



Universidade do Minho
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Letivo de 2019/2020

Trabalho Prático

Diogo Vasconcelos, a85059

Gonçalo Pinto, a83732

João Parente, a84197

José Costa, a84829

Janeiro de 2020

BD

Data de Receção	
Responsável	
Avaliação	
Observações	

Trabalho Prático

Diogo Vasconcelos, a85059

Gonçalo Pinto, a83732

João Parente, a84197

José Costa, a84829

Janeiro de 2020

Resumo

O seguinte trabalho prático pretendeu desenvolver as nossas capacidades nos conteúdos da unidade curricular em Bases de Dados, principalmente no planeamento e execução de um projeto de Sistemas Bases de Dados, (SBD), em duas componentes, modelo relacional e não relacional, com particular ênfase na análise, planeamento, modelação, arquitetura e implementação deste tipo de sistemas num caso de estudo em particular.

Área de Aplicação: Sistemas Bases de Dados.

Palavras-Chave: Análise, planeamento, modelação, arquitetura, implementação, requisitos, modelo relacional, SQL, esquema conceptual, esquema lógico, esquema físico, álgebra relacional, modelo não relacional, NoSQL, base de dados gráfica, quantidade de dados, escalabilidade, flexibilidade, confiabilidade, consistência.

Índice

1.Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	1
1.3. Motivação e Objectivos	1
2.Cenário Aplicacional	2
3. Levantamento de requisitos	3
3.1. Requisitos de Descrição	3
3.2. Requisitos de Exploração	3
3.3. Requisitos de Controlo	4
4. Modelo Relacional	5
4.1 Definição, contextualização, motivação e objetivos	5
4.2 Viabilidade do projeto	5
4.3 Construção dos diferentes modelos	6
4.3.1 Esquema conceptual	6
4.3.2 Esquema lógico	8
4.3.3 Esquema físico	9
4.4 Povoamento do SGBD	9
4.5 Resolução dos requisitos de exploração	9
4.6 Perfis de Utilização	21
5. Modelo Não Relacional	23
5.1 Definição, contextualização, motivação e objetivos	23
5.2 Viabilidade do projeto	23
5.3 Migração dos dados contidos no sistema relacional implementado para o novo sistema	24
5.4 Implementação de um conjunto de "queries"	25
6. Conclusões	28
Bibliografia	29
Referências	30
Lista de Siglas e Acrónimos	31
Anexos	32
I. Anexo A	32
II. Anexo B	37
III. Anexo C	44

Índice de Figuras

Figura 1 - Diagrama ER	7
Figura 2- Esquema Lógico	8
Figura 3- Requisito 0	10
Figura 4- Requisito 1	10
Figura 5- Requisito 2	11
Figura 6- Requisito 3	11
Figura 7- Requisito 4	12
Figura 8- Requisito 5	12
Figura 9- Requisito 6	13
Figura 10- Requisito 7	13
Figura 11- Requisito 8	14
Figura 12- Requisito 9	14
Figura 13- Requisito 10	15
Figura 14- Requisito 11	15
Figura 15- Requisito 12	16
Figura 16- Requisito 13	16
Figura 17- Requisito 14	17
Figura 18- Requisito 15	17
Figura 19- Requisito 16	17
Figura 20- Requisito 17	18
Figura 21- Requisito 18	18
Figura 22- Requisito 19	19
Figura 23- Requisito 20	19
Figura 24- Requisito 21	20
Figura 25- Requisito 22	20
Figura 27- Registrar Atleta	21
Figura 28- Registrar Categoria	21
Figura 29- Marcar Testes	21
Figura 30 – Atribuições das permissões a cada grupo de utilizadores	22
Figura 31- Requisito 0	25
Figura 32- Requisito 1	25
Figura 33- Requisito 2	25
Figura 34- Requisito 3	25
Figura 35- Requisito 4	25
Figura 36- Requisito 5	26
Figura 37- Requisito 6	26
Figura 38- Requisito 7	26
Figura 39- Requisito 8	26

Figura 40- Requisito 9	26
Figura 41- Requisito 10	26
Figura 42- Requisito 11	26
Figura 43- Requisito 12	26
Figura 44- Requisito 13	26
Figura 45- Requisito 14	27
Figura 46- Requisito 15	27
Figura 47- Requisito 16	27
Figura 48- Requisito 17	27
Figura 49- Requisito 18	27
Figura 50- Requisito 19	27
Figura 51- Requisito 20	27
Figura 52- Requisito 21	27
Figura 53- Requisito 22	27

Índice de Tabelas

Tabela 1 – Identificação e caracterização das potenciais entidades envolvidas	6
Tabela 2 - Caracterização dos diversos relacionamentos que possam existir entre as entidades estabelecidas	7

1. Introdução

No 1º semestre do 3º ano do Curso de Engenharia Informática da Universidade do Minho, existe uma Unidade Curricular denominada por Bases de Dados, que tem como objetivo ajudar os estudantes a distinguir dados, informação e conhecimento; compreender e aplicar as formas normais e definir conceptualmente o modelo de dados para um problema específico; escrever interrogações de bases de dados relacionais usando álgebra relacional; utilizar a linguagem SQL para administrar, manter e interrogar bases de dados relacionais; desenvolver aplicações informáticas que operam sobre bases de dados, incluindo aplicações baseadas na Web. Neste relatório foi descrito as diferentes etapas, soluções e estratégias implementadas de forma alcançar esses objetivos.

1.1. Contextualização

O presente trabalho descreveu o processo de análise, planeamento, arquitetura e implementação de um Sistema de Gestão de Bases de Dados em duas componentes, modelo relacional (SQL) e modelo não relacional (Neo4J).

1.2. Apresentação do Caso de Estudo

O caso de estudo definido foi o agendamento e realização de Testes Clínicos de atletas de Atletismo de diferentes modalidades e categorias.

1.3. Motivação e Objetivos

As bases de dados são uma parte integrante da nossa vida do dia-a-dia, muitas vezes não nos apercebemos que estamos a usar uma. Uma das abordagens é baseada em ficheiros, contudo esta tem grandes limitações tais como a dependência de dados, formatos de arquivos incompatíveis e consultas fixas/proliferação de programas/aplicativos. Para tornar-se mais eficaz, é necessária uma nova abordagem. Esta situação levou ao surgimento da base de dados e do Sistema de Gestão de Banco de Dados (SGBD). Uma base de dados é uma coleção compartilhada de dados relacionados logicamente, e uma descrição desses dados, concebidos para satisfazer as necessidades de informação de uma organização. O SGBD é um software que permite aos utilizadores definir, criar, manter e controlar o acesso à base de dados. Todos os acessos à base de dados são através do SGBD.

O SGBD fornece uma linguagem de definição de dados, que permite aos utilizadores definirem a base de dados, e uma linguagem de manipulação de dados, que permite que o utilizador insira, atualize, exclua e recupere dados da base de dados. Num mundo onde a quantidade de dados é cada vez maior é essencial conhecer e trabalhar com SGBD no sentido de fornecer um acesso controlado à base de dados, fornecer segurança, integridade, simultaneidade e controle de uma forma acessível.

Desta forma, a principal motivação deste trabalho foi construir um Sistema de Gestão de Bases de Dados em duas componentes como já foi dito anteriormente: modelo relacional e modelo não relacional proporcionando a aprendizagem desta linguagem de definição e manipulação de dados.

2. Cenário Aplicacional

A prestação nos jogos olímpicos que os atletas portugueses vão fazendo ao longo dos anos, há muito que não é melhor apesar das 4 medalhas de ouro na modalidade desportiva organizada mais antiga do Mundo. Todas as federações desta área de uma maneira ou de outra, tem de guardar a informação relativa aos testes clínicos que os seus atletas realizem. A forma como é mantida essa informação depende muito das pretensões de utilização e necessidades.

De forma a proporcionar o melhor desempenho dos atletas nos próximos jogos olímpicos no Japão, a federação portuguesa de atletismo pretendeu desenvolver uma pequena base de dados para registar os testes clínicos que os seus atletas façam e agendem nas diversas áreas para garantir que os atletas tenham as melhores condições de triunfar.

A federação nesse sentido estabeleceu diversas parcerias com vários médicos das mais variadas áreas de especialização que se deslocam às instalações da federação onde realizam os testes. Os membros da federação acreditam que, dessa maneira, conseguirão fazer uma melhor gestão das parcerias, bem como obter melhores resultados.

Desta forma promoveu-se várias reuniões de trabalho com os membros da federação, com alguns dos médicos e com alguns responsáveis das equipas a que os atletas estão associados para discutir o plano de trabalho para os futuros de serviços de base de dados, bem como a estrutura da base de dados. Os arquitetos da futura base de dados ao final deste tempo conseguiram elaborar uma pequena lista de requisitos para a sua construção, que irá ser apresentada posteriormente.

3. Levantamento de requisitos

3.1. Requisitos de Descrição

- Na altura da inscrição, um atleta tem de indicar o seu número de contribuinte, nome próprio, apelido, data de nascimento, nacionalidade, género, email, peso, altura, nome da equipa, morada, código postal, localidade e a sua categoria (e modalidade);
- Os médicos são caracterizados por ter um número de contribuinte, nome próprio, apelido, data de nascimento, código postal, localidade e uma especialidade;
- Existem várias modalidades sendo que estas podem ou não ter mais que uma categoria;
- Uma equipa é composta pelo nome desta, data de fundação, morada, código de postal, por atletas;
- Um teste clínico é caracterizado pelo atleta e o médico que o protagonizaram, bem como o preço, a data e hora a que esta aconteceu, o tipo do teste clínico, isto é, se é teste antidopagem ou outro tipo de problema, informação se esse teste encontra-se agendado ou não e informação como correu esse teste (se algum dos protagonistas faltou ou não);
- Um atleta só pode pertencer a uma equipa;
- Uma equipa pode ter atletas de modalidades diferentes;
- Um médico só pode fazer testes da sua especialidade;
- Um médico só pode testar um atleta de cada vez;
- Uma categoria é uma prova específica de uma modalidade;
- Um atleta pode ter mais do que uma modalidade e praticar várias categorias;
- Um teste clínico pode ser agendado por um atleta e um médico, para uma determinada hora.

3.2. Requisitos de Exploração

- Saber a idade de um atleta ou médico;
- A partir de uma equipa conseguir saber quais os testes marcados pelos atletas entre duas datas;
- A partir de uma especialidade saber o total faturado dos testes dessa;
- A partir de uma especialidade saber quais os atletas que não efetuaram qualquer teste;
- Ver o quanto foi gasto por cada atleta e ordenar estes;
- A partir de uma especialidade, uma data e um intervalo de horas ver quais os médicos disponíveis;
- Ver o total gasto por cada modalidade em testes clínicos;
- A partir de uma equipa mostrar todos os atletas dessa equipa e a sua respetiva localidade;
- Mostrar os 5 atletas que mais testes clínicos realizaram;
- Mostrar os 5 médicos e a respetiva especialidade que mais testes clínicos realizaram;
- Determinar a média de idades dos atletas por equipa;

- Saber o total gasto por género dos atletas;
- Saber quais as especialidades com menos médicos;
- Saber quais as especialidades com mais testes;
- Dado um id de um médico saber os testes que ainda atualizou;
- Um médico feito um teste seu pode atualizar o registo dela;
- Determinar o número de testes em que cada atleta faltou;
- Determinar o número de testes em que cada médico faltou;
- Dado um id de um médico saber os testes que fez;
- Saber para cada médico o valor total faturado num determinado ano;
- Saber quais as categorias distintas de cada modalidade;
- Dado um nif de um atleta apresentar todos os testes realizados;
- Ficar a conhecer quanto os atletas de uma dada equipa gastaram.

3.3. Requisitos de Controlo

- O responsável da equipa regista atletas no sistema, as categorias que estes praticam, pode consultar os testes que os seus atletas possam fazer, a localidade dos seus atletas e saber quanto foi gasto por cada atleta;
- Um atleta pode marcar testes clínicos, consultar os testes que realiza e saber quais os médicos disponíveis;
- Um médico pode aceder aos seus testes clínicos, alterar um dado teste clínico e saber quais os testes que falta atualizar;
- O administrador do sistema regista no sistema equipas, médicos e a sua respetiva especialidade;

4. Modelo Relacional

4.1. Definição, contextualização, motivação e objetivos

O modelo relacional é um modelo de dados representativo, adequado a ser o modelo subjacente de um SGBD, que se baseia no princípio de que todos os dados estão armazenados em tabelas .

No modelo relacional, todos os dados são logicamente estruturados dentro das relações (tabelas). Cada relação tem um nome e é composta por atributos (colunas) de dados, também identificados por um nome. Cada tuplo (linha) contém um valor por atributo.

A principal vantagem do modelo relacional é a sua estrutura lógica simples. Os objetivos do modelo relacional são os seguintes: permitir um elevado grau de independência de dados, fornecer motivos para lidar com a semântica dos dados, problemas de consistência e redundância e permitir a expansão de linguagens de manipulação de dados orientadas para conjuntos.

Para o desenvolvimento deste projeto seguindo o modelo relacional utilizamos a linguagem de consulta estruturada ou também denominada de SQL que é a linguagem de pesquisa declarativa padrão para base de dados relacional e desta forma utilizou o software “open-source” MySQL que utiliza a linguagem SQL como interface.

4.2. Viabilidade do projeto

A aplicação do modelo relacional a este projeto faz com ele seja viável devido a: permitir que os utilizadores categorizem e armazenem facilmente dados que posteriormente possam ser consultados e filtrados para extrair informações específicas; também é fácil de incorporar dentro de aplicações; permite ainda que uma nova categoria de dados seja adicionada, posteriormente à sua criação sem que os dados atuais sejam modificados. Outras vantagens da utilização deste modelo são: a precisão, os dados são armazenados apenas uma vez e não permite a duplicação de dados; a flexibilidade, permite que consultas complexas sejam fáceis para os utilizadores realizarem; colaboração, permite que vários utilizadores possam aceder à mesma base de dados; confiança, este modelo apresenta consistência e de fácil compreensão; e por fim garante segurança, os dados das tabelas podem ser restringidos dependendo do tipo de utilizador e permissões que este tenha.

4.3. Construção dos diferentes esquemas

4.3.1 Esquema conceptual

A finalidade do modelo conceptual é capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio. Para isto, tornou-se necessário o entendimento e a correta aplicação dos mecanismos de abstração.

Entidades formam um conjunto de “coisas” com conceitos comuns às quais desejou-se armazenar os dados.

Entidade	Descrição	Caracterização
Atleta	Termo geral que descreve todos atletas do nosso sistema.	Um atleta possui um número de contribuinte, nome próprio, apelido, data de nascimento, nacionalidade, género, email, peso, altura, nome da equipa, rua, código postal, localidade e as suas modalidades.
Equipa	Termo geral que descreve todas as equipas do nosso sistema.	Uma equipa é composta pelo nome desta, data de fundação, rua, código de postal e por atletas;
Código de Postal	Termo geral que permite armazenar os códigos de postal.	Um código de postal possui um código e a localidade.
Médico	Termo geral que descreve todos os médicos que realizam os testes clínicos do nosso sistema.	Os médicos são caracterizados por ter um número de contribuinte, nome próprio, apelido, data de nascimento, código postal e uma especialidade.
Especialidade	Termo geral que descreve todas as especialidades do médico.	Uma especialidade possui um determinado identificador e uma designação.
Modalidade	Termo geral que descreve as diferentes modalidades de atletismo presente no sistema.	Uma determinada modalidade possui um determinado identificador e uma designação.

Tabela 1 – Identificação e caracterização das potenciais entidades envolvidas.

Relacionamentos são associações entre entidades com um significado específico dentro do mundo real. Os objetos do mundo real não ocorrem de forma isolada, eles associam-se ou vinculam-se. Desta forma criou-se os possíveis relacionamentos entre as entidades acima apresentadas.

