



Universidade do Minho

Mestrado Integrado em Engenharia Informática

Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2020/2021

Centro de Reabilitação Física

**João Ferreira (89497), José Manso (87961), José
Reis (87980), Rita Gomes (87960)**

Dezembro, 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Centro de Reabilitação Física

João Ferreira (89497), José Manso (87961), José Reis (87980), Rita Gomes (87960)

Dezembro, 2020

Resumo

O presente relatório tem como objetivo fazer uma apresentação da abordagem do grupo à implementação de um sistema de base de dados para um caso prático específico.

O grupo pretende planificar e desenvolver uma base de dados para um centro de reabilitação física, possibilitando uma fácil gestão da imensa quantidade de informação que circula neste tipo de estabelecimentos. Para tal, seguiu-se a metodologia apresentada e discutida no livro '*Connolly, T., Begg C., Database Systems, A Practical Approach to Design, Implementation, and Management, Addison-Wesley, 4ª Edição, 2004*', abordando-se metodicamente todas as etapas.

Numa primeira fase, procedemos ao levantamento de requisitos básicos e sua respetiva análise a fim de traçar os objetivos primordiais. Seguidamente, foi desenhado o modelo conceptual, tendo em vista o cumprimento de todos os requisitos recolhidos na fase anterior. O modelo conceptual englobou a identificação e caracterização das entidades, dos relacionamentos e dos atributos necessários para representar fielmente o sistema. Findos esses passos, foi produzido um Diagrama ER, utilizando o brModelo, que sintetiza toda essa análise. Posteriormente, o modelo conceptual foi traduzido para o respetivo modelo lógico e subsequentemente o modelo físico, utilizando o MySQL.

Por fim, foi realizado o povoamento da base de dados implementada e procedeu-se à validação da mesma através de alguns exemplos de utilização.

Área de Aplicação: Criação e desenvolvimento de uma base de dados.

Palavras-Chave: Base de Dados, Modelo Conceptual, Modelo Lógico, Modelo Físico, Entidade, Relacionamento, Atributo.

Índice

1. Introdução	6
1.1. Contextualização	6
1.2. Apresentação do Caso de Estudo	6
1.3. Motivação e Objetivos	7
1.4. Análise da viabilidade do processo	7
1.5. Estrutura do Relatório	7
2. Levantamento e análise de requisitos	8
2.1. Método de levantamento e de análise de requisitos adotado	8
2.2. Requisitos levantados	8
2.2.1. Requisitos de descrição	8
2.2.2. Requisitos de exploração	9
2.2.3. Requisitos de controlo	10
2.3. Análise e validação geral dos requisitos	10
3. Modelação Conceptual	10
3.1. Apresentação da abordagem de modelação realizada	10
3.2. Identificação e caracterização das entidades	11
3.3. Identificação e caracterização dos relacionamentos	12
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	12
3.5. Detalhe ou generalização de entidades	16
3.6. Apresentação e explicação do diagrama ER	17
3.7. Validação do modelo de dados produzido	17
4. Modelação Lógica	19
4.1. Construção e validação do modelo de dados lógico	19
4.1.1. Entidades	19
4.1.2. Relação entre Entidades	20
4.2. Desenho do modelo lógico	22
4.3. Validação do modelo através da normalização	23
4.4. Validação do modelo com interrogações do utilizador	24
4.5. Revisão do modelo lógico produzido	25
5. Implementação Física	25
5.1. Seleção do sistema de gestão de bases de dados	25

5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	26
5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)	31
5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)	32
5.5. Estimativa do espaço em disco da base de dados e taxa de crescimento anual	32
5.6. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	34
5.7. Revisão do sistema implementado	36
6. Povoamento da Base de Dados	36
7. Conclusões e Trabalho Futuro	39
Referências	41
Lista de Siglas e Acrónimos	42

Índice de Figuras

Figura 1 – Diagrama ER.

Figura 2 - Entidades.

Figura 3 - Relacionamento com a Entidade 'Contacto'.

Figura 4 - Relacionamento entre 'Terapeuta' e 'Máquina'.

Figura 5 - Relacionamento entre 'Terapeuta' e 'Material'.

Figura 6 - Relacionamento Ternário de 'Tratamento'.

Figura 7 - Desenho do modelo lógico do CRF.

Figura 8 - Criação da Tabela 'Contacto'.

Figura 9 - Criação da Tabela 'Utente'.

Figura 10 - Criação da Tabela 'Terapeuta'.

Figura 11 - Criação da Tabela 'Médico'.

Figura 12 - Criação da Tabela 'Material'.

Figura 13 - Criação da Tabela 'Maquina'.

Figura 14 - Criação da Tabela 'Tratamento'.

Figura 15 - Criação da Tabela 'Terapeuta_has_Material'.

Figura 16 - Criação da Tabela 'Terapeuta_has_Maquina'.

Figura 17 - Query do histórico de lesões de um determinado Utente, bem como o pagamento total efetuado no seu tratamento.

Figura 18 - Query de um relatório do stock existente para saber quais os materiais em falta.

Figura 19 - Query de uma lista das máquinas não funcionais.

Figura 20 - Query de uma listagem dos utentes que frequentaram o CRF num intervalo de tempo específico.

Figura 21 - Query de um histórico de utentes que tiveram uma Lesão de um determinado tipo.

Figura 22 - Query de uma lista de tratamentos de um determinado terapeuta

Figura 23 – View do stock existente

Figura 24 – View dos contactos existentes

Figura 25 – View da informação relativa à faturação

Índice de Tabelas

Tabela 1 - Caracterização das Entidades.

Tabela 2 - Caracterização dos Relacionamentos.

Tabela 3 - Caracterização da Associação dos Atributos com a Entidade 'Utente'.

Tabela 4 - Caracterização da Associação dos Atributos com a Entidade 'Médico'.

Tabela 5 - Caracterização da Associação dos Atributos com a Entidade 'Terapeuta'.

Tabela 6 - Caracterização da Associação dos Atributos com a Entidade 'Contacto'.

Tabela 7 - Caracterização da Associação dos Atributos com a Entidade 'Máquina'.

Tabela 8 - Caracterização da Associação dos Atributos com a Entidade 'Material'.

Tabela 9 - Caracterização da Associação dos Atributos com o Relacionamento 'Tratamento'.

Tabela 10 - Tamanho máximo por entrada da Tabela 'Utente'.

Tabela 11 - Tamanho máximo por entrada da Tabela 'Médico'.

Tabela 12 - Tamanho máximo por entrada da Tabela 'Terapeuta'.

Tabela 13 - Tamanho máximo por entrada da Tabela 'Contacto'.

Tabela 14 - Tamanho máximo por entrada da Tabela 'Máquina'.

Tabela 15 - Tamanho máximo por entrada da Tabela 'Material'.

Tabela 16 - Tamanho máximo por entrada da Tabela 'Tratamento'.

1. Introdução

Nesta secção será feita uma breve introdução acerca do projeto realizado na Unidade Curricular de Bases de Dados, assim como a respetiva contextualização, apresentação do caso de estudo, os motivos para a realização deste projeto e também os objetivos que pretendemos alcançar. Por fim, será também explicada a estrutura do relatório.

1.1. Contextualização

Num mundo onde cada vez mais a população é incentivada à prática desportiva por variadíssimos motivos, surgem regularmente casos de lesões físicas. Deste modo, são exigidos meios que visam tratar essas lesões o mais rápida e eficazmente possível.

Assim sendo, o grupo decidiu proceder à criação e desenvolvimento de um sistema de base de dados de maneira a armazenar todos os dados relativos a um centro de reabilitação física.

1.2. Apresentação do Caso de Estudo

Juntamente com o crescente número de pessoas a praticar exercício físico, é necessário haver centros suficientemente capazes de suportar toda a informação dos seus utentes, e o Centro de Reabilitação Física (CRF) tem de estar apto para tal.

Então, a nível de software, é preciso recorrer a uma implementação de um sistema de base de dados para garantir a organização e facilidade de acesso a qualquer tipo de informação que esteja armazenada.

A organização desta base de dados deverá estar estruturada de forma a conter uma ficha de utente, na qual teremos acesso a informações pessoais acerca do utente e o histórico de todos os tratamentos que realizou no nosso centro. Incluirá também informação relativa ao material usado nas sessões de tratamento, para pudermos gerir

a falta de material assim como proceder à reparação ou substituição dos materiais ou máquinas que não estejam funcionais/viáveis.

