



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2020/2021

FuteProject

Gonçalo Almeida A88292

Luís Pereira A89478

Diogo Amaro A89565

Leonardo Marreiros A89537

Dezembro, 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

FuteProject

Gonçalo Almeida A88292

Luís Pereira A89478

Diogo Amaro A89565

Leonardo Marreiros A89537

Dezembro, 2020

Resumo

Este relatório foi desenvolvido no âmbito do desenvolvimento de uma Base de Dados para a empresa FuteProject, que foca a sua atividade na área do recrutamento de jogadores. Este projeto foi desenvolvido com o objetivo de tornar digital a base de dados da agência. Os relacionamentos entre agentes, jogadores e olheiros foram todos considerados e implementados da forma como os gerentes da empresa nos informaram. O papel dos agentes é recrutar jogadores e atualizar as informações dos seus jogadores. Em relação aos olheiros, estes fazem a avaliação de jogadores.

Neste relatório são apresentadas todas as fases de desenvolvimento da base de dados até à sua implementação física. Começamos por analisar o método de atuação da empresa, foram feitas entrevistas com os colaboradores e agendadas reuniões com os representantes da FuteProject. Com isto levantamos os requisitos em conjunto com o cliente o qual também os aprovou e conduziu ao início da modulação concetual.

A modelação conceptual foi desenvolvida com recurso à ferramenta “brModelo”. Este mesmo modelo foi analisado e aprovado pelos representantes da agência. O passo seguinte foi o desenvolvimento do modelo lógico, na ferramenta “MySQL Workbench” também, ele validado pelo cliente.

Após verificar que todos os requisitos eram suportados, este modelo foi implementado fisicamente. O modelo físico foi implementado com recurso a “MySQL” e “MySQL Workbench”.

Finalmente, foi dado o selo de aprovação junto com o cliente, o que deu por terminado o nosso projeto.

Área de Aplicação: Desenho, Arquitetura, Desenvolvimento e Implementação de Sistemas de Bases de Dados

Palavras-Chave: Bases de Dados, Levantamento de Requisitos, Entidades, Atributos, Relacionamentos, Modelo Concetual, Modelo Lógico, Modelo Físico, *MySQL*, *MySQL Workbench*

Índice

Resumo	i
Índice de Figuras	iv
Índice de Tabelas	v
1 Introdução	1
1.1 Contextualização	1
1.2 Apresentação do Caso de Estudo	1
1.3 Motivação e Objetivos	2
1.4 Estrutura do Relatório	2
2 Levantamento e análise de requisitos	3
2.1 Método de Levantamento e de análise de requisitos adotados	3
2.2 Especificação de requisitos	3
2.2.1 Requisitos de descrição	3
2.2.2 Requisitos de exploração	4
2.2.3 Requisitos de controlo	4
2.2.4 Análise geral dos requisitos	5
3 Modelação Conceptual	6
3.1 Apresentação da abordagem de modelação realizada	6
3.2 Identificação e caracterização das entidades	6
3.3 Identificação e caracterização dos relacionamentos	8
3.4 Identificação e Caracterização da Associação dos Atributos com as Entidades e Relacionamentos	13
3.4.1 Domínio dos Atributos	15
3.4.2 Determinação das chaves candidatas, chaves primárias e chaves alternadas	19
3.5 Detalhe ou generalização de entidades	20
3.6 Apresentação e explicação do diagrama ER	21
3.7 Validação do Modelo de Dados com o Utilizador	21
4 Modelação Lógica	24
4.1 Construção e validação dos modelos dados lógicos	24
4.1.1 Relacionamento binário (1: N)	24
4.1.2 Relacionamentos de muitos para muitos (N:M)	27
4.1.3 Entidades resultantes	27
4.2 Desenho do Modelo Lógico	28
4.3 Validação do Modelo através da Normalização	28
4.4 Validação do Modelo com as Interrogações do Utilizador	29
4.5 Revisão do Modelo Lógico com o Utilizador	31
5 Implementação Física	32
5.1 Seleção do Sistema	32

5.2	Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	32
5.3	Tradução das interrogações do utilizador para SQL	32
5.4	Escolha, definição e caracterização de índices em SQL	34
5.5	Estimativa do espaço em disco da base de dados e taxa de crescimento anual	35
5.6	Definição e caracterização das vistas de utilização em SQL	40
5.7	Definição e Caracterização dos Mecanismos de Segurança em SQL	41
5.8	Revisão do sistema implementado	42
6	Conclusão e Trabalho Futuro	44
7	Referências	45
8	Anexos	46
8.1	Anexo 1 – <i>Script</i> de Inicialização da Base de Dados	46
8.2	Anexo 2 – <i>Script</i> de Povoamento Inicial	50
8.3	Anexo 3 - Funcionalidades para o Agente	55
8.4	Anexo 4 - Funcionalidades para o Olheiro	57
8.5	Anexo 5 - <i>Script</i> de Criação das Vistas que respondem a parte das Interrogações	58
8.6	Anexo 6 - Funcionalidades Comuns ao Olheiro e Agente	58
8.7	Anexo 7 - Requisitos de Exploração	60

Índice de Figuras

Figura 1 - Relacionamento Jogador - Posição	8
Figura 2 - Relacionamento Jogador - Agente	8
Figura 3 - Relacionamento Jogador - Olheiro	9
Figura 4 - Relacionamento Jogador - Clube	9
Figura 5 - Relacionamento Jogador - Tipo	10
Figura 6 - Relacionamento Jogador - Localidade	10
Figura 7 - Relacionamento Localidade - País	11
Figura 8 - Relacionamento Jogador - Clube_anterior	11
Figura 9 - Relacionamento Jogador - Nacionalidade	12
Figura 10 - Relacionamento Olheiro - Nacionalidade	12
Figura 11 - Relacionamento Agente - Nacionalidade	13
Figura 12 - Modelo Concetual.	21
Figura 13 - - Modelo Lógico	28
Figura 14 - - Árvore representativa da Interrogação nº 1.	29
Figura 15 - Árvore representativa da Interrogação nº2	30
Figura 16 - Árvore representativa da Interrogação nº3	31
Figura 22- - Imagem do código obtido para a <i>query</i> nº 26	32
Figura 23- Imagem do código obtido para a <i>query</i> nº 28	33
Figura 24- Imagem do código obtido para a <i>query</i> nº 12	33
Figura 25- Imagem do código obtido para a <i>query</i> nº 21	33
Figura 26 - Índice idx_nome_jogador	34
Figura 27 - Índice idx_clube_antigo	34
Figura 28 - <i>view</i> para a <i>query</i> nº 19	40
Figura 29 - <i>view</i> para a <i>query</i> nº 31	40
Figura 30 - <i>view</i> para a <i>query</i> nº28	41
Figura 31 - Criação e atribuição de permissões do Agente e do Olheiro	42

Índice de Tabelas

Tabela 1 - Caracterização de todos os atributos existentes	15
Tabela 2 - Caracterização dos atributos do Olheiro	15
Tabela 3 - Caracterização dos atributos do Jogador	16
Tabela 4 - Caracterização dos atributos do Agente	16
Tabela 5 - Caracterização dos atributos do Tipo	17
Tabela 6 - Caracterização dos atributos da Localidade	17
Tabela 7 - Caracterização dos atributos do País	17
Tabela 8 - Caracterização dos atributos da Posição	17
Tabela 9 - Caracterização dos atributos da Nacionalidade	17
Tabela 10 - Caracterização dos atributos do Clube	18
Tabela 11 - Caracterização dos atributos do Clube anterior	18
Tabela 12 - Caracterização de atributos do relacionamento Jogador-Olheiro	18
Tabela 13 - Caracterização de atributos do relacionamento Jogador-Clube anterior	18
Tabela 14 - Espaço ocupado no disco por cada tipo de dados	35
Tabela 15- Espaço ocupado no disco por cada entrada na tabela Clube_anterior	35
Tabela 16 - Espaço ocupado por cada entrada na tabela Jogador	36
Tabela 17 - Espaço ocupado por cada entrada na tabela Agente	36
Tabela 18 - Espaço ocupado por cada entrada na tabela Olheiro	37
Tabela 19 - Espaço ocupado por cada entrada na tabela Nacionalidade	37
Tabela 20 - Espaço ocupado por cada entrada na tabela Pais	37
Tabela 21 - Espaço ocupado por cada entrada na tabela Clube	37
Tabela 22 - Espaço ocupado por cada entrada na tabela Localidade	38
Tabela 23 - Espaço ocupado por cada entrada na tabela Historico_Clubes	38
Tabela 24 - Espaço ocupado por cada entrada na tabela Posicao	38
Tabela 25 - Espaço ocupado por cada entrada na tabela Tipo	39
Tabela 26 - Espaço total ocupado pela base de dados	39

1 Introdução

1.1 Contextualização

A FuteProject foi criada por um pequeno grupo de agentes, no início das suas carreiras, que queriam demonstrar o perfil dos seus jogadores ao maior número de equipas possível, do modo mais simples possível, a fim de encontrar a equipa certa para os seus jogadores. Assim que os agentes fundadores encontram um jogador que querem representar este é inserido na FuteProject. Aquando da abertura da sua agência, uma equipa interessada em comprar um jogador teria de se deslocar às instalações da FuteProject e verificar os seus catálogos de forma a consultar as estatísticas e perfis dos jogadores. No entanto, com vista à expansão rápida da empresa, esta situação revelou-se ser altamente ineficiente uma vez que as equipas de maior calibre não possuíam meios de, não só deslocar-se às instalações da FuteProject, como até de saber da sua existência devido à falta da vertente online desta agência.

Os responsáveis da FuteProject viram o seu crescimento estagnado, apesar do seu enorme potencial. Esta situação conduziu à procura de uma solução que voltasse a colocar a agência no caminho para o sucesso.

1.2 Apresentação do Caso de Estudo

Os treinadores de futebol necessitam uma forma rápida e eficaz de poder averiguar jogadores com as características desejadas para as suas equipas. A FuteProject (FP) fornece para cada um dos jogadores inscritos na sua agência com um perfil detalhado: em que clube jogaram, posição que ocupam, sua reputação entre outros dados relevantes sobre o mesmo.

Para além de novos jogadores contratados para representar a agência, há também uma secção em que novos talentos se podem inscrever e mostrar o que valem para os nossos olheiros e, caso consigam passar nos seus testes, serão registados oficialmente na FP.

Seria de esperar que, sendo uma empresa em crescimento, tivessem um website em que fosse possível verificar quais os jogadores que fazem parte da agência de forma a facilitar o processo de equipas que estivessem à procura de contratações. Todavia, todo este processo é ainda feito analogicamente nos catálogos físicos da agência.

Conscientes deste problema, a FP decidiu a criação de um website onde fosse possível consular todas as informações acerca dos seus jogadores. Para isso, é preciso a criação de uma

base de dados que seja capaz de separar os jogadores de acordo as especificações pedidas pelas equipas e mostrar o perfil futebolístico do jogador em questão.

1.3 Motivação e Objetivos

Ao se aperceberem da necessidade de uma fonte de jogadores por explorar, a necessidade dos treinadores localizarem jovens talentos de uma forma simples e um forte crescimento de jogadores inscritos nesta agência, a FP decidiu criar um website com uma base de dados subjacente onde podem editar e inserir novos jogadores ao mesmo tempo que expandem o seu mercado-alvo para equipas de maior dimensão.

O objetivo deste projeto é criar uma base de dados que reúna todos os jogadores inscritos nesta agência e facilmente organizá-los de acordo com as suas características. Outro objetivo é que jovens promessas se possam registar com os seus dados mais relevantes e, de seguida, um olheiro vai observá-los e decidir se têm talento ou não para fazer parte da FuteProject.

1.4 Estrutura do Relatório

Este relatório é composto por seis capítulos.

Primeiro começamos por contextualizar e apresentar os objetivos do projeto.

O segundo capítulo, Levantamento e Análise de Requisitos, trata de como foi feito o levantamento dos requisitos. Adicionalmente, são divididos os requisitos em requisitos de descrição, exploração e controlo, assim como a análise dos mesmos.

Na Modelação Conceptual, são descritas as entidades do problema, os respetivos atributos e os relacionamentos entre as entidades. É também apresentado a caracterização e domínio dos atributos. As chaves candidatas, primárias e secundárias são também apresentadas. Por último é feita uma validação do modelo de dados resultante com o utilizador.

Em Modelação Lógica, começamos por construir o modelo lógico e explicados os diferentes relacionamentos e entidades resultantes. De seguida validamos o modelo através da normalização e é validado com interrogações do utilizador. Esta secção terminou com a revisão do modelo lógico com o utilizador.

No quinto capítulo, Implementação Física é apresentado o modelo de sistema que irá ser utilizado e a conversão do modelo lógico para o sistema de gestão de bases de dados. Além disso, as interrogações que foram antes colocadas pelo utilizador são traduzidas para SQL. Foram apresentados os índices e vistas implementados e estimado o espaço em disco da base de dados necessário e taxa de crescimento anual esperada. Finalmente, foram apresentados os mecanismos de segurança em SQL e revisto o sistema implementado com o cliente.

Finalmente, na Conclusão e Trabalho Futuro, foi feita uma retrospectiva do trabalho assim como vantagens e dificuldades na implementação da base de dados.

2 Levantamento e análise de requisitos

Imaginemos uma reunião entre treinadores de um clube à procura de novos talentos. Vários elementos dessa reunião têm opinião diferente sobre qual o jogador a contratar. Até que um deles começa a enunciar as estatísticas de cada opção em estudo. Todos ficam interessados e atentos. São referidos vários aspetos importantes como valores de mercado, estatísticas individuais, e reputação. Graças a esta informação todos concordam, baseado nas estatísticas, qual o jogador a ser contratado. Logicamente, perguntam ao seu colega de trabalho onde encontrou tanta informação ao que ele responde que foi graças à agência FuteProject e à sua organizada base de dados.

2.1 Método de Levantamento e de análise de requisitos adotados

Como forma de levantamento dos requisitos, recorreremos a várias técnicas de *fact-finding* para tentar perceber o comportamento do sistema atual com o intuito de garantir que o resultado satisfaz as necessidades da empresa.

Primeiro, começamos por observar o funcionamento da agência: o modo como os funcionários interagem com o sistema atual, as funcionalidades e as limitações deste, etc. O que permitiu compreender o serviço prestado e deu aso a um maior entendimento das possibilidades de exploração e administração do mesmo. Além disso, foi analisado o sistema de informação atual que permitiu compreender o tipo de dados armazenados e a forma como estão ligados. Finalmente, foram agendadas entrevistas onde foram discutidas diferentes vertentes que permitiram perceber o domínio do problema, as limitações sentidas pelos colaboradores, assim como pontos que deveriam ser implementados no novo sistema.

2.2 Especificação de requisitos

De acordo com o que foi discutido com os representantes da agência, foram identificados os seguintes requisitos:

2.2.1 Requisitos de descrição

1. Os agentes devem ser registados, sendo-lhes atribuído um ID. Deve ser fornecido um Nome, Nacionalidade, Data de Nascimento, Telemóvel, Email, Reputação e Idade.
2. Os olheiros devem ser registados, sendo-lhes atribuído um ID. Deve ser fornecido um Nome, Nacionalidade, Data de Nascimento, Telemóvel e Reputação.
3. Os jogadores devem ser registados, sendo-lhes atribuído um ID. Deve ser fornecido um Nome, Nacionalidade, Data de Nascimento, Agente que o representa, Olheiro que o

possa ter descoberto, Data de início de contrato assim como a de término, Localidade e País onde nasceu e perfil futebolístico.

4. Os jogadores podem-se representar em dois grupos: jovens promessas e jogadores agenciados.
5. Todos os jogadores agenciados existentes têm obrigatoriamente de ter um agente associado.
6. Os jogadores que são jovens promessas não tem um agente associado.
7. Uma mudança de clube apenas pode ser partilhada quando o agente a validar.
8. Quando um jogador muda de clube, o histórico de clubes deste deve ser atualizado.
9. Olheiros podem comentar e/ou avaliar os jogadores.
10. No final de cada época as estatísticas dos jogadores são atualizadas.

2.2.2 Requisitos de exploração

11. Obter número de jogadores agenciados.
12. Obter número de jogadores de uma determinada nacionalidade.
13. Consultar jogadores por posição.
14. Obter top 3 de agentes com mais jogadores associados.
15. Obter top 3 de jogadores com mais golos marcados.
16. Obter top 3 de jogadores de acordo com a reputação.
17. Obter top 3 de jogadores de acordo com o valor de mercado.
18. Obter lista de jogadores que jogaram num determinado clube.
19. Consultar número de golos de um determinado jogador.
20. Consultar agente de um determinado jogador.
21. Obter contacto de um agente de um determinado jogador.
22. Dado um determinado olheiro, obter lista de reputações de jogadores observados por ele.
23. Obter o top 5 de jovens promessas mais novas.
24. Obter o top 5 de melhores jogadores sem clube.
25. Obter jogadores de uma posição abaixo de um determinado preço.
26. Dado um agente ele sabe qual dos seus jogadores está mais perto de acabar contrato.
27. Devolver o histórico de clubes e o clube atual para ver a reputação.
28. Total de golos do jogador.
29. Golos marcados numa determinada equipa anterior.
30. Histórico de clubes de um jogador.
31. Jogadores que um agente representa.

