



Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Letivo de 2023/2024

“DEθ (Deteta)”

Alexandre de Oliveira Monsanto (a104358), Margarida Cunha da Silva (a104357), Maria Inês R. P. Gracias Fernandes (a104522), Pedro Manuel Macedo Rebelo (a104091), Vasco João Timóteo Gonçalves (a104527)

Maio, 2024

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

“DEθ (Deteta)”

Alexandre de Oliveira Monsanto (a104358), Margarida Cunha da Silva (a104357), Maria Inês R. P. Gracias Fernandes (a104522), Pedro Manuel Macedo Rebelo (a104091), Vasco João Timóteo Gonçalves (a104527)

Maio, 2024

Resumo

Este relatório foi realizado no âmbito da Unidade Curricular de Bases de Dados, descrevendo a criação de uma base de dados para uma agência imaginária de detetives, a DE0. Escolhemos implementar esta base de dados por se tratar de uma situação possível e que consegue ser bem fundamentada, como iremos concluir ao longo deste relatório.

A primeira parte do relatório diz respeito à definição do sistema, onde contextualizamos e fundamentamos a implementação da base de dados relacional, analisando também a sua viabilidade para a empresa DE0.

Numa segunda fase, apresentamos os requisitos que por nós foram levantados e classificados como requisitos de descrição, manipulação e controlo.

De seguida, apresentamos o modelo conceitual para a base de dados que queríamos construir, tendo sempre em conta os requisitos levantados previamente. Aqui identificamos e classificamos as entidades e os relacionamentos existentes entre estas. Uma fase fulcral desta fase foi a validação do modelo de dados com o utilizador, pois permitiu-nos avançar para a fase seguinte: a modelação lógica.

Como foi referido, a quarta parte do relatório diz respeito à modelação lógica da base de dados para a DE0. Nesta fase, apresentamos o modelo por nós desenvolvido. Além disso, validamos o mesmo através da normalização, com interrogações do utilizador e com as transações estabelecidas.

O quinto capítulo é referente à implementação física da base de dados. Nesta fase fazemos a tradução das relações base, que foram previamente definidas no modelo lógico, criamos utilizadores e fazemos o povoamento das tabelas. Depois do cálculo do espaço ocupado pela base de dados, da definição de vistas de utilização e da tradução das interrogações do utilizador (colocadas anteriormente) para SQL, procedemos à indexação e à implementação de procedimentos, funções e gatilhos.

Concluímos o projeto com algumas reflexões sobre a criação da base de dados da agência DE0, desafios com que nos deparámos e previsões de como poderá evoluir no futuro.

Área de Aplicação: Desenho e arquitetura de Sistemas de Bases de Dados

Palavras-Chave: Bases de Dados Relacionais, Levantamento de Requisitos, Modelação Conceptual, Modelação Lógica, Implementação Física

Índice

<i>Resumo</i>	<i>i</i>
<i>Índice</i>	<i>ii</i>
<i>Índice de Figuras</i>	<i>iv</i>
<i>Índice de Tabelas</i>	<i>v</i>
1. Definição do Sistema	1
1.1. Contexto de Aplicação	1
1.2. Motivação e Objetivos do Trabalho	1
1.3. Análise da Viabilidade do Processo	2
1.4. Recursos e Equipa de Trabalho	3
1.5. Plano de Execução do Projeto	3
2. Levantamento e Análise de Requisitos	5
2.1. Método de Levantamento e de Análise de Requisitos Adotado	5
2.2. Organização dos Requisitos Levantados	5
2.2.1 Requisitos de Descrição	6
2.2.2 Requisitos de Manipulação	7
2.2.3 Requisitos de Controlo	7
2.3. Análise e Validação Geral dos Requisitos	8
3. Modelação Conceptual	9
3.1. Apresentação da Abordagem de Modelação Realizada	9
3.2. Identificação e Caracterização das Entidades	9
3.3. Identificação e Caracterização dos Relacionamentos	10
3.4. Identificação e Caracterização dos Atributos das Entidades e dos Relacionamentos	10
3.5. Apresentação e Explicação do Diagrama ER Produzido	12

4. Modelação Lógica	13
4.1. Construção e Validação do Modelo de Dados Lógico	13
4.1.1 Entidades Simples	13
4.1.2 Relacionamentos binários de um-para-muitos (1:N)	14
4.1.3 Relacionamentos binários de muitos-para-muitos (N:M)	15
4.2. Apresentação e Explicação do Modelo Lógico Produzido	16
4.3. Normalização de Dados	17
4.4. Validação do Modelo com Interrogações do Utilizador	18
5. Implementação Física	20
5.1. Apresentação e explicação da base de dados implementada	20
5.2. Criação de utilizadores da base de dados	22
5.3. Povoamento da base de dados	23
5.4. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)	24
5.5. Definição e caracterização de vistas de utilização em SQL	27
5.6. Tradução das interrogações do utilizador para SQL	28
5.7. Indexação do Sistema de Dados	29
5.8. Implementação de procedimentos, funções e gatilhos	30
6. Conclusões e Trabalho Futuro	33
Correções efetuadas	35
Referências	36
Lista de Siglas e Acrónimos	37

Índice de Figuras

Figura 1 - Diagrama de Gantt CheckPoint	3
Figura 2 - Diagrama de Gantt Apresentação Final	4
Figura 2 - Diagrama ER do modelo conceptual	12
Figura 3 – Modelo lógico	16
Figura 4 – Criação do <i>schema</i> AgenciaDeteta e utilização do mesmo	20
Figura 5 – Implementação física da tabela ‘Escritório’	20
Figura 6 – Implementação física da tabela ‘Departamento’	21
Figura 7 – Implementação física da tabela ‘Detetive’	21
Figura 8 – Implementação física da tabela ‘Cliente’	21
Figura 9 – Implementação física da tabela ‘Caso’	21
Figura 10 – Implementação física da tabela ‘Pagamento’	22
Figura 11 – Implementação física da tabela ‘Caso_por_Detetive’	22
Figura 13 – Criação do utilizador ‘Cliente’	22
Figura 12 – Exemplo de povoamento por inserção de dados	23
Figura 13 – Povoamento através de um programa em Python	23
Figura 14 – View 1 em SQL	27
Figura 15 – View 2 em SQL	27
Figura 16 – View 3 em SQL	27
Figura 17 – View 4 em SQL	28
Figura 18 – Query 1 em SQL	28
Figura 19 – Query 2 em SQL	28
Figura 20– Query 3 em SQL	29
Figura 21– Query 4 em SQL	29
Figura 22– Criação do índice sobre a coluna telefone	29
Figura 23– Criação do índice sobre a coluna idCaso	30
Figura 24– Procedure 1	30
Figura 25– Função 1	31
Figura 26– Trigger 1	31
Figura 27 – Tabela de pagamento antes do update	32
Figura 28– Update que vai rodar o trigger	32
Figura 29– Tabela de pagamento depois do update	32

Índice de Tabelas

Tabela 1 – Requisitos de descrição	6
Tabela 2 – Requisitos de manipulação	7
Tabela 3 – Requisitos de controlo	8
Tabela 4 – Identificação e caracterização das entidades	9
Tabela 5 – Identificação e caracterização dos relacionamentos	10
Tabela 6 – Identificação e caracterização dos atributos das entidades	11
Tabela 7 – Estimativa do espaço ocupado pela base de dados (inicialmente)	25
Tabela 8 – Estimativa do espaço ocupado pela base de dados (após um ano)	26

1. Definição do Sistema

A agência de detetives DEθ, responsável por resolver numerosos crimes e mistérios, necessita de implementar um sistema de base de dados para facilitar a comunicação e a partilha de informação entre os diversos escritórios da agência.

1.1. Contexto de Aplicação

Em 2004, uma terrível tragédia abalou a vida do génio milionário Megálo Theta. O seu filho, Lígo Theta, foi raptado por terroristas e ordenaram ao pai que pagasse o resgate, ou iriam matá-lo. Como tinha grande parte do seu dinheiro embargado em apostas de casinos e corridas de cavalos, não pôde resgatar o filho de imediato e, como a polícia local não tinha os fundos necessários para acompanhar cada movimento dos terroristas, Megálo viu-se forçado a ser ele a investigar o rapto. Usou todos os recursos que tinha à sua disposição, até que finalmente conseguiu reaver Lígo sem ter de pagar aos terroristas.

Depois desta tragédia, Megálo percebeu que a sua vocação era de facto investigar crimes e mistérios, e começou a fazê-lo frequentemente como favor aos seus amigos (sendo cônjuges infiéis e desvios de dinheiro as principais investigações que lhe pediam para fazer).

Megálo decidiu então abrir um negócio e, 20 anos depois, em 2024, conta com a ajuda de vários detetives na sua agência, a DEθ, da qual é o presidente. Clientes de todo o país procuram-no pela sua elevadíssima taxa de sucesso na resolução de diversos casos e mistérios. Na verdade, o negócio tem corrido tão bem que ele está a pensar abrir mais escritórios para a sua agência!

1.2. Motivação e Objetivos do Trabalho

Como a DEθ começou por ser um pequeno negócio e, até agora, tinha apenas um escritório em serviço, a comunicação entre os vários funcionários, departamentos e clientes sempre foi bastante analógica, estando todos os registos internos e arquivos dos casos investigados nos últimos 20 anos armazenados em papel. Agora que Megálo se quer expandir, todos os setores dos vários escritórios devem estar na mesma página sobre os funcionários que empregam, os clientes que recebem e os casos que aceitam, para que nenhuma informação importante se perca.

O senhor Theta decidiu que já está mais que na hora de se modernizar. Como não percebe muito de tecnologias, pediu ao seu filho Lígo (que, depois de muita terapia, ultrapassou o trauma e decidiu seguir um percurso na área da Engenharia Informática) que arranjasse uma solução para o seu problema.

Lígo reuniu uma equipa, composta pelos seus amigos de curso, e decidiu implementar um sistema que permita ao pai gerir os vários escritórios da sua agência. Megálo poderá ver quais os detetives empregados por cada escritório e a sua função, assim como os casos a que cada cliente está associado e os detetives que os vão receber e tentar solucionar. O seu objetivo é facilitar a comunicação entre os diversos escritórios, armazenando e relacionando todas as informações importantes sobre os diferentes casos, funcionários e clientes numa base de dados a que todos os setores tenham acesso em qualquer um dos escritórios.

1.3. Análise da Viabilidade do Processo

A agência DEθ está comprometida em ser uma força positiva tanto para os seus clientes como para a sociedade em geral. No entanto, é, acima de tudo, uma empresa, e, por isso, todos os projetos são cuidadosamente avaliados e justificados para garantir que contribuem para o sucesso e futuro da agência, mantendo ao mesmo tempo o seu compromisso com a responsabilidade social. É por isso que a criação de uma base de dados é uma iniciativa extremamente viável e crucial para a eficiência operacional e o crescimento sustentável da agência.