1.3. Motivação e Objetivos

Um centro de reabilitação física manipula muitos dados e, desta forma, é do interesse dos gestores do CRF tornar esta manipulação e consulta mais eficaz. A melhor forma de concretizar isto será através de uma base de dados, alcançando assim uma gestão mais robusta, bem como mais segurança nas operações sobre os dados. Por exemplo, ao alterarmos um dado arbitário, temos a garantia de que não iremos ser redundantes (por exemplo, na inserção de um funcionário que já existe) e de que todos os dados serão armazenados na sua integridade. Sendo assim, será mais eficiente arquivar e guardar qualquer informação relativa a qualquer entidade (e.g. utente, funcionário) ou atributo (e.g. idade, tipo de lesão).

1.4. Análise da viabilidade do processo

O desenvolvimento deste Sistema de Gestão de Base de Dados é de considerável importância para o centro de reabilitação física, uma vez que permite a recolha e tratamento de dados essenciais para um melhor funcionamento do CRF.

Esta nova ferramenta vai não só trazer vantagens a nível de gestão de stock como também vai deixar um registo de todos os serviços prestados para ajudar no gerenciamento do centro de reabilitação.

Desta forma, espera-se que com esta nova implementação, o negócio se torne mais eficiente e organizado continuando a deixar os seus clientes satisfeitos.

1.5. Estrutura do Relatório

Neste ponto, faremos uma breve exposição daquilo que iremos incluir adiante.

A conceção da base de dados segue o método lecionado nesta unidade curricular, portanto o capítulo seguinte inicia a modelação através do levantamento dos requisitos. Nos capítulos seguintes, procede-se à construção dos modelos conceptual, lógico e físico da base de dados. Finalizamos o relatório com um povoamento simples da base de dados.

2. Levantamento e análise de requisitos

Neste tópico, temos como objetivo analisar os requisitos necessários e fundamentais para a elaboração do projeto.

2.1. Método de levantamento e de análise de requisitos adotado

Após a obtenção de informação sobre o funcionamento geral de um centro de reabilitação, através da realização de discussões com utilizadores deste tipo de estabelecimentos e da observação direta à forma como um centro de reabilitação opera, visto que vários membros do grupo já tiveram lesões que necessitassem de um tratamento neste tipo de estabelecimentos, foi criada uma lista de requisitos que a base de dados deverá satisfazer.

2.2. Requisitos levantados

2.2.1. Requisitos de descrição

Os requisitos de descrição são os seguintes:

1. Os clientes só podem aparecer com devida marcação.
2. As marcações são feitas a partir do e-mail ou por chamada telefónica.
3. Sempre que um cliente realiza um novo tratamento no centro de reabilitação é-lhe associada uma ficha.
4. A ficha do utente contém as seguintes informações:
 - Nome
 - Idade
 - Data de nascimento
 - Número de telemóvel
 - Género
 - Número do utente
 - Índice de prática desportiva
 - Histórico de lesões ou problemas de saúde
 - Tratamento receitado pelo médico

- Relatório de progresso
5. O índice de prática desportiva varia numa escala de 0-10.
 6. O cliente deve consultar o médico do centro antes de começar o tratamento com o terapeuta, de forma a que o médico o oriente para um determinado tratamento.
 7. O médico pode receitar a realização de exames, caso haja dúvidas quanto à lesão do utente.
 8. O centro de reabilitação engloba funcionários como fisioterapeutas, preparadores físicos, médicos e rececionistas.
 9. O preparador físico estará mais ligado a lesões leves e de carácter desportivo.
 10. Depois do relatório dos exames, o médico deverá seleccionar, entre um fisioterapeuta e um preparador físico, qual dos dois se adequa à lesão em questão.
 11. Cada cliente está associado a um fisioterapeuta.
 12. Cada lesão de um determinado cliente tem uma descrição e um tipo associado.
 13. Os clientes que são atletas terão um tratamento mais específico(, de acordo com a modalidade.)
 14. Para além da parte administrativa, a rececionista tratará da parte da limpeza e desinfeção, assim como da abertura e fecho do centro.
 15. Cada falta dada pelo cliente é apontada na base de dados, para ter em conta o tempo até ao término do tratamento.
 16. Os fisioterapeutas, para além de tratar de lesões, podem ainda realizar um treino de mobilidade física com o cliente.
 17. Apenas os fisioterapeutas e preparadores físicos têm acesso às máquinas.
 18. A rececionista tem a capacidade de decidir a quantidade necessária de cada material.
 19. A rececionista faz a coordenação entre a disponibilidade do médico ou fisioterapeuta/preparador físico e do cliente para proceder à marcação de uma consulta ou sessão de tratamento, respetivamente.
 20. Caso o fisioterapeuta verifique que alguma máquina esteja avariada deve comunicar à rececionista, que por sua vez regista na base de dados, para posteriormente proceder à substituição ou reparação da mesma.

2.2.2. Requisitos de exploração

Deverá ser possível obter:

1. O histórico de lesões de um determinado utente, bem como o pagamento total efetuado no seu tratamento.
2. Um relatório do stock existente para saber quais os materiais em falta.
3. Uma lista das máquinas não funcionais.
4. Uma listagem dos utentes que frequentaram o centro de reabilitação num intervalo de tempo específico.
5. Um histórico de utentes que tiveram uma lesão de um determinado tipo.
6. A lista de tratamentos de um determinado terapeuta

2.2.3. Requisitos de controlo

Os requisitos de controlo são os seguintes:

1. O centro de reabilitação física opera de segunda-feira a sexta-feira das 9h às 20h.
2. O centro de reabilitação física recebe clientes de segunda a sexta das 9h às 18h.
3. Das 18h às 20h, o centro apenas funciona para trabalho administrativo ou para limpeza e desinfeção.
4. A rececionista é a única entidade que pode manipular a base de dados.
5. O gerente pode consultar a base de dados.

2.3. Análise e validação geral dos requisitos

Previamente, foram levantados os requisitos para a base de dados do centro de reabilitação, que definem a sua estrutura, como serão explorados e acedidos. Estes requisitos foram confirmados pelos futuros utilizadores da base de dados, pois, de uma forma geral, compõem o bom funcionamento da base de dados que será implementada.

3. Modelação Conceptual

Após o levantamento de requisitos, seguiu-se para a modelação conceptual.

3.1. Apresentação da abordagem de modelação realizada

Para o desenho do modelo conceptual, o grupo decidiu fazer a sua representação através dos diagramas E-R (Entidade-Relacionamento), que é composta por entidades, relacionamentos e atributos. Sintetizando, as entidades são representativas de uma classe de objetos sobre os quais se quer guardar a informação. Os atributos representam a descrição das entidades. Os relacionamentos representam as associações relevantes entre entidades. Para modelarmos conceptualmente o caso de estudo em causa, produzimos uma análise geral dos

requisitos que permitirá a identificação das principais entidades envolvidas no sistema e respetivos relacionamentos.

De seguida, mediante os requisitos de descrição, poderemos reconhecer os atributos associados a cada entidade e relacionamento e, posteriormente, desenvolver o Diagrama ER.

Para o problema em questão, procedeu-se à criação de várias entidades tais como 'Utente', 'Rececionista', 'Médico', 'Terapeuta', 'Máquina' e 'Material'. A cada entidade teremos ligados vários atributos que definem estas, como por exemplo, a cada utente temos associado um 'Nome', 'Idade', 'Género', 'Índice de prática desportiva', etc. Um exemplo de relacionamento é um 'utiliza' entre 'Terapeuta' e 'Máquina', visto que apenas os terapeutas têm acesso autorizado às máquinas, como já referido nos requisitos anteriormente. Existem ainda as chaves primárias, que são atributos que caracterizam cada entidade unicamente, isto é, permitem a identificação única de cada registo dessa entidade possibilitando assim uma procura mais fácil e a garantia de que não há registos repetidos na base de dados.

3.2. Identificação e caracterização das entidades

A partir da análise de requisitos foram identificadas 6 entidades, que se caracterizam na tabela abaixo.

Entidade	Descrição	Ocorrência
Utente	Pessoa que pretende tratar a sua lesão, tendo por isso que proceder a uma marcação de consulta médica.	Um utente procede à marcação de uma consulta médica.
Médico	Elemento que receita um tratamento ao utente.	Um medico receita um tratamento para um utente.
Terapeuta	Funcionário que, com base no tratamento receitado pelo médico, trata o utente.	Um fisioterapeuta trata um utente conforme o tratamento fornecido pelo médico
Contacto	Conjunto de meios de comunicação disponíveis.	Um contacto é usado para transmitir informação a um cliente sobre a marcação de uma consulta.
Material	Elemento utilizado pelo terapeuta que auxilia o tratamento do utente.	O preparador físico recorre a uma bola medicinal.
Máquina	Equipamento elétrico utilizado pelo terapeuta que auxilia o tratamento do utente.	A fisioterapeuta recorre ao eletroestimulador.

Tabela 1: Caracterização das entidades.