2.2.3 Requisitos de controlo

32. Os agentes têm permissões para consultar, inserir e gerir o perfil futebolístico de um respetivo jogador.

33. Os olheiros têm permissões para avaliar e inserir comentários sobre um respetivo jogador.

2.2.4 Análise geral dos requisitos

Após o levantamento e qualificação dos requisitos foi necessária uma análise em agregado dos mesmos.

Ao longo deste projeto houve várias discussões, tais como se devia existir uma entidade chamada Utilizador. Neste caso o utilizador seria um cidadão comum que se poderia registar na FP e observar os jogadores lá representados. Eventualmente, a equipa chegou à conclusão de que este conceito era desnecessário, pois não é necessário ter uma conta para fazer esta observação.

Esta fase serviu principalmente discutir as diferenças vistas das partes interessadas, assim como dos elementos da equipa, de forma a prevenir complicações futuras relativas a eventuais diferenças nas interpretações dos requisitos do sistema de armazenamento de dados a desenvolver.

3 Modelação Conceptual

3.1 Apresentação da abordagem de modelação realizada

Após a recolha e análise dos requisitos do sistema, decidimos planear o *design* da base de dados. Para a realização deste projeto foi utilizado Diagrama ER. O Diagrama ER é um modelo não muito técnico e fácil de ser compreendido em que podemos comunicar com o utilizador sem ambiguidades e assim ir ao encontro das necessidades dos seus utilizadores finais. O modelo conceptual tem como principal objetivo representar as entidades e os atributos necessários para a realização da base de dados do FuteProject.

Ao longo da construção deste modelo conceptual é essencial a validação dos requisitos de forma a prevenir incoerências e assegurando que a futura base de dados seja eficiente na resolução de problemas.

3.2 Identificação e caracterização das entidades

De acordo com os requisitos levantados para a realização deste projeto identificamos as seguintes identidades para a satisfação deste problema:

- **Jogador**

Esta entidade vai representar cada jogador que está registado na FuteProject. Cada jogador tem associado os seguintes atributos: id, nome, data de nascimento, altura, número de internacionalizações, valor de mercado atual, reputação, golos marcados no clube atual, idade, tipo (agenciado ou promessa), posição (avançado, guarda-redes, defesa, etc), clube atual, histórico de clubes, localidade e país de nascença, nacionalidade, agente que o representa (pode não existir no caso do jogador promessa), olheiro que o descobriu (caso exista), data de início do contrato atual, data de fim do contrato atual, avaliações ou observações que podem ser deixadas pelo olheiro que o descobriu.

- **Olheiro**

Esta entidade vai representar cada funcionário que tem permissão para avaliar os jogadores. Cada olheiro tem associado os seguintes atributos: id, nome, data de nascimento, reputação, telefone, idade e nacionalidade.

- **Agente**

Esta entidade vai representar cada funcionário que tem permissão para inserir, editar ou remover jogadores na base de dados. Cada agente tem associado os seguintes atributos: id, nome, reputação, telefone, idade, data de nascimento, email, nacionalidade.

- **Localidade**

Esta entidade vai representar a localidade em que os jogadores nasceram. Tem associado a sua designação.

- **País**

Esta entidade vai representar o país em que os jogadores nasceram. Tem associado a sua designação.

- **Posição**

Esta entidade vai representar a posição que os jogadores vão ocupar em campo. Tem associado a designação da posição

.

- **Tipo**

Esta entidade vai representar a classe em que o jogador se encontra, ou seja, vai indicar se o jogador é uma “jovem promessa” ou “contratado”. Cada tipo tem associado os seguintes atributos: designação.

- **Clube**

Esta entidade vai representar o clube em que o jogador atualmente se encontra. Cada clube tem associado a sua designação.

- **Nacionalidade**

Esta entidade vai representar a nacionalidade do jogador, agente ou olheiro. Tem associado a sua designação.

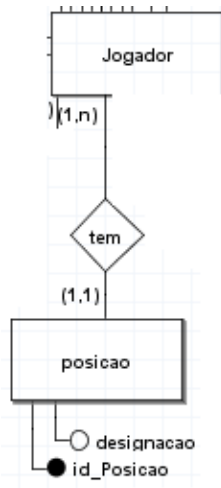
- **Clube Anterior**

Esta entidade vai representar cada clube em que o jogador já jogou. Mais uma vez, apenas tem associado a sua designação.

3.3 Identificação e caracterização dos relacionamentos

Relação entre:

1. Jogador-Posição:



Relacionamento: Jogador tem uma posição.

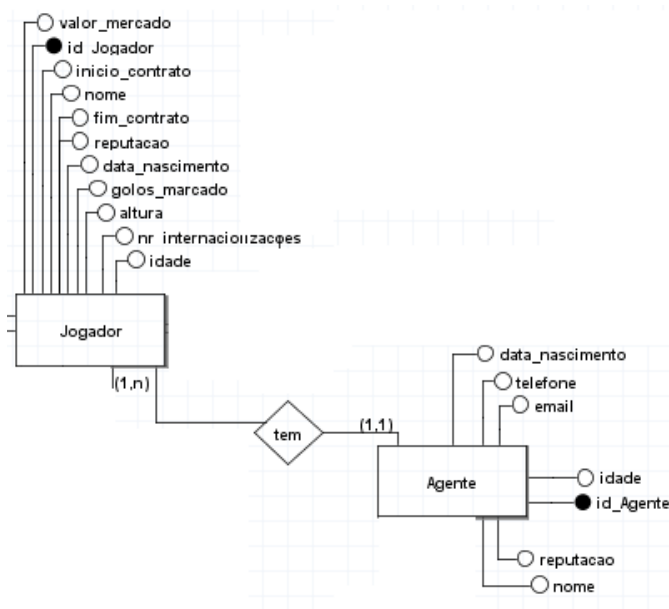
Descrição: Um jogador tem uma posição, seja avançado, defesa, guarda redes, etc. Este relacionamento traduz esse comportamento.

Cardinalidade: Jogador (1, n) - Posição (1,1)

Um jogador só tem uma posição, no entanto uma posição pode ser ocupada por vários jogadores.

Figura 1 - Relacionamento Jogador - Posição

2. Jogador-Agente:



Relacionamento: Jogador tem agente.

Descrição: Um agente representa jogadores.

Cardinalidade: Jogador (1, n) – Agente (1,1)

Um jogador tem apenas um agente, mas um agente pode ter vários jogadores.

Figura 2 - Relacionamento Jogador - Agente

3. Jogador-Olheiro

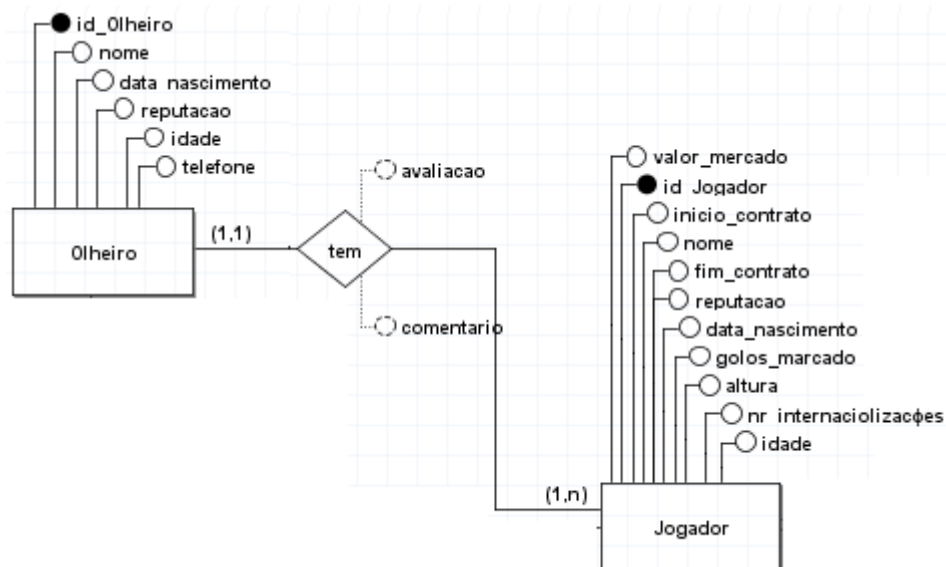


Figura 3 - Relacionamento Jogador - Olheiro

Relacionamento: Jogador tem um Olheiro

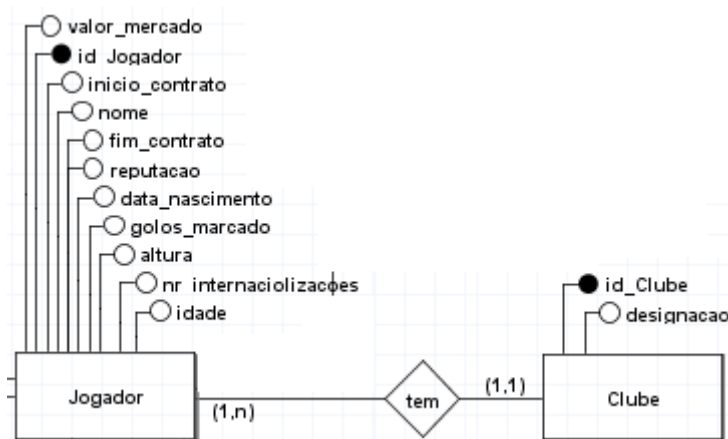
Descrição: Um jogador para ser contratado para a agência pode ter de ser visto e aceite por um olheiro e por isso decidimos que será guardado o olheiro que o descobriu.

Cardinalidade: Jogador (1, n) – Olheiro (1,1).

Um jogador foi descoberto apenas por um olheiro, mas um olheiro pode ter descoberto vários jogadores.

Atributos: Os atributos que este relacionamento possui são os seguintes: avaliação e comentário opcional que o olheiro que descobriu o jogador pode inserir no seu perfil.

4. Jogador-Clube:



Relacionamento: Jogador tem um clube.

Descrição: Clube atual em que o jogador se encontra.

Cardinalidade: Jogador (1, n) – Clube (1,1)

Um jogador pertence apenas a um clube, mas um clube tem vários jogadores.

Figura 4 - Relacionamento Jogador - Clube

5. Jogador-Tipo:

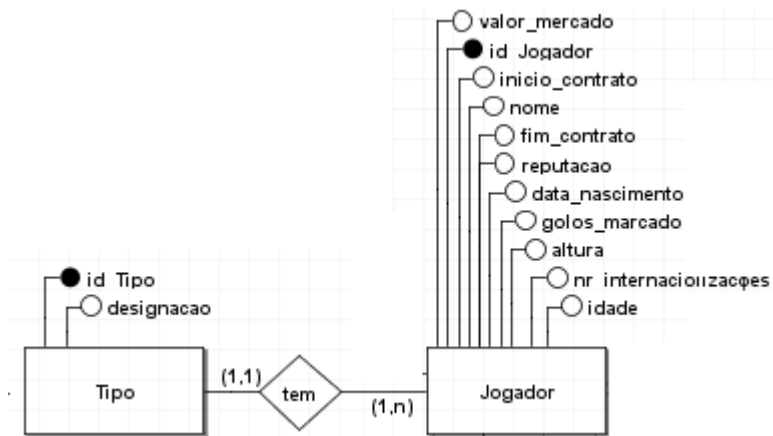


Figura 5 - Relacionamento Jogador - Tipo

Cardinalidade: Jogador (1, n) – Tipo (1,1).

Um jogador tem um tipo associado, mas um tipo pode ser associado a vários jogadores

6. Jogador-Localidade:

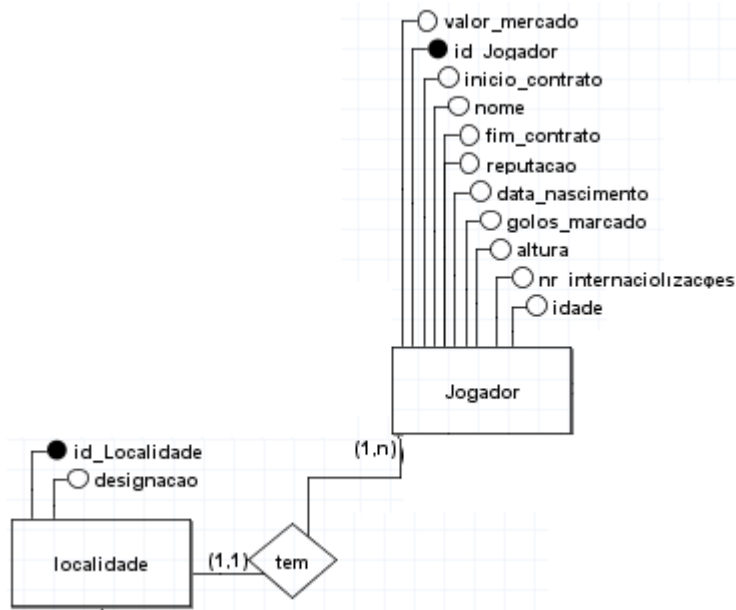


Figura 6 - Relacionamento Jogador - Localidade

Relacionamento: Jogador tem um tipo.

Descrição: Na nossa empresa existem jogadores que já estão contratados e outros que ainda estão em fase de análise para isso é necessário guardar a informação acerca do tipo de jogador que são.

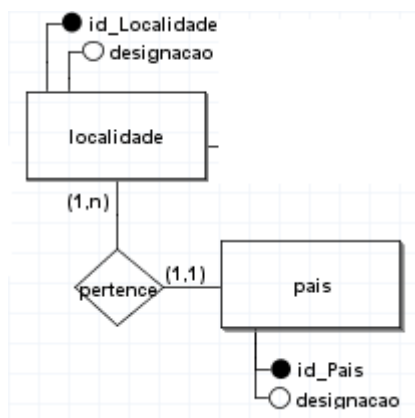
Relacionamento: Jogador tem uma localidade.

Descrição: Um jogador tem um local de nascimento que deve ser guardado, para isso usamos a localidade.

Cardinalidade: Jogador(1,n) – Localidade(1,1).

Um jogador tem apenas uma localidade, mas uma localidade pode ter vários jogadores.

7. Localidade-País:



Relacionamento: Localidade pertence a um país.

Cardinalidade: Localidade (1, n) – país (1,1)

Uma localidade pertence apenas a um país, mas um país tem várias localidades.

Figura 7 - Relacionamento Localidade - País

8. Jogador-clube_anterior:

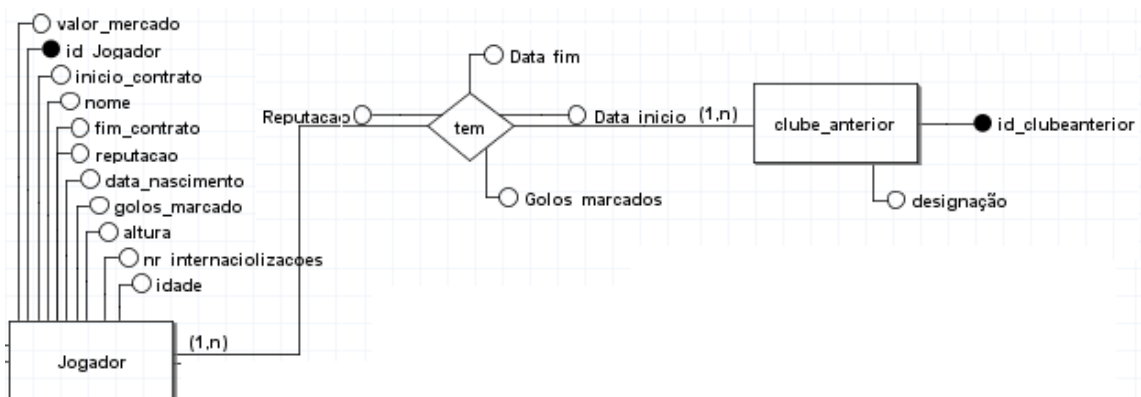


Figura 8 - Relacionamento Jogador - Clube_anterior

Relacionamento: Jogador tem clubes anteriores.

Descrição: Para ser representado o histórico de clubes, criamos este relacionamento que traduz a ligação do jogador com um dos seus clubes anteriores.

