -> fasePlayOff (se numero jornada != NULL, entao fasePlayOff = NULL)**
- **fasePlayOff -> numeroJornada (se fasePlayOff != NULL, entao jornada = NULL)**

No caso destas duas relações, encontram-se na 3ª Forma Normal, já que o lado direito é apenas constituído por atributos primos.

A única relação que não se encontra nesta forma é a seguinte, já que os atributos não são independentes uns dos outros.

- **golosMarcados,golosSofridos ->diferencaGolos**

Restrições

Equipa

- Não podem existir duas equipas com o mesmo nome
 - nome PRIMARY KEY
- Uma equipa tem obrigatoriamente um email, telefone e morada únicos
 - email UNIQUE NOT NULL
 - telefone UNIQUE NOT NULL
 - morada UNIQUE NOT NULL

Pavilhão

- Não podem existir dois pavilhões com o mesmo IdPavilhao
 - idPavilhao PRIMARY KEY
- Um pavilhão tem obrigatoriamente um nome, uma cidade e uma lotação e uma equipa atribuída
 - nome NOT NULL
 - cidade NOT NULL
 - lotacao NOT NULL
 - equipa NOT NULL
- A equipa do pavilhão deverá corresponder a uma equipa da tabela Equipa
 - equipa REFERENCES Equipa (nome)

Pessoa

- Não podem existir duas pessoas com o mesmo IdPessoa
 - idPessoa PRIMARY KEY
- Uma pessoa tem obrigatoriamente um primeiroNome, idade e pais
 - primeiroNome NOT NULL
 - idade NOT NULL
 - pais NOT NULL

Jogador

- Não podem existir dois jogadores com o mesmo IdJogador; este deve corresponder a um id da tabela Pessoa
 - idJogador PRIMARY KEY REFERENCES Pessoa (idPessoa)

- Um jogador tem, obrigatoriamente um numeroCamisola, uma posicao e uma equipa associados
 - numeroCamisola NOT NULL
 - posicao NOT NULL
 - equipa NOT NULL
- O numeroCamisola de um jogador deve estar entre 1 e 99
 - CHECK (1 < numeroCamisola < 99)
- A equipa do jogador deverá corresponder a uma equipa da tabela Equipa
 - equipa REFERENCES Equipa(nome)

Arbitro

- Não podem existir dois árbitros com o mesmo IdArbitro; este deve corresponder a um id da tabela Pessoa
 - idArbitro PRIMARY KEY REFERENCES Pessoa (idPessoa)
- Um arbitro tem obrigatoriamente associado um distrito
 - distrito NOT NULL

Treinador

- Não podem existir dois treinadores com o mesmo IdTreinador; este deve corresponder a um id da tabela Pessoa
 - idTreinador PRIMARY KEY REFERENCES Pessoa (idPessoa)
- O treinador tem obrigatoriamente uma equipa atribuída que corresponde a uma deferência de um nome da tabela Equipa
 - equipa REFERENCES Equipa (nome) NOT NULL

Jornada

- Não podem existir duas jornadas com o mesmo numero
 - numero PRIMARY KEY
- Uma Jornada tem obrigatoriamente uma dataInicio e uma dataFim únicas
 - dataInicio UNIQUE NOT NULL
 - dataFim UNIQUE NOT NULL

Playoff

- Não podem existir dois Playoff com a mesma fase; esta ter um de três valores: QUARTOS-DE-FINAL, MEIAS-FINAIS ou FINAL
 - fase PRIMARY KEY CHECK (fase = "QUARTOS-DE-FINAL" or fase = "MEIAS-FINAIS" or fase = "FINAL")
- Um Playoff tem obrigatoriamente uma dataInicio e uma dataFim únicas
 - dataInicio UNIQUE NOT NULL
 - dataFim UNIQUE NOT NULL