3.3. Identificação e caracterização dos relacionamentos

Definimos da seguinte forma os relacionamentos entre as entidades:

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
Utente	1	é tratado por	1	Médico
Utente	1	é tratado por	0..1	Terapeuta
Utente	1	Tem	1	Contacto
Médico	1	trata	N	Utente
Médico	1	fornece tratamento	N	Terapeuta
Médico	1	Tem	1	Contacto
Terapeuta	1	recebe tratamento	1	Médico
Terapeuta	1	trata	N	Utente
Terapeuta	1	utilizam	N	Material
Terapeuta	1	utilizam	N	Máquina
Terapeuta	1	Tem	1	Contacto
Contacto	1	é utilizado por	1	Médico
Contacto	1	é utilizado por	1	Utente
Contacto	1	é utilizado por	1	Terapeuta
Material	N	é usado por	N	Terapeuta
Máquina	N	é usado por	N	Terapeuta

Tabela 2: Caracterização dos relacionamentos.

3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

Depois de identificadas e caracterizadas as entidades e os respetivos relacionamentos, tendo sempre em conta os requisitos estabelecidos anteriormente, passamos à atribuição de atributos às entidades e relacionamentos.

Entidade	Atributo	Descrição	Chave Primária	Tipo e tamanho	Pode ser nulo
Utente	Nome	Nome do utente.	Não	VARCHAR(45)	Não
	Identificador	Identificador do utente.	Sim	Int	Não
	Idade	Idade do utente, calculada a partir da data de nascimento.	Não	Int	Não
	Índice de prática desportiva	Valor entre 0 e 10 que avalia a prática desportiva efetuada.	Não	int	Não
	Data de Nascimento	Data de Nascimento do utente.	Não	Date	Não
	Género	Género do utente (M, F, O).	Não	CHAR	Não

Tabela 3: Caracterização da associação dos atributos com a entidade Utente.

Entidade	Atributo	Descrição	Chave Primária	Tipo	Pode ser nulo
Médico	Identificador	Identificador do médico.	Sim	Int	Não
	Nome	Nome do médico.	Não	VARCHAR(45)	Não

Tabela 4: Caracterização da associação dos atributos com a entidade Médico.

Entidade	Atributo	Descrição	Chave Primária	Tipo	Pode ser nulo
Terapeuta	Identificador	Identificador do terapeuta.	Sim	Int	Não
	Nome	Nome do terapeuta.	Não	VARCHAR(45)	Não
	Função	Função do terapeuta: fisioterapeuta ou preparador físico.	Não	VARCHAR(15)	Não

Tabela 5: Caracterização da associação dos atributos com a entidade Terapeuta.

Entidade	Atributo		Descrição	Chave Primária	Tipo	Pode ser nulo
Contacto	Morada	Rua	Nome da rua da entidade.	Não	VARCHAR(45)	Não
		Localidade	Localidade da entidade.	Não	VARCHAR(45)	Não
		Código Postal	Código Postal da entidade.	Não	VARCHAR(8)	Não
	Número telemóvel		Número de telemóvel da entidade.	Sim	VARCHAR(9)	Não
	E-mail		Endereço de correio eletrónico da entidade.	Não	VARCHAR(45)	Sim

Tabela 6: Caracterização da associação dos atributos com a entidade Contacto.

Entidade	Atributo	Descrição	Chave Primária	Tipo	Pode ser nulo
Máquina	Identificador	Identificador da máquina	Sim	Int	Não
	Nome	Nome da máquina	Não	VARCHAR(45)	Não
	Funcional	Atributo que verifica se a máquina está funcional.	Não	TINYINT	Não

Tabela 7: Caracterização da associação dos atributos com a entidade Máquina.

Entidade	Atributo	Descrição	Chave Primária	Tipo	Pode ser nulo
Material	Nome	Nome do material.	Sim	VARCHAR(45)	Não
	Quantidade	Quantidade do material existente.	Não	Int	Não
	Stockmin	Quantidade mínima obrigatória em stock.	Não	Int	Não

Tabela 8: Caracterização da associação dos atributos com a entidade Material.

Entidade	Atributo		Descrição	Chave Primária	Tipo	Pode ser nulo
Tratamento	Id		Identificador do tratamento.	Sim	Int	Não
	Data		Data do início do tratamento.	Não	DATE	Não
	Preco		Preco do pacote de sessões de tratamento.	Não	Float	Não
	Lesão	Tipo	Avalia a gravidade da lesão.	Não	VARCHAR(15)	Não
		Descrição	Detalhes da lesão.	Não	VARCHAR(45)	Sim

Tabela 9: Caracterização da associação dos atributos com o relacionamento Tratamento.

3.5. Detalhe ou generalização de entidades

Este subcapítulo tem como finalidade generalizar (se possível) ao máximo as entidades existentes. Por outras palavras, podemos criar uma entidade que mostre o que estas têm em comum, desde que o número de semelhanças se justifique.

É possível observar que as entidades 'Terapeuta', 'Médico' e 'Utente' têm atributos em comum e se podem juntar numa entidade 'Pessoa', pois todas estas contêm um atributo 'Identificador', um relacionamento com a entidade 'Contacto' e um atributo 'Nome'. Desta forma, consideramos que estes detalhes se revelam suficientes para generalizar numa nova entidade 'Pessoa'.

Para além disso, podemos fazer uma generalização envolvendo a entidade 'Terapeuta' e a entidade 'Médico', generalizando-as numa entidade 'Profissional', visto que ambas as entidades caracterizam profissionais de saúde. Sendo assim, a nova entidade 'Profissional' teria um atributo 'Cargo' que serviria para distinguir as duas entidades envolvidas.

Para responder a esta query, teremos de usar a entidade 'Utente' com os atributos 'Identificador' e 'Nome' e, a partir daí conseguimos saber os tratamentos e, consequentemente, as lesões de um determinado utente. Para saber o pagamento efetuado em cada consulta necessitamos da entidade 'Tratamento', a qual tem associado o atributo 'Preco'.

- **Obter um relatório do stock existente para saber quais os materiais em falta.**

Para esta query, é necessário a entidade 'Material' e os seus atributos 'stockmin' e 'quantidade', que nos permitirão saber quais os materiais em falta, quando o relatório de materiais existentes é menor que o stock minimo necessário.

- **Obter uma lista das máquinas não funcionais.**

Conseguimos responder a esta query a partir da entidade 'Máquina' com o atributo booleano 'Funcional'. Dessa forma, apenas teremos que selecionar quais as máquinas que apresentam esse atributo com o valor a 0.

- **Obter uma listagem dos utentes que frequentaram o centro de reabilitação num intervalo de tempo específico.**

Para responder a esta query será útil utilizar a entidade 'Utente' e os seus atributos 'Identificador' e 'Nome'. Para além disso, através da entidade 'Tratamento' que contém a data em que o tratamento foi iniciado, será necessário recorrer a uma junção das duas tabelas para que a associação entre o identificador e o nome do utente ocorra corretamente.

- **Obter um histórico de utentes que tiveram uma lesão de um determinado tipo.**

Para obter um histórico de utentes que tiveram uma lesão de um determinado tipo será proveitoso utilizar a entidade 'Tratamento' para saber o id do respetivo tratamento correspondente ao tipo de lesão. Precisaremos também da entidade 'Utente' para saber quais os utentes associados a esta lesão. Para identificar quais os utentes requereremos os atributos 'Identificador' e 'Nome'. É de reparar que por "tipo de lesão" nos estamos a referir à sua gravidade apenas.

- **Obter a lista de tratamentos de um determinado terapeuta.**

Para esta query teremos de utilizar a entidade 'Tratamento' e a sua respetiva chave primária e, através desta, conseguimos saber os terapeutas que estão associados a este tratamento. Isto não será possível ser verificado sem selecionar primeiramente na tabela do 'Terapeuta' o identificador associado ao nome do mesmo, pois só com esse id será possível aceder diretamente à tabela 'Tratamento'.

4. Modelação Lógica

Nesta secção apresentaremos como foi construído e validado o modelo lógico construído pelo grupo.

4.1. Construção e validação do modelo de dados lógico

Depois de desenhado o modelo conceptual, passamos à elaboração do modelo lógico.

4.1.1. Entidades

Uma vez que o atributo 'Morada' é composto, quando implementado no modelo lógico, este atributo é omitido e apenas aparecem os atributos que fazem parte desta composição, neste caso, 'Localidade', 'Código-Postal' e 'Rua'. As chaves primárias e estrangeiras serão explicadas mais abaixo, quando falarmos nos relacionamentos entre entidades.