Cardinalidade: Jogador (1, n) - clube_anterior (1, n)

Um jogador pode ter vários clubes anteriores e um clube pode ter vários jogadores.

Atributos: Os atributos que este relacionamento possui são os seguintes: Data de início do contrato com esse clube anterior, Data de fim de contrato do mesmo, Golos marcados nesse clube e Reputação no final do contrato.

9. Jogador-Nacionalidade:

Relacionamento: Jogador tem nacionalidade.

Cardinalidade: Jogador (1, n) – Nacionalidade (1, 1)

Um jogador tem apenas uma nacionalidade, mas vários jogadores têm uma nacionalidade.

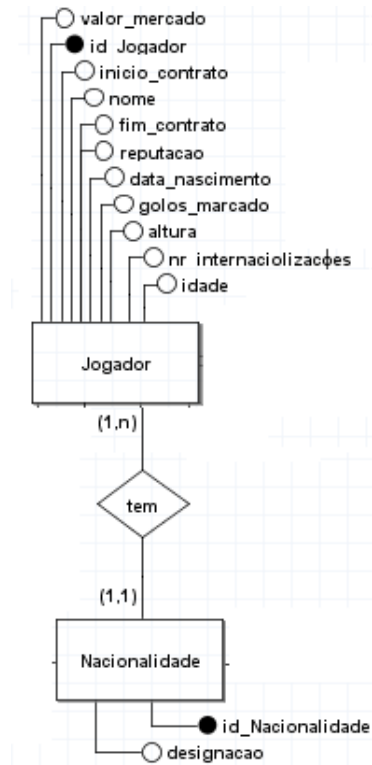


Figura 9 - Relacionamento Jogador - Nacionalidade

10. Olheiro-Nacionalidade:

Relacionamento: Olheiro tem nacionalidade

Cardinalidade: Olheiro (1, n) - Nacionalidade (1, 1)

Um olheiro tem apenas uma nacionalidade, mas, vários olheiros têm a mesma nacionalidade.

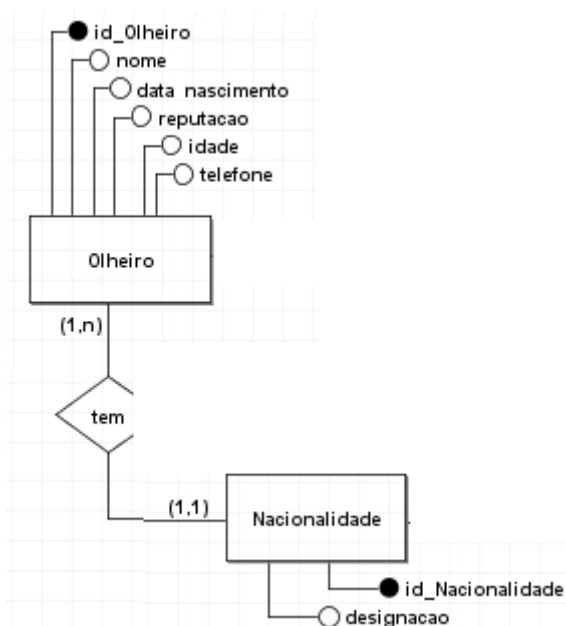


Figura 10 - Relacionamento Olheiro - Nacionalidade

11. Agente-Nacionalidade:

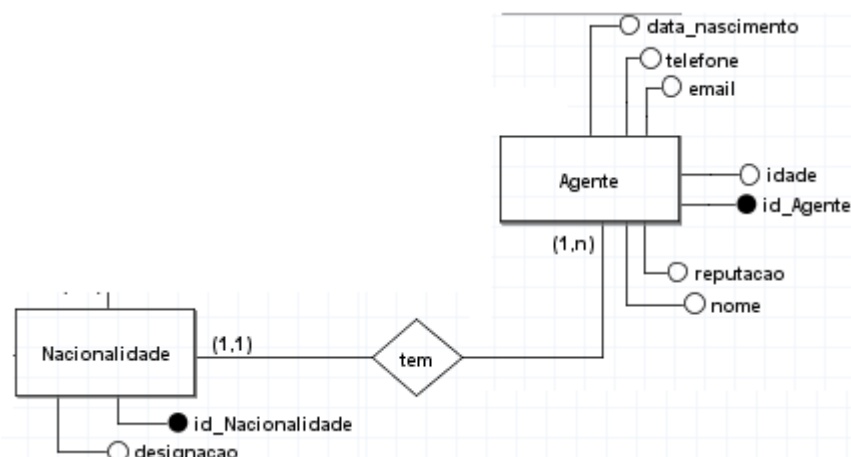


Figura 11 - Relacionamento Agente - Nacionalidade

Relacionamento: Agente tem nacionalidade

Cardinalidade: Agente (1, n) – Nacionalidade (1, 1)

Um Agente tem apenas uma nacionalidade, mas vários agentes têm a mesma nacionalidade.

3.4 Identificação e Caracterização da Associação dos Atributos com as Entidades e Relacionamentos

Entidade	Atributos	Descrição	Tipo de Atributo	Tipo de dados e tamanho
Olheiro	id_Olheiro	Identificação do Olheiro	Chave primária	INT
	nome	Nome do olheiro	Simples	VARCHAR(45)
	data_nascimento	Data de nascimento do olheiro	Simples	DATE
	reputação	Reputação do olheiro	Simples	INT
	idade	Idade do Olheiro	Derivado	INT
	telefone	Número de telefone do olheiro	Simples	VARCHAR(45)
Clube	id_Clube	Identificação do clube	Chave primária	INT
	designacao	Designação do clube	Simples	VARCHAR(45)

Jogador	id_Jogador	Identificação do jogador	Chave primaria	INT
	reputação	Reputação do jogador	Simples	INT
	data_nascimento	Data de nascimento do jogador	Simples	DATE
	golos_marcados	Total de golos marcados pelo jogador	Simples	INT
	altura	Altura do jogador em centímetros	Simples	INT
	Idade	Idade do jogador	Derivado	INT
	nr_internacionalizações	Número de vezes que jogou pela seleção	Simples	INT
	valor_mercado	Valor de mercado atual, em milhões	Simples	INT
	nome	Nome do Jogador	Simples	VARCHAR(45)
	inicio_contrato	Data de início do contrato atual	Simples	DATE
	fim_contrato	Data de fim do contrato atual	Simples	DATE
Agente	id_Agente	Identificação do agente	Chave primaria	INT
	data_nascimento	Data de Nascimento do agente	Simples	DATE
	nome	Nome do agente	Simples	VARCHAR(45)
	idade	Idade do agente	Derivado	INT
	reputação	Reputação de um agente	Simples	INT
	telefone	Número de telefone de um agente	Simples	VARCHAR(45)
	email	Email de um agente	Simples	VARCHAR(75)
Posição	id_Posicao	identificação de uma posição	Chave primaria	VARCHAR(45)
	designacao	Designação da posição	Simples	VARCHAR(45)
Localidade	id_Localidade	Identificação da localidade	Chave primaria	INT
	designação	Designação de uma localidade	Simples	VARCHAR(45)
País	id_Pais	Identificação de um país	Chave primaria	INT
	designacao	Designação de um país	Simples	VARCHAR(45)
Tipo	id_Tipo	Identificação de um tipo	Chave primaria	INT
	designacao	Designação de um país	Simples	VARCHAR(45)
Nacionalidade	id_Nacionalidade	Identifica a Nacionalidade	Chave primaria	INT
	designacao	Nome da Nacionalidade	Simples	VARCHAR(45)

clube_anterior	id_clubeanterior	Identificação do clube anterior	Chave primaria	INT
	designacao	Nome do clube	Simples	VARCHAR(45)
Jogador – Olheiro (Relacionamento)	avaliacao	Avaliação do jogador	Opcional	VARCHAR(45)
	comentario	Comentário feito pelo olheiro	Opcional	VARCHAR(45)
Jogador – Clube_anterior (Relacionamento)	Golos marcados	Golos marcados pelo jogador nesse clube	Simples	INT
	Reputacao	Reputação do jogador nesse clube	Simples	INT
	Data início	Data de início do contrato nesse clube	Simples	DATE
	Data fim	Data de fim do contrato nesse clube	Simples	DATE

Tabela 1 - Caracterização de todos os atributos existentes

A maioria dos atributos representado na Tabela 1 vão ser atributos simples (não derivados, não nulos e de valor único), com exceção da Idade que é um atributo derivado na medida em que é calculado a partir da data de nascimento.

3.4.1 Domínio dos Atributos

Todos os atributos possuem um tipo de dados e o respetivo domínio de valores que lhe podem ser atribuídos.

- **Olheiro**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Nome	VARCHAR(45)	Sequência de palavras
Idade	INT	Número inteiro positivo
Data de Nascimento	INT	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])
Reputação	INT	Número inteiro positivo menor que 10
Telefone	VARCHAR(45)	Sequência de números e/ou caracteres

Tabela 2 - Caracterização dos atributos do Olheiro

- **Jogador**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Nome	VARCHAR(45)	Sequência de palavras
Valor de mercado	INT	Número inteiro positivo
Reputação	INT	Número inteiro positivo
Altura	INT	Número inteiro positivo
Golos Marcados	INT	Número inteiro maior ou igual a 0
Nr Internacionalizações	INT	Número inteiro maior ou igual a 0
Idade	INT	Número inteiro positivo
Data de Nascimento	DATE	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])
Início Contrato	DATE	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])
Fim Contrato	DATE	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])

Tabela 3 - Caracterização dos atributos do Jogador

- **Agente**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Nome	VARCHAR(45)	Sequência de palavras
Email	VARCHAR(75)	Sequência de caracteres alfanuméricos, seguida do caracter “@” e por fim o domínio do provedor de serviço de email.
Telefone	VARCHAR(45)	Sequência de números e/ou caracteres
Reputação	INT	Número inteiro positivo menor que 10
Idade	INT	Número inteiro positivo
Data de Nascimento	DATE	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])

Tabela 4 - Caracterização dos atributos do Agente

- **Tipo**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Designação	VARCHAR(45)	Palavra ou sequência de palavras

Tabela 5 - Caracterização dos atributos do Tipo

- **Localidade**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Designação	VARCHAR(45)	Palavra ou sequência de palavras

Tabela 6 - Caracterização dos atributos da Localidade

- **País**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Designação	VARCHAR(45)	Palavra ou sequência de palavras

Tabela 7 - Caracterização dos atributos do País

- **Posição**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Designação	VARCHAR(45)	Palavra ou sequência de palavras

Tabela 8 - Caracterização dos atributos da Posição

- **Nacionalidade**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Designação	VARCHAR(45)	Palavra ou sequência de palavras

Tabela 9 - Caracterização dos atributos da Nacionalidade

- **Clube**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Designação	VARCHAR(45)	Palavra ou sequência de palavras

Tabela 10 - Caracterização dos atributos do Clube

- **Clube anterior**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo
Designação	VARCHAR(45)	Palavra ou sequência de palavras

Tabela 11 - Caracterização dos atributos do Clube anterior

- **Relacionamento Jogador-Olheiro**

Atributos	Tipo de Dados	Domínio de Valores
Avaliação	INT	Número inteiro positivo
Comentário	TEXT	Sequência de palavras

Tabela 12 - Caracterização de atributos do relacionamento Jogador-Olheiro

- **Relacionamento Jogador-Clube anterior**

Atributos	Tipo de Dados	Domínio de Valores
Golos marcados	INT	Número inteiro positivo
Reputação	INT	Número inteiro positivo
Data início	DATE	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])
Data fim	DATE	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])

Tabela 13 - Caracterização de atributos do relacionamento Jogador-Clube anterior

3.4.2 Determinação das chaves candidatas, chaves primárias e chaves alternadas

Nesta secção vamos identificar a chave primária para cada uma das entidades. Esta chave vai corresponder a um atributo que seja único (não existe hipótese de existir vários valores repetidos desse mesmo atributo) e que vai tornar possível a identificação da respetiva identidade.

Chaves candidatas são os atributos que possam servir como chave primária. Dessa seleção é escolhida uma primária e as restantes serão classificadas como chaves alternadas.

- **Olheiro:**

Chaves Candidatas: Id_Olheiro, telefone.

Chave Primária: Id_Olheiro.

Chaves Alternativas: telefone.

- **Jogador:**

Chaves Candidatas: Id_Jogador.

Chave Primária: Id_Jogador.

Chaves Alternativas: Nenhuma.

- **Clube:**

Chaves Candidatas: Id_Clube.

Chave Primária: Id_Clube.

Chaves Alternativas: Nenhuma.

- **Tipo:**

Chaves Candidatas: Id_Tipo.

Chave Primária: Id_Tipo.

Chaves Alternativas: Nenhuma.

- **Localidade:**

Chaves Candidatas: Id_Localidade.

Chave Primária: Id_Localidade.

Chaves Alternativas: Nenhuma.

- **Posicao:**

Chaves Candidatas: Id_Posicao.

Chave Primária: Id_Posicao.

Chaves Alternativas: Nenhuma.

- **Pais:**

Chaves Candidatas: Id_Pais.

Chave Primária: Id_Pais.

Chaves Alternativas: Nenhuma.

- **Agente:**

Chaves Candidatas: Id_Agente, telefone, email.

Chave Primária: Id_Agente.

Chaves Alternativas: telefone, email.

- **Nacionalidade:**

Chaves Candidatas: Id_Nacionalidade.

Chave Primária: Id_Nacionalidade.

Chaves Alternativas: Nenhuma.

- **Clube_anterior:**

Chaves Candidatas: Id_clubeanterior.

Chave Primária: Id_clubeanterior.

Chaves Alternativas: Nenhuma.

3.5 Detalhe ou generalização de entidades

No nosso projeto não foi utilizada generalização nem especialização de entidades no Modelo Conceptual.

3.6 Apresentação e explicação do diagrama ER

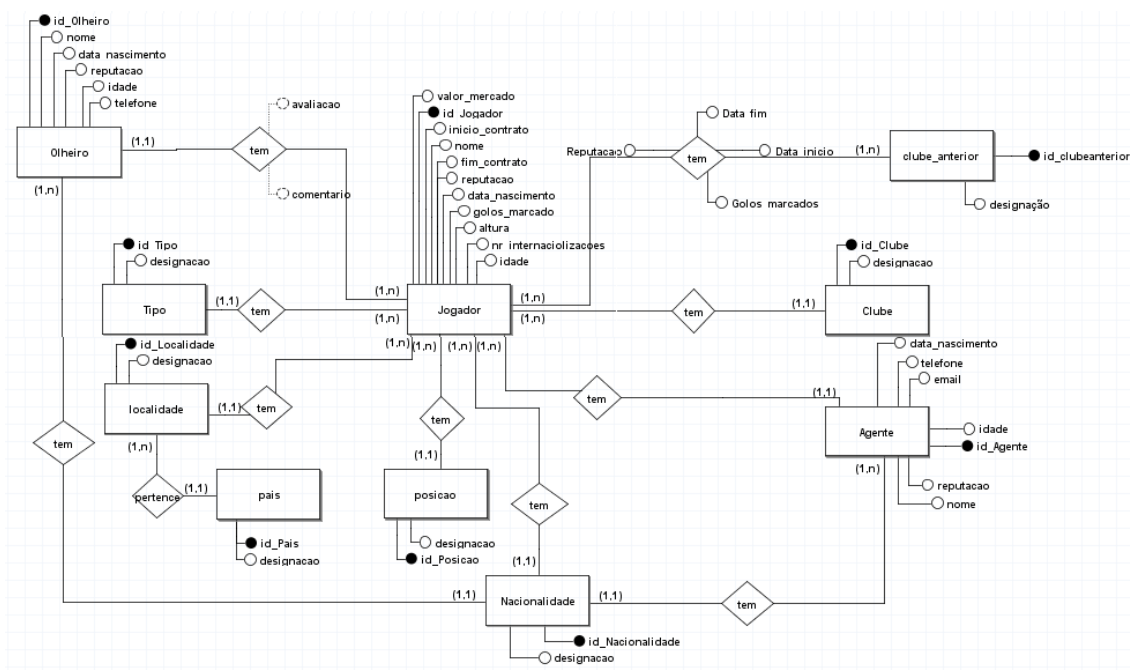


Figura 12 - Modelo Conceitual.

3.7 Validação do Modelo de Dados com o Utilizador

Terminado o Modelo de Dados, é essencial validar o modelo conceitual final. Para isso, verificámos se o nosso modelo de dados consegue responder aos requisitos impostos inicialmente com o utilizador:

1. Os agentes devem ser registados, sendo-lhes atribuído um ID. Deve ser fornecido um Nome, Nacionalidade, Data de Nascimento, Telemóvel, Reputação, Email e Idade.

