

Bases de Dados
(3º ano da Licenciatura em Ciências da Computação)
**Agendamento e Realização de Testes Clínicos de Atletas de
Atletismo**
Relatório de Desenvolvimento

Daniel Regado
(a85137)

Filipe Freitas
(a85026)

Constança Elias
(a83543)

Maria Barbosa
(a85290)

janeiro de 2020

Resumo

Este trabalho tem como objetivo criar uma base de dados com o seguinte tema: Agendamento e realização de Testes Clínicos de atletas de Atletismo de diferentes modalidades e categorias. Na primeira parte, é feita a implementação de uma base de dados relacional, passando pelas fases de planeamento, modelação, modelação conceptual, modelação lógica, modelação física e manutenção. Usamos *MySQL* como o sistema de gestão de Bases de Dados.

Na segunda parte, detalhamos a migração dos dados para uma base de dados não relacional (especificamente *MongoDB*), bem como a criação das *queries* equivalentes às utilizadas em *MySQL* para demonstrar a utilidade da base de dados no modelo relacional.

Por fim, é feita uma análise crítica ao trabalho desenvolvido e às dificuldades sentidas, e são referidas possíveis melhorias.

Área de Aplicação: Planeamento e Implementação de um Sistema de Bases de Dados

Palavras-chave: MySQL, Modelação, Modelo Lógico, Modelo Físico, MongoDB

Conteúdo

1	Introdução	5
1.1	Contextualização	5
1.2	Apresentação do Caso de Estudo	5
1.3	Motivação e Objetivos	5
1.4	Estrutura do Relatório	6
2	Base de dados Relacional	7
2.1	Construção do Modelo Conceptual	7
2.1.1	Requisitos	7
2.1.2	Perfis de utilização	7
2.1.3	Entidades	7
2.1.4	Relacionamentos	8
2.1.5	Multiplicidade dos Relacionamentos	8
2.1.6	Definição dos Atributos e chaves	8
2.1.7	Verificação da redundância no modelo	11
2.1.8	Validação do Modelo Conceptual	11
2.2	Modelo Lógico	12
2.2.1	Validação do Modelo Lógico	12
2.3	Modelo Físico	12
2.3.1	Queries e procedimentos	13
2.3.2	Vistas	15
2.3.3	Utilizadores	15
2.3.4	Povoamento	16
2.3.5	Estimativa do Espaço Ocupado	16
3	Base de Dados Não Relacional	18
3.1	Introdução	18
3.2	Migração dos dados para MongoDB	18
3.3	Queries	19
4	Conclusão	21
A	Script SQL de Preparação da Base de Dados MySQL para utilização	22
B	Script SQL de Povoamento das tabelas SQL	25
C	Script contendo todas as queries SQL construídas	38
D	Script para migração dos dados de SQL para MongoDB	40

E	Script contendo todas as queries MongoDB equivalentes às queries SQL do anexo C	42
---	---	----

Lista de Figuras

2.1	Diagrama ER do Modelo Conceptual	11
2.2	Esquema do Modelo Lógico	12
2.3	Criação física da tabela Modalidade	13
2.4	Opção <i>Cascade</i>	14
3.1	Query no ficheiro testes.sql	18
3.2	Query no ficheiro testes.sql	19
3.3	Estrutura da informação na colecção TesteClinico na base de dados MongoDB	19
3.4	Query que permite consultar as modalidades praticadas pelos atletas registados na base de Dados	19
3.5	Query que permite listar o nome dos atletas que foram submetidos a testes clínicos	19
3.6	Query que permite verificar os testes agendados, apresentando o nome dos atletas que tem testes agendados e o nome do respectivo teste	19
3.7	Query que permite verificar o tipo de testes que os atletas da modalidade "Salto em Altura" realizam	20

Lista de Tabelas

2.1	Tabela de entidades	8
2.2	Tabela da multiplicidade dos relacionamentos	8
2.3	Tabela de atributos	10
2.4	Espaço ocupado por linha para cada tabela	16
2.5	Espaço ocupado por tabela	17

Capítulo 1

Introdução

1.1 Contextualização

A associação de atletismo de Braga (AAB) tem vindo a crescer e o seu número de associados é cada vez maior. Com este aumento, a AAB tem vindo a aumentar o leque de modalidades que os seus atletas praticam. Para assegurar a saúde dos mesmos, a associação realiza alguns testes clínicos de check-up aos atletas ao longo do ano, bem como testes de controlo de Anti-Doping, para manter a integridade das suas competições. Estes testes são agendados e pagos pela associação, e é esta que faz toda a gestão dos mesmos, contado com uma clínica privada para a realização destes exames.

Nesta associação existem várias modalidades como salto, corrida e lançamentos. Para cada uma destas modalidades existem várias categorias (que são iguais para todas as modalidades): m20 (para homens com idade inferior ou igual a 20), f20 (para mulheres com idade inferior ou igual a 20), m30, f30, m15, f15, m40, f40. No entanto, estas categorias não têm relevância nos testes clínicos realizados: todos os atletas realizam os mesmos testes, independentemente da sua categoria ou modalidade.

1.2 Apresentação do Caso de Estudo

A direção da equipa de atletismo pretende criar uma base de dados para guardar toda a informação relativa ao agendamento e realização dos testes clínicos de cada atleta, de forma a facilitar a gestão da equipa e dos profissionais médicos que os realizam.

Após uma reunião com a AAB, ficou claro que é preciso pensar num sistema de gestão de base de dados como uma estrutura do ponto de vista da equipa de atletismo, ou seja, sabendo que os utilizadores principais da base de dados serão os diretores da equipa, os atletas, e os profissionais de saúde que trabalham com a equipa. Relativamente à gestão do material necessário para a realização dos testes, esta é feita pela clínica pelo que não é relevante para o caso em questão.

1.3 Motivação e Objetivos

A gestão manual do agendamento dos testes clínicos tornou-se inviável porque, como já foi referido, o número de sócios da associação tem vindo a crescer a um ritmo acelerado. Sabendo a modalidade do atleta e a hora e data das provas em que participa, bem como a disponibilidade dos profissionais, torna-se mais fácil e eficaz o agendamento dos testes clínicos. Com a criação de uma base de dados consistente, a informação torna-se mais acessível e fácil de modificar.

1.4 Estrutura do Relatório

Este relatório divide-se em quatro partes.

A primeira, correspondente a esta introdução, pretende contextualizar o trabalho e explicar a sua estrutura.

A segunda pretende explicar a modelação, estrutura e implementação da base de dados relacional, enquanto que a terceira descreve o processo usado para obter a base de dados não relacional.

Na última parte é feita uma conclusão do trabalho desenvolvido e possível trabalho futuro.

No final do relatório são também apresentados anexos com os scripts SQL e todos os scripts Python utilizados durante o desenvolvimento do trabalho, assim como uma breve explicação daquilo que faz cada um.

Capítulo 2

Base de dados Relacional

2.1 Construção do Modelo Conceptual

2.1.1 Requisitos

A AAB impõe um requisito quanto à hora de realização dos testes clínicos. À exceção dos testes de controlo anti-doping, que podem ser feitos em qualquer altura, os restantes não podem coincidir com o dia das provas. Por isso o agendamento de um teste deve ter em conta as horas das provas. As provas vêm sempre associadas a uma determinada modalidade.

De acordo com esta associação não há problema que os testes coincidam com a hora do treino pelo que não é relevante guardar informação relativamente aos treinos. Para o propósito da base de dados em questão, não é necessário distinguir a modalidade da categoria, uma vez que esta última pode ser inferida a partir da idade e do sexo do atleta. Para além disso, como já foi referido, esta não é relevante para o agendamento dos testes. Estes testes são realizados por um profissional (ou mais caso seja necessário), que tem uma equipa de profissionais associada. Isto significa que há um conjunto de médicos que coordenam os exames e quando é preciso mais do que uma pessoa para fazer o teste, a equipa entra em ação e cada profissional tem sempre a mesma equipa associada.

2.1.2 Perfis de utilização

Este sistema de gestão deve ser utilizado pela direção da AAB, para gerir o agendamento dos testes e a sua realização bem com a inserção de novos atletas. A direção da equipa é então o administrador da base de dados. Para além deste, é conveniente criar um utilizador **atleta**, que possa consultar as datas dos seus exames clínicos (à exceção dos testes de controlo anti-doping), e também um utilizador **profissional** que possa indicar se o teste foi realizado e saber os exames que tem marcados.