Entidade 1	Multiplicidade	Relação (1 para 2)	Multiplicidade	Entidade 2
Atleta	N	tem	1	Código Postal
Atleta	N	é consultado por	N	Médico
Atleta	N	pertence a	1	Equipa
Atleta	N	pratica	N	Modalidades
Médico	N	tem	1	Código Postal
Médico	N	é perito em	1	Especialidade

Tabela 2 - Caracterização dos diversos relacionamentos que possam existir entre as entidades estabelecidas.

Com base nos requisitos apresentados, nas potenciais entidades envolvidas e os diversos relacionamentos concebeu-se o seguinte diagrama ER para a base de dados requerida:

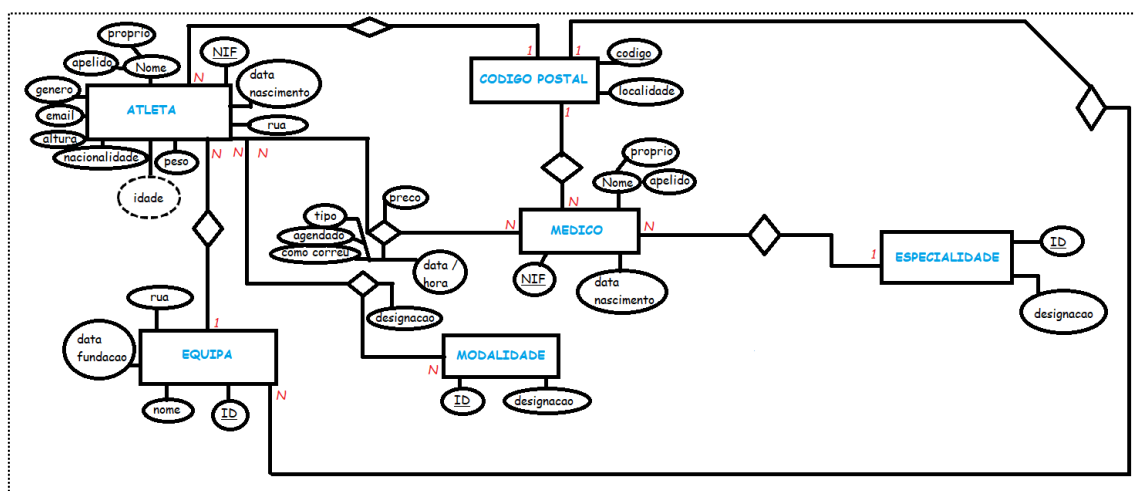


Figura 1 - Diagrama ER.

As entidades foram representadas através de um retângulo com o nome da entidade escrito em seu centro. Os relacionamentos são apresentados através de losangos. Todos os nossos relacionamentos são binários, isto é, apenas duas entidades participam no relacionamento. Os atributos são informações que caracterizam as entidades e os relacionamentos, desta forma eles encontram-se representados por círculos ligados as entidades. O atributo identificador é o que permite distinguir as diferentes identidades uma das outras, encontra-se sublinhado. Existem ainda atributos compostos que representam mais de um tipo de informação em um atributo (nome do atleta e do médico) e atributos derivados que resultam de outros atributos.

4.3.2 Esquema lógico

Depois de criado o modelo conceptual, o próximo passo foi a criação do modelo lógico. Utilizando a ferramenta visual de design de base de dados que integra desenvolvimento, administração, design, criação e manutenção de SQL em um único ambiente de desenvolvimento integrado para o sistema de base de dados MySQL, “MySQLWorkbench”, convertemos o esquema conceptual produzido na alínea anterior para o seu correspondente esquema lógico.

Seguindo as seguintes regras, todas as entidades viram uma tabela; um atributo identificador torna-se chave primária na tabela; atributos compostos tornam-se atributos simples, mapeados em colunas, uma coluna para cada atributo; atributos derivados não são mapeados; atributos multi-valorados dão origem a uma nova tabela.

Em relação aos relacionamentos, num relacionamento 1:1, escolhe-se uma das relações para receber a chave estrangeira, chave essa primária da relação que ocorre o relacionamento. Em um relacionamento 1:N, escolhe-se a relação “N” e inclui como chave estrangeira a chave primária da outra relação.. Em um relacionamento N:N, cria-se uma tabela que inclui como chave estrangeira, as chaves primarias das relações em que ocorre o relacionamento.

Com base nas regras apresentadas concebeu-se o seguinte esquema lógico:

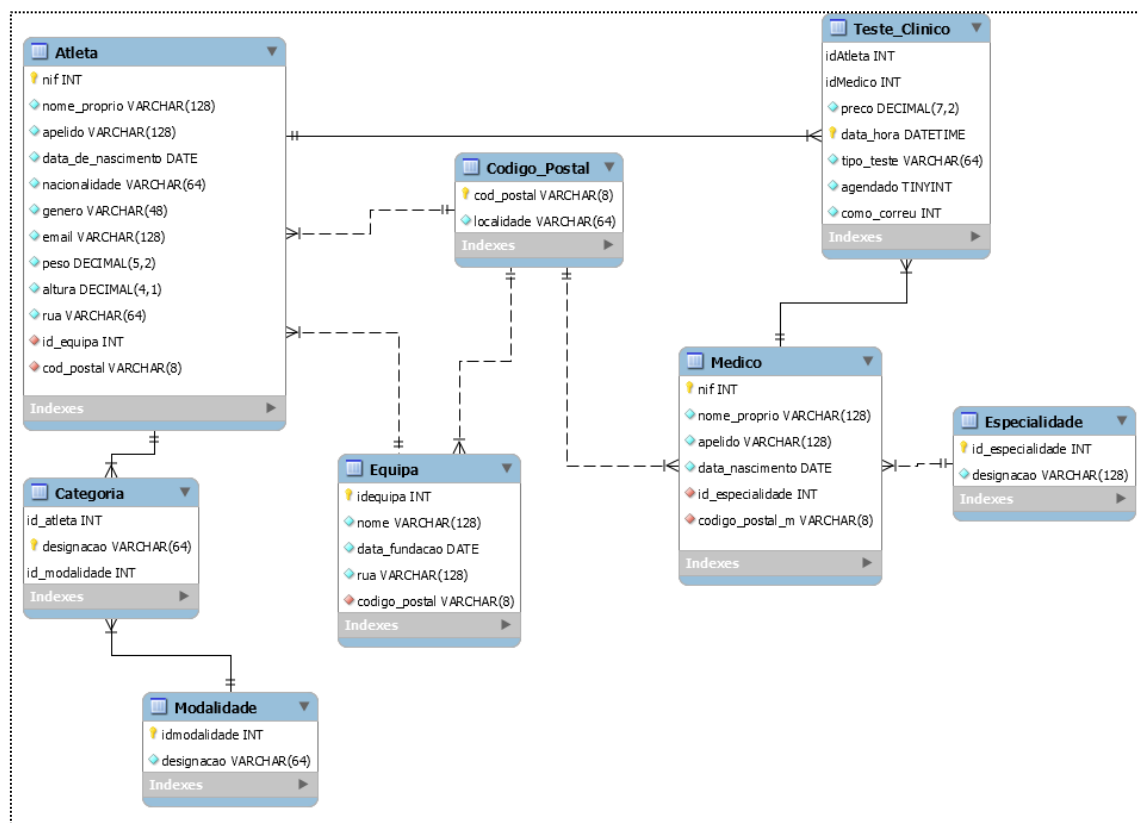


Figura 2 – Esquema lógico.

4.3.3 Esquema físico

O sistema de gestão de base de dados escolhido foi o “MySQL”, como foi dito anteriormente, porque é um sistema que apresenta um elevado desempenho, é robusto e permite vários utilizadores. Para além disso, é gratuito o que faz com que os projetos que utilizem o “MySQL” saiam logo em vantagem em relação ao custo. É um sistema seguro e por isso é usado por muitas empresas.

Partindo do modelo criado no “MySQL Workbench” como foi dito anteriormente, para traduzir o esquema lógico para o esquema físico automaticamente utilizou-se a opção “forward engineer”, como pode ser observado no Anexo A, que levou a criação do SGBD.

4.4. Povoamento do SGBD

De forma a poder cumprir os requisitos de exploração foi necessário preencher a base de dados criada com informação.

A informação que foi adicionada pode ser observada no Anexo B.

4.5. Resolução dos requisitos de exploração

Uma linguagem de bases de dados deve executar essas tarefas com esforço mínimo do utilizador, e a sua estrutura e sintaxe devem ser fáceis de aprender. Desta forma como foi referido a acima linguagem utilizada SQL tem como norma, ISO SQL, duas componentes principais “Data Definition Language” (DDL) para definir a estrutura da base de dados e controlar o acesso aos dados (o comando “Create” é um exemplo desta categoria, utilizou para criar a base de dados) e “Data Manipulation Language” (DML) para recuperação e atualização de dados (exemplo de comandos são o “Insert”, “Update” e “Delete”. Tem também como componente importante a “Data Query Language” (DQL), que lida com a manipulação dos dados, o principal comando desta categoria é o “Select”. Utilizando principalmente a última componente através dos seus comandos é possível satisfazer os requisitos de exploração acima apresentados. Contudo alguns é necessário utilizar a extensão do SQL, PLSQL, que estruturado em blocos permite controlar o fluxo de execução como também permite integrar diferentes ferramentas. E desta forma é possível completar todos os requisitos propostos.

Como foi dito anteriormente utilizou-se a linguagem SQL que é a linguagem de pesquisa declarativa padrão para base de dados relacional . Muitas das características originais do SQL foram inspiradas na álgebra relacional. A principal aplicação da utilização de álgebra relacional permite sustentar a fundamentação teórica de uma base de dados relacional , particularmente na linguagem de consulta para tais bases de dados, entre os maiores o SQL. Assim sendo recorremos ao uso de álgebra relacional pois é uma linguagem teórica com operações sobre uma ou mais relações para definir uma nova relação, sem alterar as relações iniciais. Ambos os operandos e os resultados são relações, assim a saída de uma operação pode tornar-se a entrada para outra operação. Isso permite que expressões possam ser aninhadas.

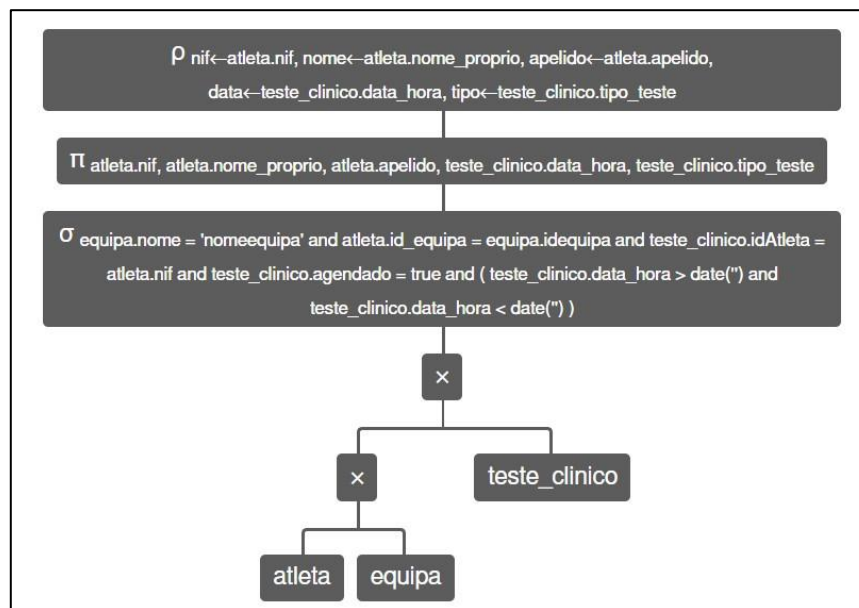
De seguida apresentamos utilizando a Álgebra Relacional, as expressões necessárias desenvolvidas, e as respetivas árvores de resolução, o que levou aos diferentes comandos para os respetivos requisitos. Alguns requisitos não apresentam as árvores de resolução porque não é possível realizar. Por outro lado alguma apresentam nas igualdades pequenos exemplos ao filtrar a informação.

- Saber a idade de um atleta ou médico:

```
DELIMITER $$
CREATE FUNCTION `idade` (dta date) RETURNS INT
Deterministic
BEGIN
    RETURN TIMESTAMPDIFF(YEAR, dta, CURDATE());
END $$
```

Figura 3 – Requisito 0.

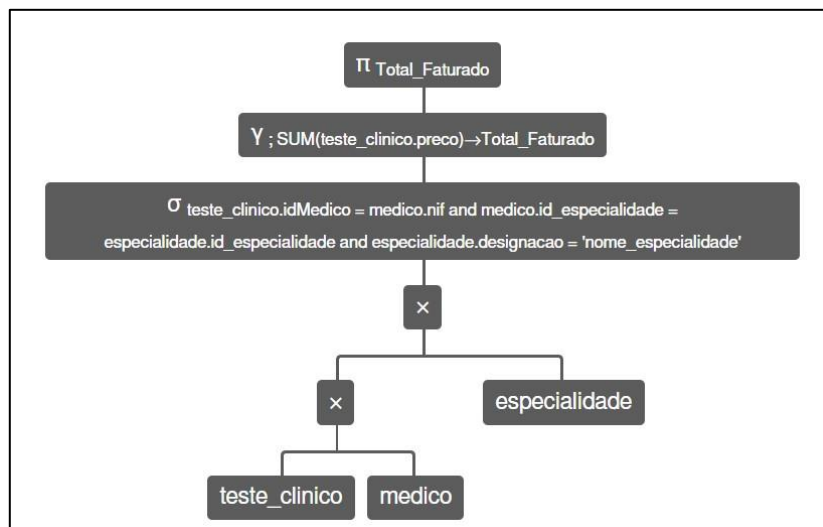
- A partir de uma equipa conseguir saber quais os testes marcados pelos atletas entre duas datas:



```
DELIMITER $$
create procedure getTestesMarcadosEquipa(in nomeequipa varchar(128), in d_a_t_a_i datetime, in d_a_t_a_f datetime)
begin
    select a.nif as nif, a.nome_proprio as nome, a.apelido as apelido, tc.data_hora as data, tc.tipo_teste as tipo
    from atleta a, equipa e, teste_clinico tc
    where e.nome=nomeequipa and a.id_equipa=e.idequipa and tc.idAtleta = a.nif
           and tc.agendado=true and (tc.data_hora between d_a_t_a_i and d_a_t_a_f);
end $$
```

Figura 4 – Requisito 1.

- A partir de uma especialidade saber o total faturado dos testes dessa:

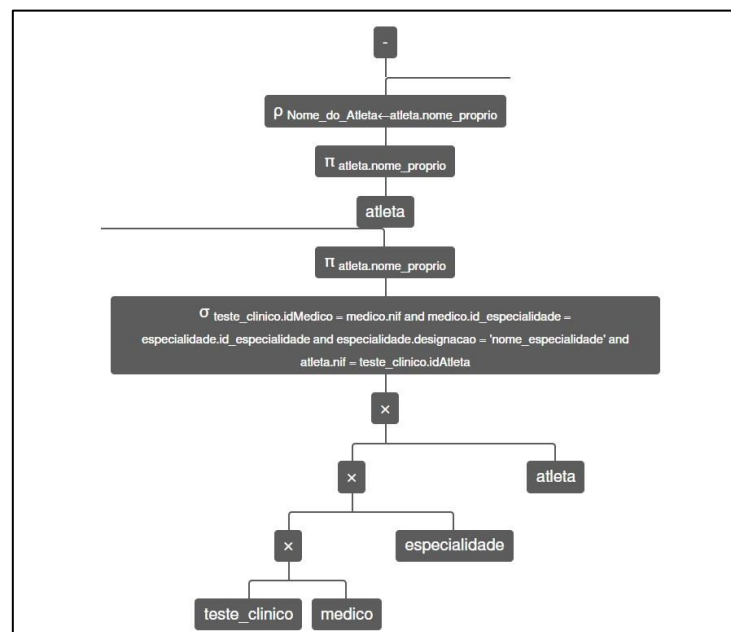


```

DELIMITER $$
create procedure total_faturado_da_especialidade(in nome_especialidade varchar(300))
begin
  select sum(t.preco) as Total_Faturado from teste_clinico t,medico m,especialidade e
  where t.idMedico=m.nif and m.id_especialidade=e.id_especialidade and e.designacao=nome_especialidade;
end $$
  
```

Figura 5 – Requisito 2.