Uma base de dados centralizada permitirá que todas as informações sobre detetives, clientes e casos sejam armazenadas num local acessível a todos os escritórios da agência. Isto elimina a necessidade de comunicação analógica e garante que todos os departamentos estejam na mesma página em tempo real.

Ao permitir o acesso instantâneo a informações relevantes, como detalhes de casos e atribuições de detetives, a base de dados facilitará a comunicação entre os vários escritórios e isto irá reduzir a possibilidade de perda ou má interpretação de várias informações importantes. Para além disso, isto vai economizar tempo e recursos, permitindo assim que as equipas se concentrem mais na resolução de casos e no atendimento ao cliente.

A digitalização dos dados irá aumentar a segurança e facilitará a proteção das informações confidenciais e sensíveis dos clientes e dos casos.

Analizando mais detalhadamente, podemos concluir que a implementação de um sistema de base de dados é não só viável, mas também essencial para o sucesso contínuo e expansão desta agência. Com uma abordagem cuidada e devidamente planeada, a transição para um sistema digital proporcionará benefícios significativos em termos de eficiência operacional, comunicação e prestação de serviços aos clientes.

1.4. Recursos e Equipa de Trabalho

Para estabelecer e operar a base de dados na agência, será fundamental contar com uma equipa diversificada e recursos materiais adequados. Em termos de recursos humanos, a equipa de desenvolvimento, amigos de curso de Ligo, desempenhará um papel crucial, composta por engenheiros de software e especialistas em bases de dados encarregados de projetar, desenvolver e manter o sistema. O presidente, Megálo Theta, supervisionará o projeto, garantindo que as decisões estratégicas estejam alinhadas com os objetivos da empresa. Além disso, a equipa operacional da agência, composta por detetives e funcionários administrativos, utilizará ativamente a base de dados nas suas atividades diárias e precisará de treino adequado para usar o sistema de forma eficaz. Pelo menos alguns membros da equipa também estarão envolvidos para garantir a manutenção básica da base de dados, como *backups* e resolução de problemas simples.

Em termos de recursos materiais, será necessário um servidor, podendo ser físico, localizado nas instalações da agência, ou na *cloud*, que tenha os recursos necessários para hospedar a base de dados. O *MySQL* será utilizado em termos de Sistema de Gestão de Base de Dados (SGDB), oferecendo recursos abrangentes para gerenciar e manipular dados de forma eficiente. Ferramentas de modelagem como *MySQLWorkbench* e *TerraER* serão utilizadas para criar modelos de dados e esquemas que servirão como base para a estrutura da base de dados.

1.5. Plano de Execução do Projeto

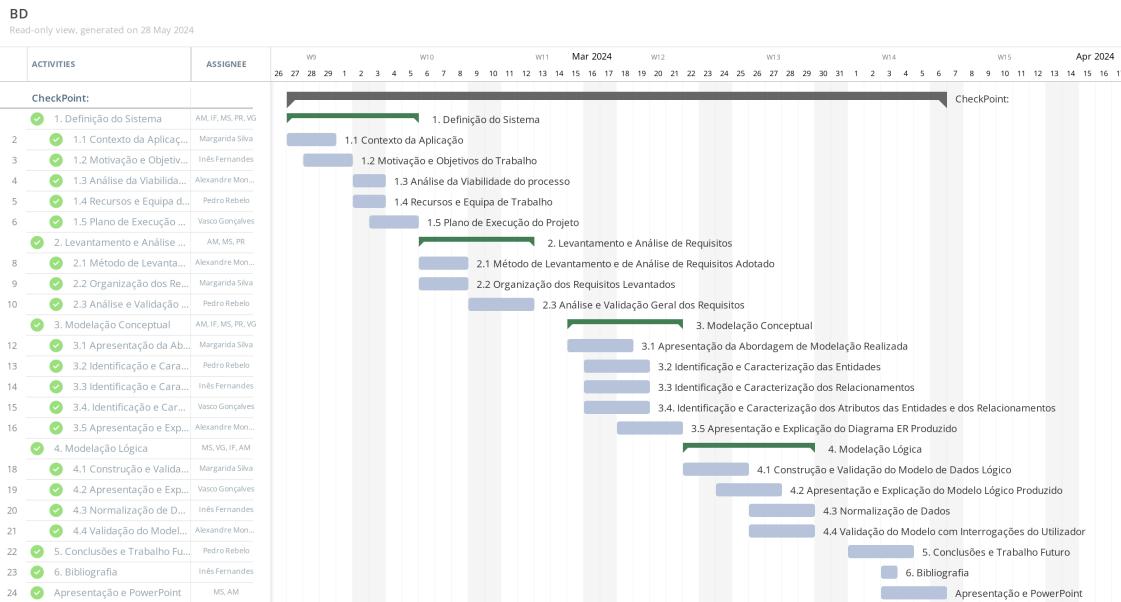


Figura 1 - Diagrama de Gantt CheckPoint

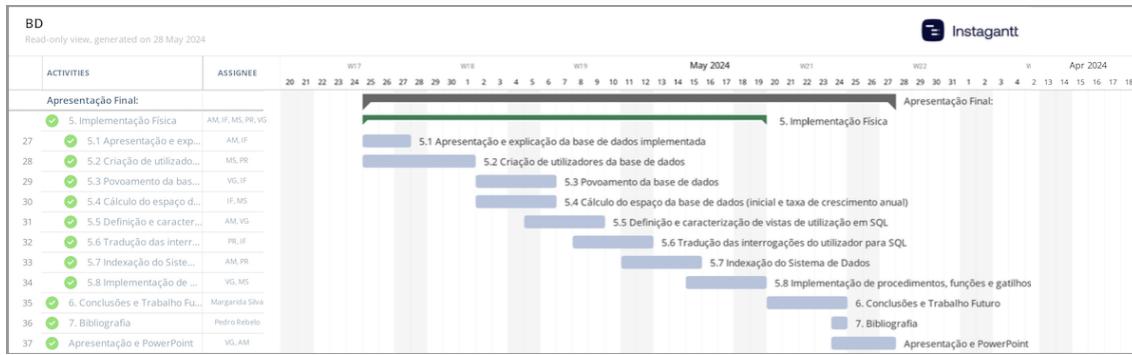


Figura 2 - Diagrama de Gantt Apresentação Final

Após uma breve análise do Diagrama de Gantt, é evidente o impacto positivo que teve na organização e na qualidade do trabalho. A estruturação clara das tarefas proporcionou uma melhor gestão do tempo e dos recursos, contribuindo significativamente para o progresso do projeto. No entanto, podemos comprovar que houve intervalos de tempo entre as tarefas devido a diversos motivos, durante a qual optamos por não avançar com determinadas atividades. Apesar desses espaços livres, comprovamos também que os mesmos não foram obstáculos à conclusão do projeto, pois conseguimos manter uma organização coerente. É importante salientar que, embora tenhamos enfrentado alguns contratemplos que levaram a desvios em relação ao Diagrama de Gantt inicialmente estabelecido, esses incidentes não comprometeram o funcionamento eficaz deste projeto.

Além disso, durante o processo, foi necessário realizar consultas extensivas com todos os funcionários para recolher e analisar os requisitos. Devido à complexidade dos dados e à necessidade de garantir uma análise e compreensão abrangente das necessidades de cada setor, previmos que essa etapa consumiria um tempo adicional.

Essa análise detalhada e a comunicação extensiva com os membros da equipa foram essenciais para garantir que o projeto avançasse de maneira eficiente. A flexibilidade demonstrada diante dos desafios encontrados e a capacidade de ajustar o plano conforme necessário foram cruciais para o sucesso geral deste projeto.

2. Levantamento e Análise de Requisitos

2.1. Método de Levantamento e de Análise de Requisitos Adotado

No desenvolvimento de sistemas de informação, a fase de levantamento de requisitos desempenha um papel crucial, pois é durante esse processo que se obtém uma compreensão clara das necessidades e expectativas desta agência. Neste contexto, foram adotadas diversas estratégias de levantamento de requisitos para garantir uma abordagem abrangente e eficaz na conceção do sistema de base de dados da DE0.

Uma das estratégias utilizadas foi a realização de reuniões diretas com Megálo. Estas reuniões foram agendadas regularmente, permitindo uma discussão detalhada sobre os processos e operações do negócio. Durante esses encontros, foram abordadas questões específicas relacionadas ao gerenciamento da agência, dos casos, interações com clientes, dos serviços, dos departamentos e dos pagamentos. A presença do próprio Megálo foi fundamental, pois ele detinha um conhecimento profundo das operações diárias e das necessidades do negócio.

Além das reuniões com Megálo, foram realizadas entrevistas individuais com outros membros da equipa da agência, como detetives, apenas para que pudessem confirmar as afirmações de Megálo e averiguar se existiam mais requisitos. A inclusão desses diversos intervenientes garantiu que todos os pontos de vista fossem considerados no processo de levantamento de requisitos.

2.2. Organização dos Requisitos Levantados

Através da análise da informação recolhida, procedemos à definição dos requisitos que a base de dados deverá suportar. Procuramos dar uma maior estruturação à informação obtida, definindo o papel de cada uma das componentes que formam o negócio da agência.

Como o objetivo da base de dados não é acompanhar o processo de investigação de cada caso (recolha de provas, testemunhas, etc.), mas sim a gestão dos vários escritórios (e dos seus detetives, casos e clientes), durante o levantamento de requisitos, a única vista de utilização que se identificou foi a do **Escritório**.

2.2.1 Requisitos de Descrição

Número	Data/Hora	Requisito	Fonte	Analista
RD1	04/03/2024 15:00	Existem escritórios	Megálo Theta	Lígo Theta
RD2	04/03/2024 15:10	Cada escritório é definido por um identificador (numérico), nome, morada (constituída por código postal, rua, localidade e porta) e contacto telefónico	Megálo Theta	Lígo Theta
RD3	04/03/2024 15:11	Existem departamentos	Megálo Theta	Lígo Theta
RD4	04/03/2024 15:12	Cada escritório tem vários departamentos	Megálo Theta	Lígo Theta
RD5	04/03/2024 15:14	O departamento é descrito por um id (numérico) e um tipo (homicídios, roubos ou sequestros)	Megálo Theta	Lígo Theta
RD6	04/03/2024 15:15	Existem detetives	Megálo Theta	Lígo Theta
RD7	04/03/2024 15:16	Cada departamento é constituído por vários detetives	Megálo Theta	Lígo Theta
RD8	04/03/2024 15:20	Cada detetive só pode trabalhar num departamento	Megálo Theta	Lígo Theta
RD9	04/03/2024 15:23	O detetive é descrito por id (nº mecanográfico), nome, morada (constituída por código postal, rua, localidade e porta), contacto telefónico e data de nascimento e um estado (de atividade)	Megálo Theta	Lígo Theta
RD10	04/03/2024 15:28	Existem casos	Megálo Theta	Lígo Theta
RD11	04/03/2024 15:34	Os detetives podem investigar vários casos de uma vez	Megálo Theta	Lígo Theta
RD12	04/03/2024 15:35	Cada caso é definido por um id (numérico) e por uma data de inicio da investigação. Pode ou não ter data de fim da investigação	Megálo Theta	Lígo Theta
RD13	04/03/2024 15:36	Existem clientes	Megálo Theta	Lígo Theta
RD14	04/03/2024 15:37	Cada caso está associado a um cliente	Megálo Theta	Lígo Theta
RD15	04/03/2024 00:30	Cada cliente pode estar associado a vários casos	Megálo Theta	Lígo Theta
RD16	05/03/2024 18:14	O cliente é definido por um id (Número de Identificação Fiscal), nome, morada (constituída por código postal, rua, localidade e porta) e contacto telefónico	Megálo Theta	Lígo Theta
RD17	06/03/2024 16:00	Existem pagamentos	Megálo Theta	Lígo Theta
RD18	06/03/2024 16:21	Cada caso tem pelo menos um pagamento associado.	Megálo Theta	Lígo Theta
RD19	06/03/2024 16:27	Cada pagamento tem um id (numérico), um valor, um estado, uma data e o método em que foi realizado (numerário, cartão, MBWay ou transferência)	Megálo Theta	Lígo Theta
RD20	06/03/2024 17:32	O valor de cada pagamento é fixo, para qualquer caso que seja investigado	Megálo Theta	Lígo Theta
RD21	06/03/2024 17:44	Um caso que não seja solucionado não terá segundo pagamento, mas terá sempre o depósito inicial	Megálo Theta	Lígo Theta

