



Universidade do Minho

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Letivo de 2019/2020

Agendamento e Realização de Testes Clínicos de Atletas de Atletismo de diferentes modalidades e categorias

Ana Afonso A85762, Hugo Faria A81283,
João Mota A80791, Pedro Silva A82522

janeiro 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Agendamento e Realização de Testes Clínicos de Atletas de Atletismo de diferentes modalidades e categorias

Ana Afonso A85762, Hugo Faria A81283,
João Mota A80791, Pedro Silva A82522

janeiro 2020

Resumo

Este projeto realiza-se no âmbito da unidade curricular Base de Dados (BD), no qual foi proposto o desenvolvimento de um sistema de agendamento e realização de testes clínicos de atletas consoante diversos fatores, nomeadamente a modalidade, categoria, clube, tipo de teste clínico a realizar, entre outros considerados relevantes para a temática abordada.

Uma BD tem como objetivo saciar a necessidade de recolha e organização de informações, podendo armazenar vários dados. Os SGBD são aplicações informáticas desenvolvidas para armazenar e gerir bases de dados, onde o utilizador tem a possibilidade de controlar o acesso aos dados com relativa facilidade, sejam estes simples ou complexos.

Aliando os conceitos teóricos consolidados ao senso comum, estabeleceram-se certos requisitos que se integram numa implementação em base de dados. Desta forma, criou-se um modelo que permite agendar e realizar testes clínicos, não autorizando a sobreposição de agendamentos para um mesmo profissional de saúde, possibilita ainda a consulta ao histórico clínico de cada utente, existindo assim um maior controlo sobre cada atleta de um determinado clube.

O projeto foi dividido em duas etapas, numa primeira etapa onde é dada primazia ao modelo relacional, caracterizado pela sua escalabilidade vertical e pelas propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), tirando partido das suas vantagens nomeadamente a portabilidade entre sistemas, a manipulação dinâmica dos dados e as múltiplas visões dos mesmos. Numa segunda etapa, utilizou-se o modelo não relacional, uma vez que este apresenta outro tipo de aspetos vantajosos tais como, maior flexibilidade (consequentemente mais simples de gerir), escalabilidade horizontal (melhor performance) e o facto de os dados serem armazenados como documentos ou grafos, por exemplo.

O trabalho concebido suporta um modelo de fácil acesso a dados clínicos sendo capaz de ser utilizado por qualquer utilizador, uma vez que foi gerado com o cuidado de ser perceptível independentemente de quem o usa.

Área de Aplicação: Arquitetura de um Sistema de Bases de Dados utilizado em Unidades de Saúde com principal foco no agendamento e realização de testes clínicos de atletas de diversas modalidades e categorias.

Palavras-Chave: Bases de Dados, SQL, *MySQL Workbench*, NoSQL, Neo4j,.

ÍNDICE

Resumo	3
1. Introdução	8
1.1. Contextualização	8
1.2. Apresentação do Caso de Estudo	8
1.3. Motivação e Objetivos	9
1.4. Estrutura do Relatório	9
2. Levantamento e Análise de Requisitos	10
2.1. Método de Levantamento e Análise de Requisitos Adotado	10
2.2. Requisitos Levantados	10
2.2.1 Requisitos de Descrição	10
2.2.2 Requisitos de Exploração	11
2.2.3 Requisitos de Controlo	11
3. Evolução da Base de Dados	12
3.1. Modelo Conceptual	12
3.1.1 Análise de Entidades	13
3.1.2 Relacionamentos	14
3.1.3 Validação do Modelo Conceptual	14
3.2. Modelo Lógico	16
3.2.1 Validação do Modelo através da Normalização	20
3.3. Modelo Físico	22
3.3.1 Seleção do Sistema de Gestão de Bases de Dados	22
3.3.2 Transição do Modelo Lógico para o Modelo Físico	22
3.3.3 Tradução das Interrogações do utilizador para SQL	23
3.3.4 Tradução das Transações do utilizador para SQL	27
3.3.5 Vistas	30
3.3.6 Estimativa do espaço em disco da BD e taxa de crescimento anual	31

3.3.7 Definição e Caracterização dos mecanismos de segurança em SQL	33
4. NoSQL	34
4.1. <i>Justificação da utilização de um Sistema NoSQL</i>	34
4.2. <i>Identificação e descrição dos objetivos da Base de Dados</i>	35
4.3. <i>Identificação e explicação do tipo de questões que serão realizadas</i>	35
4.4. <i>Estrutura base para o sistema de dados NoSQL</i>	36
4.5. <i>Identificação dos objetos de dados no sistema NoSQL</i>	36
4.6. <i>Processo de migração de dados</i>	36
4.6.1 Explicação do processo de migração de dados	36
4.6.2 Implementação do processo de migração de dados	37
4.7. <i>Apresentação das questões realizadas</i>	39
5. Conclusões e Trabalho Futuro	41
6. Referências	42
7. Lista de Siglas e Acrónimos	43
Anexos	44
I. Anexo 1 – Código DDL	44
II. Anexo 2 – Povoamento da BD	47
III. Anexo 3 – Tabelas povoadas	51
IV. Anexo 4 – Código das interrogações e transações	54
V. Anexo 5 – Script migração de dados NoSQL	65

ÍNDICE DE FIGURAS

FIGURA 1: MODELO CONCEPTUAL (DIAGRAMA E-R) DA BD	12
FIGURA 2: MODELO LÓGICO	16
FIGURA 3: EXEMPLO DOS ATRIBUTOS DE UMA TABELA	22
FIGURA 4: TABELAS OBTIDAS NO MODELO FÍSICO	22
FIGURA 5: REPRESENTAÇÃO PARCIAL DO MODELO OBTIDO EM NEO4J	39

ÍNDICE DE TABELAS

TABELA 1: TABELA ATLETA - MODELO LÓGICO	17
TABELA 2: TABELA CATEGORIA - MODELO LÓGICO	17
TABELA 3: TABELA MODALIDADE - MODELO LÓGICO	18
TABELA 4: TABELA ATLETA_HAS_MODALIDADE - MODELO LÓGICO	18
TABELA 5: TABELA PROFISSIONAL_DE_SAUDE - MODELO LÓGICO	19
TABELA 6: TABELA ESPECIALIDADE - MODELO LÓGICO	19
TABELA 7: TABELA TESTE_CLINICO - MODELO LÓGICO	20
TABELA 8: TAMANHO MÉDIO DE CADA REGISTO EM BYTES	31
TABELA 9: NÚMERO DE TUPLOS DE CADA TABELA	31
TABELA 10: TABELA PERMISSÕES UTILIZADOR	33
TABELA 11: TABELA ATLETA POVOADA	51
TABELA 12: TABELA PROFISSIONAL_DE_SAUDE POVOADA	51
TABELA 13: TABELA CATEGORIA POVOADA	52
TABELA 14: TABELA MODALIDADE POVOADA	52
TABELA 15: TABELA ESPECIALIDADE POVOADA	52
TABELA 16: TABELA TESTE_CLINICO POVOADA	53
TABELA 17: TABELA ATLETA_HAS_MODALIDADE POVOADA	53

1. Introdução

1.1. Contextualização

A intemporalidade subjacente à palavra prevenção espelha-se no ditado popular “mais vale prevenir do que remediar”, imbuídos da inevitável procura pela satisfação deste termo, ou seja, no teste periódico do bem-estar do ser humano, encontra-se a necessidade da criação deste projeto. Neste contexto, dá-se primazia à prevenção, no sentido de evitar problemas de saúde futuros, os quais poderiam ser detetados e tratados numa fase precoce.

A elaboração desta BD visa promover a deteção de anomalias respeitantes à saúde de atletas de atletismo, permitindo o agendamento para uma realização à posteriori de testes clínicos, com o intuito de verificar que todos os valores obtidos se encontram dentro dos parâmetros tabelados como normais. Este projeto permite ainda ficar a conhecer o estado clínico de um atleta através dos testes clínicos realizados, a partir dos quais é verificado se o atleta se encontra apto para participar na(s) sua(s) modalidade(s).

1.2. Apresentação do Caso de Estudo

Para a conceção deste projeto, decidiu-se que a BD implementada cumpriria os requisitos imprescindíveis, sendo estes selecionados tendo por base um protótipo de exame médico desportivo fornecido pelo Instituto Português do Desporto e Juventude (IPDJ).

Desde a escolha dos profissionais de saúde habilitados para a realização deste tipo de teste clínico, passando pelas informações básicas que precisam de ser fornecidas até aos exames necessários a efetuar, toda a elaboração suportou etapas que foram sendo deliberadas tendo em consideração serviços de saúde existentes/reais.

1.3. Motivação e Objetivos

Foi principal objetivo deste projeto a organização de um sistema, no contexto da área da saúde e ligado ao desporto, que permitisse a cada profissional de saúde o acesso à informação clínica de um atleta (existente na base de dados criada).

A motivação principal foi a criação de um meio facilitador e simplista para a consulta de dados clínicos sempre que solicitado.

1.4. Estrutura do Relatório

Ao longo do relatório serão apresentados, numa primeira fase, os requisitos levantados após entrevistas realizadas com os utilizadores da BD.

Após o levantamento de todos os requisitos, prosseguiu-se para a elaboração dos modelos Conceptual, Lógico e Físico e validação dos mesmos. Com a validação do modelo físico, foi possível a realização de questões essenciais para o correto funcionamento da BD, bem como a criação de utilizadores capazes de a utilizarem. Aquando das entrevistas com os utilizadores, foram recolhidos dados sobre o tipo de especialidades e testes clínicos necessários para a realização de um exame médico desportivo. Esta informação, previamente validada, foi transposta para a base de dados, constituindo o seu povoamento. É importante realçar que certos dados relativos aos atletas e profissionais de saúde são fictícios.

Foi ainda calculada uma estimativa dos recursos humanos e físicos gastos para a elaboração desta BD.

Concluída a fase de modelação relacional, procedeu-se à conversão para uma modelação não relacional.

Esta conversão permitiu a realização de diferentes questões sobre a base de dados, bem como uma análise crítica sobre as principais divergências entre as duas vertentes.

2. Levantamento e Análise de Requisitos

2.1. Método de Levantamento e Análise de Requisitos Adotado

A fim de desenvolver esta BD da forma mais realista possível, foi agregada informação relativa ao tipo de testes clínicos obrigatórios para realização por parte de um atleta. Deste modo, consultou-se a página do IPDJ onde se averiguou os dados pessoais necessários para o preenchimento do exame médico desportivo [1], bem como os exames a realizar.

Após a obtenção de todas as informações necessárias para a realização de testes clínicos, prosseguiu-se a diálogos com profissionais de saúde, de modo a entender da melhor forma o funcionamento e veracidade dos dados previamente obtidos.

Deste modo, foram discutidos os tipos de especialidades que se adequavam da melhor forma à realização de cada secção do exame médico desportivo.

2.2. Requisitos Levantados

2.2.1 Requisitos de Descrição

- Para que um atleta seja registado, é preciso que haja conhecimento sobre o seu nome, número de cartão de cidadão, data de nascimento, sexo, morada (rua, localidade e código postal), nacionalidade, número de telemóvel, clube de atletismo a que pertence e qual a sua aptidão física;
- Para que um profissional de saúde seja registado, é necessária a informação sobre o seu nome, data de nascimento, morada (rua, localidade e código postal) e especialidade;
- Uma especialidade é constituída por uma designação e um id;
- Uma categoria é caracterizada por uma designação e um id;
- Uma modalidade contém uma designação e um id;

- Um teste clínico dispõe de um número identificador, uma designação, uma descrição, um preço associado, uma data e hora da sua realização, um estado e uma aptidão.

2.2.2 Requisitos de Exploração

- A BD deve permitir o registo de novos atletas e profissionais de saúde;
- Deve permitir a marcação, desmarcação e remarcação de testes clínicos;
- Obter os testes clínicos realizados num determinado dia, por determinado profissional de saúde;
- Calcular o número total de testes clínicos realizados por determinado profissional de saúde;
- Apresentar os testes clínicos agendados para um atleta;
- Calcular a média de idades dos atletas registados;
- Obter a percentagem de testes clínicos realizados por atletas do sexo feminino;
- Obter a percentagem de atletas com aptidão física válida de uma determinada modalidade;
- Apresentar os atletas não aptos;
- Obter a aptidão de um atleta consoante os resultados dos testes realizados;
- Apresentar os testes clínicos que se encontram a decorrer.

2.2.3 Requisitos de Controlo

- A BD não deve permitir que sejam agendados testes clínicos com sobreposição para o mesmo profissional de saúde;
- A BD não deve permitir a inserção de um atleta com sexo indeterminado.

3. Evolução da Base de Dados

3.1. Modelo Conceptual

O primeiro passo para a conceção de uma base de dados é a elaboração de um modelo conceptual. Para isto, foi utilizado um Diagrama E-R (Diagrama Entidades-Relacionamentos), de forma a ser visualmente possível a identificação de entidades, atributos e relacionamentos.

Geralmente, neste tipo de diagramas, as entidades são representadas como sendo retângulos e os atributos, que são as propriedades de cada entidade, como círculos.

Dentro do conjunto de atributos existe no mínimo um que identifica de forma única e distinta cada tuplo de uma relação, denominado de chave primária. No diagrama E-R, este atributo é distinguido de todos os outros apresentando-se a sublinhado.

Além de entidades e atributos, os relacionamentos representam-se através de losangos. Estes são uma parte essencial de um diagrama E-R, pois especificam de que forma duas ou mais entidades estão relacionadas. Os relacionamentos são compostos por uma descrição, geralmente um verbo, que indica o tipo de relacionamento entre entidades ligadas através de linhas contínuas com o dito losango e uma cardinalidade para cada entidade envolvida no relacionamento que indica o número de elementos por relacionamento (1: 1, 1: N, N: N).

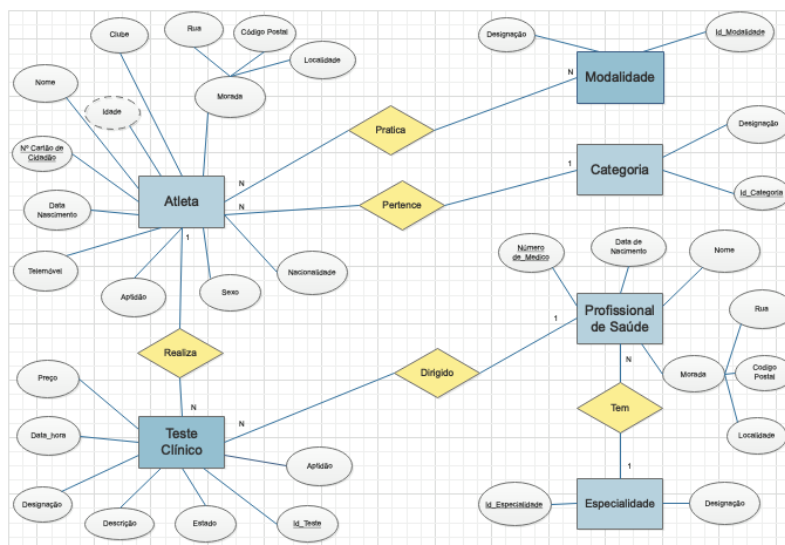


Figura 1: Modelo Conceptual (Diagrama E-R) da BD

3.1.1 Análise de Entidades

Para a elaboração da BD, foram consideradas seis entidades, sendo estas: Atleta, Modalidade, Categoria, Teste Clínico, Profissional de Saúde e Especialidade.