- A partir de uma especialidade saber quais os atletas que não efetuaram qualquer teste:

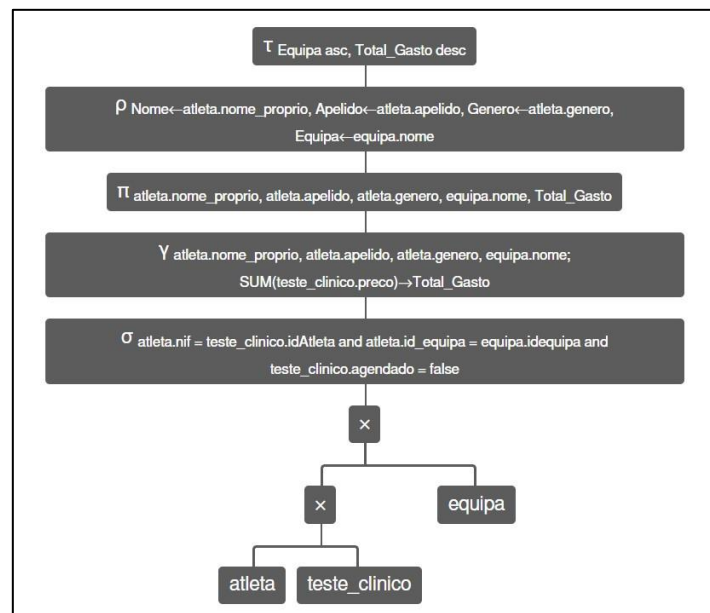


```

DELIMITER $$
create procedure atletas_nunca_testados_por_especialidade(in nome_especialidade varchar(300))
begin
  select b.nome proprio as Nome_do_Atleta from atleta b where b.nome proprio not in(select distinct a.nome proprio
  from teste_clinico t,medico m,especialidade e,atleta a
  where t.idMedico=m.nif and m.id_especialidade=e.id_especialidade
  and e.designacao=nome_especialidade and a.nif=t.idAtleta);
end $$
  
```

Figura 6 – Requisito 3.

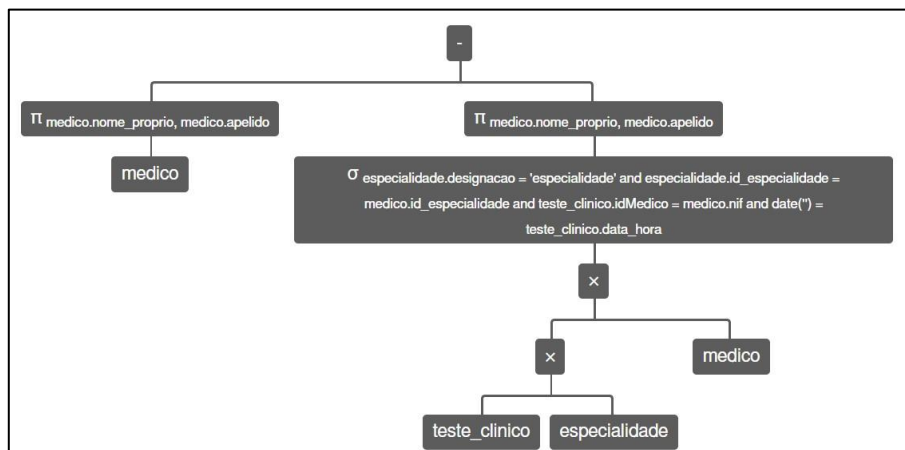
- Ver o quanto foi gasto por cada atleta e ordenar estes:



```
Create view Total_Gasto_Atleta as select a.nome_proprio as Nome, a.apelido as Apelido ,a.genero as Genero, e.nome as Equipa, sum(t.preco) as Total_Gasto
from atleta a, teste_clinico t, equipa e
where a.nif=t.idAtleta and a.id_equipa=e.idequipa and t.agendado=false group by a.nif order by Equipa, Total_Gasto desc;
```

Figura 7 – Requisito 4.

- A partir de uma especialidade, uma data e um intervalo de horas ver quais os médicos disponíveis:

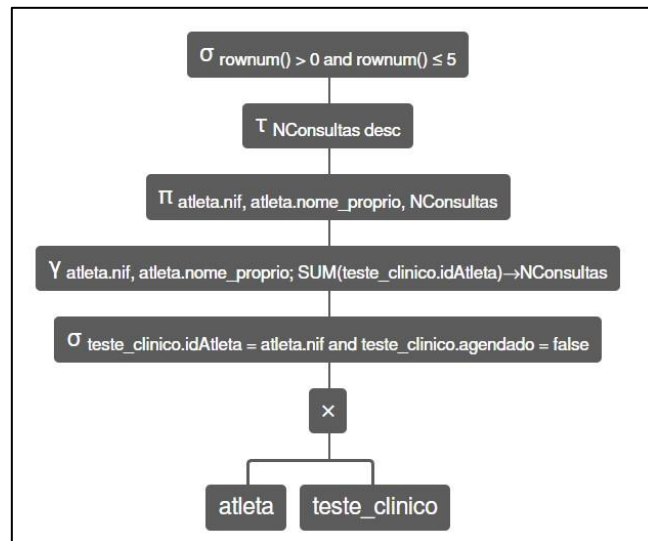


```
DELIMITER $$
create procedure medicosDisponiveis (in especialidade varchar(128), in d_a_t_a datetime)
begin
    select m.nome_proprio, m.apelido from medico m, especialidade e
    where not exists (select * from teste_clinico tc
        where e.designacao=especialidade and e.id_especialidade=m.id_especialidade
        and tc.idMedico=m.nif and d_a_t_a=tc.data_hora)
    and e.designacao=especialidade and e.id_especialidade=m.id_especialidade;
end$$
```

Figura 8 – Requisito 5.

- Ver o total gasto por cada modalidade em testes clínicos:

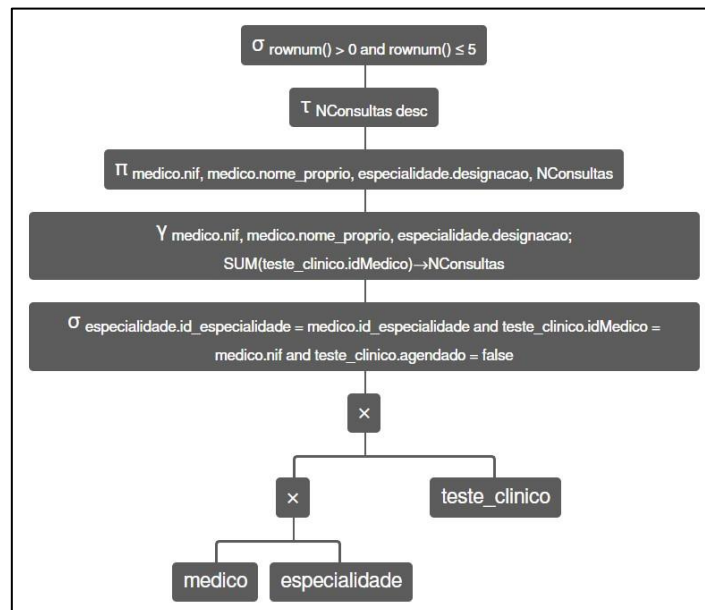
- Mostrar os 5 atletas que mais testes clínicos realizaram:



```
select a.nif,a.nome_proprio,sum(1) as NConsultas from atleta a,teste_clinico t
where t.idAtleta=a.nif and t.agendado= false group by a.nif order by NConsultas desc limit 5;
```

Figura 11 – Requisito 8.

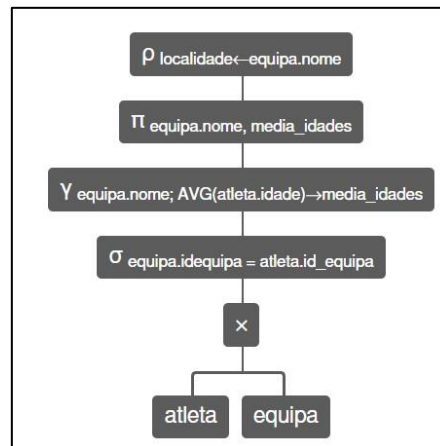
- Mostrar os 5 médicos e a respetiva especialidade que mais testes clínicos realizaram:



```
select m.nif,m.nome_proprio,e.designacao,sum(1) as NConsultas from medico m,especialidade e,teste_clinico t
where e.id_especialidade=m.id_especialidade and t.idMedico=m.nif and t.agendado=false group by nif
order by NConsultas desc limit 5;
```

Figura 12 – Requisito 9.

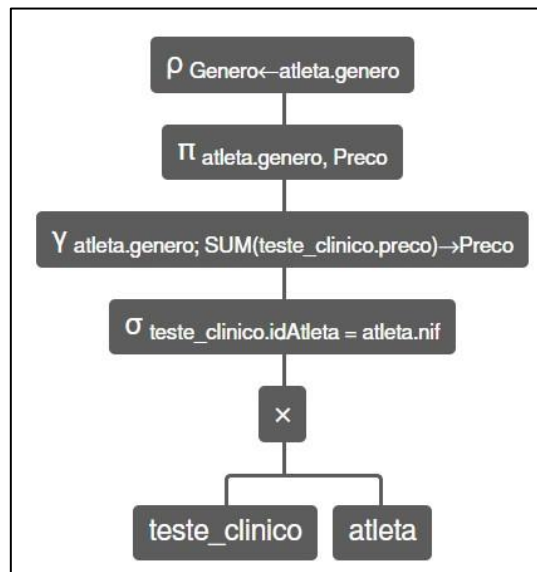
- Determinar a média de idades dos atletas por equipa:



```
Select e.nome as localidade, avg(idade(a.data_de_nascimento)) as media_idades from atleta a, equipa e
where e.idequipa=a.id_equipa group by e.nome;
```

Figura 13 – Requisito 10.

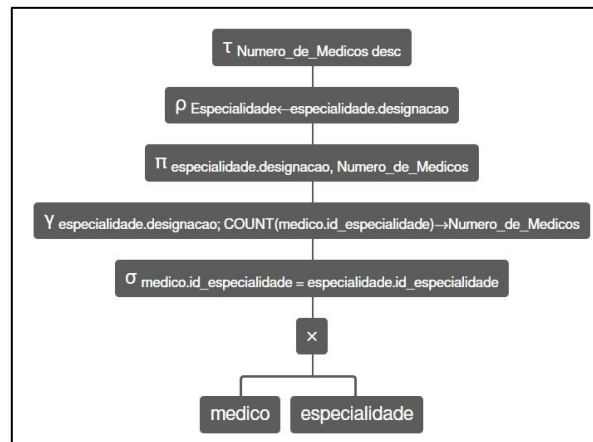
- Saber o total gasto por género dos atletas:



```
select a.genero as Genero , sum(t.preco) as Preco from teste_clinico t, atleta a where t.idAtleta=a.nif group by a.genero;
```

Figura 14 – Requisito 11.

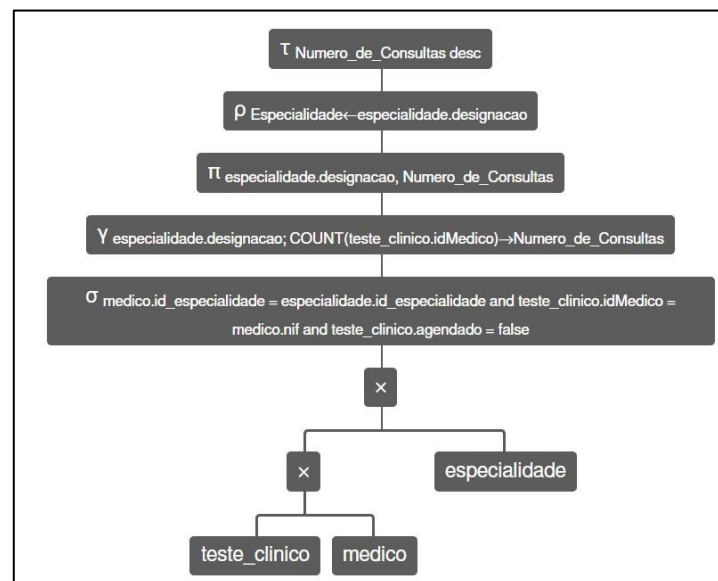
- Saber quais as especialidades com menos médicos:



```
select e.designacao as Especialidade, count(1) as Numero_de_Medicos from medico m, especialidade e
where m.id_especialidade=e.id_especialidade group by e.designacao order by Numero_de_Medicos desc;
```

Figura 15 – Requisito 12.

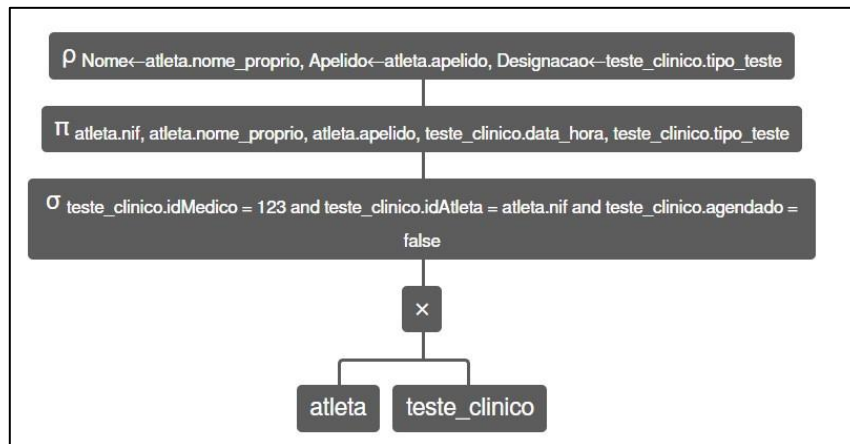
- Saber quais as especialidades com mais testes:



```
select e.designacao as Especialidade, count(1) as Numero_de_Consultas
from teste_clinico t, medico m, especialidade e
where m.id_especialidade=e.id_especialidade and t.idMedico=m.nif and t.agendado=false
group by e.designacao order by Numero_de_Consultas desc;
```

Figura 16 – Requisito 13.

- Dado um id de um médico saber os testes que ainda atualizou:



```
DELIMITER $$
create procedure faltaAtualizar(in idMed int(11))
begin
    select a.nif, a.nome proprio as Nome, a.apelido as Apelido, tc.data_hora, tc.tipo_teste as Designacao
    from atleta a, teste_clinico tc where tc.idMedico=idMed and tc.idAtleta=a.nif and t.agendado=false;
end $$
```

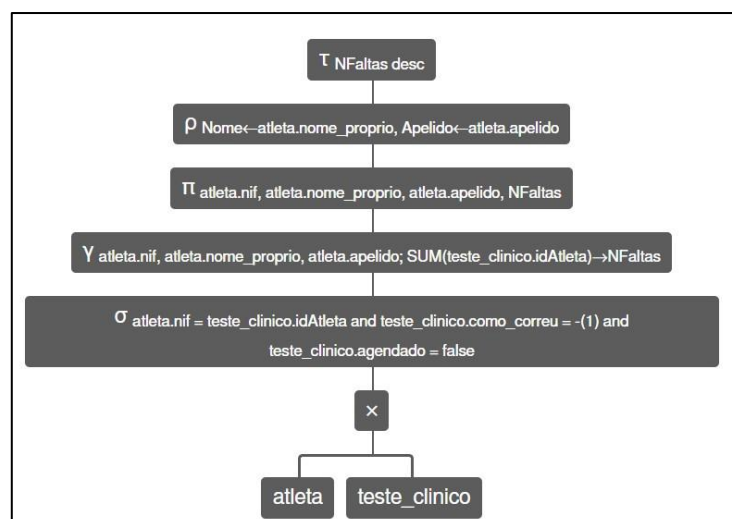
Figura 17 – Requisito 14.

