



Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Lectivo de 2023/2024

“Projeto Tricórnio”

**Gonçalo Freitas a104350, João Ferreira
a104539, Tiago Barros a104530, Leonor
Cunha a103997, Eduarda Vieira a104098**

B D Abril 2024

Data de Recepção	
Responsável	
Avaliação	
Observações	

“Projeto Tricórnio”

Gonçalo Freitas a104350, João Ferreira
a104539, Tiago Barros a104530, Leonor
Cunha a103997, Eduarda Vieira a104098

Abril 2024

Resumo

Este trabalho foi realizado no âmbito da Unidade Curricular de Bases de Dados, com foco na análise, planeamento, modulação, arquitetura e implementação deste tipo de sistemas. Ao longo deste relatório vamos desenvolver a formulação de uma base de dados de uma Agência de Detetives, apresentando as diferentes fases da elaboração e justificando as escolhas tomadas durante a produção do projeto.

A primeira parte deste projeto, refere-se à definição do sistema, onde fundamentamos a implementação de uma base de dados relacional.

Já na segunda parte, apresentamos os requisitos de descrição, manipulação e de controlo por nós definidos.

Em seguida, na terceira parte, é apresentado o modelo conceitual para a base de dados que queremos implementar. São identificadas as entidades, bem como os seus atributos e as relações entre as diferentes entidades.

Por último, na quarta parte, passamos para a implementação do modelo lógico, obtido através do modelo conceitual previamente definido.

Área de Aplicação: Desenho e arquitetura de Sistemas de Bases de Dados, no contexto de uma Agência de Detetives

Palavras-Chave: Bases de dados, Bases de dados relacionais, Modelo Conceptual, Modelo Lógico, Modelo Físico, MySQL

Índice

1. Definição do Sistema	1
1.1. Contexto de Aplicação	1
1.2. Motivação e Objetivos do Trabalho	1
1.3. Análise da Viabilidade do processo	2
1.4. Recursos e Equipa de Trabalho	3
1.5. Plano de Execução do Projeto	4
2. Levantamento e Análise de Requisitos	5
2.1. Método de Levantamento e de Análises de Requisitos Adotado	5
2.2. Organização dos Requisitos Levantados	6
2.2.1. Requisitos de Descrição	6
2.2.2. Requisitos de Manipulação	8
2.2.3. Requisitos de Controlo	9
2.3. Análise e Validação Geral dos Requisitos	9
3. Modelação Conceptual	10
3.1. Apresentação da Abordagem de Modelação Realizada	10
3.2. Identificação e Caracterização das Entidades	11
3.3. Identificação e Caracterização dos Relacionamentos	12
3.4. Identificação e Caracterização dos Atributos das Entidades e dos Relacionamentos	12
3.5. Apresentação e Explicação do Diagrama ER Produzido	16
4. Modelação Lógica	17
4.1. Construção e Validação do Modelo de Dados Lógico	17
4.1.1. Entidades Fortes	17
4.1.2. Entidades Fracas	19
4.2. Apresentação e Explicação do Modelo	20
4.3. Normalização de Dados	21
4.4. Validação do Modelo com Interrogações do Utilizador	22
5. Implementação física	24
5.1 Apresentação e explicação da base de dados implementada	24
5.2 . Criação de utilizadores da base de dados	29
5.3 . Povoamento da base de dados	30
5.4 . Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)	34

5.5 . Definição e caracterização de vistas de utilização em SQL	36
5.6 . Tradução das interrogações do utilizador para SQL	37
5.7 . Indexação do Sistema de Dados	39
5.8 . Implementação de funções	41
6 . Conclusão	42

