



O agendamento e a realização de exames imagiológicos de diferentes modalidades e tipos num determinado hospital

Docentes: José Manuel Ferreira Machado

Daniela Sofia Rijo Oliveira

Unidade Curricular: Base de Dados para a Bioinformática

Mestrado em Bioinformática

Trabalho realizado por:

Alexandra Coelho (PG45458)

Andreia Gomes (PG45463)

Catarina Ferreira (PG45467)

Daniela Lemos (PG45469)

Ano Letivo 2021/2022

7 de janeiro de 2022

Índice

Contextualização	4
Motivos e objetivos.....	4
Validação do projeto	5
Requisitos do projeto.....	6
Caraterização dos diversos perfis	7
Exames Imagiológicos.....	8
Validação do modelo conceptual.....	10
Identificação das Entidades do Modelo Conceptual.....	10
Identificação das Relacionamentos.....	11
Identificar e associar atributos com entidades e tipos de relacionamentos.....	14
Identificação dos tipos dos atributos	16
Determinar chaves candidatas, primárias e alternativas	18
Verificar redundância do modelo	20
Modelo lógico.....	20
Validação do modelo lógico.....	20
Derivar relações do modelo conceptual	21
Entidades Fortes	21
Entidades Fracas	23
Atributos Multivalor	24
Normalização	24
1ª Forma Normal	24
2ª Forma Normal	24
3ª Forma Normal	25
Validação do modelo físico.....	25
Tradução do modelo lógico para o modelo físico.....	25

Análise de transação/relação	29
Povoamento das tabelas do modelo físico	32
Execução de processos associados ao SBD.....	36
Triggers.....	36
Funções.....	36
Procedures	37
Views	38
Interfaces.....	38
Interface dos utentes: Front-end.....	40
Interface gestores: Back-end.....	42
Conclusão.....	46
Referências.....	47

Contextualização

A enorme expansão tecnológica despoletou uma maior procura na área da inovação informática por parte das empresas gerando, desta forma, uma maior competitividade entre as mesmas.

Desta forma, surge o nosso trabalho relativo a uma clínica privada, cuja não possui uma base de dados adequada para a marcação de exames imagiológicos. Esta base de dados que será criada é de extrema necessidade, uma vez que facilitará o processo de marcação e agilizará todo o processo de organização e melhor cumprimento de horários estabelecidos. Tendo em consideração as infundáveis listas de espera para a realização de exames no setor público, urge a procura de melhores alternativas como é o caso da nossa clínica. Todavia, por consequência desta grande adesão e afluência fruto da incapacidade de resposta do setor público, estes eventos despoletaram um crescimento exponencial da clínica de exames imagiológicos, tendo-se instalado o caos.

Para tal, a clínica em questão, de modo a melhorar o seu desempenho, decidiu procurar informatizar os dados dos pacientes e os respetivos exames para uma base de dados. Esta iniciativa deve-se ao aumento da procura do estabelecimento, redução do tempo e facilita o acesso de dados, bem como auxilia na gestão e processamento dos dados em causa.

Assim, procede-se ao desenvolvimento de uma base de dados capaz de agregar toda a informação necessária correspondente a cada paciente.

Motivos e objetivos

Este trabalho prático visa a implementação de uma possível base de dados funcional que poderá ser útil numa clínica de exames imagiológicos. Esta base de dados deverá responder a toda as exigências requeridas para um bom funcionamento do agendamento e realização de exames. Deste modo, pretende-se com o auxílio da linguagem SQL e da

modelação, elaborar um sistema viável e eficiente capaz de suportar uma aplicação de gestão de marcações na clínica.

Validação do projeto

De modo a proceder à validação do projeto, recorreu-se à metodologia presente nos capítulos 15, 16 e 17 do livro - Connolly, T., Begg, C., Database Systems, com as devidas adaptações à nossa situação.

Tendo em consideração o cerne do problema relativo ao armazenamento e gerenciamento dos dados de uma clínica, onde é necessário fazer o registo, recolha e acesso dos dados de forma eficiente, isto é, com um tempo de resposta razoável por parte do sistema de processamento de dados evitando *bottlenecks*, e acima de tudo fidedigna, preservando a integridade dos dados aquando da sua utilização e alterações.

Para tal, a solução passa pela utilização de um sistema de base de dados SQL, cujo possui uma padronização bastante difundida do modo de interação com um dado sistema SQL. A escolha da utilização de SQL é simples, pois esta linguagem é amplamente utilizada, nomeadamente, para o armazenamento e tratamento de uma vasta quantidade de dados. Sendo que uma das vertentes do SQL que vamos utilizar no presente projeto será o *MySQL* para a modelação lógica, uma vez que apresenta algumas vantagens como o facto de ser *open source*, isto traduz-se numa redução bastante significativa do custo de utilização para o gerenciamento dos dados. Adicionalmente, o *MySQL* apresenta uma priorização na performance, que advém da simplicidade das *features* SQL implementadas. Em oposição, também possui desvantagens como é o caso mais relevante, a ineficiência perante um extenso volume de dados, o que pode apresentar-se como uma desvantagem considerável na aplicação do sistema de base de dados à infraestrutura de dados da clínica.

Relativamente a principais fatores críticos a ter em conta para que o sistema de base de dados a ser implementado seja um sucesso, deve-se ter em consideração os argumentos apresentados a seguir.

Denote-se a necessidade de reduzir a redundância física dos dados, de modo a garantir a integridade dos mesmos e a sua tolerância a possíveis falhas, para que o mesmo

evite a perda de dados. Deve-se, ainda, permitir o acesso de forma concorrente aos dados, para que estes estejam disponíveis num tempo mínimo para todas as entidades que necessitem de aceder aos mesmos.

Requisitos do projeto

Tendo por base o problema em questão do processo de agendamento e realização de exames imagiológicos de diferentes modalidades e tipos num determinado hospital, é indispensável a seguinte informação na base de dados:

- Pacientes
- Médicos
- Marcação (agenda para a marcação de exames)
- Exames (diferentes tipos)
- Modalidades
- Equipamento (a utilizar aquando da realização dos exames)
- Especialidade
- Morada

A clínica, no nosso caso em particular, tem de possuir acesso aos dados pessoais tanto dos pacientes, bem como dos médicos. Para os pacientes necessita de informação relativa ao seu nome, data de nascimento, número de utente, número de contribuinte, contactos, género, morada e sistema de comparticipação. Relativamente aos médicos precisa do nome, data de início de serviço, número de utente, número de contribuinte, morada e especialidade correspondente.

Para a marcação dos exames é fundamental haver um armazenamento dos horários ocupados, isto é, uma agenda, bem como dos médicos disponíveis. Sendo de realçar que também temos de ter em conta a sala.

A informação sobre os tipos de exame em causa bem como as suas correspondentes modalidades, têm de constar na base de dados e ser de fácil acesso. Adicionalmente, deve-se ter em consideração o preço do respetivo exame a realizar.

Caraterização dos diversos perfis

A base de dados (BD) em questão é criada com o propósito dos pacientes procederem ao processo de marcação de exames imagiológicos. Para que seja possível a utilização deste serviço, os pacientes devem constar na base de dados, sendo necessário associar à sua ficha os seus dados pessoais, nomeadamente, o nome, número de utente, NIF, data de nascimento, género, sistema de comparticipação, morada e contactos para um bom funcionamento da BD. Deste modo, será gerada uma agenda de forma a agilizar o processo de marcação, onde nesta irão constar todas as marcações dos exames que foram agendados.

Aquando do processo de agendamento de exames, é fundamental ter em consideração a disponibilidade das salas onde se realizam os exames, bem como a disponibilidade do respetivo médico atribuído na realização do exame em questão. Tendo em conta estas variáveis, o paciente poderá, por fim, proceder à marcação nas horas disponíveis para tal, consoante o horário de expediente da clínica.

Em conformidade com a estruturação da base de dados supracitada, pretende-se a implementação de interfaces direcionadas para.

- Pacientes: consta o local de marcação dos exames e visualização de informação relativa aos mesmos.
- Administradores: tal como, médicos e gestores que gerem o agendamento e acedem aos respetivos resultados.

Denote-se a existência de pré-requisitos para os métodos de operação, sendo imperativo o registo prévio do paciente na BD, caso não o possua, para que seja possível o agendamento do exame. O registo deste poderá ser efetivado via online ou até mesmo presencialmente, tal como a marcação dos exames.

Adicionalmente, a BD deverá apresentar os horários preenchidos ou por oposição, os horários disponíveis para marcação em conformidade com a disponibilidade das salas e do médico.

Exames Imagiológicos

A imagiologia é uma das áreas da medicina que mais tem evoluído nos últimos anos, colocando ao dispor do médico radiologista tecnologias avançadas de diagnóstico e intervenção. A imagiologia é a especialidade da medicina que recorre a imagens obtidas através de radiação, ultrassons ou radiofrequência para a realização de diagnóstico. Assim sendo, são exames complementares de diagnóstico, que permitem obter e reproduzir imagens do interior do corpo humano. São também requisitados, quando necessário, para o controlo evolutivo de algumas doenças e avaliação da resposta à terapêutica instituída. Cada tipo de técnica necessita de um equipamento específico que possibilita a criação de imagens com informação particular e indicações próprias.

De entre os demais exames imagiológicos, que a nossa BD referencia como modalidade, temos a ressonância magnética (RM), a ecografia, o ecocardiograma, a radiografia (raio-X) e a endoscopia.

O exame de **ressonância magnética (RM)** é uma técnica não invasiva que permite a aquisição de imagens precisas, tanto de anatomia bem como da fisiologia do corpo humano, sendo bastante utilizada para o diagnóstico de alterações no sistema nervoso central, tecidos moles e articulações. Assume-se como um método seguro, uma vez que não recorre a radiação ionizante. Durante a realização do exame, o paciente deve manter a imobilidade, caso contrário altera a qualidade das imagens.