- Um médico feito um teste seu pode atualizar o registo dela:

```
DELIMITER $$
create procedure atualizaRegisto(in idMed int(11), in prec decimal(7,2), in quemfaltou int(11), in d_a_t_a datetime)
begin
    update teste_clinico
    set teste_clinico.prec=prec, teste_clinico.como_correu=quemfaltou, teste_clinico.como_correu.agendado=false
    where teste_clinico.agendado=true and teste_clinico.data_hora=d_a_t_a and teste_clinico.idMedico=idMed;
end $$
```

Figura 18 – Requisito 15.

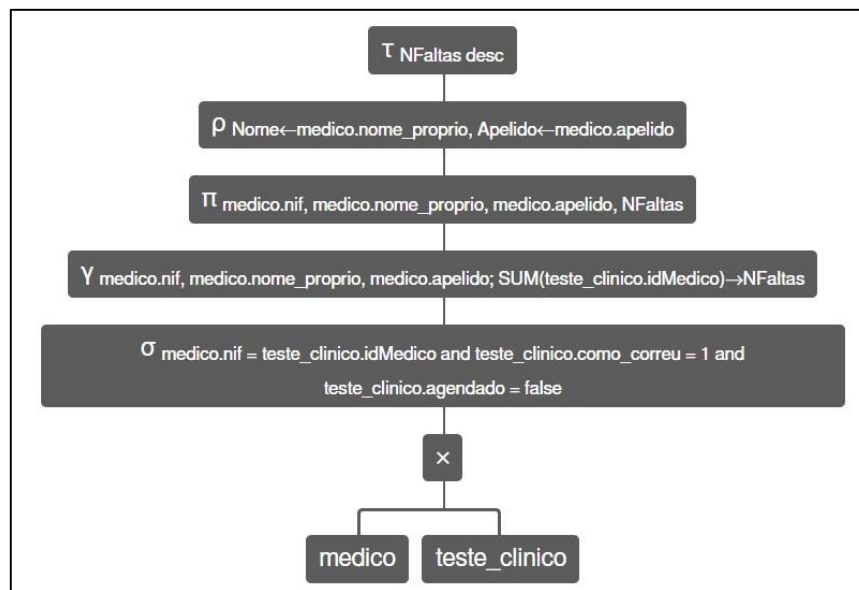
- Determinar o número de testes em que cada atleta faltou:



```
select a.nif,a.nome proprio as Nome, a.apelido as Apelido, sum(1) as NFaltas from atleta a, teste_clinico tc
where a.nif=tc.idAtleta and tc.como_correu = -1 and tc.agendado=false group by a.nif order by NFaltas desc;
```

Figura 19 – Requisito 16.

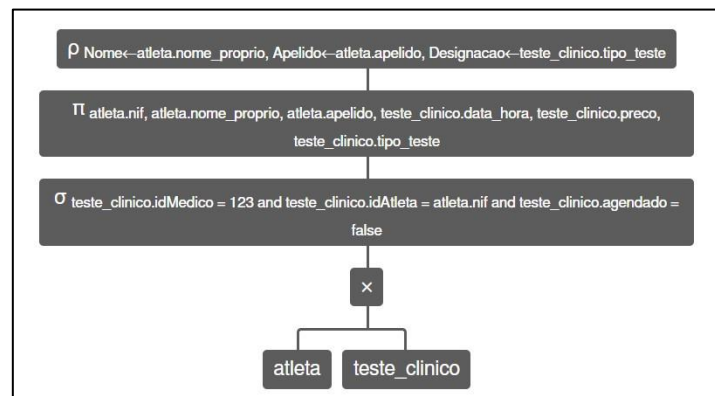
- Determinar o número de testes em que cada médico faltou:



```
select m.nif,m.nome_proprio as Nome, m.apelido as Apelido, sum(1) as NFaltas from medico m, teste_clinico tc
where m.nif=tc.idMedico and tc.como_correu = 1 and tc.agendado=false group by m.nif order by NFaltas desc;
```

Figura 20 – Requisito 17.

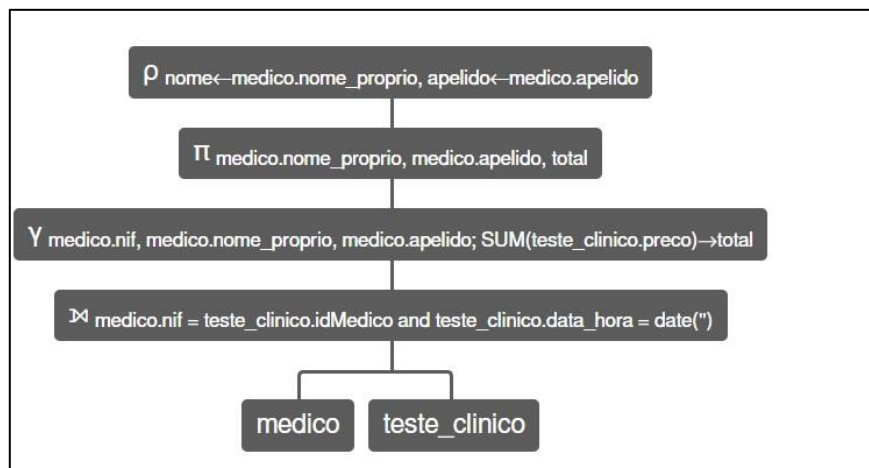
- Dado um id de um médico saber os testes que fez:



```
DELIMITER $$
create procedure getTestes(in idMed int(11))
begin
    select a.nif, a.nome_proprio as Nome, a.apelido as Apelido, tc.data_hora, tc.preco, tc.tipo_teste as Designacao
    from atleta a, teste_clinico tc where tc.idMedico=idMed and tc.idAtleta=a.nif and tc.agendado<>1;
end $$
```

Figura 21 – Requisito 18.

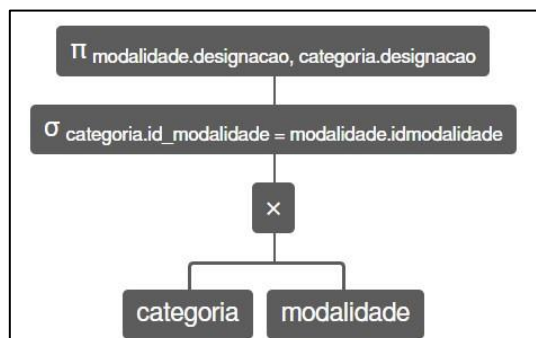
- Saber para cada médico o valor total faturado num determinado ano:



```
DELIMITER $$
create procedure valorAnual(in ano int(4))
begin
    select m.nome proprio as nome, m.apelido as apelido, ifnull(sum(tc.preco),0) as total from medico m left join teste_clinico tc on m.nif=tc.idMedico
    and year(tc.data_hora)=ano group by m.nif;
end$$
```

Figura 22 – Requisito 19.

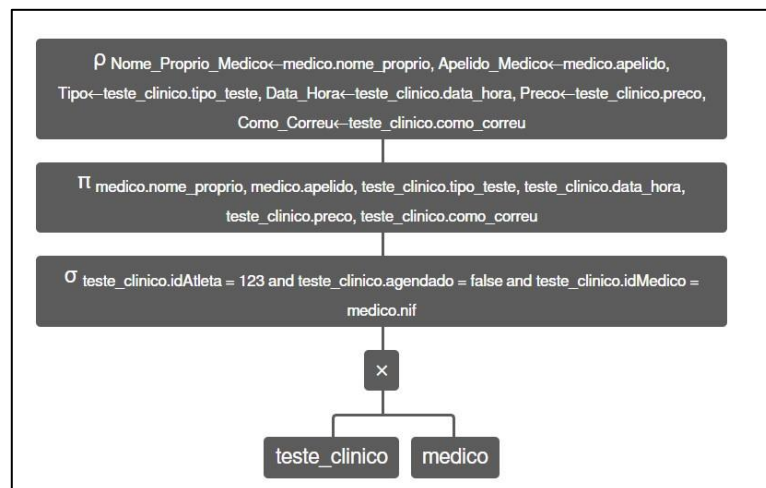
- Saber quais as categorias distintas de cada modalidade:



```
select distinct m.designacao, c.designacao from categoria c, modalidade m where c.id_modalidade=m.idmodalidade
```

Figura 23 – Requisito 20.

- Dado um nif de um atleta apresentar todos os testes realizados:



```
DELIMITER $$
create procedure verTestes (mynif int(11))
begin
    select m.nome_proprio as Nome_Proprio_Medico, m.apelido as Apelido_Medico, tc.tipo_teste as Tipo,
           tc.data_hora as Data_Hora, tc.preco as Preco, tc.como_correu as Como_Correu
    from teste_clinico tc, medico m
    where tc.idAtleta=mynif and tc.agendado=false and tc.idMedico=m.nif;
end$$
```

Figura 24 – Requisito 21.

- Ficar a conhecer quanto os atletas de uma dada equipa gastaram:



```
DELIMITER $$
create procedure verGastoEquipa (equipa varchar(64))
begin
    select tga.nome, tga.apelido, tga.genero, tga.Total_Gasto from total_gasto_atleta tga where tga.Equipa=equipa;
end$$
```

Figura 25 – Requisito 22.

4.6 Perfis de Utilização

A linguagem utilizada como foi dito anteriormente foi o SQL e como tal podemos executar umas determinadas operações sobre a base de dados existente. Como foi demonstrado anteriormente o SQL utiliza certos comandos como “Insert”, “Select”, “Update” para executar as tarefas necessárias e estes comandos são categorizados em quatro categorias a maior parte foram referidas acima para cumprir os requisitos de exploração contudo uma outra componente do SQL é “Data Control Language” (DCL) que trata principalmente de direitos, permissões e outros controlos sobre a base de dados, os principais comandos do DCL são “GRANT” e “REVOKE”.

Utilizando esta componente é possível criar alguns exemplos de forma a cumprir os requisitos de controlos apresentados.

Desta forma criou-se 3 utilizadores que representam o grupo onde de uma dada identidade, ou seja, criou-se um utilizador denominado Atleta, outro Equipa e por fim um médico.

As permissões dadas a cada um destes seria idêntica para um utilizador que seria do mesmo grupo, por exemplo os privilégios apresentados para o utilizador atleta seriam idênticos para qualquer atleta que se registasse na base de dados.

Tal como foi dito previamente, os médicos, a equipa e o código de postal é inserido pelo administrador da base de dados.

Criou-se além dos comandos apresentados, um comando para registar atletas, para registar uma categoria e um para marcar testes clínicos.

```
DELIMITER $$
create procedure registaAtleta(nif int(11), nome varchar(128), apelido varchar(128), datanascimento date, nacionalidade varchar(64), genero varchar(48), email varchar(128), peso decimal(5,2), altura decimal(5,2))
begin
    insert into atleta values(nif,nome,apelido,datanascimento,nacionalidade,genero,email,peso,altura,rua, (select e.idequipa from equipa e where e.nome=nome_equipa),codigo_posta);
end$$
```

Figura 27 – Registar Atleta.

```
DELIMITER $$
create procedure registaCategoria(design varchar(64), idA int(11), modalidade varchar(64))
begin
    insert into categoria values(idA,design,(select m.idmodalidade from modalidade m where m.designacao=modalidade));
end$$
```

Figura 28 – Registar Categoria.

```
DELIMITER $$
create procedure marcarTestes (tipo varchar(64), nifAtleta int(11), nomeproprioMedico varchar(128), apelidoMedico varchar(128),data_hora datetime)
begin
    if EXISTS (select * from testeclinico tc, medico m where m.nome_proprio=nomeproprioMedico and m.apelido=apelidoMedico and tc.idMedico=m.nif and tc.data_hora=data_hora and tc.agendado=false) then
        insert into teste_clinico values (nifAtleta,(select m.nif from medico m where m.nome_proprio=nomeproprioMedico and m.apelido=apelidoMedico),null,data_hora,tipo,true,null);
    end if;
end$$
```

Figura 29 – Marcar Testes.

Assim, atribuímos a cada grupo de utilizadores os seguintes comandos:

```
CREATE USER 'equipa'@'localhost' IDENTIFIED BY 'equipa_password';  
GRANT EXECUTE ON PROCEDURE registaAtleta TO 'equipa'@'localhost';  
GRANT EXECUTE ON PROCEDURE registaCategoria TO 'equipa'@'localhost';  
GRANT EXECUTE ON PROCEDURE verGastoEquipa TO 'equipa'@'localhost';  
GRANT EXECUTE ON PROCEDURE getTestesMarcadosEquipa TO 'equipa'@'localhost';  
GRANT EXECUTE ON PROCEDURE localidades_dos_atletas TO 'equipa'@'localhost';
```

```
CREATE USER 'atleta'@'localhost' IDENTIFIED BY 'atleta_password';  
GRANT EXECUTE ON PROCEDURE marcarTestes TO 'atleta'@'localhost';  
GRANT EXECUTE ON PROCEDURE verTestes TO 'atleta'@'localhost';  
GRANT EXECUTE ON PROCEDURE medicosDisponiveis TO 'atleta'@'localhost';
```

```
CREATE USER 'medico'@'localhost' IDENTIFIED BY 'medico_password';  
GRANT EXECUTE ON PROCEDURE atualizaRegisto TO 'medico'@'localhost';  
GRANT EXECUTE ON PROCEDURE faltaAtualizar TO 'medico'@'localhost';  
GRANT EXECUTE ON PROCEDURE getTestes TO 'medico'@'localhost';
```

Figura 30 – Atribuições das permissões a cada grupo de utilizadores.

5. Modelo Não Relacional

5.1. Definição, contextualização, motivação e objetivos

NoSQL é um termo que se refere a um tipo específico de modelo de bases de dados ou sistemas de gestão de bases de dados. O termo NoSQL é muito amplo, que não se refere a um modelo específico de bases de dados. Refere-se a toda uma panóplia de diferentes modelos que têm como principal característica não se encaixarem no modelo relacional.

Os SGBDs não relacionais surgem porque na maior parte das situações o modelo relacional não é o mais adequado para os requisitos, a situação mais comum ocorre nos casos em que os dados sejam não estruturados ou semiestruturados ou o volume de dados (“big data”) pode exigir uma nova maneira de abordar o problema do armazenamento e da recuperação desses dados.

A maioria dos SGBDs NoSQL é disponibilizado em código aberto, não possui um esquema fixo (não exige o esforço de modelação antes da introdução de quaisquer dados), tem um comportamento excelente em ambientes “cluster” e por fim a maioria das bases de dados não é rigorosa no cumprimento as restrições de ACID (Atomicidade, Consistência, Isolamento, Durabilidade), em diferentes graus porque as soluções NoSQL tem como objetivo fornecer alta disponibilidade e escalabilidade em ambiente “cluster”.

A bases de dados NoSQL são atualmente classificadas em quatro tipos principais, neste presente projeto utilizou-se uma base de dados gráfica como foi nos sugerido pelos docentes. Uma base de dados gráfica é uma base de dados que usa um modelo gráfico para representar e armazenar os dados, sendo uma alternativa ao modelo relacional. Numa base de dados gráfica, não há esquema predefinido como tal. Qualquer esquema é simplesmente um reflexo dos dados que foram inseridos. À medida que dados mais variados são inseridos, o esquema acompanha esse crescimento.

Para o desenvolvimento deste projeto seguindo o modelo não relacional utilizou o software “open-source” Neo4j que é um sistema de gestão de banco de dados de gráficos.

5.2. Viabilidade do projeto

A representação gráfica dos dados numa base de dados gráficos contrasta com a estrutura tabular apresentada no modelo de bases de dados relacional. Contudo a aplicação do modelo não relacional a este projeto faz com ele também seja viável devido à performance que oferece, as bases de dados gráficas potenciam grandes ganhos de desempenho em relação às bases de dados relacionais, principalmente quando se trata de consultas grandes em dados relacionados e permitir uma grande flexibilidade, como não há necessidade de definir um esquema, temos total flexibilidade sobre o crescimento da base de dados

5.3. Migração dos dados contidos no sistema relacional implementado para o novo sistema

De forma a cumprir os requisitos propostos acima foi necessário fazer uma migração dos dados contidos no sistema relacional implementado para o novo sistema.

