



Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Letivo de 2023/2024

WaveDetectives

Pedro Filipe Maneta Pinto (a104176)

Marco António Fernandes Brito (a104187)

Pedro Seabra Vieira (a104352)

Tomás Henrique Alves Melo (a104529)

Maio, 2024

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

WaveDetectives

Pedro Filipe Maneta Pinto (a104173)

Marco António Fernandes Brito (a104187)

Pedro Seabra Vieira (a104352)

Tomás Henrique Alves Melo (a104529)

Maio, 2024

Resumo

Este relatório foi realizado no âmbito da unidade curricular de Bases de Dados, cujo objetivo central é a criação de um sistema de base de dados. Este sistema não só enfatiza a análise e planejamento, mas também a modelação, arquitetura, implementação e validação de sistemas de bases de dados. O projeto em questão visa implementar melhorias significativas numa agência de detetives privada, que atualmente enfrenta um caso de alta complexidade e importância. Tal caso exige a introdução de novos recursos e, conseqüentemente, uma base de dados capaz de gerir um volume substancial de informações de maneira eficiente, organizada e intuitiva.

Para garantir a otimização do nosso sistema, adotamos uma abordagem metodológica dividida em duas fases principais, com cada fase sujeita a um processo rigoroso de validação. Inicialmente, focamos na definição dos requisitos funcionais e não funcionais que a base de dados deveria satisfazer. Com os requisitos estabelecidos, avançamos para a modelação concetual, onde identificamos as entidades chave e os relacionamentos pertinentes, alinhados com os requisitos previamente definidos.

Antes de proceder à próxima fase, conduzimos uma validação aprofundada do modelo concetual para assegurar a sua adequação e integridade. Com base neste, posteriormente, desenvolvemos o modelo lógico. Este modelo foi então submetido a uma análise criteriosa, utilizando técnicas de normalização para evitar redundâncias e otimizar o desempenho, bem como testes de interrogações simulados por utilizadores para verificar a usabilidade e a precisão das informações recuperadas. Este processo garante que o sistema de base de dados não apenas atenda às necessidades atuais da agência, mas que também seja uma prevenção para a adaptação de futuras exigências e expansões.

Área de Aplicação: Estruturação e implementação de uma base de dados.

Palavras-Chave: Base de Dados, Modelo Concetual, Entidades, Atributos, Modelo Lógico, Relacionamentos, Normalização, Validação, MySQL Workbench, MySQL, BrModelo, Roubo, Chamada.

Índice

RESUMO.....	III
ÍNDICE	IV
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABELAS	VII
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO DE APLICAÇÃO DO SISTEMA	1
1.1. MOTIVAÇÃO E OBJETIVOS DO TRABALHO	2
1.2. ANÁLISE DA VIABILIDADE DO PROCESSO	2
1.3. RECURSOS E EQUIPA DE TRABALHO	3
1.4. PLANO DE EXECUÇÃO DO PROJETO	3
2. LEVANTAMENTO E ANÁLISE DE REQUISITOS	6
2.1. MÉTODO DE LEVANTAMENTO E DE ANÁLISE DE REQUISITOS ADOPTADOS.....	6
2.2. ORGANIZAÇÃO DOS REQUISITOS LEVANTADOS	7
2.3. ANÁLISE E VALIDAÇÃO GERAL DOS REQUISITOS	11
3. MODELAÇÃO CONCEPTUAL	12
3.1. APRESENTAÇÃO DA ABORDAGEM DE MODELAÇÃO	12
3.2. IDENTIFICAÇÃO E CARACTERIZAÇÃO DAS ENTIDADES	12
3.3. IDENTIFICAÇÃO E CARACTERIZAÇÃO DOS RELACIONAMENTOS	13
3.4. IDENTIFICAÇÃO E CARACTERIZAÇÃO DOS ATRIBUTOS DAS ENTIDADES E DOS RELACIONAMENTOS	15
3.5. APRESENTAÇÃO E EXPLICAÇÃO DO DIAGRAMA ER PRODUZIDO	20
4. MODELAÇÃO LÓGICA.....	21
4.1. CONSTRUÇÃO E VALIDAÇÃO DO MODELO DE DADOS LÓGICO	21
4.2. APRESENTAÇÃO E EXPLICAÇÃO DO MODELO LÓGICO PRODUZIDO.....	22
4.3. NORMALIZAÇÃO DE DADOS.....	24
4.4. VALIDAÇÃO DO MODELO COM INTERROGAÇÕES DO UTILIZADOR	25
5. IMPLEMENTAÇÃO FÍSICA.....	26
5.1. APRESENTAÇÃO E EXPLICAÇÃO DA BASE DE DADOS IMPLEMENTADA	26
5.1.1. Tabela Esquadra.....	26
5.1.2. Tabela Suspeito.....	27
5.1.3. Tabela Vitima.....	27
5.1.4. Tabela Torre	27

5.1.5	<i>Tabela Agente_Polícia</i>	28
5.1.6	<i>Tabela ContactoAgente</i>	28
5.1.7	<i>Tabela ContactoEsquadra</i>	29
5.1.8	<i>Tabela ContactoVitima</i>	29
5.1.9	<i>Tabela Processo_Criminal</i>	30
5.1.10	<i>Tabela Chamada</i>	31
5.2	criação de utilizadores da base de dados	31
5.3	Povoamento da base de dados	32
5.3.1	<i>Inserção de Dados com Instruções SQL</i>	32
5.3.2	<i>Inserção de Dados com Programa Interativo</i>	33
5.4	Cálculo do espaço da base de dados (inicial e taxa de crescimento anual).....	35
5.5	Definição e caracterização de vistas de utilização em SQL	41
5.6	Tradução das interrogações do utilizador para SQL	42
5.7	Indexação do sistema de dados	47
5.8	Implementação de procedimentos, funções e gatilhos	47
5.8.1	<i>Função</i>	48
5.8.2	<i>Procedures</i>	48
5.8.3	<i>Triggers</i>	50
6	CONCLUSÕES E TRABALHO FUTURO	51
7	BIBLIOGRAFIA	52

Índice de Figuras

Figura 1 - Modelo de Gantt da “Definição do Sistema”.	4
Figura 2 - Modelo de Gantt do “Levantamento e Análise de Requisitos”.	4
Figura 3 - Modelo de Gantt do “Modelação Concetual”.	4
Figura 4 - Modelo de Gantt da “Modelação Lógica”.	5
Figura 5 - Modelo de Gantt da “Conclusões e Trabalho Futuro”.	5
Figura 6 - Modelo Concetual	20
Figura 7 - Modelo Lógico	21
Figura 8 – Transformação da entidade AgentePolicia no modelo concetual para modelo lógico.	22
Figura 9 - Transformação da entidade Esquadra no modelo concetual para modelo lógico.	22
Figura 10 - Transformação da entidade Vítima no modelo concetual para modelo lógico.	23
Figura 11 - Transformação da entidade Processo_Criminal no modelo concetual para modelo lógico.	23
Figura 12 - Transformação da entidade Suspeito no modelo concetual para modelo lógico.	23
Figura 13 - Transformação da entidade Chamada no modelo concetual para modelo lógico.	24
Figura 14 - Transformação da entidade Torre no modelo concetual para modelo lógico.	24
Figura 15 - Resultados da Query 1	42
Figura 16 - Resultados da Query 2	43
Figura 17 - Resultados da Query 3	44
Figura 18 - Resultados da Query 4	45
Figura 19 - Resultados da Query 5	46

Índice de Tabelas

Tabela 1 – Recursos e equipa de trabalho.	3
Tabela 2 – Requisitos de descrição.	10
Tabela 3 – Requisitos de manipulação.	11
Tabela 4 – Requisitos de controlo.	11
Tabela 5 – Identificação e caracterização das entidades	13
Tabela 6 – Identificação e caracterização de relacionamentos	14
Tabela 7 – Identificação e descrição das entidades e dos seus atributos.	18
Tabela 8 - Espaço que cada Datatype ocupa	35
Tabela 9 - Espaço (Bytes) tabela Esquadra	35
Tabela 10 - Espaço (Bytes) tabela Chamada	36
Tabela 11 - Espaço (Bytes) tabela Torre	36
Tabela 12 - Espaço (Bytes) tabela Agente_Polícia	36
Tabela 13 - Espaço (Bytes) tabela ContactoAgente	37
Tabela 14 - Espaço (Bytes) tabela ContactoEsquadra	37
Tabela 15 - Espaço (Bytes) tabela ContactoVítima	37
Tabela 16 - Espaço (Bytes) tabela Processo Criminal	38
Tabela 17 - Espaço (Bytes) tabela Vítima	38
Tabela 18 - Espaço (Bytes) tabela Suspeito	39
Tabela 19 - Espaço da Base de Dados com e sem povoamento	39
Tabela 20 - Estimativa do tamanho no final do 1º ano	40

1. Introdução

1.1. Contextualização de aplicação do sistema

O ano de 2019 marcou o início de uma jornada inesperada para Ramirez, um talentoso programador mexicano, e entusiasta na área da informática, cuja vida tomou um rumo dramático após ser despedido do seu emprego.

Num dia nublado, o seu telefone tocou. Do outro lado da linha estava Rui, um velho amigo dele. Rui e Ramirez são ambos apaixonados pela área da informática. Rui conhecendo Ramirez e sabendo da sua atual situação apresenta-lhe uma proposta irrecusável que poderia alegrar e mudar completamente a sua vida: a criação de uma agência de detetives privada, a WaveDetectives.

Nos meses seguintes, a agência começou a ganhar forma, resolvendo casos pequenos e às vezes sem retorno monetário.

Rui e Ramirez sentiam que faltava algo para que eles conseguissem resolver os casos de forma mais eficiente e também os casos mais desafiantes que ainda estavam por vir.

A maior dificuldade deles seria interligar casos passados e detalhes atuais para encontrar semelhanças e pistas, eles perceberam que precisavam de uma ferramenta que os ajudasse a fornecer as conexões necessárias.

Certo dia, eles receberam uma chamada da polícia. Era o inspetor Mendes, um velho conhecido de Rui, o mesmo inspetor que outrora o apanhava na sua época ingloria da pirataria. O mesmo solicitou ajuda num caso urgente de roubos de dados bancários que afetava milhares de pessoas. Perante este novo desafio, a WaveDetectives aceitou a proposta, mas sabiam que perante este desafio tão desafiador teriam de mudar algo na sua empresa e foi assim que decidiram implementar uma base de dados robusta e sofisticada, capaz de analisar e relacionar todos os casos passados e presentes.

A implementação de uma base de dados é uma solução estratégica para o desafio de interligar os casos passados com informações atuais. Ao identificar semelhanças e pistas, esta ferramenta não só facilitará a resolução de casos pendentes, mas também proporcionará uma gestão mais eficaz no futuro. Com o aumento previsto no volume de casos e no número de funcionários, uma base de dados será essencial para manter um controle rigoroso e melhorar a eficiência operacional. Apresentação do Caso de Estudo

1.1 Motivação e Objetivos do Trabalho

A motivação para a implementação de uma base de dados na WaveDetectives é clara: a necessidade de uma ferramenta que possa interligar casos passados e detalhes atuais de modo eficiente de modo que a missão do encontro de semelhanças e pistas seja simplificada. A colaboração com o inspetor Mendes em um caso de roubo de dados bancários destacou a urgência de uma ferramenta que pudesse analisar e relacionar informações de forma eficiente.

A agência, para além de procurar a interligação de casos que permitem identificar padrões e conexões entre casos passados e presentes garantindo a melhoria da capacidade de resolução de casos, procura também gerir as informações de melhor forma possível proporcionando assim uma gestão centralizada e organizada das informações dos casos garantindo o acesso rápido aos dados necessários.

1.2 Análise da Viabilidade do processo

Ramirez e Rui acreditam que com a implementação de uma base de dados para a sua agência de detetives conseguirão:

1. Redução de Custos Operacionais: A previsão e análise de tendências de casos, bem como a gestão eficiente de recursos proporcionada pela base de dados, levará a uma redução significativa dos custos operacionais, tornando a WaveDetectives mais competitiva.
2. Aumento da Eficiência na Resolução de Casos: A base de dados permitirá à WaveDetectives aumentar o número de casos resolvidos, melhorando assim a taxa de sucesso e atraindo mais clientes que procuram serviços de investigação de alta qualidade.
3. Colaboração com as Autoridades: A capacidade de resolver casos complexos, como o apresentado pelo inspetor Mendes, não só reforçará a reputação da WaveDetectives, mas também estabelecerá a agência como um parceiro valioso para as autoridades policiais. A base de dados será uma ferramenta crucial para ajudar a polícia a resolver crimes e proteger a comunidade, demonstrando o compromisso da WaveDetectives com a justiça e a segurança pública.

4. Disponibilidade de Informações Ampliada: A base de dados oferecerá um repositório extenso de informações sobre casos, padrões criminais, e tendências de investigação, o que é essencial para a preparação e execução de investigações futuras.