Índice de Figuras

Figura 1 - Diagrama de Gantt inicial	4
Figura 2 - Diagrama ER do Modelo Conceptual	16
Figura 3 - Modelo Lógico	20
Figura 4 - Código SQL para a criação da tabela Detetive	25
Figura 5 - Código SQL para a criação da tabela Cliente	25
Figura 6 - Código SQL para a criação da tabela Prova	26
Figura 7 - Código SQL para a criação da tabela Caso	27
Figura 8 - Código SQL para a criação da tabela Análise	27
Figura 9 - Código SQL para a criação da tabela Arma	28
Figura 10 - Código SQL para a criação da tabela Veículo	28
Figura 11 - Código SQL para a criação dos Utilizadores	29
Figura 12 - Código SQL para as permissões dos Utilizadores	29
Figura 13 - Código SQL para povoamento da tabela Detetive	30
Figura 14 - Código SQL para povoamento da tabela Cliente	30
Figura 15 - Código SQL para povoamento da tabela Caso	31
Figura 16 - Código SQL para povoamento da tabela Prova	31
Figura 17 - Código SQL para povoamento da tabela Veículo	31
Figura 18 - Código SQL para povoamento da tabela Análise	32
Figura 19 - Código SQL para povoamento da tabela Arma	32
Figura 20 - Código Python para povoamento das tabelas Cliente e Arma	33
Figura 21 - Código SQL para a vista Cliente_contactos	36
Figura 22 - Código SQL para a vista Detetives_contactos	36
Figura 23 - Código SQL para a vista CasosAtivos_detetives	36
Figura 24 - Código SQL para a vista Folha_Pagamento	37
Figura 25 - Código SQL para o procedimento casos_detetive	37
Figura 26 - Código SQL para o procedimento numerodeprovas_caso	38
Figura 27 - Código SQL para o procedimento analises_prova	38
Figura 28 - Código SQL para o procedimento casos_detetive	39
Figura 29 - Código SQL para o procedimento casos_cliente	39
Figura 30 - Código SQL para os índices	40
Figura 31 - Código SQL para a função dinheirogerado_detetive	41
Figura 32 - Diagrama de Gantt (parte 1) final	42

Índice de Tabelas

Tabela 1 - Requisitos de descrição	7
Tabela 2 - Requisitos de manipulação	8
Tabela 3 - Requisitos de Controlo	9
Tabela 4 - Requisitos de Descrição	11
Tabela 5 - Identificação e Caracterização dos Relacionamentos	12
Tabela 6 - Identificação e Caracterização dos Atributos das Entidades	16
Tabela 7 - Espaço ocupado pela Base de Dados	34

1. Definição do Sistema

1.1 Contexto de Aplicação

Era uma noite chuvosa em 2013 quando Alexandre Monsanto, um detetive da polícia, percebeu que a sua carreira não estava a levar o rumo que tinha imaginado. Ele estava sentado no seu escritório na esquadra, e ouvia a chuva cair lá fora enquanto os casos empilhados na sua secretária pareciam gritar por justiça não alcançada.

Foi nesse momento de reflexão que Alexandre decidiu que não aguentava mais, afinal de contas a razão pela qual se tinha tornado um detetive da polícia era para ajudar a que a justiça fosse feita sempre e sentia que não estava a contribuir para isso acontecer. Ele estava cansado de todas as restrições burocráticas e limitações do sistema, queria liberdade para investigar cada caso sem obstáculos, para seguir as pistas onde quer que o levassem, sem ser contido por regulamentos ou protocolos.

Naquela noite, Alexandre começou a traçar os planos para sua própria agência de detetives. Ele reconhecia que não podia fazer isso sozinho, precisava de uma equipa de detetives com a mesma vontade. Começou a contactar os seus colegas de trabalho mais confiáveis que compartilhavam a sua frustração com o sistema. Assim nasceu a "Sigma Spy".

1.2 Motivação e Objetivos do Trabalho

Com o passar dos anos, a Sigma Spy foi evoluindo e aumentando, contratando mais investigadores e especialistas numa variedade de campos, dentro da área. No entanto, à medida que os casos se tornavam mais complexos, ficou claro que a agência precisava de uma maneira mais eficiente de armazenar e aceder a informações. Foi então que Alexandre Monsanto, junto da sua equipa, começou a explorar a ideia de desenvolver uma base de dados interna.

Eles queriam criar um sistema que pudesse armazenar e organizar todas as informações relevantes sobre casos passados e presentes. Esse sistema não apenas facilitaria a partilha de informações entre os membros da equipa, mas também permitiria análises mais profundas e a identificação de padrões que poderiam ser cruciais para resolver casos complexos.

A necessidade de uma base de dados tornou-se evidente após um caso particularmente desafiador, no qual tiveram que lidar com uma rede complexa de conspiração corporativa e corrupção política. Sem um sistema para organizar e analisar as informações, a investigação ficou estagnada, levando a uma corrida contra o tempo para evitar uma tragédia iminente.

1.3 Análise da Viabilidade do processo

A implementação de uma base de dados interna na Sigma Spy é extremamente importante face à necessidade de um sistema centralizado para a gestão de informações. O aumento da equipa e a diversificação dos casos tornaram evidente a dificuldade de gerir de forma eficiente os dados usando métodos tradicionais, como arquivos físicos em papel.