Relativamente à entidade Atleta, decidiu-se utilizar o número de Cartão de Cidadão do mesmo para ser a chave primária. Esta decisão esteve em concordância com a escolha dos atributos necessários para esta entidade, dado que foi tomado por base o exame médico desportivo do IPDJ [1] , que contém este parâmetro como informação do atleta e visto que se trata de um elemento único que identifica inequivocamente um indivíduo, evidenciou-se como chave candidata, acabando por se tornar chave primária.

Quanto à Modalidade, Categoria e Especialidade atribuiu-se um id para identificar claramente o seu tipo e se tornar a chave primária, apresentando também como atributo a designação da modalidade, categoria e especialidade, respetivamente.

Relativamente a Teste Clínico, atribuiu-se um número para cada teste que entra na BD para que este fosse devidamente identificado. Além do referido atributo definido como chave primária de um teste clínico, este contém um preço base (atribuído consoante a especialidade do mesmo), a data e hora que cada teste se encontra agendado, a designação do teste clínico, uma breve descrição que poderá ser adicionada no ato de conclusão deste caso seja necessário deixar alguma anotação, um estado que permite saber se uma consulta está marcada, a decorrer ou concluída e uma aptidão que indica o sucesso ou insucesso do teste clínico.

Por fim, um profissional de saúde é identificado através do seu número interno e apresenta como atributos, salvo a chave primária já referida, o seu nome, data de nascimento e morada.

3.1.2 Relacionamentos

Às demais entidades e seus respectivos atributos é essencial atribuir relacionamentos entre estas.

Entre Atleta e Teste Clínico, atribuiu-se um relacionamento de 1:N, sendo que um atleta pode realizar vários testes clínicos e cada teste tem apenas um atleta associado.

Além disso, a entidade Atleta está ligada a Modalidade com um relacionamento de N:N, o que significa que um atleta pode praticar várias modalidades e cada modalidade tem vários atletas a praticá-la. Já o relacionamento entre Atleta e Categoria é do tipo 1: N.

Quanto a Teste Clínico, este também tem um relacionamento com Profissional de Saúde do tipo N:1, sendo que um profissional de saúde pode dirigir vários testes, mas um teste clínico é apenas dirigido por um profissional de saúde.

Por sua vez, Profissional de Saúde também tem um relacionamento com Especialidade, este do tipo N:1, em que um profissional de saúde tem uma especialidade e cada especialidade tem vários profissionais de saúde.

3.1.3 Validação do Modelo Conceptual

Após a concepção deste modelo é crucial passar pelo ponto da validação, no qual se verifica a possibilidade de se efetuarem transações e interrogações, mostrando assim que o modelo conceptual é válido.

A fim de assegurar a validade do modelo utilizou-se “Descrição da Transição” selecionada das formas existentes para a validação de modelos. Desta forma, verifica-se que toda a informação dada por entidades, atributos e respectivos relacionamentos é abordada pelo /encontra-se no modelo.

Apresentam-se alguns exemplos de transações e interrogações concretizadas pelo modelo:

- **Inserir um profissional de saúde:** Para a inserção de um novo profissional de saúde na BD, é necessária a introdução do seu nome, data de nascimento, morada (rua, código postal, localidade) e a sua

especialidade. Uma vez que estes parâmetros se encontram como atributos de Profissional de Saúde no modelo concebido, é possível concretizar a transação.

- **Determinar o número total de testes clínicos realizados por um determinado profissional de saúde:** Tendo em conta a existência de um relacionamento entre Profissional de Saúde e Teste Clínico, sendo que um teste clínico é dirigido por um profissional de saúde, a interrogação é passível de ser realizada.
- **Marcar um teste clínico:** Para a marcação de um teste clínico é crucial a indicação do atleta que o realiza, bem como do profissional de saúde que o dirige e da data em que este irá ser realizado. Sabendo que existe um relacionamento de 1:N entre Atleta e Teste Clínico e entre Profissional de Saúde e Teste Clínico, a transação é, de igual modo, possível.
- **Calcular a percentagem de atletas de uma modalidade com aptidão física válida:** Dado que existe um relacionamento de N:N de Atleta para Modalidade, a interrogação é válida.
- **Calcular o número de atletas de uma dada categoria:** Uma vez que se observa um relacionamento 1:N entre Atleta e Categoria, esta interrogação é também válida.

Terminada a verificação da validação do modelo conceptual, poder-se-á prosseguir para a realização do modelo lógico.

3.2. Modelo Lógico

O modelo lógico é um dos modelos fundamentais para a conceção do projeto. Este resulta de processos de normalização, através dos quais se obtém um conjunto de tabelas que permite um adequado registo dos dados, que vão de encontro aos requisitos/funcionalidades pretendidas dos utilizadores. Este modelo encontra-se abaixo representado na figura 2.

Cada tabela apresenta um atributo que atua como chave primária e um número variável de atributos que servirão como chave estrangeira (sendo estas chaves primárias noutras tabelas). Cada atributo é ainda identificado pelo tamanho e tipo de dados.

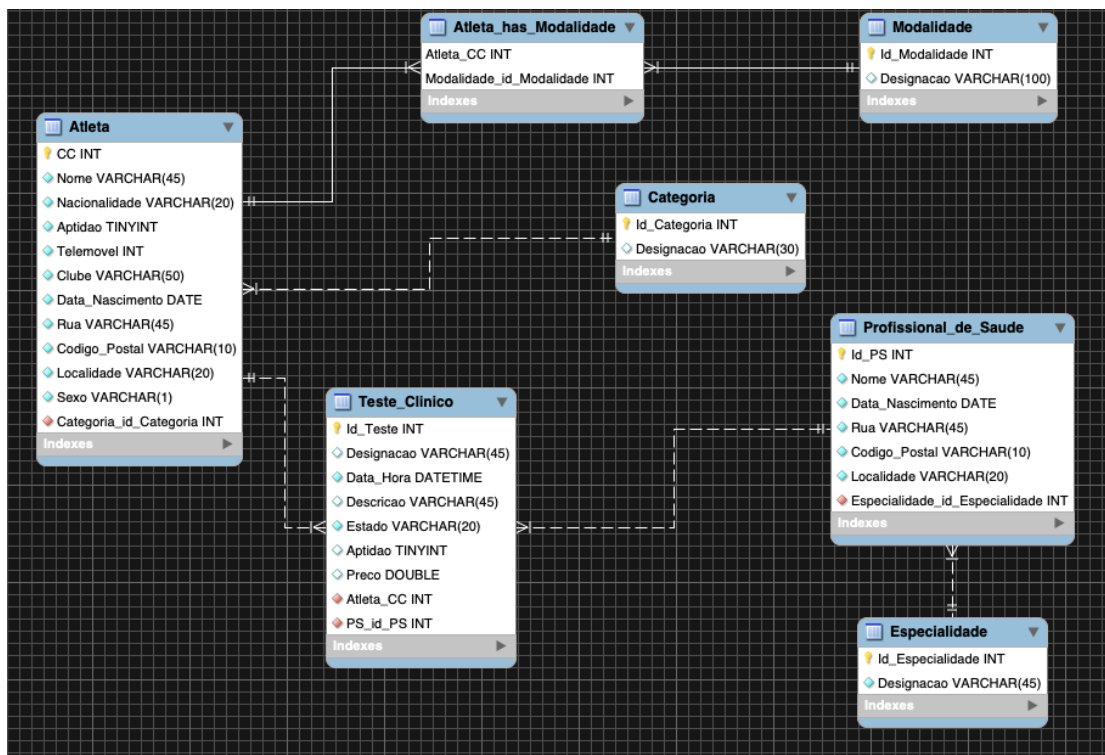


Figura 2: Modelo lógico

De seguida, irá ser apresentada uma breve descrição acerca de cada tabela, bem como os atributos que atuam como chave primária ou chave estrangeira.

Tabela Atleta

Nesta tabela encontram-se especificados os atletas registados na BD. Cada um é identificado através dos seguintes atributos: CC (número do cartão de cidadão, utilizado como identificador único do atleta), Nome, Nacionalidade, Aptidão (*tinyint* que assume o valor de 1 se o atleta estiver apto – isto é, todos os testes clínicos efetuados estarem dentro dos valores considerados normais – ou 0, caso contrário), Telemovel, Clube, Data_Nascimento, Rua, Codigo_Postal, Localidade, Sexo, Categoria_id_Categoria. Este último é a chave estrangeira relativa à categoria da qual faz parte o atleta.

Tabela Atleta		
Atributos	Chave Primária	Chave Estrangeira
CC	✓	
Nome		
Nacionalidade		
Aptidao		
Telemovel		
Clube		
Data_Nascimento		
Rua		
Codigo_Postal		
Localidade		
Sexo		
Categoria_id_Categoria		✓

Tabela 1: Tabela Atleta - Modelo Lógico

Tabela Categoria

Nesta tabela estão apresentadas as categorias subjacentes a cada atleta, como tal os atributos são: Id_Categoria (identificador único da categoria) e Designacao (nome da categoria).

Tabela Categoria		
Atributos	Chave Primária	Chave Estrangeira
Id_Categoria	✓	
Designacao		

Tabela 2: Tabela Categoria - Modelo Lógico

Tabela Modalidade

Esta tabela contém todas as modalidades possíveis de serem praticadas pelos atletas. Assim, os atributos são: Id_Modalidade (identificador único da modalidade) e Designacao (nome da modalidade).

Tabela Modalidade		
Atributos	Chave Primária	Chave Estrangeira
Id_Modalidade	✓	
Designacao		

Tabela 3: Tabela Modalidade - Modelo Lógico

Tabela Atleta_has_Modalidade

Nesta tabela encontram-se representadas as modalidades dos atletas registados. Consequentemente, os atributos são: Atleta_CC (identificador único do atleta) e Modalidade_id_Modalidade (identificador único da modalidade do respetivo atleta). A criação desta tabela deve-se à existência de um relacionamento N:N entre Atleta e Modalidade. Deste modo, a nova tabela criada tem uma chave primária composta pelas chaves primárias de Atleta e Modalidade.

Tabela Atleta_has_Modalidade		
Atributos	Chave Primária	Chave Estrangeira
Atleta_CC	✓	
Modalidade_id_Modalidade	✓	

Tabela 4: Tabela Atleta_has_Modalidade - Modelo Lógico

Tabela Profissional_de_Saude

Nesta tabela encontram-se explícitas informações acerca de um profissional de saúde, apresentando como atributos Id_PS (chave primária do profissional de saúde), Nome, Data_Nascimento, Rua, Codigo_Postal, Localidade, Especialidade_id_Especialidade (especialidade relativa a um determinado profissional de saúde).

Tabela Profissional_de_Saude		
Atributos	Chave Primária	Chave Estrangeira
Id_PS	✓	
Nome		
Data_Nascimento		
Rua		
Codigo_Postal		
Localidade		
Especialidade_id_Especialidade		✓

Tabela 5: Tabela Profissional_de_Saude - Modelo Lógico

Tabela Especialidade

Nesta tabela estão todas as especialidades que vão de encontro ao necessário para a efetuação dos testes clínicos necessários. Os atributos são: *Id_Especialidade* (identificador único da especialidade) e *Designacao* (nome da especialidade).

Tabela Especialidade		
Atributos	Chave Primária	Chave Estrangeira
Id_Especialidade	✓	
Designacao		

Tabela 6: Tabela Especialidade - Modelo Lógico

Tabela Teste_Clinico

Nesta tabela são apresentados os atributos Id_Testes (identificador único de um teste), Designacao (nome dos procedimentos que compõem um teste), Preco (preço por especialidade), data_hora (data e hora da realização do teste), Descricao (breve descrição acerca do efetuado), Estado (este pode ser um de quatro estados: marcado, desmarcado, a decorrer e concluído), Aptidao (1- caso o atleta esteja apto ou 0 – caso contrário), Atleta_CC (identificador único do atleta), PS_id_PS (identificador único do profissional de saúde) que descrevem um conjunto de informações existentes num teste clínico. Estes últimos atuam como chaves estrangeiras.

Tabela Teste_Clinico		
Atributos	Chave Primária	Chave Estrangeira
Id_Testes	✓	
Designacao		
Preco		
data_hora		
Descricao		
Estado		
Aptidão		
Atleta_CC		✓
PS_id_PS		✓

Tabela 7: Tabela Teste_Clinico - Modelo Lógico

3.2.1 Validação do Modelo através da Normalização

No seguimento da elaboração do modelo lógico é necessária a validação do mesmo através da normalização, de modo a garantir a inexistência de anomalias nas relações, através das formas normais propostas por Codd em 1970. As formas normais definem os tipos de dependências funcionais permitidas nas relações, e constituem etapas sucessivas que se podem aplicar gradualmente para eliminar as anomalias de inserção e modificação [3].

Para garantir a primeira forma normal (1FN) é necessário que todos os valores de todos os atributos sejam atômicos, de modo a assegurar a simplicidade dos valores.

A conversão dos atributos multivalorados e compostos, como é o caso de morada, em atributos simples aquando da conversão do modelo conceptual para o modelo lógico é essencial para a validação da 1FN. Assim, os atributos que compõem a morada são separados e apenas estes se encontram presentes na tabela correspondente.

Tendo em conta o exemplo do relacionamento N:N entre Modalidade e Atleta, como foi referido anteriormente, houve a necessidade da criação de uma nova relação, à qual foi atribuído o nome Atleta_has_Modalidade. No caso da inexistência da mesma, cada tuplo da relação Atleta teria de conter todas as Modalidades que o atleta em questão pratica e cada tuplo de Modalidade teria de conter a identificação de todos os atletas que a praticam o que, segundo a 1FN, não pode acontecer. Com a criação da relação Atleta_has_Modalidade, cada tuplo apenas contém a informação de um atleta e respetiva modalidade singularmente, podendo haver repetições de ambos em tuplos diferentes. Desta forma, o problema referido não se verifica.

Para a verificação da segunda forma normal (2FN), é necessário a prévia verificação da 1FN. Uma relação está na 2FN se não existirem dependências parciais, isto é, todos os atributos não candidatos são, a nível funcional, totalmente dependentes da chave candidata, o que se verifica [7].

Por fim, é indispensável a verificação da terceira forma normal (3FN). Uma relação está na 3FN se estiver na 2FN e se todos os atributos transitivamente dependentes numa chave não forem chaves primárias, o que se confirma.

Com a validação das três formas normais apresentadas é garantido que não existe redundância dos dados.

3.3. Modelo Físico

3.3.1 Seleção do Sistema de Gestão de Bases de Dados

Uma base de dados é uma coleção organizada de informação estruturada relacionada entre si que persiste durante um determinado período de tempo. É gerida por um Sistema de Gestão de Bases de Dados (SGBD), que condiciona a forma como se cria, acede e elimina informação [3].

Um SGBD deve permitir definir a estrutura de dados mais adaptada ao problema e permissões de acesso aos dados, permitir a consulta eficiente de dados, permitir a recuperação da BD e permitir operações simultâneas sobre os dados por parte de vários utilizadores.

Para a realização deste projeto, o SGBD selecionado foi o *MySQL*, que utiliza a linguagem SQL (*Structured Query Language*). A escolha deste SGBD deveu-se não só à sugestão por parte dos docentes, bem como pelo seu desempenho e estabilidade, portabilidade, facilidade de manuseamento, entre outros.

