



Universidade do Minho  
Licenciatura em Engenharia Informática

## Unidade Curricular de Bases de Dados

Ano Letivo de 2023/2024

**P.I.S.T.A.**

**Fernando Pires, Miguel Gramoso, Pedro Teixeira,  
Rafael Seara, Sara Silva**

Abril, 2024

**BD**

Data de Receção	
Responsável	
Avaliação	
Observações	

**P.I.S.T.A.**

**Fernando Pires, Miguel Gramoso, Pedro Teixeira,  
Rafael Seara, Sara Silva**

Abril, 2024

## Resumo

Este relatório foi desenvolvido no âmbito do desenvolvimento de uma Base de Dados para a empresa P.I.S.T.A., que foca a sua atividade na área de investigação de casos. Este software tem como objetivo facilitar e tornar mais eficiente a grande maioria do processo inerente à investigação de um caso. Desde a criação do próprio caso até ao controlo de detetives, suspeitos, provas e relatórios passando, por exemplo, pelo detetive responsável pelo caso e pela lista de provas do mesmo que incriminam suspeitos.

Ao longo deste relatório são apresentadas na íntegra todas as etapas do desenvolvimento desta base de dados sendo que a implementação física da mesma será abordada na segunda parte do projeto.

Em primeiro lugar, foram estudadas todas características do meio onde este software será inserido. Foram analisadas as condições em que a empresa surgiu, o seu método de trabalho e quais as suas motivações e objetivos para a implementação de uma base de dados, verificando se esta seria ou não economicamente viável.

Em seguida, através de métodos de análise e em conjunto com o cliente, foram levantados e aprovados os requisitos para o sistema que guiaram todo o processo de desenvolvimento. No fim desta fase, foi iniciada a modulação conceptual, desenvolvida com recurso à ferramenta “brModelo”, e que foi posteriormente aprovada pela P.I.S.T.A.. A partir desta desenvolveu-se em conformidade com as regras do mapeamento ER, o modelo lógico, na ferramenta “MySQL Workbench” também ele validado pelo cliente. Por fim, a Base de Dados, gerada também ela através do “MySQL Workbench”, foi implementada fisicamente, garantindo que todo o trabalho anterior convergia para uma solução correta e segura.

Concluído este processo, após a implementação da base de dados no cliente, será dado como terminado este projeto.

**Área de Aplicação:** Desenho, Arquitetura, Desenvolvimento e Implementação de Sistemas de Bases de Dados.

**Palavras-Chave:** Bases de Dados, Bases de Dados Relacionais, Análise de Requisitos, Entidades, Atributos, Relacionamentos, Modelo Conceptual, Modelo Lógico, Normalização, Interrogações.

# Índice

<b>RESUMO .....</b>	<b>I</b>
<b>ÍNDICE .....</b>	<b>II</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>IV</b>
<b>ÍNDICE DE TABELAS .....</b>	<b>V</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. CONTEXTUALIZAÇÃO .....	1
1.2. MOTIVAÇÃO E OBJETIVOS .....	2
1.3. ANÁLISE DA VIABILIDADE DO PROCESSO .....	2
1.4. RECURSOS E EQUIPA DE TRABALHO .....	3
1.5. PLANO DE EXECUÇÃO DO PROJETO .....	4
<b>2. LEVANTAMENTO E ANÁLISE DE REQUISITOS .....</b>	<b>5</b>
2.1. MÉTODO DE LEVANTAMENTO E DE ANÁLISE DE REQUISITOS ADOTADOS .....	5
2.2. REQUISITOS LEVANTADOS .....	5
2.2.1 <i>Requisitos de Descrição</i> .....	5
2.2.2 <i>Requisitos de Exploração</i> .....	6
2.2.3 <i>Requisitos de Controlo</i> .....	7
2.3. ANÁLISE DE REQUISITOS .....	7
<b>3. MODELAÇÃO CONCEPTUAL .....</b>	<b>9</b>
3.1. APRESENTAÇÃO DA ABORDAGEM DE MODELAÇÃO REALIZADA .....	9
3.2. IDENTIFICAÇÃO E CARACTERIZAÇÃO DAS ENTIDADES .....	10
3.3. IDENTIFICAÇÃO E CARACTERIZAÇÃO DOS RELACIONAMENTOS .....	11
3.4. IDENTIFICAÇÃO E CARACTERIZAÇÃO DA ASSOCIAÇÃO DOS ATRIBUTOS COM AS ENTIDADES E RELACIONAMENTOS .....	18
3.4.1 <i>Domínio dos Atributos</i> .....	20
3.4.2 <i>Chaves Candidatas, Primárias e Alternativas</i> .....	22
3.5. APRESENTAÇÃO E EXPLICAÇÃO DO DIAGRAMA ER .....	24
3.6. VALIDAÇÃO DO MODELO DE DADOS COM O UTILIZADOR .....	25

<b>4. MODELAÇÃO LÓGICA .....</b>	<b>30</b>
4.1. CONSTRUÇÃO E VALIDAÇÃO DO MODELO DE DADOS LÓGICO .....	30
4.2. APRESENTAÇÃO E EXPLICAÇÃO DO MODELO LÓGICO PRODUZIDO .....	33
4.3. VALIDAÇÃO DO MODELO ATRAVÉS DA NORMALIZAÇÃO.....	35
4.4. VALIDAÇÃO DO MODELO COM AS INTERROGAÇÕES DO UTILIZADOR .....	36
4.5. REVISÃO DO MODELO LÓGICO COM O UTILIZADOR .....	38
<b>5. ALTERAÇÕES SOBRE A PRIMEIRA PARTE .....</b>	<b>39</b>
<b>6. IMPLEMENTAÇÃO FÍSICA.....</b>	<b>40</b>
6.1. APRESENTAÇÃO E EXPLICAÇÃO DA BASE DE DADOS IMPLEMENTADA .....	40
6.2. CRIAÇÃO DE UTILIZADORES DA BASE DE DADOS .....	44
6.3. POVOAMENTO DA BASE DE DADOS .....	45
6.4. CÁLCULO DO ESPAÇO DA BASE DE DADOS .....	45
6.5. DEFINIÇÃO E CARACTERIZAÇÃO DE VISTAS DE UTILIZAÇÃO EM SQL .....	46
6.6. TRADUÇÃO DAS INTERROGAÇÕES DO UTILIZADOR PARA SQL .....	49
6.7. INDEXAÇÃO DO SISTEMA DE DADOS .....	53
6.8. IMPLEMENTAÇÃO DE PROCEDIMENTOS, FUNÇÕES E GATILHOS .....	53
<b>7. CONCLUSÕES E TRABALHO FUTURO .....</b>	<b>56</b>
<b>8. BIBLIOGRAFIA .....</b>	<b>58</b>
<b>REFERÊNCIAS.....</b>	<b>59</b>
<b>LISTA DE SIGLAS E ACRÓNIMOS.....</b>	<b>60</b>

## Índice de Figuras

Figura 1 – Diagrama de Gantt	4
Figura 2 – Relacionamento Cliente - Caso	11
Figura 3 – Relacionamento Detetive - Caso	12
Figura 4 – Relacionamento Caso - Suspeito	13
Figura 5 – Relacionamento Caso - Prova	14
Figura 6 – Relacionamento Prova - Suspeito	14
Figura 7 – Relacionamento Detetive - Prova	15
Figura 8 – Relacionamento Detetive – Relatório	16
Figura 9 – Relacionamento Relatório - Caso	17
Figura 10 – Modelo Conceptual	24
Figura 11 – Atributos do Detetive	25
Figura 12 – Atributos do Caso	26
Figura 13 – Atributos do Cliente	27
Figura 14 – Atributos do Suspeito	28
Figura 15 – Atributos da Prova	28
Figura 16 – Atributos do Relatório	29
Figura 17 – Ilustração da tabela Entidade Prova	30
Figura 18 - Ilustração da tabela Entidade Suspeito	31
Figura 19 – Ilustração da tabela Entidade Caso	31
Figura 20 – Ilustração da tabela Entidade DetetiveTelefone	32
Figura 21 – Ilustração da tabela Entidade Detetive	32
Figura 22 – Ilustração da tabela Entidade CasoDetetive	32
Figura 23 – Ilustração da tabela Entidade ClienteTelefone	33
Figura 24 – Ilustração da tabela Entidade Cliente	33
Figura 25 – Modelo Lógico	35
Figura 26 – Ilustração do modelo RelaX da consulta dos dados dos clientes de um caso	37
Figura 27 – Ilustração do modelo RelaX da consulta do detetive com mais casos	37
Figura 28 – Ilustração do modelo RelaX da consulta do caso mais caro	38

## Índice de Tabelas

Tabela 1 – Caracterização de todos os atributos existentes.	18
Tabela 2 – Caracterização dos atributos do Caso	20
Tabela 3 – Caracterização dos atributos do Cliente	20
Tabela 4 – Caracterização dos atributos do Detetive	21
Tabela 5 – Caracterização dos atributos do Suspeito	21
Tabela 6 – Caracterização dos atributos da Prova	21
Tabela 7 – Caracterização dos atributos do Relatório	21
Tabela 8 – Caracterização dos atributos de Relacionamento Detetive - Caso	22

# 1. Introdução

## 1.1. Contextualização

Numa cidade movimentada, havia uma agência de detetives de renome conhecida como P.I.S.T.A. (Procuradores de Investigação Secreta e Táticas Astutas). Fundada em 1985 por William "Bill" Montgomery, um ex-polícia que procurava uma maneira de continuar a ajudar a comunidade de uma forma diferente após se aposentar da força policial. A sua visão era fornecer serviços de investigação confiáveis e eficientes para todos os que precisassem.

A sede da P.I.S.T.A. era um edifício de tijolos vermelhos no coração da cidade, onde uma equipa dedicada de detetives trabalhava incansavelmente para resolver os casos mais intrincados. Sob a liderança do habilidoso advogado Alex Reynolds, a agência prosperou ao longo dos anos, ganhando a reputação de resolver casos que deixavam outras agências perplexas.

No entanto, um dia, um cliente insatisfeito apresentou evidências que sugeriam que Alex Reynolds não era quem dizia ser. Ele alegava que Reynolds falsificava provas para garantir resultados favoráveis aos clientes, manchando, assim, a reputação impecável da P.I.S.T.A.

Perante essas acusações, a agência enfrentou uma crise existencial. A confiança dos clientes estava abalada, e havia a necessidade urgente de rever todos os casos já tratados por Reynolds. O problema era que a P.I.S.T.A. mantinha todos os registos em papel, armazenados num antigo armazém nos arredores da cidade.

Com milhares de casos arquivados ao longo dos anos, a tarefa de rever cada documento manualmente era monumental. Dias transformaram-se em semanas, e semanas em meses, enquanto a equipa de detetives se perdia num mar de papelada.

Percebendo que o tempo perdido na procura de evidências vitais poderia custar a reputação da agência, a equipa teve uma ideia brilhante: criar uma base de dados digital que reunisse todos os casos, juntamente com todos os recursos e informações disponíveis na P.I.S.T.A.

Com a ajuda de especialistas em tecnologia, a agência digitalizou todos os documentos e estabeleceu um sistema de administração de casos online. Isso não só permitiu uma revisão mais eficiente dos casos, como também modernizou as operações da P.I.S.T.A., tornando-a mais ágil e adaptável às necessidades do mundo contemporâneo.



Enquanto a agência se reinventava digitalmente, os detetives mergulharam na tarefa de reexaminar cada caso, assegurando que a justiça fosse verdadeiramente servida e restaurando a confiança perdida dos clientes.

## 1.2. Motivação e Objetivos

A aposta da P.I.S.T.A numa base de dados digital tem como principais objetivos os seguintes pontos:

**-Eficiência Operacional:** Um dos principais objetivos seria aumentar a eficiência operacional da agência, reduzindo o tempo e os recursos necessários para revisar casos e aceder a informações relevantes. A base de dados digital permitiria uma pesquisa rápida e precisa de documentos e dados, agilizando os processos de investigação.

**-Transparência e Integridade:** Garantir transparência e integridade em todas as operações da agência é crucial para reconstruir a confiança dos clientes. A base de dados digital será projetada para garantir que todas as informações são armazenadas de forma segura e acessível, promovendo a transparência em todos os aspetos do trabalho da P.I.S.T.A.

**-Precisão:** O objetivo é garantir a precisão das informações armazenadas na base de dados. Isso significa implementar medidas de segurança robustas para proteger os dados contra adulteração ou acesso não autorizado, bem como garantir que todos os dados sejam atualizados e precisos.

**-Acessibilidade:** Facilitar o acesso às informações é fundamental para garantir que os detetives possam trabalhar de forma eficiente e eficaz. A base de dados digital será projetada com uma interface intuitiva e fácil de usar, permitindo que os detetives encontrem rapidamente as informações necessárias para resolver casos.

**-Flexibilidade e Escalabilidade:** A base de dados digital visa ser flexível e escalável o suficiente para se adaptar às necessidades em constante mudança da agência. Isso significa projetar uma infraestrutura que possa crescer e evoluir com o tempo, garantindo que a P.I.S.T.A. permanece ágil e capaz de lidar com novos desafios e demandas futuras.

## 1.3. Análise da Viabilidade do Processo

Tendo em conta os pontos apresentados anteriormente, provam-se vários os possíveis benefícios da implementação de um sistema de base de dados na P.I.S.T.A. No entanto, esta decisão deve ser viável operacional e financeiramente. Algumas das vantagens são as seguintes:

- Acesso rápido e fácil às informações: Redução significativa no tempo gasto em pesquisas, estimada em uma diminuição de 30% no tempo médio de pesquisa.