1.3 Recursos e Equipa de Trabalho

A organização detalhada dos recursos humanos e materiais necessários, bem como as responsabilidades das equipas internas e externas em um contexto específico. Ela pode estar relacionada à investigação criminal, ao desenvolvimento de software ou a qualquer outro projeto que envolva colaboração e recursos.

Recursos		Equipa de trabalho	
Recursos Humanos	Recursos Materiais	Equipa interna	Equipa externa
Desenvolvedor de Software; Equipa de análise; Agentes Policiais; Vítima.	Dispositivos com acesso à internet; Software (brModelo e MySQL Workbench);	Escrita do processo criminal; Levantamento de informações da vítima; Buscas policiais; Interrogatório aos suspeitos.	Levantamento de requisitos; Modelação do sistema; Validação do sistema; Implementação do sistema.

Tabela 1 – Recursos e equipa de trabalho.

1.4 Plano de execução do Projeto

A WaveDetectives, em colaboração com todos os participantes do projeto, delineou e organizou as atividades necessárias para o desenvolvimento da Base de Dados. Eles criaram um plano de trabalho detalhado e definiram um cronograma para a implementação das tarefas.

O diagrama GANTT ilustra a forma como as diversas etapas do projeto serão realizadas e os períodos definidos para cada tarefa. Estes períodos foram cuidadosamente planeados para garantir que cada tarefa seja concluída de forma eficiente e eficaz.

A primeira fase, “Definição do Sistema”, está programada para ser concluída num período de 8 dias. Este tempo foi alocado para garantir a construção da história e que tenhamos uma compreensão clara e completa do sistema que estamos a implementar.



Figura 1 - Modelo de Gantt da “Definição do Sistema”.

A segunda fase, “Levantamento e Análise de Requisitos”, está agendada para 4 dias. Este período reflete a necessidade de uma investigação profunda e análise crítica dos requisitos do sistema.

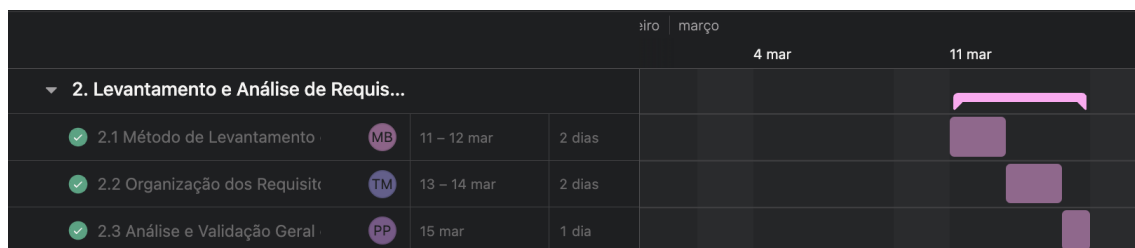


Figura 2 - Modelo de Gantt do “Levantamento e Análise de Requisitos”.

A terceira fase, “Modelagem Concetual”, tem uma duração de 9 dias. Este tempo foi alocado para permitir o desenvolvimento meticuloso do modelo concetual, tal como criar os entidades, atributos e relacionamentos.



Figura 3 - Modelo de Gantt do “Modelação Concetual”.

A quarta fase, “Modelagem Lógica”, está programada para 7 dias. Este período permite-nos desenvolver um modelo lógico detalhado e preciso, verificando e validado o modelo.



Figura 4 - Modelo de Gantt da “Modelação Lógica”.

Finalmente, a quinta fase, “Conclusões e Trabalho Futuro”, é alocada num período de 6 dias. Este tempo garante que as nossas conclusões sejam bem fundamentadas e que os trabalhos futuros sejam claramente identificados.



Figura 5 - Modelo de Gantt da “Conclusões e Trabalho Futuro”.

Cada uma destas fases é crucial para o sucesso do nosso projeto e acreditamos que os períodos alocados para cada tarefa são adequados para garantir que cada tópico seja tratado de forma eficaz.

2 Levantamento e Análise de Requisitos

2.1 Método de Levantamento e de Análise de Requisitos Adotados

Com o objetivo de compreender bem o funcionamento e as necessidades da WaveDetectives, decidimos adotar uma abordagem abrangente para coletar informações. Empregamos uma variedade de métodos de busca e obtenção de dados para abordar todas as perspectivas e cenários possíveis. Este levantamento de informações permitir-nos-á estabelecer um conjunto abrangente de requisitos básicos e não básicos que desejamos implementar na nossa base de dados, corrigindo problemas e erros pré-existentes para garantir toda a confiabilidade e utilidade possível.

Entre estes métodos de recolha de dados, baseamo-nos em:

1. Entrevistas com os Fundadores (Ramirez e Rui):

Justificação: Os fundadores têm uma visão abrangente do negócio e das necessidades da empresa.

Caracterização: Entrevistas estruturadas e informais foram realizadas para compreender as expectativas, desafios e objetivos da empresa em relação à implementação do sistema de gestão de base de dados.

2. Reuniões com a Equipa de Investigação:

Justificação: Os membros da equipa têm insights sobre os fluxos de trabalho diários e as necessidades operacionais.

Caracterização: Reuniões foram conduzidas para discutir os processos existentes, identificar lacunas e áreas de melhoria, e entender como a base de dados poderia apoiar as suas atividades.

3. Análise Documental:

Justificação: Documentos existentes podem conter requisitos implícitos ou explícitos.

Caracterização: Foram analisados documentos como relatórios de casos anteriores, procedimentos operacionais padrões e requisitos legais para identificar requisitos relevantes para a base de dados.

4. Polícia Judiciária:

Justificação: Todas as vítimas que foram à polícia fazer uma denúncia terão suas informações anotadas e teremos acesso a esse "processo".

Caracterização: A polícia vai anotar as informações (especificar) das vítimas que fizeram uma denúncia. Teremos acesso a esse "processo".

2.2 Organização dos Requisitos Levantados

O processo de levantamento de requisitos é uma etapa fundamental no desenvolvimento de sistemas, pois visa compreender e documentar as necessidades e expectativas dos utilizadores.

Para tal, a WaveDetectives recorreu a entrevistas com profissionais experientes em base de dados, questionários a lesados pelo golpe (vítimas) que indicaram os principais aspetos que deviam ser recolhidos sobre estas para que o caso seja solucionado e ainda forneceram indicações de quais elementos deveriam ser recolhidos sobre o suspeito. Para além disso, várias entrevistas ao inspetor Mendes (que foram cruciais para a organização do caso) foram feitas. O inspetor indicou como as informações do suspeito deviam ser constituídas de modo a recolher apenas as informações mais relevantes e ainda afirmou numa das reuniões feitas que a esquadra responsável pelos casos já possuía todas as informações relevantes das torres telefónicas envolvidas. Para além disto tudo, o inspetor ainda forneceu informações sobre quais os dados o processo criminal deveria ser formado de modo a sintetizar todos os aspetos mais importantes a recolher sobre este.

Perante todas as recolhas de dados, Rui e Ramirez decidiram separar os requisitos da seguinte forma:

Requisitos de descrição, para modelagem de dados, definição de estrutura e detalhes dos dados que serão armazenados na base de dados.

Requisitos de controlo, para seguranças de acesso e monitoramento de desempenho.
Requisitos de manipulação, para formas de consulta, listagem de dados e filtragem de dados.

Tipo	Nº	Data	Descrição	Fonte de informação	Analista
Descrição	RD1	15/03/2024	Cada vítima deve ser registada com: NIF, nome, idade, data de nascimento, dados da conta bancária, contacto telefónico.	Vítima	Rui
Descrição	RD2	15/03/2024	Cada suspeito deve ser registado com: id, nome, contacto, descrição.	Vítima	Rui
Descrição	RD3	15/03/2024	O processo criminal causado pelo suspeito deve ser constituído por: id, descrição do incidente, data do incidente, contacto do suspeito, o seu estado ('a decorrer' ou 'finalizado'), data de início, data de fim.	Agente da Polícia	Rui
Descrição	RD4	15/03/2024	A esquadra deve ser representa pelo seu id, nome do coordenador, contacto telefónico, departamento, endereço.	Ramirez	Rui
Descrição	RD5	15/03/2024	Agente é o responsável pelo caso e é registado com um número policial, nome, cargo, categoria, contacto telefónico.	Ramirez	Rui
Descrição	RD6	15/03/2024	Para a chamada recebida pela vítima deve ser registado o seguinte: identificador da chamada, duração da chamada, número (quer de origem quer de destino), informação.	Vítimas	Rui

Tipo	Nº	Data	Descrição	Fonte de informação	Analista
Descrição	RD7	15/03/2024	Todas as torres devem ser conter: id, alcance, coordenadas.	Ramirez	Rui
Descrição	RD8	19/03/2024	A vítima pode possuir vários contactos telefónicos.	Vítimas	Rui
Descrição	RD9	19/03/2024	O agente policial pode possuir vários contactos telefónicos.	Ramirez	Rui
Descrição	RD10	19/03/2024	A esquadra pode possuir vários contactos telefónicos.	Ramirez	Rui
Descrição	RD11	19/03/2024	O processo criminal descreve detalhadamente como foi aplicado o golpe.	Ramirez	Rui
Descrição	RD12	19/03/2024	O agente policial baseia-se na experiência de casos anteriores que foram solucionados para solucionar os novos casos.	Ramirez	Rui
Descrição	RD13	19/03/2024	O suspeito procura aplicar golpes a pessoas que possuem empregos bem remunerados.	Ramirez	Rui
Descrição	RD14	19/03/2024	As informações fornecidas pela Torre devem ser suficientes para que o suspeito seja apanhado.	Ramirez	Rui

Tipo	Nº	Data	Descrição	Fonte de informação	Analista
Descrição	RD15	19/03/2024	Todas as denúncias feitas resultaram de casos em que tivesse sido, de facto, aplicado um golpe.	Ramirez	Rui
Descrição	RD16	19/03/2024	A chamada é o método pelo qual o suspeito efetua o roubo à vítima.	Ramirez	Rui
Descrição	RD17	19/03/2024	Um endereço é formado pela localidade, pelo código postal e nome da rua.	Ramirez	Rui

Tabela 2 – Requisitos de descrição.

Tipo	Nº	Data	Descrição	Fonte de informação	Analista
Manipulação	RM1	19/03/2024	Apenas o agente policial é capaz de consultar os dados do processo criminal.	Ramirez	Rui
Manipulação	RM2	19/03/2024	Deve ser possível listar todas as torres associadas à localização do suspeito.	Ramirez	Rui
Manipulação	RM3	19/03/2024	Deve ser possível listar todos os processos criminais.	Ramirez	Rui
Manipulação	RM4	19/03/2024	Deve ser possível listar todo o histórico de golpes sofridos de uma vítima.	Ramirez	Rui

Tipo	Nº	Data	Descrição	Fonte de informação	Analista
Manipulação	RM5	19/03/2024	Deve ser possível filtrar todos os casos em que dado suspeito participou.	Ramirez	Rui
Manipulação	RM6	19/03/2024	Deve ser possível registar o nome de todas as esquadras que concluíram a caça ao suspeito com sucesso	Ramirez	Rui

Tabela 3 – Requisitos de manipulação.

Tipo	Nº	Data	Descrição	Fonte de informação	Analista
Controlo	RC1	15/03/2024	O agente responsável pelo caso é o único que pode atualizar o mesmo.	Ramirez	Rui

Tabela 4 – Requisitos de controlo.

2.3 Análise e Validação Geral dos Requisitos

Após a realização do levantamento de requisitos pelos métodos selecionados, organizamos e examinamos todos os requisitos identificados com atenção, assegurando que não houvesse falhas, contradições, duplicidades ou ambiguidades. Realizamos uma reunião com os integrantes envolvidos para validar os requisitos. Durante a discussão, surgiu uma dúvida significativa sobre o funcionamento das antenas de telecomunicações. Decidimos, para fins de simplificação da base de dados, que as antenas criariam um perímetro ao redor, sinalizando a posição do suspeito. Após uma análise minuciosa e revisão de cada requisito. Por fim, os requisitos foram aprovados por todos os participantes.

3 Modelação Concetual

3.1 Apresentação da Abordagem de Modelação

Na modelação concetual, utilizámos um diagrama ER para representar as entidades, os seus atributos e as relações entre elas, com o objetivo de estruturar informações relacionadas a um crime de roubo de informações bancárias ocorrido durante uma chamada entre o ladrão e a vítima. A finalidade desse diagrama é criar uma visualização clara e organizada dos elementos envolvidos, permitindo uma compreensão abrangente do processo. Para criar o modelo conceitual, optámos pela ferramenta brModelo, que possibilitou a criação e edição do diagrama de forma eficiente, garantindo a precisão e a legibilidade.