3.3.2 Transição do Modelo Lógico para o Modelo Físico

A implementação do modelo físico foi obtida a partir do modelo lógico construído na fase anterior, onde foram criadas sete tabelas (Figura 4), cada uma correspondente a uma entidade, em que cada coluna corresponde a um atributo dessa entidade (Figura 3). O código da criação destas tabelas encontra-se no Anexo 1 – Código DDL.

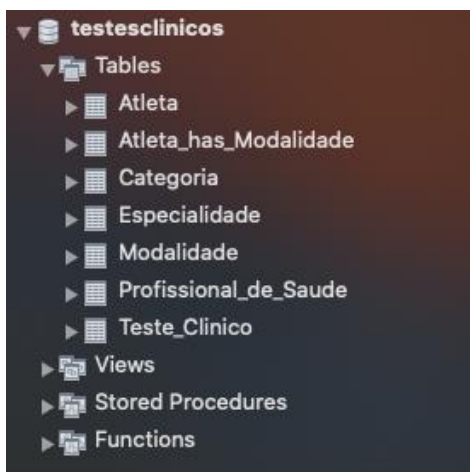


Figura 4: Tabelas obtidas no modelo físico

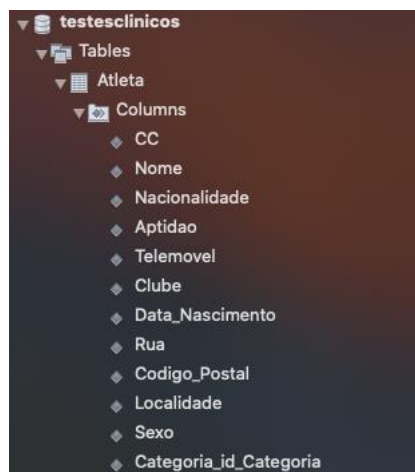


Figura 3: Exemplo dos atributos de uma tabela

3.3.3 Tradução das Interrogações do utilizador para SQL

3.3.3.1 Procedimentos

Seguem-se alguns dos procedimentos implementados na BD, bem como uma breve explicação dos mesmos. Os restantes procedimentos poderão ser encontrados em anexo.

- nTestesPorMedico

```
DELIMITER //
CREATE PROCEDURE nTestesPorMedico(IN idPS INT)
BEGIN
    SELECT COUNT(*) from Profissional_de_Saude AS PS
    JOIN Teste_Clinico as TC ON TC.PS_Id_PS
    WHERE PS.Id_PS=idPS AND TC.PS_Id_PS=idPS AND
    TC.Estado = 'Concluido';
END //
DELIMITER //
```

Este procedimento permite calcular o número total de testes clínicos realizados por um determinado profissional de saúde.

- medialddTestes

```
DELIMITER //
CREATE PROCEDURE medialddTestes()
BEGIN
    SELECT AVG(idade(Data_Nascimento)) FROM Atleta AS A
    JOIN Teste_Clinico AS TC ON TC.Atleta_CC = A.CC;
END //
DELIMITER //
```

Este procedimento calcula a média de idades dos atletas que realizaram testes.

3.3.3.2 Funções

- percentagemValidoMod

```
DELIMITER //
CREATE FUNCTION percentagemValidoMod(modalidadeP INT) RETURNS DOUBLE
DETERMINISTIC
BEGIN
    DECLARE total INT;
    DECLARE aprovados INT;
    DECLARE percentagem DOUBLE;
    SET total = (SELECT COUNT(*) FROM Atleta
        JOIN Atleta_has_Modalidade AS AhM
        ON AhM.Atleta_CC=Atleta.CC
    WHERE AhM.Modalidade_Id_Modalidade=modalidadeP);
    SET aprovados = (SELECT COUNT(*) FROM Atleta
        JOIN Atleta_has_Modalidade AS AhM
        ON AhM.Atleta_CC=Atleta.CC
        WHERE AhM.Modalidade_Id_Modalidade=modalidadeP
        AND Atleta.Aptidato=1);
    SET percentagem = (aprovados/total)*100;
    RETURN percentagem;
END//
DELIMITER //
```

Esta função permite calcular a percentagem de atletas de uma modalidade com aptidão física válida. Tal é conseguido calculando o número total de atletas dessa modalidade (armazenado na variável “total”) e o número de atletas com aptidão física igual a um (armazenado na variável “aprovados”). Posteriormente, é calculada a percentagem de atletas com aptidão igual a 1 sobre o número total de atletas da dada modalidade.

- tempoValido

```

DELIMITER //
CREATE FUNCTION tempoValido(nData DATETIME, nMedico int)
returns BOOLEAN
    DETERMINISTIC
    BEGIN
        DECLARE res BOOLEAN DEFAULT 1;
        DECLARE num INT DEFAULT 0;
        SET num = (SELECT COUNT(*) FROM Teste_Clinico AS TC
WHERE TC.Ps_Id_PS=nMedico AND
nData>=SUBTIME(TC.data_hora,TIME('00:14:59')) AND
nData<=(ADDTIME(TC.data_hora,TIME('00:14:59')) AND
TC.Estado<>'Desmarcado' AND TC.Estado<>'Concluido');
        IF(num>0) THEN SET res=0;
        END IF;
        RETURN res;
    END//
DELIMITER //

```

Esta função indica se uma determinada data/hora é válida para que um teste clínico seja marcado para um determinado profissional de saúde. Esta função retorna o valor zero (0) no caso de a data fornecida ser inválida por já existir um teste marcado para o dado profissional de saúde num intervalo de quinze minutos. Caso essa data/hora esteja disponível a função retorna um (1).

- ultimoTesteEspecialidade

```

DELIMITER //
CREATE FUNCTION ultimoTesteEspecialidade(idEspecialidade INT, CCA atleta INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE res INT;
    SET res = (SELECT Id_Testes FROM Testes_Clinicos AS TC
    JOIN Profissionais_de_Saude AS PdS
    ON TC.PS_Id_PS=PdS.Id_PS WHERE
    PdS.Especialidade_Id_Especialidade=idEspecialidade
    AND TC.Atleta_CC = CCA atleta AND TC.Estado='Concluido'
    ORDER BY TC.data_hora DESC LIMIT 1);
    RETURN res;
END //
DELIMITER //

```

A função em cima apresentada indica o Id_Testes do último teste clínico realizado por um dado atleta de uma modalidade. De modo a obter este resultado, primeiramente, são selecionados todos os testes clínicos concluídos, em que o profissional de saúde que os dirige tem como especialidade a dada como parâmetro. Seguidamente é selecionado o último teste realizado recorrendo à cláusula “ORDER BY” que permite ordenar, neste caso, por ordem decrescente de “data_hora” os testes clínicos obtidos.

3.3.4 Tradução das Transações do utilizador para SQL

- inserePS

```
DELIMITER //
CREATE PROCEDURE insereAtleta(IN CC INT,IN Nome VARCHAR(45),IN
Nacionalidade VARCHAR(20),IN Telemóvel INT,IN Clube VARCHAR(50),IN
Data_Nascimento DATE,IN Rua VARCHAR(45),IN Codigo_Postal VARCHAR(10),IN
Localidade VARCHAR(20),IN Sexo VARCHAR(1))
BEGIN
    DECLARE nCategoria INT DEFAULT 0;
    DECLARE nldade INT DEFAULT 0;
    DECLARE Error BOOLEAN DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
    START TRANSACTION;
    SET nldade = idade(Data_Nascimento);
    IF (nldade>=7 AND nldade<=9) THEN SET nCategoria=1; END IF;
    IF (nldade>=10 AND nldade<=11) THEN SET nCategoria=2;END IF;
    IF (nldade>=12 AND nldade<=13) THEN SET nCategoria=3;END IF;
    IF (nldade>=14 AND nldade<=15) THEN SET nCategoria=4;END IF;
    IF (nldade>=16 AND nldade<=17) THEN SET nCategoria=5;END IF;
    IF (nldade>=18 AND nldade<=19) THEN SET nCategoria=6;END IF;
    IF (nldade>=20 AND nldade<=34) THEN SET nCategoria=7;END IF;
    IF (nldade>=35) THEN SET nCategoria=8; END IF;
    INSERT INTO Atleta VALUES (CC,Nome,Nacionalidade,0, Telemóvel, Clube,
Data_Nascimento, Rua,Codigo_Postal,Localidade,Sexo,nCategoria);
    IF Error=1 THEN BEGIN ROLLBACK;
        SELECT('Erro a introduzir os dados');
        END;
    ELSE COMMIT;
    END IF;
END //
DELIMITER //
```

O procedimento acima apresentado permite a inserção de um novo Atleta na BD. Para tal, é necessário serem fornecidos o seu nº de cartão de cidadão, nome, nacionalidade, telemóvel, clube, data de nascimento, rua, código postal, localidade e sexo, sendo a sua categoria calculada automaticamente consoante a sua idade, que é por sua vez obtida através da função “idade” presente em anexo. É indispensável, aquando do registo de um atleta, verificar se há algum erro de inserção, como por exemplo, um género inválido. Deste modo, foi criado um trigger, controlGenero, apresentado de seguida, que impede que sejam inseridos géneros que não sejam ‘F’ ou ‘M’.

- controlGenero

```
DELIMITER //
CREATE TRIGGER controlGenero BEFORE INSERT ON Atleta
FOR EACH ROW
BEGIN
    IF(new.Sexo<>'M' AND new.Sexo<>'F')
        THEN SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT='Sexo introduzido Inválido';
    END IF;
END //
DELIMITER //
```

O *trigger* que se apresenta acima impede que no ato de inserção de um Atleta não seja possível a inserção de um sexo que não seja 'M' ou 'F'.

- MarcaTeste

```
DELIMITER //
CREATE PROCEDURE MarcaTeste(IN CC INT, IN dataHora DATETIME,
IN nPS INT)
BEGIN
    DECLARE lastId_Testes INT;
    DECLARE espec INT;
    DECLARE design VARCHAR(45);
    DECLARE Error BOOLEAN DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
    START TRANSACTION;
    SET espec = (SELECT Especialidade_Id_Especialidade
    FROM Profissional_de_Saude WHERE Id_PS=nPS);

    CASE espec
    WHEN 1 THEN SET design = 'Exame Oftalmológico';
    WHEN 2 THEN SET design = 'Exame Ectoscópico';
    WHEN 3 THEN SET design = 'Exame Biométrico';
    WHEN 4 THEN SET design = 'Exame ORL';
    WHEN 5 THEN SET design = 'Exame Estomatológico';
    WHEN 6 THEN SET design = 'Exames Complementares de
Diagnóstico';
    WHEN 7 THEN SET design = 'Exame Génito-Urinário';
    WHEN 8 THEN SET design = 'Exame Génito-Urinário';
    WHEN 9 THEN SET design = 'Exame Córdio-Circulatório e
Respiratório';
```

```

ELSE SET design = "";
END CASE;

SET lastId_Testes = (SELECT Id_Testes FROM Testes_Clinico ORDER BY
Id_Testes DESC LIMIT 1);
INSERT INTO
TESTES_CLINICO(Id_Testes,Designacao,data_hora,Estado,
Atleta_CC,PS_Id_PS) VALUES
(lastId_Testes+1,design,dataHora,'Marcado',CC,nPS);
IF Error=1 THEN BEGIN ROLLBACK;
SELECT('Não é possível agendar Teste Clínico: Possível
sobreposição de datas');
END;
ELSE COMMIT;
END IF;
END //
DELIMITER //

```

O procedimento “MarcaTeste” permite que seja marcado um teste clínico para um dado CC, data/hora e PS. No que diz respeito à designação do teste clínico, este é obtido através da especialidade do PS que dirige o mesmo. Quanto ao Id do teste, este é calculado selecionando o Id do último teste clínico registado, incrementando-o. Os atributos Descricao, Aptidao e Preco encontram-se com valor nulo, pelo que serão inseridos apenas no ato de conclusão desse teste. O estado desse teste passará então a ‘Marcado’.

- controlData

```

DELIMITER //
CREATE TRIGGER controlData BEFORE INSERT ON Testes_Clinico
FOR EACH ROW
BEGIN
IF(new.Estado<>'Concluido' AND
new.data_hora<(CURRENT_TIMESTAMP) OR
tempoValido(new.data_hora,new.PS_Id_PS)=0 )
THEN SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT='Data Indisponível';
END IF;
END //
DELIMITER //

```

O *trigger* acima apresentado serve de controlo para sobreposição de datas no ato de marcação de um teste clínico. No caso de ocorrer a tentativa de marcação de um teste clínico para um profissional de saúde, para uma hora ou num intervalo de quinze minutos de outro teste clínico desse mesmo PS, a marcação não é autorizada.

3.3.5 Vistas

Uma vista pode ser considerada uma tabela derivada da consulta de outras tabelas. Pode ser usada para evitar escrever consultas frequentes, para esconder detalhes sem interesse num determinado contexto e para simplificar a escrita de outras consultas. Permite também modificar tabelas sem alterar o que o utilizador da vista vê [3].

De seguida, encontram-se exemplos de vistas desenvolvidas neste projeto.

- aDecorrer

```
DROP VIEW IF EXISTS aDecorrer;  
CREATE VIEW aDecorrer AS  
    SELECT DISTINCT * FROM Teste_Clinico AS TC  
        WHERE TC.Estado='A decorrer';
```

A vista acima apresentada permite exibir todos os testes clínicos que se encontram a decorrer no momento.

- nAptosView

```
DROP VIEW IF EXISTS nAptosView;  
CREATE VIEW nAptosView AS  
    SELECT DISTINCT Nome AS 'Nome',  
        idade(Data_Nascimento) AS 'Idade',  
        CC AS 'Nº Cartão de Cidadão'  
    FROM Atleta WHERE Aptidao=0;
```

Esta vista será acedida pelo utilizador profissional de saúde, permitindo que este visualize todos os atletas (nome, idade e número de cartão de cidadão) que não apresentam aptidão para a prática de atletismo.

3.3.6 Estimativa do espaço em disco da BD e taxa de crescimento anual

Com o propósito obter uma boa estimativa do espaço em disco ocupado pela BD, considerou-se a dimensão desta sem qualquer povoamento. Assim, o valor inicial para a dimensão da BD é 208KiB.

De seguida, calculou-se o tamanho de todos os atributos de cada tabela, através da documentação do MySQL [4].

Tabela	Tamanho (Bytes)
Atleta	184
Profissional_de_Saude	135
Especialidade	50
Modalidade	105
Categoria	35
Atleta_has_Modalidade	8
Teste_Clinico	142

Tabela 8: Tamanho médio de cada registo em Bytes

Após a inserção do povoamento na BD, obteve-se a seguinte tabela:

Tabela	Número de Tuplos
Atleta	19
Profissional_de_Saude	11
Especialidade	9
Modalidade	16
Categoria	8
Atleta_has_Modalidade	30
Teste_Clinico	56

Tabela 9: Número de tuplos de cada tabela

Com base nas tabelas previamente apresentadas, o tamanho em disco ocupado pela BD após o seu povoamento, é dado por:

$$184*19 + 135*11 + 50*9 + 105*16 + 35*8 + 8*30 + 142*56 \\ = 15583 \text{ bytes} = 16 \text{ Kbytes}$$

No que diz respeito ao crescimento futuro da BD, assume-se que o número de modalidades, categorias e especialidades se mantém inalterado. Segundo dados obtidos no portal da “Pordata, Base de Dados de Portugal Contemporâneo” [5], prevê-se um crescimento do número de atletas a praticar atletismo de cerca de 1000 atletas por ano para a zona urbana onde são realizados os testes clínicos em questão. Esta previsão deve-se ao repentino aumento de atletas federados em Portugal de 2017 para 2018.