- **Redução de erros:** A digitalização dos dados elimina o risco de erros humanos associados ao manuseio de documentos em papel. Estima-se que a transição para uma base de dados digital reduziria os erros de entrada de dados em 25%.
- **Economia de espaço e custos:** Armazenar documentos físicos em um armazém consome espaço e recursos significativos. Com uma base de dados digital, a P.I.S.T.A. poderia economizar em custos de armazenamento e espaço físico. Estima-se uma redução de 40% nos custos associados ao armazenamento físico de documentos.
- **Melhoria na colaboração e comunicação:** Uma base de dados digital permitiria uma melhor colaboração entre os detetives, facilitando o compartilhamento de informações e a comunicação entre os membros da equipe. Isso poderia resultar em uma melhoria de 35% na eficiência da comunicação interna.
- **Segurança dos dados:** A segurança dos dados é fundamental para uma agência de detetives lidando com informações confidenciais. Uma base de dados digital ofereceria recursos avançados de segurança, como criptografia e controle de acesso, reduzindo o risco de violações de dados em 50%.

Por estes motivos, a criação de uma base de dados é uma solução viável e importante para a organização da P.I.S.T.A., contribuindo para a maior eficácia no trabalho dos detetives.

## 1.4. Recursos e Equipa de Trabalho

Para garantir o sucesso do projeto, é essencial contar com uma equipa qualificada e com os recursos materiais adequados.

### Recursos Humanos

- **Equipa de Desenvolvimento:** Responsável pelo levantamento de requisitos, modelação e implementação do sistema.
- **Administradores de Banco de Dados:** Garantem a integridade, segurança e desempenho do banco de dados.
- **Analistas de Dados e Investigadores:** Interpretam os dados e identificam padrões e anomalias.
- **Suporte Técnico:** Fornecem assistência contínua aos utilizadores do sistema.
- **Clientes:** Fornecem feedback crucial para orientar o desenvolvimento do projeto.

### Recursos Materiais

- **Hardware:** Um servidor e cinco máquinas de trabalho para suportar as operações do sistema.

- **Software (SGDB):** Sistema de Gestão de Base de Dados para armazenar, organizar e recuperar os dados.

Em resumo, a combinação cuidadosa de recursos humanos e materiais, juntamente com uma integração eficaz no processo de desenvolvimento, será essencial para o sucesso do projeto, garantindo que o sistema seja robusto, eficiente e capaz de atender às necessidades dos utilizadores finais.

## 1.5. Plano de execução do projeto

Para definir a forma como o projeto de desenvolvimento do SBD iria ser realizado, a equipa da P.I.S.T.A. reuniu-se para debater, e estabeleceram um plano concreto de trabalhos, bem como o seu cronograma de execução.

Para ilustrar este plano, foi construído um Diagrama de Gantt, incluindo todas as etapas e definindo os seus períodos de execução, bem como os vários intervenientes na realização de cada tarefa.

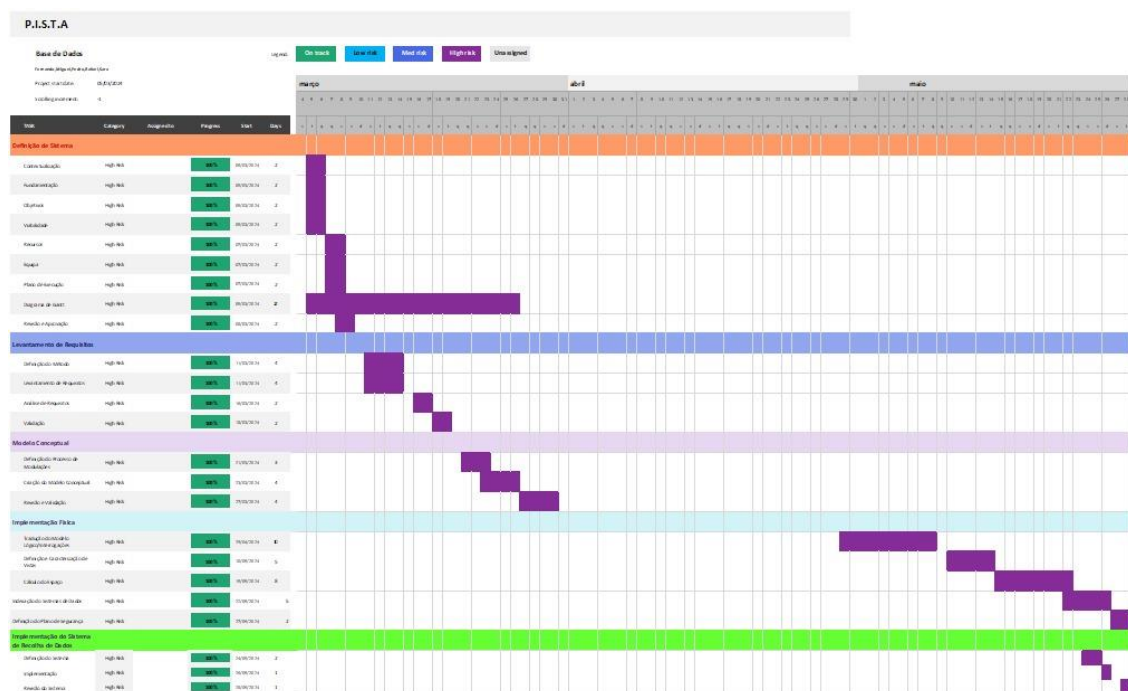


Figura 1 – Diagrama de Gantt

## 2. Levantamento e Análise de Requisitos

### 2.1. Método de Levantamento e de Análise de Requisitos Adotados

Relativamente aos métodos adotados para levantamento de requisitos, decidimos interagir e analisar o método de funcionamento do sistema atualmente implementado e a partir desse definir os requisitos necessários. Durante estas interações destacamos:

- **Entrevistas:** Foram realizadas diversas entrevistas com diferentes elementos da P.I.S.T.A, das quais foi possível analisar as necessidades para o novo sistema a implementar.
- **Observação:** Através da análise do funcionamento da empresa durante um período inicial, foi possível determinar os métodos de interação com o sistema atual e as suas limitações e capacidades.
- **Acompanhamento de casos atuais:** Acompanhamos alguns elementos da empresa durante a resolução e investigação de casos atuais. Este acompanhamento permite retirar elementos descritivos do sistema, como por exemplo os tipos de dados a armazenar e como estes se interligam.

Estes métodos permitiram o levantamento e categorização dos requisitos, que serão apresentados ao nosso empregador de maneira a garantir que o problema apresentado é o problema a ser solucionado.

### 2.2. Requisitos Levantados

#### 2.2.1 Requisitos de Descrição

1. Quando um detetive se regista, com o diretor da PISTA, este registo deve ter um nome, um contacto telefónico, um email, um estado uma classificação;

2. O detetive terá de ter necessariamente um dos seguintes estados: Ativo; Inativo;
3. A classificação de cada detetive será uma média da classificação dada pelos clientes aos casos que investiga;
4. Um ou mais detetives poderão criar e editar casos, mas apenas um é responsável pelo caso;
5. Aquando do registo de um caso deverão ser indicados um identificador único (número), uma breve descrição, um estado, uma classificação, um preço, uma data de abertura do caso e uma data de fecho do caso;
6. Todos os casos devem ser numerados sequencialmente por ordem de criação, não sendo permitido reutilizar números de casos anteriores que tenham sido fechados;
7. O caso terá de ter necessariamente um dos seguintes estados: Aberto; Fechado;
8. Cada caso deverá estar associado obrigatoriamente ao cliente que o requisitou, para que o detetive responsável possa dialogar e acertar pormenores com ele sempre que necessário;
9. Aquando do registo de um cliente deverão ser fornecidos o seu nome, um NIF, um contacto de telemóvel, um endereço, um email e um tipo;
10. O cliente terá de ter necessariamente um dos seguintes tipos: Singular; Coletivo;
11. Cada cliente poderá deixar uma classificação a cada caso que requisitou;
12. A cada caso poderão ser registados os seus suspeitos;
13. Aquando do registo de um suspeito deverão ser indicados um nome e uma breve descrição;
14. Para cada caso não pode haver mais do que um suspeito igual, ou seja, não pode haver suspeitos repetidos na lista de suspeitos do caso;
15. A cada caso poderão ser registadas as suas provas;
16. Aquando do registo de uma prova deverão ser indicados uma descrição e a data em que foi registada no sistema;
17. Para cada caso não pode haver mais do que uma prova igual, ou seja, não pode haver provas repetidas na lista de provas do caso;
18. Um caso apenas pode ser fechado quando o detetive responsável o validar;
19. Cada caso terá necessariamente de ter um relatório escrito por um detetive associado ao caso;
20. Aquando do registo de um relatório deverão ser indicados uma descrição e a data em que foi registado no sistema.

## **2.2.2 Requisitos de Exploração**

1. O detetive responsável poderá ser capaz de consultar os dados dos respetivos clientes de um dado caso por este requisitado;

2. Um detetive poderá consultar quais as provas associadas a casos que está a investigar;
3. Um detetive deverá ser capaz de consultar os suspeitos dos casos que está a investigar;
4. Um detetive deverá ser capaz de consultar os casos que está a investigar;
5. Um detetive deverá ser capaz de consultar os relatórios que preencheu dos casos que investigou;
6. Um cliente deverá ser capaz de consultar os casos que requisitou;
7. O diretor da PISTA deve poder filtrar os seus detetives por estado e por número de casos em que participou;
8. O diretor da PISTA deve poder filtrar os seus detetives, os clientes e os casos por classificação;
9. O diretor da PISTA poderá consultar, para um número indicado de clientes, os que mais dinheiro gastam em casos no geral e por tipo de caso;
10. O sistema deverá ser capaz de contabilizar o número de casos registados num determinado intervalo de tempo;
11. O sistema deverá ser capaz de contabilizar o número de casos fechados num determinado intervalo de tempo;
12. O sistema deverá ser capaz de contabilizar o número de casos a ocorrer num determinado intervalo de tempo.

### **2.2.3 Requisitos de Controlo**

1. O cliente apenas pode consultar os aspetos do sistema que a este estão direta ou indiretamente associados;
2. Classificação dos detetives apenas está visível para os diretores da agência;
3. Só o detetive responsável pode alterar o caso após este estar fechado;
4. O preço do caso apenas pode ser registado ou alterado pelo detetive responsável pelo caso;

## **2.3. Análise de Requisitos**

Após o levantamento e categorização dos requisitos do sistema, foi necessária uma análise extensa dos mesmos. Esta análise foi composta de múltiplas discussões e reuniões entre os elementos da nossa equipa, mas também com os representantes do nosso empregador, com o objetivo de impedir possíveis conflitos entre os requisitos de diferentes entidades.

Esta etapa permitiu o esclarecimento e o alinhamento das diferentes interpretações dos requisitos do sistema de armazenamento das várias partes envolvidas, facilitando trabalho e desenvolvimento futuro.

## 3. Modelação Conceptual

### 3.1. Apresentação da Abordagem de Modelação Realizada

Depois da recolha e da análise dos requisitos do sistema chegou a altura de começar a fase do planeamento do design da Base de Dados.

Com o intuito de reduzir a margem para erros e de se estabelecer um entendimento comum, ou seja, que a base de dados que será construída vá de encontro às necessidades dos seus utilizadores finais, sentimos a necessidade de utilizar um modelo, não muito técnico e fácil de ser compreendido, em que possamos comunicar com o utilizador sem ambiguidades. O modelo perfeito para esse efeito é o **Diagrama ER**.

O Diagrama ER utiliza uma abordagem top-down uma vez que começamos por identificar os dados relevantes do problema, as entidades, e os relacionamentos entre esses dados e, posteriormente, começamos a adicionar mais informação ao modelo com os atributos, quer nas entidades, quer nos relacionamentos entre as mesmas, e respetivos domínios. Por último, incluímos também a cardinalidade desses relacionamentos para que possam ser representadas algumas restrições.

O modelo conceptual tem como objetivo representar os dados presentes nos requisitos do utilizador. Ao construirmos este modelo estaremos repetidamente a fazer questões sobre entidades, relacionamentos e atributos, onde as respostas deverão estar corretamente documentadas nos requisitos.

Outro aspeto bastante positivo da utilização desta metodologia é o facto de ser completamente independente dos futuros detalhes de implementação.

Ao longo da construção deste modelo conceptual é fundamental que o mesmo seja constantemente testado e validado com o utilizador e com os seus requisitos, de modo que fique assegurado que a futura base de dados seja coerente e que efetivamente resolva um problema.



## 3.2. Identificação e Caracterização das Entidades

Atendendo aos requisitos levantados foram identificadas as entidades apresentadas em seguida.

- **Caso**

A entidade “Caso” representa a investigação que é requisitada e é representado através do seu **id**, **descrição**, **estado** (aberto, fechado), a **data de abertura** e **data de fecho** do processo, a **classificação** atribuída pelo cliente e o respetivo **preço** do serviço.

- **Detetive**

A entidade “Detetive” representa os detetives que estão contratados pela agência e fazem a investigação. É caracterizada pelo seu **id**, **nome**, **email**, **número de telemóvel**, **número de casos** que já investigou, **estado** (ativo, inativo) e a sua **classificação**, que será a média das classificações de todos os casos que investigou.

- **Cliente**

A entidade “Cliente” representa a empresa, companhia, estabelecimento ou a própria pessoa individual que pretenda requisitar uma investigação junto da agência. Cada cliente possui um **nome**, **NIF** (para possíveis efeitos de faturação), **sexo** (masculino, feminino, outro), **email**, **número de telemóvel**, **endereço** (morada) e o seu **tipo** (singular, coletivo).