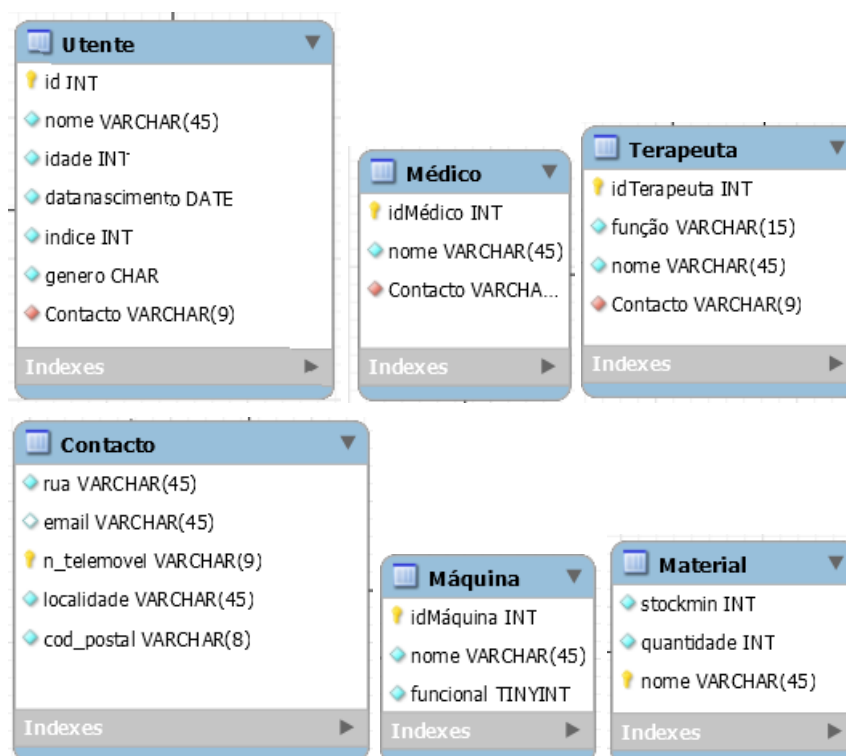


Figura 2: Entidades.

4.1.2. Relação entre Entidades

A entidade 'Contacto' tem 3 relações de 1:1, com 'Terapeuta', 'Médico' e 'Utente'. Como é uma relação de 1:1 não é preciso criar tabelas para estas relações, mas tem de haver uma maneira de identificar um das outra entidades através da entidade 'Contacto', para isso criou-se 3 chaves estrangeiras, 'Terapeuta_idTerapeuta', 'Utente_id' e 'Médico_idMédico'.

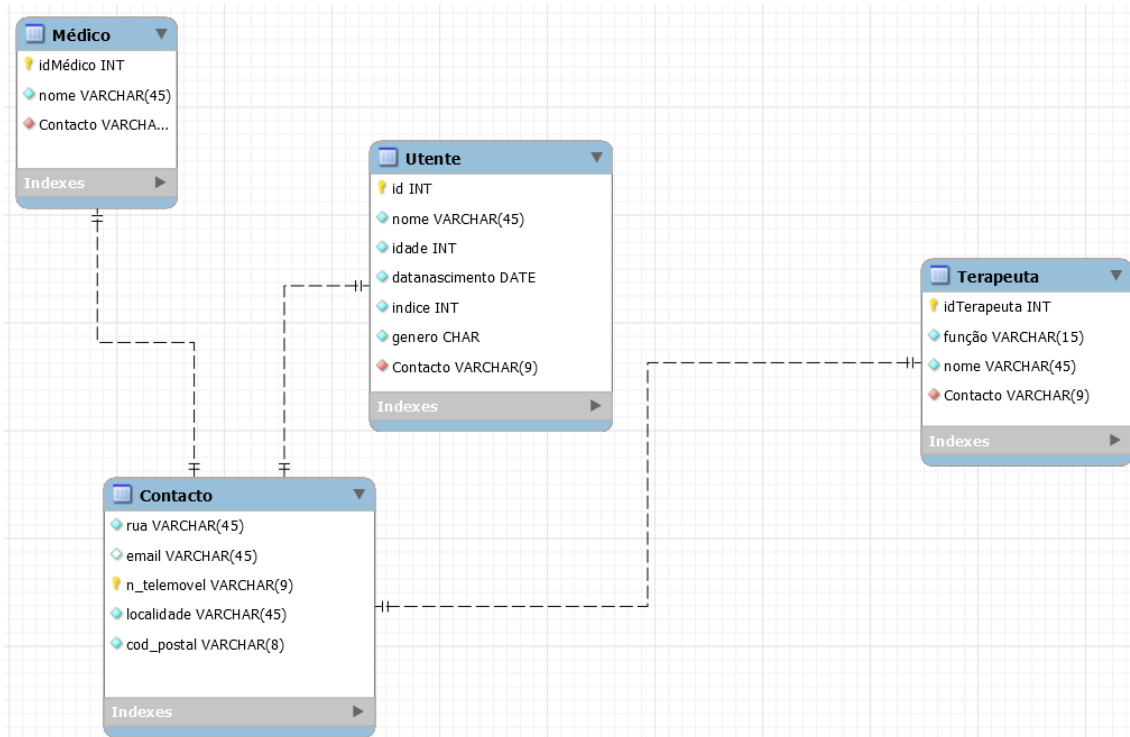


Figura 3: Relacionamento com a entidade 'Contacto'.

Como a relação entre 'Terapeuta' e 'Máquinas' é de N:M, fez-se uma nova tabela com a designação de 'Terapeuta_has_Máquinas'. Nesta estão presentes os identificadores da tabela 'Terapeuta' e da tabela 'Máquina', que são o 'Terapeuta_idTerapeuta' e 'Máquinas_idMáquinas', respetivamente, e cada um deles é uma chave estrangeira. O mesmo acontece para a relação entre 'Terapeuta' e 'Material', só que neste caso a nova tabela terá a designação de 'Terapeuta_has_Material' e os nomes dos identificadores passam a ser 'Terapeuta_idTerapeuta' e 'Material_idMaterial'.

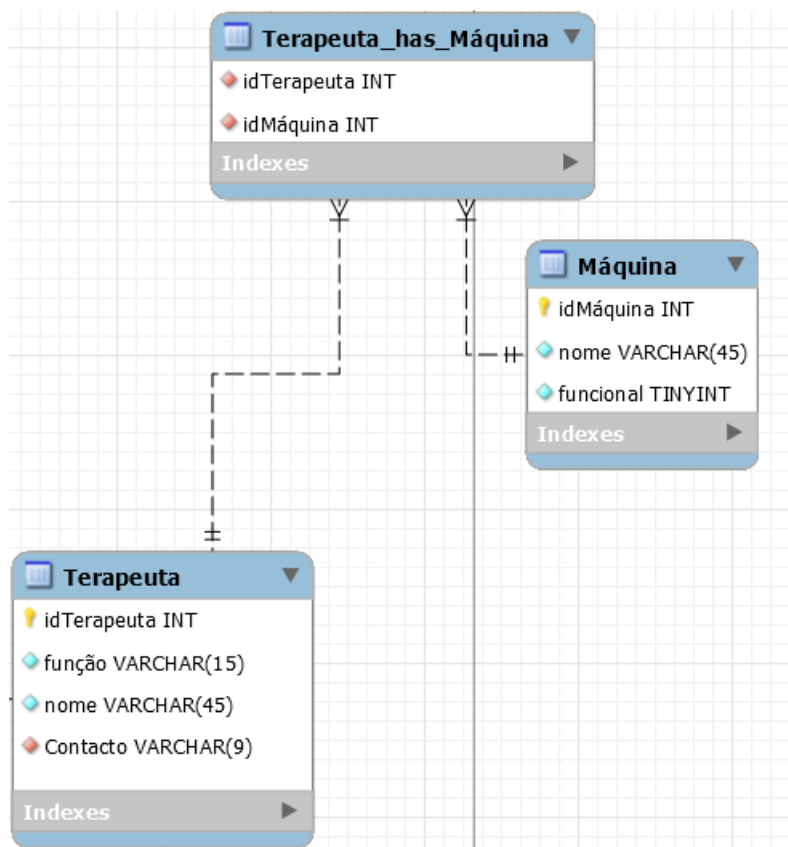


Figura 4: Relacionamento entre 'Terapeuta' e 'Máquina'.