3.2 Identificação e Caracterização das Entidades

Após a análise dos requisitos previamente estabelecidos, foram identificadas as seguintes entidades essenciais para operacionalizar a WaveDetectives e conduzir todo o seu processo de trabalho e investigação.

Designação	Descrição	Sinónimos	Ocorrência
Vítima	Pessoa que irá denunciar o crime de roubo, para a polícia, e transmitir todos as informações e detalhes, acerca do mesmo.	Cliente, Lesado	Cada vítima tem um número de identificação próprio, informações pessoais e dados financeiros, para ajudar a relacionar dados essenciais para principais alvos do crime.
Suspeito	Pessoa que é alvo de suspeita de envolvimento em um crime.	Criminoso, Ladrão	Cada suspeito executará uma chamada, que será o método utilizado para perpetrar o roubo.
Processo Criminal	Conjunto de procedimentos e informações, para investigar e julgar o crime. Sendo mais fácil assim detetar o criminoso.	-	Todos os processos criminais serão abertos quando é ocorrido um roubo e transmitido por uma vítima
Esquadra	Força de segurança responsável pela aplicação da lei e investigação de crimes.	Polícia Judiciária	As esquadras terão uma identificação única para cada uma.

Designação	Descrição	Sinónimos	Ocorrência
Chamada	Comunicação telefónica, que o suspeito realiza, para realizar o seu roubo.	Telefonema, Ligação Telefónica	Cada chamada será o método para perpetuar o roubo
Torre	Estrutura de comunicação que permite rastrear chamadas telefónicas.	-	A torre de transmissão será essencial para detetar de onde foi localizada a chamada. Cada torre terá um identificador próprio.
Agente da Polícia	Funcionário da polícia, encarregue do seu devido posto.	Polícia	Cada Agente terá o encargo pela para descoberta do criminoso.

Tabela 5 – Identificação e caracterização das entidades

- ☐ **Vítima** – Foi necessário criar uma entidade vítima, segundo o levantamento dos requisitos RD1, RD8 e RM4;
- ☐ **Suspeito** – Foi necessário criar uma entidade suspeito, segundo o levantamento dos requisitos RD2, RD13, RM2, RM4 e RM5;
- ☐ **Processo Criminal** – Foi necessário criar uma entidade Processo Criminal, segundo o levantamento dos requisitos RD3, RD11, RD15, RM1, RM4, RM5 e RC1;
- ☐ **Esquadra** – Foi necessário criar uma entidade esquadra, segundo o levantamento dos requisitos RD4, RD10 e RM6;
- ☐ **Chamada** – Foi necessário criar uma entidade chamada, segundo o levantamento dos requisitos RD6, RD13;
- ☐ **Torre** – Foi necessário criar uma entidade torre, segundo o levantamento dos requisitos RD7, RD14, RM2;
- ☐ **Agente da polícia** – Foi necessário criar uma entidade agente da polícia, segundo o levantamento dos requisitos RD5, RD9, RD12, RM1, RM4 e RC1.

3.3 Identificação e Caracterização dos Relacionamentos

Após a análise dos requisitos previamente estabelecidos, foram identificados os seguintes relacionamentos essenciais, para as ligações necessárias, que cada entidade precisa.

Entidade	Relacionamento	Cardinalidade	Participação	Entidade
Processo Criminal	envolve	N:1	T:T	Vítima
Agente da polícia	escreve	1:N	P:T	Processo Criminal
Agente da polícia	pertence	N:1	T:T	Esquadra
Processo Criminal	envolve	N:1	T:P	Suspeito
Suspeito	faz	1:N	T:T	Chamada
Vítima	recebe	1:N	T:T	Chamada
Chamada	emitida	N:1	T:P	Torre

Tabela 6 – Identificação e caracterização de relacionamentos

O relacionamento “envolve” é necessário pois a vítima é a fonte de informação de alguns detalhes cruciais do roubo. Os requisitos que justificam a criação deste relacionamento são: RD11.

O relacionamento “escreve” é necessário pois o processo criminal apenas pode ser escrito pelo Agente, com todos os detalhes do caso. Os requisitos que justificam a criação deste relacionamento são: RD11, RD12, RM1, RM3 e RC1.

O relacionamento “pertence” é necessário pois é importante saber qual a esquadra que um determinado agente está inserido. Os requisitos que justificam a criação deste relacionamento são: RM6.

O relacionamento “envolve” é necessário pois o principal objetivo do caso é descobrir o suspeito e através do processo teremos todos os detalhes anotados para a resolução do caso. Os requisitos que justificam a criação deste relacionamento são: RD11 e RM5.

O relacionamento “faz” é necessário pois o principal método de roubo do suspeito é através de chamadas telefônicas. Os requisitos que justificam a criação deste relacionamento são: RD16.

O relacionamento “recebe” é necessário pois todas as vítimas foram alvo de uma chamada fraudulenta. Os requisitos que justificam a criação deste relacionamento são: RD16.

O relacionamento “emitida” é necessário pois teremos que interligar a chamada com uma torre, que será a fonte de transmissão, necessária para se realizar a mesma, e com ela conseguirmos

saber a localização de onde foi realizada a chamada. Os requisitos que justificam a criação deste relacionamento são: RD14 e RM2.

3.4 Identificação e Caracterização dos Atributos das Entidades e dos Relacionamentos

Após a análise dos requisitos previamente estabelecidos, na modelação, foram observadas várias formas de interligação entre as entidades. Consequentemente, apresentamos uma análise elucidativa para cada uma dessas conexões

Entidade / Relacionamento	Atributo	Tipo de atributo	Descrição	Tipo e Tamanho	Nulo	Exemplo
Processo Criminal	Id	Chave	Identificador do processo	INT	N	232
	Descrição do incidente	Simples	Detalhes do incidente	TEXT	N	A vítima relata que recebeu uma chamada a dizer que era do banco. A vítima passou todos os dados. Passado uma semana a vítima reparou numa transferência desconhecida no valor de 2908€
	Data incidente	Simples	Data em que ocorreu o roubo	DATE	N	03/10/24
	Contacto suspeito	Simples	Nº telefónico do ladrão	VARCHAR (30)	S	914523457
	Status	Simples	Se está finalizado ou ainda a ocorrer o processo	BOOLEAN	N	FALSE
	Data inicio	Simples	Data de criação do processo	DATE	N	02/11/24
	Data fim	Simples	Data é que foi finalizado o processo	DATE	S	22/11/24

Entidade / Relacioname nto	Atributo	Tipo de atributo	Descrição	Tipo e Tamanho	Nulo	Exemplo
Vítima	NIF	Chave	Nº de identificação	INT	N	399954851
	Nome	Simples	Nome completo	VARCHAR (75)	N	Carlos António Bandeira
	Idade	Simples	Idade	INT	S	34
	Data Nasciment o	Simples	Data de nascimento da vítima	DATE	N	12/01/90
	Conta Bancária	Simples	Conta bancária que foi alvo de roubo	VARCHAR (100)	N	PT5000270000000123456 7833
	Contacto	Multi- valor	Contacto telefónico da vítima	VARCHAR (30)	N	923847493
	Emprego	Simples	Emprego da Vítima	VARCHAR (75)	N	Professor
Suspeito	Id	Chave	Identificador do suspeito	INT	N	345
	Nome	Simples	Nome completo	VARCHAR (75)	N	Leandro Manuel Silva
	Contacto	Simples	Contacto telefónico da suspeito	VARCHAR (30)	N	934254239
	Localizaçã o	Simples	Área de onde poderá estar o suspeito	VARCHAR (100)	S	Centro
	Descrição	Simples	Detalhes de como poderá ser o suspeito	TEXT	S	Voz grossa, ruído no fundo de vozes,(...)
Esquadra	Id	Chave	Identificador da polícia	INT	N	56
	Departame nto	Simples	Departamento da polícia	VARCHAR (75)	S	Polícia de Alijó
	Endereço	Compost o	Rua	VARCHAR (75)	S	Rua do Salgueiro, 311
			Localidade	VARCHAR (75)	S	Alijó

Entidade / Relacioname nto	Atributo	Tipo de atributo	Descrição	Tipo e Tamanho	Nulo	Exemplo
			Código Postal	VARCHAR (75)	S	4650-345
	Contacto	Multi- valor	Contacto telefónico da policia	VARCHAR (30)	S	259 950 543
	Nome_coo rdenador	Simples	Coordenador do caso na policia	VARCHAR (75)	N	Capitão Ico
Chamada	Id	Chave	Identificador da chamada	INT	N	9
	Numero Destino	Simples	Contacto telefónico que ligou	VARCHAR (75)	N	923847493
	Numero Origem	Simples	Contacto telefónico que recebeu a chamda	VARCHAR (75)	N	934254239
	Duração	Simples	Tempo de duração	VARCHAR (75)	N	5min 36s
	Info	Simples	Informação da chamada, incluindo data e hora	TIMESTA MP	N	22/11/24
Torre	Id	Chave	Identificador da Torre	INT	N	98
	Coordenad as	Simples	Coodernadas onde se localiza a Torre	VARCHAR (100)	N	41.40338, 2.17403
	Alcance	Simples	Alcance que a torre consegue emitir chamadas	INT	N	50km
Agente da Policia	Nº policial	Chave	Identificador da Operadora	INT	N	653
	Nome	Simples	Nome da operadora	VARCHAR (75)	N	Vodafone
	Cargo	Simples	Hierarquia do Agente	VARCHAR (75)	S	Comissário

Entidade / Relacioname nto	Atributo	Tipo de atributo	Descrição	Tipo e Tamanho	Nulo	Exemplo
	Contacto	Multi- valor	Contactos da operadora	VARCHAR (30)	N	911 691 200
	Categoria	Simples	Especialidade do Agente	VARCHAR (75)	N	Desarmar bombas

Tabela 7 – Identificação e descrição das entidades e dos seus atributos.

Os atributos da entidade Processo Criminal foram criados, segundo o levantamento dos seguintes requisitos:

- ☐ Id – RD3
- ☐ Descrição do incidente – RD3, RD11
- ☐ Data do incidente – RD3
- ☐ Contacto suspeito – RD3
- ☐ Status – RD3
- ☐ Data início – RD3
- ☐ Data fim – RD3

Os atributos da entidade Vítima foram criados, segundo o levantamento dos seguintes requisitos:

- ☐ NIF – RD1
- ☐ Nome – RD1
- ☐ Idade – RD1
- ☐ Data Nascimento – RD1
- ☐ Conta Bancária – RD1
- ☐ Contacto – RD1, RD8
- ☐ Emprego – RD16

Os atributos da entidade Suspeito foram criados, segundo o levantamento dos seguintes requisitos:

- ☐ Id – RD2
- ☐ Nome – RD2
- ☐ Contacto – RD2
- ☐ Localização – RM2
- ☐ Descrição – RD2

Os atributos da entidade Esquadra foram criados, segundo o levantamento dos seguintes requisitos:

- ☐ Id – RD4

- ☐ Departamento – RD4
- ☐ Endereço – RD4
- ☐ Contacto – RD4, RD10
- ☐ Nome_coordenador – RD4

Os atributos da entidade Chamada foram criados, segundo o levantamento dos seguintes requisitos:

- ☐ Id – RD6
- ☐ Número Destino – RD6
- ☐ Número Origem – RD6
- ☐ Duração – RD6
- ☐ Info – RD6

Os atributos da entidade Torre foram criados, segundo o levantamento dos seguintes requisitos:

- ☐ Id- RD7
- ☐ Coordenadas – RD7, RD14
- ☐ Alcance – RD7, RD14, RM2

Os atributos da entidade “Agente da polícia” foram criados, segundo o levantamento dos seguintes requisitos:

- ☐ N.º do agente da polícia – RD5
- ☐ Nome – RD5
- ☐ Cargo – RD5, RD12
- ☐ Contacto – RD5, RD9
- ☐ Categoria – RD5

3.5 Apresentação e Explicação do Diagrama ER Produzido

Após uma explicação detalhada de todas as entidades presentes no nosso sistema, cada um dos relacionamentos entre elas e os seus respetivos atributos, apresentamos visualmente o seguinte modelo concetual.