- **Suspeito**

A entidade “Suspeito” representa a empresa, companhia, estabelecimento ou um indivíduo singular que esteja a ser investigado. É caracterizado por um **id**, um **nome** e uma **descrição** das suas informações que forem descobertas.

- **Prova**

A entidade “Prova” representa as evidências adquiridas pelos detetives durante a investigação de um caso. Uma prova é caracterizada por um **id**, uma **data** da descoberta e uma breve **descrição** das suas informações.

- **Relatório**

A entidade “Relatório” representa a documentação e um resumo do caso, elaborado pelo detetive. Um relatório possui um **id**, uma **data de escrita** e uma **descrição** das informações relevantes do caso.

### 3.3. Identificação e Caracterização dos Relacionamentos

- **Relacionamento Cliente – Caso**

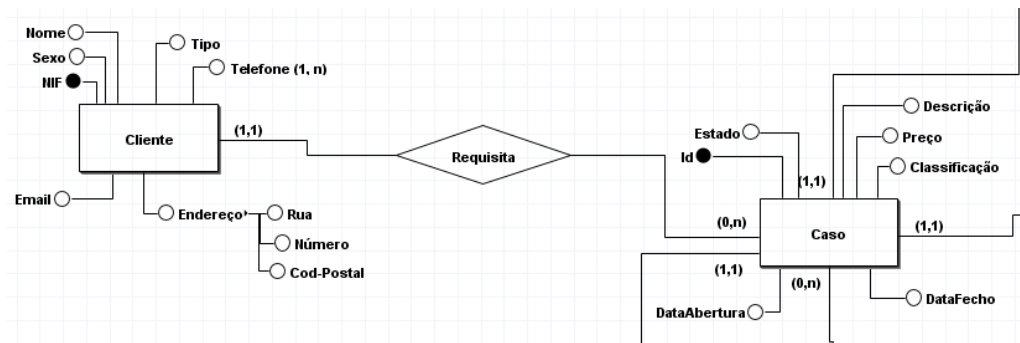


Figura 2 – Relacionamento Cliente - Caso

**Relacionamento:** Cliente Requisita Caso.

**Descrição:** A agência é abordada por clientes que têm como intenção abrir uma certa investigação. Esta informação tem de ser armazenada para podermos aceder ao pedido do cliente e para que este possa ter também todos os dados sobre os casos que lhe são associados.

**Cardinalidade:** Cliente (1,1) – Caso (1..\*).

Um cliente pode requisitar um ou vários casos (não poderá registar zero, pois, para o cliente constar na base de dados, este necessita de ter requisitado pelo menos um caso).

Um caso pertence a um e só um cliente.

**Atributos:** Este relacionamento não possui atributos.

- **Relacionamento Detetive – Caso**

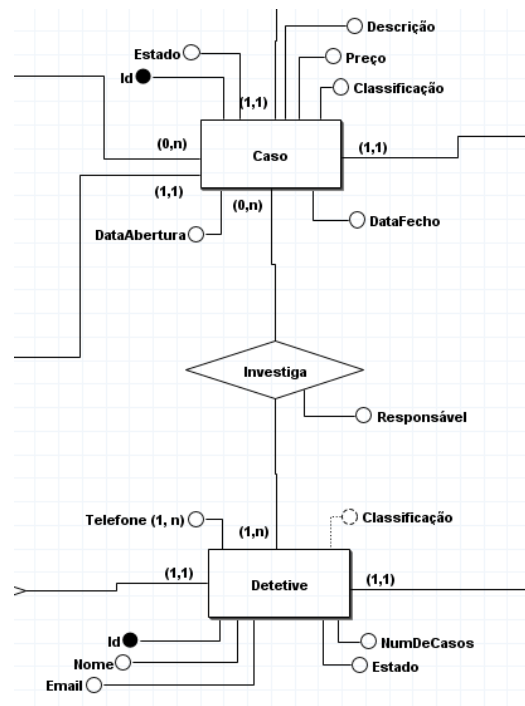


Figura 3 – Relacionamento Detetive - Caso

**Relacionamento:** Detetive Investiga Caso.

**Descrição:** Sempre que é aberto um caso é nomeado um ou mais detetives para o investigar, sendo que apenas um é o responsável sobre ele. Esta informação é guardada para que a agência possa gerenciar todos os seus casos e detetives da forma mais eficiente.

**Cardinalidade:** Detetive (1..\*) – Caso (0..\*).

Um detetive possui zero ou mais casos, pois pode não ter nenhum caso atribuído no momento, e um caso tem um ou mais detetives associados.

**Atributos:** Este relacionamento possui apenas um atributo: **responsável** que identifica qual o detetive que está responsável pelo caso.

- **Relacionamento Caso – Suspeito**

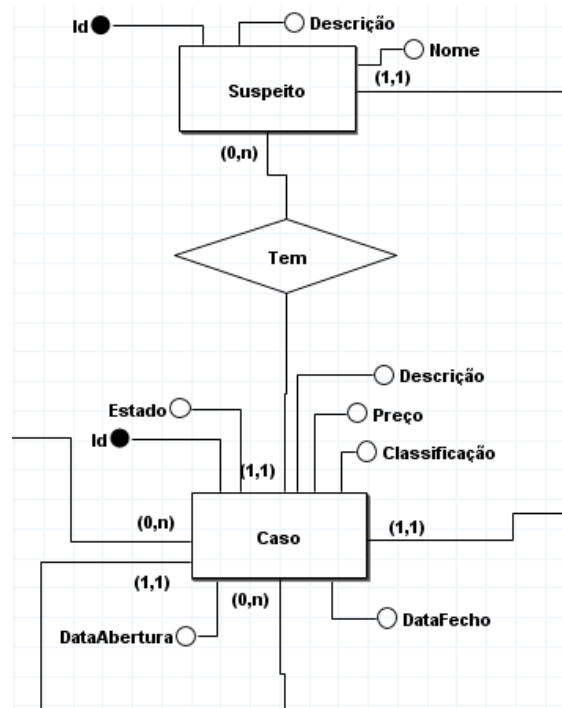


Figura 4 – Relacionamento Caso - Suspeito

**Relacionamento:** Caso Tem Suspeito.

**Descrição:** Quando um caso é aberto podem-lhe ser atribuídos zero ou mais suspeitos. Guardamos esta informação para os detetives terem acesso e procederem à devida investigação.

**Cardinalidade:** Caso (1..1) – Suspeito (0..\*).

Um suspeito para estar inserido na base de dados tem de estar associado a pelo menos um caso. Esse mesmo suspeito apenas pode estar ligado a um caso pois todos estes são diferentes e a descrição do suspeito é feita subjetivamente.

**Atributos:** Este relacionamento não possui atributos.

- **Relacionamento Caso – Prova**

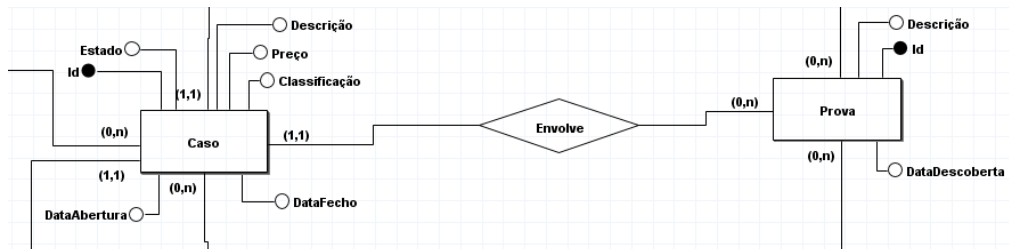


Figura 5 – Relacionamento Caso - Prova

**Relacionamento:** Caso Envolve Prova.

**Descrição:** Um caso é investigado e podem ser recolhidas provas que lhe serão associadas. Esta informação é armazenada de forma a ter a organização dos casos da agência bem gerida.

**Cardinalidade:** Caso (0..\*) – Prova (1..1).

Na eventualidade de não serem recolhidas provas durante a investigação um caso pode não ter nenhuma destas associadas. Quando uma prova é adquirida esta apenas pode ser referente a um caso.

**Atributos:** Este relacionamento não possui atributos.

- **Relacionamento Prova – Suspeito**

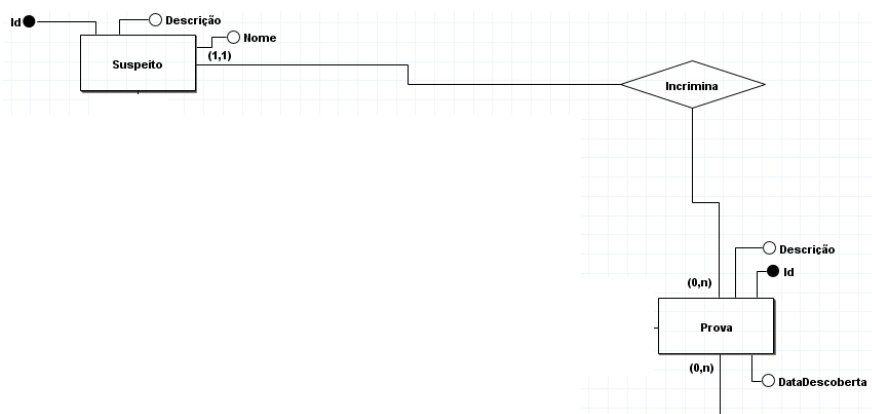


Figura 6 – Relacionamento Prova - Suspeito

**Relacionamento:** Prova Incrimina Suspeito.

**Descrição:** Quando recolhemos provas estas devem ficar armazenadas e associadas a um suspeito, para efeitos de comprovação perante o cliente que requisitou o caso.

**Cardinalidade:** Prova (0..\*) – Suspeito (1..1).

Uma prova, a existir, incrimina apenas um suspeito. No entanto, um suspeito tanto pode não ter nenhuma prova que o incrimine (podem não ter sido recolhidas) como pode ter várias.

**Atributos:** Este relacionamento não possui atributos.

- **Relacionamento Detetive – Prova**

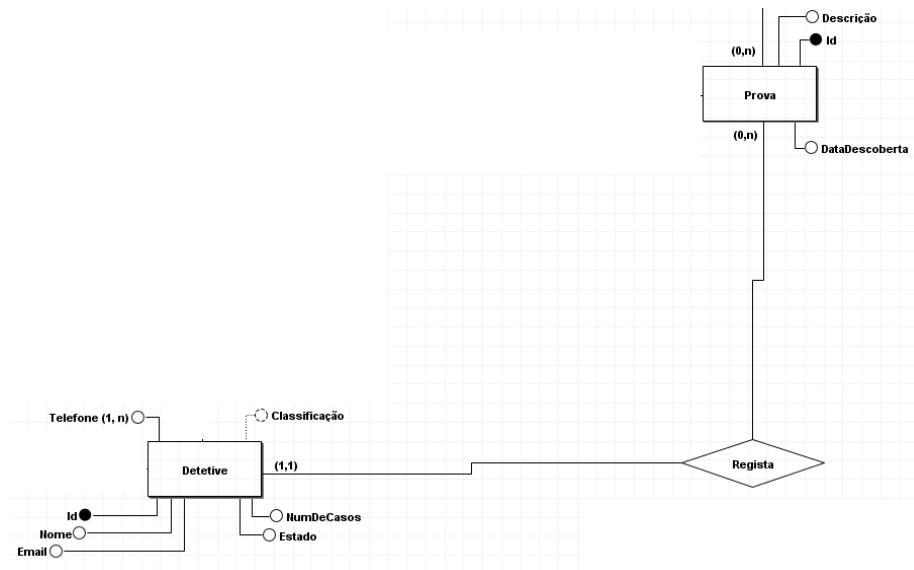


Figura 7 – Relacionamento Detetive - Prova

**Relacionamento:** Detetive Regista Prova.

**Descrição:** Um detetive durante a sua investigação pode, ou não, recolher provas. Esta informação deve ser registada no sistema pelo detetive de forma que a agência possa identificar o detetive que recolheu uma eventual prova num determinado caso.

**Cardinalidade:** Detetive (1..1) – Prova (0..\*).

**Atributos:** Este relacionamento não possui atributos.

- **Relacionamento Detetive – Relatório**

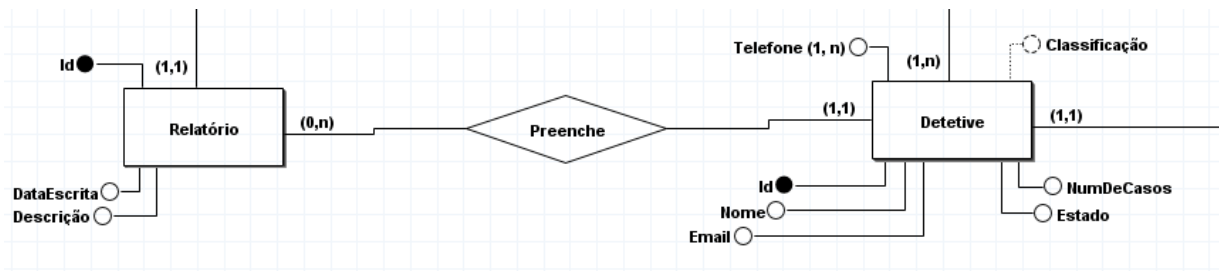


Figura 8 – Relacionamento Detetive – Relatório

**Relacionamento:** Detetive Preenche Relatório.

**Descrição:** Aquando da finalização da investigação o detetive responsável deve preencher um relatório sobre o caso. Esta informação é guardada para que a agência possa identificar qual o detetive que formulou um determinado relatório.

**Cardinalidade:** Detetive (1..1) – Relatório (0..\*).

Um detetive, caso não seja responsável por nenhum caso, não tem nenhum relatório preenchido. No entanto, um relatório é formulado apenas por um detetive (o responsável).