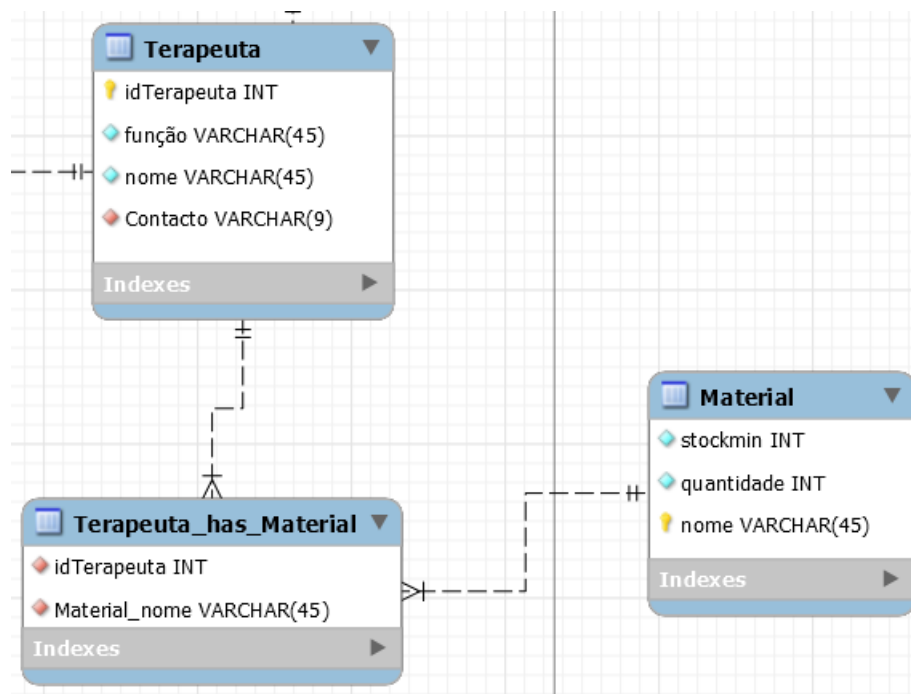


Figura 5: Relacionamento entre 'Terapeuta' e 'Material'.

A relação apresentada a seguir é uma relação tripla, denominada 'Tratamento', entre as entidades 'Médico', 'Utente' e 'Terapeuta'. Para esta relação é, também, preciso criar uma tabela que vai precisar de uma chave principal para diferenciar os vários tratamentos, que neste caso é o atributo 'Identificador'. De maneira a referenciar as outras três entidades vamos também precisar de 3 chaves estrangeiras, uma para cada entidade da relação, que são: "Médico_idMédico", 'Utente_id' e 'Terapeuta_idTerapeuta'. Para além das chaves estrangeiras, também se adicionaram mais 4 atributos à tabela de 'Tratamento': 'Data', 'Tipo', 'Descrição' e 'preço'.

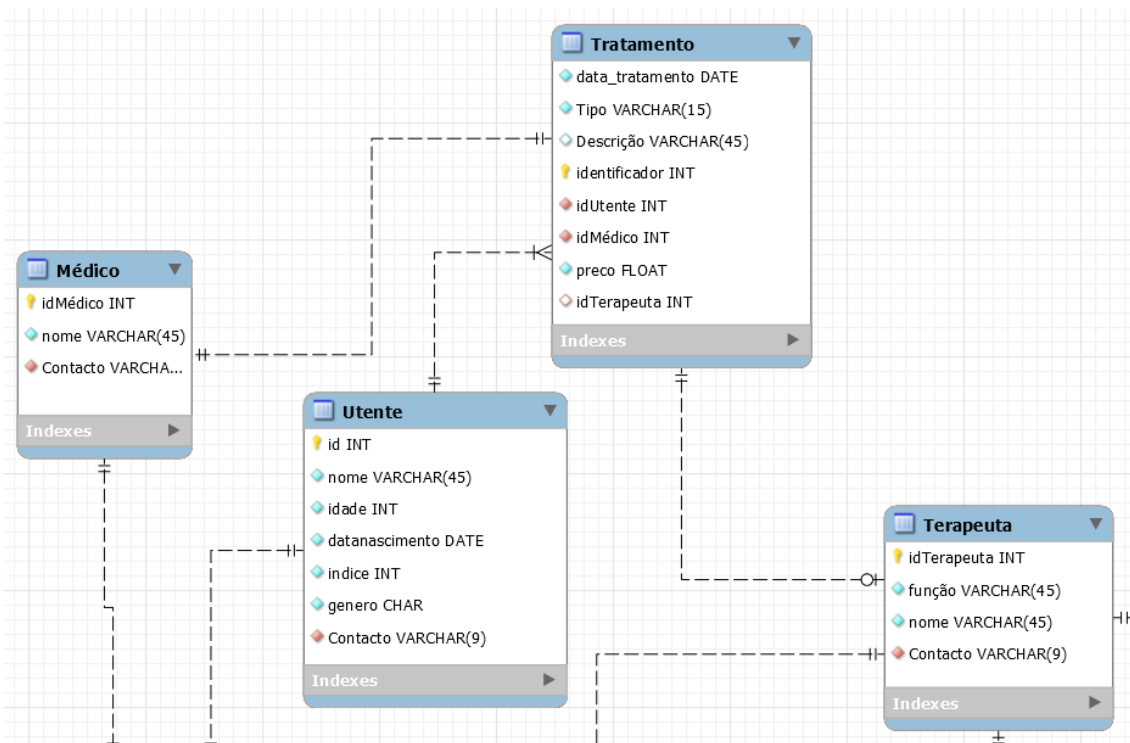


Figura 6: Relacionamento ternário de 'Tratamento'.

4.2. Desenho do modelo lógico

De seguida é apresentado o modelo lógico desenvolvido, que foi concebido com o auxílio da ferramenta MySQLworkbench.

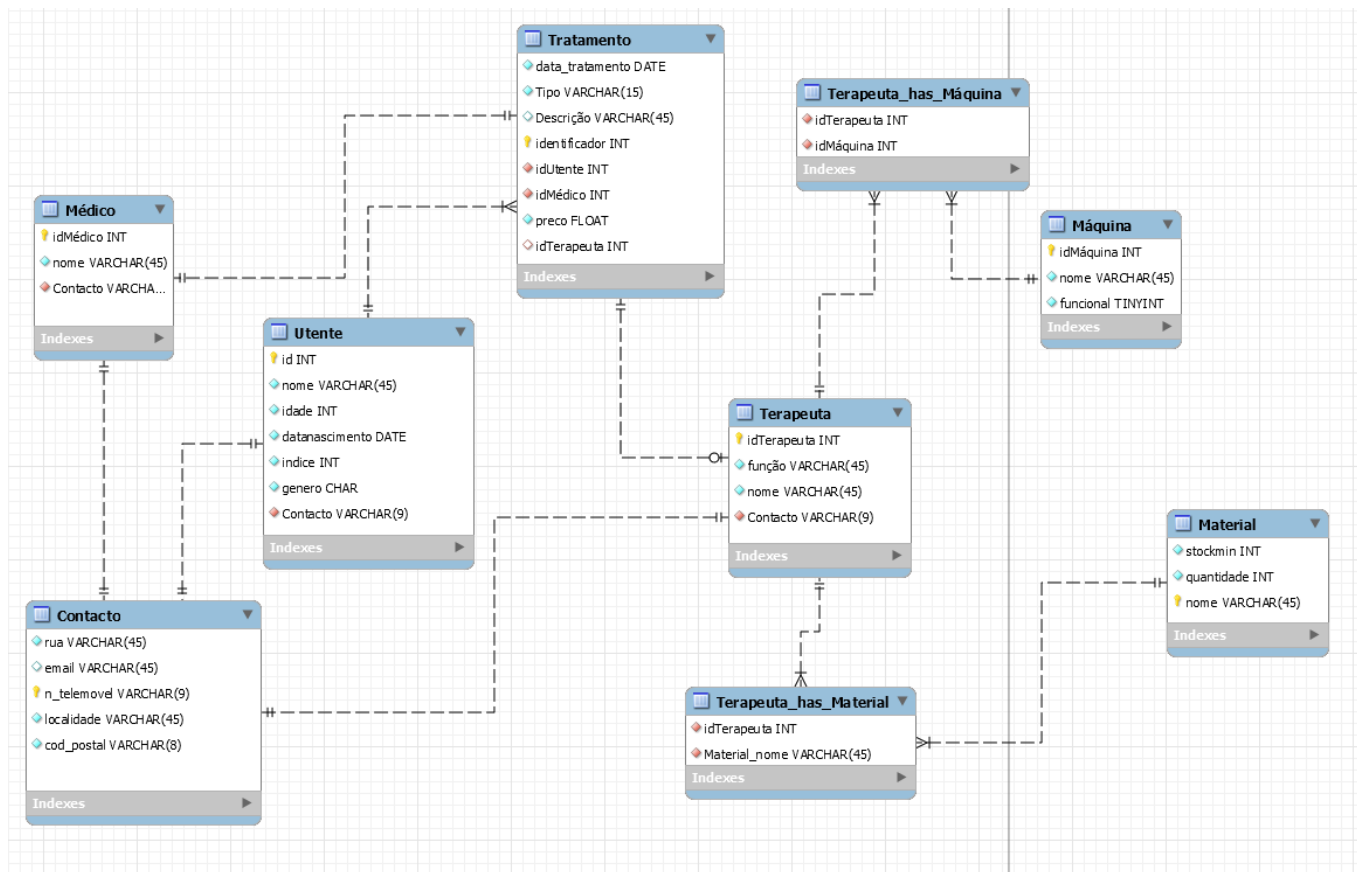


Figura 7: Desenho do modelo lógico do CRF.