Tabela 1 – Requisitos de descrição

2.2.2 Requisitos de Manipulação

Número	Data/Hora	Requisito	Fonte	Analista
RM1	04/03/2024 16:29	Deve ser possível consultar as receitas por escritório, todos os meses	Megálo Theta	Lígo Theta
RM2	04/03/2024 17:33	Deve ser possível verificar a taxa de sucesso da resolução dos casos de cada escritório	Megálo Theta	Lígo Theta
RM3	04/03/2024 17:45	Deve ser possível ver quais os detetives disponíveis para aceitar novos casos	Megálo Theta	Lígo Theta
RM4	04/03/2024 18:00	Todos os dias deve ser emitida uma lista com os casos em investigação.	Megálo Theta	Lígo Theta
RM5	04/03/2024 18:13	Deve ser possível consultar a taxa de sucesso de cada detetive na resolução de casos	Megálo Theta	Lígo Theta
RM6	04/03/2024 18:21	Deve ser possível eleger o detetive do mês, com base na taxa de sucesso de cada um na resolução dos casos desse mês	Megálo Theta	Lígo Theta
RM7	05/03/2024 14:12	Pode ser eleito um detetive do mês para cada um dos departamentos	Megálo Theta	Lígo Theta
RM8	05/03/2024 14:23	Deve ser possível consultar se os casos se encontram abertos (em investigação) ou fechados (solucionados)	Megálo Theta	Lígo Theta
RM9	05/03/2024 14:51	Pode ser possível fechar um caso se não tiver solução (adicionar uma data de fim da investigação)	Megálo Theta	Lígo Theta
RM10	05/03/2024 14:55	Quando o caso é aceite pelo escritório, o cliente tem de fazer um depósito inicial de pagamento	Megálo Theta	Lígo Theta
RM11	06/03/2024 10:25	Quando o caso é solucionado, tem de ser realizado um novo pagamento	Megálo Theta	Lígo Theta
RM12	06/03/2024 14:12	Deve ser possível obter a lista dos detetives de cada escritório	Megálo Theta	Lígo Theta
RM13	06/03/2024 14:31	Deve ser possível obter a lista de detetives que trabalham num determinado departamento	Megálo Theta	Lígo Theta
RM14	06/03/2024 17:23	Deve ser possível obter a lista de todos os clientes da agência	Megálo Theta	Lígo Theta
RM15	06/03/2024 17:57	Deve ser possível aceder ao histórico de cada cliente na agência	Megálo Theta	Lígo Theta
RM16	07/03/2024 09:35	Deve ser possível obter a lista dos clientes com pagamento em atraso	Megálo Theta	Lígo Theta

Tabela 2 – Requisitos de manipulação

2.2.3 Requisitos de Controlo

Número	Data/Hora	Requisito	Fonte	Analista
RC1	08/03/2024 15:35	Nenhum cliente deve conseguir aceder aos dados pessoais de outros clientes	Megálo Theta	Lígo Theta
RC2	08/03/2024 16:32	Nenhum cliente deve conseguir aceder aos dados pessoais dos detetives	Megálo Theta	Lígo Theta
RC3	08/03/2024 16:39	Nenhum cliente deve conseguir aceder às informações dos casos de outros clientes	Megálo Theta	Lígo Theta
RC4	08/03/2024 22:22	Um cliente só pode aceder às informações dos casos associados a si se estes se encontrarem fechados ou solucionados	Megálo Theta	Lígo Theta
RC5	09/03/2024 03:33	Os detetives não podem aceder aos dados pessoais de outros detetives	Megálo Theta	Lígo Theta
RC6	09/03/2024 11:11	Os detetives podem aceder às informações de todos os casos (abertos ou fechados) de todos os escritórios	Megálo Theta	Lígo Theta

Tabela 3 – Requisitos de controlo

2.3. Análise e Validação Geral dos Requisitos

Após a fase de levantamento de requisitos, é importante proceder à sua validação, a fim de assegurar a sua precisão, relevância e viabilidade.

Nesse contexto, foi realizada uma nova reunião com Megálo e com mais alguns detetives da agência, em que se procedeu à análise minuciosa dos requisitos registados, avaliando se estes eram claros, comprehensíveis e livres de contradições ou ambiguidades. Como não foram identificadas quaisquer falhas ou problemas nos requisitos estabelecidos, não foi necessário realizar alterações ou ajustes após a reunião.

A validação bem-sucedida dos requisitos proporciona uma base sólida para o desenvolvimento do sistema de base de dados, garantindo que é desenvolvido de acordo com as expectativas e objetivos do Senhor Theta.

3. Modelação Conceptual

Concluída a fase de levantamento de requisitos e aprovados os seus resultados, já detalhadamente analisados, passamos à próxima etapa da criação do sistema de base de dados – a sua modelação conceptual.

3.1. Apresentação da Abordagem de Modelação Realizada

O modelo conceptual é independente dos detalhes de implementação e procura representar de forma fidedigna o modelo de informação pretendido pela DE0. Para facilitar a definição e interpretação do modelo, este é acompanhado de documentação, composta por uma explicação sobre as entidades, relacionamentos e atributos identificados, assim como por um diagrama ER.

3.2. Identificação e Caracterização das Entidades

Para a realização de um modelo conceptual, é necessário definir as várias entidades existentes, recorrendo para tal aos vários requisitos levantados. A primeira entidade identificada foi o **Escritório**, uma entidade essencial e básica na implementação de uma base de dados sobre os próprios escritórios da agência. Podemos identificar também a entidade **Departamento**, outra entidade essencial que permitirá separar os serviços por tipo. A partir daí, podemos identificar a entidade **Detetive**, que irá estar associada à investigação de um **Caso**. Por sua vez, esta última associa-se a **Pagamento** e a **Cliente**.

Requisito	Entidade	Descrição	Ocorrência
RD1	Escritório	Espaços físicos onde se desenrolam os negócios da agência.	Existem vários espaços físicos onde existem vários departamentos.
RD3	Departamento	Secção em que trabalham os detetives (por exemplo: homicídios, roubos, sequestros, ...).	Existem várias secções onde trabalham vários detetives.
RD6	Detetive	Trabalhadores da empresa.	Existem vários trabalhadores que vão investigar vários casos.
RD10	Caso	Ocorrência ou problema suspeito que precisa de ser investigado e solucionado.	Existem vários casos que vão ter clientes e pagamentos associados.
RD13	Cliente	Utente dos serviços da agência.	Existem vários clientes associados aos casos.
RD17	Pagamento	Retribuição financeira.	Existem pagamentos associados aos casos.

Tabela 4 – Identificação e caracterização das entidades

3.3. Identificação e Caracterização dos Relacionamentos

O passo seguinte do processo de construção do modelo conceptual é a identificação dos relacionamentos entre as entidades estabelecidas, tendo por base os requisitos que destacámos anteriormente.

Requisito	Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
RD4	Escritório	1..1	Tem	1..N	Departamento
RD7, RD8	Departamento	1..1	constituído por	0..N	Detetive
RD11	Detetive	1..N	Investiga	0..N	Caso
RD14, RD15	Caso	1..N	associado a	1..1	Cliente
RD18	Caso	1..1	Tem	1..N	Pagamento

Tabela 5 – Identificação e caracterização dos relacionamentos

3.4. Identificação e Caracterização dos Atributos das Entidades e dos Relacionamentos

Depois de se identificar as entidades e os seus relacionamentos, é necessário identificar as informações referentes a cada um. Estes atributos são facilmente identificados analisando os requisitos levantados. Depois dessa análise, verificou-se que os relacionamentos identificados não possuem atributos, apenas as entidades.

Requisito	Entidade	Atributos	Descrição	Tipo e Tamanho dos Dados	Null (S/N)	Tipo de Atributo
RD2	Escritório	idEscritorio	Identificador do escritório	INT	N	Chave primária
RD2		Nome	Nome do escritório	VARCHAR(45)	N	Simples
RD2		telefone	Contacto telefónico do escritório	VARCHAR(14)	N	Simples
RD2		morada cod_postal rua localidade porta	Localização do escritório Código postal do escritório Rua do escritório Localidade do escritório Número da porta do escritório	INT VARCHAR(100) VARCHAR(45) INT	N N N N	Composto
RD5		idDepartamento	Identificador do departamento	INT	N	Chave primária
RD5	Departamento	tipo	Tipo do departamento	ENUM('Homicídios', 'Roubos', 'Sequestros')	N	Simples
RD9		idDetetive	Número mecanográfico	INT	N	Chave-primária
RD9	Detetive	nome	Nome do detetive	VARCHAR(100)	N	Simples
RD9		telefone	Contacto telefónico do detetive	VARCHAR(14)	N	Simples
RD9		dta_nascimento	Data de nascimento do detetive	DATE	N	Simples

Requisito	Entidade	Atributos	Descrição	Tipo e Tamanho dos Dados	Null (S/N)	Tipo de Atributo
RD9		morada cod_postal rua localidade porta	Localização do escritório Código postal do detetive Rua do detetive Localidade do detetive Número da porta do detetive	INT VARCHAR(100) VARCHAR(45) INT	N N N N	Composto
RD9		estado	Atividade do detetive	ENUM('No Ativo', 'Reformado')	N	Simples
RD16		idCliente	Número de Identificação Fiscal do cliente	INT	N	Chave primária
RD16		nome	Nome do cliente	VARCHAR(100)	N	Simples
RD16	Cliente	telefone	Contacto telefónico do cliente	VARCHAR(14)	N	Simples
RD16		morada cod_postal rua localidade porta	Localização do escritório Código postal do escritório Rua do escritório Localidade do escritório Número da porta do escritório	INT VARCHAR(100) VARCHAR(45) INT	N N N N	Composto
RD12		idCaso	Identificador do caso	INT	N	Chave primária
RD12	Caso	data_inicio	Data de início da investigação de um caso	DATE	N	Simples
RD12		data_fim	Data de fim da investigação de um caso	DATE	S	Simples
RD19		idPagamento	Identificador do pagamento	INT	N	Chave primária
RD19		valor	Valor do pagamento	INT	N	Simples
RD19	Pagamento	estado	Estado do pagamento	ENUM('Pago', 'Não Pago')	N	Simples
RD19		data_pagamento	Data em que o pagamento é efetuado	DATE	N	Simples
RD19		metodo	Método de pagamento usado	ENUM('Numerário', 'Cartão', 'MBWay', 'Transferência')	N	Simples

Tabela 6 – Identificação e caracterização dos atributos das entidades

É de justificar a escolha do tipo e tamanho dos dados do atributo “telefone” (tanto em Escritório, como em Detetive e Cliente). Este é caracterizado como VARCHAR(14) para permitir à base de dados registar números de telefone estrangeiros. A título de exemplo, um número português será registado com o indicativo 00351 e os 9 dígitos que o constituem. O tipo VARCHAR garante que os dois primeiros zeros não são eliminados pelo sistema, o que aconteceria se os dados de “telefone” fossem caracterizados com, por exemplo, INT.

