

Faculdade de Engenharia da Universidade do Porto

Faculdade de Ciências da Universidade do Porto



Rankings dos Melhores Atletas de Sempre de Atletismo Bases de Dados

Turma 2LEIC09

Equipa 906

Estudantes & Autores:

André Filipe Cardoso Barbosa - 202007398

Guilherme Cunha S. F. de Almeida - 202008866

Henrique Seabra Ferreira - 202007044

Índice

1. Contexto	3
2. Diagrama UML	4
3. Modificações UML	5
4. Esquema Relacional	6
5. Análise de Dependências Funcionais e Formas Normais	7
5.1 Dependências Funcionais	7
5.2 Violações ao Boyce-Codd e 3ª Forma Normal	9
6. Restrições	10
7. Interrogações	14
8. Gatilhos	15
8.1 Lista de Gatilhos	15
8.2 Observações sobre Gatilhos	16
9. Auto Avaliação	17

1. Contexto

Neste projeto pretende-se criar uma base de dados que englobe as melhores marcas do atletismo português.

Os dados serão organizados consoante o género e escalão do atleta, de acordo com o evento em que a marca foi estabelecida, sendo desta forma possível processar e visualizar a informação em formato *ranking*.

A entrada do atleta contém a sua identificação, clube e escalão aquando da realização da marca, bem como o local onde a prova foi realizada.

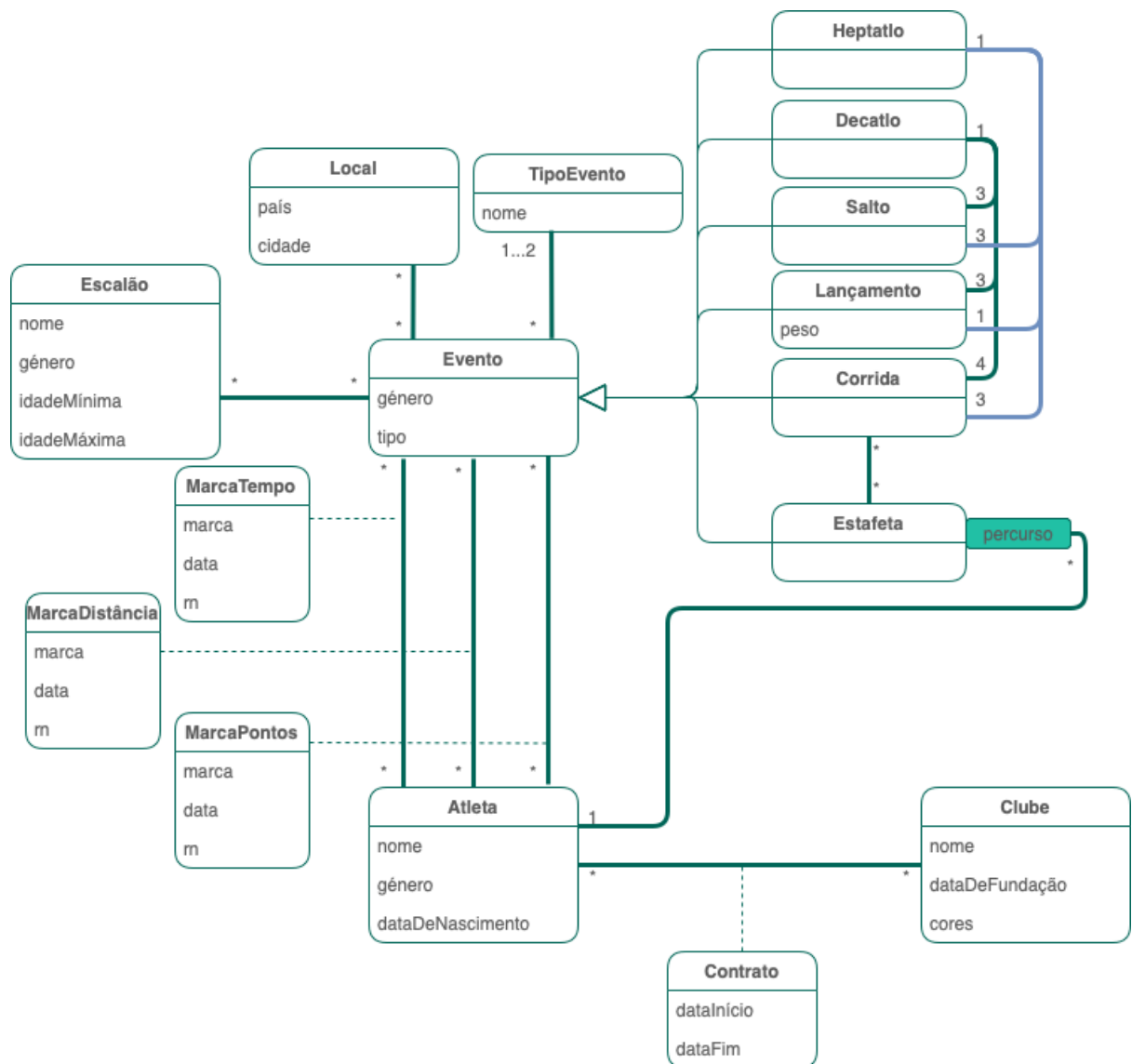
Uma vez que o objetivo do trabalho passa por organizar os atletas de acordo com as suas melhores performances, cada atleta terá apenas um registo por evento.

No atletismo existe uma variedade de **eventos** que se enquadram em categorias como saltos (triplo salto, etc.), corrida (100m, 1500m, 20km marcha, etc.), lançamentos (peso, dardo, etc.) e provas combinadas (decatlo, heptatlo, etc.). Geralmente este desporto é praticado individualmente, mas existe o caso específico das estafetas, modalidade esta que é constituída por equipas de 4 elementos ordenados pelo seu percurso.

A época desportiva é dividida em dois tipos, pista coberta e ar livre, havendo eventos específicos para cada uma.

Deste modo, é necessário completar com uma variedade de dados o nosso projeto. Assim, iremos utilizar os registos oficiais da federação portuguesa de atletismo bem como o website "[Atletismo Estatística](#)" que contém informações complementares aos rankings.

2. Diagrama UML



3. Modificações UML

Foi removida a ligação **Atleta->Escalão**, pois é possível determinar o **escalão** do **atleta** através da **data** da **marca** realizada.

Removeu-se também o atributo **distância** da classe **corrida**, pois é possível ser determinado através do atributo **tipo** da classe **evento**.

As subclasses **heptatlo**, **decatlo** e **salto**, foram colocadas “fora” da classe **evento**, de modo a ser possível realizar ligações entre as subclasses. Isto porque as provas de **heptatlo** e **decatlo** são constituídas por provas de **saltos**, **lançamentos** e **corridas**. A classe **estafeta** foi alterada para uma subclasse de **evento** e foi criada a ligação **corrida->estafeta**, exatamente porque uma estafeta é uma corrida. Assim foi possível remover a classe **marca** que estava na ligação **atleta->estafeta**.

Para ser possível distinguir se uma marca é um tempo, uma distância ou uma pontuação, foram criadas as classes **MarcaTempo**, **MarcaDistância**, **MarcaPontuação**.

Por último foi criada a classe **Contrato** na ligação **Atleta->Clube** de modo a ser possível atribuir vários clubes a um atleta ao longo da sua carreira.

4. Esquema Relacional

- Clube(idClube, nome, dataDeFundação, cores)
- Atleta(idAtleta, nome, genero, dataDeNascimento)
- Contrato(idClube->Clube, idAtleta->Atleta, dataInicio, dataFim)
- Evento(idEvento, género, tipo)
- MarcaTempo(idEvento->Evento, idAtleta->Atleta, marca, data, rn)
- MarcaDistância(idEvento->Evento, idAtleta->Atleta, marca, data, rn)
- MarcaPontos(idEvento->Evento, idAtleta->Atleta, marca, data, rn)
- Escalao(idEscalao, nome, genero, idadeMinima, idadeMaxima)
- EventoEscalao(idEvento->Evento, idEscalao->Escalao)
- Local(idLocal, pais, cidade)
- EventoLocal(idEvento->Evento, idLocal->Local)
- TipoEvento(idTipoEvento, nome)
- EventoTipoEvento(idEvento->Evento, idTipoEvento->TipoEvento)
- Heptatlo(idEvento->Evento, idHeptatlo)
- Decatlo(idEvento->Evento, idDecatlo)
- Salto(idEvento->Evento, idSalto, idDecatlo->Decatlo, idHeptatlo->Heptatlo)
- Lancamento(idEvento->Evento, idLancamento,
idDecatlo->Decatlo,idHeptatlo->Heptatlo)
- Corrida(idEvento->Evento, idCorrida, idDecatlo->Decatlo,
idHeptatlo->Heptatlo)
- Estafeta(idEvento->Evento, idEstafeta)
- CorridaEstafeta(idCorrida->Corrida, idEstafeta->Estafeta)
- Percurso(idEstafeta->Estafeta, idAtleta->Atleta, percurso)

5. Análise de Dependências Funcionais e Formas Normais

5.1 Dependências Funcionais

Clube:

idClube -> nome, dataDeFundação, cores

Atleta:

idAtleta -> nome, genero, dataDeNascimento

Contrato:

idAtleta, dataInicio -> dataFim

Evento:

idEvento -> genero, tipo

MarcaTempo:

idAtleta, idEvento -> data, marca

MarcaDistância:

idAtleta, idEvento -> data, marca

MarcaPontos:

idAtleta, idEvento -> data, marca

Escalao:

idEscalao -> nome, genero, idadeMinima, idadeMaxima, nome, genero -> idadeMinima, idadeMaxima

EventoEscalao:

Sem dependências funcionais.

Local:

idLocal -> pais, cidade

EventoLocal:

Sem dependências funcionais.

TipoEvento:

idEvento->nome

EventoTipoEvento

Sem dependências funcionais.

Heptatlo:

idEvento, idHeptatlo ->idLancamento

Decatlo:

Sem dependências funcionais.

Salto:

idEvento, idSalto ->idDecatlo, idHeptatlo

Lancamento:

idEvento, idLancamento->idDecatlo

Corrida:

idEvento, idCorrida-> idDecatlo, idHeptatlo

Estafeta:

Sem dependências funcionais.

CorridaEstafeta:

Sem dependências funcionais.

Percurso:

idEstafeta, idAtleta -> percurso

5.2 Violações ao Boyce-Codd e 3ª Forma Normal

Escalao(idEscalao, nome, genero, idadeMinima, idadeMaxima):

- Viola Boyce-Codd porque FD: nome, genero->idadeMinima, idadeMaxima não contem uma key do lado esquerdo nem é trivial
- Viola 3 Forma Normal porque FD: nome, genero->idadeMinima, idadeMaxima não contem uma key do lado esquerdo, não é trivial nem o lado direito é constituído apenas por atributos primos.

Decompor Escalao:

- Escalao1 (idEscalao, nome, genero)

FD:

idEscalao -> nome, genero

Key: {idEscalao}

Escalao1 está em FNBC. pois a FD contém uma key do lado esquerdo.

- Escalao2 (nome, genero, idadeMinima, idadeMaxima)

FD:

nome, genero -> idadeMinima, idadeMaxima

Key: {nome, genero}

Escalao2 está em FNBC. pois a FD contém uma key do lado esquerdo.

6. Restrições

Clube

Clubes diferentes têm IDs diferentes → Clube_PK → idClube PRIMARY KEY

Clube tem obrigatoriamente nome e data de fundação → nome NOT NULL, dataDeFundacao NOTNULL

Atleta

Atletas diferentes têm IDs diferentes → Atleta_PK → idAtleta PRIMARY KEY

Atleta tem obrigatoriamente nome, género e data de fundação → nome NOT NULL, genero NOT NULL, dataDeFundacao NOTNULL

Género do atleta só pode ser masculino ou feminino → Genero_Valido → genero CHECK (genero = "M" or genero = "F")

Contrato

Não pode haver dois contratos com o par (idAtleta, dataInicio) igual → CAC_PK

Contrato tem obrigatoriamente id do Atleta e de Clube e data de início → idAtleta NOT NULL, idClub NOT NULL, dataInicio NOTNULL

idAtleta e idClube devem corresponder aos ids das tabelas Atleta e Clube (FOREIGN KEYS) → idAtleta REFERENCES Atleta, idClube REFERENCES Clube

Deve existir apenas um contrato por cada combinação de Atleta, Clube e data de início de contrato → UNIQUE (idAtleta, idClube, dataInicio)

A data de fim de contrato deve ser igual ou posterior à data de início de contrato → DATAS_VALIDAS → CHECK (dataInicio <= dataFim)

Evento

Eventos diferentes têm IDs diferentes → EVE_PK → idEvento PRIMARY KEY

Evento tem obrigatoriamente género associado e tipo → genero NOT NULL, tipo NOT NULL

Género associado ao evento só pode ser masculino ou feminino → Genero_Valido → genero CHECK (genero = "M" or genero = "F")

Tipo só pode englobar as provas existentes no atletismo → Tipo_Evento_Valido → tipo

CHECK TIPO = '60M' OR ... OR TIPO = '4x400M'

MarcaTempo / MarcaDistância / MarcaPontos

Não pode haver duas marcas com o par (idAtleta, idEvento) igual → (idAtleta, idEvento)
PRIMARY KEY

As marcas (iguais na sua estrutura exceto no tipo de variável da marca) têm obrigatoriamente valor da marca e data → marca NOT NULL, data1 NOT NULL

idAtleta e idEvento devem corresponder aos ids das tabelas Atleta e Evento (FOREIGN KEYS) → idAtleta REFERENCES Atleta, idEvento REFERENCES Evento

Deve existir apenas uma marca por cada combinação de Atleta, Evento, data e valor da marca → UNIQUE (idAtleta, idEvento, data1, marca)

Escalao1

Escalões diferentes têm IDs diferentes → CE1_PK → idEscalao PRIMARY KEY

O escalão tem obrigatoriamente um nome → nome NOT NULL

Género associado ao escalão só pode ser masculino ou feminino → Genero_Valido → genero CHECK (genero = "M" or genero = "F")

Escalao2

Não pode haver duas escalões com o par (nome, genero) igual → (nome, genero)
PRIMARY KEY

O escalão tem obrigatoriamente um nome → nome NOT NULL

Idade máxima tem de ser superior à idade mínima → Idade_Valida → CHECK idadeMinima < idadeMaxima.

Género associado ao evento só pode ser masculino ou feminino → Genero_Valido → genero CHECK (genero = "M" or genero = "F")

EventoEscalao

Não pode haver duas relações EventoEscalao com o par (idEvento, idEscalao) igual → (idEvento, idEscalao) PRIMARY KEY

idEvento e idEscalao devem corresponder aos ids das tabelas Evento e Escalao (FOREIGN KEYS) → idEvento REFERENCES Evento, idEscalao REFERENCES Escalao

Local1

Locais diferentes têm IDs diferentes → idLocal PRIMARY KEY

Locais têm obrigatoriamente cidade e país → pais, cidade NOT NULL.

EventoLocal:

idEvento e idLocal devem corresponder aos ids das tabelas Evento e Local1 (FOREIGN KEYS) → idEvento REFERENCES Evento, idLocal REFERENCES Local1

Não pode haver duas relações Evento-Local com o par (idEvento, idLocal) igual → (idEvento, idLocal) PRIMARY KEY

TipoEvento:

TipoEvento diferentes têm idEvento diferente → CEL_PK → idTipoEvento PRIMARY KEY

Tipo evento associado ao tipo de evento só pode ser pc - pista coberta, ou al - ar livre → NOME_VALIDO → NOME CHECK (nome = "pc" or nome = "al")

EventoTipoEvento:

idEvento e idTipoEvento devem corresponder aos ids das tabelas Evento e TipoEvento (FOREIGN KEYS) → idEvento REFERENCES Evento, idTipoEvento REFERENCES

Heptatlo

Heptatlos diferentes têm idHeptatlo diferentes → idHeptatlo PRIMARY KEY

idEvento deve corresponder ao id da tabela Evento (FOREIGN KEYS) → idEvento REFERENCES Evento.

Decatlo

Decatlos diferentes têm idDecatlo diferentes → idDecatlo PRIMARY KEY

idEvento deve corresponder ao id da tabela Evento (FOREIGN KEYS) → idEvento REFERENCES Evento.

Salto

Saltos diferentes têm idSalto diferente → idSalto PRIMARY KEY

idEvento, idDecatlo e idHeptatlo devem corresponder aos ids das tabelas Evento, Decatlo e Heptatlo (FOREIGN KEYS) → idEvento REFERENCES Evento, idDecatlo REFERENCES Decatlo, idHeptatlo REFERENCES idHeptatlo. Decatlo, idHeptatlo REFERENCES idHeptatlo.

Lançamento

Lançamentos diferentes têm idLancamento diferente → idLancamento PRIMARY KEY

idEvento, idDecatlo e idHeptatlo devem corresponder aos ids das tabelas Evento, Decatlo e Heptatlo (FOREIGN KEYS) → idEvento REFERENCES Evento, idDecatlo REFERENCES Decatlo, idHeptatlo REFERENCES idHeptatlo. Decatlo, idHeptatlo REFERENCES idHeptatlo.

Peso só pode tomar certos valores → PESO_VALIDO → peso CHECK(PESO = 3 OR PESO = 4 OR PESO = 5 OR PESO = 6 OR PESO = 7 OR PESO = 0.500 OR PESO = 0.600 OR PESO = 0.700 OR PESO = 0.800 OR PESO = 1 OR PESO = 1.5 OR PESO = 1.75 OR PESO = 2).

Corrida

Corrida diferentes têm idCorrida diferentes → idCorrida PRIMARY KEY

idEvento, idDecatlo e idHeptatlo devem corresponder aos ids das tabelas Evento, Decatlo e Heptatlo (FOREIGN KEYS) → idEvento REFERENCES Evento, idDecatlo REFERENCES Decatlo, idHeptatlo REFERENCES idHeptatlo.

Estafeta

Estafeta diferentes têm idEstafeta diferente → idEstafeta PRIMARY KEY

idEvento deve corresponder ao id da tabela Evento (FOREIGN KEYS) → idEvento REFERENCES Evento.

Corrida_Estafeta

idCorrida e idEstafeta devem corresponder aos ids das tabelas Corrida e Estafeta (FOREIGN KEYS) → idCorrida REFERENCES Corrida, idEstafeta REFERENCES Estafeta.

Percurso

idAtleta e idEstafeta devem corresponder aos ids das tabelas Atleta e Estafeta (FOREIGN KEYS) → idAtleta REFERENCES Atleta, idEstafeta REFERENCES Estafeta.

Deve existir apenas um contrato por cada combinação de Atleta, Clube e ordem de percurso → UNIQUE (idAtleta, idEstafeta, Percurso)

Ordem no percurso da estafeta só pode tomar valores entre 1 e 4 → PERCURSO_VALIDO → percurso CHECK(PERCURSO = 1 OR PERCURSO = 2 OR PERCURSO = 3 OR PERCURSO = 4)

7. Interrogações

1. Quais são os eventos dos rankings?
2. Qual a média das marcas do ranking de lançamento de peso, sénior masculino?
3. Quais os atletas e as suas respetivas marcas que estabeleceram recorde nacional nos 100 metros séniores masculinos? Ordenado por data.
4. Na prova de salto em altura do escalão sénior, quantas dessas marcas foram em pista coberta e quantas em ar livre?
5. Quais os atletas que estão inseridos em rankings da mesma modalidade em diferentes escalões?
6. Qual o número de atletas que cada clube já teve no total?
7. Qual o local em que foram realizadas mais marcas dos rankings e o número de marcas?
8. Qual a diferença em centímetros entre a primeira e segunda melhor marca no salto em comprimento masculino?
9. Quais os atletas que participaram na melhor estafeta 4x100m e quais os seus percursos?
10. Listagem de cada uma das provas e respetivas marcas, da melhor marca de decatlo.

8. Gatilhos

8.1 Lista de Gatilhos

1. Ao adicionar provas ao um decatlo, verifica se **a prova que está a ser adicionada já existe nesse decatlo** (visto que não é possível haver duas provas do mesmo tipo) ou se **a prova a ser inserida é inválida** (não pertence às provas do decatlo).
2. Quando se adiciona uma nova marca de um atleta já inserido no ranking, caso a marca a ser adicionada é “melhor” do que a que está no ranking, **a marca antiga é removida e a mais recente é adicionada**. Caso contrário a **marca que já estava no ranking é mantida** - por se tratar de um ranking de atletas, cada atleta só pode ter uma instância em cada ranking.
3. Ao adicionar uma marca nova, verifica se o **tipo de marca** é o correto para o evento indicado. Isto é, quando adicionamos uma marca do tipo **distância**, esta nunca pode estar associada a um evento cujo resultado é em tempo.

8.2 Observações sobre Gatilhos

1. O gatilho adicionado no ficheiro *gatilho1_adiciona.sql* não é suficiente para garantir completamente a restrição desejada. Este gatilho abrange apenas as hipóteses de adicionar provas de **corrida** ao decatlo. Deste modo, é necessário criar 2 gatilhos adicionais, um para provas de **saltos** e outro para provas de **lançamentos**. A implementação em cada um destes gatilhos seria semelhante à do gatilho1, com a diferença nos **tipos de evento** que seriam aceites - saltos (comprimento, altura, vara), lançamentos (peso, disco, dardo).
2. Tal como o primeiro gatilho, o do ficheiro *gatilho2_adiciona.sql* também não satisfaz por completo a restrição que se pretendia. Como explicado anteriormente, este gatilho substitui uma marca de um atleta, caso a marca a inserir seja “melhor”, mas abrange apenas a situação de **marcas por distância**. Seria necessário adicionar mais dois gatilhos, um para **marcas por tempo** e outro para **marcas por pontos**. Gatilhos estes que seriam quase idênticos mudando apenas as especificações do tipo de marca.
3. Para o gatilho do ficheiro *gatilho3_adiciona.sql* satisfazer por completo a restrição, será necessário adicionar mais dois gatilhos, um que engloba as **marca_tempo** e outro as **marca_pontos**. A sua implementação será idêntica alterando-se apenas as provas aceites em cada um - **marca_tempo** (corrida), **marca_pontos** (decatlo, heptatlo).

9. Auto Avaliação

André Barbosa - 33.(3)%

Guilherme Almeida - 33.(3)%

Henrique Ferreira - 33.(3)%