O “Neo4j Graph Database” possui os seguintes blocos de construção: nós - o nó é uma unidade fundamental de um gráfico e contém propriedades com pares de valores-chave; relacionamentos - os relacionamentos são outro componente importante deste tipo de base de dado, cada relacionamento contém um nó inicial e um nó final e tal como nós, os relacionamentos também podem conter propriedades como pares de valor-chave; propriedades - a propriedade é um par de valores-chave para descrever nós e relacionamentos do gráfico; etiquetas - são rótulos que associam um nome comum a um conjunto de nós ou relacionamentos, um nó ou relacionamento pode conter um ou mais rótulos, é possível criar novos rótulos para nós ou relacionamentos já existentes como também remover já existentes.

Desta forma considerou-se como nós: Código de Postal, Especialidade, Modalidade, Equipa, Médico e Atleta.

Criou-se o relacionamento entre Equipa e Código de Postal com a designação de TEM_SEDE_EM, entre Médico e Código de Postal com a designação RESIDE, entre Médico e especialidade com a designação ESPECIALISTA_EM, entre Atleta e Código de Postal com a designação RESIDE, entre Atleta e Equipa com a designação ATLETA_DO, entre Atleta e Modalidade com a designação PRATICA e por fim entre Atleta e Médico com a designação E_TESTADO_POR. O nó inicial é sempre o nó referido primeiro e o nó final o segundo nó referido.

Em relação às propriedades: Código de Postal tem uma localidade e um código; Especialidade possui uma designação; Modalidade também possui uma designação; Equipa detém um nome, data de fundação e rua; Médico tem um nif, nome e data de nascimento; Atleta possui um nif, nome, data de nascimento, nacionalidade, género, email, peso, altura e rua. Aos níveis dos relacionamentos o relacionamento entre Atleta e Modalidade com a designação PRATICA possui com propriedade a categorias que aquele atleta pratica da modalidade com que tem uma relação e por fim o relacionamento entre Atleta e Médico com a designação E_TESTADO_POR tem com propriedades uma descrição, data (formato texto), preço, uma “flag” se o teste está agendado ou não e outra “flag” a indicar como correu a consulta, isto é, se algum dos intervenientes faltou a consulta.

Desta forma, introduziu-se o ficheiro presente no Anexo C e criou-se a base de dados gráfica com as características acima referidas.

5.4. Implementação de um conjunto de “queries”

Após a migração dos dados contidos no sistema relacional implementado para o novo sistema, tentou-se cumprir os requisitos de exploração acima descritos.

Desta forma, apresentamos os diferentes comandos implementados para cumprir os requisitos apresentados contudo é de realçar que alguns comandos dependem da introdução de alguns dados como trabalhou-se com uma base de dados gráfica esta não permite criar funções que receba dados como “input” por isso para alguns comandos temos a indicação do conteúdo a preencher de forma a utilizador ficar a saber o que preencher para efetuar a respetiva consulta. É de realçar que no requisito de exploração, ver o quanto foi gasto por cada atleta e ordenar estes, foi criado uma “view” porque é um requisito com alguma informação e recorrentemente pode ser consultado logo ao criar uma “view” filtrou-se o conteúdo das tabelas a serem exibidas e serviu para agrupá-las. Neste novo sistema não vai ser possível criar a “view” logo abdicamos parcialmente desse requisito nesta etapa.

- Saber a idade de um atleta ou médico:

```
match (p{nif:"Número"}) return duration.between(date(p.dataNascimento),date()).years as Idade;
```

Figura 31 – Requisito 0.

- A partir de uma equipa conseguir saber quais os testes marcados pelos atletas entre duas datas:

```
$ match (e:Equipa{nome:"Nome_Equipa"})←[:ATLETA_DO]-(a:Atleta)-[tc:E_TESTADO_POR]→(:Medico) where (tc.data ≥ "Data_Inicial" and tc.data ≤ "Data_Final") and tc.agendado=true return a.nome as nome, tc.data as data, tc.descricao as descricao
```

Figura 32 – Requisito 1.

- A partir de uma especialidade saber o total faturado dos testes dessa:

```
match(:Atleta)-[tc:E_TESTADO_POR]→(:Medico)-[:ESPECIALISTA_EM]→(:Especialidade{designacao:"Nome_Especialidade"}) where tc.agendado=false return sum(tc.preco) as Total;
```

Figura 33 – Requisito 2.

- A partir de uma especialidade saber quais os atletas que não efetuaram qualquer teste:

```
match (a:Atleta) where not (a)-[:E_TESTADO_POR]-(:Medico)-[:ESPECIALISTA_EM]→(:Especialidade{designacao:"Nome_Especialidade"}) return a
```

Figura 34 – Requisito 3.

- Ver o quanto foi gasto por cada atleta e ordenar estes:

```
match(n:Atleta)-[tc:E_TESTADO_POR]→(:Medico) where tc.agendado=false return n.nome as nome, sum(tc.preco) as total order by total desc
```

Figura 35 – Requisito 4.

- A partir de uma especialidade, uma data e um intervalo de horas ver quais os médicos disponíveis:

```
match (m:Medico) where not (:Atleta)-[:E_TESTADO_POR{data:"Data_Testes_Amarcar"}]→(m)-[:ESPECIALISTA_EM]→(:Especialidade{designacao:"Nome_Especialidade"}) and (m)-[:ESPECIALISTA_EM]→(:Especialidade{designacao:"Nome_Especialidade"}) return m
```

Figura 36 – Requisito 5.

- Ver o total gasto por cada modalidade em testes clínicos:

```
match(m:Modalidade)←[:PRATICA]-(a:Atleta)-[tc:E_TESTADO_POR]→(:Medico) where tc.agendado=false return m.designacao as designacao, sum(tc.preco) as total order by total desc
```

Figura 37 – Requisito 6.

- A partir de uma equipa mostrar todos os atletas dessa equipa e a sua respetiva localidade:

```
match (cp:CodigoPostal)←[:RESIDE]-(a:Atleta)-[:ATLETA_DO]→(:Equipa{nome:"Nome_Equipa"}) return cp.localidade as localidade, a.nome as nome
```

Figura 38 – Requisito 7.

- Mostrar os 5 atletas que mais testes clínicos realizaram:

```
match (a:Atleta)-[tc:E_TESTADO_POR]→(:Medico) where tc.agendado=false return a.nome as nome, count(tc) as consultas order by consultas desc limit 5
```

Figura 39 – Requisito 8.

- Mostrar os 5 médicos e a respetiva especialidade que mais testes clínicos realizaram:

```
match ()-[tc:E_TESTADO_POR]→(m:Medico)-[:ESPECIALISTA_EM]→(e:Especialidade) where tc.agendado=false return m.nome as nome, e.designacao as especialidade, count(tc) as consultas order by consultas desc limit 5
```

Figura 40 – Requisito 9.

- Determinar a média de idades dos atletas por equipa:

```
match (a:Atleta)-[:ATLETA_DO]→(e:Equipa) return e.nome as Equipa, avg(duration.between(date(a.dataNascimento),date()).years) as Media_Idades;
```

Figura 41 – Requisito 10.

- Saber o total gasto por género dos atletas:

```
match(a:Atleta)-[tc:E_TESTADO_POR]→(:Medico) where tc.agendado=false return a.genero as Genero, sum(tc.preco) as total order by Genero
```

Figura 42 – Requisito 11.

- Saber quais as especialidades com menos médicos:

```
match (m:Medico)-[:ESPECIALISTA_EM]→(e:Especialidade) return e.designacao as Especialidade, count(ee) as total order by total
```

Figura 43 – Requisito 12.

- Saber quais as especialidades com mais testes:

```
match (a:Atleta)-[tc:E_TESTADO_POR]-(m:Medico)-[:ESPECIALISTA_EM]→(e:Especialidade) where tc.agendado=false return e.designacao as Especialidade, count(tc) as total order by total desc
```

Figura 44 – Requisito 13.

- Dado um id de um médico saber os testes que ainda atualizou:

```
match (:Atleta)-[tc:E_TESTADO_POR]→(:Medico{nif: "Numero"}) where tc.agendado=true return tc
```

Figura 45 – Requisito 14.

- Um médico feito um teste seu pode atualizar o registo dela:

```
match (:Atleta)-[tc:E_TESTADO_POR{agendado:true,data:"Data_Testa"}]→(:Medico{nif:"Numero"}) SET tc.agendado=false,tc.preco="_", tc.faltou="_" return tc
```

Figura 46 – Requisito 15.

- Determinar o número de testes em que cada atleta faltou:

```
match (a:Atleta)-[tc:E_TESTADO_POR]→(:Medico) where tc.faltou=-1 return a.nome as Nome_Atleta, count(tc) as Faltou
```

Figura 47 – Requisito 16.

- Determinar o número de testes em que cada médico faltou:

```
match (:Atleta)-[tc:E_TESTADO_POR]→(m:Medico) where tc.faltou=1 return m.nome as Nome_Medico, count(tc) as Faltou
```

Figura 48 – Requisito 17.

- Dado um id de um médico saber os testes que fez:

```
match (a:Atleta)-[tc:E_TESTADO_POR]→(m:Medico{nif:"Numero"}) where tc.agendado=false return a.nome as nome,tc as teste_clinico
```

Figura 49 – Requisito 18.

- Saber para cada médico o valor total faturado num determinado ano:

```
match (:Atleta)-[tc:E_TESTADO_POR]→(m:Medico) where tc.agendado=false and tc.data starts with "ANO" return m.nome as nome, sum(tc.preco) as total_facturado
```

Figura 50 – Requisito 19.

- Saber quais as categorias distintas de cada modalidade:

```
match (:Atleta)-[p:PRATICA]-(m:Modalidade) return distinct m.designacao as designacao, collect(distinct p.categoria) as categoria order by m.designacao
```

Figura 51 – Requisito 20.

- Dado um nif de um atleta apresentar todos os testes realizados:

```
$ match (:Atleta{nif:"MyNIF"})-[t:E_TESTADO_POR]-(m:Medico) where t.agendado=false return t,m.nome as Medico
```

Figura 52 – Requisito 21.

- Ficar a conhecer quanto os atletas de uma dada equipa gastaram:

```
$ match (:Equipa{nome:"NomeEquipa"})←[:ATLETA_DO]-(n:Atleta)-[tc:E_TESTADO_POR]→(:Medico) where tc.agendado=false return n.nome as nome, sum(tc.preco) as total order by total desc
```

Figura 53 – Requisito 22.

6. Conclusões

O presente relatório descreveu, de forma sucinta, o processo de análise, planeamento, arquitetura e implementação de um Sistema de Gestão de Bases de Dados com o tema de agendamento e realização de Testes Clínicos de atletas de Atletismo de diferentes modalidades e categorias em duas componentes, modelo relacional utilizando o software MySQL e modelo não relacional através de uma base de dados gráfica utilizando o software Neo4J.

Após a realização deste trabalho, ficámos conscientes das diferenças entre os dois modelos, as suas vantagens e desvantagens, respetivamente.

Consideramos que os principais objetivos foram cumpridos, no entanto, no modelo relacional poderíamos ter criado no esquema conceptual um relacionamento 1:1 de forma aplicar conhecimentos, o nosso conceito de teste clínico temos consciência que não é o melhor em relação ao agendamento e o interveniente que faltou mas achamos a maneira mais razoável de resolver este problema e por fim temos consciência que poderíamos ter criado outros requisitos de exploração que levassem a explorar mais conhecimentos na linguagem SQL.

Em relação ao modelo não relacional tentou-se evitar ao máximo uma “cópia” do modelo relacional para este novo modelo mas não houve muita maneira de fugir das entidades e relacionamentos previamente estipulados.

No final deste trabalho concluímos que o modelo a utilizar neste projeto vai depender da quantidade de dados e da aplicação. Se o projeto caminhar no sentido de ser necessário escalabilidade e flexibilidade dos dados é recomendável utilizar o modelo não relacional contudo se preferir-se confiabilidade, consistência e a possibilidade de utilizar a propriedade ACID o modelo relacional é o mais recomendado.

Sentimos que a realização deste trabalho prático consolidou os nossos conhecimentos da unidade curricular de Bases de Dados bem como permitiu-nos desenvolver as nossas capacidades e adquirir experiência planeamento e execução de um projeto de Sistemas Bases de Dados.

Bibliografia

Connolly, T., Begg, C., Database Systems: A Practical Approach to Design, Implementation, and Management, Addison-Wesley, Global Edition, 26 Sep 2014. ISBN-10: 1292061189, ISBN-13: 978-1292061184.

Referências

Educba. MySQL vs NoSQL. Acedido em 17 de dezembro de 2019, em <https://www.educba.com/mysql-vs-nosql/>.

Team Manomaya. What is the importance of a database management system?. Acedido em 17 de dezembro de 2019, em <https://www.manomayasoft.com/blog/item/210-what-is-the-importance-of-a-database-management-system>.

Guru99. Relational Data Model in DBMS: Concepts, Constraints, Example. Acedido em 17 de dezembro de 2019, em <https://www.guru99.com/relational-data-model-dbms.html>.

Neo4j. Concepts: Relational to Graph. Acedido em, 17 dezembro 2019, em <https://neo4j.com/developer/graph-db-vs-rdbms/>.

DBIS subdivision of the Department of Computer Science at the University of Innsbruck. RelaX - relational algebra calculator. Acedido em 20 de dezembro de 2019, em <https://dbis-uibk.github.io/relax/>.