A **ecografia** é uma técnica de imagem médica não invasiva, que utiliza ultrassons (gerados por um aparelho denominado ecógrafo) e um computador para avaliar as estruturas anatómicas do nosso organismo. Para avaliar diferentes áreas do nosso organismo são utilizadas diferentes sondas, que emitem ultrassons com diferentes frequências, dependendo da profundidade do órgão que se pretende avaliar. É utilizada com bastante frequência para exames gerais, ginecológicos, obstétricos, abdominais, vasculares e urológicos. Tende a ser o primeiro exame de diagnóstico antes da realização de uma RM ou TAC. Esta técnica não envolve raios-X.

O **ecocardiograma** é um exame de ultrassom do coração, o qual fornece imagens obtidas através do som. Através dessas imagens, o médico especialista pode analisar se o coração está a bater normalmente, e se o fluxo sanguíneo está adequado. Desta forma também é possível diagnosticar doenças que afetam a anatomia e a fisiologia do coração, como insuficiências e malformações cardíacas.

A **radiografia** é a técnica de imagem médica mais antiga. O raio x é o exame de 1ª linha na avaliação da anatomia humana que auxilia o médico no diagnóstico de várias patologias (doenças), assim como para controlar a evolução de tratamentos. Este exame versátil permite a visualização de ossos, pulmões e o abdômen. Pode ser utilizada em várias aplicações, incluindo na obtenção de imagens em tempo real e na realização de estudos de doentes intransportáveis com equipamentos portáteis. Atualmente, os raios X, em muito maior dose, são usados para o tratamento de cânceros, de modo a destruir tumores e células cancerígenas, danificando seu DNA. Os raios-X mais comuns são o de tórax e da coluna.

A **endoscopia** é um exame que coleta imagens em tempo real do aparelho digestivo, do esôfago ao duodeno, passando pelo estômago. Além de ser utilizada como método de diagnóstico, a endoscopia pode também funcionar como um tratamento propriamente, por exemplo, por meio dela, dá para introduzir sondas no estômago ou retirar tumores.

Validação do modelo conceptual

Nesta fase da validação do projeto foi utilizada a ferramenta TerraER, de forma a criar um modelo conceptual para o sistema de base de dados que pretendemos fazer. No final obtivemos o seguinte resultado:

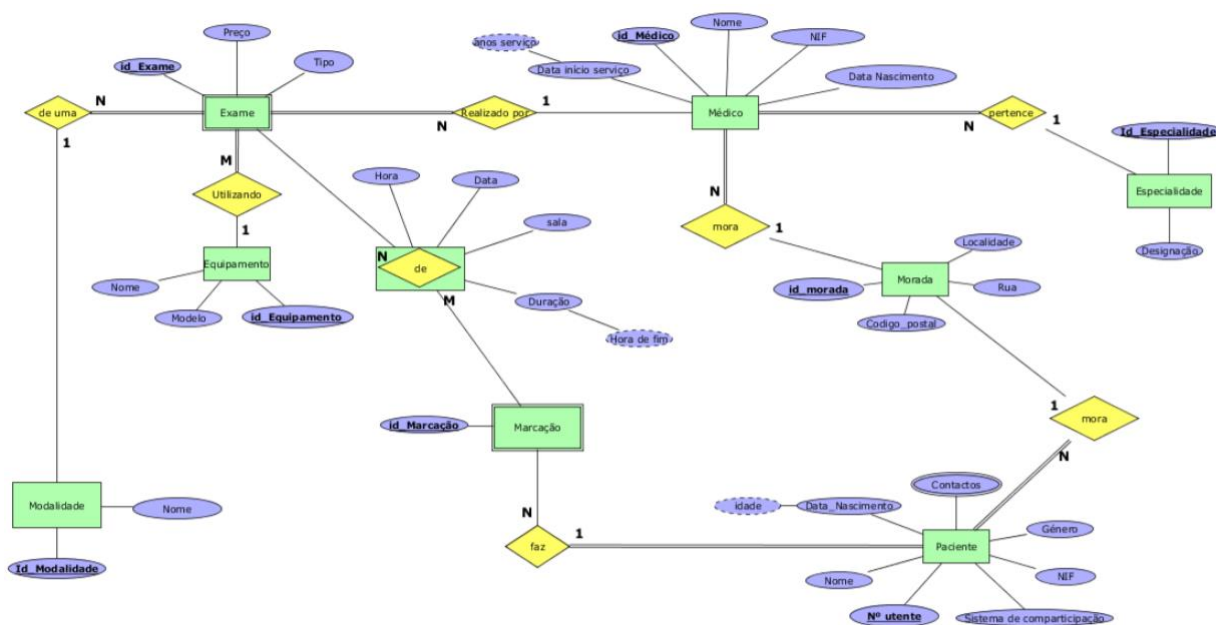


Figura 1: Representação do modelo conceptual da base de dados realizado no TerraER.

Identificação das Entidades do Modelo Conceptual

Para o modelo construído anteriormente podemos enumerar as seguintes entidades:

- Paciente
- Marcação
- Médico
- Especialidade
- Equipamento
- Exame
- Modalidade
- Morada

Tabela 1: Identificação e descrição das Entidades do Modelo Relacional.

Entidade	Descrição
Paciente	Designação utilizada para descrever os pacientes da clínica
Marcação	Designação utilizada para descrever a realização da marcação de um exame
Médico	Designação utilizada para descrever o médico que realiza o exame
Especialidade	Designação utilizada para descrever a área em que o médico que realiza o exame é especializado
Equipamento	Designação utilizada para descrever o equipamento utilizado durante o exame
Exame	Designação utilizada para descrever o tipo de especialização do exame a ser realizado
Modalidade	Designação utilizada para descrever o tipo de exame a ser realizado
Morada	Designação utilizada para descrever a residência do paciente que marca o exame

Identificação das Relacionamentos

No modelo Conceptual existem 8 tipos de relacionamentos, sendo eles entre dois tipos de entidades.

Relacionamento “Paciente” → “Marcação”

A relação entre “Paciente” e “Marcação” em termos de multiplicidade trata-se de um relacionamento de 1 : N. Isto significa que cada um dos pacientes pode fazer ‘N’ marcações.



Relacionamento “Paciente” → “Morada”

A relação entre “Paciente” e “Morada” em termos de multiplicidade trata-se de um relacionamento de N : 1. Isto significa que ‘N’ pacientes podem corresponder a 1 morada.



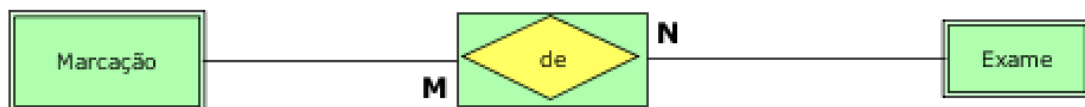
Relacionamento “Médico” → “Morada”

A relação entre “Médico” e “Morada” em termos de multiplicidade trata-se de um relacionamento de N : 1. Isto significa que ‘N’ médicos podem corresponder a 1 morada.



Relacionamento “Marcação” → “Exame”

A relação entre “Marcação” e “Exame” em termos de multiplicidade trata-se de um relacionamento de N : N. Isto significa que para ‘N’ marcações podem ser realizados ‘N’ exames.



Relacionamento “Exame” → “Modalidade”

A relação entre “Exame” e “Modalidade” em termos de multiplicidade trata-se de um relacionamento de N : 1. Isto significa que ‘N’ exames podem ser feitos para 1 modalidade.



Relacionamento “Exame” → “Médico”

A relação entre “Exame” e “Médico” em termos de multiplicidade trata-se de um relacionamento de N : 1. Isto significa que ‘N’ exames podem ser realizados por 1 médico.



Relacionamento “Médico” → “Especialidade”

A relação entre “Médico” e “Especialidade” em termos de multiplicidade trata-se de um relacionamento de N : 1. Isto significa que ‘N’ médicos podem ter 1 especialidade.



Relacionamento “Exame” → “Equipamento”

A relação entre “Exame” e “Equipamento” em termos de multiplicidade trata-se de um relacionamento de N : 1. Isto significa que ‘N’ exames podem usar 1 equipamento.



Resumidamente:

Tabela 2: Relacionamentos entre as diferentes Entidades.

Entidade	Multiplicidade	Relação	Multiplicidade	Entidade
Paciente	1	faz	N	Marcação
Paciente	N	mora	1	Morada
Médico	N	mora	1	Morada
Marcação	N	de	N	Exame
Exame	N	de uma	1	Modalidade
Exame	N	realizado por	1	Médico
Médico	N	pertence	1	Especialidade
Exame	N	utilizando	1	Equipamento

Identificar e associar atributos com entidades e tipos de relacionamentos

Nesta fase foram abordados os atributos das várias entidades presentes no modelo conceptual, uma vez que não existe nenhum atributo que afete os relacionamentos entre as entidades.

A entidade 'Paciente' tem 8 atributos.

A chave primária desta entidade é o atributo 'Nº utente', uma vez que o paciente terá este número associado na clínica. Além deste, temos alguns atributos simples, como o 'Nome', o 'Género', o 'NIF', o 'Sistema de Comparticipação', os 'Contactos', que são um atributo multivalor, e a 'Data de Nascimento', onde temos como atributo derivado a 'Idade'.

A entidade 'Morada' tem 4 atributos.

A chave primária desta entidade é o atributo 'id_morada', uma vez que a morada para cada paciente e médico terá um identificador dado pela clínica. Além deste, temos alguns atributos simples, como a 'Localidade', o 'Codigo_postal' e a 'Rua'.