Assumindo que cada atleta pratica em média duas modalidades de atletismo, prevê-se de igual forma um crescimento na tabela Atleta_has_Modalidade de 2000 por ano.

Relativamente aos testes clínicos e considerando que cada atleta tem que realizar um exame médico desportivo (constituído por 9 testes clínicos) por ano, o número de testes clínicos realizados aumentará cerca de 9000 por ano.

Por fim, assume-se que o número de profissionais de saúde se manterá quase constante.

Ao fim de um ano, será expectável que a BD tenha a dimensão de

$$\begin{aligned} &1184*19 + 135*11 + 50*9 + 105*16 + 35*8 + 2030*9 + 9142*56 \\ &= 556613\text{bytes} = 0.6\text{Mbytes} \end{aligned}$$

3.3.7 Definição e Caracterização dos mecanismos de segurança em SQL

A BD desenvolvida suporta apenas um utilizador, para além do administrador, que é o profissional de saúde. Foram definidas quais as operações a que este utilizador tem acesso.

Desta forma, o utilizador tem a capacidade de marcar, desmarcar, remarcar, iniciar e concluir um teste clínico e obter a listagem dos atletas sem aptidão física.

Na Tabela 10 estão representadas as permissões do profissional de saúde

Profissional de Saúde				
	Insert	Delete	Select	Update
MarcaTeste	✓		✓	
RemarcaTeste	✓		✓	✓
DesmarcarTeste			✓	✓
IniciaTeste			✓	✓
concluiTeste			✓	✓
naptosview			✓	

Tabela 10: Tabela permissões utilizador

De seguida, é apresentado o código de criação e das permissões do utilizador profissional de saúde

```
DROP USER IF EXISTS 'Profissional_de_Saude' @ 'localhost';
CREATE USER 'Profissional_de_Saude'@'localhost' IDENTIFIED BY '0000';
GRANT EXECUTE ON PROCEDURE testesclnicos.concluiTeste TO 'Profissional_de_Saude' @
'localhost';
GRANT EXECUTE ON PROCEDURE testesclnicos.DesmarcaTeste TO 'Profissional_de_Saude' @
'localhost';
GRANT EXECUTE ON PROCEDURE testesclnicos.IniciaTeste TO 'Profissional_de_Saude' @ 'localhost';
GRANT EXECUTE ON PROCEDURE testesclnicos.MarcaTeste TO 'Profissional_de_Saude' @
'localhost';
GRANT EXECUTE ON PROCEDURE testesclnicos.RemarcaTeste TO 'Profissional_de_Saude' @
'localhost';
GRANT SHOW VIEW ON testesclnicos.naptosview TO 'Profissional_de_Saude' @ 'localhost';
FLUSH PRIVILEGES;
SHOW GRANTS FOR 'Profissional_de_Saude' @ 'localhost';
```

4. NoSQL

4.1. Justificação da utilização de um Sistema NoSQL

Além do modelo relacional apresentado anteriormente, também foi proposta uma abordagem não relacional para a elaboração do tema.

Numa abordagem relacional, são reforçados os princípios ACID (Atomicidade, Consistência, Isolamento, Durabilidade) para preservar a integridade e reduzir anomalias, o que torna os modelos relacionais bastante consistentes.

Por outro lado, este tipo de bases de dados possuem um esquema predefinido, e têm escalabilidade vertical, o que significa que a uma única máquina devem ser adicionados recursos, o que implica um aumento de custos associados à memória.

Já num sistema *NoSQL* (*Not Only SQL*), é adotado um modelo não relacional, que se caracteriza por princípios BASE (Basicamente Disponível, Estado Leve, Eventualmente Consistente), que dá mais ênfase à disponibilidade de dados mesmo em situações de falhas em vez de garantir a consistência dos mesmos [6]. Esta possível inconsistência vai de encontro ao teorema CAP (Consistência, Disponibilidade, Tolerância ao Particionamento) que afirma que o armazenamento de dados distribuído fornece simultaneamente mais de duas das três garantias especificadas. Ao contrário de bases de dados relacionais, estas possuem escalabilidade horizontal, isto é em vez de manter um grande armazenamento de dados, o *NoSQL* reparte os dados por muitos sistemas de armazenamento.

Prevendo um crescimento futuro que a BD poderá tomar é importante ter em consideração a escalabilidade horizontal que o *NoSQL* fornece, pelo que a sua utilização se revela mais apropriada.

4.2. Identificação e descrição dos objetivos da Base de Dados

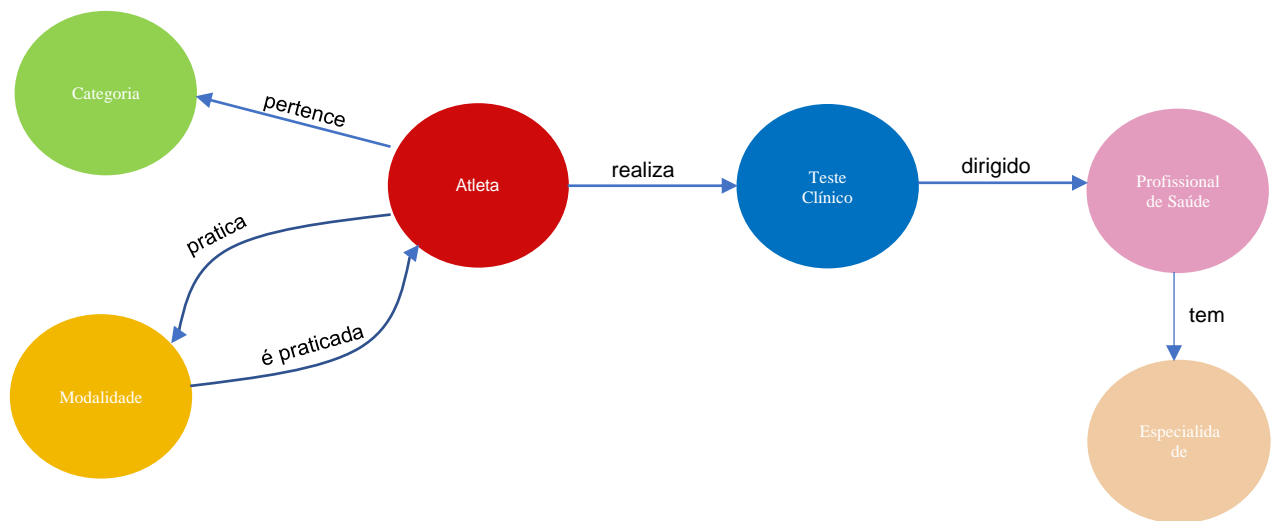
De forma a explorar o sistema *NoSQL*, foi adotado o *Neo4J*, que se caracteriza por ser um sistema de armazenamento de dados em grafos, pelo que torna a visualização de cada relação e seus respetivos relacionamentos mais clara.

4.3. Identificação e explicação do tipo de questões que serão realizadas

Foram escolhidas algumas questões a realizar que sejam capazes de demonstrar as funcionalidades de visualização em grafos do *Neo4J*.

- Questão 1: Apresentar os atletas com aptidão física positiva.
- Questão 2: Apresentar os testes clínicos que obtiveram resultado negativo de todos os atletas: A visualização por parte dos profissionais de saúde dos testes clínicos de um atleta permite o estudo do histórico clínico do mesmo, facilitando as decisões a tomar e os cuidados a ter perante este.
- Questão 3: Apresentar os testes clínicos concluídos e respetivos profissionais de saúde que os dirigem de uma determinada especialidade.
- Questão 4: Apresentar os testes clínicos marcados para um determinado profissional de saúde: A possibilidade de aceder à lista de testes clínicos a realizar por um profissional de saúde permite um conhecimento prévio do mesmo, bem como o controlo do número de consultas agendadas pelos serviços administrativos.
- Questão 5: Apresentar os testes clínicos realizados por um dado atleta, bem como os profissionais de saúde que os dirigiram: O acesso a todos os testes clínicos realizados por um atleta é de extrema importância, uma vez que permite aos profissionais de saúde ter acesso ao histórico clínico do mesmo, contribuindo para um melhor acompanhamento do atleta.

4.4. Estrutura base para o sistema de dados NoSQL



4.5. Identificação dos objetos de dados no sistema NoSQL

Os objetos de dados utilizados no sistema NoSQL foram as tabelas obtidas no modelo físico do MySQL, sendo estas: Atleta, Profissional de Saúde, Modalidade, Categoria, Teste Clínico e Especialidade.

Além dos objetos apresentados, também foram criados nodos relativos a Atleta_has_Modalidade, que resultou de uma relação N:N. Este objeto não se encontra na estrutura base apresentada, uma vez que foi criado apenas com o intuito de estabelecer o relacionamento entre Atleta e Modalidade.

4.6. Processo de migração de dados

4.6.1 Explicação do processo de migração de dados

O primeiro passo para o processo de migração de dados do modelo relacional elaborado anteriormente para o modelo não relacional em Neo4J foi converter todas as tabelas para ficheiros '.csv', através da ferramenta fornecida pelo MySQLWorkbench "*Export Results...*", pelo que estes ficaram guardados na pasta 'import' do projeto criado.

De seguida, foram criados os nodos de cada tabela através dos ficheiros '.csv' previamente obtidos, utilizando a linguagem *Cypher*.

Por fim, foram definidos os relacionamentos entre as entidades criadas no ponto anterior.

4.6.2 Implementação do processo de migração de dados

- Processo de criação de Nodos

- Criação dos nodos relativos a Atleta

```
LOAD CSV WITH HEADERS FROM "file:///atleta.csv" AS LINE
CREATE (a:Atleta)
SET a = LINE;
```

- Criação de nodos relativos a Modalidade

```
LOAD CSV WITH HEADERS FROM "file:///Modalidade.csv" AS LINE
CREATE (m:Modalidade)
SET m = LINE;
```

- Criação de nodos relativos a Atleta_has_Modalidade

```
LOAD CSV WITH HEADERS FROM "file:///Atleta_has_Modalidade.csv" AS LINE
CREATE (am:Atleta_has_Modalidade)
SET am = LINE;
```

- Criação de nodos relativos a Categoria

```
LOAD CSV WITH HEADERS FROM "file:///Categoria.csv" AS LINE
CREATE (c:Categoria)
SET c = LINE;
```

- Criação de nodos relativos a Teste_Clinico

```
LOAD CSV WITH HEADERS FROM "file:///Teste_Clinico.csv" AS LINE
CREATE (tc:Teste_Clinico)
SET tc = LINE;
```

- Criação de nodos relativos a Profissional_de_Saude

```
LOAD CSV WITH HEADERS FROM "file:///Profissional_de_Saude.csv" AS LINE
CREATE (ps:Profissional_de_Saude)
SET ps = LINE;
```

- Criação de nodos relativos a Especialidade

```
LOAD CSV WITH HEADERS FROM "file:///Especialidade.csv" AS LINE
CREATE (e:Especialidade)
SET e = LINE;
```

- Processo de criação de relacionamentos

- Criação dos relacionamentos entre Atleta, Atleta_has_Modalidade e Modalidade

```
MATCH (a:Atleta),(ahm:Atleta_has_Modalidade),(m:Modalidade)
WHERE a.CC=ahm.Atleta_CC AND
ahm.Modalidade_id_Modalidade=m.Id_Modalidade
CREATE (a)-[r:Pratica]->(m), (m)-[r1:E_Praticada]->(a)
RETURN a,m;
```

- Criação do relacionamento entre Atleta e Categoria

```
MATCH (a:Atleta),(c:Categoria)
WHERE a.Categoria_id_Categoria = c.Id_Categoria
CREATE (a)-[r:Pertence]->(c)
RETURN a,c;
```

- Criação do relacionamento entre Atleta e Teste_Clinico

```
MATCH (tc:Teste_Clinico),(a:Atleta)
WHERE tc.Atleta_CC = a.CC
CREATE (a)-[r:Realiza]->(tc)
RETURN tc,a;
```

- Criação do relacionamento entre Teste_Clinico e Profissional_de_Saude

```
MATCH (tc:Teste_Clinico),(ps:Profissional_de_Saude)
WHERE tc.PS_id_PS = ps.Id_PS
CREATE (tc)-[r:Dirigido]->(ps)
RETURN tc,ps;
```

- Criação do relacionamento entre Profissional_de_Saude e Especialidade

```

MATCH (ps:Profissional_de_Saude),(e:Especialidade)
WHERE ps.Especialidade_id_Especialidade = e.Id_Especialidade
CREATE (ps)-[r:Tem]->(e)
RETURN ps,e;

```

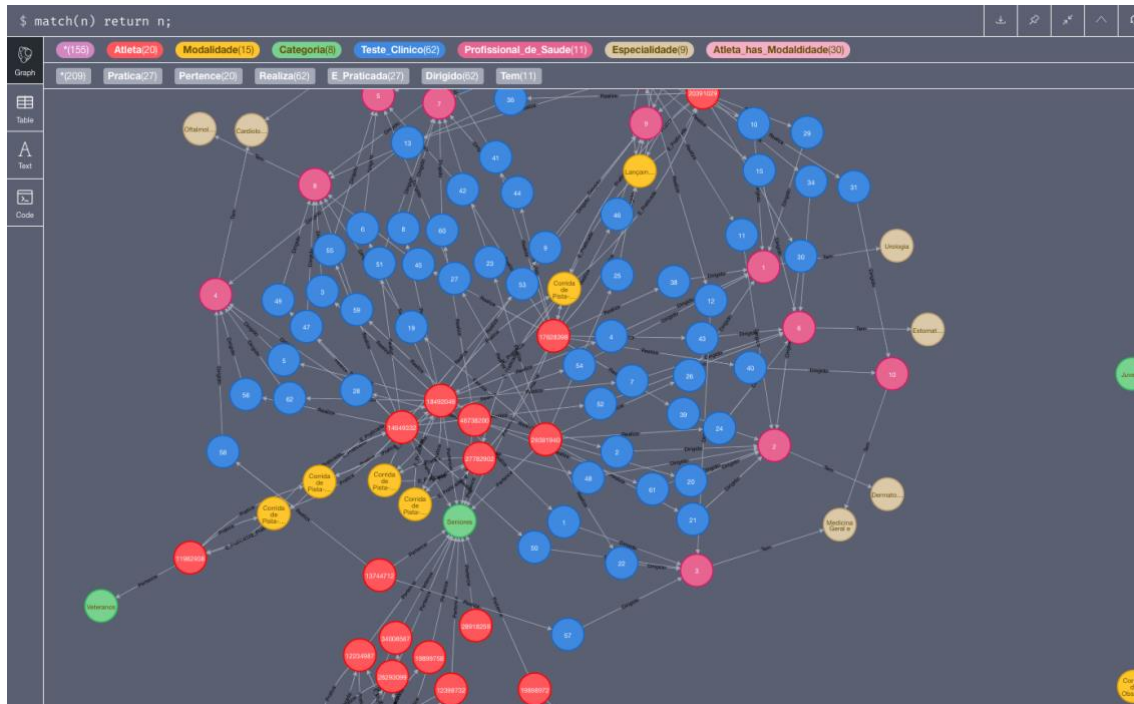


Figura 5: Representação Parcial do Modelo obtido em Neo4J

4.7. Apresentação das questões realizadas

// Query que apresenta os atletas com aptidão = 1

```

MATCH(A:Atleta)
WHERE A.Aptidao="1"
RETURN A;

```

// Query que apresenta os testes clínicos que cada atleta não obteve aptidão positiva

```

MATCH(A:Atleta)-[:Realiza]->(tc:Teste_Clinico)
WHERE tc.Aptidao="0"
RETURN A,tc;

```

//Query que apresenta os testes clínicos dirigidos por um profissional de saúde, de uma determinada especialidade. No exemplo, Especialidade = 3

```
MATCH(tc:Teste_Clinico)-[:Dirigido]->(ps:Profissional_de_Saude)-[:Tem]->(e:Especialidade)
WHERE e.Id_Especialidade="3"
RETURN tc,ps,e;
```

//Query que apresenta os testes concluídos por um dado atleta (Neste caso, CC = 14649332) e os profissionais de saúde que os dirigiram

```
MATCH(A:Atleta)-[:Realiza]->(tc:Teste_Clinico)-[:Dirigido]->(ps:Profissional_de_Saude)
WHERE tc.Estado="Concluido" AND A.CC="14649332"
RETURN tc,A,ps;
```


5. Conclusões e Trabalho Futuro

Ao longo do projeto, o grupo foi capaz de desenvolver uma base de dados que permite o agendamento de testes clínicos, evitando a sobreposição horária dos mesmos, a realização destes para cada atleta, bem como a possibilidade de obtenção de estatísticas sobre os diversos dados presentes na base de dados.