3.5. Apresentação e Explicação do Diagrama ER Produzido

Depois da identificação e caracterização das entidades, dos atributos e dos relacionamentos estabelecidos, foi possível construir um diagrama, usando as funcionalidades disponibilizadas pelo software *TerraER*.

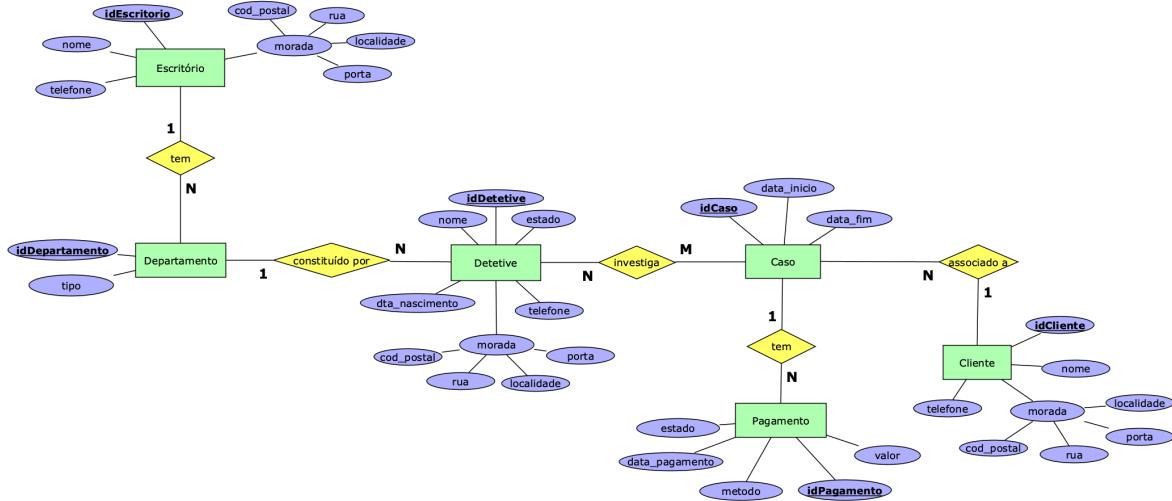


Figura 3 - Diagrama ER do modelo conceptual

Analisando o Diagrama ER apresentado, verifica-se que os relacionamentos, as entidades e os atributos que lhes correspondem se encontram em conformidade com os requisitos enunciados no Capítulo 2 deste documento. A justificação para a sua escolha está registada nos subcapítulos anteriores.

Resta, então, justificar a escolha do atributo chave primária de cada entidade:

- Nas entidades **Departamento**, **Pagamento** e **Caso**, apenas o respetivo número identificador serve como chave candidata, sendo automaticamente definido como chave primária.
- Nas entidades **Escritório**, **Cliente** e **Detetive**, tem-se como chaves candidatas o seu número identificador e o número de telefone. Uma vez que o número de telefone pode ser alterado com o passar do tempo ou ao mudar de operadora, define-se o seu identificador como chave primária, uma vez que esse número nunca se vai alterar.

4. Modelação Lógica

Após a construção do modelo conceptual, prosseguimos à sua modelação lógica.

4.1. Construção e Validação do Modelo de Dados Lógico

De forma a identificarmos e estabelecermos as relações necessárias para que o modelo lógico fosse capaz de representar as entidades, relacionamentos e atributos necessários, analisamos o nosso modelo conceptual por fases de forma a fazermos a transição de uma modelação para a outra.

O processo de derivação passa por descrever como as relações são derivadas para as estruturas que podem ocorrer num modelo de dados conceptual. No nosso modelo, tendo em conta a identificação e caracterização das entidades e relacionamentos que fizemos no capítulo 3, identificaram-se as seguintes estruturas:

1. Entidades Simples
2. Relacionamentos binários de um-para-muitos (1:N)
3. Relacionamentos binários de muitos-para-muitos (N:M)

4.1.1 Entidades Simples

Todas as entidades que considerámos no modelo conceptual vão dar origem a uma relação/tabela com todos os atributos da entidade. Os atributos compostos, como é o caso da morada, passam pela inclusão dos atributos simples relacionados.

Escritório(idEscritorio, nome, telefone, cod_postal, rua, localidade, porta)
chave_primária idEscritorio

Departamento(idDepartamento, tipo, escritorio)
chave_primária idDepartamento
chave_estrangeira escritorio referencia Escritório(idEscritorio)

Detetive (idDetetive, nome, telefone, dta_nascimento, cod_postal, rua, localidade, porta, estado, departamento)
chave_primária idDetetive
chave_estrangeira departamento referencia Departamento(idDepartamento)

Cliente (idCliente, nome, telefone, cod_postal, rua, localidade, porta)
chave_primária idCliente

Caso (idCaso, data_inicio, data_fim, cliente)

chave_primária idCaso

chave_estrangeira cliente referencia Cliente(idCliente)

Pagamento (idPagamento, valor, estado, data_pagamento, metodo, caso)

chave_primária idPagamento

chave_estrangeira caso referencia Caso(idCaso)

4.1.2 Relacionamentos binários de um-para-muitos (1:N)

Para cada relacionamento binário 1:N, a entidade do lado 'um' do relacionamento é designada como a entidade pai e a entidade do lado 'muitos' é designada como a entidade filho.

Para representar esse relacionamento, cria-se uma cópia do atributo chave primária da entidade pai na relação que representa a entidade filho, para atuar como chave estrangeira.

Como documentado no subcapítulo 3.3, o nosso modelo apresenta diversos relacionamentos deste tipo: Escritório-Departamento, Departamento-Detetive, Cliente-Caso e Caso-Pagamento.

- **Escritório-Departamento**

Departamento (idDepartamento, tipo, escritorio)

chave_primária idDepartamento

chave_estrangeira escritorio referencia Escritório(idEscritorio)

A utilização de idEscritorio como chave estrangeira de Departamento cria uma relação Escritório-Departamento de 1 para N (um escritório pode estar associado a vários departamentos).

- **Departamento-Detetive**

Detetive (idDetetive, nome, telefone, dta_nascimento, cod_postal, rua, localidade, porta, estado, departamento)

chave_primária idDetetive

chave_estrangeira departamento referencia Departamento(idDepartamento)

A utilização de idDepartamento como chave estrangeira de Detetive cria uma relação Departamento-Detetive de 1 para N (um departamento pode estar associado a vários detetives).

- **Cliente-Caso**

Caso (idCaso, data_inicio, data_fim, cliente)

chave_primária idCaso

chave_estrangeira cliente referencia Cliente(idCliente)

A utilização de idCliente como chave estrangeira de Caso cria uma relação Cliente-Caso de 1 para N (um cliente pode estar associado a vários casos).

- **Caso-Pagamento**

Pagamento (idPagamento, valor, estado, data_pagamento, metodo, caso)

chave_primária idPagamento

chave_estrangeira caso referencia Caso(idCaso)

A utilização de idCaso como chave estrangeira de Pagamento cria uma relação Caso-Pagamento de 1 para N (um caso pode estar associado a vários pagamentos).

4.1.3 Relacionamentos binários de muitos-para-muitos (N:M)

Para representar este tipo de relacionamento logicamente, deve existir uma tabela auxiliar que cria uma cópia dos atributos chave primária das entidades que participam no relacionamento, para atuarem como chaves estrangeiras.

Como documentado no subcapítulo 3.3, o nosso modelo apresenta apenas um relacionamento deste tipo: Detetive-Caso. Para o traduzir, criou-se a relação Caso_por_Detetive, cuja chave primária é uma chave composta pelas chaves estrangeiras de Detetive e de Caso.

- **Detetive-Caso**

Caso_por_Detetive (detetive, caso)

chave_primária (detetive, caso)

chave_estrangeira detetive referencia Detetive(idDetetive)

chave_estrangeira caso referencia Caso(idCaso)

A utilização de idDetetive e idCaso como chaves estrangeiras de Caso_por_Detetive cria uma relação de N para M (um detetive pode estar associado a vários casos e vice-versa). A chave primária desta relação é uma chave composta pelas chaves estrangeiras de Detetive e de Caso.

Depois da identificação das várias estruturas, pode passar-se à construção do modelo no software MySQLWorkbench.

4.2. Apresentação e Explicação do Modelo Lógico Produzido

Após a derivação das várias entidades e relacionamentos para o modelo lógico, procede-se à criação das tabelas no software MySQLWorkbench.

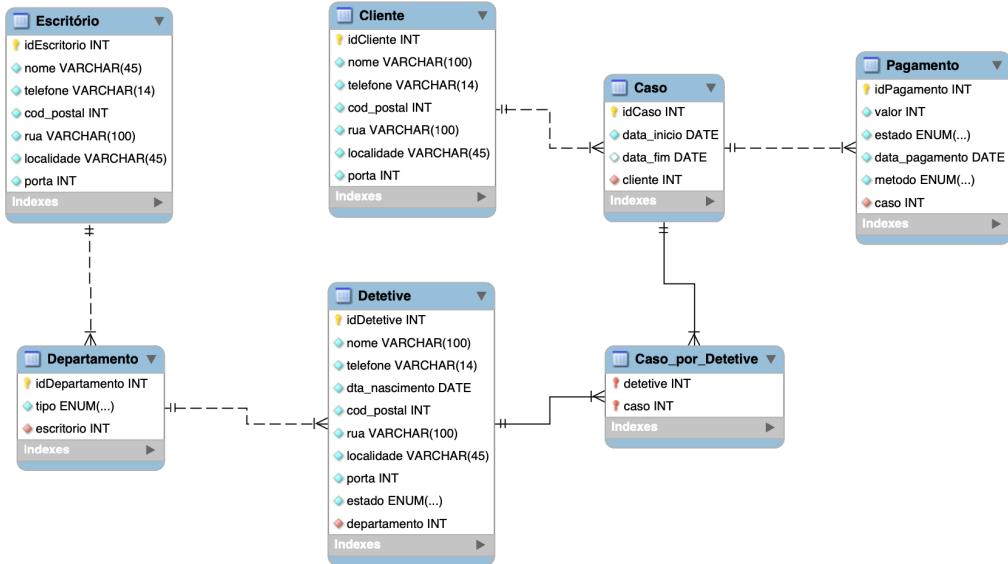


Figura 4 – Modelo lógico

As tabelas Escritório, Departamento, Detetive, Cliente, Caso e Pagamento representam as várias entidades simples que se identificaram e derivaram no subcapítulo 4.1.1.