2.1.3 Entidades

De acordo com o levantamento de requisitos, percebemos que há palavras chave como é o caso de: Atleta, Profissional, Modalidade, Prova e Teste Clínico. Chegámos então à conclusão que para o propósito da nossa base de dados são necessárias estas 5 categorias:

Entidade	Descrição
Atleta	O sócio da AAB que pratica atletismo
Modalidade	A modalidade de atletismo que o atleta pratica
Prova	Uma prova de uma determinada modalidade
Teste Clínico	O teste que vai ser realizado pelo atleta
Profissional	Quem irá realizar o teste clínico

Tabela 2.1: Tabela de entidades

2.1.4 Relacionamentos

Tendo em conta as entidades definidas, pretendemos que estas se relacionem da seguinte forma:

- Atleta *treina* Modalidade
- Modalidade *realiza* Prova
- Atleta *realiza* Teste Clínico
- Teste Clínico *realizado* por Profissional

2.1.5 Multiplicidade dos Relacionamentos

No nosso modelo só existem relacionamentos (1:N). Tendo em conta os requisitos, admitimos que os testes que eventualmente poderão precisar de mais do que um profissional para os realizar têm uma equipa fixa, ou seja, cada profissional tem uma equipa associada. Segundo a política da AAB, 1 atleta só pode praticar uma modalidade, mas cada modalidade é composta por vários atletas. De seguida apresentamos a tabela referente à multiplicidade de relacionamentos.

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
Atleta	1	Realiza	N	TesteClínico
TesteClínico	N	Realizado por	1	Profissional
Atleta	N	Treina	1	Modalidade
Modalidade	1	Realiza	N	Prova

Tabela 2.2: Tabela da multiplicidade dos relacionamentos

2.1.6 Definição dos Atributos e chaves

Tendo em conta que este sistema de base de dados tem como objetivo facilitar o agendamento e realização de testes clínicos, é necessário que permita agendar um teste e saber se foi realizado. Isso é indicado através do atributo *realizado* da entidade *Teste Clínico*. Este atributo é um valor booleano que indica se o teste já foi efetuado (*realizado* = 1) ou não (*realizado* = 0). Sempre que se aproxima a data de realização de um

teste, é enviada uma carta para a morada do atleta. Por isso é necessário guardar a morada. Os testes são pagos pela AAB, pois a quota paga pelos atletas já inclui a realização destes testes. Convém guardar o preço de cada teste para facilitar a gestão do dinheiro.

Apresenta-se agora a tabela referente aos atributos de cada entidade bem como o seu tipo de dados e a indicação das chaves primárias de cada relação.

Entidade	Atributos	Descrição	Tipo de Dados e Tamanho	Nulo	Multivalorado	Derivado
Atleta	Id (chave primária)	Número identificativo de um atleta	Inteiro	Não	Não	Não
	nome	Nome do atleta	String até 64 caracteres	Não	Não	Não
	sexo	Sexo do atleta	1 caracter	Não	Não	Não
	peso	Peso do atleta	Real	Não	Não	Não
	altura	Altura do atleta	Real	Não	Não	Não
	morada	Morada do atleta	String até 128 caracteres	Não	Não	Não
	dob	Data de nascimento do atleta	data	Não	Não	Não
Modalidade	Id (chave primária)	Número identificativo da modalidade	Inteiro	Não	Não	Não
	nome	Nome da modalidade	String até 45 caracteres	Não	Não	Não
Prova	Id (chave primária)	Número identificativo da prova	Inteiro	Não	Não	Não
	data_hora	Data e hora de realização da prova	tempo	Não	Não	Não
	duração	Duração da prova	inteiro	Não	Não	Não
Teste Clínico	Id (chave primária)	Número identificativo do teste clínico	inteiro	Não	Não	Não
	nome	Tipo de teste	String até 45 caracteres	Não	Não	Não
	realizado	Bandeira que indica se o teste está apenas agendado ou se já foi realizado	booleano	Não	Não	Não
	preco	Preço do teste	real	Não	Não	Não
	data_hora	Data e hora da realização do teste	tempo	Não	Não	Não
Profissional	Id (chave primária)	Número identificativo do profissional	Inteiro	Não	Não	Não
	nome	Nome do profissional	String até 45 caracteres	Não	Não	Não

Tabela 2.3: Tabela de atributos

Com os atributos e os relacionamentos definidos obtemos o seguinte diagrama ER (Entidade-Relação).

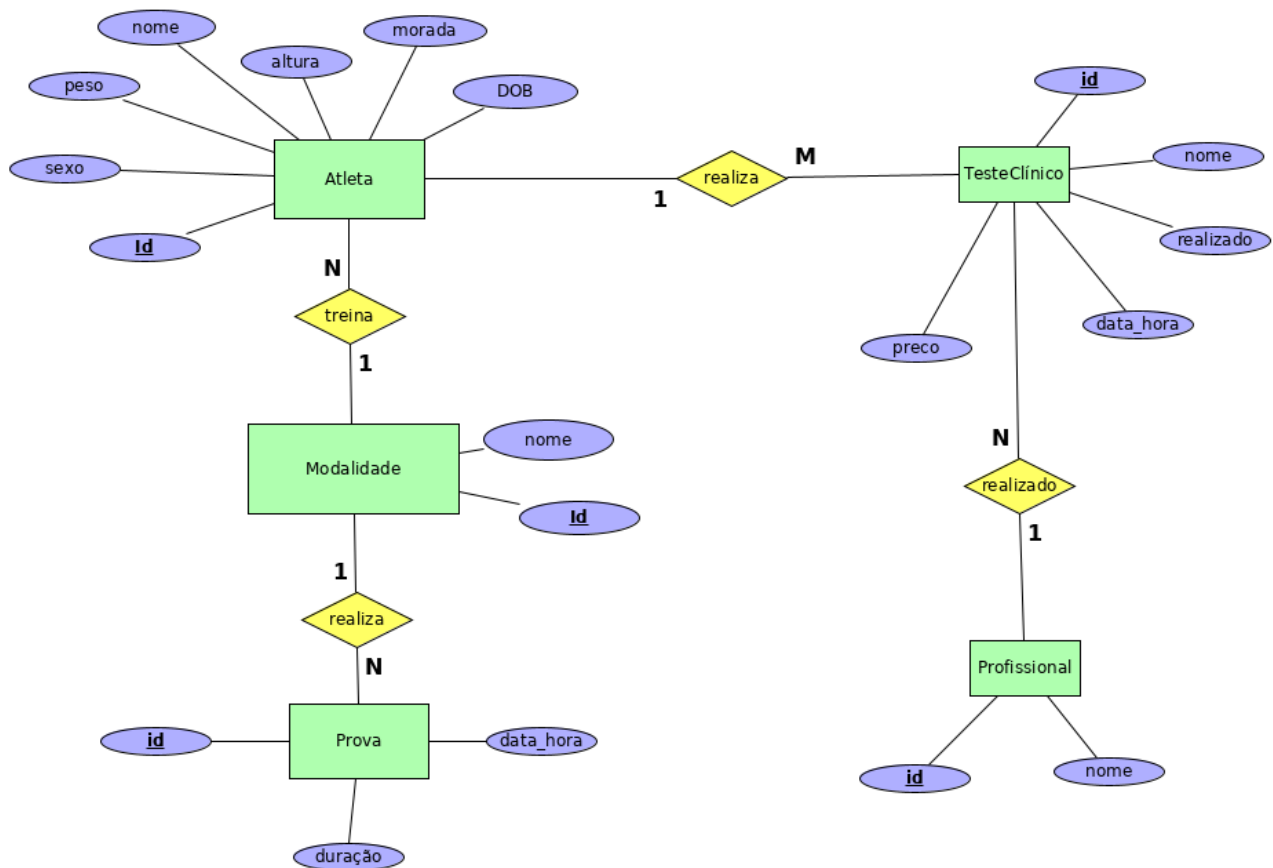


Figura 2.1: Diagrama ER do Modelo Conceptual

2.1.7 Verificação da redundância no modelo

Vamos analisar o nosso modelo de acordo com os três passos de verificação da redundância:

1. Examinar os relacionamentos (1:1)
Analisando o nosso modelo, este passo é ignorado uma vez que não existem estes relacionamentos no modelo.
2. Remover relacionamentos redundantes
Uma vez que não existem ciclos nos relacionamentos entre as entidades, não é colocado o problema da redundância.
3. Considerar a dimensão do tempo
Todas as entidades e relacionamentos que temos são indispensáveis ao propósito da base de dados e verificadas em tempo útil.

2.1.8 Validação do Modelo Conceptual

Após a realização de todas as etapas de construção do modelo, este foi revisto pela AAB e foi aprovado uma vez que permite a concretização do que foi pedido e corresponde à estrutura e organização da associação.

2.2 Modelo Lógico

Passando para o modelo lógico, foi necessário construir as tabelas referentes a cada entidade e definir as chaves estrangeiras de cada tabela. As entidades forte que o nosso sistema possui são 5. No nosso modelo não temos entidades fracas. Com base nos relacionamentos entre as entidades, definimos as chaves estrangeiras. Como apenas temos relacionamentos do tipo 1:N a chave primária da entidade “pai” (1), é utilizada como chave estrangeira na entidade “filha” (N). Já mencionámos antes que os atributos têm de ter valor, ou seja, são *NOT NULL*.