**Atributos:** Este relacionamento não possui atributos.

- **Relacionamento Relatório – Caso**

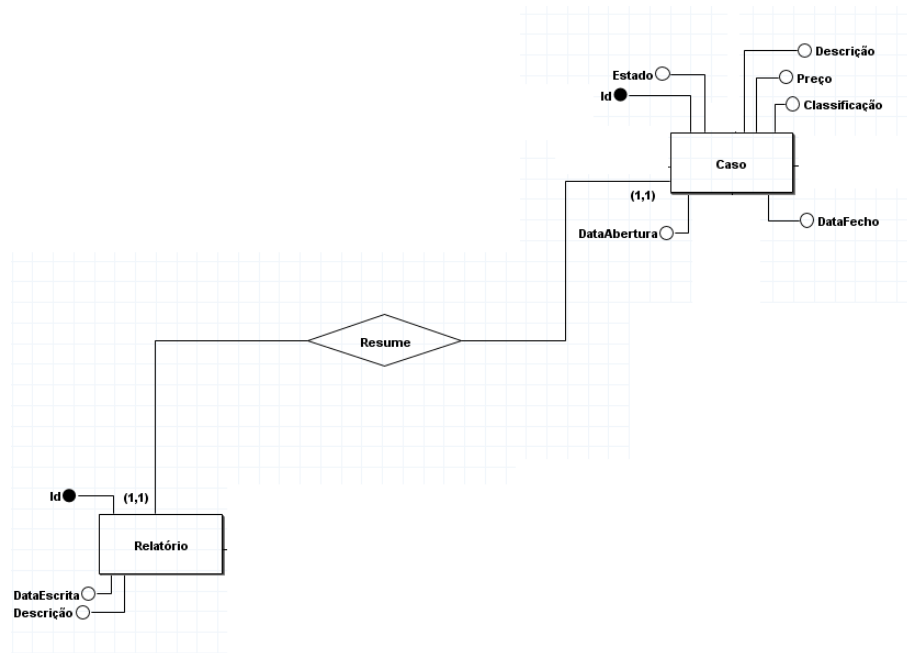


Figura 9 – Relacionamento Relatório - Caso

**Relacionamento:** Relatório Resume Caso.

**Descrição:** Quando é finalizada uma investigação, é formulado um relatório que vai resumir o caso. Esta informação deve ser armazenada para facilitar a visualização de algum detalhe importante pela agência.

**Cardinalidade:** Relatório (1..1) – Caso (1..1).

Sempre que existe um caso finalizado, este deve necessariamente ser acompanhado de um relatório logo um não pode existir sem o outro.

**Atributos:** Este relacionamento não possui atributos.



### 3.4. Identificação e Caracterização da Associação dos Atributos com as Entidades e Relacionamentos

Entidade	Atributos	Tipo de Dados	Nulo	Composto	Multivalor	Derivado	Candidato
<b>Caso</b>	Id	INT	Não	Não	Não	Não	Sim
	Estado	ENUM(10)	Não	Não	Não	Não	Não
	Descrição	TEXT(512)	Não	Não	Não	Não	Não
	Data de Abertura	DATETIME	Não	Não	Não	Não	Não
	Data de Fecho	DATETIME	Não	Não	Não	Não	Não
	Preço	DECIMAL(7,2)	Não	Não	Não	Não	Não
	Classificação	DECIMAL(4,2)	Sim	Não	Não	Não	Não
<b>Cliente</b>	NIF	INT	Não	Não	Não	Não	Sim
	Nome	VARCHAR(64)	Não	Não	Não	Não	Não
	Email	VARCHAR(45)	Não	Não	Não	Não	Sim
	Número de Telemóvel	VARCHAR(15)	Não	Não	Sim	Não	Sim
	Sexo	ENUM(2)	Não	Não	Não	Não	Não
	Tipo	ENUM(10)	Não	Não	Não	Não	Não
	Endereço	VARCHAR(128)	Não	Sim	Não	Não	Não
<b>Detetive</b>	Id	INT	Não	Não	Não	Não	Sim
	Nome	VARCHAR(64)	Não	Não	Não	Não	Não
	Email	VARCHAR(45)	Não	Não	Não	Não	Sim
	Número de Telemóvel	VARCHAR(15)	Não	Não	Sim	Não	Sim
	Número de Casos	INT	Não	Não	Não	Não	Não
	Classificação	DECIMAL(4,2)	Sim	Não	Não	Sim	Não
	Estado	ENUM(10)	Não	Não	Não	Não	Não
<b>Suspeito</b>	Id	INT	Não	Não	Não	Não	Sim
	Nome	VARCHAR(64)	Não	Não	Não	Não	Não
	Descrição	TEXT(512)	Não	Não	Não	Não	Não
<b>Prova</b>	Id	INT	Não	Não	Não	Não	Sim
	Descrição	TEXT(512)	Não	Não	Não	Não	Não
	Data de Descoberta	DATETIME	Não	Não	Não	Não	Não
<b>Relatório</b>	Id	INT	Não	Não	Não	Não	Sim
	Data de Escrita	DATETIME	Não	Não	Não	Não	Não
	Descrição	TEXT(512)	Não	Não	Não	Não	Não
<b>Relacionamento Detetive - Caso</b>	Responsável	BOOLEAN	Não	Não	Não	Não	Não

Tabela 1 – Caracterização de todos os atributos existentes.

Os atributos guardam valores que descrevem a ocorrência das entidades e representam a parte principal dos dados guardados na base de dados.

A maioria dos atributos identificados são simples, com um único valor, não são derivados e não podem ser nulos. Porém, existem algumas exceções, que passaremos a explicar em seguida.

- **Caso**

Todos os atributos desta entidade (id, estado, descrição, data de abertura, data de fecho, preço e classificação) são **simples, não derivados** e possuem **um único valor**.

A classificação poderá ser **nula** caso nenhum dos participantes classifique o evento.

- **Cliente**

Os atributos NIF, nome, email, sexo e tipo desta entidade são **simples** e possuem **um único valor**.

O atributo número de telemóvel é **simples** e **multivalorado**.

O atributo endereço é **composto** e possui **um único valor**.

Nenhum dos atributos desta entidade é **derivado**.

- **Detetive**

Todos os atributos desta entidade (id, nome, email, número de telemóvel, número de casos, classificação e estado) são **simples, com um único valor** e com **um único valor** à exceção do número de telemóvel e da classificação.

O número de telemóvel é um atributo **multivalorado**.

A classificação é um atributo **derivado** e poderá ser **nula** caso o detetive não tenha investigações.

A classificação do Detetive é calculada através da divisão da soma das classificações não nulas dos casos que está associado pelo número de classificações não nulas desses casos.

- **Suspeito**

Os atributos do Suspeito são todos não **nulos, com um único valor**, não **derivados** e não **compostos**.

- **Prova**

Todos os atributos desta entidade são **simples, com um único valor**, não **nulos** e não **derivados**.

- **Relatório**

Os atributos do Relatório são todos não **nulos, com um único valor**, não **derivados** e não **compostos**.

- **Relacionamento Detetive – Caso**

Todos os atributos deste relacionamento são **simples**, **com um único valor**, não **nulos** e não **derivados**.

### 3.4.1 Domínio dos Atributos

- **Caso**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo.
Estado	ENUM(10)	Aberto ou Fechado.
Descrição	TEXT(512)	Sequência de palavras.
Data de Abertura	DATETIME	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm]).
Data de Fecho	DATETIME	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm]).
Preço	DECIMAL(7,2)	Número decimal positivo.
Classificação	DECIMAL(4,2)	Número decimal entre 0 e 10.

Tabela 2 – Caracterização dos atributos do Caso

- **Cliente**

Atributos	Tipo de Dados	Domínio de Valores
NIF	INT	Número inteiro positivo.
Nome	VARCHAR(64)	Sequência de palavras.
Email	VARCHAR(45)	Sequência de caracteres alfanuméricos, seguida do caracter “@” e por fim o domínio do provedor de serviço de email.
Número de Telemóvel	VARCHAR(15)	Sequência de números guardados em formato <i>varchar</i> .
Sexo	ENUM(2)	Feminino ou Masculino.
Tipo	ENUM(10)	Singular ou Coletivo.
Endereço	VARCHAR(128)	Sequência de palavras.

Tabela 3 – Caracterização dos atributos do Cliente

- **Detetive**

Atributos	Tipo de Dados	Domínio de Valores
-----------	---------------	--------------------

Id	INT	Número inteiro positivo.
Nome	VARCHAR(64)	Sequência de palavras.
Email	VARCHAR(45)	Sequência de caracteres alfanuméricos, seguida do caracter "@" e por fim o domínio do provedor de serviço de email.
Número de Telemóvel	VARCHAR(15)	Sequência de números guardados em formato <i>varchar</i> .
Número de Casos	INT	Número inteiro positivo.
Classificação	DECIMAL(4,2)	Número decimal entre 0 e 10.
Estado	ENUM(10)	Ativo ou Inativo.

Tabela 4 – Caracterização dos atributos do Detetive

- **Suspeito**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo.
Nome	VARCHAR(64)	Sequência de palavras.
Descrição	TEXT(512)	Sequência de palavras.

Tabela 5 – Caracterização dos atributos do Suspeito

- **Prova**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo.
Descrição	TEXT(512)	Sequência de palavras.
Data de Descoberta	DATETIME	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm]).

Tabela 6 – Caracterização dos atributos da Prova

- **Relatório**

Atributos	Tipo de Dados	Domínio de Valores
Id	INT	Número inteiro positivo.
Data de Escrita	DATETIME	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm]).
Descrição	TEXT(512)	Sequência de palavras.

Tabela 7 – Caracterização dos atributos do Relatório

- **Relacionamento Detetive – Caso**

Atributos	Tipo de Dados	Domínio de Valores
Responsável	BOOLEAN	Número binário

Tabela 8 – Caracterização dos atributos de Relacionamento Detetive - Caso

### 3.4.2 Chaves Candidatas, Primárias e Alternativas

Com o intuito de simplificar o processo da explicação da escolha das chaves candidatas adicionamos à Tabela 1 uma coluna chamada “Candidato”. Para determinarmos se um atributo pode ser uma chave candidata pensamos como esse atributo se comporta no “mundo real”. Ou seja, um atributo será chave candidata caso não exista a hipótese de existir vários valores repetidos desse mesmo atributo.

- **Caso**

Chaves Candidatas: Id.

Chave Primária: Id.

Chaves Alternativas: Nenhuma.

Uma vez que a única chave candidata é o Id do Caso (valor sem significado prático, apenas aplicacional), será esta a chave primária do Caso.

- **Cliente**

Chaves Candidatas: NIF, número de telemóvel e e-mail.

Chave Primária: NIF.

Chaves Alternativas: Número de telemóvel e e-mail.

A chave primária do Cliente é o NIF uma vez que é a chave candidata que o representa mais eficiente e eficazmente visto ser uma identificação própria. Tanto o número de telemóvel como o e-mail poderão sofrer futuras alterações.

- **Detetive**

Chaves Candidatas: Id, número de telemóvel e e-mail.

Chave Primária: Id.

Chaves Alternativas: Número de telemóvel e e-mail.

A chave primária do Detetive é o Id do Detetive (valor sem significado prático, apenas aplicacional). Tanto o número de telemóvel como o e-mail poderão sofrer futuras alterações.

- **Suspeito**

Chaves Candidatas: Id.

Chave Primária: Id.

Chaves Alternativas: Nenhuma.

Uma vez que a única chave candidata é o Id do Suspeito (valor sem significado prático, apenas aplicativo), será esta a chave primária do Suspeito.

- **Prova**

Chaves Candidatas: Id.

Chave Primária: Id.

Chaves Alternativas: Nenhuma.

Uma vez que a única chave candidata é o Id da Prova (valor sem significado prático, apenas aplicativo), será esta a chave primária da Prova.

- **Relatório**

Chaves Candidatas: Id.

Chave Primária: Id.

Chaves Alternativas: Nenhuma.

Uma vez que a única chave candidata é o Id do Suspeito (valor sem significado prático, apenas aplicativo), será esta a chave primária do evento.

- **Relacionamento Detetive – Caso**

O relacionamento terá como chave primária uma chave composta pelas chaves primárias do Detetive (id) e Caso (id).