As chaves amarelas representam a chave-primária de cada uma das tabelas.

Cada losango azul representa os atributos de cada relação, que se caracterizaram no subcapítulo 3.4. O atributo “data_fim” foi o único caracterizado como podendo ser nulo (um caso em investigação não possui uma data de fim da mesma), por isso o seu losango respetivo encontra-se sem preenchimento.

Os relacionamentos 1:N, derivados no subcapítulo 4.1.2, representam-se através dos losangos vermelhos. Cada um destes é uma chave estrangeira. A chave primária correspondente encontra-se na tabela da entidade pai, que se “liga” à tabela da entidade filho através de uma linha tracejada.

A tabela Caso_por_Detetive representa o relacionamento N:M entre Caso e Detetive, especificado no subcapítulo 4.1.3. Esta vai fazer corresponder cada caso aos detetives que o investigam, e vice-versa, usando os seus respetivos identificadores (chaves primárias) para o fazer. As chaves estrangeiras desta relação são também as suas chaves primárias, por isso representam-se com uma chave vermelha. Esta entidade filho “liga-se” às entidades pai através de uma linha contínua.

4.3. Normalização de Dados

A normalização de um modelo relacional é muito importante uma vez que garante que as relações tenham um número mínimo, mas necessário, de atributos, de modo a que todos os requisitos de dados da agência sejam suportados. Além disso, permite que a redundância de dados seja a mínima possível.

Tendo isto em mente, procedemos à verificação do modelo através da normalização, ou seja, verificámos se o modelo estava de acordo com as Formas Normais. Normalmente, a normalização até a Terceira Forma Normal (3FN) é suficiente para garantir a ausência de redundância nos dados.

- **1^a Forma Normal (1FN)**

Uma relação está na 1FN se todos os valores dos seus atributos forem atómicos, não contendo grupos de dados repetidos. Na prática, pode dizer-se que uma relação na qual as interseções entre colunas (atributos) e linhas (registos) apenas contêm um único valor – um valor atómico – estão na 1FN.

Podemos afirmar que o nosso modelo está na 1FN pois:

- Cada tabela tem uma chave primária.
- Os valores de cada um dos atributos de uma tabela são todos atributos atómicos.
- Não existem grupos de dados repetidos.

- **2^a Forma Normal (2FN)**

Uma relação está na 2FN se todos os seus atributos não-primos (que não fazem parte de uma chave primária) forem totalmente dependentes da sua chave primária.

Assim, o nosso modelo está na 2FN porque:

- Cada uma das relações está na 1FN.
- Todos os seus atributos não-primos das relações são totalmente dependentes da sua chave primária. Isto é, não existem dependências parciais.

- **3^a Forma Normal (3FN)**

Uma tabela está na 3FN se todos os atributos que não são chaves primárias forem mutuamente independentes, não havendo assim dependências funcionais transitivas. Na prática, isto significa que os atributos que não dependem da chave primária – que dependem de outros atributos – devem ser “eliminados” da relação, ou seja, devem ser transferidos para outra tabela.

Assim, o nosso modelo está na 3FN porque:

- Cada uma das relações está na 1FN e na 2FN.
- Todos os atributos dependem única e exclusivamente da chave primária, não havendo assim dependências funcionais transitivas.

Desta forma, conclui-se que o modelo lógico está normalizado, estando de acordo com as três primeiras Formas Normais.

4.4. Validação do Modelo com Interrogações do Utilizador

Após a construção do modelo lógico e da sua normalização, um requisito básico para obter a validação do modelo é que este responda a todas as perguntas dos seus utilizadores. Para isso, Lígo e a sua equipa reuniram-se com Megálo Theta e explicaram-lhe como iria funcionar a sua base de dados. Depois testaram algumas das suas questões.

1. Qual a lista dos detetives que trabalham no escritório 1?

É necessário aceder à tabela Escritório e selecionar o escritório pretendido, que neste caso é o 1. De seguida, acede-se à tabela dos departamentos para se perceber quais são os departamentos que integram esse escritório, utilizando a chave estrangeira associada. Semelhantemente, na tabela dos detetives, acede-se aos dados armazenados na chave estrangeira “departamento” e, para todos os departamentos identificados no passo anterior, recebe-se uma lista de detetives.

2. Quais os casos associados ao cliente com o NIF 123456789?

É necessário aceder à tabela Cliente e ver qual dos clientes tem efetivamente esse Número de Identificação Fiscal (*idCliente*). De seguida obtemos na tabela Caso aqueles que têm esse número como chave estrangeira.

3. Qual foi a faturação do escritório 2 no mês de fevereiro?

Começamos por aceder à tabela Pagamento e seleccionamos todos aqueles que foram feitos no mês de fevereiro. Acedemos também ao *id* dos casos associados a esses pagamentos. Depois disso, teremos de aceder ao ou aos detetives que os investigaram, através do seu número mecanográfico (*idDetetive*). Como cada detetive está associado a um departamento, obtemos o *id* dos departamentos que receberam esses pagamentos. Depois, basta filtrar esses departamentos de forma a obter apenas aqueles que têm o escritório 2 como chave estrangeira, obtendo assim toda a faturação que o escritório 2 teve nos vários departamentos no mês de fevereiro.

4. Qual a lista dos casos de homicídio que se encontram atualmente em investigação (sem data de fim) em qualquer um dos escritórios?

Primeiramente, acede-se à tabela dos casos e seleciona-se aqueles que não têm data de fim. Depois, associando os vários casos aos detetives que os estão a investigar, conseguimos obter o departamento em que estes se encontram a trabalhar. Depois de obtermos todos os departamentos, filtramos aqueles que têm “Homicídios” no atributo “tipo” e obtemos a lista pretendida.

Depois desta demonstração por parte da equipa de Lígo, o Senhor Theta ficou bastante impressionado com o funcionamento do modelo e aprovou a continuidade do projeto.

Estando construído, normalizado e validado o modelo lógico, o próximo passo a seguir será a sua implementação física.

5. Implementação Física

5.1. Apresentação e explicação da base de dados implementada

A fim de desenvolver o esquema físico da base de dados da DE0, tivemos, inicialmente, de escolher qual o SGBD a utilizar. Uma vez que nos encontramos no domínio do modelo de dados relacional, optamos por usar o MySQL pois, além de ser aquele com o qual nos encontramos mais familiarizados, disponibiliza também de mecanismos de controlo de concorrência, que permite evitar futuros problemas no contexto da aplicação.

Para tal, é necessário realizar a tradução das relações base, que foram previamente definidas no modelo lógico, de forma a que estas e as suas restrições estejam a ser suportadas pelo SGBD.

O primeiro passo a seguir, é a criação de um *schema* onde colocaremos os nossos objetos.

```
1 • CREATE SCHEMA IF NOT EXISTS AgenciaDeteta;
2
3 • USE AgenciaDeteta;
```

Figura 5 – Criação do *schema* AgenciaDeteta e utilização do mesmo

A utilização do ‘IF NOT EXISTS’ garante que o *schema* só é criado se não existir anteriormente. O comando USE garante que as tabelas que forem definidas a seguir são criadas nesse *schema*.

Seguindo as restrições impostas no modelo lógico e na definição dos atributos, podemos proceder à criação das várias tabelas da base de dados.

```
5 • CREATE TABLE IF NOT EXISTS Escritorio (
6     idEscritorio INT NOT NULL AUTO_INCREMENT,
7     nome VARCHAR(45) NOT NULL,
8     telefone VARCHAR(14) NOT NULL,
9     cod_postal INT NOT NULL,
10    rua VARCHAR(100) NOT NULL,
11    localidade VARCHAR(45) NOT NULL,
12    porta INT NOT NULL,
13    PRIMARY KEY (idEscritorio)
14 );
```

Figura 6 – Implementação física da tabela ‘Escritório’

```

18 • - CREATE TABLE IF NOT EXISTS Departamento (
19     idDepartamento INT NOT NULL,
20     tipo ENUM('Homicídios', 'Roubos', 'Sequestros') NOT NULL,
21     escritorio INT NOT NULL,
22     PRIMARY KEY (idDepartamento),
23     FOREIGN KEY (escritorio) REFERENCES Escritório(idEscritorio)
24 );

```

Figura 7 – Implementação física da tabela ‘Departamento’

```

28 • - CREATE TABLE IF NOT EXISTS Detetive (
29     idDetetive INT NOT NULL AUTO_INCREMENT,
30     nome VARCHAR(100) NOT NULL,
31     telefone VARCHAR(14) NOT NULL,
32     dta_nascimento DATE NOT NULL,
33     cod_postal INT NOT NULL,
34     rua VARCHAR(100) NOT NULL,
35     localidade VARCHAR(45) NOT NULL,
36     porta INT NOT NULL,
37     estado ENUM('No Ativo', 'Reformado') NOT NULL,
38     departamento INT NOT NULL,
39     PRIMARY KEY (idDetetive),
40     FOREIGN KEY (departamento) REFERENCES Departamento(idDepartamento)
41 );

```

Figura 8 – Implementação física da tabela ‘Detetive’

```

45 • - CREATE TABLE IF NOT EXISTS Cliente (
46     idCliente INT NOT NULL,
47     nome VARCHAR(100) NOT NULL,
48     telefone VARCHAR(14) NOT NULL,
49     cod_postal INT NOT NULL,
50     rua VARCHAR(100) NOT NULL,
51     localidade VARCHAR(45) NOT NULL,
52     porta INT NOT NULL,
53     PRIMARY KEY (idCliente)
54 );

```

Figura 9 – Implementação física da tabela ‘Cliente’

```

58 • - CREATE TABLE IF NOT EXISTS Caso (
59     idCaso INT NOT NULL AUTO_INCREMENT,
60     data_inicio DATE NOT NULL,
61     data_fim DATE,
62     cliente INT NOT NULL,
63     PRIMARY KEY (idCaso),
64     FOREIGN KEY (cliente) REFERENCES Cliente(idCliente)
65 );

```

Figura 10 – Implementação física da tabela ‘Caso’

```

69 • CREATE TABLE IF NOT EXISTS Pagamento (
70     idPagamento INT NOT NULL AUTO_INCREMENT,
71     valor INT NOT NULL,
72     estado ENUM('Pago', 'Não Pago') NOT NULL,
73     data_pagamento DATE NOT NULL,
74     metodo ENUM('Numerário', 'Cartão', 'MBWay', 'Transferência') NOT NULL,
75     caso INT NOT NULL,
76     PRIMARY KEY (idPagamento),
77     FOREIGN KEY (caso) REFERENCES Caso(idCaso)
78 );

```

Figura 11 – Implementação física da tabela ‘Pagamento’

```

82 • CREATE TABLE IF NOT EXISTS Caso_por_Detetive (
83     detetive INT NOT NULL,
84     caso INT NOT NULL,
85     PRIMARY KEY (detetive, caso),
86     FOREIGN KEY (detetive) REFERENCES Detetive(idDetetive),
87     FOREIGN KEY (caso) REFERENCES Caso(idCaso)
88 );

```

Figura 12 – Implementação física da tabela ‘Caso_por_Detetive’

5.2. Criação de utilizadores da base de dados

O primeiro passo a seguir é definir quem poderá aceder a certos dados da base de dados. Para isso, usando o SQL, procedemos à criação de utilizadores e definimos as suas permissões de acesso ao sistema.