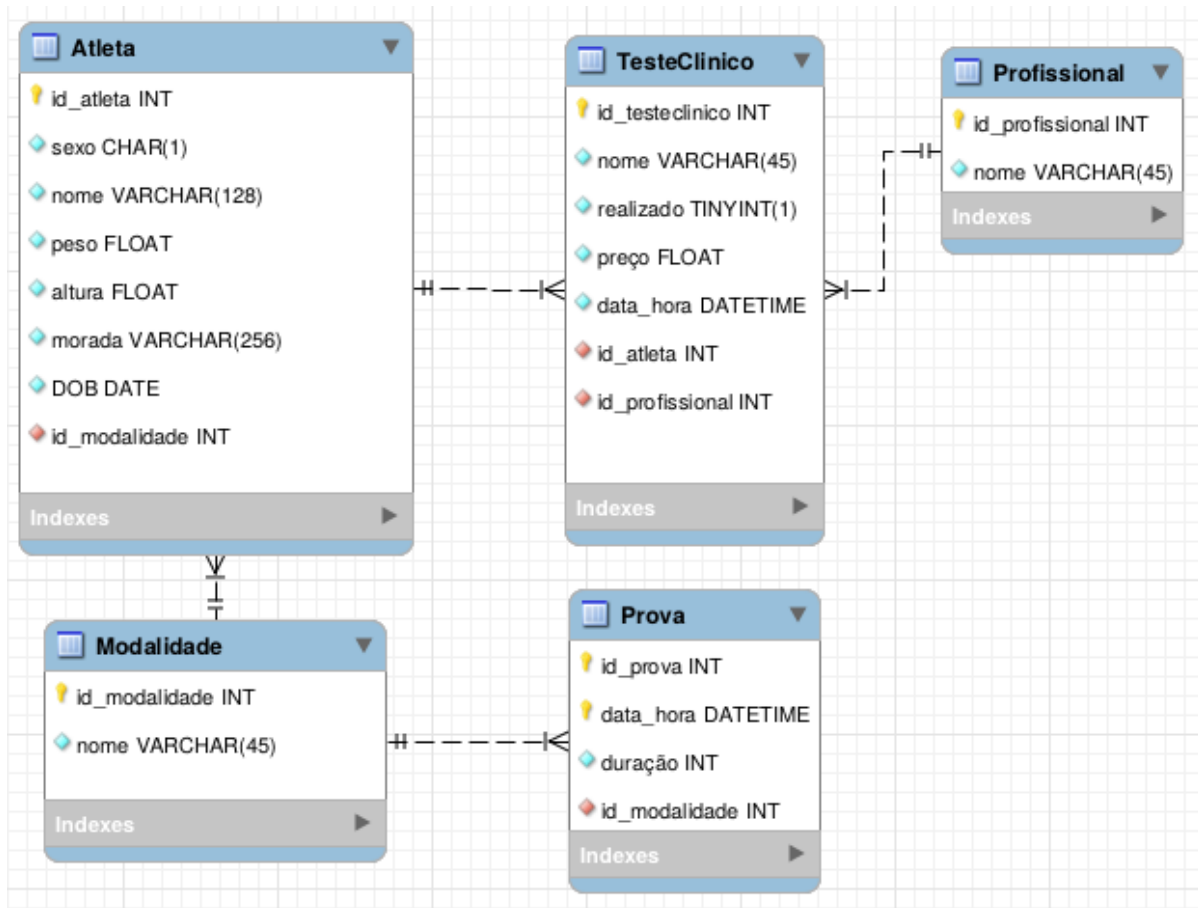


Figura 2.2: Esquema do Modelo Lógico

2.2.1 Validação do Modelo Lógico

Após analisar as dependências funcionais de cada atributo, verificámos que o modelo se encontra normalizado até à Terceira Form Normal (3FN) da Normalização. O modelo foi revisto com o utilizador e não há nada a apontar.

2.3 Modelo Físico

Esta etapa consiste na criação do modelo físico a partir do modelo lógico, através da ferramenta *Forward Engineer* do *MySQL WorkBench*. De seguida apresenta-se um exemplo da criação automática de uma tabela com a ferramenta. Ativámos a opção *AUTO INCREMENT* para geração do *id*.

```

-----
-- Table `TestesClinicos`.`Modalidade`
-----
CREATE TABLE IF NOT EXISTS `TestesClinicos`.`Modalidade` (
  `id_modalidade` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_modalidade`),
  UNIQUE INDEX `id_modalidade_UNIQUE` (`id_modalidade` ASC))
ENGINE = InnoDB;

```

Figura 2.3: Criação física da tabela Modalidade

Algumas ações essenciais à base de dados, nomeadamente poder inserir um novo tipo de teste, agendar um novo teste, desmarcar um teste, marcar um teste como realizado, inserir um novo atleta são apenas comandos de delete, update e insert. As restantes operações necessárias, obtêm-se à custa das queries e procedimentos que definimos a seguir.

2.3.1 Queries e procedimentos

Encontrámos um conjunto de queries que são relevantes tendo em conta o propósito do sistema. De seguida, apresentam-se algumas mas o ficheiro completo encontra-se no anexo C.

```

-- Query 1) Listar todas as modalidades registadas na base de dados
SELECT nome AS 'Atleta'
  FROM Modalidade;

-- Query 2) Listar os atletas registados
SELECT nome as 'Atleta'
  FROM Atleta;

-- Query 3) Listar os atletas e as suas modalidades
SELECT a.nome AS 'Atleta', m.nome AS 'Modalidade'
  FROM Atleta a, Modalidade m
 WHERE a.id_modalidade = m.id_modalidade;

-- Query 4) Listar os nome dos atletas de uma dada modalidade, por exemplo: 'Salto em
↳ Altura'
SELECT a.nome AS 'Atleta', m.nome AS 'Modalidade'
  FROM Atleta a, Modalidade m
 WHERE a.id_modalidade = m.id_modalidade
       AND m.nome = 'Salto em Altura';

-- Query 5) Listar o tipo de testes que já foram realizados
SELECT nome AS 'Tipo de Teste'
  FROM TesteClinico
 GROUP BY nome;

```

A passagem de um teste para realizado é ativada quando o profissional "diz", na aplicação existente na clínica, que o exame foi realizado. No lado da base de dados, isso traduz-se no seguinte procedimento:

```
DELIMITER //
```

```
CREATE PROCEDURE Passar_Testes_a_Realizados(id_prof int, dataHora datetime)
```

```
BEGIN
```

```
    UPDATE TesteClinico SET realizado = 1 WHERE data_hora = dataHora AND id_prof =
```

```
        ↳ id_profissional;
```

```
END //
```

```
DELIMITER ;
```

Assim sendo, um profissional poderá fazer *updates* na base de dados na tabela *TesteClinico*. Poderá também agendar novos testes e nesse caso também deve ter a permissão de inserir na tabela *TesteClinico*.

Quando um atleta deixa de ser sócio da AAB, é necessário eliminar toda a informação presente no sistema que seja referente ao atleta. Para garantir isso, tivemos de ativar a opção *CASCADE*, na *constraint* da *foreign key* *id_atleta*, para garantir que, quando se elimina um atleta, os testes a ele associados também são removidos da BD.

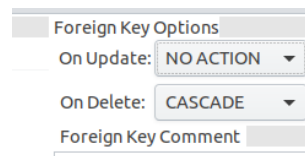


Figura 2.4: Opção *Cascade*

Para eliminar os testes que foram agendados mas não realizados e já passaram da data atual, criámos o seguinte procedimento:

```
DELIMITER //
```

```
CREATE PROCEDURE Eliminar_Testes_Nao_Realizados()
```

```
BEGIN
```

```
    DELETE FROM TesteClinico WHERE data_hora < NOW() AND realizado = 0;
```

```
END //
```

```
DELIMITER ;
```


2.3.2 Vistas

Uma vista permite guardar o resultado de uma consulta nos dados da BD. No caso da AAB, é importante ter uma vista para os atletas, para que possam consultar quando têm testes clínicos e nada mais. É importante também que os profissionais possam ver quando têm de realizar testes para que possam organizar a agenda. Assim, criámos duas vistas para serem incorporadas no respetivos utilizadores: atleta e profissional.

*-- View 1) Vista que lista os atletas que têm testes agendados nas próximas duas semanas
→ (excepto testes de controlo Anti-Doping!)*

```
CREATE VIEW Atletas_Testes_Prox2Semanas AS (  
    SELECT a.nome AS Atleta,  
           t.nome AS 'Tipo de Teste',  
           t.data_hora as 'Data/Hora'  
    FROM Atleta a, TesteClinico t  
    WHERE a.id_atleta = t.id_atleta  
           AND t.data_hora BETWEEN CURRENT_DATE() AND DATE_ADD(CURRENT_DATE(),  
→ INTERVAL 14 DAY)  
           AND t.nome NOT LIKE '%Controlo Anti-Doping%'  
);
```

*-- View 2) Vista que lista todos os profissionais que têm testes marcados durante o
→ próximo mês*

```
CREATE VIEW Profissionais_Testes_ProxMes AS (  
    SELECT p.nome AS 'Médico',  
           t.data_hora AS 'Data/Hora'  
    FROM Profissional p, TesteClinico t  
    WHERE p.id_profissional = t.id_profissional  
           AND t.data_hora BETWEEN CURRENT_DATE()  
           AND DATE_ADD(CURRENT_DATE(), INTERVAL 30 DAY)  
);
```

