

Projeto de Bases de **Dados**

1ª Entrega

Trabalho realizado por:

António Azevedo, up202108689

José Martins, up202108794

INDÍCE

A. Contexto da Base de Dados	2
B. Definição do Modelo Conceptual	7
C. Definição do Esquema Relacional	8
D. Dependências Funcionais e Formas Normais	9
E. Restrições e implementações	13
F. Avaliação de cada membro do Grupo:	18

A. Contexto da Base de Dados

Introdução

O objetivo deste trabalho é criar uma base de dados que permita ao utilizador gerir dados sobre o campeonato brasileiro de futebol, podendo o mesmo verificar desde os dados sobre as equipas, jogadores, resultados (podendo estes ser verificados jornada a jornada), e verificar a cada jornada as equipas que se encontram em condições de ir à Copa Libertadores e as que se encontram em risco de despromoção.

Jornada

A Jornada é a classe mais importante da base de dados. É o que separa e organiza os dados, visto que os resultados da nossa base de dados são organizados por jornadas. Cada uma será identificada pelo seu ID e em cada uma existirão 10 jogos. Para além disso, será possível verificar em cada jornada a classificação de cada equipa.

Jogo

Na base de dados deverão existir imensos jogos, sendo estes distinguidos pelo seu ID, devendo existir especificamente 10 jogos por jornada (na primeira povoação da base de dados não irão ser implementados os 10 jogos por jornada pela enorme quantidade de dados). Cada jogo terá uma data em que acontece. Em cada jogo devem existir vários eventos, desde Faltas, Cartões, Golos e Substituições. Para além disso, a cada jogo deve ser atribuído um árbitro, e deve ser especificada a Equipa Visitada(EquipaCasa) e a Equipa visitante(EquipaFora). Para além disso, o resultado será do tipo TEXT, mas servirá apenas para verificação do resultado, nunca para cálculos de classificação.

Equipa

A classe equipa é uma das classes centrais da base de dados, afinal, um campeonato só funciona com equipas. Cada equipa terá um nome e será identificada pelo seu ID. Uma equipa terá vários jogadores e um único treinador, não podendo ter nenhum Arbitro. Será possível verificar a classificação de cada equipa numa jornada, ou seja, será possível verificar os pontos da equipa, o lugar em que se encontra e será possível verificar se a equipa está em condições de ir às competições sul-americanas ou se está em risco de despromoção.

Pessoa

Para a formação das equipas e do jogo, como já verificamos, será necessário existirem jogadores e treinadores. Para facilitar a estruturação dos dados, criamos uma classe Pai destas três, a classe Pessoa. Cada pessoa terá Nome, Idade e Nacionalidade.

Treinador e Jogador

As classes Treinador e Jogador são classes filhas de Pessoa, tendo os seus objetos portanto todos os atributos da sua Classe Pai. Ambas estão associadas a uma equipa e não podem estar associados a mais do que uma equipa na mesma jornada.

Evento

Durante um jogo acontecem diversos eventos, e, como nos foi pedido que fosse possível verificar os vários golos em cada jogo e os seus marcadores, então decidimos que seria interessante adicionarmos também os cartões ao longo dos jogos e as substituições que acontecem ao longo do mesmo. Assim, criamos uma classe evento, de modo a que pudéssemos generalizar estes acontecimentos. Cada evento terá um minuto de jogo associado e um jogador que terá um significado diferente em cada.

Cartao

A classe Cartao é uma classe filha da classe Evento. Cada Cartao terá a sua própria cor: Amarelo ou Vermelho. Para além disso, o jogador a que está associado é o jogador que leva o cartão.

Golo

A classe Golo é uma classe filha da classe Evento. O jogador associado a partir de Evento é o jogador que marca o golo. Cada Golo estará associado a uma Equipa, que será a equipa a favor do qual o golo irá contar. Ou seja, se a Equipa associada a Golo for a equipa A, mesmo que o jogador que marque o golo seja da Equipa B (ou seja, no caso de acontecer um auto-golo).

Substituição

A classe Substituição é uma classe filha da classe Evento. Nesta classe, existirá o jogador associado por ser uma classe filha de Evento, que será o jogador substituído, ou seja, o jogador que sairá do jogo. No entanto, o mesmo tem outra ligação a outro jogador, que será o jogador substituto, ou seja, o jogador que irá entrar em jogo. Será necessário impedir que um jogador que foi substituído não possa voltar a entrar em jogo.