Os utilizadores principais da base de dados serão os clientes. Segundo os requisitos de controlo que definimos no subcapítulo 2.2.3, um cliente não deve conseguir aceder aos dados de outros clientes, dos detetives nem informações dos casos associados a outros clientes (RC1, RC2 e RC3). Apenas pode aceder às informações dos casos associados a si se se encontrarem solucionados (RC4).

```

CREATE USER 'cliente1'@'localhost' IDENTIFIED BY 'palavrapasse';

CREATE VIEW ClienteRestr AS
SELECT *
FROM Cliente
WHERE idCliente = 123456789;

CREATE VIEW CasosSolucionadoCliente1 AS
SELECT *
FROM Caso c
WHERE c.cliente = 123456789 AND c.data_fim IS NOT NULL;

GRANT SELECT ON ClienteRestr TO cliente1;
GRANT SELECT ON CasosSolucionadoCliente1 TO cliente1;

REVOKE ALL ON Cliente FROM 'cliente1'@'localhost';
REVOKE ALL ON Detetive FROM 'cliente1'@'localhost';
REVOKE ALL ON Caso FROM 'cliente1'@'localhost';

```

Figura 13 – Criação do utilizador ‘Cliente’

Outros utilizadores que poderíamos criar seriam os detetives. Os detetives não podem aceder aos dados pessoais de outros detetives, mas poderão aceder às informações de todos os casos de todos os escritórios (RC5 e RC6). Ou seja, a única restrição de permissão imposta pelos requisitos de controlo relativa aos detetives é não poderem ver as informações de outros detetives.

5.3. Povoamento da base de dados

O próximo passo a seguir é o povoamento da base de dados, que é feito de duas formas.

Na primeira, inserimos informações diretamente na base de dados através da instrução INSERT em SQL. Através deste método, conseguimos registar na base de dados todos os registo analógicos da DE0 (anteriores à criação da base de dados).

```

19    -- Povoamento da tabela "Escritório"
20 •  INSERT INTO Escritório (idEscritorio, nome, telefone, cod_postal, rua, localidade, porta)
21      VALUES
22          ('1', 'Alpha', '00351111111111', '4710111', 'Rua do Caires', 'Braga', '10'),
23          ('2', 'Beta', '00351222222222', '4940222', 'Rua 25 de Abril', 'Paredes de Coura', '20'),
24          ('3', 'Gamma', '00351333333333', '8200333', 'Rua da Oura', 'Albufeira', '30'),
25          ('4', 'Delta', '00351444444444', '4830444', 'Av. dos Bombeiros Voluntários', 'Póvoa de Lanhoso', '40'),
26          ('5', 'Epsilon', '00351555555555', '4610555', 'Rua São Martinho', 'Felgueiras', '50'),
27          ('6', 'Zeta', '00351666666666', '1500666', 'Av. Eusébio de Silva Ferreira', 'Lisboa', '60'),
28          ('7', 'Eta', '00351777777777', '4990777', 'Avenida dos Plátanos', 'Ponte de Lima', '70'),
29          ('8', 'Theta', '00351888888888', '4905888', 'Rua Direita', 'Barcelos', '80'),
30          ('9', 'Iota', '00351999999999', '4815999', 'Avenida da Liberdade', 'Guimarães', '90')
31      ;

```

Figura 14 – Exemplo de povoamento por inserção de dados

Na segunda forma, criamos um programa, na linguagem Python, que acede à base de dados e nos permite inserir dados em duas tabelas distintas (Cliente e Caso).

```

1 import mysql.connector
2
3 basedados = mysql.connector.connect(
4     host="127.0.0.1",
5     database="AgenciaDeteta",
6     user="root",
7     password="pedro2004")
8
9 mycursor = basedados.cursor()
10
11 sql1 = "INSERT INTO Cliente(idCliente,nome,telefone,cod_postal,rua,localidade,porta) VALUES(%s,%s,%s,%s,%s,%s,%s)"
12 valores1 = [("158139024","Rui Jorge Loureiro","927734210","4830502","Rua da Grila","Taíde","38"),
13             ("2502140128","Cátia Vanessa","926803555","4700432","Rua Daculá","Nogueira","321"),
14             ("199132845","Rui Redes","913914256","4720418","Avenida Mordomo Cunha","Leiria","2"),
15             ("331882003","Mário Guedes","913432991","4512512","Rua de Lima","Ponte de Lima","501"),
16             ("255621023","Juliana Pinha","969655445","4101995","Avenida da Republica","Lamaçães","132"),]
17
18 sql2 = "INSERT INTO Caso(idCaso, data_inicio,data_fim, cliente) VALUES(%s,%s,%s,%s)"
19 valores2=[("116","2025-04-27","2025-05-15","158139024"),
20           ("117","2025-04-28","2025-06-06","2502140128"),
21           ("118","2025-04-28","2025-06-16","199132845"),
22           ("119","2025-04-30","2025-07-01","331882003"),
23           ("120","2025-05-02","2025-07-01","255621023"),]
24
25 mycursor.executemany(sql1,valores1)
26 mycursor.executemany(sql2,valores2)
27
28 basedados.commit()
29
30 basedados.close()

```

Figura 15 – Povoamento através de um programa em Python

Primeiramente, importamos o módulo mysql.connector que permite a interação com uma base de dados, depois estabelecemos a conexão com a nossa base de dados (AgenciaDeteta) localizada no servidor 127.0.0.1 usando o usuário root e a senha pedro2004. Precisamos também de criar um cursor que nos permite executar comandos SQL. De seguida definimos a consulta SQL para inserir dados na tabela Cliente e Caso e especificamos os valores a serem inseridos. Para executar estas consultas SQL usamos então o cursor: “cursor.executemany()” para ambas as consultas e, por fim, usamos o commit e o close para confirmar as alterações e fechar a conexão com a base de dados, respetivamente.

5.4. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)

Outro fator importante a ter em conta durante a implementação física da base de dados é o espaço que esta irá ocupar.

Para a estimativa do espaço em disco, criou-se uma tabela com o tamanho (em bytes) de cada atributo (tendo em conta o seu tipo de dados). Multiplicou-se cada valor pelo seu número de ocorrências. Do seu somatório resulta o tamanho estimado de todos os dados armazenados até ao presente, no pior dos casos (54804 bytes).

Tabela	Atributos	Tipo de Dados	Tamanho (bytes)	Ocorrências	Subtotal
Escritório	idEscritorio	INT	4	9	36
	nome	VARCHAR(45)	46		414
	telefone	VARCHAR(14)	15		135
	cod_postal	INT	4		36
	rua	VARCHAR(100)	101		909
	localidade	VARCHAR(45)	46		414
	porta	INT	4		36
Departamento	idDepartamento	INT	4	27	108
	tipo	ENUM('Homicídios', 'Roubos', 'Sequestros')	1		27
	escritorio	INT	4		108
Detetive	idDetetive	INT	4	81	324
	nome	VARCHAR(100)	101		8181
	telefone	VARCHAR(14)	15		1215
	dta_nascimento	DATE	3		243
	cod_postal	INT	4		324
	rua	VARCHAR(100)	101		8181
	localidade	VARCHAR(45)	46		3726
	porta	INT	4		324

	estado	ENUM('No Ativo', 'Reformado')	1		81
	departamento	INT	4		324
Cliente	idCliente	INT	4		332
	nome	VARCHAR(100)	101		8383
	telefone	VARCHAR(14)	15		1245
	cod_postal	INT	4	83	332
	rua	VARCHAR(100)	101		8383
	localidade	VARCHAR(45)	46		3818
	porta	INT	4		332
Caso	idCaso	INT	4		460
	data_inicio	DATE	3		345
	data_fim	DATE	3		345
	cliente	INT	4		460
Pagamento	idPagamento	INT	4		860
	valor	INT	4		860
	estado	ENUM('Pago', 'Não Pago')	1		215
	data_pagamento	DATE	3	215	645
	metodo	ENUM('Numerário', 'Cartão', 'MBWay', 'Transferência')	1		215
	caso	INT	4		860
Caso_por_Detetive	detetive	INT	4		784
	caso	INT	4	196	784
				TOTAL	54804

Tabela 7 – Estimativa do espaço ocupado pela base de dados (inicialmente)

Para calcular a estimativa de crescimento anual da base de dados, foi agendada uma reunião entre Megálo Theta e a equipa de Lígo.

Dessa reunião, retirou-se que Megálo não está a planear voltar a expandir a DE0 nos próximos anos, por isso as únicas tabelas a receber novos dados serão as do Cliente, Caso, Pagamento e Caso_por_Detetive.

Ao analisar os dados recolhidos, Lígo verificou que, nos 3 anos anteriores à implementação da base de dados (2022, 2023 e 2024), foram resolvidos entre 8 e 9 casos em cada ano. Assim, tendo em conta que cada caso está associado a um cliente, a, no máximo, 2 pagamentos e a, em média, dois detetives, a equipa pôde calcular uma previsão de crescimento para a base de dados da agência.

Tabela	Atributos	Tipo de Dados	Tamanho (bytes)	Ocorrências	Subtotal
Cliente	idCliente	INT	4	92	368
	nome	VARCHAR(100)	101		9292
	telefone	VARCHAR(14)	15		1380
	cod_postal	INT	4		368
	rua	VARCHAR(100)	101		9292
	localidade	VARCHAR(45)	46		4232
	porta	INT	4		368
Caso	idCaso	INT	4	124	496
	data_inicio	DATE	3		372
	data_fim	DATE	3		372
	cliente	INT	4		496
Pagamento	idPagamento	INT	4	233	932
	valor	INT	4		932
	estado	ENUM('Pago', 'Não Pago')	1		233
	data_pagamento	DATE	3		699
	metodo	ENUM('Numerário', 'Cartão', 'MBWay', 'Transferência')	1		233
	caso	INT	4		932
Caso_por_Detetive	detetive	INT	4	214	856
	caso	INT	4		856
					TOTAL 57855

Tabela 8 – Estimativa do espaço ocupado pela base de dados (após um ano)

Para se realizar esta estimativa, mantiveram-se os valores do espaço ocupado pelas tabelas Escritório, Departamento e Detetive. Assim, espera-se que a base de dados ocupe 57855 bytes após o primeiro ano, o que corresponde a um crescimento de, aproximadamente, **5,57%**.