### 3.5. Apresentação e Explicação do Diagrama ER

Depois de explicada a importância e o significado de cada uma das entidades existentes no nosso sistema, cada um dos relacionamentos entre elas e os respectivos atributos, tanto das entidades, como dos relacionamentos, apresentamos o modelo conceptual:

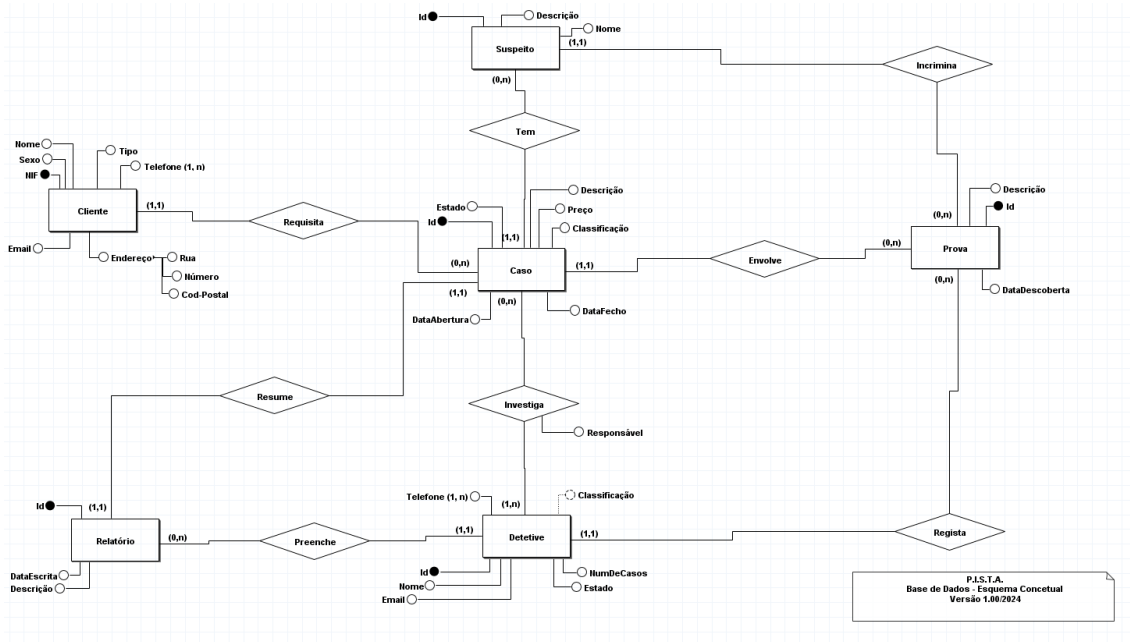


Figura 10 – Modelo Conceptual

Assim, o funcionamento do nosso sistema começa a ganhar textura. Traduzindo o Modelo Conceptual ilustrado anteriormente para texto temos que: O Cliente requisita um Caso. Esse Caso tem “Suspeito” e envolve “Prova”. O Relatório resume o Caso. O Detetive investiga o Caso e preenche o Relatório. O Detetive também regista a Prova. A Prova incrimina o Suspeito.

### 3.6. Validação do Modelo de Dados com o Utilizador

Após finalizado o modelo de dados, chegou a altura de nos reunirmos novamente com o utilizador para validarmos o nosso modelo conceptual final. Para isso, verificámos se o nosso modelo de dados consegue responder aos requisitos impostos:

1. Quando um detetive se regista, com o diretor da PISTA, este registo deve ter um nome, um contacto telefónico, um email, um estado, uma classificação e um número de casos em que participou (inicialmente 0).

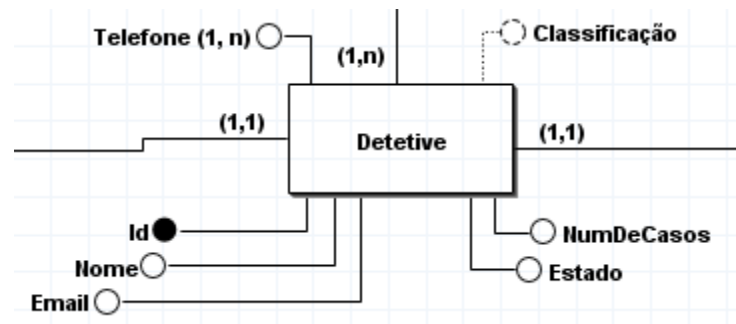


Figura 11 – Atributos do Detetive

A entidade “Detetive” tem associados todos os atributos referidos.

2. O detetive terá de ter necessariamente um dos seguintes estados: Ativo; Inativo.

O atributo tipo da entidade “Detetive” é um ENUM, com as alternativas presentes no requisito.

3. A classificação de cada detetive será uma média da classificação dada pelos clientes aos casos que investiga.

A entidade “Detetive” possui um atributo derivado chamado Classificação, como se pode ver na Figura 10

4. Um ou mais detetives poderão criar e editar casos, mas apenas um é responsável pelo caso.

Os detetives criam e editam casos requisitados pelos clientes e apenas um é responsável. Isto pode ser comprovado através do relacionamento da Figura 2 (Relacionamento entre o Detetive e o Caso).



5. Aquando do registo de um caso deverão ser indicados um identificador único (número), uma breve descrição, um estado, uma classificação, um preço, uma data de abertura do caso e uma data de fecho do caso.

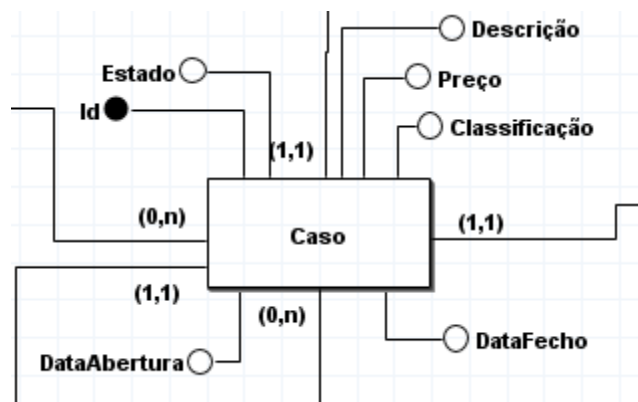


Figura 12 – Atributos do Caso

A entidade “Caso” tem associados todos os atributos referidos.

6. Todos os casos devem ser numerados sequencialmente por ordem de criação, não sendo permitido reutilizar números de casos anteriores que tenham sido fechados.

O Id do Caso é um valor sem significado prático, apenas aplicacional e é também a chave primária. Posto isto, ele é inserido sempre sequencialmente com a ordem do sistema, sempre que é registado um novo caso.

7. O caso terá de ter necessariamente um dos seguintes estados: Aberto; Fechado.

O atributo estado da entidade “Caso” é um ENUM, com as alternativas presentes no requisito.

8. Cada caso deverá estar associado obrigatoriamente ao cliente que o requisitou, para que o detetive responsável possa dialogar e acertar pormenores com ele sempre que necessário.

A Figura 1 (Relacionamento Cliente – Caso) comprova que o Cliente e o Caso têm uma associação no sistema.

9. Aquando do registo de um cliente deverão ser fornecidos o seu nome, um NIF, um contacto de telemóvel, um endereço, um email e um tipo.

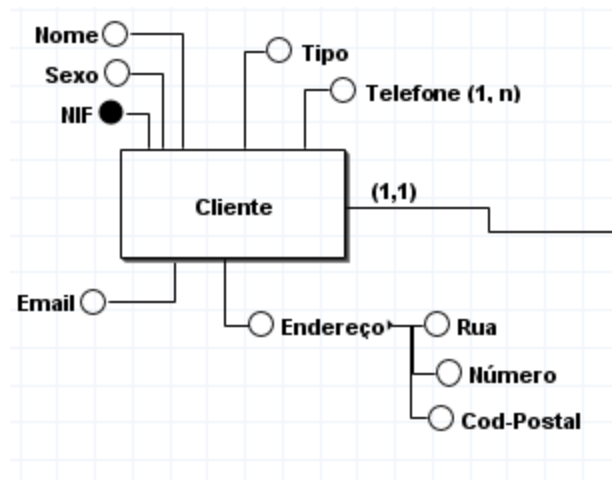


Figura 13 – Atributos do Cliente

A entidade Cliente tem associados todos os atributos referidos.

10. O cliente terá de ter necessariamente um dos seguintes tipos: Singular; Coletivo.

O atributo tipo da entidade “Cliente” é um ENUM, com as alternativas presentes no requisito.

11. Cada cliente poderá deixar uma classificação a cada caso que requisitou.

O Cliente poderá deixar uma classificação a cada Caso que requisitou, que será armazenada no atributo Classificação da entidade “Caso” (Figura 11).

12. A cada caso poderão ser registados os seus suspeitos.

Como podemos comprovar pela Figura 3 (Relacionamento Caso – Suspeito) cada Caso tem associado o(s) seu(s) Suspeito(s).

13. Aquando do registo de um suspeito deverão ser indicados um nome e uma breve descrição.

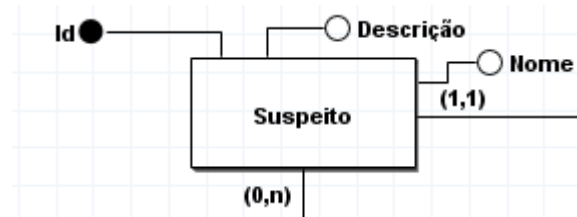


Figura 14 – Atributos do Suspeito

A entidade "Suspeito" tem associados todos os atributos referidos.

14. Para cada caso não pode haver mais do que um suspeito igual, ou seja, não pode haver suspeitos repetidos na lista de suspeitos do caso.

Cada Suspeito é identificado pelo seu Id, que é a sua chave primária, como comprovado pela Figura 13 (Atributos do Suspeito). Como a chave primária é única este requisito fica garantido.

15. A cada caso poderão ser registadas as suas provas.

A Figura 4 (Relacionamento Caso – Prova) demonstra que existe uma associação entre a entidade "Caso" e a entidade "Prova".

16. Aquando do registo de uma prova deverão ser indicados uma descrição e a data em que foi registada no sistema.

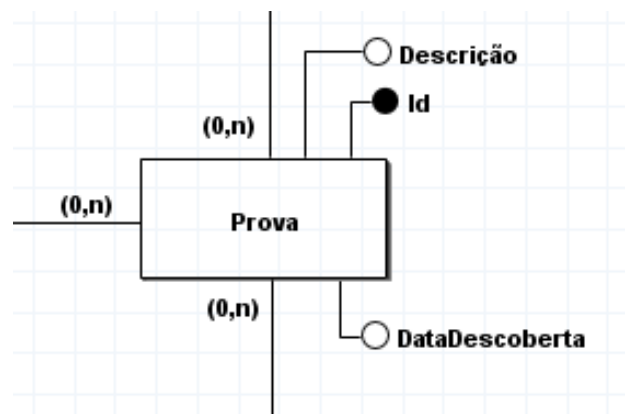


Figura 15 – Atributos da Prova

A entidade "Prova" tem associados todos os atributos referidos.

**17. Para cada caso não pode haver mais do que uma prova igual, ou seja, não pode haver provas repetidas na lista de provas do caso.**

Cada Prova é identificada pelo seu Id, que é a sua chave primária, como comprovado pela Figura 14 (Atributos da Prova). Como a chave primária é única este requisito fica garantido.

**18. Um caso apenas pode ser fechado quando o detetive responsável o validar.**

Como podemos ver pela Figura 2 (Relacionamento Detetive – Caso), cada caso tem um detetive responsável. O sistema apenas aceita que o caso seja fechado por este responsável.

**19. Cada caso terá necessariamente de ter um relatório escrito por um detetive associado ao caso.**

A Figura 8 (Relacionamento Relatório – Caso) demonstra que existe uma associação entre a entidade “Relatório” e a entidade “Caso”.

**20. Aquando do registo de um relatório deverão ser indicados uma descrição e a data em que foi registado no sistema.**

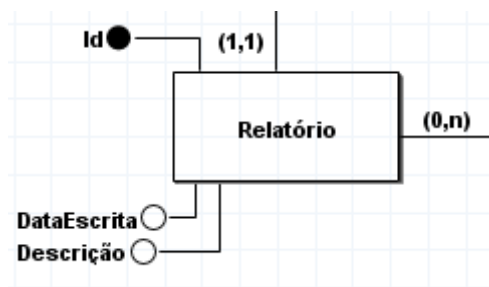


Figura 16 – Atributos do Relatório

A entidade “Relatório” tem associados todos os atributos referidos.

## 4. Modelação Lógica

### 4.1. Construção e Validação do Modelo de Dados Lógico

A modelação lógica desempenha um papel crucial na definição da estrutura da base de dados, proporcionando uma representação mais detalhada e operacional do que o modelo conceptual. No processo de construção do modelo lógico, cada entidade é mapeada para uma tabela, garantindo que todos os seus atributos sejam adequadamente representados. Além disso, os relacionamentos entre entidades são traduzidos em chaves primárias e estrangeiras, estabelecendo assim a integridade referencial entre as tabelas. Essa abordagem permite uma implementação mais precisa e eficiente da base de dados, refletindo fielmente os requisitos e estrutura do sistema em questão.

Dessa forma, procedeu-se à tradução do modelo conceptual para o modelo lógico, resultando nas seguintes tabelas:

- **Prova**



Figura 17 – Ilustração da tabela Entidade Prova

- **Suspeito**



Figura 18 - Ilustração da tabela Entidade Suspeito

- **Caso**

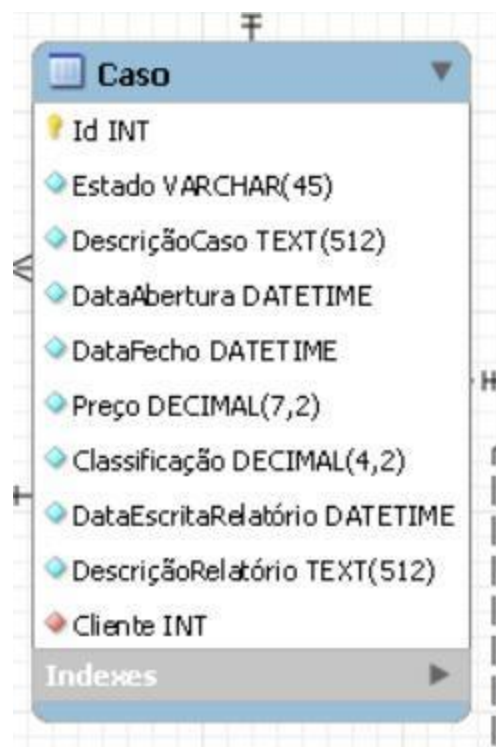


Figura 19 – Ilustração da tabela Entidade Caso

- **DetetiveTelefone**



Figura 20 – Ilustração da tabela Entidade DetetiveTelefone

- **Detetive**



Figura 21 – Ilustração da tabela Entidade Detetive

- **CasoDetetive**



Figura 22 – Ilustração da tabela Entidade CasoDetetive

- **ClienteTelefone**



Figura 23 – Ilustração da tabela Entidade ClienteTelefone

- **Cliente**

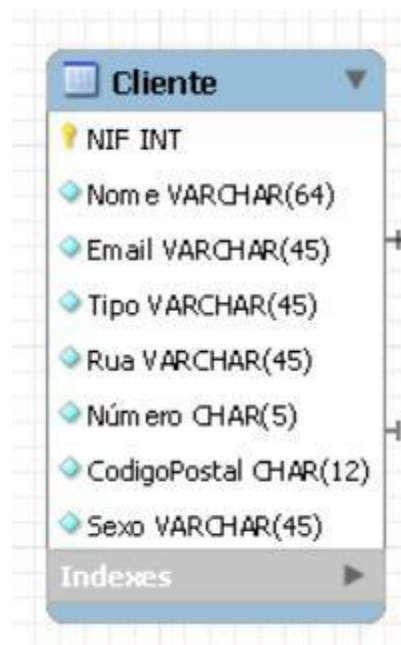


Figura 24 – Ilustração da tabela Entidade Cliente

## 4.2. Apresentação e Explicação do Modelo Lógico Produzido

Em primeiro lugar, foram identificadas as entidades e os seus atributos.

