

Bases de Dados
Campeonato nacional de hóquei em patíns
2022/23
2LEIC05 – Grupo 503

António Romão - up202108704
Francisco Campos - up202108735
José Henrique Pinheiro – up202108879

Índice

Descrição	3
Diagrama UML	4
Alterações para a segunda entrega	5
Esquema Relacional.....	6
Análise de Dependências Funcionais e Formas Normais	7
Restrições.....	10
Lista de interrogações	12
Lista de gatilhos	13
Contribuições	14

Descrição

Neste projeto pretende-se desenvolver uma Base de Dados para gerir os resultados do campeonato nacional de hóquei em patins.

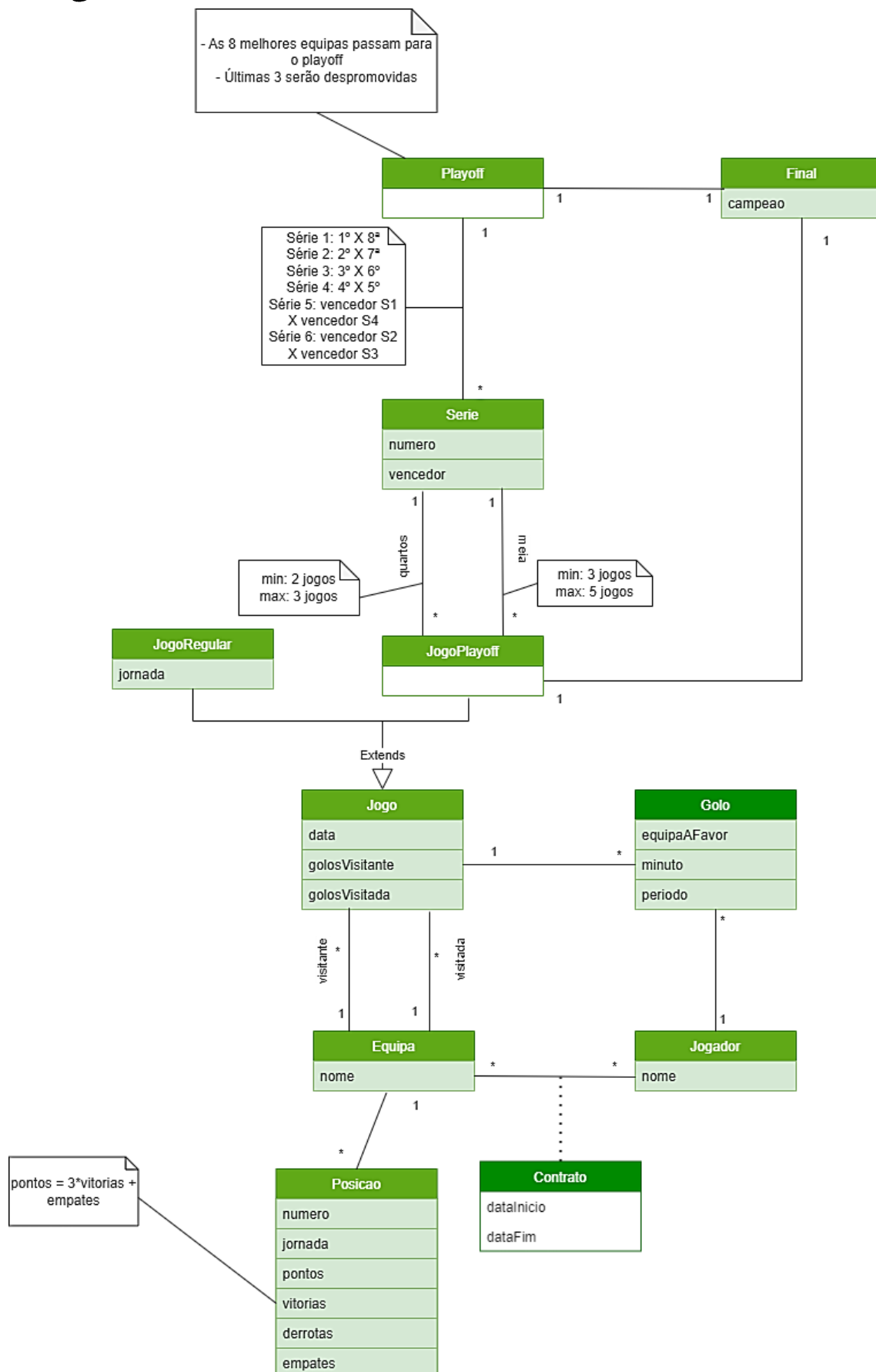
A mesma dispõe de informações diversas sobre o desenvolvimento do campeonato, entre as quais resultados jornada a jornada, marcadores dos golos, equipas que jogam em casa e jogam como visitante.