A entidade agente tem todos os atributos referidos com exceção da nacionalidade. Em relação ao atributo nacionalidade, este é conseguido através da entidade Nacionalidade. Escolhemos separar o Nacionalidade como entidade individual com o objetivo de reduzir a redundância de dados, aumentar a integridade de dados e o desempenho, uma vez que várias entidades tinham necessidade de apresentar este atributo. Isto é, foi feita a normalização dos dados. (Figura 11 - Relacionamento Agente - Nacionalidade)

2. Os olheiros devem ser registados, sendo-lhes atribuído um ID. Deve ser fornecido um Nome, Nacionalidade, Data de Nascimento, Telemóvel, Reputação e Idade.

A entidade olheiro tem todos os atributos referidos com exceção da nacionalidade a qual é acedida através na entidade Nacionalidade. (Figura 10 - Relacionamento Olheiro - Nacionalidade)

3. Os jogadores devem ser registados, sendo-lhes atribuído um ID. Deve ser fornecido um Nome, Nacionalidade, Data de Nascimento, Localidade e País de nascença, Data de início e fim de contrato, Agente que o representa, Olheiro que o tenha descoberto e perfil futebolístico. Além disso deve ser possível verificar o seu histórico de clubes.

Na medida em que se trata da entidade principal do nosso projeto, este recorre a várias entidades para aceder às diferentes informações. O jogador tem como atributos: nome, valor de mercado, reputação, data de nascimento, golos marcados, altura, número de internacionalizações, idade, data de início de contrato, e data de fim. Além disso, as outras informações requeridas são acedidas através das entidades normalizadas como: o Clube, Agente, Posição, Tipo (jogador promessa ou contratado), Clube anterior, Nacionalidade, Localidade e País. Sendo assim, o Jogador cumpre todos os requisitos esperados.

4. Os jogadores podem-se representar em dois grupos: jovens promessas e jogadores agenciados.

Escolhemos separar o Tipo de jogador como uma identidade individual com vista à normalização o que irá facilitar a introdução de dados e evitar que o tipo de jogador seja escrito de formas diferentes para o mesmo tipo. (Figura 5 - Relacionamento Jogador - Tipo)

5. Uma mudança de clube apenas pode ser partilhada quando o agente a validar.

Os agentes da FuteProject, uma vez que têm acesso total à base de dados, podem criar/editar informações de jogadores e associá-los aos respetivos clubes.

6. Quando um jogador muda de clube, o histórico de clubes deste deve ser atualizado.

Da mesma forma, como os agentes da FuteProject têm acesso total à base de dados, isto permite-lhes atualizar a informação do histórico de clubes dos jogadores.

7. Olheiros podem comentar e/ou avaliar os jogadores.

O olheiro poderá deixar uma avaliação ou comentário a cada jogador que descobriu, que será armazenada no atributo opcional avaliação e comentário, respetivamente, do relacionamento representado na Figura 3 - Relacionamento Jogador - Olheiro.

8. No final de cada época as estatísticas dos jogadores são atualizadas.

Mais uma vez, cabe aos agentes atualizar as informações dos jogadores que representam.

Sendo assim, podemos verificar que todos os requisitos de descrição são conceptualmente possíveis. Quanto aos requisitos de exploração, uma vez que todos os atributos criados foram de encontro com as necessidades expostas, todas consultas e verificações presentes nos requisitos de exploração podem ser satisfeitas. Posto isto, concluímos que o modelo de dados está validado com o utilizador.

4 Modelação Lógica

Neste capítulo será apresentado o modelo lógico. Este tem como principal objetivo ajudar na derivação dos relacionamentos sendo este essencial para o desenvolvimento da nossa solução de gestão de base de dados.

4.1 Construção e validação dos modelos dados lógicos

Depois de definidos as entidades e os tributos necessários para a realização deste projeto seguem-se a derivação e criação dos relacionamentos entre as entidades e os seus atributos de modo a concluir o Modelo de dados lógicos.

4.1.1 Relacionamento binário (1: N)

Para este relacionamento, a chave primaria do lado 1 tem de ser usada como atributo na entidade correspondente à entidade do lado N. Na entidade do lado N o atributo correspondente à chave primária do lado 1 designa-se como chave estrangeira.

- Neste caso, a chave estrangeira do Jogador é a chave primária do Olheiro, sabendo-se assim que um certo jogador tem um olheiro associado.

Olheiro (id_Olheiro, nome, data_nascimento, reputacao, telefone, idade)

Chave Primária: id_Olheiro

Jogador (id_Jogador, valor_mercado, inicio_contrato, fim_contrato, nome, reputacao, data_nascimento, golos_marcados, altura, nr_internacionalizacoes, idade)

Chave Primária: id_Jogador

Chave Estrangeira: id_Olheiro

- Neste caso, a chave estrangeira do Jogador é a chave primária do Agente, sabendo-se assim que um certo jogador tem um Agente que o representa.

Agente (id_Agente, data_nascimento, nome, reputação, telefone, email, idade)

Chave Primária: id_Agente

Jogador (id_Jogador, valor_mercado, inicio_contrato, fim_contrato, nome, reputacao, data_nascimento, golos_marcados, altura, nr_internacionalizacoes, idade).

Chave Primária: id_Jogador

Chave Estrangeira: id_Agente

- Neste caso, a chave estrangeira do Jogador é a chave primária do Tipo, sabendo-se assim que um certo jogador tem um certo tipo (Jovem promessa ou Jogador agenciado).

Tipo: (id_Tipo, designacao)

Chave Primária: id_Tipo

Jogador (id_Jogador, valor_mercado, inicio_contrato, fim_contrato, nome, reputacao, data_nascimento, golos_marcados, altura, nr_internacionalizacoes, idade)

Chave Primária: id_Jogador

Chave Estrangeira: id_Tipo

- Neste caso, a chave estrangeira do Jogador é a chave primária da Posicao, sabendo-se assim que um certo jogador faz uma certa posição.

Posicao: (id_Posicao, designacao)

Chave Primária: id_Posicao

Jogador (id_Jogador, valor_mercado, inicio_contrato, fim_contrato, nome, reputacao, data_nascimento, golos_marcados, altura, nr_internacionalizacoes, idade)

Chave Primária: id_Jogador

Chave Estrangeira: id_Posicao

- Neste caso, a chave estrangeira do Jogador é a chave primária do Clube, sabendo-se assim que um certo jogador joga num determinado clube.

Clube: (id_Clube, designacao)

Chave Primária: id_Clube

Jogador (id_Jogador, valor_mercado, inicio_contrato, fim_contrato, nome, reputacao, data_nascimento, golos_marcados, altura, nr_internacionalizacoes, idade)

Chave Primária: id_Jogador

Chave Estrangeira: id_Clube

- Neste caso, a chave estrangeira do Jogador é a chave primária da localidade, sabendo-se assim que um certo jogador tem uma localidade onde nasceu.

Localidade: (id_Localidade, designacao)

Chave Primária: id_Localidade

Jogador (id_Jogador, valor_mercado, inicio_contrato, fim_contrato, nome, reputacao, data_nascimento, golos_marcados, altura, nr_internacionalizacoes, idade)

Chave Primária: id_Jogador

Chave Estrangeira: id_Localidade

- Neste caso, a chave estrangeira do Jogador é a chave primária da Nacionalidade, sabendo-se assim que um certo jogador tem uma certa nacionalidade.

Nacionalidade: (id_Nacionalidade, designacao)

Chave Primária: id_Nacionalidade

Jogador (id_Jogador, valor_mercado, inicio_contrato, fim_contrato, nome, reputacao, data_nascimento, golos_marcados, altura, nr_internacionalizacoes, idade)

Chave Primária: id_Jogador

Chave Estrangeira: id_Nacionalidade

- Neste caso, a chave estrangeira da Localidade é a chave primária do Pais, sabendo-se assim que uma certa Localidade pertence a um país.

Pais: (id_Pais, designacao)

Chave Primária: id_Pais

Localidade: (id_Localidade, designacao)

Chave Primária: id_Localidade

Chave Estrangeira: id_Pais

- Neste caso, a chave estrangeira do Agente é a chave primária da Nacionalidade, sabendo-se assim que um certo agente tem uma certa nacionalidade.

Nacionalidade: (id_Nacionalidade, designacao)

Chave Primária: id_Nacionalidade

Agente (id_Agente, data_nascimento, nome, reputação, telefone, email, idade)

Chave Primária: id_Agente

Chave Estrangeira: id_Nacionalidade

- Neste caso, a chave estrangeira do Olheiro é a chave primária da Nacionalidade, sabendo-se assim que um certo olheiro tem uma certa nacionalidade.

Nacionalidade: (id_Nacionalidade, designacao)

Chave Primária: id_Nacionalidade

Olheiro (id_Olheiro, nome, data_nascimento, reputacao, telefone, idade)

Chave Primária: id_Olheiro

Chave Estrangeira: id_Nacionalidade

4.1.2 Relacionamentos de muitos para muitos (N:M)

Neste tipo de relacionamento, é gerado um novo relacionamento, em que existem duas chaves primárias, sendo elas cada uma das chaves primárias das entidades que formaram este relacionamento.

Histórico_clubes (reputacao, golos_marcados, data_inicio, data_fim)

Chave primária: id_Jogador, id_clubeanterior

4.1.3 Entidades resultantes

- **Olheiro** = {Id_Olheiro, Nome, Data_nascimento, Reputacao, Telefone, Idade, Id_Nacionalidade}
- **Agente** = {Id_Agente, Data_nascimento, Nome, Reputacao, Telefone, Email, Idade, Id_Nacionalidade}
- **Tipo** = {Id_Tipo, Designacao}
- **Localidade** = {Id_Localidade, Designacao, Id_Pais}
- **Pais** = {Id_Pais, Designacao}
- **Nacionalidade** = {Id_Nacionalidade, Designacao}
- **Posicao** = {Id_Posicao, Designacao}
- **Clube_anterior** = {Id_Clubeanterior, Designacao}
- **Historico_Clubes** = {Id_Jogador, Id_Clubeanterior}
- **Clube** = {Id_Clube, Designacao}
- **Jogador** = {Id_Jogador, Nome, Data_nascimento, Altura, Nr_internacionalizacoes, Valor_mercado, Reputacao, Golos_marcados, Idade, Inicio_contrato, Fim_contrato, Id_tipo, Id_Posicao, Id_Clube, Id_Localidade, Id_Agente, Id_Olheiro, Id_Nacionalidade, Avaliacao, Observacoes}

4.2 Desenho do Modelo Lógico

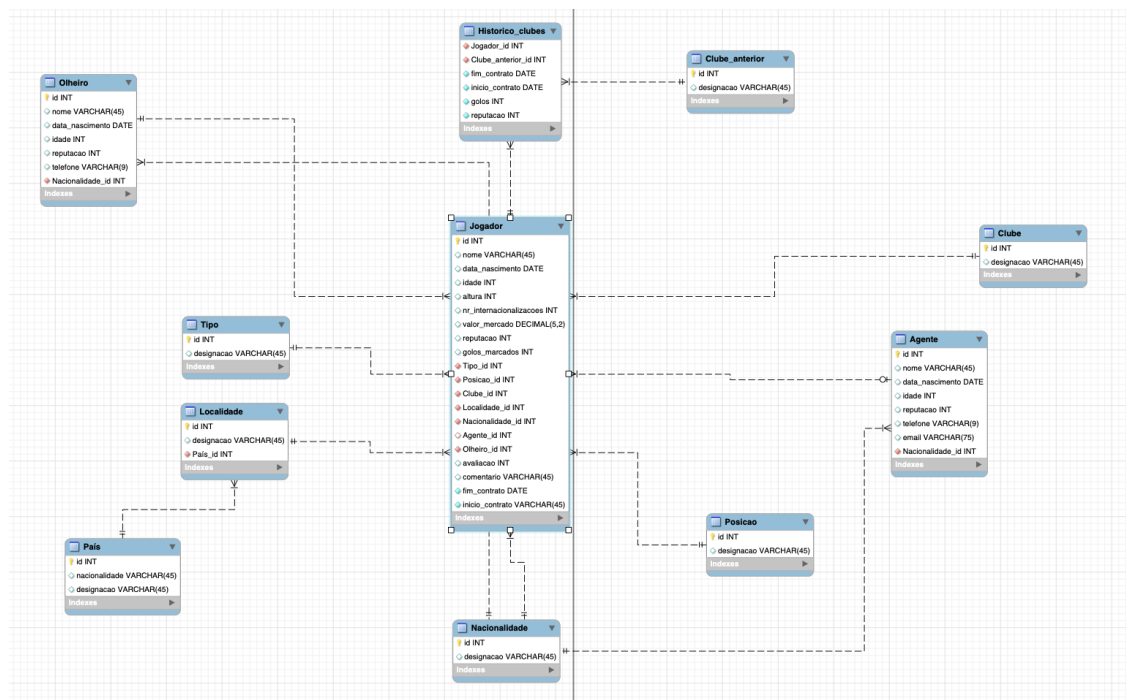


Figura 13 - - Modelo Lógico

4.3 Validação do Modelo através da Normalização

O modelo cumpre as formas normais a seguir:

- **Primeira Forma Normal.** Nesta forma os atributos precisam ser atômicos, o que significa que as tabelas não podem ter valores repetidos e nem atributos com mais de um valor. Um jogador pode ter mais do que um clube anterior, sendo assim o atributo multivalorado. Para normalizar, criamos a tabela Historico_clubes que faz a relação entre as tabelas Jogador e Clube_anterior.
- **Segunda Forma Normal.** Define que os atributos normais, ou seja, os não chave, devem depender unicamente da chave primária da tabela. Assim como as colunas da tabela que não são dependentes dessa chave devem ser removidas da tabela principal e cria-se uma tabela utilizando esses dados. Um exemplo do uso desta regra é na tabela Clube, se a designação do clube fosse um atributo do Jogador, não seria dependente na chave primária Id_Jogador mas sim da chave Id_Clube, daí a criação da tabela Clube.

- **Terceira Forma Normal.** Define que todos os atributos de uma tabela devem ser funcionalmente independentes uns dos outros, ao mesmo tempo que devem ser dependentes exclusivamente da chave primária da tabela. Um exemplo deste tipo de normalização foi a remoção do atributo Cláusula de Rescisão do jogador uma vez que este atributo iria ser calculado a partir do valor de mercado. É de notar que, apesar do atributo idade não obedecer a esta forma normal na medida em que é calculado a partir da data de nascimento, entendemos ser um atributo essencial.

4.4 Validação do Modelo com as Interrogações do Utilizador

Para verificar se o modelo lógico desenvolvido se encontra válido com base em algumas interrogações do utilizador criamos algumas interrogações relevantes adaptadas dos requisitos de exploração da base de dados e demonstramos de que forma o nosso modelo lógico consegue cumpri-las utilizando álgebra relacional.

1. Consultar o nome do Agente de um certo Jogador

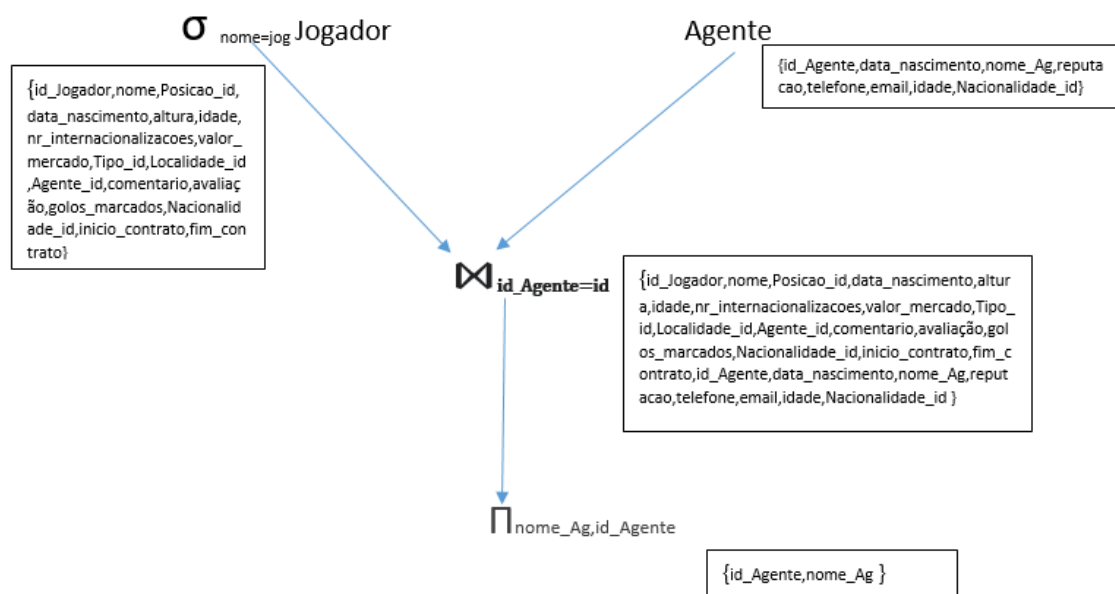


Figura 14 - - Árvore representativa da Interrogação nº 1.