Para terminar a construção do modelo de dados lógicos foram ainda seguidas as regras de derivação dos relacionamentos do modelo de dados relacional.

1. **Relacionamento binário de grau 1:N com participação obrigatória do lado N:**

São necessárias duas entidades lógicas, uma para cada entidade, e a chave primária



da entidade do lado 1 tem de ser usada como atributo na entidade correspondente à entidade do lado N.

Ocorrências:

- “Tem” entre Suspeito(N) e Caso(1)
- “Requisita” entre Caso(N) e Cliente(1)
- “Regista” entre Prova(N) e Detetive(1)
- “Envolve” entre Prova(N) e Caso(1)
- “Incrimina” entre Prova(N) e Suspeito(1)

## **2. Relacionamento binário de grau N:M:**

São sempre necessárias três entidades lógicas neste tipo de relacionamentos, uma para cada entidade e uma para o relacionamento. As chaves primárias das entidades têm de ser atributos na entidade lógica do relacionamento.

Ocorrências:

- “Investiga” entre Detetive e Caso

## **3. Relacionamento binário de grau 1:1:**

São necessárias duas entidades lógicas, uma para cada entidade, e a chave primária de qualquer um dos lados tem de ser usada como atributo na entidade correspondente à entidade do lado N. Como na única ocorrência deste relacionamento necessariamente existem as duas entidades, juntamos os atributos de uma delas na outra, resultando apenas numa entidade.

Ocorrências:

- “Resume” entre Relatório e Caso

## **4. Atributo MultiValorado:**

Os atributos multivalorados são representados através da criação de uma nova tabela, onde cada valor do atributo multivalorado é armazenado como uma entrada separada nessa tabela. Esta tabela adicional está ligada à entidade original através de uma chave estrangeira, que mantém a relação entre os valores multivalorados e a entidade principal.

Ocorrências:

- Entre Telemóvel(N) e Detetive(1)
- Entre Telemóvel(N) e Cliente(1)

## **5. Atributo Composto:**

Os atributos compostos são decompostos em atributos simples, de acordo com os subatributos que os compõem.

Ocorrências:

- Atributo Endereço com subatributos Rua, Número e Código-Postal

#### Entidades resultantes:

- **Ciente:** {NIF, Nome, Email, Tipo, Rua, Número, CodigoPostal, Sexo}
- **CienteTelefone:** {Ciente, Telefone}
- **Suspeito:** {Id, Nome, Descrição, Caso}
- **Caso:** {Id, Estado, DescriçãoCaso, DataAbertura, DataFecho, Preço, Classificação, DataEscritaRelatório, DescriçãoRelatório, Cliente}
- **Detetive:** {Id, Nome, Email, NumeroCasos, Classificação, Estado}
- **DetetiveTelefone:** {Detetive, Telefone}
- **CasoDetetive:** {IdDetetive, IdCaso, Responsável}
- **Prova:** {Id, Descrição, DataDescoberta, Suspeito, Caso, Detetive}

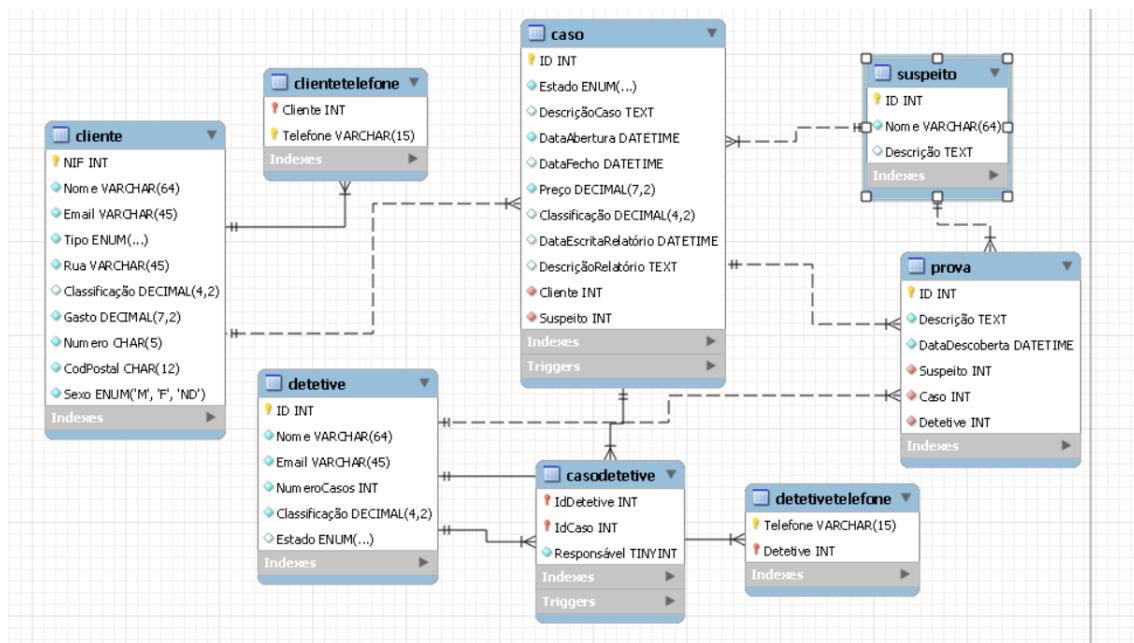


Figura 25 – Modelo Lógico

### 4.3. Validação do Modelo através da Normalização

A normalização de dados é um processo essencial neste projeto, cujo objetivo é organizar e estruturar os dados de forma a reduzir a redundância e aumentar a integridade.

Esse procedimento envolve a criação de tabelas e o estabelecimento de relações entre elas, seguindo regras específicas conhecidas como normalização. Ao normalizar uma base de dados, garantimos um armazenamento consistente e eficiente de acesso aos dados, mitigando inconsistências e tornando-a mais flexível.

Ao analisarmos o nosso modelo lógico, podemos confirmar que não há atributos duplicados e que quaisquer atributos compostos no Modelo Conceptual foram decompostos nos seus elementos constituintes numa nova tabela, satisfazendo, assim, os critérios da primeira forma normal.

Ao examinarmos a tabela do Caso, constatamos que o atributo DataFecho é unicamente dependente da chave primária, Id, assim como os demais atributos no resto do modelo. Também temos uma chave primária composta (na tabela “casodetetive”), mas como cada atributo normal depende de todas as partes da chave validamos a conformidade com a segunda forma normal.

Além disso, observamos que no nosso modelo lógico não existe qualquer atributo dependente ou proporcional a outro, cumprindo assim os requisitos da terceira forma normal.

Assim, concluímos que o nosso modelo lógico respeita as 3 fórmulas normais e se encontra, por isso, normalizado.

## **4.4. Validação do Modelo com as Interrogações do Utilizador**

De maneira a validar o modelo lógico desenvolvido, selecionamos algumas das questões mais relevantes que se encontram nos nossos requisitos da base de dados e explicamos de que forma o nosso modelo lógico consegue satisfazê-las:

## 1. Consultar os dados dos respectivos clientes de um dado caso

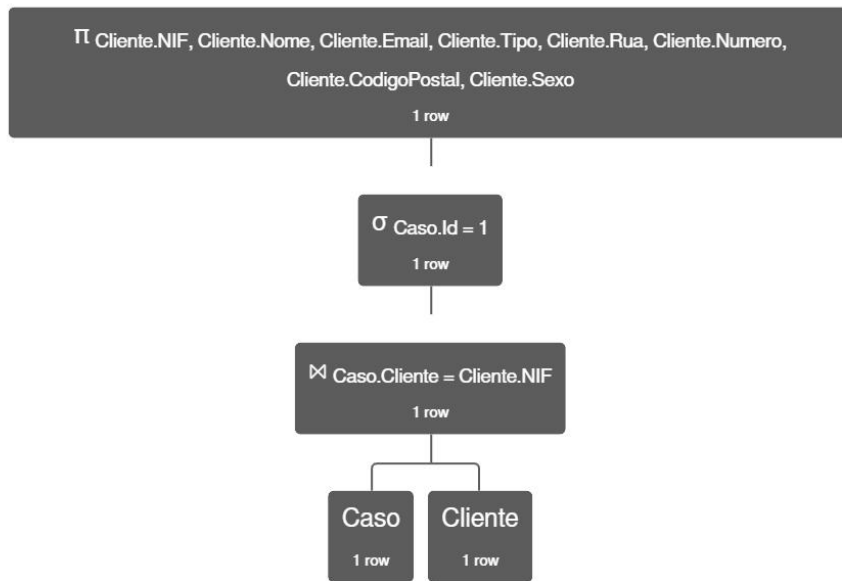


Figura 26 – Ilustração do modelo Relax da consulta dos dados dos clientes de um caso

$\pi$  Cliente.NIF, Cliente.Nome, Cliente.Email, Cliente.Tipo, Cliente.Rua, Cliente.Numero, Cliente.CodigoPostal, Cliente.Sexo  $\sigma$  Caso.Id = 1 (Caso  $\bowtie$  Caso.Cliente = Cliente.NIF Cliente)

## 2. Detetive com mais casos

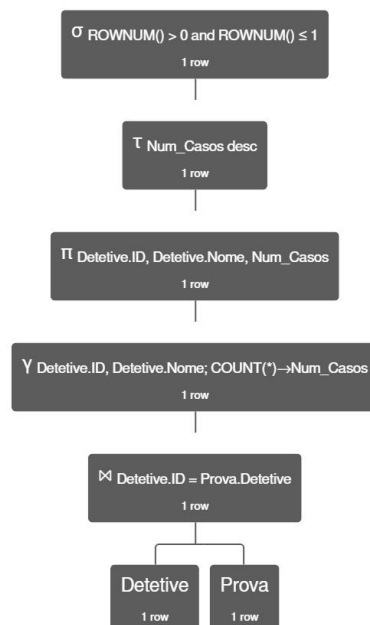


Figura 27 – Ilustração do modelo Relax da consulta do detetive com mais casos

$\sigma \text{ ROWNUM}() > 0 \text{ and } \text{ROWNUM}() \leq 1 \text{ } \tau \text{ Num\_Casos desc } \pi \text{ Detetive.ID, Detetive.Nome, Num\_Casos } \gamma \text{ Detetive.ID, Detetive.Nome; COUNT(*)} \rightarrow \text{Num\_Casos ( Detetive } \bowtie \text{ Detetive.ID = CasoDetetive.IdDetetive CasoDetetive )}$

### 3. Caso mais caro

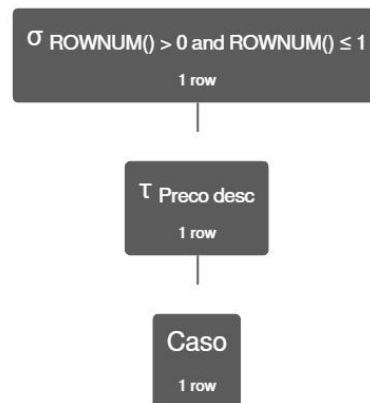


Figura 28 – Ilustração do modelo Relax da consulta do caso mais caro

$\sigma \text{ ROWNUM}() > 0 \text{ and } \text{ROWNUM}() \leq 1 \text{ } \tau \text{ Preço desc Caso}$

## 4.5. Revisão do Modelo Lógico com o Utilizador

Após concluído o desenvolvimento do modelo lógico foi agendada nova reunião com a P.I.S.T.A. a fim de validar este modelo e obter autorização para avançar para a próxima fase do projeto.

Uma vez que este modelo é significativamente mais complexo que o modelo conceptual, o primeiro passo foi apresentar em traços gerais como analisar e interpretar este tipo de modelos, para que o cliente possa, de maneira fiável, confirmar que o modelo atende às necessidades da empresa.

Depois de verificar que todas as entidades do modelo conceptual estavam presentes no modelo lógico, o cliente verificou que todas elas tinham os atributos necessários e com o domínio certo de modo a armazenar toda a informação pretendida. Finalmente, em conjunto com o cliente, foram formuladas várias perguntas de exemplo para verificar que a estrutura do modelo era capaz de responder a todas as perguntas colocadas do modo mais simples e eficiente possível.

Cumpridas todas as exigências do cliente, ele deu autorização para avançar para a próxima fase do projeto.

## **5. Alterações sobre a primeira parte**

Para a atualização do relatório de forma a coincidir com a segunda fase do projeto, fizemos algumas alterações.

Primeiramente, o Diagrama de Gantt foi atualizado, de forma a abranger tanto a primeira como a segunda fase.