Convocatória

Na liga brasileira de futebol existe uma regra que implica que não possam existir mais do que 5 jogadores estrangeiros por cada equipa a serem convocados por jogo. Assim, foi necessário criar esta classe, que possui 36 jogadores (18 por equipa). Teremos então de verificar que cada equipa não coloca mais do que 5 jogadores estrangeiros em cada convocatória, o que será implementado através de um gatilho apenas na segunda entrega. Para além disso, deve ser verificado que existe exatamente uma convocatória por jogo.

Classificação

Finalmente, a classe Classificação foi feita de modo a que o utilizador consiga verificar os pontos de uma equipa em cada jornada, podendo portanto verificar a posição de cada equipa e se cada um está em condições de ir às competições sul-americanas ou se está em risco de despromoção. A classificação será calculada percorrendo todos os jogos nas primeiras x jornadas, e irá atribuir a cada equipa:

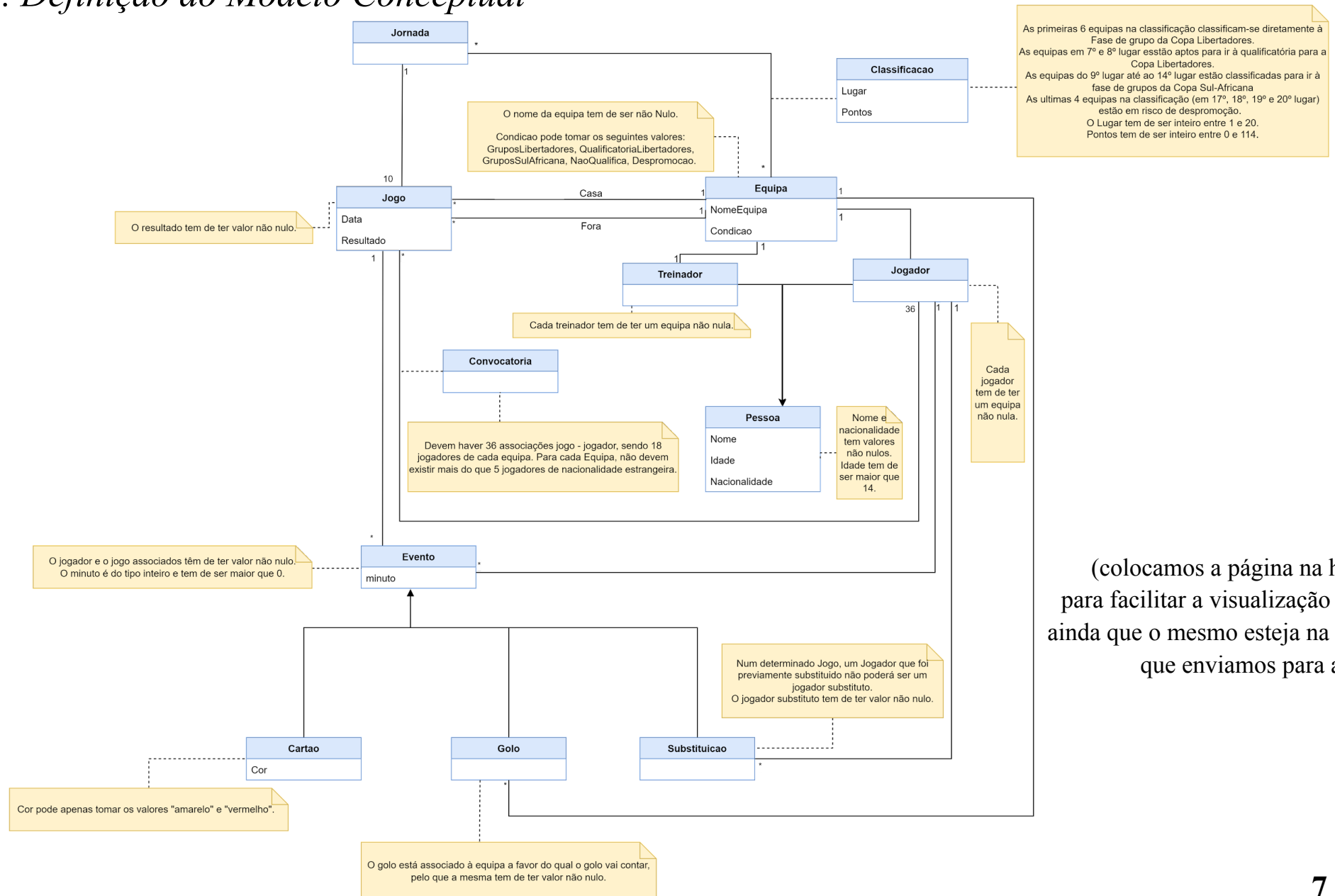
- 3 Pontos, no caso de vitória;
- 1 Ponto, no caso de empate;
- 0 Pontos, no caso de derrota;