Em primeiro lugar, são selecionados todos os tuplos de Jogador cujo nome é igual ao nome do jogador pretendido. Depois é feita a junção da relação resultante com a relação Agente em que id_Agente corresponde a Id. Por fim, projetar o componente nome da relação resultante.

2. Consultar jogadores com uma determinada nacionalidade

Em primeiro lugar, são selecionados todos os tuplos de Nacionalidade cujo nome é igual ao nome da nacionalidade pretendida. Depois é feita a junção da relação resultante com a relação Jogador em que id_Nacionalidade corresponde a Id. Por fim, projetar o componente nome da relação resultante.

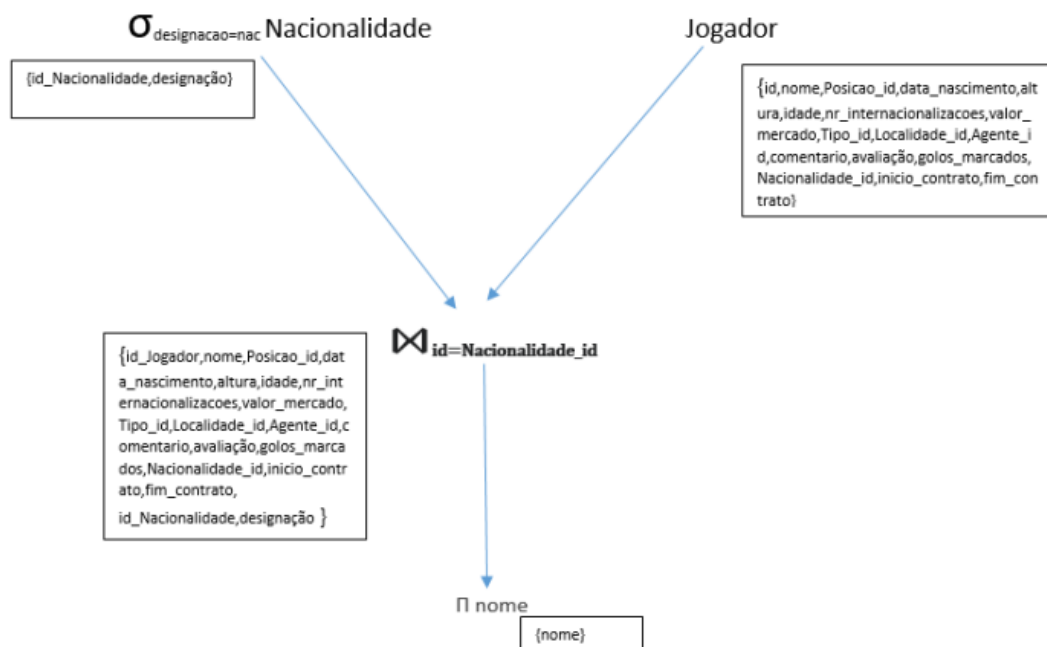


Figura 15 - Árvore representativa da Interrogação nº2

3. Obter jogadores de uma posição abaixo de um determinado preço

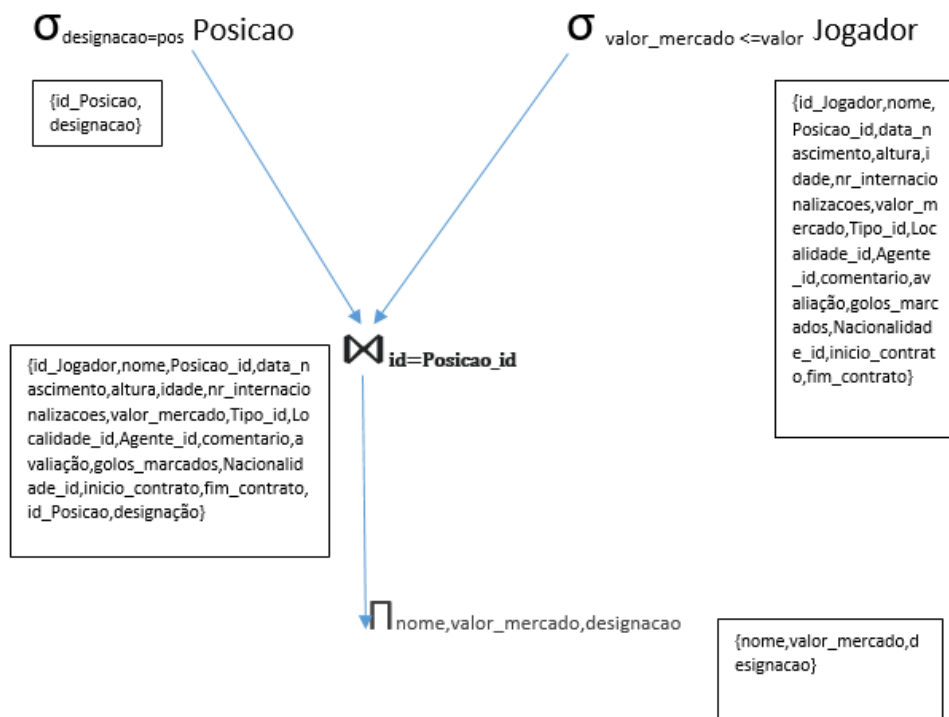


Figura 16 - Árvore representativa da Interrogação nº3

Em primeiro lugar, são seleccionados todos os tuplos de Posição cujo posição é igual ao nome da posição pretendida. Depois são seleccionados todos os tuplos de Jogador cujo valor de mercado é inferior ao valor pretendido. A seguir, é feita a junção das relações resultantes em que $id_Posicao$ corresponde a id . Por fim, projetar os componentes $nome$, $valor$ de $mercado$ e $designacao$ da relação resultante.

4.5 Revisão do Modelo Lógico com o Utilizador

Tendo concluído o modelo lógico, voltamos a reunir com o presidente da FuteProject. O objetivo desta reunião foi explicar e analisar este novo modelo e verificar que estavam presentes todas as entidades e atributos necessários de forma a armazenar toda a informação pretendida. Foram analisados cenários como prova de verificação que o modelo era capaz de responder a todas as situações colocadas de forma simples e eficiente.

Com isto, recebemos a autorização do presidente para avançar para a próxima fase do projeto, o Modelo Físico.

5 Implementação Física

5.1 Seleção do Sistema

O Sistema de Gestão de Base de Dados Relacional que nós decidimos escolher para implementar a base de dados concebida para o nosso projeto foi o “MySQL”. Decidimos escolher esta ferramenta pois, para além de ser gratuita, é fácil e simples de utilizar e apresenta um ótimo desempenho cumprindo assim todas as nossas necessidades para este trabalho.

5.2 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL

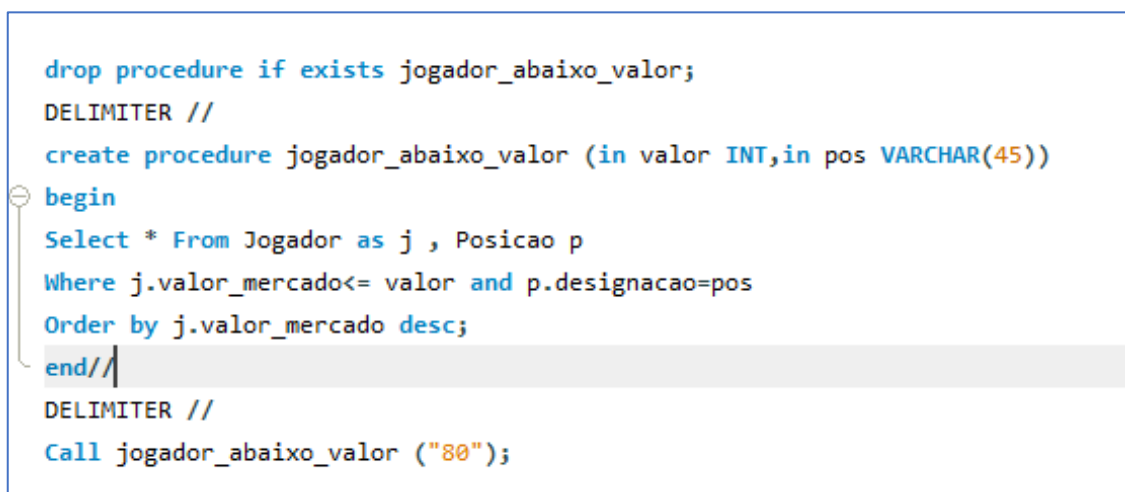
Esta etapa vai-se centrar na elaboração do Modelo Físico. Com o Modelo Lógico terminado decidimos prosseguir para a elaboração do Modelo Físico, para tal recorremos à ferramenta *Forward Engineer* presente no MySQL Workbench. Esta ferramenta vai modelar o Modelo Físico através do Modelo Lógico facilitando o processo de sustentar e processar as relações base e restrições anteriormente definidas.

5.3 Tradução das interrogações do utilizador para SQL

Nesta secção vamos apresentar alguns dos requisitos de exploração, sendo estas interrogações dos utilizadores.

Alguns exemplos de interrogações implementadas:

1. Obter os Jogadores de uma posição abaixo de um determinado valor. (RE 26)



```
drop procedure if exists jogador_abaixo_valor;
DELIMITER //
create procedure jogador_abaixo_valor (in valor INT,in pos VARCHAR(45))
begin
  select * from Jogador as j , Posicao p
  where j.valor_mercado<= valor and p.designacao=pos
  order by j.valor_mercado desc;
end//
DELIMITER //
call jogador_abaixo_valor ("80");
```

Figura 17- - Imagem do código obtido para a *query* nº 26

2. Devolver o histórico de clubes e o atual em comparar para a evolução do Jogador. (RE 28)

```
Drop procedure if exists evolucao_reputacao;
DELIMITER //
Create procedure evolucao_reputacao (in jog VARCHAR(45))
Begin
Select h.reputacao, nome, j.id, h.inicio_contrato, h.fim_contrato, c.designacao From historico_clubes h
INNER JOIN Jogador j on h.Jogador_id = j.id INNER JOIN Clube_anterior c on h.Clube_anterior_id = c.id
Where j.nome=jog
UNION
Select j.reputacao, j.nome, j.id, j.inicio_contrato, j.fim_contrato, cl.designacao From Clube cl
INNER JOIN Jogador j on j.Clube_id = cl.id
Where j.nome=jog
ORDER BY inicio_contrato;
END //
DELIMITER //
call evolucao_reputacao("Moussa Marega");
```

Figura 18- Imagem do código obtido para a *query* nº 28

3. Obter Jogadores de uma determinada nacionalidade. (RE 12)

```
drop procedure if exists jogadores_nacionalidade;
DELIMITER //
create procedure jogadores_nacionalidade(in nac VARCHAR(45))
begin
SELECT * From Nacionalidade n INNER JOIN Jogador j on n.id=j.Nacionalidade_id
where n.designacao = nac;
end //
DELIMITER //
call jogadores_nacionalidade("Português");
```

Figura 19- Imagem do código obtido para a *query* nº 12

4. Consultar um agente de um determinado Jogador. (RE 21)

```
drop procedure if exists agente_jogador;
DELIMITER //
create procedure agente_jogador (in jog VARCHAR(45))
begin
Select j.nome, a.nome , a.id From Jogador j INNER JOIN Agente a on a.id = j.Agente_id
where j.nome = jog;
end//
DELIMITER //
Call agente_jogador("Cristiano Ronaldo");
```

Figura 20- Imagem do código obtido para a *query* nº 21

5.4 Escolha, definição e caracterização de índices em SQL

A ferramenta *Foward Enginneering* cria alguns índices por predefinição, como por exemplo para as chaves primárias do projeto. No entanto, decidimos criar mais dois índices de forma a melhorar o tempo de execução das queries. Esses dois índices são:

- Índice 1:

```
Create index idx_nome_jogador on Jogador(nome);
```

Figura 21 - Índice idx_nome_jogador

Em primeiro lugar, como podemos verificar na Figura 26 - Índice idx_nome_jogador, criamos um índice (idx_nome_Jogador) para o atributo “nome” presente na entidade Jogador. Ao verificarmos que grande parte das procuras por um respetivo jogador são feitas pelo nome e não pelo seu ID. Ao criar este índice vai permitir uma maior eficiência na procura da tabela Jogador.

- Índice 2:

```
Create index idx_clube_antigo on Clube_anterior(designacao);
```

Figura 22 - Índice idx_clube_antigo

Em segundo lugar, como podemos verificar na Figura 27 - Índice idx_clube_antigo, criamos um índice (idx_clube_antigo) para o atributo “designação” presente na entidade Clube_anterior. Ao verificarmos que grande parte das procuras por um respetivo clube anterior de um respetivo jogador são feitas a partir pela sua designação e não pelo seu nome decidimos criar este índice. Ao criar este índice vai permitir uma maior eficiência na procura a tabela Clube_anterior.

5.5 Estimativa do espaço em disco da base de dados e taxa de crescimento anual

Nesta secção vamos indicar o espaço ocupado por cada entidade no disco da base de dados. É importante ter em atenção o espaço que a nossa base de dados vai ocupar e diminuindo, sempre que não corrompa o trabalho, para evitar desperdiçar espaço ocupado.

Para o cálculo das estimativas do espaço baseamo-nos no capítulo 11.8 do “MySQL” e 8.0 do *Reference Manual*. Cada valor calculado corresponde à soma dos número de bytes necessários para cada atributo.

Tipos de dados	Tamanho(bytes)
INT	4
DATE	3
VARCHAR(N)	N+1
TEXT	N

Tabela 14 - Espaço ocupado no disco por cada tipo de dados

Clube_anterior	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	designacao	VARCHAR(45)	46
Total			50

Tabela 15- Espaço ocupado no disco por cada entrada na tabela Clube_anterior

	Atributo	Tipos de Dados	Espaço no disco
Jogador	Id	INT	4
	nome	VARCHAR(45)	46
	data_nascimento	DATE	3
	altura	INT	4
	nr_internacionalizacoes	INT	4
	valor_mercado	INT	4
	idade	INT	4
	inicio_contrato	DATE	3
	fim_contrato	DATE	3
	id_Tipo	INT	4
	id_Posicao	INT	4
	id_Clube	INT	4
	id_Localidade	INT	4
	id_Agente	INT	4
	id_Olheiro	INT	4
	id_Nacionalidade	INT	4
	avaliacao	INT	4
	comentario	TEXT	-
	Total		107

Tabela 16 - Espaço ocupado por cada entrada na tabela Jogador

	Atributo	Tipos de Dados	Espaço no disco
Agente	id	INT	4
	nome	VARCHAR(45)	46
	data_nascimento	DATE	3
	reputacao	INT	4
	telefone	VARCHAR(45)	46
	idade	INT	4
	id_nacionalidade	INT	4
	email	VARCHAR(75)	76
	Total		187

Tabela 17 - Espaço ocupado por cada entrada na tabela Agente

Olheiro	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	nome	VARCHAR(45)	46
	data_nascimento	DATE	3
	reputacao	INT	4
	telefone	VARCHAR(45)	46
	idade	INT	4
	id_nacionalidade	INT	4
Total			111

Tabela 18 - Espaço ocupado por cada entrada na tabela Olheiro

Nacionalidade	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	designacao	VARCHAR(45)	46
Total			50

Tabela 19 - Espaço ocupado por cada entrada na tabela Nacionalidade

Pais	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	designacao	VARCHAR(45)	46
Total			50

Tabela 20 - Espaço ocupado por cada entrada na tabela Pais

Clube	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	designacao	VARCHAR(45)	46
Total			50

Tabela 21 - Espaço ocupado por cada entrada na tabela Clube

Localidade	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	designacao	VARCHAR(45)	46
	id_Pais	INT	4
Total			54

Tabela 22 - Espaço ocupado por cada entrada na tabela Localidade

Historico_Clubes	Atributo	Tipos de Dados	Espaço no disco
	id_Jogador	INT	4
	id_Clube_anterior	VARCHAR(45)	46
	inicio_contrato	DATE	3
	fim_contrato	DATE	3
	golos_marcados	INT	4
	reputacao	INT	4
Total			64

Tabela 23 - Espaço ocupado por cada entrada na tabela Historico_Clubes

Posicao	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	designacao	VARCHAR(45)	46
Total			50

Tabela 24 - Espaço ocupado por cada entrada na tabela Posicao

Tipo	Atributo	Tipos de Dados	Espaço no disco
	id	INT	4
	designacao	VARCHAR(45)	46
Total			50

Tabela 25 - Espaço ocupado por cada entrada na tabela Tipo

Tabela	Espaço Em disco
Jogador	$20 \times 107 = 2140$
Agente	$6 \times 187 = 1122$
Olheiro	$5 \times 111 = 555$
Posicao	$11 \times 50 = 550$
Clube_anterior	$24 \times 50 = 1200$
Pais	$13 \times 50 = 650$
Nacionalidade	$13 \times 50 = 650$
Clube	$16 \times 50 = 800$
Localidade	$24 \times 54 = 1296$
Historico_clubes	$31 \times 64 = 1984$
Tipo	$2 \times 50 = 100$
Total	11147

Tabela 26 - Espaço total ocupado pela base de dados

Observando a tabela em cima podemos ver que, neste momento, a estimativa para a nossa base de dados é cerca de 11047 bytes + tamanho do TEXT (também em bytes). Contudo, temos de ter noção que à medida que se vai adicionando mais jogadores à FuteProject e sempre que um jogador troca de clube esta estimativa vai aumentar.