Face à descoberta de algumas incorreções, foram alteradas as cardinalidades dos relacionamentos Cliente – Caso, Detetive – Caso, Caso – Suspeito e Caso – Prova.

No capítulo do modelo lógico, a figura 18 foi substituída, de forma a estar de acordo com o modelo lógico apresentado mais abaixo.

## **6. Implementação física**

### **6.1. Apresentação e explicação da base de dados implementada**

O Sistema de Gestão de Bases de Dados Relacional escolhido para implementar a base de dados concebida foi o MySQL. O MySQL é open-source, fácil de usar e possui uma grande comunidade ativa. Além destas razões, o facto do sistema de armazenamento e gestão de transações do MySQL, denominado InnoDB, seguir os princípios ACID — Atomicidade, Consistência, Isolamento e Durabilidade — constitui também um forte motivo para a adoção deste sistema de gestão de bases de dados relacionais.

Para usufruir da melhor forma dos mecanismos que o MySQL oferece, decidimos utilizar, durante todo o processo, a ferramenta de desenho, desenvolvimento e administração de bases de dados, MySQL Workbench.

A transição do modelo lógico para a implementação física em SQL envolveu a conversão das entidades e relacionamentos do modelo lógico em tabelas na base de dados, sendo necessário especificar os detalhes físicos, tais como tipos de dados e restrições. Esta abordagem sistemática assegura que a base de dados resultante seja uma representação fiel e eficiente do modelo de dados original. É de ressaltar que optámos por criar as tabelas que não possuem nos seus atributos chaves estrangeiras (cliente, suspeito e detetive) e só depois as restantes. Também é de apontar que foram introduzidas duas novas colunas na tabela cliente para melhorar o desempenho (Gasto e Classificação). Seguem-se as tabelas implementadas:

**Tabela Cliente:**

```
CREATE TABLE Cliente (  
    NIF INT PRIMARY KEY,  
    Nome VARCHAR(64) NOT NULL,  
    Email VARCHAR(45) NOT NULL,  
    Tipo ENUM ('Singular', 'Coletivo') NOT NULL,  
    Rua VARCHAR(45) NOT NULL,  
    Classificação DECIMAL(4,2),  
    Gasto DECIMAL(7,2) NOT NULL,  
    Numero CHAR(5) NOT NULL,  
    CodPostal CHAR(12) NOT NULL,  
    Sexo ENUM('M', 'F', 'ND') NOT NULL  
);
```

Figura 29 – Tabela Cliente

**Tabela ClienteTelefone:**

```
CREATE TABLE ClienteTelefone (  
    Cliente INT NOT NULL,  
    Telefone VARCHAR(15) NOT NULL,  
    PRIMARY KEY (Cliente, Telefone),  
    FOREIGN KEY (Cliente) REFERENCES Cliente (NIF)  
);
```

Figura 30 – Tabela ClienteTelefone

**Tabela Suspeito:**

```
CREATE TABLE Suspeito (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(64) NOT NULL,  
    Descrição TEXT  
);
```

Figura 31 – Tabela Suspeito



**Tabela Caso:**

```
CREATE TABLE Caso (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Estado ENUM ('Aberto', 'Fechado') NOT NULL,  
    DescriçãoCaso TEXT,  
    DataAbertura DATETIME NOT NULL,  
    DataFecho DATETIME,  
    Preço DECIMAL(7,2) NOT NULL,  
    Classificação DECIMAL(4,2) NOT NULL,  
    DataEscritaRelatório DATETIME,  
    DescriçãoRelatório TEXT,  
    Cliente INT NOT NULL,  
    Suspeito INT NOT NULL,  
    FOREIGN KEY (Cliente) REFERENCES Cliente (NIF),  
    FOREIGN KEY (Suspeito) REFERENCES Suspeito (ID)  
);
```

Figura 32 – Tabela Caso

**Tabela Detetive:**

```
CREATE TABLE Detetive (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(64) NOT NULL,  
    Email VARCHAR(45) NOT NULL,  
    NumeroCasos INT NOT NULL,  
    Classificação DECIMAL(4,2) NOT NULL,  
    Estado ENUM('Ativo', 'Inativo')  
);
```

Figura 33 – Tabela Detetive

**Tabela DetetiveTelefone:**

```
CREATE TABLE DetetiveTelefone (  
    Telefone VARCHAR(15) NOT NULL,  
    Detetive INT NOT NULL,  
    PRIMARY KEY (Telefone, Detetive),  
    FOREIGN KEY (Detetive) REFERENCES Detetive (ID)  
);
```

Figura 34 – Tabela DetetiveTelefone

**Tabela CasoDetetive:**

```
CREATE TABLE CasoDetetive (  
    IdDetetive INT NOT NULL,  
    IdCaso INT NOT NULL,  
    Responsável TINYINT NOT NULL,  
    PRIMARY KEY (IdDetetive, IdCaso),  
    FOREIGN KEY (IdDetetive) REFERENCES Detetive (ID),  
    FOREIGN KEY (IdCaso) REFERENCES Caso (ID)  
);
```

Figura 35 – Tabela CasoDetetive

**Tabela Prova:**

```
CREATE TABLE Prova (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Descrição TEXT NOT NULL,  
    DataDescoberta DATETIME NOT NULL,  
    Suspeito INT NOT NULL,  
    Caso INT NOT NULL,  
    Detetive INT NOT NULL,  
    FOREIGN KEY (Suspeito) REFERENCES Suspeito (ID),  
    FOREIGN KEY (Caso) REFERENCES Caso (ID),  
    FOREIGN KEY (Detetive) REFERENCES Detetive (ID)  
);
```

Figura 36 – Tabela Prova

## 6.2. Criação de utilizadores da base de dados

A caracterização dos utilizadores da base de dados é essencial para garantir a segurança e a eficiência na gestão dos dados. Neste contexto, apresentamos dois perfis de utilizadores distintos, cada um com permissões de trabalho específicas que refletem as suas responsabilidades e necessidades de acesso. O primeiro exemplo é o Administrador do Sistema, responsável pela manutenção geral da base de dados, configuração de permissões e monitorização de desempenho. Este perfil possui permissões amplas, incluindo leitura, escrita, atualização e eliminação de dados, além da capacidade de criar e alterar estruturas de tabelas e índices. O segundo exemplo é o Detetive, cuja principal função é a gestão de casos, relatórios, suspeitos e provas. Este perfil possui permissões restritas a leitura e consultas avançadas, sem a capacidade de modificar em completo a estrutura da base de dados. A seguir, mostramos os perfis de utilizadores criados:

### Utilizador administrador:

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'senha';

GRANT SELECT, INSERT, UPDATE, DELETE ON PISTA.* TO 'admin'@'localhost';

GRANT EXECUTE ON PROCEDURE PISTA.dados_cliente_do_caso TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.dados_prova_do_caso TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.dados_suspeito_do_caso TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_do_detetive TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.relatorios_do_detetive TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_do_cliente TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.detetives_estado_casos TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_classificacao TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.detetives_classificacao TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_data_a_data TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_data_a_data_fecho TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.maiores_clientes TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.clientes_classificacao TO 'admin'@'localhost';

GRANT EXECUTE ON PROCEDURE PISTA.adicionar_cliente TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_cliente_telefone TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_suspeito TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_caso TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_detetive TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_detetive_telefone TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_detetive_caso TO 'admin'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_prova TO 'admin'@'localhost';

GRANT SELECT ON PISTA.verDetetive TO 'admin'@'localhost';
GRANT SELECT ON PISTA.verCasos TO 'admin'@'localhost';
GRANT SELECT ON PISTA.verClientes TO 'admin'@'localhost';
GRANT SELECT ON PISTA.verSuspeitos TO 'admin'@'localhost';
GRANT SELECT ON PISTA.verProvas TO 'admin'@'localhost';
```

Utilizador detetive:

```
CREATE USER 'detetive'@'localhost' IDENTIFIED BY 'senha';

GRANT SELECT ON PISTA.Cliente TO 'detetive'@'localhost';
GRANT SELECT ON PISTA.Caso TO 'detetive'@'localhost';
GRANT SELECT ON PISTA.Suspeito TO 'detetive'@'localhost';
GRANT SELECT ON PISTA.Prova TO 'detetive'@'localhost';

GRANT EXECUTE ON PROCEDURE PISTA.dados_cliente_do_caso TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.dados_prova_do_caso TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.dados_suspeito_do_caso TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.relatorios_do_detetive TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_do_cliente TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_data_a_data TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.casos_data_a_data_fecho TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.maiores_clientes TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.clientes_classificacao TO 'detetive'@'localhost';

GRANT EXECUTE ON PROCEDURE PISTA.adicionar_suspeito TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_caso TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE PISTA.adicionar_prova TO 'detetive'@'localhost';

GRANT SELECT ON PISTA.verCasos TO 'detetive'@'localhost';
GRANT SELECT ON PISTA.verClientes TO 'detetive'@'localhost';
GRANT SELECT ON PISTA.verSuspeitos TO 'detetive'@'localhost';
GRANT SELECT ON PISTA.verProvas TO 'detetive'@'localhost';
```

### 6.3. Povoamento da base de dados

De maneira a tornar o povoamento da base de dados o mais simples e mais conveniente possível para nós, optamos por desenvolver um *script* em *Python* que fará esta tarefa. O *script* seguirá em anexo.

Este *script* possui uma ligação à base de dados através do “*mysql.connector*” e insere alguns registos através das instruções *INSERT*.

Todos os registos são gerados aleatoriamente através de funções *random* oferecidas pela linguagem. Os nomes, ruas e descrições são gerados através de escolha randomizada sobre uma lista previamente criada com alguns nomes e descrições.

### 6.4. Cálculo do espaço da base de dados

O cálculo das estimativas de espaço baseia-se no capítulo 13.7 do *MySQL 8.0 Reference Manual*: “*Data Type Storage Requirements*”.

Os valores foram calculados somando o número de bytes necessários para cada atributo e encontram-se na folha de cálculo disponibilizada em anexo.

Por cada Caso criado temos de ter pelo menos 1 Suspeito e pelo menos 1 Prova. Isto irá resultar num espaço ocupado em tabelas de 2142 bytes. A empresa espera receber 200 a 300 casos no próximo ano, o que irá representar um crescimento da base de dados em 428.4 kB a 642.6 kB.

Por cada Cliente criado temos de ter pelo menos 1 contacto associado. Isto irá resultar num espaço ocupado em tabelas de 201 bytes. A empresa espera ter um crescimento 50 a 100 clientes no próximo ano, o que irá representar um aumento do tamanho da tabela de 10.05 kB a 20.1 kB.

Por cada Detetive criado temos de ter pelo menos 1 contacto associado. Isto irá resultar num espaço ocupado em tabelas de 144 bytes. A empresa espera contratar entre 5 a 10 detetives no próximo ano, o que irá representar um aumento do tamanho da tabela de 720 bytes a 1440 bytes.

Esperamos ter também 2 a 5 detetives associados a cada caso. Isto resulta num espaço ocupado em tabelas de 18 bytes a 45 bytes.

Esperamos assim ter um crescimento da base de dados esperado de 1306.8 kB.

## 6.5. Definição e caracterização de vistas de utilização em SQL

As vistas de utilização são essenciais para uma rápida e prática visualização de dados relevantes. Seguem-se as vistas criadas:

1. **verDetetive**: Esta vista permite uma fácil compreensão dos detetives na base de dados.

```
-- View para listar detetives e seus casos associados.  
CREATE VIEW verDetetive AS  
SELECT  
    ID,  
    Nome,  
    Email,  
    NumeroCasos,  
    Classificação,  
    Estado  
FROM  
    Detetive;
```

Figura 37 – Vista verDetetive

2. **verCasos**: Oferece uma visão abrangente dos casos presentes no sistema.

```
-- View para listar casos e seus clientes associados
CREATE VIEW verCasos AS
SELECT
    Caso.ID,
    Caso.Estado,
    Caso.DescriçãoCaso,
    Caso.DataAbertura,
    Caso.DataFecho,
    Caso.Preço,
    Caso.Classificação,
    Caso.DataEscritaRelatório,
    Caso.DescriçãoRelatório,
    Cliente.Nome AS ClienteNome,
    Cliente.Email AS ClienteEmail,
    Cliente.Tipo AS ClienteTipo,
    Cliente.Rua AS ClienteRua,
    Cliente.Numero AS ClienteNumero,
    Cliente.CodPostal AS ClienteCodPostal,
    Cliente.Sexo AS ClienteSexo
FROM
    Caso
JOIN
    Cliente ON Caso.Cliente = Cliente.NIF;
```

Figura 38 – Vista verCasos

3. **verSuspeitos**: Esta vista destaca os suspeitos vinculados a cada caso.

```
-- View para listar suspeitos
CREATE VIEW verSuspeitos AS
SELECT
    Caso.ID AS CasoID,
    Caso.DescriçãoCaso AS CasoDescricao,
    Suspeito.ID AS SuspeitoID,
    Suspeito.Nome AS SuspeitoNome,
    Suspeito.Descrição AS SuspeitoDescricao
FROM
    Caso
JOIN
    Suspeito ON Caso.Suspeito = Suspeito.ID;
```

Figura 39 – Vista verSuspeitos



4. **verClientes:** Aqui, é possível visualizar todos os clientes registados no sistema, incluindo informações como nome, tipo (singular ou coletivo) e detalhes de contato.

```
-- View para listar todos os clientes
CREATE VIEW verClientes AS
SELECT
    NIF,
    Nome,
    Email,
    Tipo,
    Rua,
    Classificação,
    Preço,
    Numero,
    CodPostal,
    Sexo
FROM
    Cliente;
```

Figura 40 – Vista verClientes

5. **verProvas:** Esta vista apresenta uma visão detalhada das provas relacionadas a cada caso, juntamente com o detetive associado e o suspeito envolvido, oferecendo uma compreensão completa do progresso da investigação.