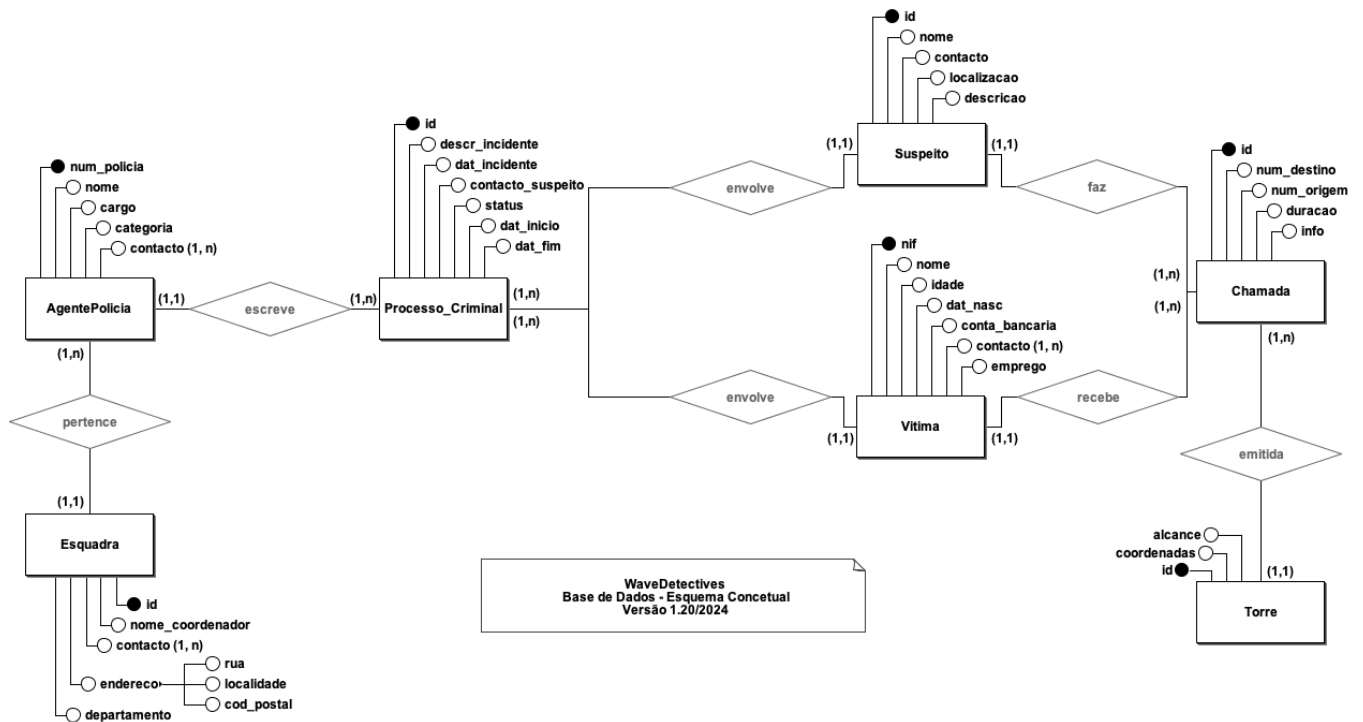


Figura 6 - Modelo Concetual

Desta forma, após obter uma melhor compreensão do modelo que pretendemos implementar na nossa base de dados para a agência de detetives WaveDetectives, podemos avançar para a modelação lógica.

4 Modelação Lógica

4.1 Construção e Validação do Modelo de Dados Lógico

Começamos por definir cada uma das entidades do modelo concetual, adicionando os respetivos atributos: atributos compostos são vistos como atributos simples, e atributos multi-valor possuem uma tabela própria que possui como chave estrangeira a chave primária da entidade; os relacionamentos 1:N também foram tratados da mesma forma. Consideramos o modelo produzido válido, uma vez que foi produzido com base no modelo concetual, e validado novamente por verificação da satisfação dos requisitos levantados previamente.

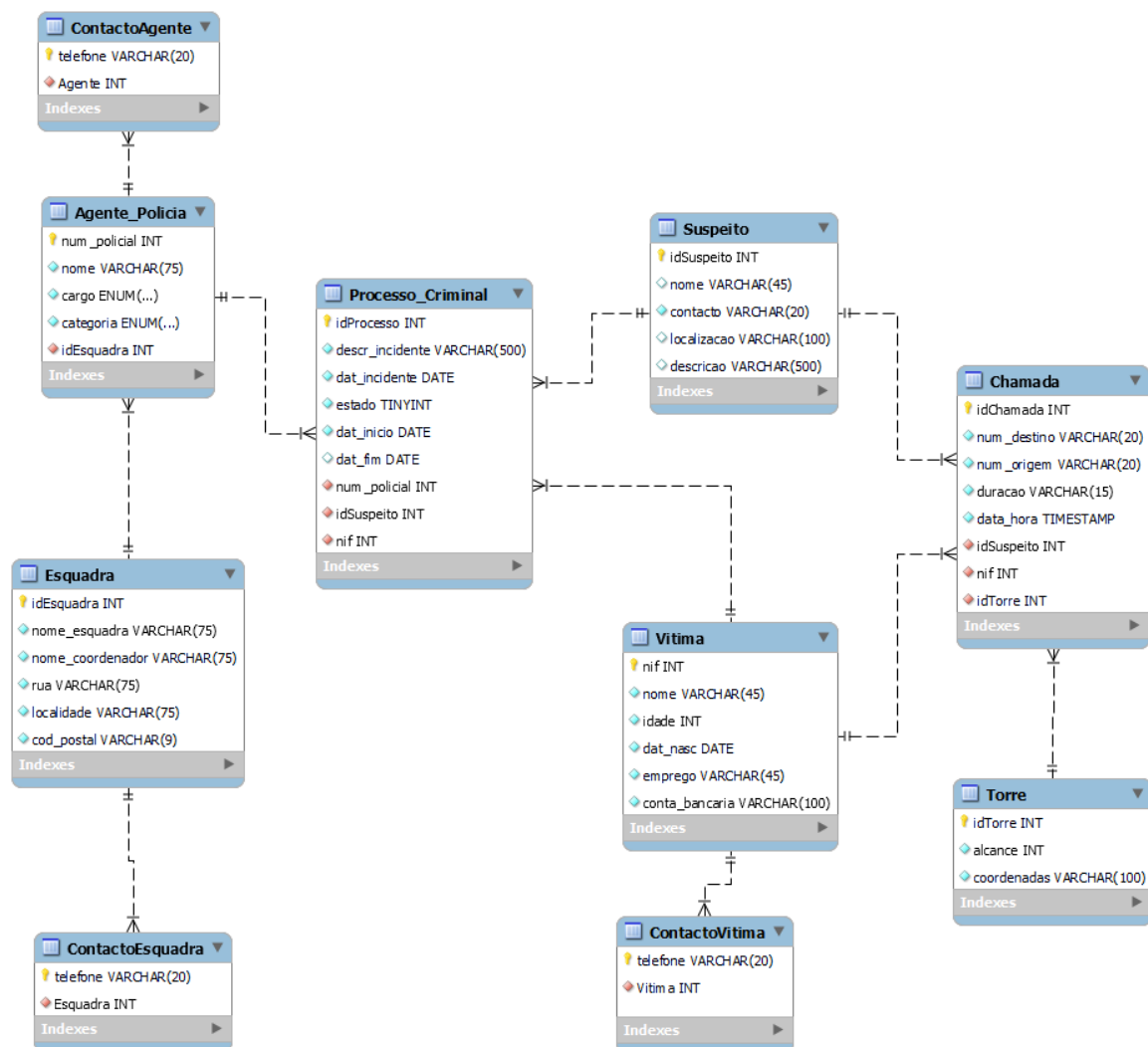


Figura 7 - Modelo Lógico

4.2 Apresentação e Explicação do Modelo Lógico Produzido

O nosso modelo lógico foi construído a partir do nosso modelo concetual. As entidades são convertidas em tabelas, assim como os relacionamentos que possuem requisitos associados ou ainda os atributos multi-valor.

O identificador de uma entidade passa a designar-se uma Primary Key ou Foreign Key caso o identificador de uma entidade esteja presente noutra sob a forma de tabela. Assim, após feita a conversão obtemos dez tabelas, das quais três foram originadas a partir de atributos multi-valor.

Seguem-se essas tabelas:

A entidade AgentePolicia deu origem a uma outra tabela contactoAgente devido ao seu atributo “contacto” ser multi-valor.

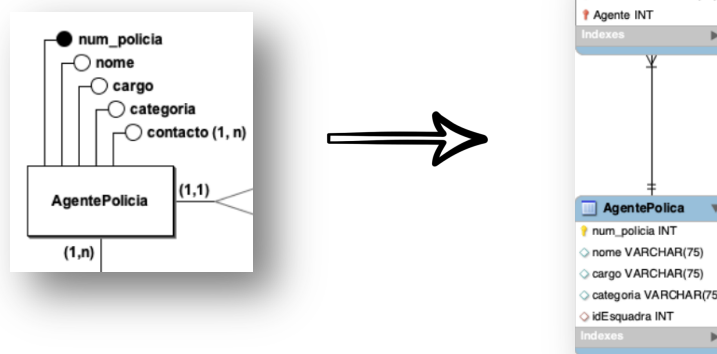


Figura 8 – Transformação da entidade AgentePolicia no modelo concetual para modelo lógico.

A entidade Esquadra deu origem a uma outra tabela contactoEsquadra devido à mesma razão explicada em cima, devido ao seu atributo multi-valor e ainda o seu atributo composto endereço é representado na tabela Esquadra pelos seus constituintes, (“rua”, “localidade”, “cod_postal”).

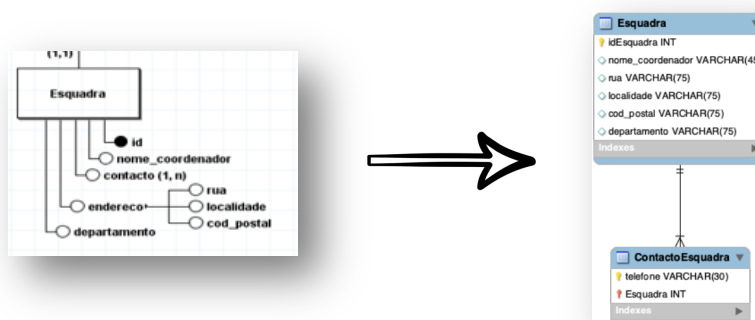


Figura 9 - Transformação da entidade Esquadra no modelo concetual para modelo lógico.

A entidade Vítima também deu origem a outra tabela contactoVítima pelas mesmas razões, o seu atributo multi-valor “contacto”.

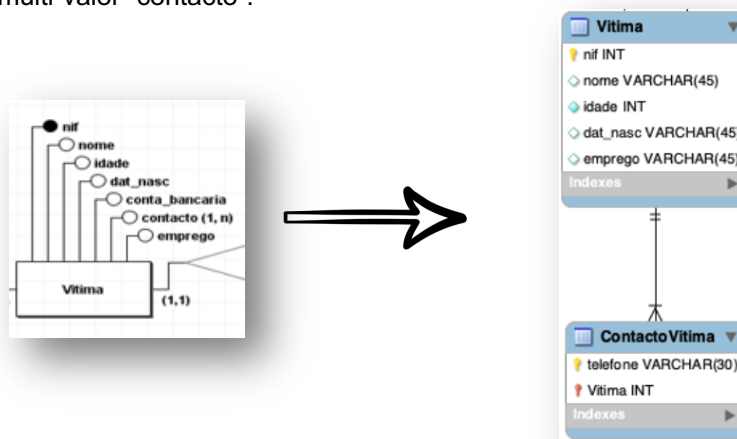


Figura 10 - Transformação da entidade Vítima no modelo concetual para modelo lógico.

O Processo_Criminal é a tabela com mais colunas, mas duas delas podem ter valores nulos, nas linhas: Contacto_Suspeito e Data_Fim. O contacto do suspeito pode ser nulo pois pode ser desconhecido, a Data_fim também pode ser nulo, pois se o processo ainda estiver em curso não terá uma data de fim.



Figura 11 - Transformação da entidade Processo_Criminal no modelo concetual para modelo lógico.

A entidade Suspeito deu origem a uma tabela Suspeito com os seus respetivos atributos.

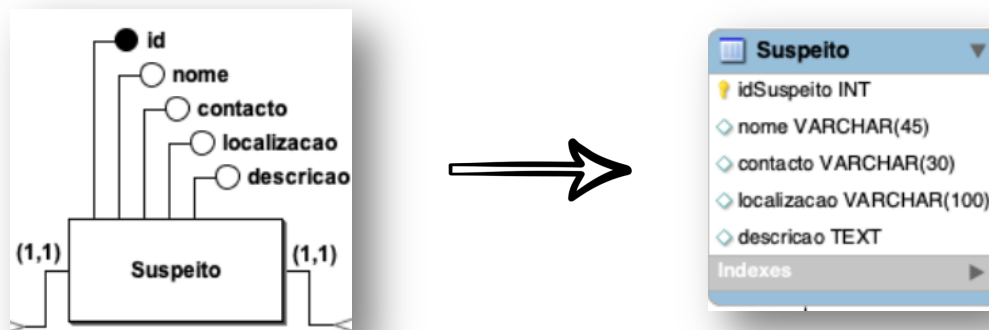


Figura 12 - Transformação da entidade Suspeito no modelo concetual para modelo lógico.

A entidade Chamada deu origem a uma tabela Chamada com os seus respetivos atributos.