A entidade 'Médico' tem 6 atributos.

A chave primária desta entidade é o atributo 'id_Medico', uma vez que o médico terá um número atribuído pela clínica. Além deste, temos alguns atributos simples, como o 'Nome', a 'Data_Nascimento', o 'NIF' e a 'Data_inicio_servico', onde temos como atributo derivado os 'anos serviço'.

A entidade 'Especialidade' tem 2 atributos.

A chave primária desta entidade é o atributo 'id_Especialidade', uma vez que a especialidade terá um número atribuído pela clínica. Além deste, temos um atributo simples sendo ele a 'Designacao'.

A entidade 'Marcação' tem 1 atributo.

A chave primária desta entidade é o atributo 'id_Marcacao', uma vez que a marcação terá um número atribuído pela clínica.

O relacionamento da 'Marcação_has_Exame' tem 5 atributos.

Este é composto por atributos simples com a 'Data', a 'Hora', a 'Sala' e a 'Duracao', onde temos como atributo derivado a 'Hora_fim'.

A entidade 'Exame' tem 3 atributos.

A chave primária desta entidade é o atributo 'id_Exame', uma vez que o exame terá um número atribuído pela clínica. Além deste, temos alguns atributos simples, como o 'Preco' e o 'Tipo'.

A entidade 'Equipamento' tem 3 atributos.

A chave primária desta entidade é o atributo 'id_Equipamento', uma vez que o equipamento terá este número atribuído pela clínica. Além deste, temos alguns atributos simples, como o 'Nome' e o 'Modelo'.

A entidade 'Modalidade' tem 2 atributos.

A chave primária desta entidade é o atributo 'id_Modalidade', uma vez que a modalidade terá um número atribuído pela clínica. Além deste, temos um atributo simples sendo ele o 'Nome'.

Identificação dos tipos dos atributos

Na tabela em baixo encontram-se expostas as características dos atributos das diferentes Entidades e relações.

Tabela 3: Características dos atributos de cada Entidade.

Entidade/Relação	Atributo	Característica
Paciente	Nº utente	Inteiro
	Nome	Até 250 caracteres
	Contactos	Exatamente 9 dígitos
	NIF	Inteiro
	Género	Até 250 caracteres

	Sistema de Participação	Até 250 caracteres
	Data de Nascimento	Exatamente 8 dígitos
	Idade	Inteiro
Morada	id_morada	Inteiro
	Localidade	Até 250 caracteres
	Código_postal	Até 8 caracteres
	Rua	Até 250 caracteres
Médico	id_Médico	Inteiro
	Nome	Até 250 caracteres
	Data Nascimento	Exatamente 8 dígitos
	NIF	Inteiro
	Data início serviço	Exatamente 8 dígitos
	anos serviço	Inteiro
Especialidade	id_Especialidade	Inteiro
	Designação	Até 45 caracteres
Marcação	id_Marcação	Inteiro
'Marcação_has_Exame'	Sala	Até 45 caracteres
	Hora	Exatamente 4 dígitos

	Data	Exatamente 8 dígitos
	Duração	Exatamente 4 dígitos
	Hora de fim	Exatamente 4 dígitos
Exame	id_Exame	Inteiro
	Preço	Inteiro
	Tipo	Até 250 caracteres
Equipamento	Id_Equipamento	Inteiro
	Nome	Até 45 caracteres
	Modelo	Até 45 caracteres
Modalidade	id_Modalidade	Inteiro
	Nome	Até 250 caracteres

Determinar chaves candidatas, primárias e alternativas

As chaves primárias das diferentes entidades presentes no modelo conceitual são as seguintes:

Paciente → n° utente

Usou-se o N° utente como chave primária uma vez que todos os pacientes este número único atribuído;

Morada → id_morada

Usou-se o id_morada como chave primária uma vez que todas as moradas têm um código único atribuído pelo hospital;

Médico → id_Médico

Usou-se o id_Médico como chave primária uma vez que todos os médicos têm um código único atribuído pelo hospital;

Especialidade → id_Especialidade

Usou-se o id_Especialidade como chave primária uma vez que todas as especialidades têm um código único atribuído pelo hospital;

Marcação → id_Marcação

Usou-se o id_Marcação como chave primária uma vez que todas as marcações têm um código único atribuído pelo hospital;

Exame → id_Exame

Usou-se o id_Exame como chave primária uma vez que todos os exames têm um código único atribuído pelo hospital;

Equipamento → id_Equipamento

Usou-se o id_Equipamento como chave primária uma vez que todos os equipamentos têm um código único atribuído pelo hospital;

Modalidade → id_Modalidade

Usou-se o id_Modalidade como chave primária uma vez que todas as modalidades têm um código único atribuído pelo hospital;

Verificar redundância do modelo

Nesta fase o modelo conceptual é examinado de modo a identificar a presença de qualquer tipo de redundância, caso seja encontrada alguma esta deverá ser eliminada. Para verificar tal acontecimento, forma seguidas três etapas:

1. Voltar a examinar as relações de um para um (1:1). Como no nosso modelo não existe qualquer tipo de relação um para um, neste passo não há nada a alterar.
2. Remover relações redundantes. Para este passo remover-se-iam ciclos, ou seja, quando conseguimos obter uma informação a partir de relações diferentes. Tal como no passo anterior, no modelo construído não existe nenhuma situação em que isto acontece, não sendo assim necessário fazer alterações.
3. Considerar dimensão do tempo. Neste último passo verificamos se o modelo elaborado continua inalterado depois de uma nova revisão. Esta condição temporal não teve nenhum efeito no modelo desenvolvido.

Modelo lógico

O modelo lógico é criado no MySQLWorkbench com base no modelo conceptual anteriormente referido. Cada entidade do modelo conceptual corresponde a uma tabela no modelo lógico, nas quais as colunas correspondem aos atributos.

Nesta fase, são estabelecidas algumas restrições como atributos não nulos e atributos nulos, bem como o tipo de dados a inserir. São definidas as chaves primárias, já referidas no modelo conceptual e as chaves estrangeiras, que podem também ser chaves primárias.

Validação do modelo lógico

Na segunda parte da validação, seguindo os vários passos da metodologia presente no livro, chegou-se ao seguinte resultado:

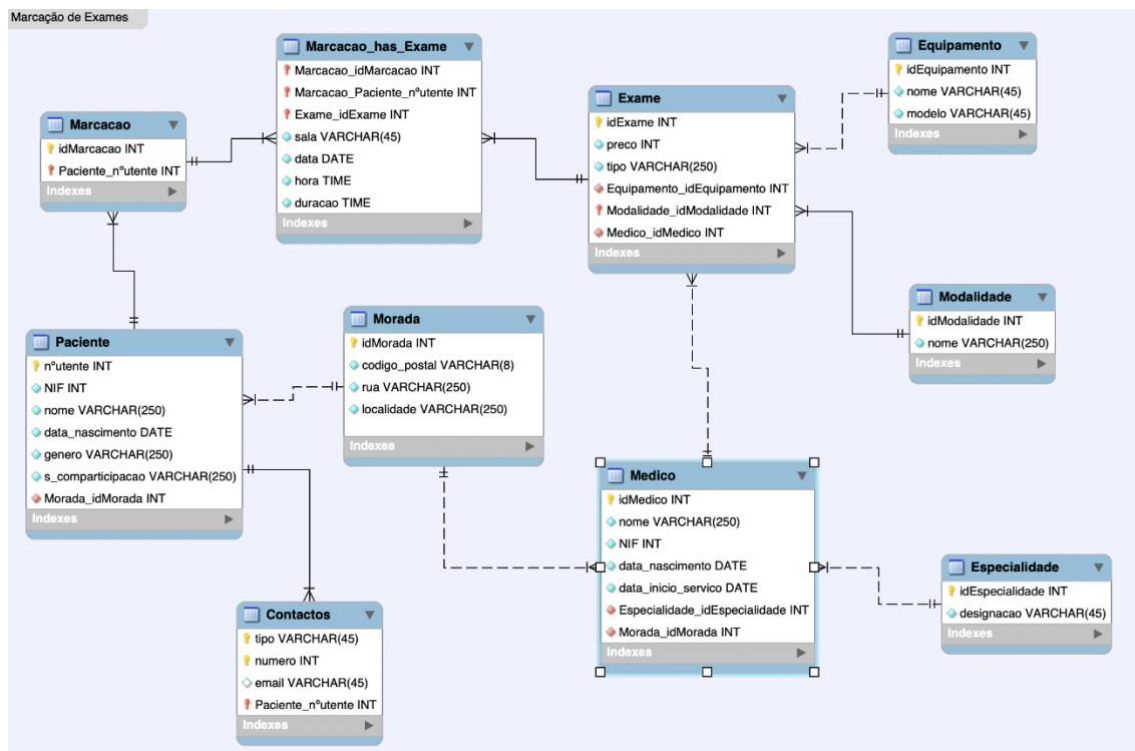


Figura 2 – Modelo Lógico da Base de Dados

Derivar relações do modelo conceptual

O objetivo desta etapa do trabalho é criar relações para o modelo lógico e assim, representar entidades, relacionamentos e atributos que foram identificados já no modelo conceptual.

A composição de cada relação, foi descrita usando DBDL (*Database Definition Language*) para modelos relacionais.

Entidades Fortes

Uma entidade forte é um tipo de entidade cuja existência não depende de nenhuma outra entidade e tem um atributo-chave. O atributo chave ajuda a identificar cada entidade de maneira única.

No nosso modelo conceptual existem 6 entidades fortes, Paciente, Médico, Morada, Especialidade, Equipamento e Modalidade.