Lista de Siglas e Acrónimos

“Structured Query Language” (SQL);

Sistemas Bases de Dados (SBD);

Sistema de Gestão de Banco de Dados (SGBD);

“Data Definition Language” (DDL);

“Data Manipulation Language” (DML);

“Data Query Language” (DQL);

“Data Control Language” (DCL);

Atomicidade, Consistência, Isolamento, Durabilidade (ACID) ;

Anexos

Anexo A

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@ @UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@ @FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@ @SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO
_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-----
-- Schema mydb
-----
```

```
-----
-- Schema mydb
-----
```

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

```
-----
-- Table `mydb`.`Codigo_Postal`
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Codigo_Postal` (
  `cod_postal` VARCHAR(8) NOT NULL,
  `localidade` VARCHAR(64) NOT NULL,
  PRIMARY KEY (`cod_postal`))
ENGINE = InnoDB;
```

```
-----
-- Table `mydb`.`Equipa`
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Equipa` (
  `idequipa` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(128) NOT NULL,
  `data_fundacao` DATE NOT NULL,
  `rua` VARCHAR(128) NOT NULL,
```

```

`codigo_postal` VARCHAR(8) NOT NULL,
PRIMARY KEY (`idequipa`),
INDEX `codpostal_idx` (`codigo_postal` ASC) VISIBLE,
CONSTRAINT `codpostal`
  FOREIGN KEY (`codigo_postal`)
  REFERENCES `mydb`.`Codigo_Postal` (`cod_postal`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Atleta`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Atleta` (
  `nif` INT NOT NULL,
  `nome proprio` VARCHAR(128) NOT NULL,
  `apelido` VARCHAR(128) NOT NULL,
  `data_de_nascimento` DATE NOT NULL,
  `nacionalidade` VARCHAR(64) NOT NULL,
  `genero` VARCHAR(48) NOT NULL,
  `email` VARCHAR(128) NOT NULL,
  `peso` DECIMAL(5,2) NOT NULL,
  `altura` DECIMAL(4,1) NOT NULL,
  `rua` VARCHAR(64) NOT NULL,
  `id_equipa` INT NOT NULL,
  `cod_postal` VARCHAR(8) NOT NULL,
  PRIMARY KEY (`nif`),
  INDEX `id_equipa_idx` (`id_equipa` ASC) VISIBLE,
  INDEX `cod_postal_idx` (`cod_postal` ASC) VISIBLE,
  CONSTRAINT `id_equipa`
    FOREIGN KEY (`id_equipa`)
    REFERENCES `mydb`.`Equipa` (`idequipa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `cod_postal`
    FOREIGN KEY (`cod_postal`)
    REFERENCES `mydb`.`Codigo_Postal` (`cod_postal`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```


ENGINE = InnoDB;

-- Table `mydb`.`Modalidade`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Modalidade` (  
  `idmodalidade` INT NOT NULL AUTO_INCREMENT,  
  `designacao` VARCHAR(64) NOT NULL,  
  PRIMARY KEY (`idmodalidade`))  
ENGINE = InnoDB;
```

-- Table `mydb`.`Categoria`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Categoria` (  
  `id_atleta` INT NOT NULL,  
  `designacao` VARCHAR(64) NOT NULL,  
  `id_modalidade` INT NOT NULL,  
  INDEX `id_atleta_idx` (`id_atleta` ASC) VISIBLE,  
  INDEX `id_modalidade_idx` (`id_modalidade` ASC) VISIBLE,  
  PRIMARY KEY (`id_atleta`, `id_modalidade`, `designacao`),  
  CONSTRAINT `id_atleta`  
    FOREIGN KEY (`id_atleta`)  
    REFERENCES `mydb`.`Atleta` (`nif`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `id_modalidade`  
    FOREIGN KEY (`id_modalidade`)  
    REFERENCES `mydb`.`Modalidade` (`idmodalidade`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `mydb`.`Especialidade`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Especialidade` (  
  `id_atleta` INT NOT NULL,  
  `id_modalidade` INT NOT NULL,  
  `id_especialidade` INT NOT NULL,  
  PRIMARY KEY (`id_atleta`, `id_modalidade`, `id_especialidade`),  
  CONSTRAINT `id_atleta`  
    FOREIGN KEY (`id_atleta`)  
    REFERENCES `mydb`.`Atleta` (`nif`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `id_modalidade`  
    FOREIGN KEY (`id_modalidade`)  
    REFERENCES `mydb`.`Modalidade` (`idmodalidade`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `id_especialidade`  
    FOREIGN KEY (`id_especialidade`)  
    REFERENCES `mydb`.`Especialidade` (`id_especialidade`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
`id_especialidade` INT NOT NULL AUTO_INCREMENT,
`designacao` VARCHAR(128) NOT NULL,
PRIMARY KEY (`id_especialidade`))
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Medico`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Medico` (
  `nif` INT NOT NULL,
  `nome proprio` VARCHAR(128) NOT NULL,
  `apelido` VARCHAR(128) NOT NULL,
  `data_nascimento` DATE NOT NULL,
  `id_especialidade` INT NOT NULL,
  `codigo_postal_m` VARCHAR(8) NOT NULL,
  PRIMARY KEY (`nif`),
  INDEX `idEspecialidade_idx` (`id_especialidade` ASC) VISIBLE,
  INDEX `codPostal_idx` (`codigo_postal_m` ASC) VISIBLE,
  CONSTRAINT `idEspecialidade`
    FOREIGN KEY (`id_especialidade`)
      REFERENCES `mydb`.`Especialidade` (`id_especialidade`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `codigo_postal_m`
    FOREIGN KEY (`codigo_postal_m`)
      REFERENCES `mydb`.`Codigo_Postal` (`cod_postal`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Teste_Clinico`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Teste_Clinico` (
  `idAtleta` INT NOT NULL,
  `idMedico` INT NOT NULL,
  `preco` DECIMAL(7,2) NULL,
  `data_hora` DATETIME NOT NULL,
```

```

`tipo_teste` VARCHAR(64) NOT NULL,
`agendado` TINYINT NOT NULL,
`como_correu` INT NULL,
INDEX `id_atleta_idx` (`idAtleta` ASC) VISIBLE,
INDEX `id_medico_idx` (`idMedico` ASC) VISIBLE,
PRIMARY KEY (`idAtleta`, `idMedico`, `data_hora`),
CONSTRAINT `idatleta`
  FOREIGN KEY (`idAtleta`)
  REFERENCES `mydb`.`Atleta` (`nif`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `idmedico`
  FOREIGN KEY (`idMedico`)
  REFERENCES `mydb`.`Medico` (`nif`)
  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;

```

Anexo B

```
insert into codigo_postal values ("4750-600","Barcelos");
insert into codigo_postal values ("4750-484","Barcelos");
insert into codigo_postal values ("4800-034","Guimarães");
insert into codigo_postal values ("4700-188","Braga");
insert into codigo_postal values ("3880-465","Ovar");
insert into codigo_postal values ("4000-039","Porto");
insert into codigo_postal values ("4910-392","Vila praia de Âncora");
```

```
insert into especialidade (designacao) values ("Cardiologia");
insert into especialidade (designacao) values ("Ortopedia");
insert into especialidade (designacao) values ("Clínica Geral");
insert into especialidade (designacao) values ("Psicologia");
insert into especialidade (designacao) values ("Nutrição");
```

```
insert into modalidade (designacao) values ("Corrida de Pista");
insert into modalidade (designacao) values ("Corrida de Obstáculos");
insert into modalidade (designacao) values ("Corrida de Estafetas");
insert into modalidade (designacao) values ("Salto em Comprimento");
insert into modalidade (designacao) values ("Salto em Altura");
insert into modalidade (designacao) values ("Lançamentos");
```

```
insert into medico values(123794561,"Maria","Cunha",'1990-12-30','5','4750-600');
insert into medico values(789456122,"Ana","Esteves",'1985-11-25',4,"4750-484");
insert into medico values(654781256,"Paula","Roberto",'1980-10-20',4,"4800-034");
insert into medico values(123456789,"José","Baião",'1975-9-15',3,"4700-188");
insert into medico values(123456798,"Carlos","Noronha",'1970-8-10',2,"4000-039");
insert into medico values(870125901,"Alfredo","Fernandes",'1965-7-5',1,"3880-465");
insert into medico values(320126902,"Manuel","Oliveira",'1960-6-3',3,"4910-392");
insert into medico values(745212389,"Joaquim","Mendes",'1955-5-2',2,"4800-034");
insert into medico values(999456123,"Augusto","Silva",'1950-4-1',1,"4700-188");
insert into medico values(999888777,"Acácio","Santos",'1945-3-30',3,"4700-188");
```

```
insert into equipa (nome,data_fundacao,rua,codigo_postal) values ("Oliveira
Sports",'2012-12-12',"Avenida Oliveira nº1","4750-600");
insert into equipa (nome,data_fundacao,rua,codigo_postal) values ("Sociedade
Recreativa de Vila Praia de Âncora",'1990-12-31',"Rua dos Emigrantes nº99","4910-392");
insert into equipa (nome,data_fundacao,rua,codigo_postal) values ("Associação
Desportiva de Ovar",'1920-1-4',"Rua do Desporto nº4","3880-465");
```

```

insert into equipa (nome,data_fundacao,rua,codigo_postal) values ("Os
Ceramistas",'2010-10-10',"Lugar da Bouça nº70","4750-484");

insert into equipa (nome,data_fundacao,rua,codigo_postal) values("SC Braga",'1950-5-
13',"Praça da República nº50","4000-039");


insert into atleta values("456135469","Diogo","Vasconcelos",'1999-1-
1',"Portugues","M","pokemonlovesu2@gmail.com",'50.0','150.0',"Rua do Amor nº 69",4,"4750-
484");

insert into atleta values("846100164","Gonçalo","Pinto",'1999-2-
2',"Português","M","gon_pin@gmail.com",'75.0','175.0',"Rua da Liberdade nº 25",3,"3880-465");

insert into atleta values("967852145","José","Costa",'1998-3-
3',"Indiano","M","helpmewithmycurry@gmail.com",'100.0','180.0',"Rua do Caril nº 100",1,"4750-
600");

insert into atleta values("326455115","João","Parente",'1999-4-
4',"Portugues","M","papij@gmail.com",'70.0','177.0',"Rua do Ouro nº 420",2,"4910-392");

insert into atleta values("456123456","Pedro","Sánchez",'1970-5-
5',"Espanhol","M","barcelona@gmail.com",'85.0','185.0',"Rua da Prata nº 112",1,"4800-034");

insert into atleta values("451351515","Emmanuel","Macron",'1975-6-
6',"Frances","M","eder@gmail.com",'78.7','178.7',"Avenida Paris nº4",5,"4700-188");

insert into atleta values("789542132","Nicolás","Maduro",'1980-7-
7',"Venezuelano","M","verde@gmail.com",'105.0','190.0',"Rua das Caracas nº99",3,"4000-039");

insert into atleta values("421221511","Xi","Jinping",'1985-8-
8',"Chines","M","ping@gmail.com",'90.0','185.0',"Rua do Arroz nº100",1,"4750-600");

insert into atleta values("218454844","João","Lourenço",'1990-9-
9',"Angolano","F","lour_o@gmail.com",'65.0','170.0',"Travessa de Luanda nº20",4,"4750-484");

insert into atleta values("121654151","Maria","Araújo",'1995-10-
10',"Portuguesa","F","influencer@gmail.com",'50.0','160.0',"Rua do Cobre nº11",4,"4800-034");

insert into atleta values("781214845","Leonor","Martins",'2000-11-
11',"Portuguesa","F","imagirl@gmail.com",'77.7','177.7',"Rua do Rego nº 505",5,"4700-188");

insert into atleta values("103548484","Matilde","Rodrigues",'2002-12-
12',"Portuguesa","F","sougira@gmail.com",'66.6','166.6',"Rua da Lomba nº 33",3,"3880-465");

insert into atleta values("784218111","Beatriz","Lopes",'2004-12-
24',"Portuguesa","F","love1D@gmail.com",'50.0','145.0',"Avenida 8 de Julho nº8",3,"4000-039");

insert into atleta values("103582472","Carolina","Silva",'2010-12-
25',"Portuguesa","F","funklover@gmail.com",'40.0','140.0',"Rua de Vilarinho nº 111",2,"4910-
392");

insert into atleta values("721811241","Mariana","Santos",'2010-12-
31',"Portuguesa","F","family.carreira.forever@gmail.com",'55.5','155.5',"Rua da Constituição nº
40",5,"4700-188");

```

```

insert into categoria values("967852145","Triplo Salto",4);
insert into categoria values("967852145","Salto em Comprimento Normal",4);
insert into categoria values("967852145","100 metros",2);
insert into categoria values("967852145","400 metros",2);
insert into categoria values("967852145","Corrida de Estafetas Normal",3);
insert into categoria values("967852145","Salto em Altura sem vara",5);
insert into categoria values("967852145","Salto em Altura com vara",5);
insert into categoria values("456123456","100 metros",1);
insert into categoria values("456123456","200 metros",1);
insert into categoria values("456123456","400 metros",1);
insert into categoria values("456123456","100 metros",2);
insert into categoria values("456123456","400 metros",2);
insert into categoria values("456123456","Corrida de Estafetas Normal",3);
insert into categoria values("421221511","Lançamento de peso",6);
insert into categoria values("421221511","Lançamento de dardo",6);
insert into categoria values("421221511","Lançamento de disco",6);
insert into categoria values("421221511","Lançamento de martelo",6);
insert into categoria values("421221511","1500 metros",1);
insert into categoria values("421221511","3000 metros",1);
insert into categoria values("421221511","3000 metros",2);
insert into categoria values("326455115","Triplo Salto",4);
insert into categoria values("326455115","Salto em Comprimento Normal",4);
insert into categoria values("326455115","100 metros",2);
insert into categoria values("326455115","400 metros",2);
insert into categoria values("326455115","Corrida de Estafetas Normal",3);
insert into categoria values("326455115","Salto em Altura sem vara",5);
insert into categoria values("326455115","Salto em Altura com vara",5);
insert into categoria values("103582472","100 metros",1);
insert into categoria values("103582472","200 metros",1);
insert into categoria values("103582472","400 metros",1);
insert into categoria values("103582472","100 metros",2);
insert into categoria values("103582472","400 metros",2);
insert into categoria values("103582472","Corrida de Estafetas Normal",3);
insert into categoria values("789542132","Lançamento de peso",6);
insert into categoria values("789542132","Lançamento de disco",6);
insert into categoria values("789542132","Lançamento de martelo",6);
insert into categoria values("789542132","1500 metros",1);
insert into categoria values("846100164","Triplo Salto",4);
insert into categoria values("846100164","100 metros",1);
insert into categoria values("846100164","Corrida de Estafetas Normal",3);

```

```

insert into categoria values("103548484","100 metros",1);
insert into categoria values("103548484","400 metros",2);
insert into categoria values("103548484","Corrida de Estafetas Normal",3);
insert into categoria values("784218111","200 metros",1);
insert into categoria values("784218111","400 metros",2);
insert into categoria values("784218111","Corrida de Estafetas Normal",3);
insert into categoria values("456135469","Triplo Salto",4);
insert into categoria values("456135469","Salto em Comprimento Normal",4);
insert into categoria values("456135469","100 metros",2);
insert into categoria values("456135469","400 metros",2);
insert into categoria values("456135469","Corrida de Estafetas Normal",3);
insert into categoria values("218454844","Salto em Comprimento Normal",4);
insert into categoria values("218454844","Corrida de Estafetas Normal",3);
insert into categoria values("218454844","Salto em Altura com vara",5);
insert into categoria values("121654151","1000 metros",1);
insert into categoria values("121654151","500 metros",2);
insert into categoria values("451351515","Triplo Salto",4);
insert into categoria values("451351515","Salto em Comprimento Normal",4);
insert into categoria values("451351515","100 metros",2);
insert into categoria values("451351515","400 metros",2);
insert into categoria values("451351515","Corrida de Estafetas Normal",3);
insert into categoria values("451351515","Salto em Altura sem vara",5);
insert into categoria values("451351515","Salto em Altura com vara",5);
insert into categoria values("781214845","Lançamento de peso",6);
insert into categoria values("781214845","Lançamento de dardo",6);
insert into categoria values("781214845","Lançamento de disco",6);
insert into categoria values("781214845","Lançamento de martelo",6);
insert into categoria values("721811241","100 metros",1);
insert into categoria values("721811241","100 metros",2);

```

```

insert into teste_clinico values("967852145","870125901",'10.50','2016-1-5
14:30:00',"Colesterol Alto",false,0);
insert into teste_clinico values("967852145","123456798",'19.99','2017-3-10
15:00:00',"Distensão muscular",false,0);
insert into teste_clinico values("967852145","123456789",'10.00','2018-6-15
15:30:30',"Rotina",false,0);
insert into teste_clinico values("967852145","654781256",'9.99','2019-9-20
16:00:00',"Ansiedade",false,0);
insert into teste_clinico values("967852145","123794561",'15.00','2016-12-26
16:30:00',"Excesso de Peso",false,0);