No fim, ordenam-se as equipas por pontos e será possível calcular o atributo condicao de cada equipa a partir das seguintes condições:

- As primeiras 6 equipas na classificação classificam-se diretamente à Fase de grupo da Copa Libertadores,
- As equipas em 7º e 8º lugar estão aptos para ir à qualificatória para a Copa Libertadores;
- As equipas do 9º lugar até ao 14º lugar estão classificadas para ir à fase de grupos da Copa Sul-Africana;
- As últimas 4 equipas na classificação (em 17º, 18º, 19º e 20º lugar) estão em risco de despromoção.

No entanto esta última parte do cálculo da condicao será apenas feito na segunda entrega do projeto.

B. Definição do Modelo Conceptual



(colocamos a página na horizontal para facilitar a visualização do UML, ainda que o mesmo esteja na pasta .zip que enviamos para a entrega)

C. Definição do Esquema Relacional

Jornada (IdJornada)

Jogo (IdJogo, Data, IdJornada ->Jornada, IdEquipaFora -> Equipa, IdEquipaCasa -> Equipa, Resultado)

Equipa (IdEquipa, NomeEquipa, condicao)

Pessoa (IdPessoa, Nome, Idade, Nacionalidade)

Treinador (IdPessoa -> Pessoa, IdEquipa -> Equipa)

Jogador (IdPessoa -> Pessoa, IdEquipa -> Equipa)

Evento (IdEvento, IdJogo -> Jogo, IdPessoa -> Jogador, Minuto)

Cartao (IdEvento -> Evento, Cor)

Golo (IdEvento -> Evento, IdEquipa -> Equipa)

Substituição (IdEvento -> Evento, IdJogadorSubstituto -> Jogador)

Convocatória (IdJogador -> Jogador, IdJogo -> Jogo)

Classificação (IdEquipa -> Equipa, IdJornada -> Jornada, Pontos, Lugar)

D. Dependências Funcionais e Formas Normais

De modo a justificar as dependências funcionais que se encontram em BCNF e em 3NF, sabemos que para se encontrar em:

- BCNF, em qualquer dependência $A \rightarrow B$ não trivial, A é uma chave ou superchave.
- 3NF, em qualquer dependência $A \rightarrow B$ não trivial, A é uma chave ou super chave, ou B é membro de uma chave da relação.

Jornada:

Não existem dependências funcionais para jornada, uma vez que a mesma tem apenas o seu ID como atributo e identificador.

Jogo:

Dependências Funcionais na 3NF:

- $\{IdJogo\} \rightarrow \{IdJornada, IdEquipaFora, IdEquipaCasa, IdArbitro, IdConvocatoria, Resultado\}$
- $\{IdEquipaCasa, IdEquipaFora\} \rightarrow \{IdJogo\}$
- $\{IdEquipaFora, IdJornada\} \rightarrow \{IdJogo\}$
- $\{IdEquipaCasa, IdJornada\} \rightarrow \{IdJogo\}$

BCNF:

- $\{IdJogo\} \rightarrow \{IdJornada, IdEquipaFora, IdEquipaCasa, IdConvocatoria, Resultado\}$

Equipa:

Dependências Funcionais na 3NF:

- $\{IdEquipa\} \rightarrow \{NomeEquipa, condicao\}$
- $\{NomeEquipa\} \rightarrow \{IdEquipa\}$

BCNF:

- $\{IdEquipa\} \rightarrow \{NomeEquipa, condicao\}$

Pessoa:

$\{IdPessoa\} \rightarrow \{Nome, Idade, Nacionalidade\}$

As dependências funcionais de Pessoa encontram-se tanto na BCNF como na 3NF porque a única dependência é que o seu ID implique todos os seus dados. Não existe nenhuma relação entre os outros atributos que implique outros.