Paciente: N° utente (INT), NIF (INT), Nome (VARCHAR (250)), Data de nascimento (DATE), Idade (VARCHAR(45)), Género (VARCHAR (250)), Sistema comparticipação (VARCHAR (250)), idMorada (INT)

- **Chave Primária:** N° utente (INT)
- **Chave Estrangeira:** idMorada (INT)

Médico: IdMédico (INT), Nome (VARCHAR (250)), NIF (INT), Data de nascimento (DATE), Data de início de serviço (DATE), Anos de serviço (VARCHAR(45)), IdEspecialidade (INT), IdMorada (INT)

- **Chave Primária:** idMédico (INT)
- **Chaves Estrangeiras:** idEspecialidade (INT), idMorada (INT)

Morada: IdMorada (INT), Código postal (VARCHAR (8)), Rua (VARCHAR (250)), Localidade (VARCHAR (250))

- **Chave Primária:** idMorada (INT)

Especialidade: IdEspecialidade (INT), Designação (VARCHAR (45))

- **Chave Primária:** idEspecialidade (INT)

Equipamento: IdEquipamento (INT), Nome (VARCHAR (45)), modelo (VARCHAR (45))

- **Chave Primária:** idEquipamento (INT)

Modalidade: IdModalidade (INT), Nome (VARCHAR (250))

- **Chave Primária:** idModalidade (INT)

Entidades Fracas

As entidades fracas referem-se a entidades em que a sua existência depende de outras entidades, não tem um atributo de chave. Em vez disso, uma entidade fraca tem uma chave parcial ou discriminadores. Esses discriminadores, juntamente com o atributo-chave da entidade forte, atuam como uma chave primária para a entidade fraca.

No nosso modelo conceptual existem 2 entidades fracas, Exame e Marcação.

Exame: idExame (INT), Preço (INT), Tipo (VARCHAR (250)), IdEquipamento (INT), IdModalidade (INT), IdMedico (INT)

- **Chave Primária:** idExame (INT), IdModalidade (INT)
- **Chave Estrangeira:** IdEquipamento (INT), IdModalidade (INT), IdMedico (INT)

Marcação: idMarcação (INT), N°utente (INT)

- **Chave Primária:** idMarcação (INT), N°utente (INT)
- **Chave Estrangeira:** N°utente (INT), idMarcação (INT)

Entidades de relação

As entidades de relação são entidades que se relacionam com outras 2 entidades numa relação de n:m, sendo esta também considerada uma entidade fraca.

No nosso modelo conceptual existe uma entidade de relação, a Marcação_has_Exame.

Marcação_has_Exame: idMarcação (INT), N°utente (INT), idExame (INT), sala (VARCHAR (45)), Data (DATE), Hora_inicio (TIME), Duração (TIME), Hora_fim (TIME)

- **Chave Primária:** idMarcação (INT), N°utente (INT), idExame (INT)
- **Chave Estrangeira:** idMarcação (INT), N°utente (INT), idExame (INT)

Atributos Multivalor

Um atributo multivalor é um atributo que pode ter mais de um valor associado. Existe um atributo multivalor no nosso modelo, Contactos. Este atributo faz parte do Paciente.

Contactos: tipo (VARCHAR (45)), Número (INT), Email (VARCHAR (45)), N°utente (INT)

- **Chave Primária:** tipo (VARCHAR(45)), Número (INT)
- **Chave Estrangeira:** N°utente (INT)
- **Chave Alternativa:** Email (VARCHAR(45))

Normalização

1ª Forma Normal

O primeiro passo do processo de normalização (1ª FN), visa eliminar grupos de valores repetidos para um dado atributo, ou conjunto de atributos num dado tuplo. A solução passa por decompor a relação inicial criando novas relações, tantas quantas o número de grupos que se repetem. A nova relação terá como atributos a chave primária da relação de onde é originária, bem como os atributos que fazem parte do grupo que se repete.

Analisando o nosso modelo, verifica-se que para a entidade médico, por exemplo, podem ser marcados vários exames.

Assim, após a 1ºFN é possível constatar que em todas as tabelas do modelo não existe a ocorrência de grupos repetidos.

2ª Forma Normal

Uma relação na 2ºFN é uma relação em que todos os atributos não pertencentes a qualquer chave candidata, devem depender da totalidade da chave.

Como anteriormente foi provado, o modelo encontra-se de acordo com a 1ª forma normal, que é o primeiro requisito necessário para estar de acordo com a segunda. Outra condição necessária, à 2ª forma normal, é que todas as chaves não primárias sejam dependentes da chave primária – algo que, através da observação das tabelas existentes, é também possível de confirmar. Existem, no entanto, chaves que poderiam por si mesmas identificar uma tabela, mas, no entanto, foram anteriormente consideradas chaves candidatas, o que as torna também dependentes.

3ª Forma Normal

Uma relação na 3ªFN é uma relação em que não existem dependências funcionais entre atributos não chave (dependências transitivas).

Para as tabelas do nosso modelo se encontrarem de acordo com a 3ª forma normal, é necessário que cumpram também a 2ª FN.

Como esta condição se aplica, verifica-se depois se alguma das colunas existentes na tabela é dependente de outra(s), algo que não acontece em nenhuma das tabelas existentes no modelo lógico construído.

Validação do modelo físico

Tradução do modelo lógico para o modelo físico

Após concluída a validação do modelo lógico, partiu-se para a criação do modelo físico. Para isso, foi usada a ferramenta Forward Engineer do software MySQL Workbench, de modo a elaborar automaticamente o código SQL. O código gerado para cada tabela é o seguinte:

```
CREATE TABLE IF NOT EXISTS `trabalho`.`Morada` (  
  `idMorada` INT NOT NULL,  
  `codigo_postal` VARCHAR(8) NOT NULL,  
  `rua` VARCHAR(250) NOT NULL,  
  `localidade` VARCHAR(250) NOT NULL,
```

```

        PRIMARY KEY (`idMorada`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `trabalho`.`Paciente` (
  `n°utente` INT NOT NULL,
  `NIF` INT NOT NULL,
  `nome` VARCHAR(250) NOT NULL,
  `data_nascimento` DATE NOT NULL,
  `idade` VARCHAR(45) NULL,
  `genero` VARCHAR(250) NOT NULL,
  `s_comparticipacao` VARCHAR(250) NOT NULL,
  `Morada_idMorada` INT NOT NULL,
  PRIMARY KEY (`n°utente`),
  INDEX `fk_Paciente_Morada1_idx` (`Morada_idMorada` ASC)
  VISIBLE,
  CONSTRAINT `fk_Paciente_Morada1`
    FOREIGN KEY (`Morada_idMorada`)
      REFERENCES `trabalho`.`Morada` (`idMorada`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `trabalho`.`Marcacao` (
  `idMarcacao` INT NOT NULL,
  `Paciente_n°utente` INT NOT NULL,
  PRIMARY KEY (`idMarcacao`, `Paciente_n°utente`),
  INDEX `fk_Marcacao_Paciente_idx` (`Paciente_n°utente` ASC)
  VISIBLE,
  CONSTRAINT `fk_Marcacao_Paciente`
    FOREIGN KEY (`Paciente_n°utente`)
      REFERENCES `trabalho`.`Paciente` (`n°utente`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `trabalho`.`Equipamento` (
  `idEquipamento` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `modelo` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idEquipamento`))

```

```
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `trabalho`.`Modalidade` (  
  `idModalidade` INT NOT NULL,  
  `nome` VARCHAR(250) NOT NULL,  
  PRIMARY KEY (`idModalidade`))  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `trabalho`.`Especialidade` (  
  `idEspecialidade` INT NOT NULL,  
  `designacao` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idEspecialidade`))  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `trabalho`.`Medico` (  
  `idMedico` INT NOT NULL,  
  `nome` VARCHAR(250) NOT NULL,  
  `NIF` INT NOT NULL,  
  `data_nascimento` DATE NOT NULL,  
  `data_inicio_servico` DATE NOT NULL,  
  `anos_servico` VARCHAR(45) NULL,  
  `Especialidade_idEspecialidade` INT NOT NULL,  
  `Morada_idMorada` INT NOT NULL,  
  PRIMARY KEY (`idMedico`),  
  INDEX `fk_Medico_Especialidadel_idx`  
  (`Especialidade_idEspecialidade` ASC) VISIBLE,  
  INDEX `fk_Medico_Moradal_idx` (`Morada_idMorada` ASC) VISIBLE,  
  CONSTRAINT `fk_Medico_Especialidadel`  
    FOREIGN KEY (`Especialidade_idEspecialidade`)  
    REFERENCES `trabalho`.`Especialidade` (`idEspecialidade`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Medico_Moradal`  
    FOREIGN KEY (`Morada_idMorada`)  
    REFERENCES `trabalho`.`Morada` (`idMorada`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `trabalho`.`Exame` (  
  `idExame` INT NOT NULL,
```

```

`preco` INT NOT NULL,
`tipo` VARCHAR(250) NOT NULL,
`Equipamento_idEquipamento` INT NOT NULL,
`Modalidade_idModalidade` INT NOT NULL,
`Medico_idMedico` INT NOT NULL,
PRIMARY KEY (`idExame`, `Modalidade_idModalidade`),
INDEX `fk_Exame_Equipamento1_idx` (`Equipamento_idEquipamento`
ASC) VISIBLE,
INDEX `fk_Exame_Modalidade1_idx` (`Modalidade_idModalidade`
ASC) VISIBLE,
INDEX `fk_Exame_Medico1_idx` (`Medico_idMedico` ASC) VISIBLE,
CONSTRAINT `fk_Exame_Equipamento1`
FOREIGN KEY (`Equipamento_idEquipamento`)
REFERENCES `trabalho`.`Equipamento` (`idEquipamento`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Exame_Modalidade1`
FOREIGN KEY (`Modalidade_idModalidade`)
REFERENCES `trabalho`.`Modalidade` (`idModalidade`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Exame_Medico1`
FOREIGN KEY (`Medico_idMedico`)
REFERENCES `trabalho`.`Medico` (`idMedico`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `trabalho`.`Contactos` (
`tipo` VARCHAR(45) NOT NULL,
`numero` INT NOT NULL,
`email` VARCHAR(45) NULL,
`Paciente_n°utente` INT NOT NULL,
PRIMARY KEY (`tipo`, `numero`, `Paciente_n°utente`),
INDEX `fk_Contactos_Paciente1_idx` (`Paciente_n°utente` ASC)
VISIBLE,
CONSTRAINT `fk_Contactos_Paciente1`
FOREIGN KEY (`Paciente_n°utente`)
REFERENCES `trabalho`.`Paciente` (`n°utente`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)