O modelo gerado abarca as principais funcionalidades que uma Unidade de Saúde requer, tais como a possibilidade de acesso aos resultados de aptidão dos testes realizados pelos atletas a qualquer momento (por parte dos profissionais de saúde); à informação crucial que identifica claramente cada atleta e cada profissional de saúde e o estado do teste clínico, entre outros.

A elaboração deste projeto permitiu uma melhor percepção da diferença entre os sistemas de bases de dados relacional e não relacional, possibilitando a compreensão das situações em que é apropriada a aplicação de cada um deles.

Numa perspetiva futura, a fim de melhorar este projeto, seria importante a implementação de uma interface gráfica que permitisse uma visualização mais atrativa das funcionalidades da base de dados. Para além disso, o acréscimo do povoamento da BD seria, de igual modo, um fator importante, na medida em que se conseguiria aproximar o modelo da realidade e, desta forma, conseguir obter outro tipo de resultados de performance, capazes de verificar a viabilidade da base de dados, bem como estabelecer uma barreira entre o modelo relacional e não relacional. No que diz respeito à manutenção da BD, seria crucial implementar mais mecanismos de atualização, tais como a atualização da categoria do atleta mediante o avanço da idade, e segurança dos dados.

Em suma, apesar de certas dificuldades sentidas (nomeadamente na transição do modelo relacional para o não relacional), o grupo tem consciência que o trabalho realizado foi de extrema importância, dado que as bases de dados são uma parte integrante da nossa vida no dia-a-dia.

6. Referências

- [1] Idesporto.pt. (2020). [online] Available at: http://www.idesporto.pt/ficheiros/file/modelo_v0_9.pdf [Accessed 1 Nov. 2019].
- [2] Idesporto.pt. (2020). [online] Available at: http://www.idesporto.pt/ficheiros/file/modelo_v0_9.pdf [Accessed 1 Nov. 2019].
- [3] Gouveia, F. (2014). Fundamentos de Bases de Dados. 1st ed. Lisboa: FCA - Editora de Informática, Lda.
- [4] Dev.mysql.com. (n.d.). MySQL :: MySQL 8.0 Reference Manual :: 11.7 Data Type Storage Requirements. [online] Available at: <https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html> [Accessed 29 Dec. 2019].
- [5] Pordata.pt. (2020). Praticantes desportivos federados total e por algumas federações desportivas. [online] Available at: https://www.pordata.pt/Portugal/Praticantes+desportivos+federados+total+e+por+algumas+federações+desportivas-2226?fbclid=IwAR2Lx1OwU_DbT-LPY9NbhnU07_ryw5NAG2crMt1ARnxckGBE-cUEtKq46Og [Accessed 29 Dec. 2019].
- [6] Neo4j Graph Database Platform. (n.d.). Graph Databases for Beginners: ACID vs. BASE Explained. [online] Available at: <https://neo4j.com/blog/acid-vs-base-consistency-models-explained/> [Accessed 30 Dec. 2019].
- [7] Connolly, T., Begg, C., Database Systems, A Practical Approach to Design, Implementation, and Management, Addison-Wesley, 4ª Edição, 2004.

7. Lista de Siglas e Acrónimos

BD	Base de Dados
SGBD	Sistemas de Gestão de Base de Dados
SQL	<i>Structured Query Language</i>
IPDJ	Instituto Português do Desporto e Juventude
PS	Profissional de Saúde
1FN	Primeira Forma Normal
2FN	Segunda Forma Normal
3FN	Terceira Forma Normal
CC	Número de Cartão de Cidadão
NoSQL	<i>Not Only SQL</i>

Anexos

Nesta secção encontra-se informação adicional necessária para uma melhor compreensão do relatório.

I. Anexo 1 – Código DDL

```
-- 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_D
ATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTIT
UTION';

-----
-- Schema testesclinicos
-----

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

-----
-- Table `testesclinicos`.`Categoria`
-----
CREATE TABLE IF NOT EXISTS `testesclinicos`.`Categoria` (
  `Id_Categoria` INT NOT NULL,
  `Designacao` VARCHAR(30) NULL,
  PRIMARY KEY (`Id_Categoria`))
ENGINE = InnoDB;

-----
-- Table `testesclinicos`.`Atleta`
-----
CREATE TABLE IF NOT EXISTS `testesclinicos`.`Atleta` (
  `CC` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Nacionalidade` VARCHAR(20) NOT NULL,
  `Aptidao` TINYINT NOT NULL,
  `Telemovel` INT NOT NULL,
  `Clube` VARCHAR(50) NOT NULL,
  `Data_Nascimento` DATE NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `Codigo_Postal` VARCHAR(10) NOT NULL,
  `Localidade` VARCHAR(20) NOT NULL,
  `Sexo` VARCHAR(1) NOT NULL,
```

```

`Categoria_id_Categoria` INT NOT NULL,
PRIMARY KEY (`CC`),
INDEX `fk_Atleta_Categoria1_idx` (`Categoria_id_Categoria` ASC) VISIBLE,
CONSTRAINT `fk_Atleta_Categoria1`
FOREIGN KEY (`Categoria_id_Categoria`)
REFERENCES `testesclinicos`.`Categoria` (`Id_Categoria`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `testesclinicos`.`Modalidade`
-----

```

```

CREATE TABLE IF NOT EXISTS `testesclinicos`.`Modalidade` (
  `Id_Modalidade` INT NOT NULL,
  `Designacao` VARCHAR(100) NULL,
  PRIMARY KEY (`Id_Modalidade`))
ENGINE = InnoDB;

```

```

-----
-- Table `testesclinicos`.`Especialidade`
-----

```

```

CREATE TABLE IF NOT EXISTS `testesclinicos`.`Especialidade` (
  `Id_Especialidade` INT NOT NULL,
  `Designacao` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Id_Especialidade`))
ENGINE = InnoDB;

```

```

-----
-- Table `testesclinicos`.`Profissional_de_Saude`
-----

```

```

CREATE TABLE IF NOT EXISTS `testesclinicos`.`Profissional_de_Saude` (
  `Id_PS` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Data_Nascimento` DATE NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `Codigo_Postal` VARCHAR(10) NOT NULL,
  `Localidade` VARCHAR(20) NOT NULL,
  `Especialidade_id_Especialidade` INT NOT NULL,
  PRIMARY KEY (`Id_PS`),
  INDEX `fk_Profissional de Saúde_Especialidade1_idx`
(`Especialidade_id_Especialidade` ASC) VISIBLE,
CONSTRAINT `fk_Profissional de Saúde_Especialidade1`
FOREIGN KEY (`Especialidade_id_Especialidade`)
REFERENCES `testesclinicos`.`Especialidade` (`Id_Especialidade`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `testesclinicos`.`Teste_Clinico`
-----

```

```

-----
CREATE TABLE IF NOT EXISTS `testesclinicos`.`Teste_Clinico` (
  `Id_Testes` INT NOT NULL,
  `Designacao` VARCHAR(45) NULL,
  `Data_Hora` DATETIME NOT NULL,
  `Descricao` VARCHAR(45) NULL,
  `Estado` VARCHAR(20) NOT NULL,
  `Aptidao` TINYINT NULL,
  `Preco` DOUBLE NULL,
  `Atleta_CC` INT NOT NULL,
  `PS_id_PS` INT NOT NULL,
  PRIMARY KEY (`Id_Testes`),
  INDEX `fk_Testes Clínicos_Atleta1_idx` (`Atleta_CC` ASC) VISIBLE,
  INDEX `fk_Testes Clínicos_Profissional de Saúde1_idx` (`PS_id_PS` ASC) VISIBLE,
  CONSTRAINT `fk_Testes Clínicos_Atleta1`
    FOREIGN KEY (`Atleta_CC`)
    REFERENCES `testesclinicos`.`Atleta` (`CC`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Testes Clínicos_Profissional de Saúde1`
    FOREIGN KEY (`PS_id_PS`)
    REFERENCES `testesclinicos`.`Profissional_de_Saude` (`Id_PS`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `testesclinicos`.`Atleta_has_Modalidade`
-----

```

```

CREATE TABLE IF NOT EXISTS `testesclinicos`.`Atleta_has_Modalidade` (
  `Atleta_CC` INT NOT NULL,
  `Modalidade_id_Modalidade` INT NOT NULL,
  INDEX `fk_Atleta_has_Modalidade_Modalidade1_idx` (`Modalidade_id_Modalidade`
ASC) VISIBLE,
  INDEX `fk_Atleta_has_Modalidade_Atleta1_idx` (`Atleta_CC` ASC) VISIBLE,
  PRIMARY KEY (`Atleta_CC`, `Modalidade_id_Modalidade`),
  CONSTRAINT `fk_Atleta_has_Modalidade_Atleta1`
    FOREIGN KEY (`Atleta_CC`)
    REFERENCES `testesclinicos`.`Atleta` (`CC`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Atleta_has_Modalidade_Modalidade1`
    FOREIGN KEY (`Modalidade_id_Modalidade`)
    REFERENCES `testesclinicos`.`Modalidade` (`Id_Modalidade`)
    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;

```

II. Anexo 2 – Povoamento da BD

USE testesclínicos;

-- povoamento modalidade

```
INSERT INTO modalidade(Id_Modalidade,Designacao) VALUES (1,'Corrida de Pista-100m'),
(2,'Corrida de Pista-200m'),
(3,'Corrida de Pista-400m'),
(4,'Corrida de Pista-800m'),
(5,'Corrida de Pista-1500m'),
(6,'Corrida de Obstáculos-100m'),
(7,'Corrida de Obstáculos-200m'),
(8,'Corrida de Obstáculos-2000m'),
(9,'Corrida de Estafetas-2000m'),
(10,'Salto em Comprimento'),
(11,'Salto em Altura'),
(12,'Lançamento de Peso'),
(13,'Lançamento de Dardo'),
(14,'Lançamento de Disco'),
(15,'Lançamento de Martelo');
```

-- povoamento especialidade

```
INSERT INTO especialidade(Id_Especialidade,Designacao) VALUES (1,'Oftalmologia'),
(2,'Dermatologia'),
(3,'Medicina Geral e Familiar'),
(4,'Otorrinolaringologia'),
(5,'Estomatologia'),
(6,'Radiologia'),
(7,'Urologia'),
(8,'Ginecologia-Obstetrícia'),
(9,'Cardiologia');
```

-- povoamento categoria

```
INSERT INTO Categoria(Id_Categoria,Designacao) VALUES (1,'Benjamins A'),
(2,'Benjamins B'),
(3,'Infantis'),
(4,'Iniciados'),
(5,'Juvenis'),
(6,'Juniões'),
(7,'Seniores'),
(8,'Veteranos');
```

-- povoamento profissional de saúde

```
INSERT INTO Profissional_de_Saude
(Id_PS,Nome,Data_Nascimento,Rua,Codigo_Postal,Localidade,Especialidade_Id_Esp
ecialidade) VALUES
(1,'António Ribeiro','1867/04/12','Rua do Capitão','4710-409','Braga',7),
(2,'José Vieira','1964/10/23','Rua São Marco','4710-423','Braga',2),
(3,'Rita Santos','1986/03/07','Rua das Nogueiras',4800-036,'Guimarães',3),
(4,'Luís Medeiros','1969/07/30','Avenida Dos Soldados','4710-129','Braga',9),
(5,'Ana Meireles','1988/12/23','Rua de Teibães','4820-769','Fafe',8),
(6,'Rui Mota','1978/01/12','Travessa da Padaria','4000-784','Porto',5),
(7,'Isabel Pinto','1982/09/1','Rua do Capitão','4710-409','Braga',4),
```

(8,'Maria Pereira','1990/09/9','Rua das Flores','1000-112','Lisboa',1),
 (9,'João Garcia','1973/03/28','Rua Ferreira Santos','4800-797','Guimarães',6),
 (10,'Mariana Nobre','1985/06/13','Avenida da Grande Guerra','4760-219','Famalicão',3),
 (11,'José Rangel','1957/04/15','Rua das Bicas','4000-811','Porto',9);

-- povoamento atleta

INSERT INTO Atleta

(CC,Nome,Nacionalidade,Aptidao,Telemovel,Clube,Data_Nascimento,Rua,Codigo_Postal,Localidade,Sexo,Categoria_Id_Categoria) VALUES (18492049,'Pedro Pimentel','Portuguesa','1','918499890','Sporting Clube de Braga','1996/04/01','Rua Candido Moreira','4710-420','Braga','M',7),
 (17283940,'Hugo Silva','Portuguesa','0','918399841','Sporting Clube de Braga','2001/03/13','Rua São Marco','4710-423','Braga','M',6),
 (29381940,'Nélson Silvestre','Angolana','1','968199332','Vitória Sport Clube','1995/11/25','Travessa dos Livros','4710-877','Braga','M',7),
 (20391029,'Mariana Correia','Portuguesa','1','938817283','Sporting Clube de Braga','2000/03/18','Rua das Macieiras','4800-324','Guimarães','F',6),
 (17628398,'Félix Nogueira','Portuguesa','1','936527778','Clube de Atletismo de Fafe','1993/11/05','Rua Doutor Silva','4710-032','Braga','M',7),
 (14649332,'Luísa Fonseca','Portuguesa','1','914563219','Sporting Clube de Braga','1995/04/25','Rua do Pomar','4800-776','Guimarães','F',7),
 (12234987,'Roberto Caiado','Brasileira','0','935374563','Sporting Clube de Braga','1999/01/03','Rua General Gomes','4710-087','Braga','M',7),
 (19898972,'Filipe Martins','Portuguesa','1','924235302','Sporting Clube de Braga','1998/09/22','Travessa da Terceira','4710-001','Braga','M',7),
 (34008567,'Martim Ribeiro','Portuguesa','0','967821342','Vitória Sport Clube','1999/12/01','Travessa da Ribeirinha','4800-100','Guimarães','M',7),
 (19444493,'Soraia Couto','Portuguesa','1','936782937','Sporting Clube de Braga','2006/06/08','Rua Ferreira Santos','4800-797','Guimarães','F',3),
 (29481029,'Daniel Leal','Portuguesa','1','918493023','Sporting Clube de Braga','2005/11/12','Rua dos Anjos','4730-122','Vila Verde','M',4),
 (19492039,'Catarina Alves','Portuguesa','1','918492009','Sport Lisboa e Benfica','2001/02/21','Rua Nova de Santa Cruz','4710-409','Braga','F',6),
 (12398732,'Óscar Silva','Portuguesa','1','962892998','Sporting Clube de Braga','1987/07/28','Rua Da Chuva','4710-064','Braga','M',7),
 (11982938,'Manuel Carvalho','Portuguesa','1','919929839','Sporting Clube de Braga','1982/02/12','Rua do Sol','4705-224','Aveleda','M',8),
 (28293099,'Margarida Portugal','Portuguesa','1','933323564','Vitória Sport Clube','1995/12/31','Avenida dos Soldados','4710-129','Braga','F',7),
 (46738200,'Gabriel Soares','Brasileira','1','962269974','Centro de Atletismo do Porto','1998/10/30','Avenida da Liberdade','4710-251','Braga','M',7),
 (28918259,'Patrícia Ribeiro','Portuguesa','0','938877123','Sporting CLube de Braga','1991/05/06','Rua dos Amigos','4700-004','Palmeira','F',7),
 (27782902,'Inês Ramalho','Portuguesa','1','961119229','Centro de Atletismo do Porto','1987/01/24','Rua dos Bombeiros','4710-556','Braga','F',7),
 (19899758,'Jerónimo Cunha','Portuguesa','0','939918491','Vitória Sport Clube','1985/07/04','Rua Paricular Silva da Cunha','4800-111','Guimarães','M',7);