4.3. Validação do modelo através da normalização

Nesta secção iremos validar o modelo lógico através da normalização, com a intenção de reduzir a redundância de dados e garantir a sua integridade.

O processo de normalização pretende analisar as dependências funcionais presentes nas tabelas que constituem o modelo de base de dados, diminuindo progressivamente a possibilidade de ocorrência de erros de consistência durante a atualização ou alteração da informação contida num sistema de base de dados. Este processo compreende o uso de um conjunto de regras, chamados de formas normais. Caso o sistema de base de dados respeite as primeiras três regras, então ele está na terceira forma normal (3FN). Contudo, a terceira forma normal é considerada o nível mínimo necessário para grande parte das aplicações.

- **Primeira Forma Normal (1FN)**

Para que uma tabela esteja na primeira forma normal deve cumprir a seguinte regra: a interseção de cada linha e coluna poderá conter apenas um valor. Para não

existirem atributos repetidos procedeu-se à criação de uma entidade designada por 'Contacto', que tem atributos como 'morada', 'número telemóvel' e 'e-mail'. Desta forma, todas as tabelas cumprem a primeira forma normal.

- **Segunda forma normal (2FN)**

A segunda forma normal diz-nos que deve cumprir a 1FN e todas as chaves não primárias devem ser funcionalmente dependentes da chave primária por completo, isto significa que todas as tabelas têm uma chave primária com apenas um atributo. Assim sendo concluímos que as tabelas respeitam também a 2FN, visto que qualquer atributo de uma relação no modelo lógico depende apenas do atributo chave da relação correspondente, e nenhuma das relações existentes contém dependências parciais.

Consequentemente, ao ser validada a primeira forma normal e verificada também a segunda, as tabelas cumprem a segunda forma normal.

- **Terceira forma normal (3FN)**

Após verificar a 2FN, verificaremos se as tabelas estão na 3FN que diz que estas não devem ter chaves não primárias, cujos atributos sejam transitivamente dependentes da chave primária da tabela. Sendo que isto se confirma em todas as tabelas, conclui-se que não existem dependências transitivas e, por isso, o modelo lógico encontra-se na terceira forma normal.

4.4. Validação do modelo com interrogações do utilizador

Verifiquemos agora, analisando o modelo lógico efetuado, se este responde às interrogações do utilizador de modo a cumprir os requisitos de exploração definidos para a base de dados.

1ª Query: O histórico de lesões de um determinado utente, bem como o pagamento total efetuado no seu tratamento.

$\Pi_{data_tratamento, tipo, descrição, preço} (\sigma_{idUtente = (\Pi_{id} (\sigma_{nome = cliente} (utente)))} (tratamento))$

2ª Query: Um relatório do stock existente para saber quais os materiais em falta.

$\pi_{nome} (\sigma_{quantidade < stockmin} (material))$

3ª Query: Uma lista das máquinas não funcionais.

$\pi_{idMáquina} (\sigma_{funcional = 0} (máquina))$

4ª Query: Uma listagem dos utentes que frequentaram o centro de reabilitação num intervalo de tempo específico.

$\pi_{t.idUtente, u.nome} (utente \bowtie_{t.idUtente = u.id} (\sigma_{month(t.data_tratamento) \leq max \wedge month(t.data_tratamento) \geq min} (tratamento)))$

5ª Query: Um histórico de utentes que tiveram uma lesão de um determinado tipo.

$\pi_{u.nome} (utente \bowtie_{t.idUtente = u.id} (\sigma_{t.tipo = tipo} (tratamento)))$

6ª Query: A lista de tratamentos de um determinado terapeuta.

$\pi_{u.nome, t.data_tratamento, t.tipo, t.descrição} (utente \bowtie_{t.idUtente = u.id} (\sigma_{idTerapeuta = tidTerapeuta} (\sigma_{nome = terapeutaNome} (terapeuta))))$

4.5. Revisão do modelo lógico produzido

Após uma análise junto do utilizador conclui-se que o modelo lógico desenvolvido pelo grupo aborda de forma correta os conceitos requeridos para o mesmo.

5. Implementação Física

5.1. Seleção do sistema de gestão de bases de dados

Tendo em mente aquilo que nos foi lecionado ao longo deste semestre, a escolha da ferramenta de gestão de base de dados revelou-se fácil, visto que o MySQL é aquela com a qual temos mais confiança devido ao facto de ser o programa utilizado nas aulas da UC de Base de Dados, e como consequência disso foi a ferramenta que decidimos adotar com o objetivo de realizarmos o esquema físico com a qualidade e eficiência pretendidas. Outros fatores que nos levou a escolher o MySQL foi o facto de se tratar de um sistema gratuito, popular, de fácil manuseamento

e de alto grau de desempenho e segurança de dados. Assim esperamos obter um produto final consistente e que cumpra todos os objetivos definidos anteriormente.

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

Usando o MySQL como sistema de gestão, começamos por criar um script que irá ser a base do nosso sistema.

O modelo físico obtêm-se através da opção Forward Engineer no separador Database.

```
39 • CREATE TABLE IF NOT EXISTS `crf`.`Contacto` (  
40     `rua` VARCHAR(45) NOT NULL,  
41     `email` VARCHAR(45) NULL,  
42     `n_telemovel` VARCHAR(9) NOT NULL,  
43     `localidade` VARCHAR(45) NOT NULL,  
44     `cod_postal` VARCHAR(8) NOT NULL,  
45     PRIMARY KEY (`n_telemovel`))  
46     ENGINE = InnoDB;
```

Figura 8: Criação da tabela 'Contacto'.

```
-- Table `crf`.`Utente`  
-----  
• CREATE TABLE IF NOT EXISTS `crf`.`Utente` (  
    `id` INT NOT NULL,  
    `nome` VARCHAR(45) NOT NULL,  
    `idade` INT NOT NULL,  
    `datanascimento` DATE NOT NULL,  
    `indice` INT NOT NULL,  
    `genero` CHAR NOT NULL,  
    `Contacto` VARCHAR(9) NOT NULL,  
    PRIMARY KEY (`id`),  
    INDEX `fk_Utente_Contacto1_idx` (`Contacto` ASC) VISIBLE,  
    CONSTRAINT `fk_Utente_Contacto1`  
        FOREIGN KEY (`Contacto`)  
        REFERENCES `crf`.`Contacto` (`n_telemovel`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
    ENGINE = InnoDB;
```

Figura 9: Criação da tabela 'Utente'.

```
-- Table `crf`.`Terapeuta`  
  
• CREATE TABLE IF NOT EXISTS `crf`.`Terapeuta` (  
  `idTerapeuta` INT NOT NULL,  
  `função` VARCHAR(15) NOT NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  `Contacto` VARCHAR(9) NOT NULL,  
  PRIMARY KEY (`idTerapeuta`),  
  INDEX `fk_Terapeuta_Contacto1_idx` (`Contacto` ASC) VISIBLE,  
  CONSTRAINT `fk_Terapeuta_Contacto1`  
    FOREIGN KEY (`Contacto`)  
    REFERENCES `crf`.`Contacto` (`n_telemovel`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Figura 10: Criação da tabela 'Terapeuta'.

```
-- Table `crf`.`Médico`  
  
• CREATE TABLE IF NOT EXISTS `crf`.`Médico` (  
  `idMédico` INT NOT NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  `Contacto` VARCHAR(9) NOT NULL,  
  PRIMARY KEY (`idMédico`),  
  INDEX `fk_Médico_Contacto_idx` (`Contacto` ASC) VISIBLE,  
  CONSTRAINT `fk_Médico_Contacto`  
    FOREIGN KEY (`Contacto`)  
    REFERENCES `crf`.`Contacto` (`n_telemovel`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Figura 11: Criação da tabela 'Médico'.

```

-----
-- Table `crf`.`Material`
-----
• CREATE TABLE IF NOT EXISTS `crf`.`Material` (
  `stockmin` INT NOT NULL,
  `quantidade` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`nome`))
ENGINE = InnoDB;

```

Figura 12: Criação da tabela 'Material'.

```

-----
-- Table `crf`.`Máquina`
-----
• CREATE TABLE IF NOT EXISTS `crf`.`Máquina` (
  `idMáquina` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `funcional` TINYINT NOT NULL,
  PRIMARY KEY (`idMáquina`))
ENGINE = InnoDB;

```

Figura 13: Criação da tabela 'Máquina'.

```

● ○ CREATE TABLE IF NOT EXISTS `crf`.`Tratamento` (
    `data_tratamento` DATE NOT NULL,
    `Tipo` VARCHAR(15) NOT NULL,
    `Descrição` VARCHAR(45) NULL,
    `identificador` INT NOT NULL,
    `idUtente` INT NOT NULL,
    `idMédico` INT NOT NULL,
    `idTerapeuta` INT NULL,
    `preco` FLOAT NOT NULL,
    PRIMARY KEY (`identificador`),
    INDEX `fk_Tratamento_Utentel_idx` (`idUtente` ASC) VISIBLE,
    INDEX `fk_Tratamento_Médicol_idx` (`idMédico` ASC) VISIBLE,
    INDEX `fk_Tratamento_Terapeuta1_idx` (`idTerapeuta` ASC) VISIBLE,
    CONSTRAINT `fk_Tratamento_Utentel`
        FOREIGN KEY (`idUtente`)
        REFERENCES `crf`.`Utente` (`id`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Tratamento_Médicol`
        FOREIGN KEY (`idMédico`)
        REFERENCES `crf`.`Médico` (`idMédico`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Tratamento_Terapeuta1`
        FOREIGN KEY (`idTerapeuta`)
        REFERENCES `crf`.`Terapeuta` (`idTerapeuta`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 14: Criação da tabela 'Tratamento'.