```

```

insert into teste_clinico values("456123456","999456123",'10.00','2017-1-30
09:00:00',"Consumo exagerado de gordura e açúcar",false,0);
insert into teste_clinico values("456123456","745212389",'50.00','2018-2-3
09:30:00',"Ruptura de ligamento",false,0);
insert into teste_clinico values("456123456","320126902",'4.99','2019-3-3
10:00:00',"Antidopagem",false,0);
insert into teste_clinico values("421221511","999888777",'2.99','2016-5-5
10:30:00',"Rotina",false,0);
insert into teste_clinico values("421221511","789456122",'24.99','2017-7-7
11:00:00',"Depressão",false,0);
insert into teste_clinico values("326455115","870125901",'18.99','2018-10-10
11:30:00',"Hipertensão",false,0);
insert into teste_clinico values("326455115","999456123",'7.99','2019-6-1
12:00:00',"Stress",false,0);
insert into teste_clinico values("326455115","870125901",'14.99','2019-7-1
12:30:00',"Diabetes",false,0);
insert into teste_clinico values("103582472","123456798",'16.99','2019-6-1
13:00:00',"Tendinite",false,0);
insert into teste_clinico values("103582472","654781256",'29.99','2019-7-2
13:00:00',"Transtorno Obsessivo Compulsivo",false,0);
insert into teste_clinico values("846100164","999888777",'2.99','2018-6-13
16:30:00',"Rotina",false,0);
insert into teste_clinico values("846100164","123456789",'4.99','2019-6-5
16:30:00',"Antidopagem",false,0);
insert into teste_clinico values("789542132","123794561",'15.00','2018-4-26
17:00:00',"Excesso de Peso",false,0);
insert into teste_clinico values("789542132","320126902",'4.99','2019-6-5
17:00:00',"Antidopagem",false,0);
insert into teste_clinico values("103548484","123794561",'10.00','2018-12-18
17:30:00',"Definição de Dieta",false,0);
insert into teste_clinico values("103548484","999888777",'4.99','2019-6-5
17:30:00',"Antidopagem",false,0);
insert into teste_clinico values("784218111","123794561",'10.00','2018-12-18
18:00:00',"Definição de Dieta",false,0);
insert into teste_clinico values("784218111","123456789",'4.99','2019-6-5
18:00:00',"Antidopagem",false,0);
insert into teste_clinico values("456135469","123456798",'39.99','2016-1-27
08:00:00',"Distensão muscular",false,0);
insert into teste_clinico values("456135469","745212389",'49.00','2017-12-27
20:00:00',"Ruptura de ligamento",false,0);

```



```

insert into teste_clinico values("218454844","123456798",'59.99','2016-12-27
08:00:00',"Distensão muscular",false,0);
insert into teste_clinico values("218454844","745212389",'69.00','2017-1-27
20:00:00',"Ruptura de ligamento",false,0);
insert into teste_clinico values("121654151","123456798",'79.99','2018-1-27
08:00:00',"Distensão muscular",false,0);
insert into teste_clinico values("121654151","745212389",'89.00','2018-12-27
20:00:00',"Ruptura de ligamento",false,0);
insert into teste_clinico values("451351515","870125901",'18.99','2019-12-9
10:30:00',"Hipertensão",false,0);
insert into teste_clinico values("451351515","999888777",'2.99','2019-12-10
10:30:00',"Rotina",false,0);
insert into teste_clinico values("781214845","999456123",'7.99','2019-12-11
10:30:00',"Stress",false,0);
insert into teste_clinico values("781214845","999888777",'2.99','2019-12-12
10:30:00',"Rotina",false,0);
insert into teste_clinico values("721811241","870125901",'14.99','2019-12-13
10:30:00',"Diabetes",false,0);
insert into teste_clinico values("721811241","999888777",'2.99','2019-12-20
10:30:00',"Rotina",false,0);
insert into teste_clinico values("967852145","123456789",null,'2020-1-6
16:20:00',"Rotina",true,null);
insert into teste_clinico values("456123456","320126902",null,'2020-1-7
16:20:00',"Antidopagem",true,null);
insert into teste_clinico values("421221511","999888777",null,'2020-1-8
16:20:00',"Rotina",true,null);
insert into teste_clinico values("326455115","123456789",null,'2020-1-9
16:20:00',"Rotina",true,null);
insert into teste_clinico values("103582472","320126902",null,'2020-1-10
16:20:00',"Antidopagem",true,null);
insert into teste_clinico values("846100164","999888777",null,'2020-1-13
16:20:00',"Rotina",true, null);
insert into teste_clinico values("789542132","123456789",null,'2020-1-14
16:20:00',"Rotina",true, null) ;
insert into teste_clinico values("103548484","320126902",null,'2020-1-15
16:20:00',"Antidopagem",true,null);
insert into teste_clinico values("784218111","999888777",null,'2020-1-16
16:20:00',"Rotina",true, null);
insert into teste_clinico values("456135469","123456789",null,'2020-1-17
16:20:00',"Rotina",true, null);

```

```

insert into teste_clinico values("218454844","320126902",null,'2020-1-20
16:20:00',"Antidopagem",true, null);
insert into teste_clinico values("121654151","999888777",null,'2020-1-21
16:20:00',"Rotina",true, null);
insert into teste_clinico values("451351515","123456789",null,'2020-1-22
16:20:00',"Rotina",true, null);
insert into teste_clinico values("781214845","320126902",null,'2020-1-23
16:20:00',"Antidopagem",true, null);
insert into teste_clinico values("721811241","999888777",null,'2020-1-24
16:20:00',"Rotina",true, null);
insert into teste_clinico values("456135469","123794561",'9.99','2019-12-16
12:00:00',"Definição de Dieta",false, -1);
insert into teste_clinico values("846100164","999456123",'19.99','2019-12-17
12:00:00',"Stress",false, 1);
insert into teste_clinico values("967852145","745212389",'29.99','2019-12-18
12:00:00',"Tendinite",false, 1);
insert into teste_clinico values("326455115","999888777",'39.99','2019-12-19
12:00:00',"Antidopagem",false, 1);
insert into teste_clinico values("451351515","789456122",'49.99','2019-12-20
12:00:00',"Depressão",false, -1);

```

Anexo C

```
CREATE (cp1 : CodigoPostal {localidade: 'Barcelos', codigo: '4750-600'})
CREATE (cp2: CodigoPostal {localidade: 'Barcelos', codigo: '4750-484'})
CREATE (cp3 : CodigoPostal {localidade: 'Guimarães', codigo: '4800-034'})
CREATE (cp4 : CodigoPostal {localidade: 'Braga', codigo: '4700-188'})
CREATE (cp5 : CodigoPostal {localidade: 'Ovar', codigo: '3880-465'})
CREATE (cp6 : CodigoPostal {localidade: 'Porto', codigo: '4000-039'})
CREATE (cp7 : CodigoPostal {localidade: 'Vila praia de Âncora', codigo: '4910-392'})
```

```
CREATE (e1 : Especialidade {designacao : 'Cardiologia'})
CREATE (e2 : Especialidade {designacao : 'Ortopedia'})
CREATE (e3 : Especialidade {designacao : 'Clinica Geral'})
CREATE (e4: Especialidade {designacao : 'Psicologia'})
CREATE (e5 : Especialidade {designacao : 'Nutrição'})
```

```
CREATE (mo1 : Modalidade {designacao : 'Corrida de Pista'})
CREATE (mo2: Modalidade {designacao : 'Corrida de Obstáculos'})
CREATE (mo3 : Modalidade {designacao : 'Corrida de Estafetas'})
CREATE (mo4 : Modalidade {designacao : 'Salto em Comprimento'})
CREATE (mo5 : Modalidade {designacao : 'Salto em Altura'})
CREATE (mo6 : Modalidade {designacao : 'Lançamentos'})
```

```
CREATE (eq1 : Equipa {nome: 'Oliveira Sports', dataFundacao: '2012-12-12' , rua: 'Avenida Oliveira nº1'})
```

```
CREATE (eq2 : Equipa {nome: 'Sociedade Recreativa de Vila Praia de Âncora', dataFundacao: '1990-12-31' , rua: 'Rua dos Emigrantes nº99'})
```

```
CREATE (eq3: Equipa {nome: 'Associação Desportiva de Ovar', dataFundacao: '1920-1-4', rua: 'Rua do Desporto nº4'})
```

```
CREATE (eq4 : Equipa {nome: 'Os Ceramistas', dataFundacao: '2010-10-10', rua: 'Lugar da Bouça nº70'})
```

```
CREATE (eq5 : Equipa {nome: 'SC Braga', dataFundacao: '1950-5-13' , rua: 'Praça da República nº50'})
```

```
CREATE (eq1)-[:TEM_SEDE_EM {}]->(cp1)
CREATE (eq2)-[:TEM_SEDE_EM {}]->(cp7)
CREATE (eq3)-[:TEM_SEDE_EM {}]->(cp5)
CREATE (eq4)-[:TEM_SEDE_EM {}]->(cp2)
CREATE (eq5)-[:TEM_SEDE_EM {}]->(cp4)
```

```

CREATE (m1 : Medico {nif: 123794561, nome: 'Maria Cunha', dataNascimento: '1990-12-30'})
CREATE (m2 : Medico {nif: 789456122, nome: 'Ana Esteves', dataNascimento: '1985-11-25'})
CREATE (m3 : Medico {nif: 654781256, nome: 'Paula Roberto', dataNascimento: '1980-10-20'})
CREATE (m4 : Medico {nif: 123456789, nome: 'José Baião', dataNascimento: '1975-9-15'})
CREATE (m5 : Medico {nif: 123456798, nome: 'Carlos Noronha', dataNascimento: '1970-8-10'})
CREATE (m6 : Medico {nif: 870125901, nome: 'Alfredo Fernandes', dataNascimento: '1965-7-5'})
CREATE (m7 : Medico {nif: 320126902, nome: 'Manuel Oliveira', dataNascimento: '1960-6-3'})
CREATE (m8 : Medico {nif: 745212389, nome: 'Joaquim Mendes', dataNascimento: '1955-5-2'})
CREATE (m9 : Medico {nif: 999456123, nome: 'Augusto Silva', dataNascimento: '1950-4-1'})
CREATE (m10 : Medico {nif: 999888777, nome: 'Acácio Santos', dataNascimento: '1945-3-30'})

```

```

CREATE (m1)-[:RESIDE {}]->(cp1)
CREATE (m2)-[:RESIDE {}]->(cp2)
CREATE (m3)-[:RESIDE {}]->(cp3)
CREATE (m4)-[:RESIDE {}]->(cp4)
CREATE (m5)-[:RESIDE {}]->(cp5)
CREATE (m6)-[:RESIDE {}]->(cp6)
CREATE (m7)-[:RESIDE {}]->(cp7)
CREATE (m8)-[:RESIDE {}]->(cp3)
CREATE (m9)-[:RESIDE {}]->(cp4)
CREATE (m10)-[:RESIDE {}]->(cp4)

```

```

CREATE (m1)-[:ESPECIALISTA_EM {}]->(e5)
CREATE (m2)-[:ESPECIALISTA_EM {}]->(e4)
CREATE (m3)-[:ESPECIALISTA_EM {}]->(e4)
CREATE (m4)-[:ESPECIALISTA_EM {}]->(e3)
CREATE (m5)-[:ESPECIALISTA_EM {}]->(e2)
CREATE (m6)-[:ESPECIALISTA_EM {}]->(e1)
CREATE (m7)-[:ESPECIALISTA_EM {}]->(e3)
CREATE (m8)-[:ESPECIALISTA_EM {}]->(e2)
CREATE (m9)-[:ESPECIALISTA_EM {}]->(e1)

```

CREATE (m10)-[:ESPECIALISTA_EM {}]->(e3)

CREATE (a1 : Atleta {nif:456135469, nome: 'Diogo Vasconcelos', dataNascimento: '1999-1-1', nacionalidade: 'Portugues', genero: 'M', email:'pokemonlovesu2@gmail.com', peso: 50.0, altura: 150.0, rua:'Rua do Amor nº 69'})

CREATE (a2 : Atleta {nif:846100164, nome: 'Gonçalo Pinto', dataNascimento: '1999-2-2', nacionalidade: 'Portugues', genero: 'M', email:'gon_pin@gmail.com', peso: 75.0, altura: 175.0, rua:'Rua da Liberdade nº 25'})

CREATE (a3 : Atleta {nif:967852145, nome: 'José Costa', dataNascimento: '1998-3-3', nacionalidade: 'Indiano', genero: 'M', email:'helpmewithmycurry@gmail.com', peso: 100.0, altura: 180.0, rua:'Rua do Caril nº 100'})

CREATE (a4 : Atleta {nif:326455115, nome: 'João Parente', dataNascimento: '1999-4-4', nacionalidade: 'Portugues', genero: 'M', email:'papj@gmail.com', peso: 70.0, altura: 177.0, rua:'Rua do Ouro nº 420'})

CREATE (a5 : Atleta {nif:456123456, nome: 'Pedro Sánchez', dataNascimento: '1970-5-5', nacionalidade: 'Espanhol', genero: 'M', email:'barcelona@gmail.com', peso: 85.0, altura: 185.0, rua:'Rua da Prata nº112'})

CREATE (a6 : Atleta {nif:451351515, nome: 'Emmanuel Macron', dataNascimento: '1975-6-6', nacionalidade: 'Frances', genero: 'M', email:'eder@gmail.com', peso: 78.7, altura: 178.7, rua:'Avenida Paris nº4'})

CREATE (a7 : Atleta {nif:789542132, nome: 'Nicolás Maduro', dataNascimento: '1980-7-7', nacionalidade: 'Venezuelano', genero: 'M', email:'verde@gmail.com', peso:105, altura: 190.0, rua:'Rua das Caracas nº99'})

CREATE (a8 : Atleta {nif:421221511, nome: 'Xi Jinping', dataNascimento: '1985-8-8', nacionalidade: 'Chines', genero: 'M', email:'ping@gmail.com', peso: 90.0, altura: 185.0, rua:'Rua do Arroz nº100'})

CREATE (a9 : Atleta {nif:218454844, nome: 'João Lourenço', dataNascimento: '1990-9-9', nacionalidade: 'Angolano', genero: 'M', email:'lour_o@gmail.com', peso: 65.0, altura: 170.0, rua:'Travessa de Luanda nº20'})

CREATE (a10 : Atleta {nif:121654151, nome: 'Maria Araújo', dataNascimento: '1995-10-10', nacionalidade: 'Portuguesa', genero: 'F', email:'influencer@gmail.com', peso: 50.0, altura: 160.0, rua:'Rua do Cobre nº11'})

CREATE (a11 : Atleta {nif:781214845, nome: 'Leonor Martins', dataNascimento: '2000-11-11', nacionalidade: 'Portuguesa', genero: 'F', email:'imagirl@gmail.com', peso: 77.7, altura: 177.7, rua:'Rua do Rego nº 505'})

CREATE (a12 : Atleta {nif:103548484, nome: 'Matilde Rodrigues', dataNascimento: '2002-12-12', nacionalidade: 'Portuguesa', genero: 'F', email:'sougira@gmail.com', peso: 66.6, altura: 166.6, rua:'Rua da Lomba nº 33'})

CREATE (a13 : Atleta {nif:784218111, nome: 'Beatriz Lopes', dataNascimento: '2004-12-24', nacionalidade: 'Portuguesa', genero: 'F', email:'love1D@gmail.com', peso: 50.0, altura: 145.0, rua:'Avenida 8 de Julho nº8'})

CREATE (a14 : Atleta {nif:103582472, nome: 'Carolina Silva', dataNascimento: '2010-12-25', nacionalidade: 'Portuguesa', genero: 'F', email:'funklover@gmail.com', peso: 40.0, altura: 140.0, rua:'Rua de Vilarinho nº 111'})

CREATE (a15 : Atleta {nif:721811241, nome: 'Mariana Santos', dataNascimento: '2010-12-31', nacionalidade: 'Portuguesa', genero: 'F', email:'family.carreira.forever@gmail.com', peso: 55.5, altura: 155.5, rua:'Rua da Constituição nº 40'})

CREATE (a1)-[:RESIDE {}]->(cp2)

CREATE (a2)-[:RESIDE {}]->(cp5)

CREATE (a3)-[:RESIDE {}]->(cp1)

CREATE (a4)-[:RESIDE {}]->(cp7)

CREATE (a5)-[:RESIDE {}]->(cp3)

CREATE (a6)-[:RESIDE {}]->(cp4)

CREATE (a7)-[:RESIDE {}]->(cp6)

CREATE (a8)-[:RESIDE {}]->(cp1)

CREATE (a9)-[:RESIDE {}]->(cp2)

CREATE (a10)-[:RESIDE {}]->(cp3)

CREATE (a11)-[:RESIDE {}]->(cp4)

CREATE (a12)-[:RESIDE {}]->(cp5)

CREATE (a13)-[:RESIDE {}]->(cp6)

CREATE (a14)-[:RESIDE {}]->(cp7)

CREATE (a15)-[:RESIDE {}]->(cp4)

CREATE (a1)-[:ATLETA_DO {}]->(eq4)

CREATE (a2)-[:ATLETA_DO {}]->(eq3)