-- povoamento teste clínico

INSERT INTO Teste_Clinico

(Id_Testes,Designacao,data_hora,Descricao,Estado,Aptidao,Preco,Atleta_CC,PS_Id_PS) VALUES
 (1,'Exame Biométrico','2019/12/20 12:00:00','Valores normais','Concluido','1',null,18492049,3),

(2,'Exame Dermatologico','2019/12/20 13:00:00','Valores normais','Concluido','1',null,18492049,2),
 (3,'Exame Oftalmologico','2019/12/20 13:30:00','Valores normais','Concluido','1',null,18492049,8),
 (4,'Exame Urologico','2019/12/20 14:00:00','Valores normais','Concluido','1',null,18492049,1),
 (5,'Exame Cardio-Circulatorio','2019/12/20 15:00:00','Valores normais','Concluido','1',null,18492049,4),
 (6,'Exame Ginecologico','2019/12/20 15:30:00','Valores normais','Concluido','1',null,18492049,5),
 (7,'Exame Estomatologico','2019/12/20 16:00:00','Valores normais','Concluido','1',null,18492049,6),
 (8,'Exame ORL','2019/12/20 16:45:00','Valores normais','Concluido','1',null,18492049,7),
 (9,'Exame radiologico','2019/12/20 15:00:00','Valores normais','Concluido','1',null,18492049,9),
 (10,'Exame Urologico','2020/02/03 10:00:00',null,'Marcado',null,null,17283940,1),
 (11,'Exame Dermatologico ','2020/02/03 10:30:00',null,'Marcado',null,null,17283940,2),
 (12,'Exame Biométrico','2020/02/03 11:00:00',null,'Marcado',null,null,17283940,3),
 (13,'Exame Cardio-Circulatorio','2020/02/03 11:30:00',null,'Marcado',null,null,17283940,4),
 (14,'Exame Ginecologico','2020/02/03 14:00:00',null,'Marcado',null,null,17283940,5),
 (15,'Exame Estomatologico','2020/02/03 15:00:00',null,'Marcado',null,null,17283940,6),
 (16,'Exame ORL','2020/02/03 17:00:00',null,'Marcado',null,null,17283940,7),
 (17,'Exame Oftalmologico','2020/02/04 11:00:00',null,'Marcado',null,null,17283940,8),
 (18,'Exame Radiologico','2020/02/03 17:30:00',null,'Marcado',null,null,17283940,9),
 (19,'Exame Biométrico','2019/10/10 12:16:00','Valores normais','Concluido','1',null,29381940,8),
 (20,'Exame Dermatologico','2019/10/11 12:16:00','Valores normais','Concluido','1',null,29381940,2),
 (21,'Exame Dermatologico','2019/10/12 12:16:00','Valores anormais','Concluido','0',null,29381940,2),
 (22,'Exame Biométrico','2019/10/12 13:10:00','Valores normais','Concluido','1',null,29381940,3),
 (23,'Exame ORL','2019/10/12 13:45:00','Valores normais','Concluido','1',null,29381940,7),
 (24,'Exame Estomatologico','2019/10/15 12:16:00','Valores normais','Concluido','1',null,29381940,6),
 (25,'Exame Radiologico','2019/10/15 17:10:00','Valores normais','Concluido','1',null,29381940,9),
 (26,'Exame Urologico','2019/10/15 18:00:00','Valores normais','Concluido','1',null,29381940,1),
 (27,'Exame Genito Urinario ','2019/10/15 18:45:00','Valores normais','Concluido','1',null,29381940,5),
 (28,'Exame Cardio-Circulatorio','2019/10/15 19:15:00','Valores normais','Concluido','1',null,29381940,4),
 (29,'Exame Urologico','2020/03/03 10:00:00',null,'Marcado',null,null,20391029,1),
 (30,'Exame Dermatologico ','2020/03/03 10:30:00',null,'Marcado',null,null,20391029,2),
 (31,'Exame Biométrico','2020/03/03 11:00:00',null,'Marcado',null,null,20391029,10),
 (32,'Exame Cardio-Circulatorio','2020/03/03 11:30:00',null,'Marcado',null,null,20391029,11),
 (33,'Exame Ginecologico','2020/03/03 14:00:00',null,'Marcado',null,null,20391029,5),
 (34,'Exame Estomatologico','2020/03/03 15:00:00',null,'Marcado',null,null,20391029,6),
 (35,'Exame ORL','2020/03/03 17:00:00',null,'Marcado',null,null,20391029,7),
 (36,'Exame Oftalmologico','2020/03/04 11:00:00',null,'Marcado',null,null,20391029,8),

(37,'Exame Radiologico','2020/03/03 17:30:00',null,'Marcado',null,null,20391029,9),
 (38,'Exame Urologico','2019/11/23 10:00:00','Valores Normais','Concluido','1',null,17628398,1),
 (39,'Exame Dermatologico ','2020/03/20 10:30:00',null,'Marcado',null,null,17628398,2),
 (40,'Exame Biométrico','2020/03/24 11:00:00',null,'Marcado',null,null,17628398,10),
 (41,'Exame Cardio-Circulatorio','2019/11/23 11:30:00','Valores Normais','Concluido','1',null,17628398,11),
 (42,'Exame Ginecologico','2020/11/23 14:00:00',null,'Marcado',null,null,17628398,5),
 (43,'Exame Estomatologico','2020/03/25 15:00:00',null,'Marcado',null,null,17628398,6),
 (44,'Exame ORL','2019/11/23 17:00:00','Valores Normais','Concluido','1',null,17628398,7),
 (45,'Exame Oftalmologico','2019/11/23 11:00:00','Valores Normais','Concluido','1',null,17628398,8),
 (46,'Exame Radiologico','2020/03/31 17:30:00',null,'Marcado',null,null,17628398,9),
 (47,'Exame Biométrico','2019/09/10 12:16:00','Problema na vista','Concluido','0',null,14649332,8),
 (48,'Exame Dermatologico','2019/09/11 12:16:00','Valores normais','Concluido','1',null,14649332,2),
 (49,'Exame Dermatologico','2019/09/12 12:16:00','Valores Normais','Concluido','1',null,14649332,8),
 (50,'Exame Biométrico','2019/09/12 13:10:00','Valores normais','Concluido','1',null,14649332,3),
 (51,'Exame ORL','2019/09/12 13:45:00','Valores normais','Concluido','1',null,14649332,7),
 (52,'Exame Estomatologico','2019/09/15 12:16:00','Valores normais','Concluido','1',null,14649332,6),
 (53,'Exame Radiologico','2019/09/15 17:10:00','Valores normais','Concluido','1',null,14649332,9),
 (54,'Exame Urologico','2019/09/15 18:00:00','Valores normais','Concluido','1',null,14649332,1),
 (55,'Exame Genito Urinario ','2019/09/15 18:45:00','Valores normais','Concluido','1',null,14649332,5),
 (56,'Exame Cardio-Circulatorio','2019/09/15 19:15:00','Valores normais','Concluido','1',null,14649332,4);

-- povoamento atleta_has_modalidade

INSERT INTO Atleta_has_Modalidade(Atleta_CC, Modalidade_Id_Modalidade)
 VALUES

(18492049,1),
 (18492049,2),
 (18492049,3),
 (18492049,4),
 (17283940,12),
 (20391029,12),
 (17628398,12),
 (14649332,4),
 (14649332,5),
 (20391029,2),
 (12234987,15),
 (12234987,16),
 (19898972,10),
 (34008567,16),
 (19444493,1),
 (19444493,2),
 (19444493,3),

(29481029,10),
 (29481029,10),
 (19492039,7),
 (12398732,16),
 (12398732,15),
 (11982938,4),
 (11982938,5),
 (28293099,14),
 (28293099,15),
 (27782902,1),
 (27782902,2),
 (27782902,3),
 (19899758,16);

III. Anexo 3 – Tabelas povoadas

CC	Nome	Nacionalidade	Aptidao	Telemovel	Clube	Data_Nascimento	Rua	Codigo_Postal	Localidade	Sexo	Categoria_id_Categoria
► 1944449	Soraia Couto	Portuguesa	1	936782937	Sporting Clube de Braga	2006-06-08	Rua Ferreira Santos	4800-797	Guimarães	F	3
11982938	Manuel Carvalho	Portuguesa	1	919929839	Sporting CLube de Braga	1982-02-12	Rua do Sol	4705-224	Aveleda	M	8
12234987	Roberto Caiado	Brasileira	0	935374563	Sporting Clube de Braga	1999-01-03	Rua General Gomes	4710-087	Braga	M	7
12398732	Oscar Silva	Portuguesa	1	962892998	Sporting Clube de Braga	1987-07-28	Rua Da Chuva	4710-064	Braga	M	7
14649332	Luisa Fonseca	Portuguesa	1	914563219	Sporting Clube de Braga	1995-04-25	Rua do Pomar	4800-776	Guimarães	F	7
17283940	Hugo Silva	Portuguesa	0	918399841	Sporting Clube de Braga	2001-03-13	Rua São Marco	4710-423	Braga	M	6
17628398	Félix Nogueira	Portuguesa	0	936527778	Clube de Atletismo de Fafe	1993-11-05	Rua Doutor Silva	4710-032	Braga	M	7
18492049	Pedro Pimentel	Portuguesa	1	918499890	Sporting Clube de Braga	1996-04-01	Rua Candido Moreira	4710-420	Braga	M	7
19492039	Catarina Alves	Portuguesa	1	918492009	Sport Lisboa e Benfica	2001-02-21	Rua Nova de Santa Cruz	4710-409	Braga	F	6
19898972	Filipe Martins	Portuguesa	1	924235302	Sporting Clube de Braga	1998-09-22	Travessa da Terceira	4710-001	Braga	M	7
19899758	Jerónimo Cunha	Portuguesa	0	939918491	Vitória Sport Clube	1985-07-04	Rua Paricular Silva da...	4800-111	Guimarães	M	7
20391029	Mariana Correia	Portuguesa	0	938817283	Sporting Clube de Braga	2000-03-18	Rua das Macieiras	4800-324	Guimarães	F	6
27782902	Inês Ramalho	Portuguesa	1	961119229	Centro de Atletismo do P...	1987-01-24	Rua dos Bombeiros	4710-556	Braga	F	7
28293099	Margarida Port...	Portuguesa	1	933323564	Vitória Sport Clube	1995-12-31	Avenida dos Soldados	4710-129	Braga	F	7
28918259	Patrícia Ribeiro	Portuguesa	0	938877123	Sporting CLube de Braga	1991-05-06	Rua dos Amigos	4700-004	Palmeira	F	7
29381940	Nélson Silvestre	Angolana	0	968199332	Vitória Sport Clube	1995-11-25	Travessa dos Livros	4710-877	Braga	M	7
29481029	Daniel Leal	Portuguesa	1	918493023	Sporting Clube de Braga	2005-11-12	Rua dos Anjos	4730-122	Vila Verde	M	4
34008567	Martim Ribeiro	Portuguesa	0	967821342	Vitória Sport Clube	1999-12-01	Travessa da Ribeirinha	4800-100	Guimarães	M	7
46738200	Gabriel Soares	Brasileira	1	962269974	Centro de Atletismo do P...	1998-10-30	Avenida da Liberdade	4710-251	Braga	M	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabela 11: Tabela Atleta povoada

	Id_PS	Nome	Data_Nascimento	Rua	Codigo_Postal	Localidade	Especialidade_id_Especialidade
►	1	António Ribeiro	1867-04-12	Rua do Capitão	4710-409	Braga	7
	2	José Vieira	1964-10-23	Rua São Marco	4710-423	Braga	2
	3	Rita Santos	1986-03-07	Rua das Nogueiras	4764	Guimarães	3
	4	Luís Medeiros	1969-07-30	Avenida Dos Soldados	4710-129	Braga	9
	5	Ana Meireles	1988-12-23	Rua de Teibães	4820-769	Fafe	8
	6	Rui Mota	1978-01-12	Travessa da Padaria	4000-784	Porto	5
	7	Isabel Pinto	1982-09-01	Rua do Capitão	4710-409	Braga	4
	8	Maria Pereira	1990-09-09	Rua das Flores	1000-112	Lisboa	1
	9	João Garcia	1973-03-28	Rua Ferreira Santos	4800-797	Guimarães	6
	10	Mariana Nobre	1985-06-13	Avenida da Grande Guerra	4760-219	Famalicão	3
	11	José Rangel	1957-04-15	Rua das Bicas	4000-811	Porto	9
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabela 12: Tabela Profissional_de_Saude povoada

	Id_Categoria	Designacao
▶	1	Benjamins A
	2	Benjamins B
	3	Infantis
	4	Iniciados
	5	Juvenis
	6	Juniores
	7	Seniores
	8	Veteranos
	NULL	NULL

Tabela 13: Tabela Categoria povoada

	Id_Modalidade	Designacao
▶	1	Corrida de Pista-100m
	2	Corrida de Pista-200m
	3	Corrida de Pista-400m
	4	Corrida de Pista-800m
	5	Corrida de Pista-1500m
	6	Corrida de Obstáculos-100m
	8	Corrida de Obstáculos-200m
	9	Corrida de Obstáculos-2000m
	10	Corrida de Estafetas-2000m
	11	Salto em Comprimento
	12	Salto em Altura
	13	Lançamento de Peso
	14	Lançamento de Dardo
	15	Lançamento de Disco
	16	Lançamento de Martelo
	NULL	NULL

Tabela 14: Tabela Modalidade povoada

	Id_Especialidade	Designacao
▶	1	Oftalmologia
	2	Dermatologia
	3	Medicina Geral e Familiar
	4	Otorrinolaringologia
	5	Estomatologia
	6	Radiologia
	7	Urologia
	8	Ginecologia-Obstetrícia
	9	Cardiologia
	NULL	NULL