```

-----
-- Table `crf`.`Terapeuta_has_Material`
-----
CREATE TABLE IF NOT EXISTS `crf`.`Terapeuta_has_Material` (
  `idTerapeuta` INT NOT NULL,
  `Material_nome` VARCHAR(45) NOT NULL,
  INDEX `fk_Terapeuta_has_Material_Material1_idx` (`Material_nome` ASC) VISIBLE,
  INDEX `fk_Terapeuta_has_Material_Terapeuta1_idx` (`idTerapeuta` ASC) VISIBLE,
  CONSTRAINT `fk_Terapeuta_has_Material_Terapeuta1`
    FOREIGN KEY (`idTerapeuta`)
      REFERENCES `crf`.`Terapeuta` (`idTerapeuta`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Terapeuta_has_Material_Material1`
    FOREIGN KEY (`Material_nome`)
      REFERENCES `crf`.`Material` (`nome`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 15: Criação da tabela 'Terapeuta_has_Material'.

```

-----
-- Table `crf`.`Terapeuta_has_Máquina`
-----
CREATE TABLE IF NOT EXISTS `crf`.`Terapeuta_has_Máquina` (
  `idTerapeuta` INT NOT NULL,
  `idMáquina` INT NOT NULL,
  INDEX `fk_Terapeuta_has_Máquina_Máquina1_idx` (`idMáquina` ASC) VISIBLE,
  INDEX `fk_Terapeuta_has_Máquina_Terapeuta1_idx` (`idTerapeuta` ASC) VISIBLE,
  CONSTRAINT `fk_Terapeuta_has_Máquina_Terapeuta1`
    FOREIGN KEY (`idTerapeuta`)
      REFERENCES `crf`.`Terapeuta` (`idTerapeuta`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Terapeuta_has_Máquina_Máquina1`
    FOREIGN KEY (`idMáquina`)
      REFERENCES `crf`.`Máquina` (`idMáquina`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 16: Criação da tabela 'Terapeuta_has_Máquina'.

5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)

Todas as interrogações referidas em 4.4 foram traduzidas para SQL. De seguida, apresentamos alguns scripts escritos em SQL com o intuito de responder a interrogações fundamentais para a base de dados criada.

1ª Query:

```
select data_tratamento, tipo, Descrição, preco from tratamento
  where idUtente =
    (select id from utente where nome = "Cristiano Ronaldo dos Santos Aveiro");
```

Figura 17: Query do histórico de lesões de um determinado utente, bem como o pagamento total efetuado no seu tratamento.

2ª Query:

```
select nome from material where quantidade < stockmin;
```

Figura 18: Query de um relatório do stock existente para saber quais os materiais em falta.

3ª Query:

```
select idMáquina, nome from máquina where funcional = 0;
```

Figura 19: Query de uma lista das máquinas não funcionais.

4ª Query:

```
select tratamento.idUtente, utente.nome from tratamento
inner join utente on tratamento.idUtente = utente.id
where month(tratamento.data_tratamento) <= 8 && month(tratamento.data_tratamento) >= 5
group by utente.nome
order by utente.nome asc;
```

Figura 20: Query de uma listagem dos utentes que frequentaram o centro de reabilitação num intervalo de tempo específico.

5ª Query:

```
select utente.nome from tratamento as t
inner join utente on t.idUtente = utente.id
where t.Tipo = "Médio"
group by utente.nome
order by utente.nome asc;
```

Figura 21: Query de um histórico de utentes que tiveram uma lesão de um determinado tipo.

6ª Query:

```
select u.nome, t.data_tratamento, t.tipo, t.Descrição from tratamento as t
inner join utente as u on t.idUtente = u.id
where idTerapeuta =
(select idTerapeuta from terapeuta where nome = "Pedro Miguel Coelho Lopes")
order by data_tratamento asc;
```

Figura 22: Query de uma lista de tratamentos de um determinado terapeuta.

5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)

Através da utilização de índices em SQL é possível aumentar a eficácia da base de dados em geral, facilitando a organização e consulta de uma tabela. Contudo, a definição de índices extra aumenta o tempo despendido pelo sistema ao realizar inserções. Sendo assim, optamos por não colocar índices extra em nenhuma das relações, isto porque a maior parte das operações realizadas no sistema são de inserção de dados por parte da Rececionista.

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

Com o desenvolvimento de uma Base de Dados é necessário ter em conta o espaço em disco que será necessário para armazenar todos os dados produzidos.

Cada registo de informação ocupa espaço físico na memória, dependendo do tipo de dados a que este está associado.

Atributo	Tamanho máximo (Bytes)
Nome	45
Idade	4
Índice de prática desportiva	4
Data de nascimento	10
Género	1

Tabela 10: Tamanho máximo por entrada da tabela 'Utente'.

Atributo	Tamanho máximo (Bytes)
Identificador	4
Nome	45

Tabela 11: Tamanho máximo por entrada da tabela 'Médico'.

Atributo	Tamanho máximo (Bytes)
Identificador	4
Nome	45
Função	15

Tabela 12: Tamanho máximo por entrada da tabela 'Terapeuta'.

Atributo		Tamanho máximo (Bytes)
Morada	Rua	45
	Localidade	45
	Código postal	8
Número de telemóvel		9
E-mail		45

Tabela 13: Tamanho máximo por entrada da tabela 'Contacto'.

Atributo	Tamanho máximo (Bytes)
Identificador	4
Nome	45
Funcional	4

Tabela 14: Tamanho máximo por entrada da tabela 'Máquina'.

Atributo	Tamanho máximo (Bytes)
Nome	45
Quantidade	4
Stock mínimo	4

Tabela 15: Tamanho máximo por entrada da tabela 'Material'.

Atributo		Tamanho máximo (Bytes)
Identificador		4
Data		10
Preço		4
Lesão	Tipo	15
	Descrição	45

Tabela 16: Tamanho máximo por entrada da tabela 'Tratamento'.

5.6. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

O grupo decidiu elaborar três vistas de utilização com as seguintes finalidades:

- Verificar o stock de material existente.

```
-- View Stock
CREATE VIEW stockMateriaisView AS
    SELECT nome,
           quantidade,
           stockmin
    FROM material
    ORDER BY nome ASC;
```

Figura 23: View do Stock existente.

- Apresentar a lista de contactos de todos os utentes.

```
-- View Contactos Utentes
create view contactosUtentes as
    select u.nome, c.n_telemovei, c.email, c.rua, c.localidade, c.cod_postal
    from utente as u
    inner join contacto as c
    on u.contacto = c.n_telemovei;
```

Figura 24: View de Contactos dos Utentes

- Apresentar toda a informação relativa à faturação do CRF.

```
-- View faturacao
create view infoFaturacao as
    select sum(preco) as 'Total Faturado',
           max(preco) as 'Tratamento mais caro',
           avg(preco) as 'Preço médio de tratamento',
           min(preco) as 'Tratamento mais barato',
           count(preco) as 'Número de tratamentos'
    from tratamento;
```

Figura 25: View da informação relativa à faturação

5.7. Revisão do sistema implementado

Após terminada a implementação física, foi apresentado o sistema final de base de dados ao centro de reabilitação, que se mostrou bastante satisfeito com as particularidades do mesmo.

6. Povoamento da Base de Dados

Neste subcapítulo, iremos apresentar o conjunto de inserções que foram feitas às diversas tabelas criadas.