Obs: Decidimos que o email vai ocupar 76 bytes (mais 30 bytes que o resto dos VARCHAR's) porque o email de uma pessoa pode ser bastante extenso. Para identificar as entidades decidimos escolher inteiros para ocupar menos espaço possível e continuar a ser executável. No tipo de Dados, onde se encontra "TEXT" decidimos não dar um valor definitivo visto que, este atributo, pode conter um número infundável de bytes, (o olheiro pode ter muito que dizer sobre esse jogador).

5.6 Definição e caracterização das vistas de utilização em SQL

Nesta secção vamos apresentar algumas das *views* que consideramos que serão mais uteis para os utilizadores da base de dados.

- **Exemplo 1:**

```
-- Aplicada na querie 19 para ir buscar os jogadores que passaram por um clube
Drop view if exists historico_clubes_clube;
Create view historico_clubes_clube as
Select j.nome, c.designacao, t.inicio_contrato From Historico_clubes t INNER JOIN Clube_anterior c on
c.id=t.Clube_anterior_id INNER JOIN Jogador j on t.Jogador_id = j.id;
```

Figura 23 - *view* para a *query* nº 19

A Figura 28 - *view* para a *query* nº 19 representa a *view* implementada para ir buscar os jogadores que passaram por um determinado clube.

- **Exemplo 2:**

```
-- Aplicada na querie 31 para ir buscar os clubes pelos quais um determinado jogador já passou
Drop view if exists historico_clubes_jogador;
Create view historico_clubes_jogador as
Select j.nome as nome , ca.designacao as Clube_anterior, h.inicio_contrato as inicio_contrato , h.fim_contrato as fim_contrato From Jogador j
INNER JOIN Historico_clubes h on j.id = h.Jogador_id
INNER JOIN Clube_anterior ca on h.Clube_anterior_id=ca.id;
```

Figura 24 - *view* para a *query* nº 31

A Figura 29 - *view* para a *query* nº 31 representa a *view* implementada para ir buscar os clubes pelos quais um determinado jogador já passou.

- **Exemplo 3:**

A Figura 30 - *view* para a *query* nº28 representa a *view* implementada para ir buscar os dados estatísticos de um respetivo jogador num respetivo clube, como por exemplo nº de golos marcados naquele clube ou a reputação que o clube deu àquele Jogador.


```

15  -- Aplicada na querie 28 para ir buscar a info da passagem do jogador por um clube
16  Drop view if exists historico_reputacao_jogador;
17  Create view historico_reputacao_jogador as
18  SELECT j.id, nome, h.inicio_contrato, h.fim_contrato, c.designacao, h.reputacao
19  FROM Historico_clubes h
20  INNER JOIN Jogador j on h.Jogador_id = j.id
21  INNER JOIN Clube_anterior c on h.Clube_anterior_id = c.id
22  UNION
23  SELECT j.id, j.nome, j.inicio_contrato, j.fim_contrato, cl.designacao, j.reputacao From Clube cl
24  INNER JOIN Jogador j on j.Clube_id = cl.id;

```

Figura 25 - view para a query nº28

5.7 Definição e Caracterização dos Mecanismos de Segurança em SQL

Na realização deste projeto decidimos que seria necessário implementar medidas de segurança de modo a assegurar a proteção dos dados presentes na nossa Base de Dados. Para tal, é essencial que a nossa própria base de dados seja um mecanismo de defesa contra qualquer tipo de más intenções para contra a FuteProject.

Assim, implementamos as permissões que cada entidade vai ter, ou seja, vamos dar permissão às entidades para modificar ou adicionar atributos à base de dados que lhe competem modificar. Por exemplo, como verificamos na Figura 31 - Criação e atribuição de permissões do Agente e do Olheiro, cada agente vai ter permissão para: atualizar os dados de um jogador ao final de cada época, adicionar ou remover um jogador da FuteProject e quando um jogador troca de clube, este atualiza o clube atual do jogador e adiciona o clube antigo ao histórico de clubes.

Apenas o olheiro vai ter permissões para adicionar um novo jogador à categoria Jovem Promessa e quando assim o faz, tem a possibilidade comentar a sua avaliação e um comentário sobre o Jogador.

```

1 • USE FUTEPROJECT;
2
3 • CREATE ROLE Agentes;
4 • CREATE ROLE Olheiros;
5
6   ### Agente
7 • Grant SELECT,UPDATE,INSERT ON FuteProject.* TO Agentes;
8 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addClubeAnterior TO Agentes;
9 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addHistorico_clube TO Agentes;
10 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.criaclube TO Agentes;
11 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.resetaDadosJogador TO Agentes;
12 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.alteraStatusJogador TO Agentes;
13 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.adicionaJogador TO Agentes;
14 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.removeJogador TO Agentes;
15 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.recrutarJovemPromessa TO Agentes;
16 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addLocalidade TO Agentes;
17 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addPais TO Agentes;
18 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addNacionalidade TO Agentes;
19
20   ### Olheiro
21 • Grant INSERT, SELECT ON FuteProject.* TO Olheiros;
22 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.adicionaJovemPromessa TO Olheiros;
23 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addLocalidade TO Olheiros;
24 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addPais TO Olheiros;
25 • Grant EXECUTE ON PROCEDURE FUTEPROJECT.addNacionalidade TO Olheiros;
26   ##
27
28 • CREATE USER 'Jorge_Mendes'@'localhost' IDENTIFIED BY 'Jorge_mendes';
29 • GRANT Agentes to 'Jorge_Mendes'@'localhost';
30
31 • CREATE USER 'Aurelio Pereira'@'localhost' IDENTIFIED BY 'Aurelio_Pereira';
32 • GRANT Olheiro to 'Aurelio Pereira'@'localhost';
33

```

Figura 26 - Criação e atribuição de permissões do Agente e do Olheiro

5.8 Revisão do sistema implementado

Com a implementação da base de dados concluída decidimos reunirmo-nos uma última vez com a administração da FuteProject e validar/confirmar o projeto e, se estiver tudo dentro dos objetivos pretendidos, dar por concluído o desenvolvimento deste sistema. De modo a apresentar o nosso trabalho para a administração da FuteProject decidimos seguir estes quatro tópicos:

- No primeiro tópico mostramos que a base de dados cumpria os vários objetivos definidos pela FuteProject e demonstramos algumas capacidades adicionais que achamos interessantes implementar.
- No segundo tópico demonstramos o “código” e lógica, que nós seguimos, que tornou capaz o funcionamento da base de dados , ou seja , mostramos as operações necessárias para a inserção de dados no sistema, tal como adicionar um novo jogador ou um novo agente, e como o sistema reage quando estes novos dados prontos a inserir estão incorretos.
- No terceiro tópico mostramos à administração as estimativas de espaço ocupado pela base de dados e os mecanismos de segurança “instalados” de modo a proteger a informação presente na base de dados e garantir que cada entidade só modifica o que tiver permissão para modificar.
- Último tópico consistiu em retirar todas as possíveis dúvidas que os Agentes responsáveis pela FuteProject possam ter sobre a implementação do projeto.

Esclarecidas todas as dúvidas, administração deu-nos o contrato e luz verde para imediata implementação do sistema.

6 Conclusão e Trabalho Futuro

A base de dados desenvolvida vai permitir, sem dúvida alguma, uma melhor organização da agência e consequentemente impulsionar a FuteProject para um novo patamar de excelência e organização. Estas qualidades vão ser estimadas e tornar a FuteProject mais desejada e com mais jogadores interessados.

Este software vai representar uma melhor estratégia na apresentação dos jogadores contratados a futuras equipas interessadas. Futuras equipas, que, graças à nova base de dados, podem ser de qualquer parte do mundo pois este software vai-se tornar disponível para todos os clubes do mundo aumentando exponencialmente a visibilidade dos jogadores. Assim como, graças à categoria jovens promessas, vai tornar muito mais fácil encontrar novos jogadores com o talento necessário para representar a FuteProject. De modo a que estes objetivos sejam cumpridos da forma mais segura possível foi essencial garantir, desde o início, a consistência, integridade e segurança das informações fornecidas à nossa base de dados.

Posto isto, ao longo deste relatório, foram apresentadas todas as condições do sistema em que a base de dados se iria inserir, as necessidades do mesmo e métodos utilizados.

Na fase inicial, foi abordado o contexto que envolve a FuteProject e os principais objetivos pretendidos e analisamos os requisitos necessários para a implementação da mesma. Depois de definidos o principal requisito foi elaborado um modelo conceptual e de seguida transformamos o mesmo num modelo lógico, implementamos este num sistema de gestão de base de dados relacional e transformando assim o projeto num modelo físico.

No início da realização deste projeto sentimos alguma dificuldade em compreender quais as entidades necessárias para cumprir de forma mais eficiente o projeto em mãos, sendo que, à medida que avançávamos e compreendíamos um pouco mais, fizemos bastantes alterações no modelo original e a lógica a seguir no trabalho.

A realização deste projeto permitiu-nos compreender e perceber o desenvolvimento de uma base de dados, do início ao fim, e aprofundar/consolidar os nossos conhecimentos adquiridos ao longo deste semestre.

Com este trabalho concluído, esperamos ter cumprido todos os requisitos propostos pelo docente.

7 Referências

- Connolly, T. and Begg, C. (2005). Database Systems, A Practical Approach to Design, Implementation, and Management. 4th ed. Addison-Wesley. Mysql. (2018).
- MySQL. [Online]. [26 November 2018]. Available from: <https://dev.mysql.com/doc/refman/8.0/en/>.
- Transfermarkt.pt. 2020. Mercado De Transferências, Rumores, Valores De Mercado E Notícias. [online] Available at: <https://www.transfermarkt.pt/>
- Pt.soccerwiki.org. 2020. Soccerwiki.Org. [online] Available at: <https://pt.soccerwiki.org/wiki.php>

8 Anexos

8.1 Anexo 1 – *Script* de Inicialização da Base de Dados

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN
_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUB
STITUTION';

-----
-- Schema FuteProject
-----

-----
-- Schema FuteProject
-----
CREATE SCHEMA IF NOT EXISTS `FuteProject` DEFAULT CHARACTER SET
utf8 ;
USE `FuteProject` ;

-----
-- Table `FuteProject`.`Tipo`
-----
CREATE TABLE IF NOT EXISTS `FuteProject`.`Tipo` (
  `id` INT NOT NULL,
  `designacao` VARCHAR(45) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-----
-- Table `FuteProject`.`Posicao`
-----
CREATE TABLE IF NOT EXISTS `FuteProject`.`Posicao` (
  `id` INT NOT NULL,
  `designacao` VARCHAR(45) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-----
-- Table `FuteProject`.`Clube`
-----
CREATE TABLE IF NOT EXISTS `FuteProject`.`Clube` (
  `id` INT NOT NULL,
  `designacao` VARCHAR(45) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
```

-- Table `FuteProject`.`País`

```
CREATE TABLE IF NOT EXISTS `FuteProject`.`País` (  
  `id` INT NOT NULL,  
  `nacionalidade` VARCHAR(45) NULL,  
  `designacao` VARCHAR(45) NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

-- Table `FuteProject`.`Localidade`

```
CREATE TABLE IF NOT EXISTS `FuteProject`.`Localidade` (  
  `id` INT NOT NULL,  
  `designacao` VARCHAR(45) NULL,  
  `País_id` INT NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `fk_Localidade_País1_idx` (`País_id` ASC) VISIBLE,  
  CONSTRAINT `fk_Localidade_País1`  
    FOREIGN KEY (`País_id`)  
    REFERENCES `FuteProject`.`País` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `FuteProject`.`Nacionalidade`

```
CREATE TABLE IF NOT EXISTS `FuteProject`.`Nacionalidade` (  
  `id` INT NOT NULL,  
  `designacao` VARCHAR(45) NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

-- Table `FuteProject`.`Agente`

```
CREATE TABLE IF NOT EXISTS `FuteProject`.`Agente` (  
  `id` INT NOT NULL,  
  `nome` VARCHAR(45) NULL,  
  `data_nascimento` DATE NULL,  
  `idade` INT NULL,  
  `reputacao` INT NULL,  
  `telefone` VARCHAR(9) NULL,  
  `email` VARCHAR(75) NULL,  
  `Nacionalidade_id` INT NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `fk_Agente_Nacionalidade1_idx` (`Nacionalidade_id` ASC) VISIBLE,  
  CONSTRAINT `fk_Agente_Nacionalidade1`  
    FOREIGN KEY (`Nacionalidade_id`)  
    REFERENCES `FuteProject`.`Nacionalidade` (`id`)  
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `FuteProject`.`Olheiro`
-----
```

```
CREATE TABLE IF NOT EXISTS `FuteProject`.`Olheiro` (
  `id` INT NOT NULL,
  `nome` VARCHAR(45) NULL,
  `data_nascimento` DATE NULL,
  `idade` INT NULL,
  `reputacao` INT NULL,
  `telefone` VARCHAR(9) NULL,
  `Nacionalidade_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Olheiro_Nacionalidade1_idx` (`Nacionalidade_id` ASC) VISIBLE,
  CONSTRAINT `fk_Olheiro_Nacionalidade1`
    FOREIGN KEY (`Nacionalidade_id`)
      REFERENCES `FuteProject`.`Nacionalidade` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `FuteProject`.`Jogador`
-----
```

```
CREATE TABLE IF NOT EXISTS `FuteProject`.`Jogador` (
  `id` INT NOT NULL,
  `nome` VARCHAR(45) NULL,
  `data_nascimento` DATE NULL,
  `idade` INT NULL,
  `altura` INT NULL,
  `nr_internacionalizacoes` INT NULL,
  `valor_mercado` DECIMAL(5,2) NULL,
  `reputacao` INT NULL,
  `golos_marcados` INT NULL,
  `Tipo_id` INT NOT NULL,
  `Posicao_id` INT NOT NULL,
  `Clube_id` INT NOT NULL,
  `Localidade_id` INT NOT NULL,
  `Nacionalidade_id` INT NOT NULL,
  `Agente_id` INT NULL,
  `Olheiro_id` INT NOT NULL,
  `avaliacao` INT NULL,
  `comentario` VARCHAR(45) NULL,
  `fim_contrato` DATE NOT NULL,
  `inicio_contrato` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Jogador_Tipo1_idx` (`Tipo_id` ASC) VISIBLE,
  INDEX `fk_Jogador_Posicao1_idx` (`Posicao_id` ASC) VISIBLE,
  INDEX `fk_Jogador_Clube1_idx` (`Clube_id` ASC) VISIBLE,
  INDEX `fk_Jogador_Localidade1_idx` (`Localidade_id` ASC) VISIBLE,
  INDEX `fk_Jogador_Agente1_idx` (`Agente_id` ASC) VISIBLE,
  INDEX `fk_Jogador_Nacionalidade1_idx` (`Nacionalidade_id` ASC) VISIBLE,
  INDEX `fk_Jogador_Olheiro1_idx` (`Olheiro_id` ASC) VISIBLE,
```



```

CONSTRAINT `fk_Jogador_Tipo1`
  FOREIGN KEY (`Tipo_id`)
    REFERENCES `FuteProject`.`Tipo` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Jogador_Posicao1`
  FOREIGN KEY (`Posicao_id`)
    REFERENCES `FuteProject`.`Posicao` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Jogador_Clube1`
  FOREIGN KEY (`Clube_id`)
    REFERENCES `FuteProject`.`Clube` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Jogador_Localidade1`
  FOREIGN KEY (`Localidade_id`)
    REFERENCES `FuteProject`.`Localidade` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Jogador_Agente1`
  FOREIGN KEY (`Agente_id`)
    REFERENCES `FuteProject`.`Agente` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Jogador_Nacionalidade1`
  FOREIGN KEY (`Nacionalidade_id`)
    REFERENCES `FuteProject`.`Nacionalidade` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Jogador_Olheiro1`
  FOREIGN KEY (`Olheiro_id`)
    REFERENCES `FuteProject`.`Olheiro` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `FuteProject`.`Clube_anterior`
-----