Tabela 15: Tabela Especialidade povoada

Id_Testes	Designacao	Data_Hora	Descricao	Estado	Aptidao	Preco	Atleta_CC	PS_id_PS
1	Exame Biométrico	2019-12-20 12:00:00	Valores normais	Concluido	1	NULL	18492049	3
2	Exame Dermatologico	2019-12-20 13:00:00	Valores normais	Concluido	1	NULL	18492049	2
3	Exame Oftalmologico	2019-12-20 13:30:00	Valores normais	Concluido	1	NULL	18492049	8
4	Exame Urologico	2019-12-20 14:00:00	Valores normais	Concluido	1	NULL	18492049	1
5	Exame Cardio-Circulatorio	2019-12-20 15:00:00	Valores normais	Concluido	1	NULL	18492049	4
6	Exame Ginecologico	2019-12-20 15:30:00	Valores normais	Concluido	1	NULL	18492049	5
7	Exame Estomatologico	2019-12-20 16:00:00	Valores normais	Concluido	1	NULL	18492049	6
8	Exame ORL	2019-12-20 16:45:00	Valores normais	Concluido	1	NULL	18492049	7
9	Exame radiologico	2019-12-20 15:00:00	Valores normais	Concluido	1	NULL	18492049	9
10	Exame Urologico	2020-02-03 10:00:00	NULL	Marcado	NULL	NULL	17283940	1
11	Exame Dermatologico	2020-02-03 10:30:00	NULL	Marcado	NULL	NULL	17283940	2
12	Exame Biométrico	2020-02-03 11:00:00	NULL	Marcado	NULL	NULL	17283940	3
13	Exame Cardio-Circulatorio	2020-02-03 11:30:00	NULL	Marcado	NULL	NULL	17283940	4
14	Exame Ginecologico	2020-02-03 14:00:00	NULL	Marcado	NULL	NULL	17283940	5
15	Exame Estomatologico	2020-02-03 15:00:00	NULL	Marcado	NULL	NULL	17283940	6
16	Exame ORL	2020-02-03 17:00:00	NULL	Marcado	NULL	NULL	17283940	7
17	Exame Oftalmologico	2020-02-04 11:00:00	NULL	Marcado	NULL	NULL	17283940	8
18	Exame Radiologico	2020-02-03 17:30:00	NULL	Marcado	NULL	NULL	17283940	9
19	Exame Biométrico	2019-10-10 12:16:00	Valores normais	Concluido	1	NULL	29381940	8
20	Exame Dermatologico	2019-10-11 12:16:00	Valores normais	Concluido	1	NULL	29381940	2
21	Exame Dermatologico	2019-10-12 12:16:00	Valores anorm...	Concluido	0	NULL	29381940	2
22	Exame Biométrico	2019-10-12 13:10:00	Valores normais	Concluido	1	NULL	29381940	3
23	Exame ORL	2019-10-12 13:45:00	Valores normais	Concluido	1	NULL	29381940	7
24	Exame Estomatologico	2019-10-15 12:16:00	Valores normais	Concluido	1	NULL	29381940	6
25	Exame Radiologico	2019-10-15 17:10:00	Valores normais	Concluido	1	NULL	29381940	9
26	Exame Urologico	2019-10-15 18:00:00	Valores normais	Concluido	1	NULL	29381940	1
27	Exame Genito Urinario	2019-10-15 18:45:00	Valores normais	Concluido	1	NULL	29381940	5
28	Exame Cardio-Circulatorio	2019-10-15 19:15:00	Valores normais	Concluido	1	NULL	29381940	4
29	Exame Urologico	2020-03-03 10:00:00	NULL	Marcado	NULL	NULL	20391029	1
30	Exame Dermatologico	2020-03-03 10:30:00	NULL	Marcado	NULL	NULL	20391029	2
31	Exame Biométrico	2020-03-03 11:00:00	NULL	Marcado	NULL	NULL	20391029	10
32	Exame Cardio-Circulatorio	2020-03-03 11:30:00	NULL	Marcado	NULL	NULL	20391029	11
33	Exame Ginecologico	2020-03-03 14:00:00	NULL	Marcado	NULL	NULL	20391029	5
34	Exame Estomatologico	2020-03-03 15:00:00	NULL	Marcado	NULL	NULL	20391029	6
35	Exame ORL	2020-03-03 17:00:00	NULL	Marcado	NULL	NULL	20391029	7
36	Exame Oftalmologico	2020-03-04 11:00:00	NULL	Marcado	NULL	NULL	20391029	8
37	Exame Radiologico	2020-03-03 17:30:00	NULL	Marcado	NULL	NULL	20391029	9
38	Exame Urologico	2019-11-23 10:00:00	Valores Normais	Concluido	1	NULL	17628398	1
39	Exame Dermatologico	2020-03-20 10:30:00	NULL	Marcado	NULL	NULL	17628398	2
40	Exame Biométrico	2020-03-24 11:00:00	NULL	Marcado	NULL	NULL	17628398	10
41	Exame Cardio-Circulatorio	2019-11-23 11:30:00	Valores Normais	Concluido	1	NULL	17628398	11
42	Exame Ginecologico	2020-11-23 14:00:00	NULL	Marcado	NULL	NULL	17628398	5
43	Exame Estomatologico	2020-03-25 15:00:00	NULL	Marcado	NULL	NULL	17628398	6
44	Exame ORL	2019-11-23 17:00:00	Valores Normais	Concluido	1	NULL	17628398	7
45	Exame Oftalmologico	2019-11-23 11:00:00	Valores Normais	Concluido	1	NULL	17628398	8
46	Exame Radiologico	2020-03-24 17:30:00	NULL	Marcado	NULL	NULL	17628398	9
47	Exame Biométrico	2019-09-10 12:16:00	Problema na v...	Concluido	0	NULL	14649332	8
48	Exame Dermatologico	2019-09-11 12:16:00	Valores normais	Concluido	1	NULL	14649332	2
49	Exame Dermatologico	2019-09-12 12:16:00	Valores Normais	Concluido	1	NULL	14649332	8
50	Exame Biométrico	2019-09-12 13:10:00	Valores normais	Concluido	1	NULL	14649332	3
51	Exame ORL	2019-09-12 13:45:00	Valores normais	Concluido	1	NULL	14649332	7
52	Exame Estomatologico	2019-09-15 12:16:00	Valores normais	Concluido	1	NULL	14649332	6
53	Exame Radiologico	2019-09-15 17:10:00	Valores normais	Concluido	1	NULL	14649332	9
54	Exame Urologico	2019-09-15 18:00:00	Valores normais	Concluido	1	NULL	14649332	1
55	Exame Genito Urinario	2019-09-15 18:45:00	Valores normais	Concluido	1	NULL	14649332	5
56	Exame Cardio-Circulatorio	2019-09-15 19:15:00	Valores normais	Concluido	1	NULL	14649332	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabela 16: Tabela Teste_Clinico povoada

Atleta_CC	Modalidade_id Modalida...
18492049	1
19444493	1
27782902	1
18492049	2
19444493	2
20391029	2
27782902	2
18492049	3
19444493	3
27782902	3
11982938	4
14649332	4
18492049	4
11982938	5
14649332	5
19492039	7
29481029	9
1989972	10
29481029	10
17283940	12
17628398	12
20391029	12
12398732	13
12234987	14
28293099	14
12234987	15
12398732	15
19899758	15
28293099	15
34008567	15

Tabela 17: Tabela Atleta_has_Modalidade povoada

IV. Anexo 4 – Código das interrogações e transações

USE testesClinicos;

-- ***** PROCEDURES *****

-- Procedure para inserção de um novo Profissional de Saúde

```
DELIMITER //
CREATE PROCEDURE inserePS( IN Nome VARCHAR(45),IN Data_Nascimento DATE,IN
Rua VARCHAR(45),IN Codigo_Postal VARCHAR(10),IN Localidade VARCHAR(20),IN
Especialidade INT)
BEGIN
    DECLARE lastId_PS INT;
    DECLARE Error BOOLEAN DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
    START TRANSACTION;

    SET lastId_PS = (SELECT Id_PS FROM Profissional_de_Saude
                     ORDER BY Id_PS DESC LIMIT 1);
    INSERT INTO Profissional_de_Saude
(Id_PS,Nome,Data_Nascimento,Rua,Codigo_Postal,Localidade,Especialidade_Id_Especial
idade) VALUES
(lastId_PS+1,Nome,Data_Nascimento,Rua,Codigo_Postal,Localidade,Especialidade);

    IF Error=1
    THEN
        BEGIN
            ROLLBACK;
            SELECT('Erro a introduzir os dados');
        END;
    ELSE COMMIT;
    END IF;
END //
DELIMITER //
```

-- Procedure para inserção de um novo Atleta

```
DELIMITER //
CREATE PROCEDURE insereAtleta(IN CC INT,IN Nome VARCHAR(45),IN Nacionalidade
VARCHAR(20),IN Telemóvel INT,IN Clube VARCHAR(50),IN Data_Nascimento DATE,IN
Rua VARCHAR(45),IN Codigo_Postal VARCHAR(10),IN Localidade VARCHAR(20),IN
Sexo VARCHAR(1))
BEGIN
    DECLARE nCategoria INT DEFAULT 0;
    DECLARE nIdade INT DEFAULT 0;
    DECLARE Error BOOLEAN DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
    START TRANSACTION;
    SET nIdade = idade(Data_Nascimento);

    IF (nIdade>=7 AND nIdade<=9)
    THEN SET nCategoria=1;
    END IF;
    IF (nIdade>=10 AND nIdade<=11)
    THEN SET nCategoria=2;
```

```

        END IF;
    IF (nldade>=12 AND nldade<=13)
        THEN SET nCategoria=3;
    END IF;
    IF (nldade>=14 AND nldade<=15)
        THEN SET nCategoria=4;
    END IF;
    IF (nldade>=16 AND nldade<=17)
        THEN SET nCategoria=5;
    END IF;
    IF (nldade>=18 AND nldade<=19)
        THEN SET nCategoria=6;
    END IF;
    IF (nldade>=20 AND nldade<=34)
        THEN SET nCategoria=7;
    END IF;
    IF (nldade>=35)
        THEN SET nCategoria=8;
    END IF;

INSERT INTO Atleta VALUES (CC,Nome,Nacionalidade,0,Telemóvel, Clube,
Data_Nascimento,Rua, Codigo_Postal,Localidade, Sexo,nCategoria);
    IF Error=1
    THEN
        BEGIN
            ROLLBACK;
            SELECT('Erro a introduzir os dados');
            END;
        ELSE COMMIT;
    END IF;
END //
DELIMITER //

```

-- Procedure que calcula o número total de testes clinicos realizados por um determinado profissional de saúde

```

DELIMITER //
CREATE PROCEDURE nTestesPorMedico(IN idPS INT)
    BEGIN
        SELECT COUNT(*) from Profissional_de_Saude AS PS
        JOIN Teste_Clinico as TC ON TC.PS_Id_PS
        WHERE PS.Id_PS=idPS AND TC.PS_Id_PS=idPS
        AND TC.Estado = 'Concluido';

    END //
DELIMITER //

```


-- Procedure que obtém os Testes Clinicos realizados por um dado profissional de saúde num dado dia

```
DELIMITER //
CREATE PROCEDURE nTestesPorMedicoEData(IN idPS INT, IN data DATE)
BEGIN
    SELECT * from Teste_Clinico AS TC
    JOIN Profissional_de_Saude as PS ON TC.PS_Id_PS
    WHERE PS.Id_PS=idPS AND TC.PS_Id_PS=idPS
    AND DATE(TC.data_hora)=data
    AND TC.Estado = 'Concluido';
END //
DELIMITER //
```

-- Procedure que apresenta os testes clinicos que um atleta tem marcados

```
DELIMITER //
CREATE PROCEDURE mostraTestesPorRealizar(IN CC INT)
BEGIN
    SELECT * from Teste_Clinico AS TC
    JOIN Atleta as A ON A.CC=TC.Atleta_CC
    WHERE TC.Estado='Marcado' AND TC.Atleta_CC=CC;
END //
DELIMITER //
```

-- Procedure que marca um teste clínico

```
DELIMITER //
CREATE PROCEDURE MarcaTeste(IN CC INT, IN dataHora DATETIME, IN nPS INT)
BEGIN
    DECLARE lastId_Testes INT;
    DECLARE espec INT;
    DECLARE design VARCHAR(45);
    DECLARE Error BOOLEAN DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;

    START TRANSACTION;

    SET espec = (SELECT Especialidade_Id_Especialidade
                  FROM Profissional_de_Saude
                  WHERE Id_PS=nPS);

    CASE espec
        WHEN 1 THEN SET design = 'Exame Oftalmológico';
        WHEN 2 THEN SET design = 'Exame Ectoscópico';
        WHEN 3 THEN SET design = 'Exame Biométrico';
        WHEN 4 THEN SET design = 'Exame ORL';
        WHEN 5 THEN SET design = 'Exame Estomatológico';
        WHEN 6 THEN SET design = 'Exames Complementares de Diagnóstico';
        WHEN 7 THEN SET design = 'Exame Génito-Urinário';
        WHEN 8 THEN SET design = 'Exame Génito-Urinário';
        WHEN 9 THEN SET design = 'Exame Córdio-Circulatório e Respiratório';
    ELSE SET design = "";
    END CASE;
```



```

SET lastId_Testes = (SELECT Id_Testes
                     FROM Testes_Clinico
                     ORDER BY Id_Testes DESC LIMIT 1);
INSERT INTO TESTES_CLINICO (Id_Testes,Designacao,data_hora, Estado, Atleta_CC,
PS_Id_PS) VALUES
(lastId_Testes+1,design,dataHora,'Marcado',CC,nPS);

        IF Error=1
        THEN
            BEGIN
                ROLLBACK;
                SELECT('Não é possível agendar Teste Clinico: Possível
sobreposição de datas');
            END;
        ELSE COMMIT;
        END IF;
    END //
DELIMITER //

-- Procedure que desmarca um teste clínico

DELIMITER //
CREATE PROCEDURE DesmarcaTeste(IN CC INT, IN dataHora DATETIME)
BEGIN
    DECLARE Error BOOLEAN DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
    START TRANSACTION;

    UPDATE Testes_Clinico AS TC SET TC.Estado='Desmarcado'
        WHERE TC.Atleta_CC=CC AND TC.data_hora=dataHora;
    IF Error=1
    THEN
        BEGIN
            ROLLBACK;
            SELECT('Não é possível desmarcar Teste Clinico');
        END;
    ELSE COMMIT;
    END IF;
END //
DELIMITER //

-- Procedure que remarca uma consulta

DELIMITER //
CREATE PROCEDURE RemarcaTeste (IN CC INT, IN dataHora DATETIME,IN
Nova_dataHora DATETIME)
BEGIN
    DECLARE profSaude INT;
    DECLARE Error BOOLEAN DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
    START TRANSACTION;

    SET profSaude= (SELECT PS_Id_PS FROM Testes_Clinico as TC
        WHERE TC.Atleta_CC=CC AND TC.data_hora=dataHora);

    CALL DesmarcaTeste(CC,dataHora);

```

```

        CALL MarcaTeste(CC,Nova_dataHora,profSaude);

        IF Error=1
            THEN
        ROLLBACK;
            ELSE COMMIT;
        END IF;
    END //
DELIMITER //

-- Procedure que inicia uma consulta

DELIMITER //
CREATE PROCEDURE IniciaTeste(IN idTeste INT)
    BEGIN
        DECLARE ndata DATETIME;
        DECLARE Error BOOLEAN DEFAULT 0;
        DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
        START TRANSACTION;

        SET ndata = (SELECT data_hora FROM Teste_Clinico AS TC
                     WHERE TC.Id_Testes=idTeste);

        IF(CURRENT_TIMESTAMP>TIME(ndata)
        AND CURRENT_TIMESTAMP<ADDTIME(TIME(ndata),"10:00"))
            THEN
                UPDATE Teste_Clinico AS TC
                SET TC.Estado='A decorrer'
                WHERE TC.Id_Testes=idTeste;
        END IF;
        IF Error=1
            THEN
                ROLLBACK;
            ELSE COMMIT;
        END IF;
    END //
DELIMITER //