Treinador:

$\{IdPessoa\} \rightarrow \{IdEquipa\}$

As dependências funcionais encontram-se tanto na BCNF como na 3NF porque não existem outras dependências funcionais e esta cumpre as regras explicadas no início desta secção.

Jogador:

$\{IdPessoa\} \rightarrow \{IdEquipa\}$

As dependências funcionais encontram-se tanto na BCNF como na 3NF porque não existem outras dependências funcionais e esta cumpre as regras explicadas no início desta secção.

Evento:

$\{\text{IdEvento}\} \rightarrow \{\text{IdJogo}, \text{IdPessoa}, \text{Minuto}\}$

Exceto o IdEvento nenhum outro atributo pode implicar outro, portanto podemos concluir que as dependências funcionais encontram-se tanto na BCNF como na 3NF uma vez que não existem outras dependências funcionais, e esta cumpre as regras explicadas no início desta secção.

Cartão:

$\{\text{IdEvento}\} \rightarrow \{\text{Cor}\}$

A dependência funcional de Cartão encontra-se tanto na BCNF como na 3NF porque a única dependência é que o seu ID implique todos os seus dados. Não existe nenhuma relação entre os outros atributos que implique outros. Neste caso, o IdEvento irá implicar qual é a cor do cartão, mas a cor do cartão nunca poderá indicar qual é o evento, até porque no mesmo jogo podem haver vários cartões amarelos, por exemplo.

Golo:

$\{\text{IdEvento}\} \rightarrow \{\text{Equipa}\}$

A dependência funcional de Golo encontra-se tanto na BCNF como na 3NF porque a única dependência é que o seu ID implique todos os seus dados. Não existe nenhuma relação entre os outros atributos que implique outros. Neste caso, o IdEvento irá implicar a favor de qual equipa é o golo, mas o golo nunca poderá indicar qual é o evento, até porque no mesmo jogo podem haver vários golos a favor da mesma equipa, por exemplo.

Substituição:

{IdEvento} -> {IdJogadorSubstituto}

A dependência funcional de Substituição encontra-se tanto na BCNF como na 3NF porque a única dependência é que o seu ID implique todos os seus dados. Não existe nenhuma relação entre os outros atributos que implique outros. Por exemplo, em vários jogos um específico jogador pode ser substituído de campo várias vezes.

Convocatória:

Não existem dependências funcionais, uma vez que a convocatória é uma classe de associação entre jogador e jogo e a sua chave primária é {IdJogo, IdJogador}.

Classificação:

{IdJornada, IdEquipa} -> {Pontos, Lugar }

Não existe mais nenhuma dependência funcional, e visto que a dependência funcional existente verifica as regras explicadas no início desta secção, podemos concluir que esta dependência funcional está tanto na BCNF como na 3NF.

E. Restrições e implementações

Jornada:

- O ID da Jornada será primary key da Jornada, portanto será único e não nulo;
 - idJornada PRIMARY KEY (idJornada)

Jogo:

- O ID do Jogo será primary key da classe, portanto será único e não nulo;
 - idJogo PRIMARY KEY (idJogo)
- Um jogo estará associado a exatamente uma Jornada, exatamente duas Equipas, uma em “casa” e outra “fora”, exatamente uma Data e exatamente a um Árbitro, todas estas remetendo para a sua respectiva tabela, e todas estas sendo não nulas;
 - idJornada FOREIGN KEY (idJornada) REFERENCES JORNADA(idJornada)
 - idEquipaFora FOREIGN KEY (idEquipaFora) REFERENCES EQUIPA(idEquipa)
 - idEquipaCasa FOREIGN KEY (idEquipaCasa) REFERENCES EQUIPA(idEquipa)
 - idArbitro FOREIGN KEY (idArbitro) REFERENCES ARBITRO(idArbitro)
- Um jogo terá um resultado com valor não nulo.
 - Resultado NOT NULL

Equipa:

- O ID da Equipa será primary key, portanto será único e não nulo;
 - idEquipa PRIMARY KEY (idEquipa)
- O nome da equipa deverá ser único e não nulo, pois não poderá existir duas equipas com o mesmo nome, nem uma equipa sem nome;
 - NomeEquipa UNIQUE NOT NULL

Pessoa:

- O ID da Pessoa será primary key, portanto será único e não nulo;
 - idPessoa PRIMARY KEY (idPessoa)
- O nome, idade e nacionalidade são não nulos;
 - nome NOT NULL
 - idade NOT NULL
 - nacionalidade NOT NULL
- A menor idade aceite seria de um jogador, visto que os treinadores relacionados ao campeonato têm idades superiores a 18. Como um jogador pode assinar contrato a partir dos 14 anos, então idade tem de ter valor igual ou superior a 14;
 - CHECK (idade >= 14)

Treinador:

- O idPessoa será a primary key do Treinador, portanto terá de ser único e não nulo;
 - idPessoa PRIMARY KEY (idPessoa) REFERENCES PESSOA(idPessoa)
- Cada Treinador terá uma única equipa não nula;
 - idEquipa FOREIGN KEY (idEquipa) REFERENCES Equipa(idEquipa)

Jogador:

- O idPessoa será a primary key do Jogador, portanto terá de ser único e não nulo;
 - idPessoa PRIMARY KEY (idPessoa) REFERENCES PESSOA(idPessoa)
- Cada Jogador terá de pertencer a uma única equipa não nula;
 - idEquipa FOREIGN KEY (idEquipa) REFERENCES Equipa(idEquipa)

Evento:

- O ID do Evento será primary key, portanto será único e não nulo;
 - idEvento PRIMARY KEY (idEvento)
- Cada evento estará associado a um Jogo e a um Jogador, tendo estes de ser não nulos;
 - idJogo FOREIGN KEY (idJogo) REFERENCES JOGO(idJogo)
 - idJogador FOREIGN KEY (idJogador) REFERENCES JOGADOR(idPessoa)
- Cada evento estará associado a um minuto onde o mesmo ocorreu, sendo este não nulo;
 - minuto NOT NULL
 - CHECK (minuto >= 0)

Cartão:

- O idEvento será primary key do Cartão, logo será único e não nulo;
 - idEvento PRIMARY KEY (idEvento) REFERENCES EVENTO(idEvento)
- Um Cartão será sempre de cor amarela ou cor vermelha, logo o atributo cor será não nulo e restrito a essas duas opções;
 - CHECK (cor = amarelo OR cor = vermelho)

Golo:

- O idEvento será primary key do Golo, logo será único e não nulo;
 - idEvento PRIMARY KEY (idEvento) REFERENCES EVENTO(idEvento)
- Um golo estará sempre associado à equipa marcadora do mesmo, logo o atributo idEquipa será único e não nulo;
 - idEquipa FOREIGN KEY (idEvento) REFERENCES EQUIPA(idEquipa)

Substituição:

- O idEvento será primary key da Substituição, logo será único e não nulo;
 - idEvento PRIMARY KEY (idEvento) REFERENCES EVENTO(idEvento)
- Uma Substituição estará sempre associada a um jogador substituto, logo o atributo idJogador será único e não nulo;
 - idJogador FOREIGN KEY (idEvento) REFERENCES JOGADOR(idPessoa)
- No mesmo Jogo, um jogador que foi previamente substituído não poderá ser um jogador substituto (será implementado para a segunda entrega)

Convocatória:

- Uma convocatória estará associada a exatamente um Jogo e a 36 Jogadores diferentes, 18 de cada equipa do Jogo associado, logo os atributos idJogo e idJogador serão únicos e não nulos;
 - idJogo FOREIGN KEY (idJogo) REFERENCES JOGO(idJogo)
 - idJogador FOREIGN KEY (idJogo) REFERENCES JOGADOR(idJogador)
- Numa mesma convocatória, uma equipa não poderá ter mais que 5 jogadores convocados com nacionalidade não brasileira;
 - Esta restrição será implementada com um gatilho

Classificação:

O cálculo da classificação é feito com gatilhos, pelo que será implementado apenas na segunda entrega do projeto.

- Cada pontuação terá de ser maior ou igual a 0, e menor ou igual a 114;
 - CHECK(Pontos >= 0 AND Pontos <=114)
- Cada lugar terá de ser maior que 1 e menor que 20;
 - CHECK(Lugar >= 1 AND Lugar <= 20)

F. Avaliação de cada membro do Grupo:

O grupo trabalhou em conjunto e houve bastante pesquisa feita pelos dois membros. O António trabalhou maioritariamente em SQL e na pesquisa de dados. O José trabalhou principalmente na elaboração do relatório, UML e na estruturação dos dados. Na avaliação, consideramos que ambos os membros investiram a mesma quantidade de esforço para o trabalho, pelo que consideramos que ambos deveremos ter a mesma classificação.