GoloMarcado

- Não podem existir dois GoloMarcado com o mesmo idGolo
 - idGolo PRIMARY KEY
- Um GoloMarcado tem obrigatoriamente um minuto associado
 - minuto NOT NULL
- Um GoloMarcado tem obrigatoriamente um idJogador que é uma referência de uma instância idJogador da tabela Jogador
 - idJogador REFERENCES Jogador(idJogador) NOT NULL
- Um GoloMarcado tem obrigatoriamente um idJogo que é uma referência de uma instância idJogo da tabela Jogo
 - idJogo REFERENCES Jogo(idJogo) NOT NULL

Jogo

- Não podem existir dois Jogos com o mesmo idJogo
 - idJogo PRIMARY KEY
- Um Jogo tem, obrigatoriamente, uma data, golosMarcadosVisitante e golosMarcadosVisitado; os golosMarcadosVisitante e golosMarcadosVisitado são igual ou superior a 0;
 - data NOT NULL
 - golosMarcadosVisitado CHECK (golosMarcadosVisitado >= 0) NOT NULL
 - golosMarcadosVisitante CHECK (golosMarcadosVisitante >= 0) NOT NULL
- Um Jogo apresenta um nomeEquipaVisitada e nomeEquipaVisitante que são referência de instâncias da tabela Equipa
 - nomeEquipaVisitada REFERENCES Equipa(nome) NOT NULL
 - nomeEquipaVisitante REFERENCES Equipa(nome) NOT NULL

- Um Jogo apresenta um arbitro que deve corresponder a um idArbitro da tabela Arbitro
 - arbitro REFERENCES Arbitro (idArbitro) NOT NULL
- Um Jogo apresenta um numeroJornada que deve corresponder a um numero na tabela Jornada. Apresenta também uma fasePlayoff que deve corresponder a uma fase na tabela Playoff; Um jogo pode corresponder a um jogo da fase regular (numeroJornada != Null and fasePlayoff = Null) ou pode corresponder a um jogo da fase de playOff (numeroJornada = Null and fasePlayOff != Null)
 - numeroJornada REFERENCES Jornada(numero)
 - fasePlayoff REFERENCES Playoff(fase)
 - CHECK ((numeroJornada IS NULL AND fasePlayOff IS NOT NULL) OR (numeroJornada IS NOT NULL AND fasePlayOff IS NULL))

Para além destas restrições, inserimos na nossa base de dados as restrições ON DELETE e ON UPDATE, para manter a integridade dos dados armazenados.

Interrogações

Introduzimos na nossa base de dados um conjunto de interrogações, no sentido de extrair da base de dados informações de interesse relativas ao campeonato.

1. Melhor marcador de cada equipa
2. Maior diferença de golos num jogo
3. Percentagem de vitórias em casa e fora de cada equipa
4. Ver a tabela de classificação final
5. Equipas com a maior e menor médias de idades do campeonato
6. Jogos ganhos, empatados e perdidos por cada equipa na fase regular do campeonato
7. Equipa que sofreu menos golos
8. Cinco árbitros com mais jogos arbitrados
9. Número de golos marcados em cada pavilhão
10. Equipas que marcaram pelo menos um golo em todos os jogos em que participaram

Gatilhos

Adicionamos também os seguintes gatilhos:

1. Sempre que é atualizado um determinado Jogo atualizamos os atributos `pontuacaoTotal`, `jogosGanhos` e `jogosEmpatados` da tabela **Equipa**.
2. Sempre que é adicionado uma nova linha na tabela `GoloMarcado`, incrementamos os atributos `golosMarcadosVisitante` da tabela **Jogador** se este for jogador da `equipaVisitante` e `golosMarcadosVisitado` se o **Jogador** for da `equipaVisitada`.
3. Sempre que é adicionado um novo **Jogador** verifica se já existe um mesmo jogador com o mesmo `numeroCamisola`. Se já existir, desencadeia um erro.
4. Sempre que é adicionada uma nova linha na tabela `Jogo` atualizamos os atributos `pontuacaoTotal`, `jogosGanhos` e `jogosEmpatados` da tabela **Equipa**. Se a equipa ganhar são somados 3 pontos, se empatar, 1 ponto e se perder 0 pontos. Este trigger complementa o primeiro, e garante os casos iniciais (quando não há ainda golos marcados e os jogos estão empatados).