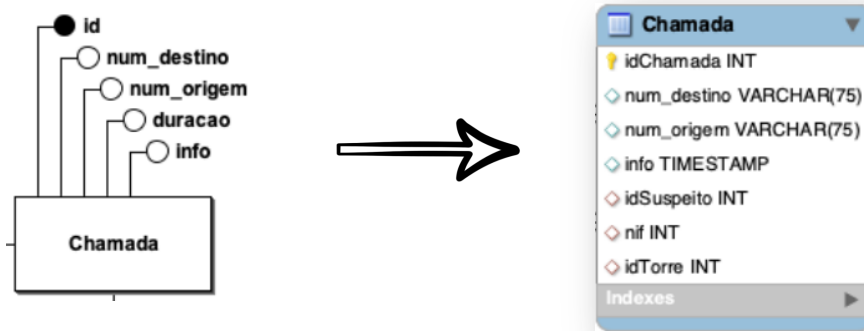


Figura 13 - Transformação da entidade Chamada no modelo conceitual para modelo lógico.

E por fim a entidade Torre deu origem também a uma tabela Torre com os seus respetivos atributos.

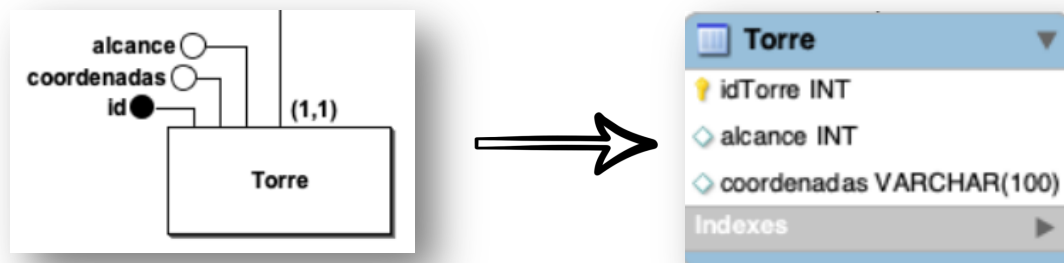


Figura 14 - Transformação da entidade Torre no modelo conceitual para modelo lógico.

4.3 Normalização de Dados

O nosso modelo de base de dados cumpre as três formas normais (1FN, 2FN e 3FN), o que indica que está devidamente normalizado.

Na Primeira Forma Normal (1FN), cada tabela do nosso modelo tem uma chave primária e todos os seus atributos são atômicos, sem valores duplicados ou atributos compostos ou multi-valor.

Na Segunda Forma Normal (2FN), todos os atributos não-chave nas nossas tabelas dependem unicamente da chave primária da respetiva tabela.

Finalmente, na Terceira Forma Normal (3FN), verificamos que não existem atributos no nosso modelo que possam ser derivados de outros atributos que não a chave primária.

Portanto, podemos afirmar que o nosso modelo está normalizado, pois cumpre todas as três formas normais. Esta normalização ajuda a prevenir a redundância de dados e a melhorar o desempenho do modelo, evitando anomalias na inclusão, exclusão e modificação de registos.

4.4 Validação do Modelo com Interrogações do Utilizador

Para validar o nosso modelo de base de dados através de várias interrogações do utilizador. Estas interrogações foram concebidas para testar a eficácia e a precisão do nosso modelo em responder a perguntas reais que podem surgir durante a utilização da base de dados. Ao responder corretamente a estas interrogações, podemos confirmar que o nosso modelo está bem estruturado e pronto para ser implementado. As interrogações do utilizador que vamos utilizar para a validação do nosso modelo são as seguintes:

1. Listar em quais casos um determinado agente está envolvido.

Para determinar em quais casos um agente específico está envolvido, realizamos uma junção entre a tabela Caso e a tabela Agente, com base no ID do agente. Em seguida, filtramos os resultados usando uma condição de igualdade com o ID do agente que estamos investigando, resultando numa lista de todos os casos em que o agente está envolvido.

2. Listar todas as chamadas associadas a uma determinada torre.

Para determinar quais chamadas estão associadas a uma torre específica, realizamos uma junção entre a tabela Chamada e a tabela Torre, com base no ID da torre. Em seguida, filtramos os resultados usando uma condição de igualdade com o ID da torre que estamos investigando, resultando numa lista de todas as chamadas associadas a essa torre.

3. Listar todos os processos criminais associados a uma determinada vítima.

Para determinar quais processos criminais estão associados a uma vítima específica, realizamos uma junção entre a tabela Processo Criminal e a tabela Vítima, com base no NIF da vítima. Em seguida, filtramos os resultados usando uma condição de igualdade com o NIF da vítima que estamos investigando, resultando numa lista de todos os processos criminais associados a essa vítima.

4. Listar os nomes dos policiais que pertencem a uma determinada esquadra.

Para determinar quais policiais estão lotados numa esquadra específica, realizamos uma junção entre a tabela Agente e a tabela Esquadra, com base no ID da esquadra. Em seguida, filtramos os resultados usando uma condição de igualdade com o ID da esquadra que estamos investigando, resultando numa lista dos nomes de todos os policiais lotados nessa esquadra.

5 Implementação Física

5.1 Apresentação e explicação da base de dados implementada

O processo de construção do esquema físico a partir do modelo lógico, utilizando MySQL Workbench, seguindo diretrizes do esquema lógico, proporciona uma base sólida para a realização de operações complexas, análises detalhadas e manutenção eficiente dos dados, atendendo às necessidades do sistema de gerenciamento de processos criminais de forma robusta e escalável.

Para começarmos a criar um esquema de base de dados, inicialmente precisamos inicializar um SCHEMA caso não exista, neste caso criar o esquema WaveDetectives.

```
CREATE SCHEMA IF NOT EXISTS WaveDetectives;  
USE WaveDetectives;
```

5.1.1 Tabela Esquadra

Para construção da Tabela Esquadra, foram inicialmente definidos todos os atributos constituintes de cada tabela. Neste caso, foram definidos o idEsquadra, nome_coordenador, rua, localidade, cod_postal e nome_esquadra. Foi definido como Primary Key o idEsquadra, com os atributos UNSIGNED e AUTO_INCREMENT para garantir unicidade e eficiência na indexação, gerando automaticamente um novo valor para a chave primária sempre que um novo registro é inserido. O mecanismo de armazenamento escolhido foi o InnoDB pela sua grande capacidade no gerenciamento de transações, economia de espaço para otimizar o uso do armazenamento e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS Esquadra (  
    idEsquadra INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    nome_coordenador VARCHAR(75) NOT NULL,  
    rua VARCHAR(75) NOT NULL,  
    localidade VARCHAR(75) NOT NULL,  
    cod_postal VARCHAR(9) NOT NULL,  
    nome_esquadra VARCHAR(75) NOT NULL,  
    PRIMARY KEY (idEsquadra)  
)  
ENGINE = InnoDB;
```

5.1.2 Tabela Suspeito

Para construção da Tabela Suspeito, foram definidos os atributos idSuspeito, nome, contacto, localização e descrição. O idSuspeito foi definido como Primary Key, com os atributos UNSIGNED e AUTO_INCREMENT para garantir unicidade e eficiência na indexação. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS Suspeito (  
  idSuspeito INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  nome VARCHAR(45) NULL,  
  contacto VARCHAR(20) NOT NULL,  
  localizacao VARCHAR(100) NULL,  
  descricao TEXT NULL,  
  PRIMARY KEY (idSuspeito)  
)  
ENGINE = InnoDB;
```

5.1.3 Tabela Vitima

Para construção da Tabela Vitima, foram definidos os atributos nif, nome, idade, dat_nasc, emprego e conta_bancaria. O nif foi definido como Primary Key, garantindo unicidade (número de identificação fiscal). O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS Vitima (  
  nif INT NOT NULL,  
  nome VARCHAR(45) NOT NULL,  
  idade INT NOT NULL,  
  dat_nasc DATE NOT NULL,  
  emprego VARCHAR(45) NOT NULL,  
  conta_bancaria VARCHAR(100) NOT NULL,  
  PRIMARY KEY (nif)  
)  
ENGINE = InnoDB;
```

5.1.4 Tabela Torre

Para construção da Tabela Torre, foram definidos os atributos idTorre, alcance e coordenadas. O idTorre foi definido como Primary Key, com os atributos UNSIGNED e AUTO_INCREMENT para garantir unicidade e eficiência na indexação. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS Torre (
  idTorre INT UNSIGNED NOT NULL AUTO_INCREMENT,
  alcance INT NOT NULL,
  coordenadas VARCHAR(100) NOT NULL,
  PRIMARY KEY (idTorre)
)
ENGINE = InnoDB;
```

5.1.5 Tabela Agente_Policia

Para construção da Tabela Agente_Policia, foram definidos os atributos num_policial, nome, cargo, categoria e idEsquadra. O num_policial foi definido como Primary Key, com os atributos UNSIGNED e AUTO_INCREMENT para garantir unicidade e eficiência na indexação. Foi adicionada uma chave estrangeira idEsquadra referenciando a tabela Esquadra para assegurar que os relacionamentos entre as tabelas permanecem consistentes e facilitarem a navegação e consultas. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS Agente_Policia (
  num_policial INT UNSIGNED NOT NULL AUTO_INCREMENT,
  nome VARCHAR(75) NOT NULL,
  cargo ENUM('Diretor nacional', 'Diretor nacional adjunto', 'Chefe', 'Subchefe',
    'Investigador', 'Coordenador de investigação', 'Assistente') NULL,
  categoria ENUM('Civil', 'Militar', 'Operações Especiais', 'Cibernética',
    'Linha da Frente', 'Intervenção') NOT NULL,
  idEsquadra INT UNSIGNED NOT NULL,
  PRIMARY KEY (num_policial),
  CONSTRAINT fk_idEsquadraAgente FOREIGN KEY (idEsquadra) REFERENCES Esquadra (idEsquadra)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE = InnoDB;
```

5.1.6 Tabela ContactoAgente

Para construção da Tabela ContactoAgente, foram definidos os atributos telefone e Agente_Policia. O telefone foi definido como Primary Key para garantir unicidade. Foi adicionada uma chave estrangeira Agente_Policia referenciando a tabela Agente_Policia para assegurar que os relacionamentos entre as tabelas permanecem consistentes e facilitarem a navegação e consultas. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS ContactoAgente (
    telefone VARCHAR(20) NOT NULL,
    num_policial INT UNSIGNED NOT NULL,
    PRIMARY KEY (telefone),
    CONSTRAINT fk_num_policialContacto FOREIGN KEY (num_policial)
    REFERENCES Agente_Policia (num_policial)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE = InnoDB;
```

5.1.7 Tabela ContactoEsquadra

Para construção da Tabela ContactoEsquadra, foram definidos os atributos telefone e Esquadra. O telefone foi definido como Primary Key para garantir unicidade. Foi adicionada uma chave estrangeira Esquadra referenciando a tabela Esquadra para assegurar que os relacionamentos entre as tabelas permanecem consistentes e facilitarem a navegação e consultas. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS ContactoEsquadra (
    telefone VARCHAR(20) NOT NULL,
    idEsquadra INT UNSIGNED NOT NULL,
    PRIMARY KEY (telefone),
    CONSTRAINT fk_idEsquadraContacto FOREIGN KEY (idEsquadra)
    REFERENCES Esquadra (idEsquadra)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE = InnoDB;
```

5.1.8 Tabela ContactoVitima

Para construção da Tabela ContactoVitima, foram definidos os atributos telefone e Vitima. O telefone foi definido como Primary Key para garantir unicidade. Foi adicionada uma chave estrangeira Vitima referenciando a tabela Vitima para assegurar que os relacionamentos entre as tabelas permanecem consistentes e facilitarem a navegação e consultas. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS ContactoVitima (
    telefone VARCHAR(20) NOT NULL,
    nif INT NOT NULL,
    PRIMARY KEY (telefone),
    CONSTRAINT fk_nifContacto FOREIGN KEY (nif) REFERENCES Vitima (nif)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
ENGINE = InnoDB;
```

5.1.9 Tabela Processo_Criminal

Para construção da Tabela Processo_Criminal, foram definidos os atributos idProcesso, descr_incidente, dat_incidente, contacto_suspeito, estado, dat_inicio, dat_fim, num_policial, idSuspeito e nif. O idProcesso foi definido como Primary Key, com os atributos UNSIGNED e AUTO_INCREMENT para garantir unicidade e eficiência na indexação. Foram adicionadas chaves estrangeiras num_policial, idSuspeito e nif referenciando as tabelas Agente_Policia, Suspeito e Vitima, respectivamente, para assegurar que os relacionamentos entre as tabelas permanecem consistentes e facilitarem a navegação e consultas. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS Processo_Criminal (
    idProcesso INT UNSIGNED NOT NULL AUTO_INCREMENT,
    descr_incidente VARCHAR(500) NOT NULL,
    dat_incidente DATE NOT NULL,
    estado TINYINT NOT NULL,
    dat_inicio DATE NOT NULL,
    dat_fim DATE NULL,
    num_policial INT UNSIGNED NOT NULL,
    idSuspeito INT UNSIGNED NOT NULL,
    nif INT NOT NULL,
    PRIMARY KEY (idProcesso),
    CONSTRAINT fk_nifProcesso FOREIGN KEY (nif) REFERENCES Vitima (nif)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_idSuspeitoProcesso FOREIGN KEY (idSuspeito)
        REFERENCES Suspeito (idSuspeito)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_num_policialProcesso FOREIGN KEY (num_policial)
        REFERENCES Agente_Policia (num_policial)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
ENGINE = InnoDB;
```

5.1.10 Tabela Chamada

Para construção da Tabela Chamada, foram definidos os atributos idChamada, num_destino, num_origem, info, idSuspeito, nif e idTorre. O idChamada foi definido como Primary Key, com os atributos UNSIGNED e AUTO_INCREMENT para garantir unicidade e eficiência na indexação. Foram adicionadas chaves estrangeiras idSuspeito, nif e idTorre referenciando as tabelas Suspeito, Vitima e Torre, respectivamente, para assegurar que os relacionamentos entre as tabelas permanecem consistentes e facilitarem a navegação e consultas. O mecanismo de armazenamento escolhido foi o InnoDB pela sua robustez no gerenciamento de transações e suporte a chaves estrangeiras.