```

```

CREATE TABLE IF NOT EXISTS `FuteProject`.`Clube_anterior` (
  `id` INT NOT NULL,
  `designacao` VARCHAR(45) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

```

-----
-- Table `FuteProject`.`Historico_clubes`
-----

```

```

CREATE TABLE IF NOT EXISTS `FuteProject`.`Historico_clubes` (
  `Jogador_id` INT NOT NULL,
  `Clube_anterior_id` INT NOT NULL,
  `fim_contrato` DATE NOT NULL,
  `inicio_contrato` DATE NOT NULL,
  `golos` INT NOT NULL,

```

```

`reputacao` INT NOT NULL,
INDEX `fk_Jogador_has_Historico_clubes_Historico_clubes1_idx`
(`Clube_anterior_id` ASC) VISIBLE,
INDEX `fk_Jogador_has_Historico_clubes_Jogador1_idx` (`Jogador_id` ASC)
VISIBLE,
CONSTRAINT `fk_Jogador_has_Historico_clubes_Jogador1`
FOREIGN KEY (`Jogador_id`)
REFERENCES `FuteProject`.`Jogador` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Jogador_has_Historico_clubes_Historico_clubes1`
FOREIGN KEY (`Clube_anterior_id`)
REFERENCES `FuteProject`.`Clube_anterior` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

8.2 Anexo 2 – *Script* de Povoamento Inicial

```

USE `FuteProject`;

INSERT INTO `FuteProject`.`Tipo`
(id,designacao)
VALUES
("01","Contratado"),
("02","Jovem Promessa");

INSERT INTO `FuteProject`.`Posicao`
(id,designacao)
VALUES
("01","Guarda-Redes"),
("02","Defesa Central"),
("03","Lateral Esquerdo"),
("04","Lateral Direito"),
("05","Médio Defensivo"),
("06","Médio Centro"),
("07","Médio Ofensivo"),
("08","Extremo Esquerdo"),
("09","Extremo Direito"),
("10","Segundo Avançado"),
("11","Ponta de Lança");

INSERT INTO `FuteProject`.`Clube`
(id,designacao)
VALUES
("01","FC Porto"),
("02","Juventus"),
("03","Wolves"),
("04","Liverpool"),
("05","Barcelona"),

```

```

("06","Real Madrid"),
("07","Lyon"),
("08","Dortmund"),
("09","Atlético Madrid"),
("10","AC Milan"),
("11","Manchester United"),
("12","Everton"),
("13","PSG"),
("14","FC Porto B"),
("15","Famalicão"),
("16","Académica");

```

```

INSERT INTO `FuteProject`.`País`
(id,designacao)
VALUES
("01","Portugal"),
("02","México"),
("03","Máli"),
("04","Inglaterra"),
("05","Alemanha"),
("06","Holanda"),
("07","Noruega"),
("08","Espanha"),
("09","Suécia"),
("10","França"),
("11","Colômbia"),
("12","Argentina"),
("13","Brasil");

```

```

INSERT INTO `FuteProject`.`Nacionalidade`
(id,designacao)
VALUES
("01","português"),
("02","mexicano"),
("03","maliano"),
("04","inglês"),
("05","alemão"),
("06","holandês"),
("07","noruegues"),
("08","espanhol"),
("09","sueco"),
("10","francês"),
("11","colombiano"),
("12","argentino"),
("13","brasileiro");

```

```

INSERT INTO `FuteProject`.`Localidade`
(id,designacao,País_id)
VALUES
("01","Madeira","01"),
("02","Hermosillo","02"),
("03","Santa Maria da Feira","01"),
("04","Porto","01"),
("05","Les Ullis","03"),
("06","Liverpool","04"),
("07","Monchengladbach","05"),

```

```

("08","Sevilla","08"),
("09","Moordrecht","06"),
("10","Leeds","04"),
("11","Viseu","01"),
("12","Breda","06"),
("13","Malmo","09"),
("14","Lagny-sur-Marne","10"),
("15","Cucuta","11"),
("16","Rosario","12"),
("17","Barcelos","01"),
("18","Famalicão","01"),
("19","Aveiro","01"),
("20","São Paulo","13"),
("21","Londres","04"),
("22","Amsterdão","06"),
("23","Lisboa","01"),
("24","Braga","01");

```

```

INSERT INTO `FuteProject`.`Agente`
(id,data_nascimento,idade,nome,reputacao,telefone,email,Nacionalidade_id)
VALUES
("01","1966-01-07","54","Jorge
Mendes","10","931238129","jorgemendes@gmail.com","01"),
("02","1985-09-23","35","Valério
Sousa","8","912934726","valeriosousa@gmail.com","01"),
("03","1976-03-11","44","Gilson
Ribeiro","7","921353172","gilson76ribeiro@gmail.com","13"),
("04","1980-02-28","40","Hipólito
Crispim","8","912645859","hipocrispo@gmail.com","01"),
("05","1985-11-09","35","John
Blake","9","912276645","j_blake@gmail.com","04"),
("06","1960-11-30","60","Mino
Raiola","6","917639758","mraiola@gmail.com","06");

```

```

INSERT INTO `FuteProject`.`Olheiro`
(id,nome,data_nascimento,idade,reputacao,telefone,Nacionalidade_id)
VALUES
("01","Aurélío Pereira","1975-05-15","45","8","963838381","01"),
("02","Rod Ruddick","1989-12-27","31","7","933728983","04"),
("03","Gilmar Francisco","1970-01-03","50","9","923609390","13"),
("04","Ricardo Bochini","1983-06-19","37","8","910293930","02"),
("05","Rafa Monfort","1981-09-23","39","6","951938128","08");

```

```

-- ECT * FROM `FuteProject`.`Jogador`;
INSERT INTO `FuteProject`.`Jogador`

```

```

(id,nome,data_nascimento,idade,altura,nr_internacionalizacoes,valor_mercado,rep
utacao,golos_marcados,Tipo_id,Posicao_id,Clube_id,Localidade_id,Nacionalidade
_id,Agente_id,Olheiro_id,avaliacao,comentario,fim_contrato,inicio_contrato)
VALUES
("01","Cristiano Ronaldo","1985-02-
05","35","187","204","60","97","89","01","08","02","01","01","01","10","Melhor de
todos os tempos","2023-07-15","2018-07-13"),
("02","Jesús Corona","1993-01-
06","27","173","42","30","89","49","01","09","01","02","02","02","02",null,null,"2021-
09-13","2016-10-25"),

```

("03", "Rúben Neves", "1997-03-13", "23", "180", "18", "50", "91", "110", "01", "05", "03", "03", "01", "03", "03", null, null, "2022-08-29", "2017-06-29"),

("04", "Fábio Silva", "2002-07-19", "18", "185", "0", "25", "78", "67", "01", "11", "03", "04", "01", "04", "04", null, null, "2024-07-18", "2020-07-15"),

("05", "Moussa Marega", "1991-04-14", "29", "183", "24", "100", "89", "89", "01", "10", "01", "05", "03", "05", "05", "10", "Um bocado melhor que o Ronaldo", "2021-09-15", "2017-07-29"),

("06", "Trent Alexander-Arnold", "1998-10-07", "22", "180", "12", "110", "93", "12", "01", "04", "04", "06", "04", "06", "01", null, null, "2021-09-19", "2015-06-14"),

("07", "Marc-André ter Stegen", "1992-04-30", "28", "187", "24", "75", "95", "0", "01", "01", "05", "07", "05", "01", "02", null, null, "2022-07-13", "2014-07-17"),

("08", "Sérgio Ramos", "1986-03-30", "34", "184", "178", "14", "95", "95", "01", "03", "06", "08", "08", "02", "03", null, null, "2023-08-04", "2013-08-14"),

("09", "Memphis Depay", "1994-02-13", "26", "176", "54", "45", "83", "120", "01", "08", "07", "09", "06", "03", "04", null, null, "2021-09-02", "2014-08-12"),

("10", "Erling Haaland", "2000-07-21", "20", "194", "7", "100", "90", "80", "01", "11", "08", "10", "07", "04", "05", null, null, "2024-08-19", "2020-01-10"),

("11", "João Félix", "1999-11-10", "21", "180", "11", "100", "85", "48", "01", "10", "09", "11", "01", "05", "01", null, null, "2023-09-14", "2019-08-05"),

("12", "Virgil Van Dijk", "1991-07-08", "29", "193", "34", "80", "96", "43", "01", "03", "04", "12", "06", "06", "02", null, null, "2021-09-09", "2017-08-14"),

("13", "Zlatan Ibrahimovic", "1981-10-03", "39", "195", "116", "3", "81", "45", "01", "11", "10", "13", "09", "01", "03", "9", "Nada a apontar. Todos conhecem o Zlatan", "2021-09-13", "2019-07-03"),

("14", "Paul Pogba", "1993-03-15", "27", "191", "74", "80", "93", "67", "01", "06", "11", "14", "10", "02", "04", null, null, "2021-10-01", "2018-09-14"),

("15", "James Rodriguez", "1991-07-12", "29", "181", "76", "35", "85", "79", "01", "07", "12", "15", "11", "03", "05", null, null, "2021-08-13", "2018-09-05"),

("16", "Ángel Di María", "1988-02-14", "32", "180", "102", "32", "84", "128", "01", "09", "13", "16", "12", "04", "01", null, null, "2021-06-29", "2018-07-30"),

("17", "Gonçalo Almeida", "2000-05-19", "20", "186", "0", "5", "70", "15", "02", "11", "14", "17", "01", null, "02", "5", "Melhor jogador que já passou pelo Arrentela", "2021-09-13", "2019-10-02"),

("18", "Diogo Amaro", "2000-06-26", "20", "193", "0", "3", "69", "0", "02", "01", "15", "18", "01", null, "03", null, null, "2022-10-01", "2018-08-06"),

("19", "Luís Pereira", "2000-06-21", "20", "190", "0", "3", "65", "2", "02", "02", "01", "17", "01", null, "04", null, null, "2021-06-29", "2019-08-13"),

("20", "Leonardo Marreiros", "2000-12-05", "19", "183", "0", "3", "68", "5", "02", "09", "16", "19", "01", null, "05", null, null, "2021-06-25", "2019-09-16");

```
INSERT INTO `FuteProject`.`Clube_anterior`
```

```
(id,designacao)
```

```
VALUES
```

```
("01","Manchester United"),  
("02","Real Madrid"),  
("03","Sporting"),  
("04","Southampton"),  
("05","Celtic"),  
("06","FC Porto"),  
("07","PSV"),  
("08","Juventus"),  
("09","Salzburg"),  
("10","PSG"),  
("11","AC Milan"),  
("12","Barcelona"),  
("13","Inter"),  
("14","Benfica"),  
("15","Monaco"),  
("16","Bayern"),  
("17","Sevilla"),  
("18","Liverpool Formation"),  
("19","Twente"),  
("20","Monterrey"),  
("21","Marítimo"),  
("22","Vitória SC"),  
("23","Amiens SC"),  
("24","Monchengladbach");
```

```
INSERT INTO `FuteProject`.`Historico_clubes`
```

```
(Jogador_id,Clube_anterior_id,fim_contrato,inicio_contrato,golos,reputacao)
```

```
VALUES
```

```
("01","01","2009-07-01","2003-08-12","118","80"),  
("01","02","2018-07-13","2009-07-01","408","95"),  
("01","03","2003-08-12","1998-08-14","10","65"),  
("02","19","2016-10-25","2013-06-08","15","60"),  
("02","20","2013-06-08","2009-07-14","7","57"),  
("03","06","2017-06-29","2014-07-14","8","60"),  
("04","06","2020-07-15","2018-07-14","2","65"),  
("05","21","2017-07-29","2015-07-13","18","65"),  
("05","22","2015-07-13","2014-06-14","13","65"),  
("05","23","2014-06-14","2012-07-20","9","55"),  
("06","18","2015-06-14","2010-07-13","5","40"),  
("07","24","2014-07-17","2010-08-19","1","60"),  
("08","17","2013-08-14","2009-07-14","6","65"),  
("09","01","2018-07-09","2014-08-12","15","60"),  
("09","07","2014-08-12","2010-09-09","10","50"),  
("10","09","2020-01-10","2017-07-27","29","70"),  
("11","14","2019-08-05","2018-09-18","14","70"),  
("12","04","2017-08-14","2015-09-02","4","50"),  
("12","05","2015-09-02","2012-08-14","1","40"),  
("13","13","2013-07-14","2009-08-15","26","65"),  
("13","12","2015-08-10","2013-07-14","14","70"),  
("13","11","2017-09-28","2015-08-10","28","70"),  
("13","08","2009-08-15","2006-07-17","14","60"),  
("13","10","2019-07-03","2017-09-28","80","80"),  
("14","08","2018-09-14","2013-08-08","31","75"),  
("15","06","2013-07-23","2008-09-25","16","70"),
```

```

("15","15","2015-08-27","2013-07-23","19","75"),
("15","02","2018-09-05","2015-08-27","13","75"),
("16","14","2014-07-29","2010-09-07","37","70"),
("16","02","2017-09-01","2014-07-29","23","75"),
("16","01","2018-07-30","2017-09-01","9","70");

```

8.3 Anexo 3 - Funcionalidades para o Agente

```

-- funções agente
USE `FuteProject`;

```

```

-- Função que cria clube anterior

```

```

Drop procedure if exists addClubeAnterior;
DELIMITER //
CREATE PROCEDURE addClubeAnterior ( IN designacao VARCHAR(128))
Begin
DECLARE n INT;
declare teste VARCHAR(45);
SET n =(SELECT COUNT(*) From Clube_anterior);
set teste = (select c.designacao from Clube_anterior c where c.designacao =
designacao);
IF (teste is null) then
INSERT INTO Clube_anterior(id,designacao)
Values(n+1,designacao);
END IF;
END //
DELIMITER //

```

```

-- criar Historivo_clube

```

```

Drop procedure if exists addHistorico_clube;
DELIMITER //
Create procedure addHistorico_clube (IN id_jogador INT,inicio_contrato DATE
,fim_contrato DATE,golos INT,id_clubeAntigo INT, reputacao INT)
BEGIN
INSERT INTO Historico_clubes
(Jogador_id,Clube_anterior_id,inicio_contrato,fim_contrato,golos,reputacao)
Values (id_jogador,id_clubeAntigo,inicio_contrato,fim_contrato,golos, reputacao);
END //
DELIMITER //

```

```

-- cria uma entrada na tabela clube

```

```

Drop procedure if exists criaClube;
DELIMITER //
CREATE PROCEDURE criaClube ( IN designacao VARCHAR(128))
Begin
declare teste VARCHAR(45);
DECLARE n INT;
set teste = (select c.designacao from Clube c where c.designacao = designacao);
IF (teste is null) then
SET n =(SELECT COUNT(*) From Clube);

```

```

INSERT INTO Clube (id,designacao)
Values(n+1,designacao);
END IF;
END //
DELIMITER //

```

```

-- iniciar dados de jogador num novo clube
Drop procedure if exists resetaDadosJogador;
DELIMITER //
CREATE PROCEDURE resetaDadosJogador ( IN id_novoClube VARCHAR(128),
inicio_contrato date, fim_contrato date, id_jogador int)
Begin
Update Jogador
SET Clube_id=id_novoClube,golos_marcados=0, inicio_contrato=inicio_contrato,
fim_contrato=fim_contrato
where id=id_jogador;
END //
DELIMITER //

```

```

DELIMITER //
Create procedure alteraStatusJogador(in id_jogador INT, in golos INT,in valor INT,
in rep INT, IN nr_inter INT)
BEGIN
UPDATE Jogador
SET golos_marcados = golos, reputacao=rep, valor_mercado=valor,
nr_internacionalizacoes=nr_inter, idade=idade(j.dataNascimento)
Where j.id=id_jogador;
END //
DELIMITER //

```

-- agente adiciona um jogador à base de dados

```

Drop procedure if exists adicionaJogador;
DELIMITER //
Create procedure adicionaJogador(in nome VARCHAR(45), dataNascimento date,
idade int, altura int, nr_internacionalizacoes int, valor_mercado int, reputacao int,
golos_marcados int, tipo_id int, posicao_id int, clube_id int,
localidade_id int, nacionalidade_id int, agente_id int, olheiro_id int, avaliacao int,
comentario text, fim_contrato date, inicio_contrato date)
BEGIN
UPDATE Jogador

SET n =(SELECT COUNT(*) From Jogador);
INSERT INTO Jogador
(id,nome,data_nascimento,idade,altura,nr_internacionalizacoes,valor_mercado,rep
utacao,golos_marcados,Tipo_id,Posicao_id,Clube_id,Localidade_id,Nacionalidade
_id,Agente_id,Olheiro_id,avaliacao,comentario,fim_contrato,inicio_contrato)
Values(n+1, nome, dataNascimento, idade, altura, nr_internacionalizacoes,
valor_mercado, reputacao, golos_marcados, tipo_id, posicao_id, clube_id,