```
-- Povoamento da tabela "Utente"
insert into `crf`.`utente`
(id, nome, idade, datanascimento, indice, genero, contacto)
values
(0, "Joaquim Ricardo da Silva Campos", 55, "1965-01-22", 2, 'M', "950123456"),
(1, "Carla Madalena Sepúlveda Pinto", 23, "1997-04-04", 9, 'F', "945627884"),
(2, "Maria Rita Rodrigues Sousa", 19, "2001-11-08", 7, 'F', "923451226"),
(3, "Cláudio Gomes Rocha", 28, "1992-03-13", 9, 'M', "925554126"),
(4, "Mariana Sousa Machado", 20, "2000-10-10", 10, 'M', "916542585"),
(5, "Cristiano Ronaldo dos Santos Aveiro", 35, "1985-02-05", 10, 'M', "925627410"),
(6, "Jéssica Marques Oliveira", 19, "2000-12-29", 8, 'F', "962513300"),
(7, "Joana Gonçalves Faria", 22, "1998-09-02", 5, 'F', "939633004"),
(8, "António Miguel Lopes Pires", 40, "1980-08-05", 4, 'M', "917475630")
;

insert into Médico
(idMédico, nome, contacto)
values
(0, "Nuno Manuel da Silva Teixeira", "924562411"),
(1, "Ana Filipa Veiga Pereira", "911230455")
;
```

```

insert into terapeuta
(idTerapeuta, função, nome, contacto)
values
(0, "Fisioterapeuta", "Ana Cristina Dias Soares", "923514788"),
(1, "Preparador físico", "Pedro Miguel Coelho Lopes", "93648222"),
(2, "Fisioterapeuta", "Rúben Carvalho Ferreira", "932147854"),
(3, "Preparador físico", "Maria Natália Gomes Campos", "936998329")
;

```

```

insert into material
(stockmin, quantidade, nome)
values
(4, 4, "Bola medicinal 4kg"),
(3, 3, "Bola medicinal 2kg"),
(4, 6, "Discos 2kg"),
(6, 6, "Tapete"),
(2, 1, "Step Aeróbico"),
(2, 0, "Creme de massagem NIVEA")
;

```

```

insert into tratamento
(data_tratamento, tipo, descrição, identificador, idUtente, idMédico, idTerapeuta, preco)
values
("2020-03-18", "Grave", "Tendinite na virilha", 0, 0, 0, 0, 285.50),
("2020-07-06", "Leve", "Deslocação do ombro", 1, 1, 0, 0, 171.80),
("2020-08-14", "Muito Grave", "Fratura da clavícula", 2, 2, 1, 2, 440.22),
("2020-06-12", "Médio", "Distensão abdominal", 3, 3, 1, 1, 220.40),
("2020-01-02", "Muito Leve", "Torção do tornozelo", 4, 4, 0, 3, 189.99),
("2020-05-12", "Muito Grave", "Instabilidade na rótula", 5, 5, 0, 0, 420.00),
("2020-07-03", "Grave", "Distensão muscular na virilha", 6, 6, 1, 3, 360.70),
("2020-09-11", "Muito Grave", "Rotura de ligamentos no joelho", 7, 7, 1, 1, 490.77),
("2020-08-02", "Médio", "Rotura de ligamentos no pulso", 8, 8, 0, 0, 310.40),
("2020-04-18", "Médio", "Distensão no bíceps", 9, 3, 0, 0, 180.99),
("2020-06-09", "Leve", "Falta de mobilidade na anca", 10, 6, 1, 0, 120.55),
("2020-05-30", "Leve", "Distensão no gêmeo", 11, 1, 1, 1, 192.12),
("2020-09-28", "Médio", "Distensão muscular nas costas", 12, 4, 0, 1, 220.10)
;

```

```

insert into máquina
(idMáquina, nome, funcional)
values
(0, "Eletroestimulador", 1),
(1, "Passadeira", 1),
(2, "Passadeira", 1),
(3, "Eletroestimulador", 1),
(4, "Eletroestimulador", 0)
;

```

```

insert into contacto
-- (rua, email, n_telemovel, Médico_idMédico, Terapeuta_idTerapeuta, Utente_id, localidade, cod_postal)
(rua, email, n_telemovel, localidade, cod_postal)
values
("Rua Fernão Sousa, n 42", "jrsc00@gmail.com", "950123456", "Braga", "4720-444"),
("Rua dos Congregados, n 24", "cmcp20@gmail.com", "945627884", "Braga", "4715-424"),
("Rua Amélia do Carmo, n 124", "mrrs22@hotmail.com", "923451226", "Braga", "4715-234"),
("Rua Sá Valente, n 220", "cgr1212@gmail.com", "925554126", "Braga", "4710-253"),
("Rua Mário Fernandes, n 7", "msm.12@gmail.com", "916542585", "Braga", "4715-440"),
("Rua das Pereiras, n 42", "crsa33@hotmail.com", "925627410", "Braga", "4720-111"),
("Rua Carmo Ferreira, n 88", "jmo.o@gmail.com", "962513300", "Braga", "4715-787"),
("Rua Manuel Gomes Faria, n 40", "jgfg99@gmail.com", "939633004", "Braga", "4720-777"),
("Rua Faria Sousa, n 4", "amlp@hotmail.com", "917475630", "Braga", "4715-567"),
("Rua Palmeira Campos, n 200", "nmst@gmail.com", "924562411", "Braga", "4720-890"),
("Rua dos Namorados, n 23", "afvp@gmail.com", "911230455", "Braga", "4710-343"),
("Rua das Camélias, n 55", "acds33@hotmail.com", "923514788", "Braga", "4715-743"),
("Rua dos Congregados, n 20", "pmcl22@gmail.com", "93648222", "Braga", "4720-311"),
("Rua dos Capelistas, n 12", "rcf44@gmail.com", "932147854", "Braga", "4715-987"),
("Rua Senhora a Branca, n 75", "mngc11@gmail.com", "936998329", "Braga", "4730-874")
;

```

```

insert into terapeuta_has_material
(idTerapeuta, Material_nome)
values
(0, "Bola medicinal 2kg"),
(0, "Tapete"),
(1, "Bola medicinal 4kg"),
(3, "Step Aeróbico"),
(2, "Discos 2kg"),
(1, "Discos 2kg"),
(0, "Tapete")
;

```



```
insert into terapeuta_has_máquina
(idTerapeuta, idMáquina)
values
(0, 0),
(1, 1),
(1, 2),
(2, 4),
(3, 3),
(3, 1)
;
```

7. Conclusões e Trabalho Futuro

O desafio proposto pelo grupo foi modelar a estrutura básica de uma base de dados para gestão de um centro de reabilitação física. Os sistemas de base de dados requerem um esforço muitíssimo menor para a organização e manipulação dos dados, tendo também um índice de eficácia elevadíssimo comparativamente a métodos tradicionais (suporte em papel), o que torna este meio de organização, um meio cada vez mais cobiçado pelo mundo do negócio.

Finalizado o trabalho proposto, concluímos que a implementação de uma base de dados requer diversas fases consecutivas para que a sua criação seja coerente e garanta a satisfação do utilizador.

Para que uma base de dados seja estável, bem estruturada, eficaz, e ao mesmo tempo fácil de aceder e manipular, esta requer um atento trabalho de pré-construção, em que o "construtor" da base de dados terá de ter em conta todas as exigências do seu cliente. Sendo assim, todo o processo deverá seguir todos os passos necessários até que a base de dados esteja pronta, começando na contextualização do trabalho em causa, seguido de uma boa interpretação, passando para uma construção sólida e organizada dos modelos necessários à construção desta (conceptual, lógico, físico).

É extremamente difícil concluir um trabalho deste tipo na perfeição, uma vez que qualquer base de dados pode sempre ser melhorada seja por meio de

aperfeiçoamento da sua estrutura, seja por modificação na eficiência das "queries". Para além disso, existem várias maneiras de abordar uma determinada situação, sendo raros os casos em que exista uma solução universal e perfeitamente correta.

Sendo que o sistema ainda apresenta algumas lacunas, foram identificadas as seguintes propostas a implementar futuramente:

- Adicionar mais entidades (estagiários, funcionários de limpeza, etc.);
- Registar todas as sessões de tratamento/consultas médicas;
- Abrangir a base de dados de forma a criar um *franchise* de centros de reabilitação física.

De um modo geral e após as várias etapas e validação efetuada em cada um dos modelos, obteve-se a confirmação de que o sistema foi implementado com sucesso, capaz de responder adequadamente aos requisitos identificados e garantindo a integridade de dados.

Referências

[1] Connolly, T., Begg C. (2004), *Database Systems, A Pratical Approach to Design, Implementation, and Management*, Addison-Wesley, 4ªa Edição

[2] Machado, D. [Diego] (2015), *Normalização em Bancos de Dados*.

[3] Miranda, W. [William], *Normalização de Dados*

Lista de Siglas e Acrónimos

BD Base de Dados

CRF Centro de Reabilitação Física

ER Entidade-Relacionamento