```
CREATE TABLE IF NOT EXISTS Chamada (  
  idChamada INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  num_destino VARCHAR(20) NOT NULL,  
  num_origem VARCHAR(20) NULL,  
  duracao VARCHAR(15) NOT NULL,  
  data_hora TIMESTAMP NOT NULL,  
  idSuspeito INT UNSIGNED NOT NULL,  
  nif INT NOT NULL,  
  idTorre INT UNSIGNED NOT NULL,  
  PRIMARY KEY (idChamada),  
  CONSTRAINT fk_idSuspeitoChamada FOREIGN KEY (idSuspeito)  
  REFERENCES Suspeito (idSuspeito)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT fk_nifChamada FOREIGN KEY (nif) REFERENCES Vitima (nif)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT fk_idTorreChamada FOREIGN KEY (idTorre) REFERENCES Torre (idTorre)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION  
)  
ENGINE = InnoDB;
```

5.2 Criação de utilizadores da base de dados

```
-- Rui Analista de Dados  
CREATE USER 'rui'@'localhost' IDENTIFIED WITH mysql_native_password BY 'RuiSenha123!';  
GRANT SELECT ON WaveDetectives.* TO 'rui'@'localhost';  
  
-- Ramirez "Admin"  
CREATE USER 'Ramirez'@'localhost' IDENTIFIED WITH mysql_native_password BY 'RamirezSenha123!';  
GRANT ALL PRIVILEGES ON WaveDetectives.* TO 'Ramirez'@'localhost' WITH GRANT OPTION;  
  
-- Polícias  
CREATE USER 'policia'@'localhost' IDENTIFIED WITH mysql_native_password BY 'PoliciaSenha123!';  
GRANT SELECT, INSERT, UPDATE ON WaveDetectives.* TO 'policia'@'localhost';  
  
FLUSH PRIVILEGES;
```

A criação de utilizadores é essencial para assegurar a segurança, a integridade e a eficiência no acesso aos dados. Diferentes utilizadores podem ter diferentes necessidades e níveis de acesso, desde administradores que precisam de controle total sobre o sistema até analistas de dados e outros funcionários que necessitam apenas de permissões específicas. Definir corretamente os perfis de utilização e as permissões de trabalho garante que cada utilizador possa executar suas funções de maneira eficiente, ao mesmo tempo em que protege os dados contra acessos não autorizados e operações indevidas.

Através da construção de utilizadores na nossa base de dados definimos que:

Rui é um analista de dados que precisa consultar informações da base de dados. Por isso, ele recebe a permissão `SELECT`, que permite apenas a leitura dos dados.

Ramirez é o administrador do sistema, com permissão completa (`ALL PRIVILEGES`) para executar qualquer operação no base de dados. O `'WITH GRANT OPTION'` permite que ele conceda permissões a outros utilizadores.

A última parte é destinada aos polícias que necessitam consultar (`SELECT`), inserir (`INSERT`) e atualizar (`UPDATE`) dados na base de dados, mas não têm permissão para eliminar ou fazer alterações na estrutura do banco de dados.

5.3 Povoamento da base de dados

5.3.1 Inserção de Dados com Instruções SQL

Na foto em baixo vemos um exemplo de uma instrução em SQL de inserção de dados, neste caso a tabela "Vitima" armazenar informações sobre vítimas, incluindo o número de identificação fiscal (nif), nome, data de nascimento, profissão e conta bancária de cada vítima. O comando SQL `INSERT` faz isso mesmo, adiciona os dados nessa mesma tabela. O mesmo acontece para a tabela "ContactoVitima" e na tabela "Torre". Ou seja, os comandos SQL fornecidos inserem dados de exemplo nas tabelas do banco de dados, cada uma armazenando informações específicas.

```
-- Inserir dados na tabela Vitima
INSERT INTO Vitima (nif, nome, dat_nasc, emprego, conta_bancaria)
VALUES
  (123456789, 'Ana Silva', '1993-05-14', 'Engenheira', 'PT50000201231234567890154'),
  (234567890, 'Bruno Costa', '1978-08-20', 'Professor', 'PT50000201231234567890155'),
  (345678901, 'Carla Ferreira', '1995-02-10', 'MÃ©dica', 'PT50000201231234567890156'),
  (456789012, 'David Martins', '1973-11-25', 'Advogado', 'PT50000201231234567890157'),
  (567890123, 'Eva Souza', '2001-01-30', 'Estudante', 'PT50000201231234567890158'),
  (678901234, 'Fernando Rocha', '1989-07-18', 'Engenheira', 'PT50000201231234567890159');

-- Inserir dados na tabela ContactoVitima
INSERT INTO ContactoVitima (telefone, nif)
VALUES
  ('925556677', 123456789),
  ('925556688', 234567890),
  ('925556699', 345678901),
  ('925556700', 456789012),
  ('925556711', 567890123),
  ('925556722', 678901234);

-- Inserir dados na tabela Torre
INSERT INTO Torre (alcance, coordenadas)
VALUES
  (100, '41.40338, 2.17403'),
  (80, '36.50368, 1.13676'),
  (150, '40.71278, -74.0060'),
  (120, '34.05223, -118.2437'),
  (200, '51.5074, -0.1278'),
  (90, '48.8566, 2.3522');
```

5.3.2 Inserção de Dados com Programa Interativo

O programa interativo permite a inserção de dados de forma dinâmica em várias tabelas da base de dados. O menu principal permite escolher a tabela e, em seguida, inserir os dados necessários. Para conseguir fazer a inserção de dados na tabela o utilizador é preciso ter permissões para conseguir aceder.

```
Digite o usuário do banco de dados: Ramirez
Digite a senha do banco de dados: RamirezSenha123!
Conexão ao banco de dados bem-sucedida!
```

O menu principal permite ao usuário selecionar a opção para inserir dados interactivamente ou sair do programa. Ao escolher a opção de inserção, o usuário é direccionado para um submenu onde pode selecionar a tabela específica.


```

Menu de Inserção de Dados
1. Inserir dados interativamente
2. Sair
Escolha (1/2): 1
Escolha a tabela para inserir dados:
1. Esquadra
2. Suspeito
3. Torre
4. Agente Policia
5. Processo Criminal
6. Vitima
Escolha (1/2/3/4/5/6): 1

```

Um exemplo de inserção de dados na tabela "Esquadra", onde o usuário fornece detalhes como o nome do coordenador, rua, localidade, código postal e nome da esquadra.

```

Nome do Coordenador: Pedro Rodrigues
Rua: Avenida Portugal
Localidade: Porto
Código Postal: 4000-020
Nome da Esquadra: Esquadra Municipal
Dados inseridos com sucesso!

```

Na inserção de dados na tabela "Processo Criminal", já é um pouco diferente pois é preciso que já tenha outros dados, uma vez que possui chaves estrangeiras, onde o usuário fornece a descrição do incidente, data do incidente, estado, data de início, data de fim, número policial (Chave Estrangeira), ID do suspeito (Chave Estrangeira) e NIF da vítima (Chave Estrangeira).

```

Escolha (1/2/3/4/5/6): 5
Descrição do Incidente: Ladrao roubou 5 milhões da conta da vitima após dizer que a chamada era do Banco.
Data do Incidente (YYYY-MM-DD): 2024-02-24
Estado (0 ou 1): 1
Data de Inicio (YYYY-MM-DD): 2024-03-01
Data de Fim (YYYY-MM-DD) ou deixe vazio: 2024-04-25
Número Policial: 1
ID do Suspeito: 1
NIF da Vítima: 250469804
Dados inseridos com sucesso!

```

Concluindo, o programa foi projetado para facilitar a inserção de dados nas tabelas do banco de dados de forma interativa e eficiente. As capturas de tela mostram a interface do programa e os passos seguidos para inserir dados em diferentes tabelas. O código completo do programa está disponível em um arquivo anexo para consulta detalhada.

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

Para calcular a dimensão inicial da base de dados e a taxa de crescimento anual com base no modelo físico que implementamos em SQL, precisamos considerar o tipo de dados de cada atributo e o espaço que cada um ocupa.

Tendo em conta os conhecimentos prévios relativamente ao tamanho ocupado por datatypes como INT, VARCHAR, ENUM, entre outros, calculamos o espaço ocupado em disco por cada atributo pertencente a cada tabela, de forma a gerar uma estimativa da ocupação total em disco da nossa base de dados.

Date type	Storage Required
TINYINT	1 byte
VARCHAR	L + 1 bytes if column values require 0 – 255 bytes, L + 2 bytes if values may require more than 255 bytes
ENUM	1 or 2 bytes, depending on the number of enumeration values (65,535 values maximum)
DATE	3 bytes
INT	4 bytes
TIMESTAMP	4 bytes

Tabela 8 - Espaço que cada Datatype ocupa

Esquadra

Atributo	Tipo	Tamanho (Bytes)
Id	INT	4
Nome_esquadra	VARCHAR(75)	76
Rua	VARCHAR(75)	76
Localidade	VARCHAR(75)	76
cod_postal	VARCHAR(9)	10
Nome_coordenador	VARCHAR(75)	76
Total		318

Tabela 9 - Espaço (Bytes) tabela Esquadra

Chamada

Atributo	Tipo	Tamanho (Bytes)
Id	INT	4
Numero Destino	VARCHAR(20)	21
Numero Origem	VARCHAR(20)	21
Duração	VARCHAR(15)	16
Data_Hora	TIMESTAMP	4
idSuspeito	INT	4
nif	INT	4
IdTorre	INT	4
	Total	78

Tabela 10 - Espaço (Bytes) tabela Chamada

Torre

Atributo	Tipo	Tamanho (Bytes)
Id	INT	4
Coordenadas	VARCHAR(100)	101
Alcance	INT	4
	Total	109

Tabela 11 - Espaço (Bytes) tabela Torre

Agente da Policia

Atributo	Tipo	Tamanho (Bytes)
Nº policial	INT	4
Nome	VARCHAR(75)	76
Cargo	ENUM	1
Categoria	ENUM	1
idEsquadra	INT	4
	Total	86

Tabela 12 - Espaço (Bytes) tabela Agente_Polícia

ContactoAgente

Atributo	Tipo	Tamanho (Bytes)
telefone	VARCHAR(30)	31
Agente Policia	INT	4
	Total	35

Tabela 13 - Espaço (Bytes) tabela ContactoAgente

Contacto Esquadra

Atributo	Tipo	Tamanho (Bytes)
telefone	VARCHAR(30)	31
Esquadra	INT	4
	Total	35

Tabela 14 - Espaço (Bytes) tabela ContactoEsquadra

Contacto Vitima

Atributo	Tipo	Tamanho (Bytes)
telefone	VARCHAR(30)	31
Vitima	INT	4
	Total	35

Tabela 15 - Espaço (Bytes) tabela ContactoVítima

Processo Criminal

Atributo	Tipo	Tamanho (Bytes)
Id	INT	4
Descrição do incidente	VARCHAR(500)	502
Data do incidente	DATE	3
Contacto suspeito	VARCHAR(30)	31
Status	BOOLEAN	1
Data inicio	DATE	3
Data fim	DATE	3
num_policial	INT	4
idSuspeito	INT	4
Nif	INT	4
Total		559

Tabela 16 - Espaço (Bytes) tabela Processo Criminal

Vítima

Atributo	Tipo	Tamanho (Bytes)
NIF	INT	4
Nome	VARCHAR(45)	46
Idade	INT	4
Data Nascimento	DATE	3
Conta Bancária	VARCHAR(35)	36
Emprego	VARCHAR(45)	46
Total		139

Tabela 17 - Espaço (Bytes) tabela Vítima

Suspeito

Atributo	Tipo	Tamanho (Bytes)
Id	INT	4
Nome	VARCHAR(45)	46
Contacto	VARCHAR(20)	21
Localização	VARCHAR(100)	101
Descrição	VARCHAR(500)	502
Total		674

Tabela 18 - Espaço (Bytes) tabela Suspeito

Tabela	Espaço (Bytes)	Nº Entradas	Total (Bytes)
Processo Criminal	318	11	3498
Agente policia	86	8	688
Esquadra	318	3	954
Suspeito	674	6	4044
Vitima	139	6	834
Torre	109	6	654
Chamada	172	11	1892
ContactoAgente	35	8	280
ContactoEsquadra	35	3	105
ContactoVitima	35	6	210
Total	1921		13159

Tabela 19 - Espaço da Base de Dados com e sem povoamento

Desta forma, o tamanho total da nossa base de dados seria, sem povoamento, seria de **1921 Bytes**. Com povoamento, através da multiplicação de todas as entradas de cada tabela, seria de **13159 Bytes**.

Agora, para estimar o crescimento anual, precisamos assumir quantos registros novos serão adicionados a cada tabela por ano.

Suponhamos as seguintes taxas de crescimento anual para cada tabela (número de novos registros por ano):

- ☐ Esquadra: 20 novos registros
- ☐ Suspeito: 500 novos registros
- ☐ Vítima: 300 novos registros
- ☐ Torre: 50 novos registros
- ☐ Agente_Polícia: 80 novos registros
- ☐ ContactoAgente: 20 novos registros
- ☐ ContactoEsquadra: 20 novos registros
- ☐ ContactoVítima: 300 novos registros
- ☐ Processo_Criminal: 700 novos registros
- ☐ Chamada: 1000 novos registros

Tabela	Espaço inicial (Bytes)	Novos Registros/ano	Tamanho Bytes/ano
Processo Criminal	559	700	391300
Agente polícia	86	80	6880
Esquadra	318	20	6360
Suspeito	674	500	337000
Vitima	139	300	41700
Torre	109	50	5450
Chamada	78	1000	78000
ContactoAgente	35	20	700
ContactoEsquadra	35	20	700
ContactoVítima	35	300	10500
			878590

Tabela 20 - Estimativa do tamanho no final do 1º ano

Através da análise da tabela de calculo, conseguimos observar que a estimativa do crescimento anual, tendo em conta o povoamento do nosso modelo faria um total de **878590 Bytes**. Este crescimento daria uma taxa de crescimento de **8476%**, dado pela seguinte fórmula:

$$\left(\frac{878590-13159}{13159} \times 100\right).$$

5.5 Definição e caracterização de vistas de utilização em SQL

Uma view é uma tabela virtual que não existe fisicamente. É uma representação de dados que são derivados de uma ou mais tabelas existentes. É criada por uma consulta que reúne esses dados num formato que faz sentido para o usuário de forma muito mais simplificada e acessível.