CREATE (a3)-[:ATLETA_DO {}]->(eq1)

CREATE (a4)-[:ATLETA_DO {}]->(eq2)

CREATE (a5)-[:ATLETA_DO {}]->(eq1)

CREATE (a6)-[:ATLETA_DO {}]->(eq5)

CREATE (a7)-[:ATLETA_DO {}]->(eq3)

CREATE (a8)-[:ATLETA_DO {}]->(eq1)

CREATE (a9)-[:ATLETA_DO {}]->(eq4)

CREATE (a10)-[:ATLETA_DO {}]->(eq4)

CREATE (a11)-[:ATLETA_DO {}]->(eq5)

CREATE (a12)-[:ATLETA_DO {}]->(eq3)

CREATE (a13)-[:ATLETA_DO {}]->(eq3)

CREATE (a14)-[:ATLETA_DO {}]->(eq2)

CREATE (a15)-[:ATLETA_DO {}]->(eq5)

CREATE (a3)-[:PRATICA {categoria:'Triplo Salto'}]->(mo4)

CREATE (a3)-[:PRATICA {categoria:'Normal'}]->(mo4)

CREATE (a3)-[:PRATICA {categoria:'100m'}]->(mo2)

CREATE (a3)-[:PRATICA {categoria:'400m'}]->(mo2)

CREATE (a3)-[:PRATICA {categoria:'Normal'}]->(mo3)

CREATE (a3)-[:PRATICA {categoria:'Sem vara'}]->(mo5)

CREATE (a3)-[:PRATICA {categoria:'Com vara'}]->(mo5)

CREATE (a5)-[:PRATICA {categoria:'100m'}]->(mo1)

CREATE (a5)-[:PRATICA {categoria:'200m'}]->(mo1)

CREATE (a5)-[:PRATICA {categoria:'400m'}]->(mo1)

CREATE (a5)-[:PRATICA {categoria:'100m'}]->(mo2)

CREATE (a5)-[:PRATICA {categoria:'400m'}]->(mo2)

CREATE (a5)-[:PRATICA {categoria:'Normal'}]->(mo3)

CREATE (a8)-[:PRATICA {categoria:'Peso'}]->(mo6)

CREATE (a8)-[:PRATICA {categoria:'Dardo'}]->(mo6)

CREATE (a8)-[:PRATICA {categoria:'Disco'}]->(mo6)

CREATE (a8)-[:PRATICA {categoria:'Martelo'}]->(mo6)

CREATE (a8)-[:PRATICA {categoria:'1500m'}]->(mo1)

CREATE (a8)-[:PRATICA {categoria:'3000m'}]->(mo1)

CREATE (a8)-[:PRATICA {categoria:'1500m'}]->(mo1)

CREATE (a8)-[:PRATICA {categoria:'3000m'}]->(mo1)

CREATE (a8)-[:PRATICA {categoria:'3000m'}]->(mo2)

CREATE (a4)-[:PRATICA {categoria:'Triplo Salto'}]->(mo4)

CREATE (a4)-[:PRATICA {categoria:'Normal'}]->(mo4)

CREATE (a4)-[:PRATICA {categoria:'100m'}]->(mo2)

CREATE (a4)-[:PRATICA {categoria:'400m'}]->(mo2)

CREATE (a4)-[:PRATICA {categoria:'Normal'}]->(mo3)

CREATE (a4)-[:PRATICA {categoria:'Sem vara'}]->(mo5)

CREATE (a4)-[:PRATICA {categoria:'Com vara'}]->(mo5)

CREATE (a14)-[:PRATICA {categoria:'100m'}]->(mo1)

CREATE (a14)-[:PRATICA {categoria:'200m'}]->(mo1)

CREATE (a14)-[:PRATICA {categoria:'400m'}]->(mo1)

CREATE (a14)-[:PRATICA {categoria:'100m'}]->(mo2)

CREATE (a14)-[:PRATICA {categoria:'400m'}]->(mo2)

CREATE (a14)-[:PRATICA {categoria:'Normal'}]->(mo3)

```

CREATE (a2)-[:PRATICA {categoria:'Triplo Salto'}]->(mo4)
CREATE (a2)-[:PRATICA {categoria:'100m'}]->(mo1)
CREATE (a2)-[:PRATICA {categoria:'Normal'}]->(mo3)
CREATE (a7)-[:PRATICA {categoria:'Peso'}]->(mo6)
CREATE (a7)-[:PRATICA {categoria:'Disco'}]->(mo6)
CREATE (a7)-[:PRATICA {categoria:'Martelo'}]->(mo6)
CREATE (a7)-[:PRATICA {categoria:'1500m'}]->(mo1)
CREATE (a12)-[:PRATICA {categoria:'100m'}]->(mo1)
CREATE (a12)-[:PRATICA {categoria:'400m'}]->(mo2)
CREATE (a12)-[:PRATICA {categoria:'Normal'}]->(mo3)
CREATE (a13)-[:PRATICA {categoria:'200m'}]->(mo1)
CREATE (a13)-[:PRATICA {categoria:'400m'}]->(mo2)
CREATE (a13)-[:PRATICA {categoria:'Normal'}]->(mo3)

```

```

CREATE (a1)-[:PRATICA {categoria:'Triplo Salto'}]->(mo4)
CREATE (a1)-[:PRATICA {categoria:'Normal'}]->(mo4)
CREATE (a1)-[:PRATICA {categoria:'100m'}]->(mo2)
CREATE (a1)-[:PRATICA {categoria:'400m'}]->(mo2)
CREATE (a1)-[:PRATICA {categoria:'Normal'}]->(mo3)
CREATE (a9)-[:PRATICA {categoria:'Normal'}]->(mo4)
CREATE (a9)-[:PRATICA {categoria:'Normal'}]->(mo3)
CREATE (a9)-[:PRATICA {categoria:'Com vara'}]->(mo5)
CREATE (a10)-[:PRATICA {categoria:'1000m'}]->(mo1)
CREATE (a10)-[:PRATICA {categoria:'500m'}]->(mo2)

```

```

CREATE (a6)-[:PRATICA {categoria:'Normal'}]->(mo4)
CREATE (a6)-[:PRATICA {categoria:'Normal'}]->(mo4)
CREATE (a6)-[:PRATICA {categoria:'100m'}]->(mo2)
CREATE (a6)-[:PRATICA {categoria:'400m'}]->(mo2)
CREATE (a6)-[:PRATICA {categoria:'Normal'}]->(mo3)
CREATE (a6)-[:PRATICA {categoria:'Sem vara'}]->(mo5)
CREATE (a6)-[:PRATICA {categoria:'Com vara'}]->(mo5)
CREATE (a11)-[:PRATICA {categoria:'Peso'}]->(mo6)
CREATE (a11)-[:PRATICA {categoria:'Dardo'}]->(mo6)
CREATE (a11)-[:PRATICA {categoria:'Disco'}]->(mo6)
CREATE (a11)-[:PRATICA {categoria:'Martelo'}]->(mo6)
CREATE (a15)-[:PRATICA {categoria:'100m'}]->(mo1)
CREATE (a15)-[:PRATICA {categoria:'100m'}]->(mo2)

```



```

CREATE (a3)-[:E_TESTADO_POR{descricao:'Colesterol Alto',data:'2016-1-5
14:30:00',preco:10.50,agendado:false,faltou:0}]->(m6)
CREATE (a3)-[:E_TESTADO_POR{descricao:'Distensão muscular',data:'2017-3-10
15:00:00',preco:19.99,agendado:false,faltou:0}]->(m5)
CREATE (a3)-[:E_TESTADO_POR{descricao:'Rotina',data:'2018-6-15
15:30:00',preco:10.00,agendado:false,faltou:0}]->(m4)
CREATE (a3)-[:E_TESTADO_POR{descricao:'Ansiedade',data:'2019-9-20
16:00:00',preco:9.99,agendado:false,faltou:0}]->(m3)
CREATE (a3)-[:E_TESTADO_POR{descricao:'Excesso de Peso',data:'2016-12-26
16:30:00',preco:15.00,agendado:false,faltou:0}]->(m1)
CREATE (a5)-[:E_TESTADO_POR{descricao:'Consumo exagerado de gordura e
açúcar',data:'2017-1-30 09:00:00',preco:10.00,agendado:false,faltou:0}]->(m9)
CREATE (a5)-[:E_TESTADO_POR{descricao:'Ruptura de ligamento',data:'2018-2-3
09:30:00',preco:50.00,agendado:false,faltou:0}]->(m8)
CREATE (a5)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2019-3-3
10:00:00',preco:4.99,agendado:false,faltou:0}]->(m7)
CREATE (a8)-[:E_TESTADO_POR{descricao:'Rotina',data:'2016-5-5
10:30:00',preco:2.99,agendado:false,faltou:0}]->(m10)
CREATE (a8)-[:E_TESTADO_POR{descricao:'Depressão',data:'2017-7-7
11:00:00',preco:24.99,agendado:false,faltou:0}]->(m2)

CREATE (a4)-[:E_TESTADO_POR{descricao:'Hipertensão',data:'2018-10-10
11:30:00',preco:18.99,agendado:false,faltou:0}]->(m6)
CREATE (a4)-[:E_TESTADO_POR{descricao:'Stress',data:'2019-6-1
12:00:00',preco:7.99,agendado:false,faltou:0}]->(m9)
CREATE (a4)-[:E_TESTADO_POR{descricao:'Diabetes',data:'2019-7-1
12:30:00',preco:14.99,agendado:false,faltou:0}]->(m6)
CREATE (a14)-[:E_TESTADO_POR{descricao:'Tendinite',data:'2018-6-2
13:00:00',preco:16.99,agendado:false,faltou:0}]->(m5)
CREATE (a14)-[:E_TESTADO_POR{descricao:'Transtorno Obsessivo
Compulsivo',data:'2019-7-2 13:30:00',preco:29.99,agendado:false,faltou:0}]->(m3)

CREATE (a2)-[:E_TESTADO_POR{descricao:'Rotina',data:'2018-6-13
16:30:00',preco:2.99,agendado:false,faltou:0}]->(m10)
CREATE (a2)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2019-6-5
16:30:00',preco:4.99,agendado:false,faltou:0}]->(m4)
CREATE (a7)-[:E_TESTADO_POR{descricao:'Excesso de Peso',data:'2018-4-26
17:00:00',preco:15.00,agendado:false,faltou:0}]->(m1)
CREATE (a7)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2019-6-5
17:00:00',preco:4.99,agendado:false,faltou:0}]->(m7)

```

```

CREATE (a12)-[:E_TESTADO_POR{descricao:'Definição de Dieta',data:'2018-12-18
17:30:00',preco:10.00,agendado:false,faltou:0}]->(m1)
CREATE (a12)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2019-6-5
17:30:00',preco:4.99,agendado:false,faltou:0}]->(m10)
CREATE (a13)-[:E_TESTADO_POR{descricao:'Definição de Dieta',data:'2018-12-18
18:00:00',preco:10.00,agendado:false,faltou:0}]->(m1)
CREATE (a13)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2019-6-5
18:00:00',preco:4.99,agendado:false,faltou:0}]->(m4)

CREATE (a1)-[:E_TESTADO_POR{descricao:'Distensão muscular',data:'2016-1-27
8:00:00',preco:39.99,agendado:false,faltou:0}]->(m5)
CREATE (a1)-[:E_TESTADO_POR{descricao:'Ruptura de ligamento',data:'2017-12-27
20:00:00',preco:49.00,agendado:false,faltou:0}]->(m8)
CREATE (a9)-[:E_TESTADO_POR{descricao:'Distensão muscular',data:'2016-12-27
8:00:00',preco:59.99,agendado:false,faltou:0}]->(m5)
CREATE (a9)-[:E_TESTADO_POR{descricao:'Ruptura de ligamento',data:'2017-1-27
20:00:00',preco:69.00,agendado:false,faltou:0}]->(m8)
CREATE (a10)-[:E_TESTADO_POR{descricao:'Distensão muscular',data:'2018-12-27
8:00:00',preco:79.99,agendado:false,faltou:0}]->(m5)
CREATE (a10)-[:E_TESTADO_POR{descricao:'Ruptura de ligamento',data:'2018-12-27
20:00:00',preco:89.00,agendado:false,faltou:0}]->(m8)

CREATE (a6)-[:E_TESTADO_POR{descricao:'Hipertensão',data:'2019-12-9
10:30:00',preco:18.99,agendado:false,faltou:0}]->(m6)
CREATE (a6)-[:E_TESTADO_POR{descricao:'Rotina',data:'2019-12-10
10:30:00',preco:2.99,agendado:false,faltou:0}]->(m10)
CREATE (a11)-[:E_TESTADO_POR{descricao:'Stress',data:'2019-12-11
10:30:00',preco:7.99,agendado:false,faltou:0}]->(m9)
CREATE (a11)-[:E_TESTADO_POR{descricao:'Rotina',data:'2019-12-12
10:30:00',preco:2.99,agendado:false,faltou:0}]->(m10)
CREATE (a15)-[:E_TESTADO_POR{descricao:'Diabetes',data:'2019-12-13
10:30:00',preco:14.99,agendado:false,faltou:0}]->(m6)
CREATE (a15)-[:E_TESTADO_POR{descricao:'Rotina',data:'2019-12-20
10:30:00',preco:2.99,agendado:false,faltou:0}]->(m10)

CREATE (a3)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-6
16:20:00',preco:"",agendado:true,faltou:"}]->(m4)
CREATE (a5)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2020-1-7
16:20:00',preco:"",agendado:true,faltou:"}]->(m7)

```

```

CREATE (a8)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-8
16:20:00',preco:'',agendado:true,faltou:''}]>-(m10)
CREATE (a4)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2020-1-9
16:20:00',preco:'',agendado:true,faltou:''}]>-(m4)
CREATE (a14)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-10
16:20:00',preco:'',agendado:true,faltou:''}]>-(m7)
CREATE (a2)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2020-1-13
16:20:00',preco:'',agendado:true,faltou:''}]>-(m10)
CREATE (a7)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-14
16:20:00',preco:'',agendado:true,faltou:''}]>-(m4)
CREATE (a12)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2020-1-15
16:20:00',preco:'',agendado:true,faltou:''}]>-(m7)
CREATE (a13)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-16
16:20:00',preco:'',agendado:true,faltou:''}]>-(m10)
CREATE (a1)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2020-1-17
16:20:00',preco:'',agendado:true,faltou:''}]>-(m4)
CREATE (a9)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-20
16:20:00',preco:'',agendado:true,faltou:''}]>-(m7)
CREATE (a10)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2020-1-21
16:20:00',preco:'',agendado:true,faltou:''}]>-(m10)
CREATE (a6)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-22
16:20:00',preco:'',agendado:true,faltou:''}]>-(m4)
CREATE (a11)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2020-1-23
16:20:00',preco:'',agendado:true,faltou:''}]>-(m7)
CREATE (a15)-[:E_TESTADO_POR{descricao:'Rotina',data:'2020-1-24
16:20:00',preco:'',agendado:true,faltou:''}]>-(m10)

CREATE (a1)-[:E_TESTADO_POR{descricao:'Definição de Dieta',data:'2019-12-16
12:00:00',preco:9.99,agendado:false,faltou:-1}]>-(m1)
CREATE (a2)-[:E_TESTADO_POR{descricao:'Stress',data:'2019-12-17
12:00:00',preco:19.99,agendado:false,faltou:1}]>-(m9)
CREATE (a3)-[:E_TESTADO_POR{descricao:'Tendinite',data:'2019-12-18
12:00:00',preco:29.99,agendado:false,faltou:1}]>-(m8)
CREATE (a4)-[:E_TESTADO_POR{descricao:'Antidopagem',data:'2019-12-19
12:00:00',preco:39.99,agendado:false,faltou:1}]>-(m10)
CREATE (a6)-[:E_TESTADO_POR{descricao:'Depressão',data:'2019-12-20
12:00:00',preco:49.99,agendado:false,faltou:-1}]>-(m2)

```