Para além disto, é coberta ainda uma fase de PlayOff, a qual inclui Quartos de Final, Meias Finais e a Final.

Jornadas e séries são mutuamente exclusivos, isto é, a fase de PlayOff só começa quando a fase Regular acaba.

O . Em caso de empate, entram os critérios de desempate (presente na secção de [Restrições](#)).

Díagrama UML



Alterações para a segunda entrega

Face à entrega anterior, decidiu-se modificar ligeiramente a estrutura da base de dados (refletida no diagrama UML acima).

Eliminou-se a classe *Jornada*, transferindo o seu atributo *numero* para as classes *JogoRegular* e *Posicao* como *jornada*.

Esta alteração permitiu que a base de dados armazene a posição ao longo das diferentes jornadas, bem como jogos referentes à mesma, possibilitando uma análise mais pormenorizada do desenvolvimento do campeonato.

Deste modo, a relação entre *Equipa* e *Posicao* deixou de ser 1 para 1 e passou a ser 1 para muitos, uma vez que cada equipa tem várias posições no decorrer das jornadas.

Perante a mudança acima referida, decidiu-se ainda remover a relação entre *Equipa* e *Playoff*, uma vez que, na modelagem atual do problema, se torna numa relação sem qualquer pertinência, sendo possível saber as equipas que frequentaram esta fase através dos jogos da mesma.

Para além disso, foram adicionados campos de data de início e fim à classe *Contrato*.

Esquema Relacional

Equipa (idEquipa, nome)

Contrato (idEquipa → Equipa, idJogador → Jogador, dataInicio, dataFim)

Jogador (idJogador, nome)

Posicao (numero, jornada, pontos, vitorias, empates, derrotas, idEquipa → Equipa)

Jogo (idJogo, data, golosVisitante, golosVisitada)

Visitada (idJogo → Jogo, idEquipa → Equipa)

Visitante (idJogo → Jogo, idEquipa → Equipa)

Golo (idGolo, idJogo → Jogo, idJogador → Jogador, equipaAFavor, minuto, periodo)

JogoRegular (idJogo → Jogo, jornada)

JogoPlayOff (idJogo → Jogo)

QuartoDeFinal (idJogorPlayOff → jogoPlayoff, numeroSerie → Serie)

MeiaFinal (idJogorPlayOff → jogoPlayOff, numeroSerie → Serie)

Final (idFinal, campeao)

PlayOff (idPlayOff, idFinal → Final)

Serie (numero, vencedor, idPlayOff → PlayOff)

Análise de Dependências Funcionais e Formas Normais

Equipa (idEquipa, nome)

FDs: {idEquipa} → {nome}
{nome} → {idEquipa}

Esta relação está na Boyce-Codd Normal Form, uma vez que, em todas as dependências funcionais na forma $\{A\} \rightarrow \{B\}$, A é uma superkey (como {nome}⁺ = {nome, idEquipa}, então a dependência funcional {nome} → {idEquipa} não invalida a BCNF). Como todas as formas BCNF estão também na 3^a Forma Normal (BCNF é um subset de 3NF), então esta relação também está na 3NF.

Contrato (idEquipa → Equipa, idJogador → Jogador, dataInicio, dataFim)

FDs: {idEquipa, idJogador} → {dataInicio, dataFim}

*Justificação 1

Jogador(idJogador, nome)

FDs: {idJogador} → {nome}

*Justificação 1

Posicao (numero, jornada, pontos, vitorias, empates, derrotas, idEquipa → Equipa)

FDs: {numero, jornada} → {pontos, vitorias, empates, derrotas, idEquipa}
{idEquipa, jornada} → {pontos, vitorias, empates, derrotas, numero}
{vitorias, empates, derrotas} → {pontos}

Para a relação Posicao, consideramos que a FD {pontos} → {numero} não era válida, uma vez que existem outros critérios para além dos pontos que determinam o numero da Posicao (por exemplo, caso duas equipas tenham os mesmos pontos, existem critérios de desempate como a diferença de golos).

Assim sabemos que a relação está na 1NF, uma vez que cada atributo é atômico, isto é representa um valor único (ou seja não é uma lista, ou um set). Também sabemos que a relação está na 2NF, uma vez que nenhum atributo non-prime (atributo que não pertence a uma chave candidata) é funcionalmente dependente de um prime (vitorias, empates e derrotas não são prime). No entanto, a relação Posicao não está na 3NF. Isto acontece porque {vitorias, empates, derrotas} não é uma superkey, e pontos não é um atributo do tipo prime. Seguidamente, como BCNF é um subset de 3NF, então se a Posicao não obedece à 3NF, então também não estará na forma BCNF.