2.3.3 Utilizadores

Para além do administrador, que é a direção da AAB, existe mais dois utilizadores, como já foi referido anteriormente. O atleta pode ver quanto tem provas e testes clínicos marcados. O profissional pode ver quando tem testes agendados e pode fazer inserir e dizer que o teste foi realizado (à custa do procedimento).

```
CREATE USER 'atleta'@'localhost' IDENTIFIED BY 'atleta';  
GRANT SELECT ON Atletas_Testes_Prox2Semanas TO 'atleta'@'localhost';  
GRANT SELECT ON TestesClinicos.Atletas_ProvasAgendadas TO 'atleta'@'localhost';  
SHOW GRANTS FOR 'atleta'@'localhost';
```

```
CREATE USER 'profissional'@'localhost' IDENTIFIED BY 'prof';  
GRANT SELECT ON Profissionais_Testes_ProxMes TO 'profissional'@'localhost';  
GRANT EXECUTE ON PROCEDURE Passar_Testes_a_Realizado TO 'profissional'@'localhost';  
GRANT INSERT, UPDATE ON TestesClinicos.TesteClinico TO 'profissional'@'localhost';  
SHOW GRANTS FOR 'profissional'@'localhost';
```

2.3.4 Povoamento

De forma a poder testar os requisitos e a finalidade da base de dados foi feito um povoamento com alguns dados. Após obtenção de uma lista de atletas e indicação das modalidades e tipos de teste, foi feito um script em Python para gerar a restante informação, nomeadamente as provas das várias modalidades e os testes. Foram incorporadas algumas restrições por forma a podermos testar o que pretendíamos e garantir a consistência da base de dados. Por exemplo, o script garante que o mesmo médico não realiza 2 consultas à mesma hora. O script de povoamento gerado encontra-se no anexo B.

2.3.5 Estimativa do Espaço Ocupado

O espaço ocupado por cada tipo de dados é o seguinte:

- Int - 4 bytes
- Varchar(n) - 1 + n bytes
- Char - 1 byte
- Date - 3 bytes
- Datetime - 8 bytes
- TinyInt - 1 byte
- Float - 4 bytes

Tendo em conta estes valores e os tipos de dados dos atributos de cada tabela, obtemos o total de bytes por linha para cada entidade.

Tabela	Total de bytes por linha
Atleta	407
Modalidade	50
Prova	16
TesteClinico	71
Profissional	50
Total de bytes	598

Tabela 2.4: Espaço ocupado por linha para cada tabela

Fazendo um cálculo com a quantidade atual de atletas, modalidades, profissionais, testes e provas obtemos os seguintes valores de espaço ocupado para cada tabela do modelo físico.

Tabela	Total de bytes por tabela
Atleta	105*407
Modalidade	15*50
Prova	15*16
TesteClinico	71*35
Profissional	31*50
Total de bytes	47760

Tabela 2.5: Espaço ocupado por tabela

Segundo os dados fornecidos pela AAB, há um saldo positivo de 20 atletas a cada ano, ou seja, a associação tem cerca de mais 20 atletas por ano. Admitindo que são marcados nesse ano cerca de 6 testes por atleta, e admitindo que o número de modalidades se mantém e o número de provas por ano é cerca de 100, então o aumento do espaço ocupado na base de dados num ano é:

$$407*20 + 71*6*20 + 100*16 = 18260 \text{ bytes}$$

Isto significa que volume de dados aumenta cerca de 38% por ano.

Capítulo 3

Base de Dados Não Relacional

A administração da associação de atletismo de Braga decidiu migrar os dados da base de dados relacional para uma BD não relacional, concretamente a **MongoDB**, uma vez que esta permite escalonamento horizontal dos dados e torna a procura de grandes quantidades de informação mais eficiente.

3.1 Introdução

A MongoDB é uma base de dados NoSQL, orientada a documentos, bastante flexível e de fácil escalonamento. Para além disso uma colecção pode conter vários documentos e cada documento pode ter um tamanho diferente e conter sub documentos com um esquema próprio, a que chamamos documentos incorporados.

3.2 Migração dos dados para MongoDB

Para efectuarmos a migração dos dados em MySQL para MongoDB, criámos um script em Python (que se encontra em anexo) para tornar mais rápido e menos trabalhoso todo o processo. Recorremos a bibliotecas próprias do Python para escrever a base de dados NoSQL.

Após o processo de migração a base de dados vai conter duas colecções: TesteClinico e Prova.

Começou-se por estabelecer conexão com o MongoDB e proceder-se a leitura de 2 queries em SQL, (guardadas nos documentos provas.sql e testes.sql) que vão permitir a construção das duas colecções.

A primeira permite criar uma tabela que associa cada teste clínico ao seu tipo, à data de realização e ao preço. E para cada um dos testes realizados, é associado o atleta que foi submetido ao teste bem como as restantes informações pertinentes sobre ele, como o nome, idade, sexo, peso, morada e a modalidade que pratica. Para além disto, guardamos também a informação relacionada com o profissional que realiza o teste clínico.

```
SELECT tc.nome, tc.realizado, tc.preço, tc.data_hora, atl.nome AS 'nome_atleta', atl.sexo AS  
'sexo_atleta', m.nome AS 'modalidade_atleta', atl.peso AS 'peso_atleta', atl.morada as  
'morada_atleta', atl.DOB as 'DOB_atleta', prof.nome as 'nome_profissional' FROM TesteClinico tc  
INNER JOIN Atleta atl ON atl.id_atleta = tc.id_atleta  
INNER JOIN Profissional prof ON prof.id_profissional = tc.id_profissional  
INNER JOIN Modalidade m ON atl.id_modalidade = m.id_modalidade;
```

Figura 3.1: Query no ficheiro testes.sql

A query presente no ficheiro provas.sql por sua vez permitiu criar uma tabela que a cada prova associa o nome da modalidade, a duração e a data/hora de realização.

```
SELECT p.data_hora, p.duração, m.nome FROM Prova p
INNER JOIN Modalidade m ON p.id_modalidade = m.id_modalidade;
```

Figura 3.2: Query no ficheiro testes.sql

Após a leitura da Base de Dados em MySQL, o nosso script converte as informações estruturadas por documentos para MongoDB, originado as duas colecções.

A base de dados resultante da migração foi obtida através de documentos incorporados. Na imagem, abaixo, facilmente se verifica que Atleta e Profissional correspondem a documentos incorporados num outro documento.

```
{
  "_id" : ObjectId("5e0f78b4af65c5ce68f0857b"),
  "tipo" : "Análises",
  "realizado" : false,
  "preço" : 14,
  "data_hora" : ISODate("2021-10-14T11:00:00Z"),
  "atleta" : {
    "nome" : "Joan Manuel Cedeño Rodriguez",
    "sexo" : "M",
    "modalidade" : "Barreiras 100m",
    "peso" : 91,
    "morada" : "Mire de Tibães, Braga",
    "DOB" : ISODate("1978-11-10T00:00:00Z")
  },
  "profissional" : {
    "nome" : "Tarciso"
  }
}
```

Figura 3.3: Estrutura da informação na colecção TesteClinico na base de dados MongoDB

3.3 Queries

Apresentamos em seguida algumas queries, equivalentes as definidas e apresentadas no modelo relacional e que permitem comprovar a operacionalidade do sistema não relacional.

```
db.TesteClinico.find({}, {"atleta.modalidade": true, "_id": false}).pretty()
```

Figura 3.4: Query que permite consultar as modalidades praticadas pelos atletas registados na base de Dados

```
db.TesteClinico.find({"realizado":true}, {"atleta.nome":1, "_id":0}).pretty()
```

Figura 3.5: Query que permite listar o nome dos atletas que foram submetidos a testes clínicos

```
db.TesteClinico.find({"realizado":false}, {"atleta.nome":1, "_id":0, "tipo":1 }).pretty()
```

Figura 3.6: Query que permite verificar os testes agendados, apresentando o nome dos atletas que tem testes agendados e o nome do respectivo teste

```
db.TesteClinico.find({"atleta.modalidade" : "Salto em Altura"},{"_id":0,"tipo":1 }).pretty()
```

Figura 3.7: Query que permite verificar o tipo de testes que os atletas da modalidade "Salto em Altura" realizam

No Apêndice E apresentamos um conjunto de queries, em MongoDB, que consideramos pertinentes tendo em conta o conteúdo da nossa base de dados, e que são equivalentes às queries em *MySQL* definidas anteriormente.

Capítulo 4

Conclusão

Este trabalho permitiu consolidar toda a matéria lecionada na UC de Bases de Dados. Ao passar por todas as fases da implementação de uma BD aplicadas a uma situação real pudémos ficar com uma noção do que é ser um utilizador comum de uma base de dados e quais os procedimentos necessários para criar e manter uma base de dados de acordo com o que é pedido.

A parte que inicialmente trouxe mais problemas foi a definição do modelo conceptual, que demorou algum tempo até chegar ao seu estado final. Tendo este modelo decidido, os restantes passos foram mais fáceis. Percebemos, com esta dificuldade, que este é uma etapa crucial na implementação e que, se não for bem feita e projetada, pode dar muitos problemas no futuro, nomeadamente a necessidade de se implementar de novo a base de dados.