```

```
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `trabalho`.`Marcacao_has_Exame` (
  `Marcacao_idMarcacao` INT NOT NULL,
  `Marcacao_Paciente_n°utente` INT NOT NULL,
  `Exame_idExame` INT NOT NULL,
  `sala` VARCHAR(45) NOT NULL,
  `data` DATE NOT NULL,
  `hora_inicio` TIME NOT NULL,
  `duracao` TIME NOT NULL,
  `hora_fim` TIME NULL,
  PRIMARY KEY (`Marcacao_idMarcacao`,
  `Marcacao_Paciente_n°utente`, `Exame_idExame`),
  INDEX `fk_Marcacao_has_Exame_Exame1_idx` (`Exame_idExame` ASC)
  VISIBLE,
  INDEX `fk_Marcacao_has_Exame_Marcacao1_idx`
  (`Marcacao_idMarcacao` ASC, `Marcacao_Paciente_n°utente` ASC)
  VISIBLE,
  CONSTRAINT `fk_Marcacao_has_Exame_Marcacao1`
    FOREIGN KEY (`Marcacao_idMarcacao`
, `Marcacao_Paciente_n°utente`)
    REFERENCES `trabalho`.`Marcacao` (`idMarcacao`
, `Paciente_n°utente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Marcacao_has_Exame_Exame1`
    FOREIGN KEY (`Exame_idExame`)
    REFERENCES `trabalho`.`Exame` (`idExame`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Análise de transação/relação

Neste ponto, proceder-se-á análise de transações/relações do modelo físico, de forma a entender de maneira mais exata a atividade nas diferentes tabelas.

Para obter resultados, testar-se-á as *queries* listadas:

1. Quantos exames foram realizados por cada paciente?
2. De todas os exames realizados na clínica, qual foi a data e hora do

primeiro e último exame da clínica?

3. Qual o preço dos exames realizados pela paciente Carla Guimarães?

4. Procurar todos os pacientes que tiveram exames de manhã.

5. Qual é o nome de todos os médicos e a respetiva especialidade que cada um exerce?

Para os implementar foram utilizados códigos SQL, que são os seguintes:

Quantos exames foram realizados por cada paciente?

```
select p.nome, count(*) from Marcacao m, Paciente p
where m.Paciente_nºutente=p.nºutente
group by p.nome
order by count(*) desc;
```

Tendo em conta o nome de cada paciente foi feita a contagem de cada exame realizado ordenando de forma decrescente, isto é, os pacientes que realizaram mais exames aparecem primeiro.

De todos os exames realizados na clínica, qual foi a data e hora do primeiro e último exame da clínica?

```
(select m.data, m.hora_inicio from Marcacao_has_Exame m
order by m.data, m.hora_inicio limit 1)
union all
((select m2.data, m2.hora_inicio from Marcacao_has_Exame m2
order by m2.data desc, m2.hora_inicio desc) limit 1);
```

Ordenou-se primeiramente de forma crescente os dois atributos para que o primeiro valor (mínimo) corresponda ao primeiro exame realizado na clínica e de seguida, ordenou-se por ordem decrescente para selecionar qual foi o último realizado.

Qual o preço dos exames realizados pela paciente Carla Guimarães?

```
select distinct p.nome,ex.tipo, ex.preco from Paciente p, Exame
ex, Marcacao m, Marcacao_has_Exame me
where p.nºutente=m.Paciente_nºutente and
m.Paciente_nºutente=me.Marcacao_Paciente_nºutente and
me.Exame_idExame =ex.idExame and p.nome = 'Carla Guimarães';
```

Tendo-se em consideração o nome específico do paciente e o tipo do exame a realizar foi obtido o preço pago pela paciente Carla Guimarães para todos os exames que realizou.

Procurar todos os pacientes que realizaram exames de manhã.

```
select p.nome,me.hora_inicio from paciente p,marcacao_has_exame
me
where me.hora_inicio between '9:00:00' and '12:00:00' and
me.Marcacao_Paciente_nºutente=p.nºutente
order by me.hora_inicio;
```

Para procurar todos os pacientes que realizaram exames de manhã, pressupôs-se o horário de realização entre as 9h e as 12h. Para tal seleccionamos os nomes dos pacientes no horário em causa por ordem crescente (desde o início do dia).

Qual é o nome de todos os médicos e a respetiva especialidade que cada um exerce?

```
select m.nome, e.designacao
from Medico m
inner join Especialidade e
on m.Especialidade_idEspecialidade=e.idEspecialidade;
```

Visa obter o nome de cada médico e a especialidade respetiva por meio do inner join.

Povoamento das tabelas do modelo físico

O povoamento das tabelas do modelo físico foi executado utilizando dados aleatórios, para assim nos ser possível validar os nossos modelos.

Para os implementar foram utilizados os seguintes códigos SQL:

```
INSERT INTO trabalho.marcacao_has_exame (hora_fim) SELECT
hora_fim(hora_inicio,duracao) FROM marcacao_has_exame;

INSERT INTO trabalho.Paciente (idade) SELECT
idade(data_nascimento) FROM Paciente;

INSERT INTO trabalho.Medico (anos_servico) SELECT
idade(data_inicio_servico) FROM Medico;

INSERT INTO trabalho.Morada (idMorada, codigo_postal, rua,
localidade) VALUES (1, "4710-654","Rua das Canas 256", "Braga"),
(2,"4714-542","Rua das Maias 12","Braga"), (3,"4710-
008","Travessa da Lage Gualtar 05", "Braga"), (4,"4767-
501","Avenida Engenheiro Pinheiro Braga Gavião 9020","Vila Nova
de Famalicão"), (5, "4804-534", "Largo Cónego José Maria Gomes
08", "Guimarães"), (6,"4723-120","Ferreiros e Gondizalves 456",
"Braga"), (7,"4725-102", "Lomar e Arcos 20", "Braga"), (8,"4743-
456","Merelim (Sao Pedro) e Frossos 59", "Braga"), (9,"4763-
102", "Tadim 01", "Braga"), (10,"4670-580", "Tadim 25",
"Braga"), (11, "4700-033","Real, Dume e Semelhe 36", "Braga"),
(12, "4702-125","Pedralva 125", "Braga" ), (13,"4789-036","Este
(Sao Pedro e Sao Mamede) 23", "Braga"), (14, "4701-145","Crespos
e Pousada 258", "Braga"), (15, "4769-003", "Esporoes 458",
"Braga"), (16, "4725-009", "Sobreposta 80", "Braga"), (17,
"4788-118","Maximinos, Se e Cividade 14", "Braga");

INSERT INTO trabalho.Equipamento (idEquipamento, Nome, Modelo)
VALUES (1, "Sonda Linear","DX-523"),(2,"Digital
Direto","PerformX Digi AT"),(3,"Convencional","BReeZe -
Arcom"),(4,"Endoscópio","AR-240"),(5,"Sonda
Ultrassónica","Breil-34D"),(6,"NeuVantage","1.5T -
Neusoft"),(7,"NeuVantage","0.35T - Neusoft");
```



```

INSERT INTO trabalho.Modalidade (idModalidade, nome) VALUES
(1,"Ressonância
Magnética"), (2,"Endoscopia"), (3,"Ecocardiograma"), (4,"Ecografia"
), (5,"Radiografia");

INSERT INTO trabalho.Especialidade (idEspecialidade, designacao)
VALUES (11, "Gastrenterologia"), (13,"Radiologia"), (12,
"Cardiologia");

INSERT INTO trabalho.Paciente (n°utente, NIF, nome,
data_nascimento, idade, genero, s_comparticipacao,
Morada_idMorada) VALUES (460505826,215685778,"Joana
Ferreira","1975-12-08",idade(data_nascimento), "F", "SNS",
6), (182552802,002855422,"Ana Luisa Pereira","1953-10-
13",idade(Data_Nascimento),"F","AdvanceCare",
8), (134330630,220348416,"José Lopes","1960-08-
12",idade(data_nascimento),"M", "null",
15), (218637581,870542216,"Roberto Pereira","2002-10-
18",idade(data_nascimento),"M","SNS",
17), (125285547,135615512,"Carla Guimarães","1972-03-
03",idade(data_nascimento),"F","Multicare",
7), (271480643,344785105,"Rita Rodrigues","1972-03-
03",idade(data_nascimento),"F","SNS",9), (161432848,626725828,"An
a Faria","2003-09-
29",idade(data_nascimento),"F","null",12), (176616010,510716528,"
Elsa Almeida","1967-06-
09",idade(data_nascimento),"F","Medicare",14), (177551020,1736218
15,"Gonçalo Martins","1953-10-
10",idade(data_nascimento),"M","AdvanceCare",10),
(257278212,423064602,"Francisco Ribeiro","2004-09-
25",idade(data_nascimento),"M","Medicare",16), (275365586,2604634
37,"Joao Pedro Pais","1970-01-
26",idade(data_nascimento),"M","null",
11), (158248466,287376541,"Guilherme Monteiro","1994-10-
03",idade(data_nascimento),"M","SNS",13);

INSERT INTO trabalho.Medico (idMedico, nome, NIF,
data_nascimento, data_inicio_servico, anos_servico,
Especialidade_idEspecialidade, Morada_idMorada) VALUES (1,"João
Pereira",152342376, "1982-09-03","2012-05-
22",idade(data_inicio_servico),13,2), (2,"Gaspar
Carvalho",295829393,"1965-02-01","2007-03-
25",idade(data_inicio_servico),11,5), (3,"Sara

```

```

Pereira",192348395,"1990-12-27","2011-07-
04",idade(data_inicio_servico),13,1),(4,"Jorge
Santos",214235234,"1980-03-29","2010-02-
15",idade(data_inicio_servico),12,4),(5,"Raquel
Oliveira",285301352,"1978-12-06","2007-05-
24",idade(data_inicio_servico),11,3);

INSERT INTO trabalho.Marcacao (idMarcacao, Paciente_n°utente)
VALUES
(0,460505826),(1,177551020),(2,125285547),(3,125285547),(4,16143
2848),(5,218637581),(6,182552802),(7,182552802),(8,257278212),(9
,158248466),(10,177551020),(11,275365586),(12,134330630),(13,158
248466),(14,134330630),(15,271480643),(16,176616010),(17,1252855
47),(18,460505826),(19,158248466),(20,275365586);

INSERT INTO trabalho.Contactos (tipo, numero, email,
Paciente_n°utente) VALUES
("tel",967044406,"TNjyiOlA@gmail.com",460505826),("tel",93388369
5,"mGgxMgS@gmail.com",182552802),("telef",252003698,"null",13433
0630),("tel",968216431,"fHvMLviMBC@gmail.com",218637581),("tel",
969306343,"bcYfpAgyG@gmail.com",125285547),("tel",962537975,"tqC
vHuHW@gmail.com",271480643),("tel",915730981,"null",161432848), (
"tel",930916347,"TeCxcjnu@gmail.com",176616010),("telef",2541238
00,"null",177551020),("tel",917081676,"QvlaGehys@gmail.com",2572
78212),("tel",963304780,"TGHatQlS@gmail.com",275365586),("tel",9
37179165,"mfftqIrJPb@gmail.com",158248466);

INSERT INTO trabalho.Exame (idExame, preco, tipo,
Equipamento_idEquipamento, Modalidade_idModalidade,
Medico_idMedico) VALUES
(1,40,"Abdominal",3,4,1),(2,30,"Torácica",2,5,3),(3,30,"Mama",2,
4,1),(4,30,"Obstétrica",3,4,1),(5,30,"Mão",4,5,1),(6,30,"Pé",4,5
,3),(7,40,"Lombar",4,5,3),(8,40,"Face",4,5,1),(9,50,"Digestiva
Alta",4,2,2),(10,50,"Digestiva
Baixa",4,2,5),(11,40,"Transtorácico",1,3,4),(12,40,"Fetal",1,3,4
),(13,40,"Transesofágico",5,3,4),(14,100,"Tórax",7,1,1),(15,100,
"Crânio",7,1,3),(16,100,"Coluna",6,1,1);

INSERT INTO trabalho.Marcacao_has_Exame (Marcacao_idMarcacao,
Marcacao_Paciente_n°utente, Exame_idExame, sala, data,
hora_inicio, duracao, hora_fim) VALUES (0,460505826,3,"sala
1","2022-05-
26","11:30:00","00:15:00",hora_fim(hora_inicio,duracao)),(1,1775
51020,8,"sala 3","2021-12-

```

30","17:00:00","00:10:00",hora_fim(hora_inicio,duracao)), (2,1252
 85547,4,"sala 2","2021-10-
 13","9:30:00","00:15:00",hora_fim(hora_inicio,duracao)), (3,12528
 5547,1, "sala 1","2022-02-
 10","9:30:00","00:15:00",hora_fim(hora_inicio,duracao)), (4,16143
 2848,2,"sala 4","2022-05-
 03","16:30:00","00:10:00",hora_fim(hora_inicio,duracao)), (5,2186
 37581,5,"sala 4","2022-05-
 10","10:00:00","00:10:00",hora_fim(hora_inicio,duracao)), (6,1825
 52802,6,"sala 4","2021-09-
 27","14:30:00","00:10:00",hora_fim(hora_inicio,duracao)), (7,1825
 52802,9, "sala 5" ,"2022-04-
 14","10:30:00","00:40:00",hora_fim(hora_inicio,duracao)), (8,2572
 78212,7,"sala 4","2021-12-
 17","10:00:00","00:10:00",hora_fim(hora_inicio,duracao)), (9,1582
 48466,10,"sala 5","2022-05-
 23","10:00:00","00:40:00",hora_fim(hora_inicio,duracao)), (10,177
 551020,11,"sala 6","2021-10-
 28","09:00:00","00:30:00",hora_fim(hora_inicio,duracao)), (11,275
 365586,15,"sala 7","2022-03-
 30","11:00:00","01:00:00",hora_fim(hora_inicio,duracao)), (12,134
 330630,7,"sala 4","2022-02-
 11","15:00:00","00:10:00",hora_fim(hora_inicio,duracao)), (13,158
 248466,13,"sala 6","2021-10-
 01","09:30:00","00:30:00",hora_fim(hora_inicio,duracao)), (14,134
 330630,14,"sala 7","2022-03-
 25","14:00:00","01:00:00",hora_fim(hora_inicio,duracao)), (15,271
 480643,4,"sala 2","2021-09-
 22","17:00:00","00:30:00",hora_fim(hora_inicio,duracao)), (16,176
 616010,12,"sala 2","2021-11-
 12","14:30:00","00:30:00",hora_fim(hora_inicio,duracao)), (17,125
 285547,2,"sala 4","2022-03-
 09","16:30:00","00:10:00",hora_fim(hora_inicio,duracao)), (18,460
 505826,8,"sala 3","2022-07-
 13","15:30:00","00:10:00",hora_fim(hora_inicio,duracao)), (19,158
 248466,10,"sala 5","2022-01-
 06","15:00:00","00:30:00",hora_fim(hora_inicio,duracao)), (20,275
 365586,16,"sala 8","2022-01-
 31","9:30:00","01:00:00",hora_fim(hora_inicio,duracao));

Execução de processos associados ao SBD

Para testar a base de dados, procedemos à criação e execução de funções, *views* e *triggers*. Para os implementar foram utilizados códigos SQL, que são os seguintes:

Triggers

```
DELIMITER $$
CREATE TRIGGER verify_exists
before insert on trabalho.Contactos
for each row
BEGIN
    IF (new.email = '' or new.email not like '_%@__%.__%') then
        set new.email = NULL;
    end if;
END$$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER verify_exists
before insert on trabalho.marcacao_has_exame
for each row
begin
    IF ((NEW.sala in (select sala from
trabalho.marcacao_has_exame)) )
    and (NEW.hora_inicio in (select hora_inicio from
trabalho.marcacao_has_exame))
    and (NEW.data in (select data from
trabalho.marcacao_has_exame))
    then signal sqlstate '45000' set message_text = 'Não é
possível marcar nestas condições';
    END IF;
end$$
DELIMITER ;
```

Os triggers verificam se não existem marcações na mesma hora, data e sala, adicionalmente, verificam se a inserção do email é válida.

Funções

```
DELIMITER //
```

```

create function idade(date1 datetime)
returns int
Begin
    return timestampdiff(year,date1,curDate());
End //
DELIMITER;

DELIMITER //
create function hora_fim(hora_inicio time,duracao time)
returns time
Begin
    return addtime(hora_inicio, duracao);
End //
DELIMITER ;

```

As funções criadas foram as da idade e hora_fim, a função idade foi utilizada para calcular a idade respetiva de cada utente assim como os anos de serviço de cada médico. A hora final de cada exame tendo em conta a duração do mesmo foi calculada pela função hora_fim.

Procedures

```

DELIMITER //
CREATE PROCEDURE falecimento ( utente int)
begin
delete from Marcacao where Paciente_n°utente=utente;
delete from Paciente where n°utente=utente;
delete from Contactos where Paciente_n°utente=utente;
delete from Marcacao_has_exame where
Marcacao_Paciente_n°utente=utente;
end //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE cancelado ( Marcacao int)
begin
    delete from Marcacao where idMarcacao=Marcacao;
    delete from Marcacao_has_exame where
Marcacao_idMarcacao=Marcacao;

```

```
end //
DELIMITER ;
```

O procedimento ‘cancelado’ visa eliminar marcações de exames cancelados pelos pacientes quando estes ligam para a clínica a desmarcar. Enquanto o procedimento ‘falecimento’ pretende eliminar todas as informações relativas a um paciente que faleceu.

Views

```
CREATE VIEW paciente_seguro as (select
p.nome,p.data_nascimento,p.s_comparticipacao,p.genero,p.idade
from Paciente p);
```

Relativamente à view, esta cria uma tabela virtual dos pacientes somente com informações não confidenciais dos mesmos que é vista pelos administradores.

Interfaces

Tal como referido anteriormente, em conformidade com a BD estruturada, devem ser implementadas interfaces direcionadas a:

- Pacientes: consta o local de marcação dos exames e visualização de informação relativa aos mesmos.
- Administradores: tal como, médicos e gestores que gerem o agendamento e acedem aos respetivos resultados.

Na interface utilizada pelos pacientes, a página inicial para a marcação de exames imagiológicos remete para o seu login, caso os mesmos não tenham efetuado um pré-registo anteriormente, serão redirecionados para uma página onde irão registar-se. Para tal, terão de preencher os seus dados pessoais, desde o nome, data de nascimento, género, número de utente, NIF, telemóvel, morada, email e sistema de participação. Após o preenchimento dos campos supracitados, o processo de registo do paciente na plataforma de marcação de exames encontra-se finalizado e os seus dados encontram-se associados ao seu registo pessoal. De seguida, o próximo passo no processo de agendamento é referente à marcação do exame propriamente dito, sendo que nesta interface o paciente terá de

preencher os campos em conformidade com a modalidade, o tipo (especialidade) do exame, o médico que pretende que realize o exame, bem como o horário disponível e que mais se adequa ao paciente (conveniente). Neste último parâmetro, o paciente tem à sua disposição um calendário pré-definido para que selecione o dia e a hora disponíveis para o exame. Por fim, na terceira página da interface surge a confirmação dos dados e da marcação do exame em questão no horário selecionado.

Relativamente à interface destinada aos administradores, como é o caso dos médicos, existem 4 páginas que os mesmos têm acesso onde podem realizar pesquisas, editar informação e até mesmo trabalhar. Após *Login* do Médico, em que para tal é necessário colocar o respetivo id e a palavra-passe, é apresentada uma interface que permite pesquisar exames e pacientes.

Na primeira página surge o separador “Agenda” onde é possível a visualização dos exames agendados que um dado médico (o qual fez o login) tem a realizar num determinado dia. No segundo separador, surgem os “Pacientes” onde é possível a pesquisa de um paciente em particular, permitindo, assim, o acesso aos dados pessoais do mesmo. Para tal, é necessário o número de utente ou o seu nome. No seguimento do terceiro separador referente a “Médicos” é exequível a visualização da listagem de todos os médicos que fazem parte da clínica tal qual as respetivas especialidades e dados pessoais. Por fim, na última página alusiva a “Exames” é possível a consulta de todos os exames agendados, podendo ser feita também uma pesquisa específica para um determinado exame, sendo unicamente necessário o id do mesmo. Desta forma, obtém-se informações bastante pertinentes acerca do mesmo, desde a sua modalidade, tipo, o equipamento necessário, o médico e a sala onde se irá realizar, o horário estipulado (e a respetiva duração) e o custo associado.

Na implementação das interfaces, o *Oracle* foi o sistema gerenciador de base de dados (SGBD), representando um dos mecanismos de base de dados relacional mais fidedigno e amplamente utilizado. O sistema é desenvolvido tendo por base uma estrutura de BD relacional, em que os objetos de dados podem ser acedidos diretamente pelos utilizadores ou até mesmo por um *front-end* de um aplicativo, através do SQL que corresponde à linguagem de consulta estruturada.

É de realçar que a arquitetura do *Oracle* se ramifica entre o lógico e o físico, possibilitando uma modulação da estrutura física, a qual pode ser adicionada e alterada sem comprometer a atividade da base de dados, os seus dados ou utilizadores. Desta forma, é exequível a conceção de um sistema robusto, o qual apresenta uma menor probabilidade de falha. Assim sendo, uma das maiores vantagens do *Oracle* é o facto de ser totalmente escalável, na medida em que é utilizado por inúmeras empresas responsáveis pelo gerenciamento e processamento de dados em redes de área ampla e local.

Em suma, este sistema visa “alimentar” o website de *front-end* utilizado pelos pacientes para a marcação de exames imagiológicos, tal como o programa de *back-end* utilizado pelos administradores.

Interface dos utentes: Front-end

Das várias ferramentas e tecnologias que provêm de diferentes linguagens de programação analisadas por especialistas, chegou-se à conclusão de que o conjunto de ferramentas básicas utilizado na criação de um website incluem HTML, CSS e JavaScript. O HTML e CSS não são propriamente linguagens de programação, mas sim componentes responsáveis pela estrutura da página e informações sobre o estilo, constituindo assim o front-end das páginas e aplicações da web.

Com o HTML conseguimos obter a estrutura básica do site que posteriormente é melhorada e modificada por outras tecnologias, como CSS e JavaScript, de modo a colocar neste o conteúdo desejado.

Já quanto ao CSS este permite ao programador especificar quando um determinado estilo entra em vigor. Por outras palavras, o CCS determina como os elementos HTML de um site devem aparecer no front-end da página web.

JavaScript é a linguagem de script onde o tipo de uma variável é interpretado ao mesmo tempo que é executado. É utilizada para a manipulação de DOM (Document Object Model), e permite adicionar às informações existentes caixas de confirmação, frases de alerta e novas identidades, podendo assim criar sites interativos com um comportamento dinâmico.

The screenshot shows the 'Ficha pessoal' form in the 'Marcação exame' section. The form includes the following fields:

- Nome Completo *
- Nº utente *
- NIF *
- Data de nascimento *
- Género *
- Telemóvel *
- Telefone
- Email
- Morada (rua, Código-postal, localidade) *
- Sistema de Participação *

A green asterisk indicates mandatory fields. A 'SEGUINTE' button is located at the bottom right. The breadcrumb trail is: Home > Apoio ao cliente > Marcação exame > Ficha pessoal.

Figura 3 – Representação da Interface Front-end (website onde são marcados os exames pelo paciente)

The screenshot shows the 'Exame' form in the 'Marcação exame' section. The form includes the following fields:

- Modalidade *
- Tipo *
- Médico *
- Data *
- Hora *

A green asterisk indicates mandatory fields. An 'ENVIAR' button is located at the bottom right. The breadcrumb trail is: Home > Apoio ao cliente > Marcação exame > Ficha pessoal > Exame.

Figura 4 – Representação da Interface Front-end (website onde são marcados os exames pelo paciente)

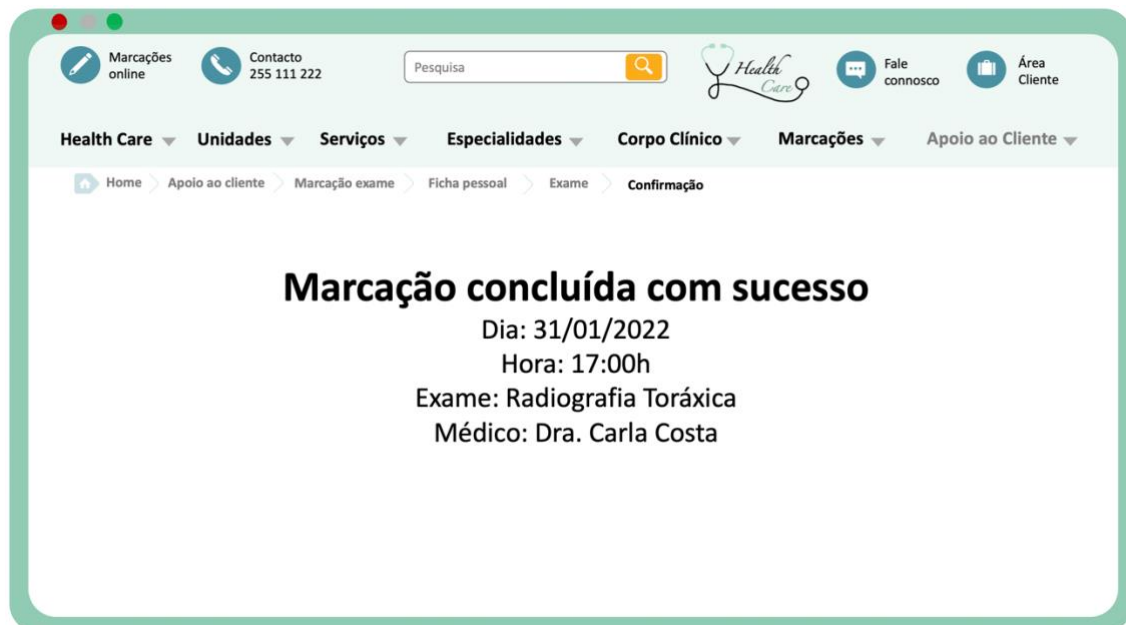


Figura 5 – Representação da Interface Front-end (aviso que aparece no website após concluir com sucesso a marcação do exame)

Interface gestores: Back-end

De modo a criar uma interface para os administradores pudessem organizar as informações de cada exame marcado, poderia optar-se pelo Python.

Uma interface atua como um modelo para projetar classes. Interfaces também definem métodos da mesma forma que classes, mas métodos abstratos, enquanto classe contém métodos não abstratos. Métodos abstratos são aqueles métodos sem implementação ou sem corpo. Portanto, a interface apenas define o método abstrato sem implementação. A implementação desses métodos abstratos é definida por classes que implementam uma interface, o Python poderia ser usado para esta finalidade.

O design da interface Python é diferente de outras linguagens de programação como C++, Java, C# e Go; uma diferença é que todas essas linguagens usam a palavra-chave “interface”, enquanto Python não a usa. Outra diferença é que Python não requer que uma classe que implemente uma interface forneça a definição para todos os métodos abstratos de uma interface.

Os programadores de Python estão familiarizados com várias estruturas da Web que podem ser usadas para criar sites. Django, Grok, WebPy, TurboGears, WebApp2,

Pyramid e Flask são algumas das estruturas Python mais usadas que podem ajudá-lo a criar um site do zero.

Existem principalmente dois frameworks de back-end Python para desenvolvimento web, sendo eles Django e Flask. Para tal, pode-se utilizar qualquer um deles e usar o framework para desenvolvimento web. Flask é um micro framework web de Python, desta forma não necessita de ferramentas ou bibliotecas específicas. É um framework muito simples comparativamente ao Django.

Entre os dois frameworks apresentados para back-end, o mais adequado é o Django, uma vez que é o framework web Python mais usado. Este requer que certos arquivos sejam colocados em certas pastas e certas regras devem ser seguidas ao escrever o código. Ao fazer o desenvolvimento de back-end, é necessário também lidar com base de dados, sendo importante utilizar qualquer uma das tecnologias de base de dados como MySQL.

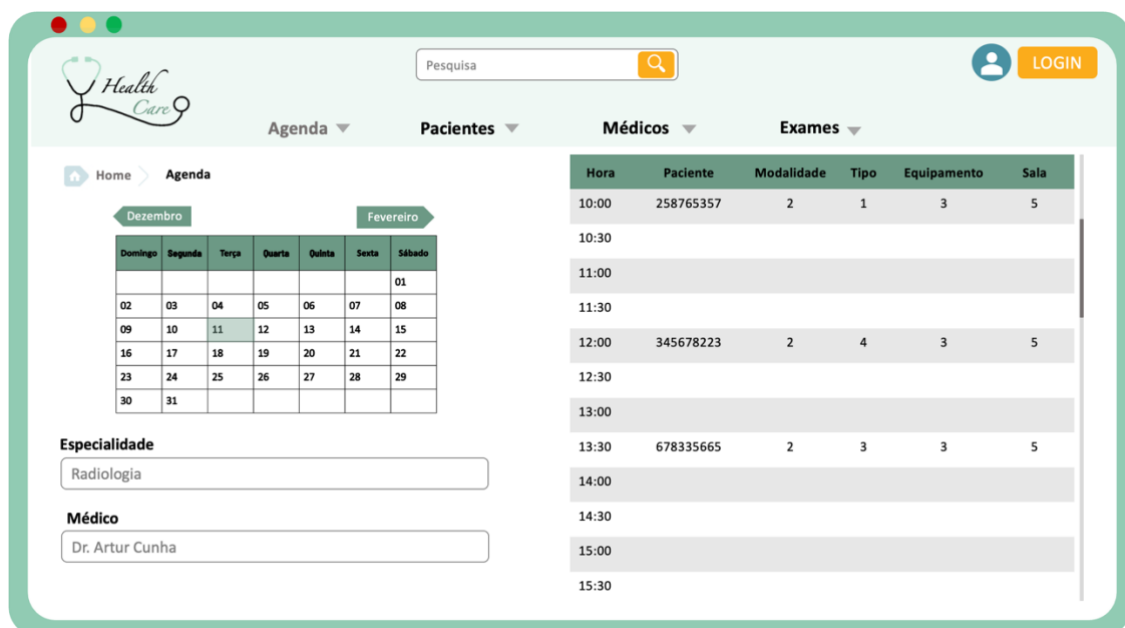


Figura 6 – Representação da Interface Back-end (visualização, através do administrador, da informação sobre a consulta marcada pelos pacientes)

The screenshot shows the 'Pacientes' (Patients) section of the Health Care system. The top navigation bar includes 'Home', 'Pacientes', 'Médicos', and 'Exames'. A search bar is present. The main content area is titled 'Ficha pessoal' (Personal File). On the left, there is a table of patients. On the right, there are form fields for patient details.

Nº Utente	Nome
112565688	António da Silva Ferreira
154533378	Ana Maria Costa
246599557	Constança Augusta Guimarães
351546156	Ricardo Castro de Sousa Amorim
155846222	José Maria Almeida Martins
515451654	Maria Manuela dos Santos
185533322	Ana Carolina Rodrigues
125998645	Joaquim da Silva Pereira
323163545	Filipe Marques Carneiro

Ficha pessoal

Nome completo: António da Silva Ferreira

Nº utente: 112565688 NIF: 257389356

Data nascimento: 18/11/1991 Género: Masculino

Telemóvel: 918874569 Telefone: 253654892

Email: antonio.ferreira@gmail.com

Morada: Rua de Santa Maria, 4000-888, Porto

Sistema de Participação: SNS

Figura 7 – Representação da Interface Back-end (visualização, através do administrador, da informação pessoal do paciente)

The screenshot shows the 'Médicos' (Doctors) section of the Health Care system. The top navigation bar includes 'Home', 'Pacientes', 'Médicos', and 'Exames'. A search bar is present. The main content area is titled 'Ficha pessoal' (Personal File). On the left, there is a table of doctors. On the right, there are form fields for doctor details.

ID	Nome
11641	Manuel Fernandes Pereira
15554	António José Ferreira
25423	Joana Carneiro Alves
15485	Teresa Santos Conde
15765	Daniel Silva Couto
14455	Rodrigo Cruz Falcão
13018	Filipe António Brandão
12564	Ana Rita Oliveira
16853	Ana Ribeiro Sousa

Ficha pessoal

ID: 368733897

Nome completo: Manuel Fernandes Pereira

Especialidade: Cardiologia

NIF: 368733897

Data nascimento: 22/03/1984 Data início serviço: 03/05/2015

Morada: Rua do Souto, 4700-231, Braga

Figura 8 – Representação da Interface Back-end (visualização, através do administrador, da informação pessoal do médico)

The screenshot displays the 'Exames' (Exams) section of the Health Care Back-end Interface. The interface includes a top navigation bar with a search bar, a user profile icon, and a 'LOGIN' button. Below the navigation bar, there are tabs for 'Agenda', 'Pacientes', 'Médicos', and 'Exames'. The 'Exames' tab is active, showing a table of existing exams and a form for adding a new exam.

Exames Table:

ID	Modalidade	Equipamento	Médico
345676	1	3	11641
743875	3	2	24567
279046	2	5	34568
154852	2	5	34568
157655	6	1	45678
144551	4	4	34678
130185	1	3	78456
125641	3	2	67768
168528	2	5	24567

Form for Adding a New Exam:

The form includes the following fields:

- ID:** 368733897
- Tipo:** Abdominal
- Modalidade:** Ecografia
- Equipamento:** Ecógrafo
- Sala:** 2
- Data:** 23/01/2022
- Hora:** 15:00
- Duração:** 00:30
- Preço(€):** 50

Figura 9 – Representação da Interface Back-end (visualização, através do administrador, da informação acerca do exame a realizar pela clínica)

Conclusão

Em virtude do que desenvolvemos neste trabalho, foi-nos possível perceber um pouco a complexidade que está inerente a criação de uma base de dados, sendo que é necessária bastante atenção na identificação das entidades, tendo em conta a realidade, assim como nos seus relacionamentos.

Tendo em mente a criação de um modelo para o controlo e gestão de agendamentos para uma clínica de imagiologia passamos por vários processos sequenciais: desenho do modelo conceptual, do modelo lógico e por fim a passagem ao modelo físico, assim como as respetivas validações. Para testar a base de dados, procedemos então à criação e execução de funções, views e triggers, implementando assim um SBD relacional.

Realizamos também a normalização dos dados de modo a conseguirmos eliminar redundâncias desnecessárias e aumentar assim o desempenho e fluidez da nossa base de dados.

Futuramente seria também interessante tornar este modelo mais complexo, por exemplo adicionando mais atributos às entidades já existentes, assim como adicionar as demais entidades que estejam presentes no processo de realização de um exame no mundo real, incluindo recursos humanos, técnicos e outros recursos associados. Poderia também ser estabelecida uma entidade “agenda” que permitisse ter uma visão global dos exames previstos para determinado dia, para determinado médico ou mesmo para cada equipamento, outro ponto de especial interesse seria criar flags ao invés das procedures que utilizamos no caso de falecimento de um paciente ou cancelamento de um exame, uma vez que os dados não devem ser eliminados permanentemente de uma base de dados mas sim marcados como falecido ou cancelado.

Seria também eminente a necessidade aprofundar a aplicação das formas normais (4ª e 5ª FN), de modo a eliminara problemas de manutenção, diminuir os custos de espaço de armazenamento e também problemas de desempenho.

Referências

- [1] Connolly, T., Begg, C., Database Systems: A Practical Approach to Design, Implementation, and Management, Addison Wesley, 4.ª Edição, 2004
- [2] Techopedia.com. (2019). What is Oracle Database (Oracle DB)? - Definition from Techopedia. [online] Disponível em: <https://www.techopedia.com/definition/8711/oracle-database> [Acedido: 20 Dezembro 2021].
- [3] SAUDE BEM ESTAR (2021). Raio x (RX) [online]. Disponível em: <https://www.saudebemestar.pt/pt/exame/imagiologia/raio-x/> [Acedido: 22 Dezembro 2021].
- [4] VEJA SAUDE (2021). O que é e para que serve a endoscopia? Disponível em: <https://saude.abril.com.br/medicina/o-que-e-e-para-que-serve-a> [Acedido: 21 Dezembro]
- [5] Nibib.nih.gov. (2021). Magnetic Resonance Imaging (MRI). [online] Disponível em: <https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri> [Acedido: 23 Dezembro 2021].
- [6] Radiologiadealbufeira.pt. (2021). TAC - Radiologia de Albufeira. [online] Disponível em: <https://www.radiologiadealbufeira.pt/tac.html> [Acedido: 21 Dec. 2021].
- [7] Kolowich, L. (2019). Web Design 101: How HTML, CSS, and JavaScript Work. [online] Blog.hubspot.com. Disponível em: https://blog.hubspot.com/marketing/web-design-html-css-javascript?fbclid=IwAR0QpSPlhxWBok6SP07golGuM8k7mIIRgtjY1kY_Ok6CmDe6r1J9bvBSUQc [Acedido: 22 Dezembro 2021].
- [8] Pythonista planet (2020) How to Become a Python Back-end Developer [online] Disponível em: <https://pythonistaplanet.com/how-to-become-a-python-backend-developer/> [Acedido: 12 Janeiro]
- [9] EDUCBA (2020) Interface in Python [online] Disponível em: <https://www.educba.com/interface-in-python/> [Acedido: 12 Janeiro]