Isto acontece porque o conjunto das vitorias e empates é funcionalmente dependente dos pontos (visto que pontos = 3*vitorias+empates). Esta violação pode ser corrigida retirando os atributos vitorias, empates e derrotas, ou tornando-os numa variável temporária, para o cálculo dos pontos (ou seja, não necessita de ser obrigatoriamente atributos). Se tal acontecer, então a relação Posicao apenas terá dependências funcionais que obedecem à BCNF, e, consequentemente, à 3FN. Assim, acabamos por na entrega final, através do uso de triggers, implementar a relação dada acima, fazendo com que os pontos não necessitem de ser diretamente alimentados na base de dados. Deste modo, a dependência funcional {vitorias, empates, derrotas} → {pontos} deixa de ser relevante na relação Posicao.

Jogo(idJogo, data, golosVisitante, golosVisitada)

FDs: {idJogo} → {data, golosVisitante, golosVisitada}
*Justificação1

Visitada(idJogo → Jogo, idEquipa → Equipa)

FDs: -
*Justificação2

Visitante(idJogo → Jogo, idEquipa → Equipa)

FDs: -
*Justificação2

Golo(idGolo, idJogo → Jogo, idJogador → Jogador, equipaAFavor, minuto, periodo)

FDs: {idGolo} → {idJogo, idJogador, equipaAFavor, minuto, periodo}
*Justificação1

JogoRegular(idJogo → Jogo, jornada)

FDs: {idJogo} → {numeroJornada}
*Justificação1

JogoPlayoff(idJogo → Jogo)

FDs: -
*Justificação2

QuartoDeFinal(idJogorPlayOff → jogoPlayOff, numeroSerie → Serie)

FDs: {idJogorPlayOff} → {numeroSerie}
*Justificação1

MeiaFinal(idJogorPlayOff → jogoPlayOff, numeroSerie → Serie)

FDs: {idJogorPlayOff} → {numeroSerie}
*Justificação1

Final(idFinal, campeao, idPlayoff → Playoff)

FDs: {idFinal} → {campeao, idPlayoff}
*Justificação1

Playoff(idPlayoff, idFinal → Final)

FDs: -
*Justificação2

Serie(numero, vencedor, idPlayOff → PlayOff)

FDs: {numero} → {vencedor, idPlayOff}
*Justificação1

Justificação1

Esta relação está na Boyce-Codd Normal Form, uma vez que, em todas as dependências funcionais na forma $\bar{A} \rightarrow \bar{B}$, \bar{A} é uma superkey. Como todas as formas BCNF estão também na 3ª Forma Normal (BCNF é um subset de 3NF), então esta relação também está na 3NF.

Justificação2

Esta relação está na Boyce-Codd Normal Form, pois não tem dependências funcionais não triviais. Como todas as formas BCNF estão também na 3ª Forma Normal (BCNF é um subset de 3NF), então esta relação também está na 3NF.

Restrições

Equipa:

- cada Equipa tem um nome, que é uma string não nula
- cada Equipa é identificada pelo idEquipa não nulo

Contrato:

- cada Contrato é identificado pelo idJogador e idEquipa
- cada Contrato refere-se a um Jogador (através do idJogador)
- cada Contrato refere-se a uma Equipa (através do idEquipa)

Jogador:

- cada Jogador tem um nome, que é uma string não nula
- cada Jogador é identificado pelo idJogador

Posicao:

- cada Posicao é identificada por um numero e pelo numero da jornada, que são dois inteiros positivos
- cada Posicao tem pontos que é um inteiro positivo ou nulo
- cada Posicao tem vitorias que é um inteiro positivo ou nulo
- cada Posicao tem empates que é um inteiro positivo ou nulo
- cada Posicao tem derrotas que é um inteiro positivo ou nulo

Jogo:

- cada Jogo é identificado pelo idJogo
- cada Jogo tem uma data, que é do tipo data
- cada Jogo tem golosVisitante, que é um inteiro positivo ou nulo
- cada Jogo tem golosVisitada, que é um inteiro positivo ou nulo

Visitada:

- Visitada refere-se a um Jogo (através do idJogo) e a uma Equipa (através do idEquipa)
- Visitada é identificada pelo idJogo

Visitante:

- Visitante refere-se a um Jogo (através do idJogo) e a uma Equipa (através do idEquipa)
- Visitante é identificada pelo idJogo

Golo:

- cada Golo é identificado pelo idGolo
- cada Golo refere-se a um Jogo (através do idJogo) e a um Jogador (através do idJogador)
- cada Golo tem uma equipaAFavor, que é um inteiro positivo (idEquipa)
- cada Golo tem um minuto (em que foi marcado), que é um inteiro positivo ou nulo (caso este seja marcado nos segundos iniciais)
- cada Golo tem um periodo (onde foi marcado), que é um inteiro positivo

JogoRegular:

- cada JogoRegular é identificado pelo idJogo
- cada JogoRegular refere-se a um Jogo (através do idJogo)

JogoPlayoff:

- cada JogoPlayoff é identificado pelo idJogo
- cada JogoPlayoff refere-se a um Jogo (através do idJogo)

QuartoDeFinal:

- cada QuartoDeFinal é identificado pelo idPlayoff
- cada QuartoDeFinal refere-se a um Playoff (através do idPlayoff) e a uma Serie (através do numeroSerie)

MeiaFinal:

- cada MeiaFinal é identificado pelo idPlayoff
- cada MeiaFinal refere-se a um Playoff (através do idPlayoff) e a uma Serie (através do numeroSerie)

Final:

- cada Final é identificada pelo idFinal
- cada Final tem um campeão, que é uma string não nula

Playoff:

- cada Playoff é identificado pelo idPlayoff
- cada Playoff refere-se a uma final (através do idFinal)

Serie:

- cada Serie é identificada pelo numero
- um numero é um inteiro positivo
- um vencedor é uma string não nula
- cada Serie refere-se a um Playoff (através do idPlayoff)

Critérios de Desempate:

1. Será melhor classificada a equipa que, nos jogos da fase da prova realizados entre ambos, tenha obtido o maior número de pontos;
2. Será melhor classificada a equipa que, nos jogos da fase da prova, realizados entre ambos, tenha conseguido uma maior diferença entre os golos marcados e sofridos;
3. Se o empate persistir, será melhor classificada a equipa que tenha conseguido uma maior diferença entre os golos marcados e sofridos ao longo da Fase da Prova em que as equipas empatadas intervierem diretamente;
4. Será melhor classificada a equipa que tiver o maior quociente geral na prova, resultante da divisão dos golos marcados pelos golos sofridos ao longo da fase da prova em que as equipas, pontualmente empatadas, intervieram diretamente

Passagem a fase de Playoff:

- As equipas que se encontrem nos primeiros 8 lugares da tabela no fim da fase regular têm acesso a fase de Playoff

Lista de interrogações

- 1) Equipa com a menor diferença de golos
- 2) Equipa que jogaram mais jogos de Playoff
- 3) Melhor marcador de cada equipa
- 4) Tabela de classificação final
- 5) Tabela do desenvolvimento das series da fase de Playoff
- 6) Percentagem de golos do melhor marcador de cada equipa face ao total de golos da mesma
- 7) Média dos golos marcados por período
- 8) Jogador que marcou mais golos contra a equipa vencedora
- 9) Campo mais difícil (equipa com mais vitórias em casa)
- 10) Maior comeback (equipa que no fim do primeiro período estava a perder e depois ganhou o jogo)

NOTA:

Em algumas destas queries foram utilizadas views, não só por uma questão de conveniência como também por forma a torná-las mais legíveis.

Lista de gatilhos

- 1) Calcula os pontos depois de realizar uma inserção na tabela de uma dada posição na POSICAO, contendo os atributos (equipa, numero, jornada, vitorias, empates, derrotas).
- 2) A cada inserção de um golo na tabela GOLO atualiza o resultado na tabela JOGO bem como os dados da respetiva equipa na tabela POSICAO.
- 3) Verifica se o número de uma série e de uma jornada é menor do que 6 e 26, respetivamente.

Observações:

O gatilho 1 foi pensado por forma a preencher um dos atributos da tabela POSICAO, que depende de uma formula cujas variáveis são atributos da mesma tabela.

O gatilho 3, tem como propósito manter a coerência da base de dados, substituindo restrições implementados com *checks* na entrega anterior.

Deste modo, tanto 1 como 3 devem estar ativos antes de se efetuar o povoamento.

Contrariamente, gatilho 2 foi construído por forma a permitir a correção de lapsos nos golos após o termino do jogo, devendo de ser aplicado apenas posteriormente ao povoamento.

Contribuições

Consideramos que neste trabalho o grupo apresentou uma postura de trabalho e interesse pelo projeto, tendo todos os elementos contribuído, de diferentes formas, nas diversas tarefas.

No entanto, evidenciou-se especial destaque (nos seguintes pontos) dos elementos:

- A. todos
- B. up202108879
- C. up202108735
- D. up202108879
- E. todos
- F. todos
- G. up202108735/ up202108879
- H. todos