Pensamos que a nível futuro seria interessante aumentar o volume de dados da base de dados para poder fazer uma melhor comparação da eficiência garantida pelas duas vertentes abordadas neste trabalho.

Apêndice A

Script SQL de Preparação da Base de Dados MySQL para utilização

```
-- Configuração necessária para poder apagar certos dados da base de dados
SET SQL_SAFE_UPDATES = 0;

-- View 1) Vista que lista os atletas que têm testes agendados nas próximas duas semanas
-- ↳ (excepto testes de controlo Anti-Doping!)
CREATE VIEW Atletas_Testes_Prox2Semanas AS (
    SELECT a.nome AS Atleta,
           t.nome AS 'Tipo de Teste',
           t.data_hora AS 'Data/Hora'
    FROM Atleta a, TesteClinico t
    WHERE a.id_atleta = t.id_atleta
          AND t.data_hora BETWEEN CURRENT_DATE() AND DATE_ADD(CURRENT_DATE(),
          ↳ INTERVAL 14 DAY)
          AND t.nome NOT LIKE '%Controlo Anti-Doping%'
);

-- View 2) Vista que lista todos os profissionais que têm testes marcados durante o
-- ↳ próximo mês
CREATE VIEW Profissionais_Testes_ProxMes AS (
    SELECT p.nome AS 'Médico',
           t.data_hora AS 'Data/Hora'
    FROM Profissional p, TesteClinico t
    WHERE p.id_profissional = t.id_profissional
          AND t.data_hora BETWEEN CURRENT_DATE()
          AND DATE_ADD(CURRENT_DATE(), INTERVAL 30 DAY)
);

-- View 3) Vista que associa a todos os profissionais quais os atletas a quem esse
-- ↳ profissional já realizou testes
CREATE VIEW Profissionais_Atletas AS (
    SELECT p.nome AS 'Médico',
           a.nome AS 'Atleta'
```



```

        FROM Profissional p, Atleta a, TesteClinico t
        WHERE t.id_profissional = p.id_profissional
              AND t.id_atleta = a.id_atleta
        ORDER BY p.nome ASC
    );

-- View 4) Vista que associa a cada atleta as provas de atletismo que este vai realizar
CREATE VIEW Atletas_ProvasAgendadas AS (
    SELECT a.nome AS Atleta,
           m.nome AS Modalidade,
           p.data_hora AS 'Data/Hora'
    FROM Atleta a, Modalidade m, Prova p
    WHERE m.id_modalidade = a.id_modalidade
          AND m.id_modalidade = p.id_modalidade
);

-- View 5) Vista que associa a cada teste o atleta e o profissional envolvido
CREATE VIEW Teste_Profissional_Atleta AS (
    SELECT a.nome as Atleta,
           t.nome as 'Teste',
           p.nome as 'Médico'
    FROM Atleta a, TesteClinico t, Profissional p
    WHERE t.id_profissional = p.id_profissional
          AND t.id_atleta = a.id_atleta
    ORDER BY a.nome ASC
);

-- Preparação da Base de Dados para os utilizadores
FLUSH PRIVILEGES;

DELIMITER //
CREATE PROCEDURE Eliminar_Testes_Nao_Realizados()
BEGIN
    DELETE FROM TesteClinico WHERE data_hora < NOW() AND realizado = 0;
END //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE Passar_Testes_a_Realizado(id_prof int, dataHora datetime)
BEGIN
    UPDATE TesteClinico SET realizado = 1 WHERE data_hora = dataHora AND id_prof =
        ↪ id_profissional;
END //
DELIMITER ;

-- Apagar os utilizadores se já existirem
DROP USER IF EXISTS 'atleta'@'localhost';
DROP USER IF EXISTS 'profissional'@'localhost';

```

```

-- Criar os utilizadores da base de dados e dar-lhes privilégios
CREATE USER 'atleta'@'localhost' IDENTIFIED BY 'atleta';
GRANT SELECT ON Atletas_Testes_Prox2Semanas TO 'atleta'@'localhost';
GRANT SELECT ON TestesClinicos.Atletas_ProvasAgendadas TO 'atleta'@'localhost';
SHOW GRANTS FOR 'atleta'@'localhost';

CREATE USER 'profissional'@'localhost' IDENTIFIED BY 'prof';
GRANT SELECT ON Profissionais_Testes_ProxMes TO 'profissional'@'localhost';
GRANT EXECUTE ON PROCEDURE Passar_Teste_a_Realizado TO 'profissional'@'localhost';
GRANT INSERT, UPDATE ON TestesClinicos.TesteClinico TO 'profissional'@'localhost';
SHOW GRANTS FOR 'profissional'@'localhost';

```

Apêndice B

Script SQL de Povoamento das tabelas SQL

```
INSERT INTO Modalidade (nome) VALUES ('Salto em Comprimento');
INSERT INTO Modalidade (nome) VALUES ('Salto em Altura');
INSERT INTO Modalidade (nome) VALUES ('Estafetas');
INSERT INTO Modalidade (nome) VALUES ('Sprint 60m');
INSERT INTO Modalidade (nome) VALUES ('Sprint 100m');
INSERT INTO Modalidade (nome) VALUES ('Sprint 200m');
INSERT INTO Modalidade (nome) VALUES ('Sprint 400m');
INSERT INTO Modalidade (nome) VALUES ('Barreiras 60m');
INSERT INTO Modalidade (nome) VALUES ('Barreiras 100m');
INSERT INTO Modalidade (nome) VALUES ('Barreiras 400m');
INSERT INTO Modalidade (nome) VALUES ('Lançamento do Peso');
INSERT INTO Modalidade (nome) VALUES ('Lançamento do Martelo');
INSERT INTO Modalidade (nome) VALUES ('Lançamento do Dardo');
INSERT INTO Modalidade (nome) VALUES ('Lançamento do Disco');
INSERT INTO Modalidade (nome) VALUES ('Maratona');
INSERT INTO Profissional (nome) VALUES ('Constantino');
INSERT INTO Profissional (nome) VALUES ('Tarciso');
INSERT INTO Profissional (nome) VALUES ('Olivia');
INSERT INTO Profissional (nome) VALUES ('Derick');
INSERT INTO Profissional (nome) VALUES ('Khadija');
INSERT INTO Profissional (nome) VALUES ('Papa-Ndiogou');
INSERT INTO Profissional (nome) VALUES ('Marius');
INSERT INTO Profissional (nome) VALUES ('Manish');
INSERT INTO Profissional (nome) VALUES ('Vinildo');
INSERT INTO Profissional (nome) VALUES ('Humaira');
INSERT INTO Profissional (nome) VALUES ('Rayna');
INSERT INTO Profissional (nome) VALUES ('Rafia');
INSERT INTO Profissional (nome) VALUES ('Nelson');
INSERT INTO Profissional (nome) VALUES ('Jamie');
INSERT INTO Profissional (nome) VALUES ('Maura');
INSERT INTO Profissional (nome) VALUES ('Manteg');
INSERT INTO Profissional (nome) VALUES ('Nayra');
```

```

INSERT INTO Profissional (nome) VALUES ('Adame');
INSERT INTO Profissional (nome) VALUES ('Lisandra');
INSERT INTO Profissional (nome) VALUES ('Enzo');
INSERT INTO Profissional (nome) VALUES ('Helaine');
INSERT INTO Profissional (nome) VALUES ('Kadirkhan');
INSERT INTO Profissional (nome) VALUES ('Lavonne');
INSERT INTO Profissional (nome) VALUES ('Seerat');
INSERT INTO Profissional (nome) VALUES ('Mansirat');
INSERT INTO Profissional (nome) VALUES ('Steven');
INSERT INTO Profissional (nome) VALUES ('Brianna');
INSERT INTO Profissional (nome) VALUES ('Stanislawa');
INSERT INTO Profissional (nome) VALUES ('Niah');
INSERT INTO Profissional (nome) VALUES ('Florinda');
INSERT INTO Profissional (nome) VALUES ('Patrício');
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Alberto Rui Vilaverde Monteiro da Silva', '96', '203', 'Merelim (São Pedro) e
→ Frossos, Braga', '1979-7-26', 5);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Alexandre Esteves Miranda', '107', '212', 'Escudeiros e Penso (Santo Estêvão e São
→ Vicente), Braga', '1991-4-2', 13);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Ana Luísa Pereira de César', '86', '173', 'Pedralva, Braga', '1989-6-25', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Ana Margarida Reis Maia Campos', '96', '232', 'Santa Lucrecia de Algeriz e Navarra,
→ Braga', '1986-11-3', 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Ana Rita Miranda Rosendo', '105', '180', 'Braga (Maximinos, Sé e Cividade), Braga',
→ '1990-5-29', 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'André Alves dos Santos', '101', '203', 'Sobrepota, Braga', '1993-10-29', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'André Bernardo Coelho Nunes', '108', '205', 'Guisande e Oliveira (São Pedro),
→ Braga', '1977-11-25', 9);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'André Daniel Freitas Ferreira', '93', '223', 'Sobrepota, Braga', '1988-9-10', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'André Sousa Figueiredo', '80', '220', 'Esporões, Braga', '1988-10-14', 15);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Beatriz Freitas Rocha', '55', '158', 'Celeirós, Aveleda e Vimieiro, Braga',
→ '1978-7-3', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Bernardo José Barbosa Alves', '76', '188', 'Priscos, Braga', '1979-12-31', 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Bruno Alexandre Dias Novais de Sousa', '125', '215', 'Ruilhe, Braga', '1988-11-14',
→ 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Bruno Alves Martins Carvalho', '86', '218', 'São Victor, Braga', '1985-12-9', 2);