```
CREATE VIEW ProcessosPorVitima AS
SELECT pc.idProcesso,
pc.descr_incidente,
pc.dat_incidente,
pc.estado, pc.dat_inicio,
pc.dat_fim,
v.nome AS nome_vitima
FROM Processo_Criminal pc
INNER JOIN Vitima v ON pc.nif = v.nif;
```

A view ProcessosPorVitima irá ajudar na consolidação de todos os processos criminais associados a uma vítima específica. Consideramos a implementação desta bastante importante, pois fornece uma ferramenta valiosa para os agentes. Com esta view, eles podem facilmente identificar e analisar os diversos processos criminais aos quais uma vítima está ligada o que poderá ajudar a identificar padrões e tendências, o que pode ser crucial para a resolução de casos e a prevenção de futuros incidentes.

```
CREATE VIEW ChamadasPorTorre AS
SELECT t.idTorre, c.num_origem AS coordenadas_torre, COUNT(*) AS total_chamadas
FROM Chamada c
JOIN Torre t ON c.idTorre = t.idTorre
GROUP BY t.idTorre, c.num_origem;
```

A view ChamadasPorTorre pode ser uma ferramenta crucial na localização das chamadas suspeitas, identificando a antena de onde a chamada foi feita. A análise de padrões nas chamadas pode revelar atividades suspeitas, como por exemplo um volume incomum de chamadas de uma torre específica que podem levar a que sejam tomadas algumas medidas para monitorar mais de perto as chamadas e prevenir futuros ataques.


```
CREATE VIEW VitimasPorIdade AS
SELECT CASE
    WHEN idade < 18 THEN 'Menor de 18'
    WHEN idade BETWEEN 18 AND 30 THEN '18-30'
    WHEN idade BETWEEN 31 AND 50 THEN '31-50'
    ELSE 'Mais de 50'
END AS faixa_etaria,
COUNT(*) AS total_vitimas
FROM Vitima
GROUP BY faixa_etaria;
```

A view VitimasPorIdade poderá ser útil principalmente para fornecer uma visão geral da distribuição etária das vítimas. Isso poderá ajudar as autoridades a entender melhor a demografia das vítimas, podendo chegar padrões que possam prevenir ocorrências futuras.

5.6 Tradução das interrogações do utilizador para SQL

As queries SQL apresentadas a seguir foram desenvolvidas para responder a perguntas específicas relacionadas às operações diárias da aplicação, como a listagem de chamadas associadas a torres, identificação de polícias por esquadra, monitoramento de processos criminais em andamento e concluídos, e a contabilização de incidentes geridos por cada agente policial. Estas consultas são projetadas para facilitar a extração de informações relevantes e fornecer uma visão clara sobre como está a ocorrer os processos criminais e todos os detalhes dos mesmos.

- 1. Listar todas as chamadas associadas a uma determinada torre.

```
SELECT T.idTorre,
       C.idChamada,
       C.num_destino,
       C.num_origem,
       C.data_hora AS data_hora_chamada
FROM Chamada C
INNER JOIN Torre T ON C.idTorre = T.idTorre
WHERE T.idTorre = 1;
```

	idTorre	idChamada	num_destino	num_origem	data_hora_chamada
▶	1	1	925556699	921348654	2023-09-05 13:10:44
	1	6	925556722	923456789	2023-12-31 16:25:15
	1	11	925556677	925126642	2024-01-10 10:54:22

Figura 15 - Resultados da Query 1

Esta query seleciona o ID da torre, o ID da chamada, o número de origem e o número de destino da chamada e ainda a data e hora da chamada, alterando o nome da coluna para 'data_hora_chamada'. É indicada a tabela Chamada como principal referenciada por 'C'. É feita a junção interna com a tabela Torre referenciada como T, utilizando a condição ON C.idTorre = T.idTorre, que estabelece a correspondência entre o idTorre na tabela Chamada e na tabela Torre. Filtra os resultados para incluir apenas as chamadas associadas à torre com o ID escolhido (neste caso, idTorre=1) pelo WHERE.

2. Listar os nomes dos policias que pertencem a uma determinada esquadra.

```
SELECT Esquadra.idEsquadra, Agente_Policia.num_policial, Agente_Policia.nome
FROM Agente_Policia
INNER JOIN Esquadra
WHERE Esquadra.idEsquadra=1
ORDER BY Agente_Policia.nome ASC;
```

	idEsquadra	num_policial	nome
►	1	2	Carlos Silva
	1	6	Catarina Mendes
	1	1	Claudio
	1	8	Fernando Mendes
	1	7	Gonçalo Maia
	1	3	Jorge Mendes
	1	5	Maria Brito
	1	4	Sergio Calado

Figura 16 - Resultados da Query 2

Esta query seleciona o ID da esquadra, o número do agente e o nome dos agentes que pertencem a uma determinada esquadra. Indica a tabela Agente_Policia como principal, referenciada por Agente_Policia. É feita a junção interna com a tabela Esquadra, utilizando a condição ON Agente_Policia.idEsquadra = Esquadra.idEsquadra, que estabelece a correspondência entre o idEsquadra na tabela Agente_Policia e na tabela Esquadra. Filtra os resultados para incluir apenas os policiais que pertencem à esquadra com o ID escolhido (neste caso, idEsquadra = 1) pelo WHERE. Por fim, ordena os resultados em ordem alfabética (de forma ascendente) pelo nome do agente em ORDER BY Agente_Policia.nome ASC.

3. Lista todos os processos criminais que ainda estão a decorrer detalhando o caso, o agente responsável, a vítima e suspeito do caso.

```
SELECT Processo_Criminal.idProcesso,
Processo_Criminal.descr_incidente,
Agente_Policia.num_policial ,
Agente_Policia.nome AS nome_policia_responsavel,
Vitima.nif, Vitima.nome AS nome_vitima ,
Suspeito.idSuspeito,
Suspeito.nome AS nome_suspeito,
Processo_Criminal.estado
AS estado_processo
FROM Processo_Criminal
INNER JOIN Agente_Policia ON Processo_Criminal.num_policial=Agente_Policia.num_policial
INNER JOIN Suspeito ON Processo_Criminal.idSuspeito=Suspeito.idSuspeito
INNER JOIN Vitima ON Processo_Criminal.nif=Vitima.nif
WHERE Processo_Criminal.estado=0;
```

	idProcesso	descr_incidente	num_policial	nome_policia_responsavel	nif	nome_vitima	idSuspeito	nome_suspeito	estado_processo
▶	4	Roubo de identidade e informações bancárias a...	2	Carlos Silva	234567890	Bruno Costa	2	NULL	0
	9	Fraude bancária por telefonema, vítima engana...	3	Jorge Mendes	345678901	Carla Ferreira	6	NULL	0
	10	Fraude financeira por telefonema, vítima foi con...	6	Catarina Mendes	678901234	Fernando Rocha	6	NULL	0
	11	Esquema de phishing via telefonema, acesso a ...	1	Claudio	123456789	Ana Silva	6	NULL	0

Figura 17 - Resultados da Query 3

A query seleciona as seguintes colunas idProcesso da tabela Processo_Criminal, descr_incidente também da tabela Processo_Criminal, num_policial e nome da tabela Agente_Policia (o nome do agente foi renomeado para nome_policia_responsável para mais fácil identificação), nif e nome da tabela Vítima (nome da tabela Vítima foi renomeado para nome_vitima), idSuspeito e nome da tabela Suspeito (nome da tabela Suspeito foi renomeado para nome_suspeito) e ainda o estado da tabela Processo_Criminal (renomeado para estado_processo). Indica a tabela Processo_Criminal como principal. É indicada a tabela Processo_Criminal como principal.

As junções internas com a tabela Agente_Policia utilizando a condição ON Processo_Criminal.num_policial = Agente_Policia.num_policial é útil para verificar o agente do processo criminal responsável. A junção com a tabela Suspeito utilizando a condição ON Processo_Criminal.idSuspeito = Suspeito.idSuspeito é útil para identificar o suspeito do processo. A junção com a tabela Vítima utilizando a condição ON Processo_Criminal.nif = Vitima.nif é útil para verificar qual é a vítima que sofreu o golpe.

Por fim, é utilizado o WHERE para filtrar apenas os processos criminais cujo estado seja equivalente a 0, isto é, processos criminais que ainda estão abertos.

4. Lista o número de processos criminais geridos por cada agente no ano de 2023.

```
SELECT Agente_Policia.nome , COUNT(Processo_Criminal.idProcesso) AS numero_incidentes
FROM Agente_Policia
INNER JOIN Processo_Criminal ON Agente_Policia.num_policial =Processo_Criminal.num_policial
WHERE Processo_Criminal.dat_inicio BETWEEN '2023-01-01' AND '2023-12-31'
GROUP BY Agente_Policia.nome
ORDER BY numero_incidentes DESC;
```

	nome	numero_incidentes
►	Catarina Mendes	2
	Jorge Mendes	1
	Claudio	1
	Maria Brito	1
	Carlos Silva	1
	Sergio Calado	1

Figura 18 - Resultados da Query 4

A query seleciona as seguintes colunas: nome da tabela Agente_Policia e uma contagem (COUNT) do número de processos criminais (idProcesso), renomeada como numero_incidentes. Indica a tabela Agente_Policia como principal.

Realiza uma junção interna (INNER JOIN) com a tabela Processo_Criminal utilizando a condição ON Agente_Policia.num_policial = Processo_Criminal.num_policial. Após isso, filtra os resultados para incluir apenas os processos iniciados no ano de 2023, usando a condição WHERE Processo_Criminal.dat_inicio BETWEEN '2023-01-01' AND '2023-12-31'. É Agrupado os resultados pelo nome do agente (GROUP BY Agente_Policia.nome). Por fim, é feita a ordenação dos resultados pela contagem de incidentes em ordem decrescente (ORDER BY numero_incidentes DESC) por razões de organização, facilidade na procura e estética.

5. Lista os policiais responsáveis pela resolução de casos e as esquadras que estes representam, ou seja, as esquadras que são responsáveis por casos que já foram concluídos e lista ainda o numero de incidentes por cada agente.