Embora a implementação de uma base de dados interna tenha vários custos significativos associados, o retorno sobre o investimento será compensado a longo prazo. A melhoria na eficiência operacional, a capacidade de resolver casos mais rapidamente e a proteção aprimorada dos dados dos clientes podem resultar numa melhoria da agência, ou seja, um crescimento também a nível financeiro.

Em suma, a implementação de uma base de dados interna na Sigma Spy é altamente viável, pois visto que a dimensão da agência e a sua capacidade financeira no momento permitem um investimento deste tipo, além de ser uma necessidade, a eficiência operacional irá melhorar, além disso, o dinheiro gasto será recuperado no futuro visto que a agência vai evoluir e tornar-se capaz de resolver mais casos com uma maior rapidez.

1.4 Recursos e Equipa de Trabalho

Para garantir o sucesso do projeto de implementação da base de dados interna na Sigma Spy, uma série de recursos humanos e materiais serão essenciais.

Em termos de recursos humanos, será necessária uma equipa de desenvolvimento de software qualificada para projetar, desenvolver e implementar a base de dados interna. Esses profissionais serão responsáveis por criar e manter um sistema robusto, seguro e intuitivo que atenda às necessidades específicas da agência.

Esta equipa trabalhará em estreita colaboração com os membros da agência, principalmente com o líder Alexandre Monsanto, para entender aquilo que estes pretendem, garantindo assim o sistema mais eficiente possível.

Já em termos de recursos materiais, serão necessários um Hardware e software adequados para hospedar e executar a base de dados interna. Será utilizado o MySQL para Sistema de Gestão de Base de Dados, o que permite manipular dados de forma eficiente. Será ainda utilizado o MySQL Workbench e o brModelo como ferramentas de modelagem, ou seja, permitem criar modelos de dados e esquemas fornecendo uma base para a estrutura da Base de Dados.

Além disso, numa próxima fase, será preciso desenvolver programas para a recolha de dados e criação de painéis de análise. O Python é a escolha ideal para essa recolha de dados devido à sua flexibilidade, enquanto o Microsoft Power BI

Desktop pode ser empregado na elaboração de painéis de análise visualmente chamativos e informativos.

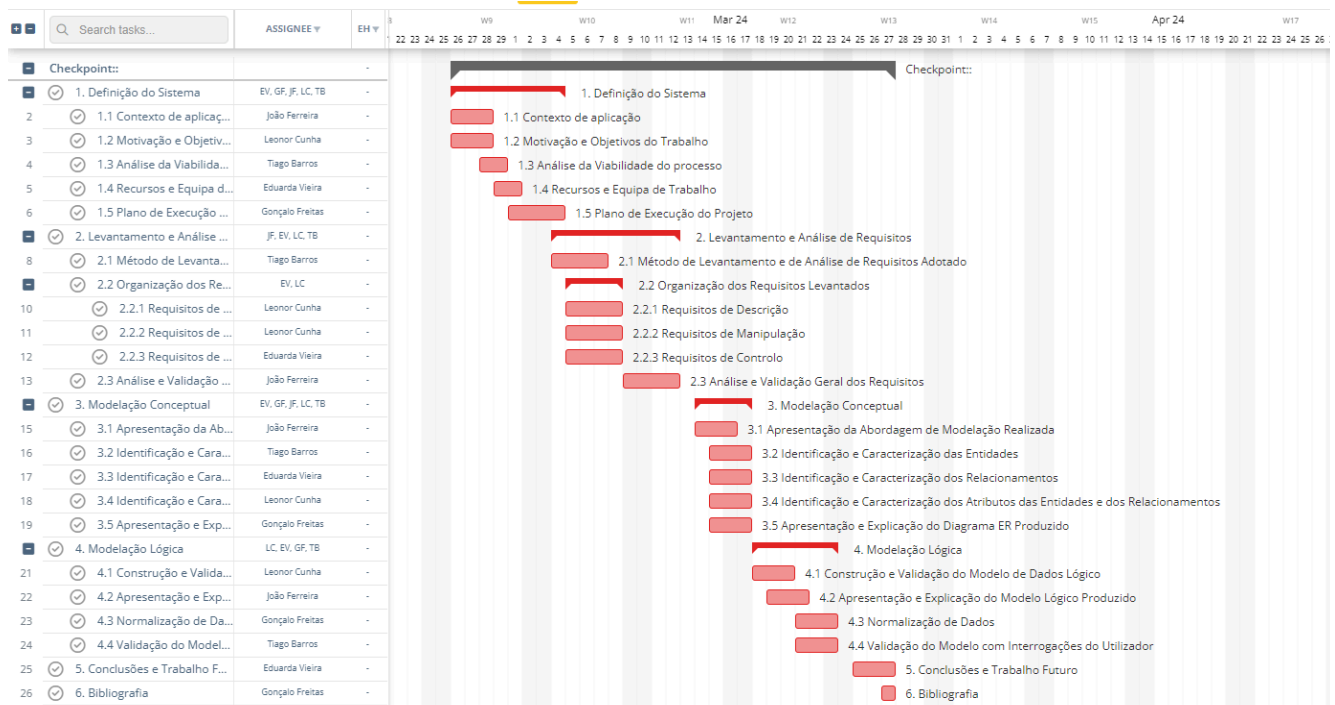


Figura 1 - Diagrama de Gantt inicial

No contexto da implementação da base de dados interna na Sigma Spy, é essencial estabelecer um plano de execução claro e abrangente. Assim sendo, Alexandre Monsanto criou o plano acima, para tentar garantir que o trabalho era cumprido de forma séria e dentro de prazo.

Inicialmente, para a Definição do Sistema, foi dado um prazo de 7 dias, pois será necessário entrar em contacto com a equipa responsável pelo desenvolvimento do sistema de Base de Dados e marcar uma reunião com os mesmos, com o objetivo de expor o que é pretendido para a agência e perceber se existem condições para levar o projeto para a frente.

Já para o Levantamento e Análise de Requisitos estabeleceu-se um limite de 8 dias, visto que serão necessárias reuniões entre Alexandre e a equipa de desenvolvimento de sistema de forma a identificar e documentar as necessidades essenciais do sistema, após isso a equipa terá de traçar o plano e reunir novamente com a agência para expor o mesmo e perceber se há pontos a corrigir.

Para a terceira fase, Modelação Conceptual, foi definido um período de 3 dias, pois será necessário analisar os requisitos recolhidos e determinar as entidades, os atributos e os relacionamentos para assim desenvolver o diagrama.

Quanto à Modelação Lógica, foi estimado um tempo de 5 dias, de forma a que a equipa consiga desenvolver um modelo que esteja em conformidade com a modelação conceptual feita anteriormente, ponderando quais chaves primárias e secundárias definir e que ligações existem entre as várias tabelas, criadas a partir das entidades e os seus atributos.

Por fim, será necessário fazer conclusões e analisar o trabalho futuro a ser feito, para esse tópico foi previsto um tempo de 3 dias tendo em conta que é preciso analisar todo o trabalho desenvolvido anteriormente e fazer uma retrospectiva do mesmo, percebendo assim o que correu bem e o que é preciso melhorar no futuro.

2. Levantamento e Análise de Requisitos

2.1 Método de Levantamento e de Análises de Requisitos Adotado

A fase de levantamento de requisitos é uma parte fundamental do desenvolvimento de um sistema de informação. É nesta fase que se compreende as prioridades e as necessidades da agência. Para garantir isto foram postas em prática várias formas de levantamento de requisitos.

Foram realizadas diversas reuniões com Alexandre Monsanto para perceber claramente os seus principais requisitos. Nestas reuniões discutiram-se questões

específicas sobre a organização da agência dos quais se podem enumerar: casos, apoio aos clientes, outros serviços e pagamentos.

A comunicação aberta e direta com Alexandre permitiu que todos os requisitos que eram esperados pelo mesmo fossem cumpridos e ajudou ainda a perceber os que detinham mais importância para Alexandre.

2.2 Organização dos Requisitos Levantados

Adotou-se as categorizações principais: descrição, manipulação e controlo. Essa organização permitiu uma compreensão clara dos requisitos, garantindo que todas as necessidades essenciais do sistema fossem identificadas e documentadas de maneira adequada.

Cada requisito foi cuidadosamente caracterizado, incluindo informações como tipo, número de identificação, data de identificação, descrição detalhada, fonte de informação e analista responsável pela sua identificação e documentação.

Por meio da colheita inicial de informações, os requisitos foram organizados e documentados de forma apropriada, ao qual proporciona uma base sólida para o desenvolvimento e implementação da base de dados, alinhada com as necessidades e expectativas da agência. Podemos, assim, proceder à definição dos requisitos que o sistema deverá suportar.

2.2.1 Requisitos de Descrição

Nº	Data/Hora	Descrição	Área/Vista	Analista	Fonte
----	-----------	-----------	------------	----------	-------

R.D1	2024/03/03 21:26:00	Todos os casos investigados devem ser registados com um número de identificação único, nome do caso, data de início da investigação, data de fim da investigação, estado do processo (resolvido, em processo, não resolvido), a categoria em que esse caso está inserido, o montante de pagamento, a forma de pagamento e o estado do pagamento	Casos	Leonor Cunha	Alexandre Monsanto
R.D2	2024/03/03 22:53:00	São considerados funcionários da agência: detetives	Detetives	João Ferreira	Alexandre Monsanto
R.D3	2024/03/03 23:05:00	Devem ser armazenados dados pessoais dos clientes, tais como nome, email, contacto telefónico, morada, NIF	Clientes	Leonor Cunha	Alexandre Monsanto
R.D4	2024/03/03 23:21:00	Detetives possuem um único id, especialidade, nome, data de nascimento, morada, telefone, email, NIF e o seu estado (ativo, inativo)	Detetives	Eduarda Vieira	Alexandre Monsanto
R.D5	2024/03/03 23:26:00	Cada caso pode possuir ou não provas relacionadas a esse caso (pode ter mais do que uma prova)	Caso	Gonçalo Freitas	Alexandre Monsanto
R.D6	2024/03/03 23:27:00	Cada prova deve possuir um único id de prova, o id do caso a sua descrição a sua data de obtenção, detetive responsável pela coleta	Provas	Gonçalo Freitas	Alexandre Monsanto
R.D7	2024/03/03 23:53:00	Um cliente/caso apenas pode ter um detetive	Cliente/Detetive	Tiago Barros	Alexandre Monsanto
R.D8	2024/03/03 23:57:00	Durante uma investigação podem ser requisitados veículos	Veículos	João Ferreira	Alexandre Monsanto
R.D9	2024/03/03 23:59:00	Cada veículo deve possuir um id único, o id do caso, o id do detetive que o requisitou, a sua data de requisito, a data de devolução, a capacidade do mesmo, quantidade de gasolina e o estado do veículo	Veículos	Leonor Cunha	Alexandre Monsanto
R.D10	2024/03/04 00:04:00	Apenas é possível a cada detetive, requisitar um só veículo	Veículos/Detetive	Eduarda Vieira	Alexandre Monsanto
R.D11	2024/03/04 16:45	Cada detetive só possui uma especialidade mas uma especialidade pode ter vários detetives	Detetive/Especialidade	Leonor Cunha	Alexandre Monsanto
R.D12	2024/03/04 17:08:00	Um detetive pode estar a trabalhar para vários clientes	Detetive/Cliente	Gonçalo Freitas	Alexandre Monsanto
R.D13	2024/03/11 14:28:00	Cada cliente deve pagar pelo seu respetivo caso	Caso/Pagamento	Gonçalo Freitas	Alexandre Monsanto
R.D14	2024/03/25 17:02:00	Os pagamentos só podem ser efetuados por multibanco, dinheiro vivo, transferência bancária ou Mb-way	Caso/Pagamento	Gonçalo Freitas	Alexandre Monsanto

R.D15	2024/03/30 22:50:00	Deve existir armas cada arma deve possuir um id único, nº de licença, data de fabrico, descrição(modelo), nº de munições	Armas	Tiago Barros	Alexandre Monsanto
R.D16	2024/03/30 23:05:00	É permitido requisitar mais que uma arma por detetive	Armas	Tiago Barros	Alexandre Monsanto
R.D17	2024/03/31 00:10:00	As análises que foram requeridas devem ser identificadas com descrição, id de prova, id de análise, data de análise, data de resultados, id de caso	Laboratório/Análises	Leonor Cunha	Alexandre Monsanto
R.D18	2024/03/31 00:15:00	Análises só podem ser requeridas pelos Detetives	Laboratório/Análises	João Ferreira	Alexandre Monsanto
R.D19	2024/03/31 00:23:00	Cada detetive pode requisitar o número de análises que quiser	Laboratório/Análises	Eduarda Vieira	Alexandre Monsanto
R.D20	2024/03/31 01:03:00	Cada folha de pagamento está associada a um caso	Pagamento	Gonçalo Freitas	Alexandre Monsanto
R.D21	2024/03/31 1:10:00	Cada caso está associado a um cliente	Caso	Gonçalo Freitas	Alexandre Monsanto
R.D22	2024/04/2 17:24:00	Para cada prova podem ser requeridas análises	Prova	Gonçalo Freitas	Alexandre Monsanto

Tabela 1 - Requisitos de descrição

2.2.2 Requisitos de Manipulação

Nº	Data/Hora	Descrição	Área/Vista	Analista	Fonte
R.M1	2024/03/03 21:48:00	Os casos só podem ser registados por um detetive	Casos	Tiago Barros	Alexandre Monsanto
R.M2	2024/03/03 21:54:00	A cada momento, deve ser possível obter o histórico de investigação/casos de cada detetive	Detetives	Gonçalo Freitas	Alexandre Monsanto
R.M3	2024/03/03 23:11:00	Deve ser possível inserir clientes, casos e detetives	Casos/Funcionários/Clientes	Gonçalo Freitas	Alexandre Monsanto
R.M4	2024/03/03 23:22:00	No final do dia deve ser possível consultar um relatório diário, onde consta o número de casos ativos e detetives ocupados com casos	Casos/Funcionários/Clientes	Gonçalo Freitas	Alexandre Monsanto

R.M5	2024/03/03 23:26	Os clientes podem registar o feedback de um caso	Clientes	Tiago Barros	Alexandre Monsanto
R.M6	2024/03/03 23:36:00	Deve ser possível aceder ao histórico de investigações de cada detetive	Casos/Detetives	Tiago Barros	Alexandre Monsanto
R.M7	2024/03/03 23:52:00	Deve ser possível consultar todos os clientes de um detetive	Cliente/Detetive	Leonor Cunha	Alexandre Monsanto
R.M8	2024/03/30 22:50:00	Durante uma investigação pode ser requisitadas armas	Armas	Eduarda Vieira	Alexandre Monsanto

Tabela 2 - Requisitos de manipulação

2.2.3 Requisitos de Controlo

Nº	Data/Hora	Descrição	Área/Vista	Analista	Fonte
R.C1	2024/03/03 23:30:00	Uma prova só pode ter um caso associado	Provas	João Ferreira	Alexandre Monsanto
R.C2	2024/03/03 23:33:00	Só os detetives devem ter a capacidade de registar notas durante as investigações	Detetives	João Ferreira	Alexandre Monsanto
R.C3	2024/03/03 23:38:00	Um cliente não pode aceder a dados sobre outros clientes ou outros casos	Clientes/Casos	Gonçalo Freitas	Alexandre Monsanto
R.C5	2024/03/30 23:03:00	Somente os detetives podem requisitar armas	Armas	Tiago Barros	Alexandre Monsanto
R.C6	2024/03/30 23:11:00	Apenas detetives podem requerer análises ao laboratório	Laboratório/Análises	Leonor Cunha	Alexandre Monsanto
R.C7	2024/03/30 23:15:00	Só os detetives podem alterar dados registados	Casos/Detetives	Gonçalo Freitas	Alexandre Monsanto

Tabela 3 - Requisitos de Controlo

2.3 Análise e Validação Geral dos Requisitos

Depois do levantamento de requisitos estar completo procedeu-se à sua validação para confirmar que ia de acordo com a visão da agência.

Para isso foram realizadas novas reuniões com Alexandre em que foi analisado requisito a requisito para confirmar que não existiam erros ou redundâncias e perceber se os requisitos eram claros e fáceis de compreender. Com esta reunião confirmamos que não se encontrava qualquer falha ou ambiguidade nos requisitos e não foi necessário realizar correções ou alterações.

Esta validação é uma boa fundação para o desenvolvimento bem sucedido do sistema de base de dados que é esperado pela agência.

3. Modelação Conceptual

3.1 Apresentação da Abordagem de Modelação Realizada

Na realização da modelagem para a agência de detetives Sigma Spy, o objetivo principal foi garantir a integridade dos dados, a eficiência na recuperação de informações e a flexibilidade para expansões e atualizações futuras. Iniciamos com um levantamento abrangente de requisitos, identificando todas as informações necessárias para organizar a base de dados da agência, incluindo casos, clientes, suspeitos, evidências e outros elementos cruciais para suas operações.

Dividimos esses requisitos em categorias como descrição, manipulação e controlo para uma melhor compreensão. Isso permitiu-nos caracterizar as entidades específicas da Sigma Spy de forma mais precisa, determinando os seus atributos essenciais e os relacionamentos entre elas. Por exemplo, identificamos os atributos dos casos, como data, localização e descrição, bem como os relacionamentos entre os casos, os clientes envolvidos e os suspeitos associados.

Após a conclusão do diagrama ER, concentramo-nos na normalização da base de dados. