Projeto de Bases de Dados 21/22

Alexandre Correia *202007042*

Henrique Silva *202007242*

Tiago Branquinho *202005567*

Grupo 1006

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Índice

Contexto

O nosso grupo tomou a iniciativa de escolher uma temporada de Moto GP como tema para a construção de uma base de dados de modo a clarificar este desporto motorizado, cuja primeira edição ocorreu em 1949. Para tal, recorreu-se a um diagrama UML.

A competição é disputada entre um conjunto de marcas, cada uma distinguida pelo seu nome, nacionalidade e ano de ingresso. Por sua vez, a cada uma destas podem estar associadas diversas equipas.

Equipas pertencentes à mesma marca caracterizam-se, para além do seu nome, pelo seu tipo. Geralmente, a principal é considerada a de "fábrica" e as restantes as de "satélite". Ambas são compostas por vários colaboradores, que se dividem em engenheiros e, pelo menos, dois pilotos. Tal como os engenheiros, os pilotos são conhecidos pelo seu nome, nacionalidade e data de nascimento, sendo-lhes ainda atribuído um número, um tipo (principal ou reserva) e o número de pontos acumulados. Cada um pode apenas conduzir uma mota. Por outro lado, cada engenheiro é especializado num determinado ramo.

Cada veículo é caracterizado pelo seu peso, bem como pela sua potência e uma velocidade máxima inferior a 340km/h. Para além disso, estes podem ser conduzidos por vários pilotos, e têm de estar equipados com dois pneus. Existem várias categorias de pneumáticos, estabelecidas de acordo com o seu tipo, rigidez e marca, sendo esta igual para todos os que pertencem à mesma temporada da competição.

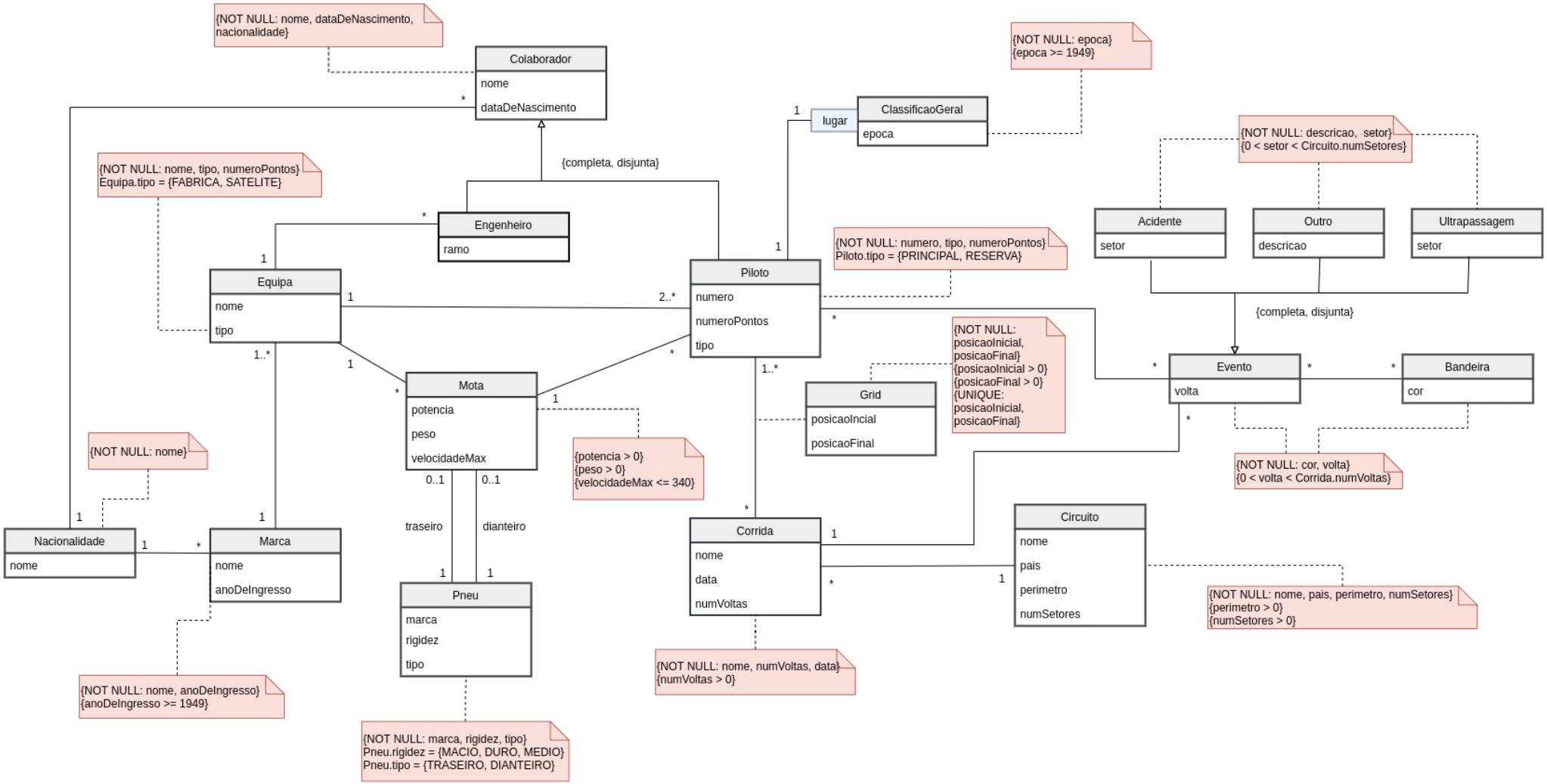
A competição descrita trava-se numa sequência pré-definida de corridas. A cada uma está atribuído um nome, uma data, um número de voltas e, evidentemente, um circuito. As posições de partida e de chegada dos pilotos são relevantes para a prova e, como tal, devem ser registadas numa *grid*.

Às diversas pistas visitadas ao longo da competição estão associados um nome, um país, um perímetro e um número de setores. É de salientar que podem ser travadas várias corridas no mesmo circuito.

Em cada corrida são registados os eventos que a marcam. Com efeito, podemos dividi-los em vários tipos, tais como ultrapassagens e acidentes, aos quais correspondem um setor, e outros, como, por exemplo, a ocorrência de precipitação. A cada um está associado uma volta e, por fim, podem, ou não, culminar numa ou várias bandeiras. Uma bandeira terá sempre uma cor associada, dependente do tipo de evento que sinaliza.

Em todas as épocas existe uma classificação geral que tem em conta o desempenho dos pilotos ao longo do ano. Assim sendo, o vencedor é determinado de acordo com os pontos adquiridos ao longo da mesma.

Diagrama UML



Esquema Relacional

- Nacionalidade (idNacionalidade, nome)
- Marca (idMarca, nome, anoDeIngresso, idNacionalidade->Nacionalidade)
- Equipa (idEquipa, nome, tipo, idMarca->Marca)
- Mota (idMota, potencia, peso, velocidadeMax, idEquipa->Equipa)
- Pneu (idPneu, marca, rigidez, tipo, idMota->Mota)
- Colaborador (idColaborador, nome, dataDeNascimento, idNacionalida->Nacionalidade)
- Engenheiro (idColaborador->Colaborador, ramo, idEquipa->Equipa)
- Piloto (idColaborador->Colaborador, numero, numeroPontos, tipo, idEquipa->Equipa, idMota->Mota)
- Circuito (idCircuito, nome, pais, perimetro, numSetores)
- Corrida (idCorrida, nome, data, numVoltas, idCircuito->Circuito)
- Grid (idCorrida->Corrida, idColaborador->Piloto, posicaoInicial, posicaoFinal)
- ClassificacaoGeral (epoca, lugar, idColaborador->Piloto)
- Evento (idEvento, volta, idCorrida->Corrida)
- Acidente (idEvento->Evento, setor)
- Outro (idEvento->Evento, descricao)
- Ultrapassagem (idEvento->Evento, setor)
- Bandeira (idBandeira, cor)
- PilotoCorrida (idPiloto->Piloto, idCorrida->Corrida)
- PilotoEvento(idPiloto->Piloto, idEvento->Evento)
- EventoBandeira (idEvento->Evento, idBandeira->Bandeira)

Análise de Dependências Funcionais e Formas Normais

- Nacionalidade(idNacionalidade, nome)
  - FDs: idNacionalidade->nome
  - Formas: BCNF:sim, 3NF:sim
- Marca(idMarca, nome, anoDeIngresso, idNacionalidade->Nacionalidade)
  - FDs: idMarca->nome, anoDeIngresso, idNacionalidade
  - Formas: BCNF:sim, 3NF:sim
- Equipa(idEquipa, nome, tipo, idMarca->Marca)
  - FDs: idEquipa->nome, tipo, idMarca
  - Formas: BCNF:sim, 3NF:sim
- Mota(idMota, potencia, peso, velocidadeMax, idEquipa->Equipa)
  - FDs: idMota->potencia, peso, velocidadeMax, idEquipa
  - Formas: BCNF:sim, 3NF:sim
- Pneu(idPneu, marca, rigidez, tipo, idMota->Mota)
  - FDs: idPneu->marca, rigidez, tipo, idMota
  - Formas: BCNF:sim, 3NF:sim
- Colaborador(idColaborador, nome, dataDeNascimento, idNacionalida->Nacionalidade)
  - FDs: idColaborador->nome, dataDeNascimento, idNacionalidade
  - Formas: BCNF:sim, 3NF:sim
- Engenheiro(idColaborador->Colaborador, ramo, idEquipa->Equipa)
  - FDs: idColaborador->ramo, idEquipa
  - Formas: BCNF:sim, 3NF:sim
- Piloto(idColaborador->Colaborador, numero, numeroPontos, tipo, idEquipa->Equipa, idMota->Mota)
  - FDs: idColaborador->numero, numeroPontos, tipo, idEquipa, idMota
  - Formas: BCNF:sim, 3NF:sim
- Circuito(idCircuito, nome, pais, perimetro, numSetores)
  - FDs: idCircuito->nome, pais, perimetro, numSetores
  - Formas: BCNF:sim, 3NF:sim
- Corrida(idCorrida, nome, data, numVoltas, idCircuito->Circuito)
  - FDs: idCorrida->nome, data, numVoltas, idCircuito
  - Formas: BCNF:sim, 3NF:sim
- Grid(idCorrida->Corrida, idColaborador->Piloto, posicaoInicial, posicaoFinal)
  - FDs: idCorrida, idColaborador->posicaoInicial, posicaoFinal
  - Formas: BCNF:sim, 3NF:sim

- ClassificacaoGeral(epoca, lugar, idColaborador->Piloto)
  - FDs: Triviais
  - Formas: BCNF:sim, 3NF:sim
- Evento(idEvento, volta, idCorrida->Corrida)
  - FDs: idEvento->volta, idCorrida
  - Formas: BCNF:sim, 3NF:sim
- Acidente(idEvento->Evento,setor)
  - FDs: idEvento->setor
  - Formas: BCNF:sim, 3NF:sim
- Outro(idEvento->Evento, descricao)
  - FDs: idEvento->descricao
  - Formas: BCNF:sim, 3NF:sim
- Ultrapassagem(idEvento->Evento, setor)
  - FDs: idEvento->setor
  - Formas: BCNF:sim, 3NF:sim
- Bandeira(idBandeira, cor)
  - FDs: idBandeira->cor
  - Formas: BCNF:sim, 3NF:sim
- PilotoCorrida(idPiloto->Piloto, idCorrida->Corrida)
  - FDs: Triviais
  - Formas: BCNF:sim, 3NF:sim
- PilotoEvento(idPiloto->Piloto, idEvento->Evento)
  - FDs: Triviais
  - Formas: BCNF:sim, 3NF:sim
- EventoBandeira(idEvento->Evento, idBandeira->Bandeira)
  - FDs: Triviais
  - Formas: BCNF:sim, 3NF:sim

Como em todas as relações e todas as dependências funcionais, em que A->B, o conjunto de atributos A permite identificar todos os atributos da relação, podemos concluir que A é uma *superkey*.

Assim sendo, todas as relações estão em Forma Normal de *Boyce-Codd*(BCNF) e na 3ª Forma Normal(3NF).

Lista e Forma de Implementação das Restrições

Marca:

- Não pode haver duas marcas com o mesmo ID
  - idMarca PRIMARY KEY
- Todas as marcas têm de ter um nome associado
  - nome NOT NULL
- Todas as marcas têm de ter um ano de ingresso associado, sendo este maior ou igual ao ano da primeira temporada da competição (1949)
  - anoDeIngresso NOT NULL CHECK (anoDeIngresso >= 1949)
- O ID da nacionalidade de uma marca corresponde ao ID de uma nacionalidade.
  - idNacionalidade REFERENCES Nacionalidade(idNacionalidade)

Equipa:

- Não pode haver duas equipas com o mesmo ID
  - idEquipa PRIMARY KEY
- Todas as equipas têm de ter um nome associado
  - nome NOT NULL
- Todas as equipas têm de ter um tipo, que deve ser [de] satélite ou [de] fábrica.
  - tipo TEXT NOT NULL CHECK (tipo = "FABRICA" OR tipo = "SATELITE")
- O ID da marca de uma equipa corresponde ao ID de uma marca.
  - idMarca REFERENCES Marca(idmarca)

Colaborador:

- Não pode haver dois colaboradores com o mesmo ID
  - idColaborador PRIMARY KEY
- Todos os colaboradores têm de ter um nome associado
  - nome NOT NULL
- Todos os colaboradores têm de ter uma data de nascimento
  - dataDeNascimento NOT NULL
- O ID da nacionalidade de um colaborador corresponde ao ID de uma nacionalidade.
  - idNacionalidade REFERENCES Nacionalidade(idNacionalidade)

Engenheiro:

- Não pode haver dois engenheiros com o mesmo ID, correspondendo este a um ID de um colaborador.
  - idColaborador PRIMARY KEY

- Todos os engenheiros têm de ter um ramo no qual estão especializados
    - ramo NOT NULL
- O ID da equipa de um engenheiro corresponde ao ID de uma equipa.
    - idEquipa REFERENCES Equipa(idEquipa)

Piloto:

- Não pode haver dois pilotos com o mesmo ID, correspondendo este a um ID de um colaborador.
    - idColaborador PRIMARY KEY
- Todos os pilotos têm de ter um número associado
    - numero NOT NULL
- Todos os pilotos têm de ter um número de pontos associado
    - numeroPontos NOT NULL
- Todos os pilotos têm de ter um tipo, que deve ser principal ou [de] reserva.
    - tipo TEXT NOT NULL CHECK(tipo = "PRINCIPAL" OR tipo = "RESERVA")
- O ID da equipa de um piloto corresponde ao ID de uma equipa.
    - idEquipa REFERENCES Equipa(idEquipa)

Mota:

- Não pode haver duas motas com o mesmo ID
    - idMota PRIMARY KEY
- Todas as motas têm de ter uma potência associada, que nunca pode ser nula ou negativa
    - potencia NOT NULL CHECK(potencia > 0)
- Todas as motas têm de ter um peso, que nunca pode ser nulo ou negativo
    - peso NOT NULL CHECK(peso > 0)
- Todas as motas têm de ter uma velocidade máxima atingida, que nunca pode ser maior que 340 km/h
    - velocidadeMax NOT NULL CHECK(velocidadeMax <= 340)
- O ID da equipa de uma mota corresponde ao ID de uma equipa.
    - idEquipa REFERENCES Equipa(idEquipa)

Pneu:

- Não pode haver dois pneus com o mesmo ID
    - idPneu PRIMARY KEY
- Todos os pneus têm de ter uma marca
    - marca NOT NULL
- Todos os pneus têm de ter uma rigidez própria, podendo estes ser macios, médios ou duros
    - rigidez NOT NULL CHECK(rigidez = "MACIO" OR rigidez = "MEDIO" OR rigidez = "DURO")
- Todos os pneus têm de ter um tipo, podendo estes ser traseiros ou dianteiros
    - tipo NOT NULL CHECK(tipo = "TRASEIRO" OR tipo = "DIANTEIRO")
- O ID da mota de um pneu corresponde ao ID de uma mota.
    - idMota REFERENCES Mota(idMota)

Corrida:

- Não pode haver duas corridas com o mesmo ID
  - idCorrida PRIMARY KEY
- Todas as corridas têm de ser nomeadas
  - nome NOT NULL
- Todas as corridas têm de ter uma data associada
  - data NOT NULL
- Todas as corridas têm de ter um número de voltas, sempre positivo
  - numVoltas NOT NULL CHECK(numVoltas > 0)
- O ID do circuito de uma corrida corresponde ao ID de um circuito.
  - idCircuito REFERENCES Circuito(idCircuito)

Circuito:

- Não pode haver dois circuitos com o mesmo ID
  - idCircuito PRIMARY KEY
- Todos os circuitos têm de ser nomeados
  - nome NOT NULL
- Todos os circuitos têm de estar associados a um país
  - pais NOT NULL
- Todos os circuitos têm de ter um perímetro, sempre positivo
  - perimetro NOT NULL CONSTRAINT CHECK(perimetro > 0)
- Todos os circuitos têm de ter um número de setores, sempre positivo
  - numSetores NOT NULL CONSTRAINT CHECK(numSetores > 0)

Grid:

- Não pode haver duas grids com o mesmo ID de corrida e de colaborador
  - PRIMARY KEY(idCorrida, idColaborador)
- O ID da corrida de uma grid corresponde ao ID de uma corrida.
  - idCorrida REFERENCES Corrida(idCorrida)
- O ID do colaborador de uma grid corresponde ao ID de colaborador de um piloto.
  - idPiloto REFERENCES Piloto(idPiloto)
- Todos as grids têm de ter uma posição inicial, sempre positiva
  - posicaoInicial NOT NULL CHECK(posicaoInicial > 0)
- Todos as grids têm de ter uma posição final, sempre positiva
  - posicaoFinal NOT NULL CHECK(posicaoFinal > 0)
- Um conjunto formado por um certo ID de corrida e uma certa posição inicial é único
  - UNIQUE(idCorrida, posicaoInicial)
- Um conjunto formado por um certo ID de corrida e uma certa posição final é único
  - UNIQUE(idCorrida, posicaoFinal)

Classificação Geral:

- O ID da colaborador de uma classificação final corresponde ao ID de colaborador de um piloto.
    - idColaborador REFERENCES Piloto(idColaborador)
- A época de uma classificação geral tem de representar um ano maior que 1949 (ano de início do Moto GP)
    - epoca CHECK(epoca >= 1949)
- O lugar de uma classificação geral tem de ser positivo
    - lugar CHECK(lugar > 0)
- Não pode haver duas classificações gerais associadas à mesma época , ao mesmo lugar, e ao mesmo ID de colaborador
    - PRIMARY KEY(epoca, lugar, idColaborador)

Evento:

- Não pode haver dois eventos com o mesmo ID
    - idEvento PRIMARY KEY
- Todos os eventos têm de ocorrer durante uma volta, sempre com número positivo
    - volta NOT CHECK(volta > 0)
- O ID da corrida de um evento corresponde ao ID de uma corrida.
    - idCorrida REFERENCES Corrida(idCorrida)

Ultrapassagem:

- Não pode haver duas ultrapassagens com o mesmo ID do evento, correspondendo este a um ID de um evento.
    - idEvento PRIMARY KEY REFERENCES Evento(idEvento)
- Todas as ultrapassagens têm de ter ocorrido num setor, sempre com número positivo
    - setor NOT NULL CHECK(setor > 0)

Acidente:

- Não pode haver dois acidentes com o mesmo ID do evento, correspondendo este a um ID de um evento.
    - idEvento PRIMARY KEY REFERENCES Evento(idEvento)
- Todos os acidentes têm de ter ocorrido num setor, sempre com número positivo
    - setor NOT NULL CHECK(setor > 0)

Outro:

- Não pode haver dois outros tipos de evento com o mesmo ID do evento, correspondendo este a um ID de um evento.
    - idEvento PRIMARY KEY REFERENCES Evento(idEvento)
- Todos esses outros tipos de evento têm de ter uma descrição associada
    - descricao NOT NULL

Bandeira:

- Não pode haver duas bandeiras com o mesmo ID
  - idBandeira PRIMARY KEY
- Todas as bandeiras têm de ter uma cor
  - cor NOT NULL

PilotoCorrida:

- O ID de um colaborador corresponde ao ID de colaborador de um piloto
  - idColaborador REFERENCES Piloto(idColaborador)
- O ID de uma corrida corresponde ao ID da tabela corrida
  - idCorrida REFERENCES Corrida(idCorrida)
- Não pode haver um tuplo com a mesma conjugação de idColaborador e idCorrida.
  - PRIMARY KEY(idColaborador, idCorrida)

PilotoEvento:

- O ID de um colaborador corresponde ao ID de colaborador de um piloto
  - idColaborador REFERENCES Piloto(idColaborador)
- O ID de um evento corresponde ao ID da tabela evento
  - idEvento REFERENCES Evento(idEvento)
- Não pode haver um tuplo com a mesma conjugação de idColaborador e idEvento.
  - PRIMARY KEY(idColaborador, idEvento)

EventoBandeira:

- O ID de um evento corresponde ao ID da tabela evento
  - idEvento REFERENCES Evento(idEvento)
- O ID de uma Bandeira corresponde ao ID da tabela bandeira
  - idBandeira REFERENCES Bandeira(idBandeira)
- Não pode haver um tuplo com a mesma conjugação de idEvento e idBandeira.
  - PRIMARY KEY(idEvento, idBandeira)

Nacionalidade:

- Não pode haver duas nacionalidades com o mesmo ID
  - idNacionalidade PRIMARY KEY
- Todas as nacionalidades têm de ter um nome atribuído
  - nome NOT NULL

As restrições aplicadas a atributos que estão relacionados entre duas tabelas, como por exemplo a volta de um evento ser menor que o número de voltas da corrida em questão, serão implementadas com *triggers* na terceira entrega. Além do mais, apesar de não mencionadas as restrições ON DELETE e ON UPDATE, estas foram implementadas na criação da base de dados no ficheiro “criar.sql”.