```

```

INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Bruno Miguel Faria Fernandes', '116', '207', 'Nogueiró e Tenões, Braga',
→ '1977-2-12', 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Carlos César Madureira Romano', '114', '230', 'Arentim e Cunha, Braga', '1986-6-6',
→ 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Catarina Freitas da Cruz', '59', '198', 'Guisande e Oliveira (São Pedro), Braga',
→ '1986-2-20', 15);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Celso André de Carvalho Rodrigues', '114', '188', 'Esporões, Braga', '1981-1-7',
→ 11);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Daniela Eduarda Cruz Fernandes', '93', '208', 'Escudeiros e Penso (Santo Estêvão e
→ São Vicente), Braga', '1975-5-12', 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Daniela Sofia Pereira Arruda', '112', '212', 'Arentim e Cunha, Braga', '1994-4-28',
→ 3);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Diogo Araújo Ferreira', '85', '204', 'Ferreiros e Gondizalves, Braga', '1989-2-16',
→ 6);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Diogo Miguel Alves Rocha', '119', '236', 'Lomar e Arcos, Braga', '1979-1-31', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Diogo Paulo Lopes de Vasconcelos', '110', '201', 'Padim da Graça, Braga',
→ '1985-8-31', 5);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Diogo Pinto Ribeiro', '98', '214', 'Merelim (São Pedro) e Frossos, Braga',
→ '1992-11-11', 3);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Duarte Manuel Vilar de Oliveira', '114', '208', 'Celeirós, Aveleda e Vimieiro,
→ Braga', '1978-9-19', 5);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Eduardo José Azevedo Ferreira Araújo', '98', '192', 'Este (São Pedro e São Mamede),
→ Braga', '1987-3-31', 3);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Fábio Fernandes Silva', '101', '216', 'Guisande e Oliveira (São Pedro), Braga',
→ '1985-9-8', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Fernanda Manuela Alves Dias', '87', '207', 'Ferreiros e Gondizalves, Braga',
→ '1984-6-14', 11);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Filipe Gonçalves de Macedo', '130', '204', 'Braga (Maximinos, Sé e Cividade),
→ Braga', '1979-10-1', 7);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Filipe José Vieira Barbosa', '84', '188', 'Cabreiros e Passos (São Julião), Braga',
→ '1977-3-14', 5);

```

```

INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Francisco Egídio Coutinho Martins Correia', '95', '196', 'Tebosa, Braga',
→ '1988-12-30', 3);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Gonçalo Leal da Mota Meireles Moreira', '98', '198', 'Real, Dume e Semelhe, Braga',
→ '1984-11-12', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Goncalo Nuno Esperança Quesado', '79', '213', 'Lamas, Braga', '1989-11-12', 11);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Guilherme de Araujo Soares', '132', '253', 'Sobreposta, Braga', '1976-7-14', 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Guilherme João Cerqueira da Mota Miranda', '50', '221', 'Priscos, Braga',
→ '1985-8-7', 6);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Helena Poças Martins', '139', '203', 'Merelim (São Pedro) e Frossos, Braga',
→ '1990-2-6', 13);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Henrique Veiga da Paz', '102', '200', 'Crespos e Pousada, Braga', '1983-2-18', 10);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Hugo André Coelho Cardoso', '94', '226', 'Esporões, Braga', '1989-7-6', 13);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Hugo André da Silva Veloso Moreira', '105', '185', 'Priscos, Braga', '1988-3-8', 8);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Hugo Fernandes Matias', '93', '187', 'Nogueira, Fraião e Lamações, Braga',
→ '1991-3-12', 8);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Hugo Filipe Duarte Carvalho', '80', '207', 'Celeirós, Aveleda e Vimieiro, Braga',
→ '1990-4-10', 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Joana Esteves Dantas', '86', '177', 'Ruilhe, Braga', '1994-8-1', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Joan Manuel Cedeño Rodriguez', '68', '224', 'Mire de Tibães, Braga', '1978-11-10',
→ 9);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'João Bernardo Coutinho Barreiros de Freitas', '68', '198', 'Esporões, Braga',
→ '1988-12-25', 8);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'João Carlos Pereira Rodrigues', '122', '173', 'Celeirós, Aveleda e Vimieiro, Braga',
→ '1990-4-18', 13);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'João da Cunha e Costa', '82', '211', 'Pedralva, Braga', '1983-11-1', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'João Duarte Vila Verde Silva', '77', '183', 'Merelim (São Paio), Panoias e Parada de
→ Tibães, Braga', '1976-1-9', 7);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'João Manuel da Rocha Rodrigues', '84', '197', 'Sobreposta, Braga', '1993-12-13', 8);

```

```

INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Manuel Silva Antunes', '141', '198', 'Crespos e Pousada, Braga', '1983-6-10',
↳ 14);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Nuno Brandão Silva', '113', '191', 'Figueiredo, Braga', '1994-2-2', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Nuno Cardoso Gonçalves de Abreu', '131', '180', 'Lomar e Arcos, Braga',
↳ '1984-9-11', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Pedro Dias Fernandes', '71', '198', 'Sobreposta, Braga', '1982-3-3', 7);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Pedro Magalhães Teixeira', '84', '214', 'Merelim (São Pedro) e Frossos, Braga',
↳ '1984-9-5', 10);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Pedro Oliveira Meira', '82', '193', 'Sobreposta, Braga', '1995-4-18', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Pedro Rodrigues Azevedo', '138', '202', 'Priscos, Braga', '1995-9-13', 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'João Tiago Oliveira Sousa Nogueira Lopes', '100', '211', 'Merelim (São Paio),
↳ Panoias e Parada de Tibães, Braga', '1984-8-12', 10);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Joaquim Francisco Barata Simões', '111', '217', 'Espinho, Braga', '1980-2-23', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Jorge Manuel Loureiro Vieira', '138', '191', 'Palmeira, Braga', '1987-10-15', 3);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Jorge Miguel da Silva Brandão Gonçalves', '105', '185', 'Tebosa, Braga',
↳ '1984-6-21', 8);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'José Diogo Xavier Monteiro', '63', '210', 'Priscos, Braga', '1981-8-16', 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Luís Alberto Diogo Pereira de Lima', '129', '189', 'Sequeira, Braga', '1985-8-21',
↳ 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Luís Daniel da Silva Félix', '74', '222', 'Ruilhe, Braga', '1984-3-24', 8);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Luis Filipe da Silva Costa', '98', '215', 'Braga (São José de São Lázaro e São João
↳ do Souto), Braga', '1994-4-1', 6);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Luís Miguel Aquino Silva', '82', '214', 'São Vicente, Braga', '1987-1-28', 9);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Luís Miguel Arrepiá Ferreira', '87', '220', 'Cabreiros e Passos (São Julião),
↳ Braga', '1977-11-24', 15);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Luís Pedro da Silva Vila', '87', '172', 'Pedralva, Braga', '1975-6-2', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
↳ 'Marco Alexandre Félix de Lima', '71', '209', 'Adaúfe, Braga', '1983-11-23', 5);

```



```

INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Marco Antonio Rodrigues Sampaio', '136', '226', 'Guisande e Oliveira (São Pedro),
→ Braga', '1979-10-10', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Maria Francisca Tavares Martins Fernandes', '113', '165', 'Crespos e Pousada,
→ Braga', '1988-8-23', 6);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Maria Inês de Azevedo Mourao Alves', '104', '191', 'Palmeira, Braga', '1980-6-14',
→ 11);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Maria José Borges Pires', '112', '180', 'Real, Dume e Semelhe, Braga', '1979-8-30',
→ 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Maria Laura de Araújo Barbosa', '118', '216', 'Esporões, Braga', '1991-6-5', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Maria Luísa Faria Silva', '113', '227', 'Ferreiros e Gondizalves, Braga',
→ '1989-6-26', 7);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Maria Miguel Albuquerque Regueiras', '111', '213', 'Real, Dume e Semelhe, Braga',
→ '1990-7-24', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Mariana Gonçalves Marques', '121', '204', 'Figueiredo, Braga', '1989-4-19', 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Mariana Lino Lopes Costa', '95', '224', 'Braga (São José de São Lázaro e São João do
→ Souto), Braga', '1987-1-8', 11);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Miguel Cardoso Pereira', '95', '215', 'Celeirós, Aveleda e Vimieiro, Braga',
→ '1980-6-2', 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Miguel Rúben Silva de Oliveira', '92', '171', 'Ferreiros e Gondizalves, Braga',
→ '1982-10-10', 10);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Néilson Miguel de Oliveira Estevão', '115', '212', 'Celeirós, Aveleda e Vimieiro,
→ Braga', '1991-5-25', 15);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Nuno Azevedo Alves da Cunha', '97', '189', 'Nogueiró e Tenões, Braga', '1988-8-2',
→ 11);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro André Correia Neves Parpot', '84', '195', 'Sobreposta, Braga', '1995-9-30',
→ 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Daniel Fernandes Abreu', '91', '218', 'Real, Dume e Semelhe, Braga',
→ '1993-8-16', 5);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Filipe Carvalho Barbosa', '84', '207', 'São Vicente, Braga', '1981-6-9', 1);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Gomes Machado Monteiro Fernandes', '89', '186', 'Braga (Maximinos, Sé e
→ Cividade), Braga', '1976-3-12', 1);