```

-- Procedure que conclui uma consulta

```

DELIMITER //
CREATE PROCEDURE concluiTeste(IN idTeste INT, IN Descricao VARCHAR(45), IN
Aptidao TINYINT)
    BEGIN
        DECLARE CCid INT;
        DECLARE nEstado VARCHAR(20);
        DECLARE espec INT;
        DECLARE Error BOOLEAN DEFAULT 0;
        DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Error =1;
        START TRANSACTION;

        SET espec = especialidadeTC(idTeste);

        SET CCid = (SELECT Atleta_CC FROM Teste_Clinico as TC
                   WHERE TC.Id_Testes=idTeste);
    END //
DELIMITER //

```

```

SET nEstado = (SELECT Estado from Teste_Clinico AS TC
                WHERE TC.Id_Testes=idTestes);
IF( nEstado = 'A decorrer')
    THEN
UPDATE Teste_Clinico AS TC
SET TC.Preco = precoEspec(espec)
    WHERE TC.Id_Testes=idTestes;

UPDATE Teste_Clinico AS TC
    SET TC.Aptidao = Aptidao
    WHERE TC.Id_Testes=idTestes;

    UPDATE Teste_Clinico AS TC
SET TC.Estado='Concluido'
    WHERE TC.Id_Testes=idTestes;

UPDATE Teste_Clinico AS TC
SET TC.Descricao=Descricao
    WHERE TC.Id_Testes=idTestes;

    END IF;
    IF Error=1
        THEN
            ROLLBACK;
        ELSE COMMIT;
    END IF;
END //
DELIMITER //

-- Procedure que apresenta a média de idades

DELIMITER //
CREATE PROCEDURE mediaIdadesTestes()
    BEGIN
        SELECT AVG(idade(Data_Nascimento)) FROM Atleta AS A
        JOIN Teste_Clinico AS TC ON TC.Atleta_CC = A.CC;
    END //
DELIMITER //

-- ***** FUNCTIONS *****

-- Função que calcula a percentagem de testes clinicos a atletas do sexo feminino

DELIMITER //
CREATE FUNCTION percentagemFEM() RETURNS DOUBLE
DETERMINISTIC
    BEGIN
        DECLARE total INT;
        DECLARE n INT;
        DECLARE percentagem DOUBLE;
        SET total = (SELECT COUNT(*) FROM Teste_Clinico AS TC
                    WHERE TC.Estado='Concluido');
        SET n = (SELECT COUNT(*)
                FROM Teste_Clinico AS TC JOIN Atleta AS A

```

```

        ON TC.Atleta_CC=A.CC
        WHERE A.Sexo='F' AND TC.Estado='Concluido');
SET percentagem = (n/total)*100;

    RETURN percentagem;
END //
DELIMITER //

-- Função que calcula a percentagem de atletas de uma modalidade com aptidao fisica
válida

DELIMITER //
CREATE FUNCTION percentagemValidoMod(modalidadeP INT) RETURNS DOUBLE
DETERMINISTIC
BEGIN
    DECLARE total INT;
    DECLARE aprovados INT;
    DECLARE percentagem DOUBLE;
    SET total = (SELECT COUNT(*) FROM Atleta
        JOIN Atleta_has_Modalidade AS AhM
        ON AhM.Atleta_CC=Atleta.CC
        WHERE
AhM.Modalidade_Id_Modalidade=modalidadeP);
    SET aprovados = (SELECT COUNT(*) FROM Atleta
        JOIN Atleta_has_Modalidade AS AhM
        ON AhM.Atleta_CC=Atleta.CC
        WHERE
AhM.Modalidade_Id_Modalidade=modalidadeP
        AND Atleta.Aptidao = 1);
    SET percentagem = (aprovados/total)*100;

    RETURN percentagem;
END//
DELIMITER //

-- Função que verifica se existe alguma consulta marcada para o mesmo médico numa
determinada data e hora ou num intervalo menor que 15min após essa data

DELIMITER //
CREATE FUNCTION tempoValido(nData DATETIME, nMedico int) returns BOOLEAN
DETERMINISTIC
BEGIN
    DECLARE res BOOLEAN DEFAULT 1;
    DECLARE num INT DEFAULT 0;
    SET num = (SELECT COUNT(*) FROM Teste_Clinico AS TC
        WHERE TC.Ps_Id_PS=nMedico
        AND nData>=SUBTIME(TC.data_hora,TIME('00:14:59'))
        AND
nData<=(ADDTIME(TC.data_hora,TIME('00:14:59'))))
        AND TC.Estado<>'Desmarcado'
        AND TC.Estado<>'Concluido');
    IF(num>0) THEN SET res=0;
    END IF;
    RETURN res;
END//
DELIMITER //

```

-- Função que calcula a idade a partir de um Date

```
DELIMITER //
CREATE FUNCTION idade(dN DATE) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE ano INT;
    SET ano = TIMESTAMPDIFF(YEAR,dN,CURDATE());
    RETURN ano;
END //
DELIMITER //
```

-- Função que devolve o id do último teste de uma dada especialidade para um dado Atleta

```
DELIMITER //
CREATE FUNCTION ultimoTesteEspecialidade(idEspecialidade INT,CCAtleta INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE res INT;
    SET res = (SELECT Id_Testes FROM Testes_Clinico AS TC
                JOIN Profissional_de_Saude AS PdS ON
TC.PS_Id_PS=PdS.Id_PS
                WHERE PdS.Especialidade_Id_Especialidade=idEspecialidade
                AND TC.Atleta_CC = CCAtleta AND TC.Estado='Concluido'
                ORDER BY TC.data_hora DESC LIMIT 1);
    RETURN res;
END //
DELIMITER //
```

-- Função que dado um id de um teste indica a aptidão obtida nesse teste

```
DELIMITER //
CREATE FUNCTION aptTeste(idTeste INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE res INT;
    SET res = (SELECT Aptidao FROM Testes_Clinico AS TC
                WHERE TC.Id_Testes=idTeste);
    RETURN res;
END //
DELIMITER //
```

-- Função que indica quantos testes de diferentes especialidades foram realizados por um dado atleta

```
DELIMITER //
CREATE FUNCTION quantosTestes(idUtilizador INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE res INT DEFAULT 0;
    DECLARE id INT DEFAULT -1;
```

```

        SET id = ultimoTesteEspecialidade(1,idUtilizador);
        if(id IS NOT NULL ) THEN Set res = (res+1);END IF;
        SET id = ultimoTesteEspecialidade(2,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        SET id = ultimoTesteEspecialidade(3,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        SET id = ultimoTesteEspecialidade(4,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        SET id = ultimoTesteEspecialidade(5,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        SET id = ultimoTesteEspecialidade(6,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        SET id = ultimoTesteEspecialidade(7,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        SET id = ultimoTesteEspecialidade(8,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        SET id = ultimoTesteEspecialidade(9,idUtilizador);
        if(id IS NOT NULL) THEN Set res = (res+1); END IF;
        RETURN RES;
    END //;
DELIMITER //

-- Função que indica o número de últimos testes realizados com a Aptidao a 0

DELIMITER //
CREATE FUNCTION quantosTestesZero(idUtilizador INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE res INT DEFAULT 0;
    DECLARE id INT DEFAULT -1;
    SET id = ultimoTesteEspecialidade(1,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(2,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(3,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(4,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(5,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(6,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(7,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(8,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    SET id = ultimoTesteEspecialidade(9,idUtilizador);
        if(id IS NOT NULL AND (aptTeste(id)=0)) THEN Set res = (res+1); END IF;
    RETURN RES;
END //;
DELIMITER //

```

-- Função que indica a especialidade de um Teste Clínico

```
DELIMITER //
CREATE FUNCTION especialidadeTC(idTeste INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE especialidade INT;
    SET especialidade = (SELECT Id_Especialidade FROM Especialidade AS E
                        JOIN Profissional_de_Saude AS PS
                        ON
E.Id_Especialidade=PS.Especialidade_id_Especialidade
                        JOIN Teste_Clinico AS TC ON TC.PS_id_PS=PS.Id_PS
                        WHERE
TC.Id_Testes=idTeste);
    RETURN especialidade;
END //
DELIMITER //
```

-- Função que devolve o preço consoante a especialidade

```
DELIMITER //
CREATE FUNCTION precoEspec(idEspec INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE nPreco INT;
    CASE idEspec
    WHEN 1 THEN SET nPreco=7.5;
    WHEN 2 THEN SET nPreco=7.5;
    WHEN 3 THEN SET nPRECO=5;
    WHEN 4 THEN SET nPRECO=7.5;
    WHEN 5 THEN SET nPRECO=7.5;
    WHEN 6 THEN SET nPRECO=7.5;
    WHEN 7 THEN SET nPRECO=5;
    WHEN 8 THEN SET nPRECO=5;
    WHEN 9 THEN SET nPRECO=5;
    else SET nPreco=0;
    END CASE;
    RETURN nPreco;
END //
DELIMITER //
```

-- Função que indica o número de atletas de uma dada categoria

```
DELIMITER //
CREATE FUNCTION nAtletasCategoria(idCat INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE nAtleta INT DEFAULT 0;

    SET nAtleta = (SELECT COUNT(*) FROM Atleta AS A
                    WHERE A.Categoria_id_Categoria = idCat);

    RETURN nAtleta;
END //
DELIMITER //
```

```
-- ***** TRIGGERS *****

-- Trigger de controlo de inserção do sexo que nao seja M ou F – Atleta

DELIMITER //
CREATE TRIGGER controlGenero BEFORE INSERT ON Atleta
FOR EACH ROW
BEGIN
    IF(new.Sexo<>'M' AND new.Sexo<>'F')
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='Sexo introduzido
Inválido';
    END IF;
END //
DELIMITER //

-- Trigger de controlo para marcações de datas indisponíveis

DELIMITER //
CREATE TRIGGER controlData BEFORE INSERT ON Teste_Clinico
FOR EACH ROW
BEGIN
    IF(new.Estado<>'Concluido' AND
new.data_hora<(CURRENT_TIMESTAMP)
OR tempoValido(new.data_hora,new.PS_Id_PS)=0 )
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='Data Indisponível';
    END IF;
END //
DELIMITER //

-- Trigger que altera o estado de Aptidao de um Atleta consoante os resultados dos Testes
Concluido após um update em Teste_Clinico

DELIMITER //
CREATE TRIGGER atualizaAptidaoU AFTER UPDATE ON Teste_Clinico
FOR EACH ROW
BEGIN
    IF(quantosTestes(new.Atleta_CC)=9 AND quantosTestesZero(new.Atleta_CC)=0)
    THEN UPDATE Atleta SET Atleta.Aptidao=1 WHERE
Atleta.CC=new.Atleta_CC;
    END IF;
    IF(quantosTestes(new.Atleta_CC)<>9 OR
quantosTestesZero(new.Atleta_CC)<>0)
    THEN UPDATE Atleta SET Atleta.Aptidao=0 WHERE
Atleta.CC=new.Atleta_CC;
    END IF;
END //
DELIMITER //

-- Trigger que altera o estado de Aptidao de um Atleta consoante os resultados dos Testes
Concluido após um Insert em Teste_Clinico

DELIMITER //
CREATE TRIGGER atualizaAptidaoI AFTER INSERT ON Teste_Clinico
FOR EACH ROW
BEGIN
    IF(quantosTestes(new.Atleta_CC)=9 AND quantosTestesZero(new.Atleta_CC)=0)
    THEN UPDATE Atleta SET Atleta.Aptidao=1
    WHERE Atleta.CC=new.Atleta_CC;
```



```

        END IF;
        IF(quantosTestes(new.Atleta_CC)<>9 OR quantosTestesZero(new.Atleta_CC)<>0)
        THEN UPDATE Atleta SET Atleta.Aptidao=0
            WHERE Atleta.CC=new.Atleta_CC;
        END IF;
    END //
DELIMITER //

```

```
-- ***** VIEWS *****
```

```
-- View que mostra os testes a decorrer no momento
```

```

DROP VIEW IF EXISTS aDecorrer;
CREATE VIEW aDecorrer AS
    SELECT DISTINCT * FROM Teste_Clinico AS TC
        WHERE TC.Estado='A decorrer';

```

```
-- View que apresenta os atletas não aptos
```

```

DROP VIEW IF EXISTS nAptosView;
CREATE VIEW nAptosView AS
    SELECT DISTINCT Nome AS 'Nome',
        idade(Data_Nascimento) AS 'Idade',
        CC AS 'Nº Cartão de Cidadão'
    FROM Atleta WHERE Aptidao=0;

```

V. Anexo 5 – Script migração de dados NoSQL

```

// Create Atleta
LOAD CSV WITH HEADERS FROM "file:///atleta.csv" AS LINE
CREATE (a:Atleta)
SET a = LINE;

// Create Atleta_has_Modalidade
LOAD CSV WITH HEADERS FROM "file:///Atleta_has_Modalidade.csv" AS LINE
CREATE (am:Atleta_has_Modalidade)
SET am = LINE;

// Create Modalidade
LOAD CSV WITH HEADERS FROM "file:///Modalidade.csv" AS LINE
CREATE (m:Modalidade)
SET m = LINE;

// Create Categoria
LOAD CSV WITH HEADERS FROM "file:///Categoria.csv" AS LINE
CREATE (c:Categoria)
SET c = LINE;

// Create Teste_Clinico
LOAD CSV WITH HEADERS FROM "file:///Teste_Clinico.csv" AS LINE
CREATE (tc:Teste_Clinico)

```

```

SET tc = LINE;

// Create Profissional_de_Saude
LOAD CSV WITH HEADERS FROM "file:///Profissional_de_Saude.csv" AS LINE
CREATE (ps:Profissional_de_Saude)
SET ps = LINE;

// Create Especialidade
LOAD CSV WITH HEADERS FROM "file:///Especialidade.csv" AS LINE
CREATE (e:Especialidade)
SET e= LINE;

MATCH (tc:Teste_Clinico),(a:Atleta)
WHERE tc.Atleta_CC = a.CC
CREATE (a)-[r:Realiza]->(tc)
RETURN tc,a;

MATCH (a:Atleta),(c:Categoria)
WHERE a.Categoria_id_Categoria = c.Id_Categoria
CREATE (a)-[r:Pertence]->(c)
RETURN a,c;

MATCH (ps:Profissional_de_Saude),(e:Especialidade)
WHERE ps.Especialidade_id_Especialidade = e.Id_Especialidade
CREATE (ps)-[r:Tem]->(e)
RETURN ps,e;

MATCH (tc:Teste_Clinico),(ps:Profissional_de_Saude)
WHERE tc.PS_id_PS = ps.Id_PS
CREATE (tc)-[r:Dirigido]->(ps)
RETURN tc,ps;

MATCH (a:Atleta),(ahm:Atleta_has_Modalidade),(m:Modalidade)
WHERE a.CC=ahm.Atleta_CC AND ahm.Modalidade_id_Modalidade=m.Id_Modalidade
CREATE (a)-[r:Pratica]->(m), (m)-[r1:E_Praticada]->(a)
RETURN a,m;

```