```
-- View para listar provas
CREATE VIEW verProvas AS
SELECT
    p.ID AS ProvaID,
    p.Descrição AS ProvaDescrição,
    c.ID AS CasoID,
    c.DescriçãoCaso AS CasoDescrição,
    d.ID AS DetetiveID,
    d.Nome AS DetetiveNome,
    s.ID AS SuspeitoID,
    s.Nome AS SuspeitoNome
FROM
    Prova p
JOIN
    Caso c ON p.Caso = c.ID
JOIN
    Detetive d ON p.Detetive = d.ID
JOIN
    Suspeito s ON p.Suspeito = s.ID;
```

Figura 40 – Vista verProvas

Estas vistas proporcionam uma maneira eficaz de aceder e interpretar os dados, que permite uma análise mais rápida e informada das informações armazenadas na base de dados.

## 6.6. Tradução das interrogações do utilizador para SQL

Tendo em conta os requisitos de exploração traçados e a confirmação do método a ser seguido para o tratamento das interrogações do utilizador, foram criadas as seguintes queries SQL:

- a) O detetive responsável poderá ser capaz de consultar os dados dos respetivos clientes de um dado caso por este requisitado:

```
DELIMITER $$
CREATE PROCEDURE dados_cliente_do_caso(IN IDdocaso INT)
BEGIN
SELECT
    cliente.*
FROM
    cliente
    INNER JOIN caso ON cliente.NIF = caso.Cliente
WHERE
    caso.ID = IDdocaso;
END $$
DELIMITER ;
```

Figura 44 – Procedure dados\_cliente\_do\_caso

- b) Um detetive poderá consultar quais as provas associadas a casos que está a investigar:

```
DELIMITER $$
CREATE PROCEDURE dados_prova_do_caso(IN IDdocaso INT)
BEGIN
SELECT
    prova.*
FROM
    prova
    INNER JOIN caso ON prova.ID = caso.ID
WHERE
    caso.ID = IDdocaso;
END $$
DELIMITER ;
```

Figura 45 – Procedure dados\_prova\_do\_caso



- c) Um detetive deverá ser capaz de consultar os suspeitos dos casos que está a investigar:

```
DELIMITER $$
CREATE PROCEDURE dados_suspeito_do_caso(IN IDdocaso INT)
BEGIN
    SELECT
        suspeito.*
    FROM
        suspeito
    INNER JOIN caso ON suspeito.ID = caso.Suspeito
    WHERE
        caso.ID = IDdocaso;
END $$
DELIMITER ;
```

Figura 46 – Procedure dados\_suspeito\_do\_caso

- d) Um detetive deverá ser capaz de consultar os casos que está a investigar:

```
DELIMITER $$
CREATE PROCEDURE casos_do_detetive(IN IDdetetive INT)
BEGIN
    SELECT
        caso.*
    FROM
        caso
    INNER JOIN casodetetive ON caso.ID = casodetetive.IdCaso
    INNER JOIN detetive ON casodetetive.IdDetetive = detetive.ID
    WHERE
        detetive.ID = IDdetetive;
END $$
DELIMITER ;
```

Figura 47 – Procedure casos\_do\_detetive

- e) Um detetive deverá ser capaz de consultar os relatórios que preencheu dos casos que investigou:

```
DELIMITER $$
CREATE PROCEDURE relatorios_do_detetive(IN IDdetetive INT)
BEGIN
    SELECT
        caso.DescriçãoRelatório,
        caso.DataEscritaRelatório
    FROM
        caso
    INNER JOIN casodetetive ON caso.ID = casodetetive.IdCaso
    INNER JOIN detetive ON casodetetive.IdDetetive = detetive.ID
    WHERE
        caso.Estado = 2 AND detetive.ID = IDdetetive;
END $$
DELIMITER ;
```

Figura 48 – Procedure relatórios\_do\_detetive

- f) Um cliente deverá ser capaz de consultar os casos que requisitou:

```
DELIMITER $$
CREATE PROCEDURE casos_do_cliente(IN NIFcliente INT)
BEGIN
SELECT
    caso.*
FROM
    caso
    INNER JOIN cliente ON caso.Cliente = cliente.NIF
WHERE
    cliente.NIF = NIFcliente;
END $$
DELIMITER ;
```

Figura 49 – Procedure casos\_do\_cliente

- g) O diretor da PISTA deve poder filtrar os seus detetives por estado e por número de casos em que participou:

```
DELIMITER $$
CREATE PROCEDURE detetives_estado_casos(IN numero_casos INT, IN estado INT)
BEGIN
SELECT
    *
FROM
    detetive
WHERE
    NumeroCasos >= numero_casos AND Estado = estado;
END $$
DELIMITER ;
```

Figura 50 – Procedure detetives\_estados\_casos

- h) O diretor da PISTA deve poder filtrar os seus detetives, os clientes e os casos por classificação:

```
DELIMITER $$
CREATE PROCEDURE casos_classificacao()
BEGIN
SELECT
    caso.ID,
    caso.Classificação
FROM
    caso
ORDER BY
    Classificação;
END $$
DELIMITER ;
```

Figura 51 – Procedure casos\_classificacao

- i) O diretor da PISTA poderá consultar, para um número indicado de clientes, os que mais dinheiro gastam em casos no geral e por tipo de caso:

```
DELIMITER $$
CREATE PROCEDURE maiores_clientes(IN numero_clientes INT)
BEGIN
SELECT
    cliente.Nome,
    cliente.Gasto
FROM
    cliente
ORDER BY gasto DESC
LIMIT numero_clientes;
END $$
DELIMITER ;
```

Figura 52 – Procedure maiores\_clientes

- j) O sistema deverá ser capaz de contabilizar o número de casos registrados num determinado intervalo de tempo:

```
DELIMITER $$
CREATE PROCEDURE casos_data_a_data(IN data_inicio DATE, IN data_fim DATE)
BEGIN
SELECT
    COUNT(*) AS row_count
FROM
    caso
WHERE
    DataAbertura BETWEEN data_inicio AND data_fim;
END $$
DELIMITER ;
```

Figura 53 – Procedure casos\_data\_a\_data

- k) O sistema deverá ser capaz de contabilizar o número de casos fechados num determinado intervalo de tempo:

```
DELIMITER $$
CREATE PROCEDURE casos_data_a_data_fecho(IN data_inicio DATE, IN data_fim DATE)
BEGIN
SELECT
    COUNT(*) AS row_count
FROM
    caso
WHERE
    (DataFecho BETWEEN data_inicio AND data_fim)
    AND
    (Estado = 1);
END $$
DELIMITER ;
```

Figura 54 – Procedure casos\_data\_a\_data\_fecho

## 6.7. Indexação do Sistema de Dados

De maneira a aumentar a eficiência das procuras, optamos por adicionar alguns índices. A introdução destes índices permite reduzir o tempo de execução das procuras e ao mesmo tempo permitem ordenar a informação atribuída.

Alguns dos índices criados foram as chaves estrangeiras (Suspeito, Caso e Detetive) na tabela Prova.

```
-- Indices Prova
CREATE INDEX idx_prova_suspeito ON Prova (Suspeito);
CREATE INDEX idx_prova_caso ON Prova (Caso);
CREATE INDEX idx_prova_detetive ON Prova (Detetive)
```

Figura 55 - Índices da tabela Prova

Outros índices adicionados foram os relativos aos casos, que permitem melhorar a eficiência de procuras na tabela.

```
-- Indices Caso
CREATE INDEX idx_caso_cliente ON Caso (Cliente);
CREATE INDEX idx_caso_suspeito ON Caso (Suspeito);
CREATE INDEX idx_caso_dataabertura ON Caso (DataAbertura);
CREATE INDEX idx_caso_estado ON Caso (Estado);
```

Figura 56 - Índices tabela Caso

## 6.8. Implementação de procedimentos, funções e gatilhos

Os triggers apresentados a seguir foram desenvolvidos para automatizar a atualização de determinados campos em diferentes tabelas da base de dados, em resposta a operações de inserção e atualização na tabela de casos. Estes triggers garantem a consistência e integridade dos dados ao calcular e atualizar automaticamente os gastos dos clientes, a classificação dos clientes e dos detetives, assim como o número de casos atribuídos a cada

detetive. Através da utilização destes triggers, conseguimos minimizar a necessidade de intervenção manual e reduzir a probabilidade de erros, assegurando que as informações refletidas na base de dados estão sempre atualizadas e corretas. Abaixo, detalhamos cada um dos triggers implementados.

- **Trigger atualizar\_cliente\_gasto:**

```
DELIMITER $$
CREATE TRIGGER atualizar_cliente_gasto
AFTER INSERT ON Caso
FOR EACH ROW
BEGIN
    UPDATE Cliente
    SET Gasto = Gasto + NEW.Preço
    WHERE NIF = NEW.Cliente;
END$$
DELIMITER ;
```

Figura 57 – Trigger atualizar cliente gasto

- **Trigger atualizar\_cliente\_classificacao:**

```
DELIMITER $$
CREATE TRIGGER atualizar_cliente_classificacao
AFTER UPDATE ON Caso
FOR EACH ROW
BEGIN
    DECLARE avg_classificacao DECIMAL(4,2);

    SELECT AVG(Classificação) INTO avg_classificacao
    FROM Caso
    WHERE Cliente = NEW.Cliente
    AND Estado = 2;

    UPDATE Cliente
    SET Classificação = avg_classificacao
    WHERE NIF = NEW.Cliente;
END$$
DELIMITER ;
```

Figura 58 – Trigger atualizar cliente classificação

- **Trigger update\_detetive\_classificacao:**

```

DELIMITER $$
CREATE TRIGGER update_detetive_classificacao
AFTER UPDATE ON Caso
FOR EACH ROW
BEGIN
    DECLARE avg_classificacao DECIMAL(4,2);
    DECLARE detetive_id INT;

    SELECT IdDetetive INTO detetive_id
    FROM CasoDetetive
    WHERE IdCaso = NEW.ID;

    SELECT AVG(c.Classificação) INTO avg_classificacao
    FROM Caso c
    JOIN CasoDetetive cd ON c.ID = cd.IdCaso
    WHERE cd.IdDetetive = detetive_id
    AND c.Estado = 2;

    UPDATE Detetive
    SET Classificação = avg_classificacao
    WHERE ID = detetive_id;
END$$
DELIMITER

```

Figura 59 – Trigger update detetive classificação

- **Trigger incrementar\_numero\_casos:**

```

DELIMITER $$
CREATE TRIGGER incrementar_numero_casos
AFTER INSERT ON CasoDetetive
FOR EACH ROW
BEGIN
    UPDATE Detetive
    SET NumeroCasos = NumeroCasos + 1
    WHERE ID = NEW.IdDetetive;
END$$
DELIMITER

DROP TRIGGER IF EXISTS update_detetive_classificacao;
DROP TRIGGER IF EXISTS incrementar_numero_casos;

```

Figura 60 – Trigger incrementar número casos

## 7. Conclusões e Trabalho Futuro

Tendo em conta a base de dados desenvolvida, acreditamos que iremos ajudar a P.I.S.T.A a crescer tanto como empresa como agência de investigação. A implementação desta tecnologia proporciona à empresa uma capacidade de resposta adequada às suas crescentes necessidades, garantindo consistência, integridade, segurança, concorrência, acessibilidade e facilidade de manutenção.

Os resultados obtidos são fruto de um longo processo de estudo, dividido em múltiplas partes que se uniram para formar um sistema coeso e robusto. Esta divisão permitiu um processo estável na criação do sistema de armazenamento de dados, sempre com o foco em atender a totalidade dos requisitos estabelecidos.

Ao longo deste relatório, foram apresentadas todas as condições do sistema onde a base de dados se inseriria, as suas necessidades e os métodos utilizados. Inicialmente, abordámos o contexto da P.I.S.T.A, detalhando e analisando os requisitos da base de dados desejada. Em seguida, apresentámos uma modelagem conceptual, transformando-a num modelo lógico, que foi posteriormente implementado num sistema de gestão de base de dados relacional (SGBDR). Finalmente, a implementação do sistema foi realizada com base em SQL.

Os benefícios de utilizar uma base de dados relacional são inúmeros, incluindo a facilitação da estruturação e acesso aos dados de forma organizada e eficiente, a flexibilidade para realizar consultas complexas e integrar diferentes tipos de dados, e as restrições de integridade que ajudam a manter a precisão e a consistência dos dados. A escalabilidade permite que a base de dados cresça juntamente com as necessidades da empresa, suportando um aumento no volume de dados e nas operações. A facilidade de manutenção, com operações como backups e atualizações sendo mais simples de realizar, também é um destaque.

A nossa solução foi projetada com a possibilidade de futuras expansões e melhorias em mente. Funcionalidades adicionais que podem ser implementadas posteriormente incluem ferramentas de Business Intelligence (BI) e Analytics para obter insights mais profundos e a capacidade de integrar a base de dados com outras aplicações e sistemas da P.I.S.T.A, aumentando a eficiência operacional.

Em resumo, a implementação desta base de dados relacional coloca a P.I.S.T.A numa posição vantajosa, preparada para enfrentar os desafios futuros com um sistema de gestão de dados sólido e eficiente.



## **8. Bibliografia**

<https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html>

## Referências

Connolly, T. and Begg, C. (2005). Database Systems, A Practical Approach to Design, Implementation, and Management. 4th ed. Addison-Wesley.

## **Lista de Siglas e Acrónimos**

**P.I.S.T.A.** Peritos em Investigação e Solução de Tramas Alternadas

**BD** Base de Dados

**NIF** Número de Identificação Fiscal