```



```

INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Miguel Borges Rodrigues', '96', '213', 'Merelim (São Pedro) e Frossos, Braga',
→ '1995-1-5', 2);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Miguel Queiros Gomes', '107', '194', 'Real, Dume e Semelhe, Braga',
→ '1982-10-13', 9);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Nuno Fernandes Freitas', '98', '186', 'Guisande e Oliveira (São Pedro),
→ Braga', '1989-4-11', 10);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Rafael Bernardo Medeiros', '77', '193', 'Crespos e Pousada, Braga',
→ '1994-2-26', 6);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Pedro Rafael Martins de Oliveira', '95', '200', 'Morreira e Trandeiras, Braga',
→ '1992-8-29', 6);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Peter Ferreira Vala', '84', '240', 'Lomar e Arcos, Braga', '1975-9-2', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Rafael Antunes Simões', '94', '222', 'Padim da Graça, Braga', '1977-4-28', 9);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Renato Jorge Cruzinha da Silva', '102', '224', 'Nogueiró e Tenões, Braga',
→ '1981-4-9', 11);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Ricardo José Santos Veloso', '99', '194', 'Mire de Tibães, Braga', '1984-9-28', 7);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Ricardo Miguel da Silva Pereira', '120', '187', 'Lomar e Arcos, Braga', '1990-6-28',
→ 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Ricardo Vieira Carvalho', '59', '190', 'Lamas, Braga', '1976-5-30', 9);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Rui Alexandre Dias Neto', '91', '176', 'Cabreiros e Passos (São Julião), Braga',
→ '1975-8-16', 12);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Rui Filipe Chaves', '87', '195', 'Ruilhe, Braga', '1976-7-16', 8);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Rui Filipe Traila de Sousa', '86', '244', 'Ruilhe, Braga', '1978-2-3', 4);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Rui Miguel Pimenta e Cunha', '119', '210', 'Crespos e Pousada, Braga', '1979-7-29',
→ 15);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Rui Nuno Borges Cruz Oliveira', '77', '216', 'Gualtar, Braga', '1989-10-3', 14);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Rui Pedro Neto Reis', '104', '169', 'Real, Dume e Semelhe, Braga', '1992-7-2', 3);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Shahzod Yusupov', '93', '207', 'Lamas, Braga', '1990-4-5', 14);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('F',
→ 'Tânia Filipa Amorim da Rocha', '116', '178', 'Merelim (São Paio), Panoias e Parada
→ de Tibães, Braga', '1986-6-14', 2);

```

```

INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Tiago Gabriel Pereira da Silva', '81', '201', 'Tebosa, Braga', '1984-10-18', 13);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Tiago Gomes Campos', '90', '183', 'Figueiredo, Braga', '1984-8-17', 10);
INSERT INTO Atleta (sexo, nome, peso, altura, morada, DOB, id_modalidade) VALUES ('M',
→ 'Válter Ferreira Picas Carvalho', '103', '209', 'Santa Lucrécia de Algeriz e Navarra,
→ Braga', '1992-11-9', 5);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-1-4T10-11', 1699,
→ 1);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-4-10T9-47', 1755,
→ 2);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-2-13T13-7', 1471,
→ 3);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-11-23T17-42', 1156,
→ 4);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-12-27T12-4', 1260,
→ 5);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2022-6-8T14-40', 1968,
→ 6);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2021-10-6T11-51', 64, 7);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2022-9-11T14-37', 104,
→ 8);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-7-5T11-32', 314, 9);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-4-26T16-36', 875,
→ 10);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-12-5T12-48', 1576,
→ 11);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2021-10-19T11-40', 2042,
→ 12);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2020-10-27T9-33', 1469,
→ 13);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2021-8-10T9-14', 668,
→ 14);
INSERT INTO Prova (data_hora, duração, id_modalidade) VALUES ('2022-8-26T16-34', 1286,
→ 15);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 11, '2019-10-13T12:30', 11, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 12, '2019-04-02T14:15', 11, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2021-11-08T13:30', 86, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2019-05-01T14:00', 86, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2019-08-21T09:00', 86, 1);

```

```

INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2021-06-03T09:30', 2, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 7, '2019-08-14T11:30', 55, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2021-12-14T15:00', 84, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2020-09-18T16:00', 84, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2020-03-01T17:15', 16, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2021-09-27T11:00', 16, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2019-01-31T11:15', 31, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 9, '2021-11-15T11:00', 77, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 6, '2021-07-01T09:45', 44, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 8, '2020-05-22T11:45', 44, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 9, '2021-06-06T15:30', 44, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2020-01-10T17:30', 103, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2020-01-25T17:30', 103, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 12, '2020-12-04T15:30', 5, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 15, '2019-01-09T09:30', 5, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 8, '2021-09-11T15:30', 98, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 7, '2020-04-19T17:30', 15, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2019-06-28T09:00', 15, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 12, '2020-05-21T13:30', 15, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2021-07-06T09:15', 101, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 15, '2019-08-05T14:00', 101, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2020-12-20T12:00', 105, 1);

```

```

INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 6, '2020-05-31T11:15', 105, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2021-11-06T10:45', 105, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2020-01-14T13:30', 35, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2021-09-11T14:15', 35, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 5, '2019-09-05T15:30', 102, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2021-10-26T12:00', 72, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2021-03-22T12:30', 72, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 7, '2019-12-24T12:15', 90, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 12, '2020-02-19T16:15', 90, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 15, '2019-04-06T16:00', 66, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2020-05-09T15:30', 66, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2020-02-10T09:45', 66, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2019-07-13T15:45', 98, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2020-06-27T11:30', 98, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2020-09-05T09:00', 4, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 6, '2020-06-01T10:15', 4, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2020-01-11T09:30', 86, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2019-11-09T15:15', 86, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 7, '2020-09-02T11:15', 86, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2021-12-25T09:00', 62, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2021-12-23T15:15', 62, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 9, '2021-02-03T17:15', 62, 1);

```

```

INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2020-12-23T13:30', 26, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2020-03-25T16:15', 26, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 8, '2019-11-17T09:30', 81, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2021-08-20T12:45', 81, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2019-12-11T11:45', 81, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2020-01-27T10:00', 11, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2021-06-19T16:30', 11, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2021-06-27T10:15', 11, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2021-03-01T14:15', 3, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2019-12-10T15:00', 53, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 15, '2019-07-02T16:30', 53, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 7, '2020-10-28T16:45', 53, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 6, '2020-04-13T10:30', 4, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 14, '2019-01-12T09:15', 4, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 6, '2019-06-15T17:45', 89, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2019-09-28T15:30', 89, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2021-02-27T10:30', 89, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2019-03-30T17:30', 13, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 7, '2020-06-06T17:15', 13, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 12, '2021-11-08T14:00', 23, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2020-01-08T12:30', 81, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 1, 12, '2019-04-12T12:15', 81, 1);

```



```

INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 7, '2020-10-05T15:00', 81, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2021-07-23T11:15', 75, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2021-01-02T09:00', 4, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2021-03-25T11:45', 4, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2021-11-11T12:15', 54, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 7, '2021-04-18T15:00', 54, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 8, '2021-06-04T15:45', 53, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2020-07-08T15:15', 53, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 9, '2020-07-30T10:30', 53, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 9, '2021-01-20T16:15', 68, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 7, '2019-10-11T15:00', 68, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 12, '2019-07-24T15:45', 92, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 12, '2021-08-06T11:30', 92, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 10, '2021-05-25T17:00', 53, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 8, '2021-06-25T16:00', 45, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2021-10-11T13:45', 33, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 11, '2021-07-02T15:30', 6, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 6, '2021-04-12T11:15', 6, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 15, '2019-01-14T15:45', 79, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 13, '2021-12-22T10:45', 79, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 5, '2019-03-05T12:15', 93, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
→ VALUES ('Controlo Anti-Doping', 0, 14, '2021-03-27T16:30', 93, 1);

```

```
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
↪ VALUES ('Controlo Anti-Doping', 0, 11, '2021-05-23T14:00', 93, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
↪ VALUES ('Controlo Anti-Doping', 1, 15, '2020-01-03T13:15', 65, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
↪ VALUES ('Controlo Anti-Doping', 1, 13, '2020-01-04T14:15', 40, 1);
INSERT INTO TesteClinico (nome, realizado, preço, data_hora, id_atleta, id_profissional)
↪ VALUES ('Controlo Anti-Doping', 0, 10, '2020-01-05T12:45', 50, 1);
```