```



```

localidade_id, nacionalidade_id, agente_id, olheiro_id, avaliacao, comentario,
fim_contrato, inicio_contrato);
END //
DELIMITER //

```

-- agente remove um jogador da base de dados

```

Drop procedure if exists removeJogador;
DELIMITER //
Create procedure removeJogador(in idJogador INT)
BEGIN
delete from Historico_clubes h where h.Jogador_id = idJogador;
delete from Jogador j where j.id = idJogador;
END //
DELIMITER //

```

```

call removeJogador("02");

```

-- agente recruta uma jovem promessa

```

Drop procedure if exists recrutarJovemPromessa;
DELIMITER //
Create procedure recrutarJovemPromessa(in idJogador INT, agente_id int)
BEGIN
UPDATE Jogador
SET Tipo_id="01", Agente_id=agente_id
WHERE id=idJogador;
END //
DELIMITER //

```

8.4 Anexo 4 - Funcionalidades para o Olheiro

-- funções do olheiro

-- adicionar uma jovem promessa à base de dados

```

Drop procedure if exists adicionaJovemPromessa;
DELIMITER //
Create procedure adicionaJovemPromessa(in nome VARCHAR(45),
dataNascimento date, idade int, altura int, nr_internacionalizacoes int,
valor_mercado int, reputacao int,
golos_marcados int, posicao_id int, clube_id int,
localidade_id int, nacionalidade_id int, olheiro_id int, avaliacao int, comentario text,
fim_contrato date, inicio_contrato date)
BEGIN
UPDATE Jogador

SET n =(SELECT COUNT(*) From Jogador);
INSERT INTO Jogador
(id,nome,data_nascimento,idade,altura,nr_internacionalizacoes,valor_mercado,rep
utacao,golos_marcados,Tipo_id,Posicao_id,Clube_id,Localidade_id,Nacionalidade
_id,Agente_id,Olheiro_id,avaliacao,comentario,fim_contrato,inicio_contrato)
Values(n+1, nome, dataNascimento, idade, altura, nr_internacionalizacoes,
valor_mercado, reputacao, golos_marcados, "02", posicao_id, clube_id,

```

```

localidade_id, nacionalidade_id, null, olheiro_id, avaliacao, comentario,
fim_contrato, inicio_contrato);
END //
DELIMITER //

```

8.5 Anexo 5 - Script de Criação das Vistas que respondem a parte das Interrogações

```

-- Aplicada na querie 19 para ir buscar os jogadores que passaram por um clube
Drop view if exists historico_clubes_clube;
Create view historico_clubes_clube as
Select j.nome, c.designacao, t.inicio_contrato From Historico_clubes t INNER JOIN
Clube_anterior c on
c.id=t.Clube_anterior_id INNER JOIN Jogador j on t.Jogador_id = j.id;

```

```

-- Aplicada na querie 31 para ir buscar os clubes pelos quais um determinado
jogador já passou
Drop view if exists historico_clubes_jogador;
Create view historico_clubes_jogador as
Select j.nome as nome , ca.designacao as Clube_anterior, h.inicio_contrato as
inicio_contrato , h.fim_contrato as fim_contrato From Jogador j
INNER JOIN Historico_clubes h on j.id = h.Jogador_id
INNER JOIN Clube_anterior ca on h.Clube_anterior_id=ca.id;

```

```

-- Aplicada na querie 28 para ir buscar a info da passagem do jogador por um clube
Drop view if exists historico_reputacao_jogador;
Create view historico_reputacao_jogador as
SELECT j.id, nome, h.inicio_contrato, h.fim_contrato, c.designacao, h.reputacao
FROM Historico_clubes h
INNER JOIN Jogador j on h.Jogador_id = j.id
INNER JOIN Clube_anterior c on h.Clube_anterior_id = c.id
UNION
SELECT j.id, j.nome, j.inicio_contrato, j.fim_contrato, cl.designacao, j.reputacao
From Clube cl
INNER JOIN Jogador j on j.Clube_id = cl.id;

```

8.6 Anexo 6 - Funcionalidades Comuns ao Olheiro e Agente

-- funções usadas pelos agentes e olheiros

-- adicionar localidade nova

```

Drop procedure if exists addLocalidade;
DELIMITER //
CREATE PROCEDURE addLocalidade ( IN designacao VARCHAR(45), idPais
INT)
Begin

```

```

DECLARE n INT;
declare teste VARCHAR(45);
SET n =(SELECT COUNT(*) From Localidade);
set teste = (select l.designacao from Localidade l where l.designacao =
designacao);
IF (teste is null) then
INSERT INTO Localidade(id,designacao,País_id)
Values(n+1,designacao,idPaís);
END IF;
END //
DELIMITER //

```

-- adicionar país novo

```

Drop procedure if exists addPaís;
DELIMITER //
CREATE PROCEDURE addPaís (IN designacao VARCHAR(45))
Begin
DECLARE n INT;
declare teste VARCHAR(45);
SET n =(SELECT COUNT(*) From País);
set teste = (select p.designacao from País p where p.designacao = designacao);
IF (teste is null) then
INSERT INTO País(id,designacao)
Values(n+1,designacao);
END IF;
END //
DELIMITER //

```

-- adicionar nacionalidade nova

```

Drop procedure if exists addNacionalidade;
DELIMITER //
CREATE PROCEDURE addNacionalidade (IN designacao VARCHAR(45))
Begin
DECLARE n INT;
declare teste VARCHAR(45);
SET n =(SELECT COUNT(*) From Nacionalidade);
set teste = (select n.designacao from Nacionalidade n where n.designacao =
designacao);
IF (teste is null) then
INSERT INTO Nacionalidade(id,designacao)
Values(n+1,designacao);
END IF;
END //
DELIMITER //

```

8.7 Anexo 7 - Requisitos de Exploração

```
USE `FuteProject`;
```

```
SELECT *  
FROM Jogador;
```

-- Query 11 - Obter número de jogadores agenciados

```
drop procedure if exists nr_agenciados;
```

```
DELIMITER //
```

```
create procedure nr_agenciados()
```

```
Begin
```

```
SELECT COUNT(id) FROM Jogador
```

```
WHERE Tipo_id=1;
```

```
end //
```

```
DELIMITER //
```

```
call nr_agenciados();
```

-- 12. Obter número de jogadores de uma determinada nacionalidade.

```
drop procedure if exists jogadores_nacionalidade;
```

```
DELIMITER //
```

```
create procedure jogadores_nacionalidade(in nac VARCHAR(45))
```

```
begin
```

```
SELECT * From Nacionalidade n INNER JOIN Jogador j on n.id=j.Nacionalidade_id
```

```
where n.designacao = nac;
```

```
end //
```

```
DELIMITER //
```

```
call jogadores_nacionalidade("Português");
```

-- 13. Consultar jogadores por posição.

```
drop procedure if exists jogadores_posicao;
```

```
DELIMITER //
```

```
create procedure jogadores_posicao(in pos VARCHAR(45))
```

```
begin
```

```
SELECT * FROM Posicao p INNER JOIN Jogador j on j.Posicao_id = p.id
```

```
where p.designacao=pos;
```

```
end //
```

```
DELIMITER ;
```

```
call jogadores_posicao("Extremo Direito");
```

-- 14. Obter top 3 de agentes com mais jogadores associados.

```
drop procedure if exists top3Agentes;
```

```
DELIMITER //
```

```
create procedure top3Agentes()
```

```
Begin
```

```
SELECT Count(j.Agente_id),a.nome
```

```
From Jogador j INNER JOIN Agente a on a.id = j.Agente_id
```

```
Group by (j.Agente_id)
```

```

Order by (Count(j.Agente_id)) DESC
Limit 3;
end //
DELIMITER //

```

```

call top3Agentes();

```

```

-- 15.  Obter top 3 de jogadores com mais golos marcados.
SET GLOBAL log_bin_trust_function_creators = 1;
Drop function if exists soma_golos_l;
DELIMITER //
CREATE FUNCTION soma_golos_l(jog VARCHAR(45)) RETURNS INT
BEGIN
DECLARE golos INT;
SET golos = (SELECT SUM(h.golos) FROM Jogador j inner join Historico_clubes h
on j.id = h.Jogador_id WHERE j.nome = jog);
RETURN golos;
END //
DELIMITER //

```

```

drop procedure if exists top3Golos;
DELIMITER //
create procedure top3Golos()
Begin
SELECT j.nome, soma_golos_l(j.nome)+j.golos_marcados as Golos_Totais From
Jogador j
Where j.Tipo_id=1
Group by (j.id)
Order by (Golos_Totais) desc
Limit 3;
end //
DELIMITER //
call top3Golos();

```

```

-- 16.  Obter top 3 de jogadores com maior reputacao.

```

```

drop procedure if exists top3Reputacao;
DELIMITER //
create procedure top3Reputacao()
begin
SELECT * From Jogador j
Where j.Tipo_id=1
Order by (j.reputacao) desc
Limit 3;
end //
DELIMITER //

```

```

call top3Reputacao();

```

```

-- 17.  Obter top 3 de jogadores com maior valor de mercado.

```

```

drop procedure if exists top3Valor;
DELIMITER //
create procedure top3Valor()
begin

```

```

SELECT * From Jogador j
Where j.Tipo_id=1
Order by (j.valor_mercado) desc
Limit 3;
end //
DELIMITER //

```

```

call top3Valor();

```

-- 18. Obter lista de jogadores que jogaram num determinado clube.

```

drop procedure if exists historico_clube;
DELIMITER //
create procedure historico_clube (in clube VARCHAR(45))
begin
SELECT nome, ca.designacao, id, ca.inicio_contrato from historico_clubes_clube
ca, clube c
where c.designacao = clube and ca.designacao = c.designacao
order by ca.inicio_contrato;
end //
DELIMITER //
call historico_clube("Real Madrid");

```

-- 19. Consultar número de golos de um determinado jogador no clube atual.

```

drop procedure if exists golos_jogador;
DELIMITER //
create procedure golos_jogador (in jog VARCHAR(45))
begin
Select j.nome, j.golos_marcados From Jogador j
Where j.nome = jog;
end//
DELIMITER //
Call golos_jogador("Cristiano Ronaldo");

```

-- 20. Consultar agente de um determinado jogador.

```

drop procedure if exists agente_jogador;
DELIMITER //
create procedure agente_jogador (in jog VARCHAR(45))
begin
Select j.nome, a.nome , a.id From Jogador j INNER JOIN Agente a on a.id =
j.Agente_id
where j.nome = jog;
end//
DELIMITER //
Call agente_jogador("Cristiano Ronaldo");

```

-- 21. Consultar telefone de agente de um determinado jogador.

```

drop procedure if exists telefone_agente_jogador;
DELIMITER //
create procedure telefone_agente_jogador (in jog VARCHAR(45))
begin
Select j.nome, a.nome , a.telefone From Jogador j INNER JOIN Agente a on a.id =
j.Agente_id

```

```

where j.nome = jog;
end//
DELIMITER //
Call telefone_agente_jogador("Cristiano Ronaldo");

```

-- 22. Dado um determinado olheiro, obter lista de reputações de jogadores observados por ele.

```

drop procedure if exists olheiro_jogador_reputacao;
DELIMITER //
create procedure olheiro_jogador_reputacao (in o VARCHAR(45))
begin
Select j.nome as Nome , j.reputacao as Reputacao From Jogador j
INNER JOIN Olheiro ol on ol.id=j.Olheiro_id
Where ol.nome=o;
end//
DELIMITER //
Call olheiro_jogador_reputacao ("Aurélio Pereira");

```

-- 23. Obter o top 5 de jovens promessas mais novas.

```

SET GLOBAL log_bin_trust_function_creators = 1;
DELIMITER //
CREATE FUNCTION idade (dta date) RETURNS INT
BEGIN
RETURN TIMESTAMPDIFF(MONTH, dta, CURDATE());
END //
DELIMITER //

```

```

drop procedure if exists top5MaisJovens;
DELIMITER //
create procedure top5MaisJovens()
begin
Select * From Jogador j
Where j.Tipo_id=2
Order by idade(j.data_nascimento)
Limit 5;
end //
DELIMITER //

```

```

call top5MaisJovens();

```

-- 24. Obter o top 5 de melhores jogadores sem clube.

```

drop procedure if exists top5SemClube;
DELIMITER //
create procedure top5SemClube()
begin
Select * From Jogador j
Where j.Clube_id = null
Order by j.reputacao
Limit 5;

```

```

end //
DELIMITER //

call top5SemClube();

-- 25 - Obter jogadores de uma posição abaixo de um determinado preço.

drop procedure if exists jogador_abaixo_valor;
DELIMITER //
create procedure jogador_abaixo_valor (in valor INT,in pos VARCHAR(45))
begin
Select * From Jogador j INNER JOIN Posicao p on j.posicao_id = p.id
Where j.valor_mercado<= valor and p.designacao=pos
Order by j.valor_mercado desc;
end//
DELIMITER //
Call jogador_abaixo_valor ("80", "Ponta de Lança");

-- 26 acabar contrato

Drop function if exists mesescontratoj;
DELIMITER //
CREATE FUNCTION mesescontratoj (data date) RETURNS INT
BEGIN
RETURN TIMESTAMPDIF(MONTH, CURDATE(), data);
END //
DELIMITER //

Drop procedure if exists meses_contrato;
DELIMITER //
create procedure meses_contrato (in n VARCHAR(45))
BEGIN
Select j.nome , mesescontratoj(j.fim_contrato) as Meses_de_contrato From
Jogador j
INNER JOIN Agente a on a.id=j.Agente_id
Where a.nome=n;
End //
DELIMITER //
Call meses_contrato("Jorge Mendes");

-- 27-Devolver o historico de clubes e o clube atual para ver a reputação

Drop procedure if exists evolucao_reputacao;
DELIMITER //
Create procedure evolucao_reputacao (in jog VARCHAR(45))
Begin
SELECT * FROM historico_reputacao_jogador h
Where h.nome=jog
ORDER BY inicio_contrato;
END //
DELIMITER //
call evolucao_reputacao("Cristiano Ronaldo");

```


-- 28 Golos marcados

```
Drop function if exists soma_golos;
DELIMITER //
CREATE FUNCTION soma_golos(jog VARCHAR(45)) RETURNS INT
BEGIN
DECLARE golos INT;
SET golos = (SELECT SUM(h.golos) FROM Jogador j inner join Historico_clubes h
on j.id = h.Jogador_id WHERE j.nome = jog);
RETURN golos;
END //
DELIMITER //
```

```
Drop procedure if exists golos_jogador;
DELIMITER //
Create procedure golos_jogador(in jog VARCHAR(45))
Begin
Select j.nome, soma_golos(jog) + j.golos_marcados as Golos
From Jogador j
Where j.nome= jog;
-- GROUP BY j.nome;
END //
DELIMITER //
call golos_jogador("Cristiano Ronaldo");
```

-- 29

```
Drop procedure if exists golos_clube;
DELIMITER //
Create procedure golos_clube (in jog VARCHAR(45), in equip VARCHAR(45))
Begin
Select j.nome,ca.designacao as Equipa , h.golos
From Jogador j INNER JOIN Historico_clubes h on j.id=h.Jogador_id
INNER JOIN Clube_anterior ca on h.Clube_anterior_id=ca.id
WHERE j.nome = jog and ca.designacao = equip;
END //
DELIMITER //
call golos_clube("Cristiano Ronaldo","Manchester United");
```

-- 30 Histórico de clubes de um jogador

```
Drop procedure if exists historicoClubesJogador;
DELIMITER //
CREATE PROCEDURE historicoClubesJogador ( IN designacao VARCHAR(128))
Begin
Select Clube_anterior, ca.inicio_contrato, ca.fim_contrato from
historico_clubes_jogador ca, Jogador j
where j.nome = designacao and j.nome = ca.nome
Order by ca.inicio_contrato;
END //
DELIMITER //
```

```
call historicoClubesJogador("Cristiano Ronaldo");
```

-- 31 Jogadores agenciados por um agente

```
Drop procedure if exists jogadoresAgente;  
DELIMITER //  
CREATE PROCEDURE jogadoresAgente (IN designacao VARCHAR(45))  
Begin  
Select j.nome as Nome from Jogador j INNER JOIN Agente a on j.Agente_id = a.id  
where a.nome = designacao;  
END //  
DELIMITER //  
  
call jogadoresAgente("Jorge Mendes");
```