É de notar que esta estimativa foi realizada para o pior caso possível (a cada um dos nove novos casos estar associado um cliente novo, dois pagamentos e dois detetives), o que pode não acontecer na realidade.

5.5. Definição e caracterização de vistas de utilização em SQL

Ao criar um SGBD, um fator importante são os acessos que cada utilizador pode realizar.

Resumidamente, devemos levar em consideração as perspetivas de utilização. Nesse contexto, as vistas (ou views) em SQL desempenham um papel crucial.

As vistas são objetos de bancos de dados que atuam como tabelas virtuais, sendo definidas por consultas SQL que selecionam dados de uma ou mais tabelas reais (ou outras vistas). Elas permitem simplificar consultas complexas, restringir o acesso a determinados dados e apresentar uma estrutura de dados personalizada aos usuários.

Alguns exemplos de vistas são os seguintes:

View 1: Detalhes de Casos e Detetives

Uma view que une informações sobre casos e os detetives atribuídos a eles.

```
5 • CREATE VIEW Casos_e_Detectives AS
6     SELECT C.idCaso, C.data_inicio, C.data_fim, CL.nome AS Cliente, D.nome AS Detetive, D.departamento
7     FROM Caso C
8     INNER JOIN Cliente CL ON C.cliente = CL.idCliente
9     INNER JOIN Caso_por_Detective CD ON C.idCaso = CD.caso
10    INNER JOIN Detetive D ON CD.detetive = D.idDetetive;
```

Figura 16 – View 1 em SQL

View 2: Pagamentos por Caso

Uma view que mostra os pagamentos realizados para cada caso.

```
14 • CREATE VIEW vw_Pagamentos_Casos AS
15     SELECT C.idCaso, CL.nome AS Cliente, P.valor, P.estado, P.data_pagamento, P.metodo
16     FROM Pagamento P
17     JOIN Caso C ON P.caso = C.idCaso
18     JOIN Cliente CL ON C.cliente = CL.idCliente;
```

Figura 17 – View 2 em SQL

View 3: Informações do Cliente e Seus Casos

Uma view que agrupa informações sobre clientes e os casos relacionados a eles.

```
22 • CREATE VIEW vw_Clientes_Casos AS
23     SELECT CL.idCliente, CL.nome, CL.telefone, CL.localidade, C.idCaso, C.data_inicio, C.data_fim
24     FROM Cliente CL
25     INNER JOIN Caso C ON CL.idCliente = C.cliente;
```

Figura 186 – View 3 em SQL

View 4: Detectives e seus Departamentos

Uma view que lista detetives e os departamentos a que pertencem.

```

29 • CREATE VIEW vw_Detectives_Departamentos AS
30     SELECT D.idDetective,D.nome AS Detective,DP.idDepartamento AS Departamento,D.telefone,D.localidade
31     FROM Detective AS D
32     INNER JOIN Departamento AS DP ON D.departamento = DP.idDepartamento;

```

Figura 19 – View 4 em SQL

5.6. Tradução das interrogações do utilizador para SQL

Com base no modelo lógico, na criação e no povoamento da base de dados, realizamos consultas SQL para atender às perguntas e necessidades do gerente. As queries em SQL permitiram extrair informações relevantes da base de dados, aplicar filtros e combinar dados de diferentes tabelas para obter resultados específicos.

Query1: Qual a lista dos detectives que trabalham no escritório 1?

Criamos um procedure para obter o idDetective o Nome de cada detetive que trabalham em cada escritório.

```

1      DELIMITER $$ 
2 •  CREATE PROCEDURE query1 (IN Id INT)
3      BEGIN
4          SELECT Detective.idDetective , Detective.nome
5          FROM Detective AS D INNER JOIN Departamento AS De ON D.departamento = De.idDepartamento
6                          INNER JOIN Escritorio AS E ON De.escritorio = E.idEscritorio
7          WHERE E.idEscritorio = Id;
8      END
9      $$ 
10     DELIMITER ;

```

Figura 20 – Query 1 em SQL

Query2: Quais os casos associados ao cliente com o NIF 123456789?

Criamos um procedure para obter os Id dos casos de um cliente em específico.

```

16     DELIMITER $$ 
17 •  CREATE PROCEDURE query2 (IN nif INT)
18     BEGIN
19         SELECT C.idCaso
20         FROM Casos AS C INNER JOIN Cliente AS Cl ON C.Cliente=Ca.idCliente
21         WHERE Cl.idCliente=nif;
22     END
23     $$ 
24     DELIMITER ;

```

Figura 21 – Query 2 em SQL

Query3: Qual foi a faturaçao do escritorio 2 no mês de fevereiro?

Criamos uma function para obter a faturaçao de um escritorio específico em um certo mês.

```

26      DELIMITER $$ 
27 •  CREATE FUNCTION query3(idEsc INT, mes INT)
28          RETURNS FLOAT DETERMINISTIC
29      BEGIN
30          RETURN(SELECT sum(P.valor)
31                  FROM Pagamento AS P INNER JOIN Caso AS C ON P.caso=C.idCaso
32                                  INNER JOIN Caso_por_Detetive AS CD ON c.idCaso=CD.caso
33                                  INNER JOIN Detetive AS D ON CD.detetive=D.idDetetive
34                                  INNER JOIN Departamento AS De ON D.departamento=DE.idDepartamento
35                                  INNER JOIN Escritório AS E ON De.escritorio=E.idEscritorio
36                  WHERE idEsc=E.idEscritorio AND mes=MONTH(P.data_pagamento));
37      END
38  $$ 
39  DELIMITER ;

```

Figura 22– Query 3 em SQL

Query4: Qual a lista dos casos de homicídio que se encontram atualmente em investigação (sem data de fim) em qualquer um dos escritórios?

Uma view que lista todos os casos que ainda estão ativos (onde data_fim é nula).

```

42      DELIMITER $$ 
43 •  CREATE PROCEDURE query4()
44      BEGIN
45          SELECT C.idCaso
46          FROM Caso AS C
47          WHERE C.data_fim IS NULL;
48      END
49  $$ 
50  DELIMITER ;

```

Figura 23– Query 4 em SQL

5.7. Indexação do Sistema de Dados

A indexação desempenha um papel fundamental na otimização do desempenho e da eficiência das consultas num sistema de base de dados. De forma a minimizar o tempo de resposta às *queries* feitas, decidimos utilizar índices, que são estruturas de dados que nos irão permitir aceder a um ficheiro/arquivo mais facilmente.

Como uma das coisas mais procuradas são os contactos dos clientes, para a facilitar e agilizar a comunicação, quando necessária, com os clientes, que necessita da consulta dos seus contactos telefónicos, concedemos um índice á coluna telefone da tabela Cliente:

```
CREATE INDEX idx_telefone_do_cliente ON Cliente(telefone);
```

Figura 242– Criação do índice sobre a coluna telefone

Utilizando a mesma lógica que utilizámos para a tabela dos clientes, podemos ver que na tabela dos casos o ID destes é uma coluna frequentemente procurada pelos clientes, dessa forma, decidimos conceder-lhe um índice:

```
CREATE INDEX idx_id_do_caso ON Caso(idCaso);
```

Figura 253– Criação do índice sobre a coluna idCaso

5.8. Implementação de procedimentos, funções e gatilhos

Um procedimento é uma coleção de instruções SQL pré-compiladas armazenadas na base de dados que podem ser invocadas mais tarde.

No nosso modelo, achamos relevante criar um *procedure* que nos devolve todos os casos (e os respetivos clientes) de um certo departamento num ano específico.

```
58      DELIMITER $$  
59 •  CREATE PROCEDURE procedure1(ano INT, dep INT)  
60      BEGIN  
61          SELECT C.idCaso,C1.nome  
62          FROM Caso AS C INNER JOIN Cliente AS C1 ON C.cliente=C1.idCliente  
63              INNER JOIN Caso_por_Detetive AS CD ON C.idCaso=CD.caso  
64                  INNER JOIN Detetive AS D ON CD.detetive=D.idDetetive  
65                      INNER JOIN Departamento AS De ON D.departamento=DE.idDepartamento  
66                          WHERE DE.idDepartamento = 1 AND YEAR(C.data_inicio)=2004;  
67      END  
68      $$  
69  DELIMITER ;
```

Figura 264– Procedure 1

Uma função é um programa armazenado que devolve um único valor. Tipicamente, usam-se funções para encapsular fórmulas ou regras de negócio comuns que são reutilizáveis entre instruções SQL ou programas armazenados.

Uma função que achamos conveniente para criar para o nosso trabalho, foi uma função que ao receber o NIF de determinado cliente nos devolvesse o total de gastos que esse cliente teve na agência.

```

46      DELIMITER $$ 
47 •  CREATE FUNCTION funcao1(NIF INT)
48          RETURNS FLOAT DETERMINISTIC
49      BEGIN
50          RETURN(SELECT sum(P.valor)
51                  FROM Pagamento AS P INNER JOIN Caso AS C ON P.caso=C.idCaso
52                                  INNER JOIN Cliente AS Cl ON c.cliente=Cl.idCliente
53                  WHERE Cl.idCliente=NIF);
54      END $$ 
55  DELIMITER ;

```

Figura 275– Função 1

Um trigger é invocado automaticamente quando uma operação de alteração específica (instrução INSERT, UPDATE ou DELETE) é executada sobre uma determinada tabela. Os triggers são úteis para tarefas como a aplicação de regras comerciais ou até para validação de dados quando inseridos na base de dados.

No nosso projeto decidimos criar um trigger que quando fazemos UPDATE na tabela caso e damos um caso como finalizado, tornando a data_fim NOT NULL, criamos automaticamente o segundo pagamento ficando com o valor, método e caso do primeiro, assumindo o estado como ‘Pago’ e a data_pagamento como a data em que foi finalizado o caso.

```

81      DELIMITER $$ 
82 •  CREATE TRIGGER trigger1
83      AFTER UPDATE ON Caso
84      FOR EACH ROW
85      BEGIN
86          IF OLD.data_fim IS NULL AND NEW.data_fim IS NOT NULL THEN
87              INSERT INTO Pagamento (valor, estado, data_pagamento, metodo, caso)
88                  SELECT valor, 'Pago', NEW.data_fim, metodo, caso
89                  FROM Pagamento
90                  WHERE caso = NEW.idCaso
91                  ORDER BY idPagamento
92                  LIMIT 1;
93          END IF;
94      END $$ 
95  DELIMITER ;

```

Figura 286– Trigger 1

Um caso concreto de aplicação deste trigger é por exemplo, assumimos que o caso com id 115 foi finalizado 10/12/2025 e nesse dia foi levantado o segundo pagamento, e ao atualizar o caso como concluído, vai também criar o outro pagamento com os mesmos valores e na data em que este caso foi finalizado:

```

108 •   select*
109     from Pagamento INNER JOIN Caso ON Pagamento.caso=Caso.idCaso
110     WHERE Caso.idCaso=115;
111

```

	idPagamento	valor	estado	data_pagamento	metodo	caso	idCaso	data_inicio	data_fim	cliente
▶	215	1500	Pago	2025-04-25	MBWay	115	115	2025-04-25	NULL	678478901

Figura 297 – Tabela de pagamento antes do update

```

UPDATE Caso
SET data_fim = '2025-12-10'
WHERE idCaso = 115;

```

Figura 308– Update que vai rodar o trigger

```

108 •   select*
109     from Pagamento INNER JOIN Caso ON Pagamento.caso=Caso.idCaso
110     WHERE Caso.idCaso=115;
111

```

	idPagamento	valor	estado	data_pagamento	metodo	caso	idCaso	data_inicio	data_fim	cliente
▶	215	1500	Pago	2025-04-25	MBWay	115	115	2025-04-25	2025-12-10	678478901
	222	1500	Pago	2025-12-10	MBWay	115	115	2025-04-25	2025-12-10	678478901

Figura 319– Tabela de pagamento depois do update

6. Conclusões e Trabalho Futuro

A equipa de Ligo considera que o projeto de implementação de um sistema de bases de dados para a agência representa uma iniciativa crucial para modernizar e otimizar as suas operações e promover uma comunicação mais eficiente entre os vários escritórios e departamentos da agência.

O projeto, após análise da viabilidade, evidencia vários benefícios, tanto operacionais como estratégicos, que serão alcançados através do meio da transição para um meio digital. A conjunção adequada de recursos humanos e materiais resulta numa abordagem planeada e flexível, fundamental para o sucesso da implementação.

A modelação conceptual e lógica do sistema de bases de dados foi realizada de forma abrangente, identificando entidades, relacionamentos e atributos essenciais para atender às necessidades da agência. A migração do modelo conceptual para o modelo lógico demonstra como as estruturas foram derivadas e validadas para garantir uma representação mais precisa dos dados.

Assim sendo, a primeira fase deste projeto revela bastantes pontos fortes: a análise detalhada dos requisitos, a comunicação eficaz entre os membros da equipa e o uso das ferramentas adequadas para a modelação e para uma boa implementação do sistema de bases de dados.

A implementação física do sistema de bases de dados, foi desenvolvida no *MySQLWorkbench*. Explicando e justificando a criação de cada tabela, apresentámos o esquema físico resultante. Criámos utilizadores com perfis de utilização e permissões específicas, garantindo a segurança e a correta atribuição de acessos. Realizámos o povoamento da base de dados através de instruções *INSERT* e um programa automatizado em Python, assegurando a correta inserção e integridade dos dados. Calculámos a dimensão inicial e a taxa de crescimento anual da base de dados, com base no tipo de dados dos vários atributos, usando uma folha de cálculo para maior precisão. Desenvolvemos as vistas (VIEW) necessárias para consultas eficientes e traduzimos as interrogações dos utilizadores em queries SQL otimizadas. Além disso, definimos índices para melhorar o desempenho das consultas e implementámos procedimentos, funções e gatilhos para assegurar a integridade e funcionalidade dos dados.

O seu desenvolvimento é crucial para a implementação de um sistema de gestão da agência. Um dos aspetos positivos mais notáveis do projeto foi a implementação de técnicas de indexação, que resultaram numa melhoria significativa no desempenho das consultas. Ao criar índices específicos para as tabelas mais acedidas, conseguimos reduzir o tempo de resposta da base de dados ao utilizador. Essa otimização foi crucial para a eficiência do sistema, permitindo uma recuperação de dados mais rápida e eficaz. Além disso, a indexação facilitou a execução de consultas complexas, tornando o sistema mais ágil e responsivo.

Como qualquer projeto, ao longo do tempo, perceberam-se alguns erros durante o processo. A falta de documentação em algumas etapas prejudicou a consistência do projeto e atrasou a correção de erros, afetando a fluidez do desenvolvimento.

Algumas decisões iniciais incorretas exigiram correções que atrasaram o progresso do projeto. Realizamos também alterações importantes, como o ajuste do tipo de dados de alguns atributos (tipo de departamento, método e estado de pagamento, estado do detetive e telefone do escritório, detetive e cliente).

Ainda assim, consideramos que estes problemas não comprometeram a qualidade do projeto, sendo facilmente resolvidos. Tendo isto em conta, podemos afirmar que os grandes alicerces da solução encontrada são as fases de definição do problema e de desenvolvimento do modelo conceptual.

É relevante referir que o modelo lógico deriva do modelo conceptual, que, posteriormente, deu origem à implementação física. O modelo conceptual assumiu um papel de base para todo o trabalho, o que significa que, se este fosse incorreto e descuidado, resultaria num sistema obsoleto.

Para o futuro, será importante realizar uma revisão contínua do sistema de base de dados para garantir que continue a atender às necessidades em constante evolução da agência. Além disso, a expansão do sistema para incluir funcionalidades adicionais, como análise de dados avançada e integração com outras plataformas, pode ser considerada para melhorar ainda mais a eficiência e eficácia das operações da DE0.

Posteriormente, é necessário desenvolver programas para recolha de dados e criação de painéis de análise. Python será uma escolha adequada para a recolha de dados devido à sua versatilidade, enquanto o Microsoft Power BI Desktop pode ser utilizado para criar painéis de análise visualmente atraentes e informativos. Ao garantir a disponibilidade e competência adequadas desses recursos, a DE0 estará bem posicionada para implementar uma base de dados eficaz que atenda às suas necessidades operacionais e estratégicas.

Correções efetuadas

Após a finalização da segunda fase da construção da base de dados, é importante notar alguns aspetos que foram alterados após a entrega da primeira fase. Estas pequenas alterações deveram-se tanto ao aconselhamento dos docentes da Unidade Curricular, como à existência de certas incorreções observadas durante o desenvolvimento da segunda fase do projeto.

Assim, enunciamos nesta secção alguns deles que, embora não sejam pontos fulcrais para o desenvolvimento da base de dados, contribuem para a sua evolução e bom funcionamento. Consideramos que as alterações ao resumo inicial deste relatório não são de justificação necessária, uma vez que apenas foi adicionado o processo que seguimos na construção da segunda fase do projeto.

1. Alteração do tipo de dados do atributo “telefone” nas entidades “Escritório”, “Detetive” e “Cliente” de INT para VARCHAR(14).

Após a apresentação da fase I do projeto, foi-nos aconselhado fazer esta alteração para permitir à base de dados registar números de telefone estrangeiros. A título de exemplo, um número português será registado na base de dados com o indicativo 00351 e os 9 dígitos que o constituem. O tipo VARCHAR garante que os dois primeiros zeros não são eliminados pelo sistema, o que aconteceria se os dados de “telefone” fossem caracterizados com INT, como estava anteriormente.

2. Alteração do tipo de dados dos atributos “tipo” (da entidade “Departamento”) e “metodo” (da entidade “Pagamento”) de VARCHAR(45) para ENUM(...).

Estas alterações tiveram por base o mesmo intuito. A título de exemplo, esta modificação procura reduzir a possibilidade do departamento “Homicídios” ser introduzido na base de dados como “homicidios”. O tipo ENUM('Homicídios', 'Roubos', 'Sequestros'), ao contrário do tipo VARCHAR(45), garante que só existem esses três tipos de departamento. O mesmo para os métodos de pagamento, que foram alterados também de VARCHAR(45) para ENUM('Numerário', 'Cartão', 'MBWay', 'Transferência').

3. Alteração do tipo de dados dos atributos “estado” das entidades “Detetive” e “Pagamento” de BINARY(1) para ENUM(...).

Durante a implementação física da base de dados e a criação de *queries* SQL, observámos que o tipo BINARY(1) nos devolia “BLOB” na coluna “estado” ao selecionar as informações de cada tabela após o seu povoamento.

Por não conseguirmos resolver o erro e por falta de esclarecimento de dúvidas perante os docentes da Unidade Curricular, não achámos que fosse pertinente a entrega deliberada do projeto nestas condições. Por isso, foi decidido pela equipa alterar o “estado” do Detetive para ENUM('No Ativo', 'Reformado') e o do Pagamento para ENUM('Pago', 'Não Pago').

Referências

- Connolly, T., Begg, C., Database Systems: A Practical Approach to Design, Implementation, and Management, Addison- Wesley, Global Edition, 26 Sep 2014. ISBN-10: 1292061189, ISBN-13: 978-1292061184
- Belo, O. – https://elearning.uminho.pt/bbcswebdav/pid-1428103-dt-content-rid-7870421_1/xid-7870421_1 (“O Ciclo de Vida do Desenvolvimento de uma Base de Dados”);
- Belo, O. – https://elearning.uminho.pt/bbcswebdav/pid-1428103-dt-content-rid-7870424_1/xid-7870424_1 (“A Mercearia da D. Acácia”);
- Belo, O. – https://elearning.uminho.pt/bbcswebdav/pid-1428103-dt-content-rid-8376411_1/xid-8376411_1 (“Normalização de Dados”)
- Ferreira, D. – https://elearning.uminho.pt/bbcswebdav/pid-1441475-dt-content-rid-8134933_1/xid-8134933_1 (“PL04 - Modelação Conceptual”)
- Ferreira, D. – https://elearning.uminho.pt/bbcswebdav/pid-1446084-dt-content-rid-8184990_1/xid-8184990_1 (“PL05 - Modelação Lógica”)
- Ferreira, D. – https://elearning.uminho.pt/bbcswebdav/pid-1455084-dt-content-rid-8273227_1/xid-8273227_1 (“PL06 - Modelação Lógica e Normalização”)
- Ferreira, D. – https://elearning.uminho.pt/bbcswebdav/pid-1457450-dt-content-rid-8367912_1/xid-8367912_1 (“PL07 – Normalização e Álgebra Relacional”)
- Ferreira, D. – https://elearning.uminho.pt/bbcswebdav/pid-1465311-dt-content-rid-8577697_1/xid-8577697_1 (“PL10 – SQL Avançada”)
- MySQL. (n.d.). Storage requirements. MySQL 8.0 Reference Manual. Retrieved May, 2024, from <https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html>

Lista de Siglas e Acrónimos

BD	Base de Dados
DW	Data Warehouse
OLTP	<i>On-Line Analytical Processing</i>
ER	Entidade-Relacionamento
SGBD	Sistema de Gestão de Base de Dados
SBDR	Sistema de Base de Dados Relacional
SBD	Sistema de Base de Dados
S/N	Sim/Não
1FN	Primeira Forma Normal
2FN	Segunda Forma Normal
3FN	Terceira Forma Normal
NIF	Número de Identificação Fiscal