Apêndice C

Script contendo todas as queries SQL construídas

```
-- Configuração necessária para poder apagar certos dados da base de dados
SET SQL_SAFE_UPDATES = 0;

-- Query 1) Listar todas as modalidades registradas na base de dados
SELECT nome AS 'Atleta'
  FROM Modalidade;

-- Query 2) Listar os atletas registrados
SELECT nome as 'Atleta'
  FROM Atleta;

-- Query 3) Listar os atletas e as suas modalidades
SELECT a.nome AS 'Atleta', m.nome AS 'Modalidade'
  FROM Atleta a, Modalidade m
 WHERE a.id_modalidade = m.id_modalidade;

-- Query 4) Listar os nome dos atletas de uma dada modalidade, por exemplo: 'Salto em
↳ Altura'
SELECT a.nome AS 'Atleta', m.nome AS 'Modalidade'
  FROM Atleta a, Modalidade m
 WHERE a.id_modalidade = m.id_modalidade
       AND m.nome = 'Salto em Altura';

-- Query 5) Listar o tipo de testes que já foram realizados
SELECT nome AS 'Tipo de Teste'
  FROM TesteClinico
 GROUP BY nome;

-- Query 6) Verificar os testes agendados, retornando o nome dos atletas que tem testes
↳ agendados
SELECT a.nome AS 'Atleta'
  FROM TesteClinico t, Atleta a
```



```

WHERE a.id_atleta = t.id_atleta
      AND t.realizado = 0
GROUP BY a.nome;

-- Query 7) O nome dos atletas que foram submetidos a testes clinicos e a quantos testes
↳ foram submetidos
SELECT DISTINCT a.nome AS 'Atleta',
                (SELECT COUNT(*) FROM TesteClinico t WHERE a.id_Atleta = t.id_atleta AND
                ↳ t.realizado = 1) AS 'Número de Testes Realizados'
FROM Atleta a
WHERE (SELECT COUNT(*) FROM TesteClinico t WHERE a.id_Atleta = t.id_atleta AND
↳ t.realizado = 1) >= 1;

-- Query 8) Verificar os dias em que um determinado atleta (por exemplo "LUIS PEDRO
↳ BARBOSA FERREIRO") tem algum teste clinico
SELECT DISTINCT t.data_hora AS 'Data/Hora'
FROM Atleta a, TesteClinico t
WHERE a.nome = 'Luís Pedro Barbosa Ferreira'
      AND a.id_atleta = t.id_atleta
      AND t.realizado = 0;

-- Query 9) Nome dos profissionais que trabalham na clínica
SELECT nome AS 'Médico'
FROM Profissional;

-- Query 10) Número de profissionais que trabalham na clínica
SELECT DISTINCT COUNT(*)
FROM Profissional;

-- Query 11) Número dos profissionais que já realizaram pelo menos 1 teste clínico
SELECT DISTINCT COUNT(*) AS 'Número de Profissionais'
FROM Profissional p
WHERE (SELECT COUNT(*) FROM TesteClinico t WHERE p.id_profissional =
↳ t.id_profissional) >= 1;

-- Query 12) Atletas cujo o peso se encontra dentro de um intervalo (i.e., têm um peso
↳ saudável). Por exemplo 60 e 90 kg
SELECT a.nome
FROM Atleta a
WHERE a.peso >= 60
      AND a.peso <= 90;

```

Apêndice D

Script para migração dos dados de SQL para MongoDB

```
#!/bin/python3

import mysql.connector
from pymongo import MongoClient
from datetime import date
from datetime import datetime
from getpass import getpass
import sys

if __name__ == "__main__":
    print("Connecting to MySQL database...")

    mydb = mysql.connector.connect(
        host = "localhost",
        user = "root",
        passwd = getpass("MySQL password: "),
        database = "TestesClinicos",
        auth_plugin = "mysql_native_password"
    )

    cursor = mydb.cursor()

    print("Querying MySQL TestesClinicos...")

    query_prova = open("prova.sql").read()
    query_teste = open("teste.sql").read()

    cursor.execute(query_teste)
    results_teste = cursor.fetchall()

    cursor.execute(query_prova)
    results_prova = cursor.fetchall()
```

```

print("Converting entries to documents...")

converted_results_prova = list(map(lambda entry: {"data_hora": entry[0], "duração":
→ entry[1], "modalidade": entry[2]}, results_prova))
converted_results_teste = list(map(lambda entry: {"tipo": entry[0], "realizado":
→ bool(entry[1]), "preço": entry[2], "data_hora": entry[3], "atleta": {"nome":
→ entry[4], "sexo": entry[5], "modalidade": entry[6], "peso": entry[7], "morada":
→ entry[8], "DOB": datetime.combine(entry[9], datetime.min.time())}, "profissional"
→ : {"nome": entry[10]}}}, results_teste))

print("Connecting to MongoDB...")

mongo_client = MongoClient('localhost', 27017)

if "TestesClinicos" in mongo_client.list_database_names():
    print("TestesClinicos already exists.")
    answer = input("You wish to drop it? 'y' to Yes, anything else to No.\n")

    if answer.lower() == 'y':
        mongo_client.drop_database("TestesClinicos")

    else:
        print("Cancelled.")
        sys.exit()

testesClinicos = mongo_client["TestesClinicos"]
prova = testesClinicos["Prova"]
testeClinico = testesClinicos["TesteClinico"]

print("Inserting documents...")

for document in converted_results_prova:
    prova.insert_one(document)

for document in converted_results_teste:
    testeClinico.insert_one(document)

print("Done!")

```

Apêndice E

Script contendo todas as queries MongoDB equivalentes às queries SQL do anexo C

```
// Query 1) Listar todas as modalidades registradas na base de dados:
db.TesteClinico.find({}, {"atleta.modalidade": true, "_id": false}).pretty()

// Query 2) Listar os atletas registrados:
db.TesteClinico.find({}, {"atleta.nome": true, "_id" : false}).pretty()

// Query 3) Listar os atletas e as suas modalidades:
db.TesteClinico.find({}, {"atleta.nome" : 1, "atleta.modalidade" : true, "_id" :
→ false}).pretty()

// Query 4) Listar os nome dos atletas da modalidade: 'salto em altura'
db.TesteClinico.find({'atleta.modalidade': "Salto em Altura" }, {"atleta.nome" : true,
→ "_id" : false}).pretty()

// Query 5) Listar o tipo de testes que se podem realizar na clinica:
db.TesteClinico.find({}, {"tipo": true, "_id" : false}).pretty()

// Query 6) Listar o nome dos atletas que tem testes agendados:
```

```

db.TesteClinico.find({"realizado":false}, {"atleta.nome":true , "_id":false}).pretty()

// Query 7) Listar o nome dos atletas que foram submetidos a testes clinicos:
db.TesteClinico.find({"realizado":true}, {"atleta.nome":true , "_id":false}).pretty()

// Query 8) Lista dos dias em o atleta "André sousa figueiredo" tem algum teste clinico
→ marcado:
db.TesteClinico.find({"atleta.nome" : "André Sousa Figueiredo", "realizado" :
→ false}, {"data_hora" : 1, "_id" : 0 }).pretty()

// Query 9) Lista o nome dos profissionais que trabalham na clinica:
db.TesteClinico.distinct("profissional.nome").pretty()

// Query 10) Número de profissionais que trabalham na clinica:
db.TesteClinico.distinct("profissional.nome").length

// Query 11) Número de profissionais que ja realizaram pelo menos 1 teste clinico:
db.TesteClinico.distinct("profissional.nome", {"realizado": true}).length

//Query 12) Atletas cujo o peso se encontra entre 2 paramentros (60Kg E 90Kg) considerado
→ peso saudável:
db.TesteClinico.find({"atleta.peso": {$gt : 60 , $lt:90}}, {"atleta.nome": true,
→ "_id" : false}).pretty()

// View 1
db.TesteClinico.find({"tipo": {$ne : "Controlo Anti-Doping"}, "data_hora": {$gt : new
→ Date(), $lt : new Date(Date.now() + 1000 * 3600 * 24 * 14)}}), {"atleta.nome": true,
→ "tipo": true, "data_hora": true, "_id" : false}).pretty()

// View 2
db.TesteClinico.find({"data_hora": {$gt : new Date(), $lt : new Date(Date.now() + 1000 *
→ 3600 * 24 * 30)}}), {"profissional.nome": true, "data_hora": true, "_id" :
→ false}).pretty()

// View 3) Lista que associa a todos os profissionais quais os atletas a quem esse
→ profissional ja realizou testes
db.TesteClinico.find({}, {"_id":false, "profissional.nome": true, "atleta.nome" : true
→ }).pretty()

```

```
//View 5) Lista para cada teste_Clinico o atleta e o profissional envolvidos:  
db.TesteClinico.find({}, {"tipo": true, "profissional.nome": true, "atleta.nome" :  
  ↳ true, "_id" : false}).pretty()
```

Bibliografia

BEGG, Carolyn; COLLONY, Thomas. *Database Systems: A Practical Approach to Design, Implementation, and Management*, Sixth Edition, Global Edition