```
SELECT DISTINCT Agente_Policia.num_policial, Agente_Policia.nome AS nome_policia,
Esquadra.idEsquadra, Esquadra.nome_esquadra,
COUNT(Processo_Criminal.idProcesso) AS numero_incidentes,
Processo_Criminal.estado AS estado_processo
FROM Agente_Policia
INNER JOIN Esquadra ON Agente_Policia.idEsquadra=Esquadra.idEsquadra
INNER JOIN Processo_Criminal ON Agente_Policia.num_policial=Processo_Criminal.num_policial
WHERE Processo_Criminal.estado=1

GROUP BY
    Agente_Policia.num_policial,
    Agente_Policia.nome,
    Esquadra.idEsquadra,
    Esquadra.nome_esquadra

ORDER BY Agente_Policia.nome ASC;
```

	num_policial	nome_policia	idEsquadra	nome_esquadra	numero_incidentes	estado_processo
►	2	Carlos Silva	1	Polícia de Aljô	1	1
	6	Catarina Mendes	3	Polícia de Braga	1	1
	1	Claudio	1	Polícia de Aljô	1	1
	3	Jorge Mendes	2	Polícia de Paços de Ferreira	1	1
	5	Maria Brito	3	Polícia de Braga	1	1
	4	Sergio Calado	2	Polícia de Paços de Ferreira	2	1

Figura 19 - Resultados da Query 5

A query seleciona as seguintes colunas: num_policial da tabela Agente_Policia, nome da tabela Agente_Policia (renomeado para nome_policia), idEsquadra e nome_esquadra da tabela Esquadra, a contagem (COUNT) do número de processos criminais (idProcesso), renomeada para numero_incidentes que contará o número de idProcessos (que terá o efeito de contar o número de processos) e o estado da tabela Processo_Criminal (renomeado para estado_processo). É indicada a tabela Agente_Policia como principal.

São feitas junções internas (INNER JOIN) com as tabelas Esquadra e Processo_Criminal. A junção com a tabela Esquadra utilizando a condição ON Agente_Policia.idEsquadra = Esquadra.idEsquadra é feita para representar as esquadras que os agentes associados representam e a junção com a tabela Processo_Criminal utilizando a condição ON Agente_Policia.num_policial = Processo_Criminal.num_policial é usada para listar os processos criminais associados a cada agente. Filtra os resultados para incluir apenas os processos que estão resolvidos (com estado = 1), usando a condição WHERE Processo_Criminal.estado = 1.

É feita a agrupação dos resultados pelas colunas num_policial, nome, idEsquadra e nome_esquadra. Por fim, é feita a ordenação dos resultados pelo nome do policial em ordem ascendente (ORDER BY Agente_Policia.nome ASC).

5.7 Indexação do Sistema de Dados

Os Índices são estruturas de dados criadas com o objetivo de melhorar a velocidade das consultas de dados mais rapidamente. São usados para facilitar a busca de informações em uma tabela com o menor número possível de operações de leitura, tornando a busca mais rápida e eficiente (SELECT e WHERE, por exemplo), mas por outro lado abrandam operações de inserção e atualização (INSERT e UPDATE, por exemplo).

Tendo em conta as considerações anteriores devemos tentar apenas definir índices para as colunas que são consultadas com alguma frequência mas não são atualizadas com frequência. Uma opção de índice é a coluna coordenadas da tabela Torre, visto que, ao procurar por entradas na tabela das Torres, uma tabela que pode conter várias entradas, a primeira opção será procurar pelas coordenadas, em vez de procurar pelo idTorre, a chave primária.

```
CREATE INDEX Torre_coordenadas ON Torre(coordenadas);
```

Outros índices que poderão valer a pena implementar são nas colunas dat_incidente, idSuspeito, nif, da tabela Processo Criminal, visto que esta tabela conterá imensas entradas, e pode ser útil poder encontrar entradas nesta com base na sua data.

```
CREATE INDEX PC_dataIncidente ON Processo_Criminal(dat_incidente);  
CREATE INDEX PC_idSuspeito ON Processo_Criminal(idSuspeito);  
CREATE INDEX PC_nif ON Processo_Criminal(nif);
```

5.8 Implementação de procedimentos, funções e gatilhos

Procedures, funções e gatilhos são elementos essenciais no gerenciamento de uma base de dados. Eles permitem a automação de tarefas, garantem a integridade dos dados e facilitam a execução de operações complexas de maneira eficiente e segura. Procedures são rotinas armazenadas, que podem ser chamadas para executar uma sequência de instruções SQL. Funções são semelhantes às procedures, mas retornam um valor único e podem ser usadas em consultas SQL. Gatilhos são scripts que são automaticamente quando eventos específicos ocorrem em uma tabela, como inserções, atualizações ou deleções.

5.8.1 Função

1. Indica se o suspeito tem ou não mais que um processo.

```
DELIMITER $$

CREATE FUNCTION fn_mais_de_um_processo (idSuspeito INT)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
    DECLARE num_processos INT;
    SELECT COUNT(*) INTO num_processos
    FROM Processo_Criminal
    WHERE idSuspeito = idSuspeito;
    RETURN IF(num_processos > 1, 1, 0);
END $$

DELIMITER ;

-- executar
SELECT fn_mais_de_um_processo(1);
```

Esta função retorna um valor booleano (1 para verdadeiro e 0 para falso), indicando se um suspeito está envolvido em mais de um processo criminal. Isso é útil para análises rápidas sobre o envolvimento de suspeitos em múltiplos casos.

5.8.2 Procedures

1. Lista os suspeitos de um dado agente passando o id do agente como argumento.

```
DELIMITER $$

CREATE PROCEDURE sp_suspeitos_por_agente (IN idAgente INT)
BEGIN
    SELECT
        S.idSuspeito,
        S.nome,
        S.contacto,
        S.localizacao,
        S.descricao
    FROM Suspeito S
    INNER JOIN Processo_Criminal P ON S.idSuspeito = P.idSuspeito
    WHERE P.num_policial = idAgente;
END $$

DELIMITER ;
```

Ao executar este procedimento, são retornados os detalhes dos suspeitos, incluindo o ID, nome, contato, localização e descrição. A função utiliza uma junção interna (INNER JOIN) entre as tabelas Suspeito e Processo_Criminal com base no idSuspeito, filtrando os resultados para

incluir apenas aqueles associados ao agente especificado pelo argumento idAgente. Para utilizar este procedimento, basta chamar a função com o ID do agente desejado, como no exemplo

```
CALL sp_suspeitos_por_agente(1);
```

 que listará todos os suspeitos associados ao agente com o número de policial 1.

2. Lista as chamadas feitas por um dado suspeito dando o seu id

```
DELIMITER $$
CREATE PROCEDURE sp_historico_chamadas_suspeito (IN p_idSuspeito INT)
BEGIN
    SELECT
        C.idChamada,
        C.num_origem,
        C.num_destino,
        C.data_hora,
        C.duracao,
        T.coordenadas AS torre_coordenadas,
        T.alcance AS torre_alcance
    FROM Chamada C
    JOIN Torre T ON C.idTorre = T.idTorre
    WHERE C.idSuspeito = p_idSuspeito;
END $$

DELIMITER ;

-- executar
CALL sp_historico_chamadas_suspeito(1);
```

O procedimento armazenado `sp_historico_chamadas_suspeito` foi criado para listar o histórico de chamadas feitas por um determinado suspeito, identificado pelo seu ID (`p_idSuspeito`). Ao executar este procedimento, são retornados detalhes das chamadas, incluindo o ID da chamada, número de origem, número de destino, data e hora da chamada, duração, coordenadas da torre e alcance da torre. A função utiliza uma junção interna (JOIN) entre as tabelas Chamada e Torre com base no `idTorre`, filtrando os resultados para incluir apenas aquelas chamadas associadas ao suspeito especificado pelo argumento `p_idSuspeito`. Para utilizar este procedimento, basta chamar a função com o ID do suspeito desejado, como no exemplo `CALL sp_historico_chamadas_suspeito(1);` que listará todas as chamadas associadas ao suspeito com o ID 1.

5.8.3 Triggers

1. Calcular a idade automaticamente ao inserir ou atualizar a tabela Vítima.

```
DELIMITER //
```

```
CREATE TRIGGER update_idade_before_insert
BEFORE INSERT ON Vítima
FOR EACH ROW
BEGIN
    SET NEW.idade = TIMESTAMPDIF(YEAR, NEW.dat_nasc, CURDATE());
END //
```

```
CREATE TRIGGER update_idade_before_update
BEFORE UPDATE ON Vítima
FOR EACH ROW
BEGIN
    SET NEW.idade = TIMESTAMPDIF(YEAR, NEW.dat_nasc, CURDATE());
END //
```

```
DELIMITER ;
```

Os triggers `update_idade_before_insert` e `update_idade_before_update` garantem que a idade da vítima, na tabela `Vítima`, seja calculada automaticamente com base na data de nascimento fornecida. O trigger `update_idade_before_insert` é executado antes de um novo registro ser inserido, calculando a idade com a função `TIMESTAMPDIF(YEAR, NEW.dat_nasc, CURDATE())`. De forma semelhante, o trigger `update_idade_before_update` é executado antes da atualização de um registro existente, recalculando a idade sempre que a data de nascimento for modificada. Ambos os triggers asseguram a precisão da idade na tabela `Vítima`.

2. Impedir Atualização de Status de Processo para Finalizado se Não Houver Data de Fim

```
DELIMITER //
```

```
CREATE TRIGGER impedir_finalizacao_sem_data_fim
BEFORE UPDATE ON Processo_Criminal
FOR EACH ROW
BEGIN
    IF NEW.estado = 1 AND NEW.dat_fim IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é possível finalizar um processo sem definir a data de fim.';
    END IF;
END;
//
DELIMITER ;
```

O trigger `impedir_finalizacao_sem_data_fim` impede que o status de um processo criminal seja atualizado para "finalizado" sem que uma data de fim seja definida. Este é acionado antes da atualização na tabela `Processo_Criminal` e verifica se o novo estado é 1 e se a data de fim é nula. Se estas condições forem verdadeiras, ele gera uma exceção e impede a atualização, mantendo a integridade dos dados ao garantir que todos os processos finalizados tenham uma data de conclusão registrada.

6 Conclusões e trabalho futuro

Ao concluirmos este trabalho, podemos afirmar que cada processo foi conduzido com dedicação e empenho. Desde a contextualização do sistema até à modelação Física. Cada etapa foi planejada e executada para garantir a qualidade e eficácia do sistema de bases de dados proposto. Ao definir o contexto de aplicação, compreendemos a importância de entender profundamente as necessidades do caso de estudo, o que nos permitiu estabelecer objetivos claros e relevantes. Durante o levantamento e análise de requisitos, adotamos métodos para garantir que todas as exigências fossem capturadas e organizadas de forma coerente. A modelação concetual e lógica refletiu a complexidade do sistema, proporcionando uma base sólida para o desenvolvimento do modelo físico que realizamos em linguagem SQL, que foi essencial para definir a estrutura exata do banco de dados, assegurando que cada tabela, chave primária, chave estrangeira, índices, normalização e até interrogações dos utilizadores, fossem implementados de acordo com os requisitos do sistema.

Como grupo, reconhecemos a importância fundamental da fase inicial do projeto no desenvolvimento global do projeto. Está claro para nós que os alicerces de uma base de dados robusta são estabelecidos numa primeira fase. A tarefa de criar uma narrativa e ‘animar’ uma futura base de dados foi particularmente interessante para nós. No entanto, enfrentamos desafios como por exemplo no levantamento de requisitos, pois, assim como acontece em reuniões reais, nem tudo é decidido de uma vez, é necessário um processo iterativo de avanços e recuos até alcançarmos um consenso satisfatório.

Relativo à parte mais focado na construção do modelo físico, sentimos maior dificuldade na construção e caracterização de utilizadores, e também a implementação da indexação no sistema de dados, pois não estávamos familiarizados com o conceito e quais as razões para implementar os mesmos.

Relativamente a um trabalho futuro, achamos interessante implementar a criação de uma plataforma web para o sistema WaveDetectives não só tornaria o sistema mais acessível e fácil de usar, mas também ampliaria as suas capacidades de análise e manipulação de dados. Com uma interface intuitiva e funcionalidades avançadas, os usuários poderão explorar e gerenciar os dados de forma mais eficiente, contribuindo para a tomada de decisões informadas e a eficácia geral das operações.

De maneira geral, estamos satisfeitos com o nosso projeto e confiantes de que a nossa Base de Dados, estará apta e será competente, para resolver todos os casos que os detetives enfrentarão, com todas as exigências propostas, e todos os objetivos que inicialmente definimos.

7. Bibliografia

Oracle. (2024). Data Type Storage Requirements Reference Manual.
dev.mysql.com/doc/refman/8.0/en/storage-requirements.html

Belo, O. (2022). A Merceria da D. Acácia.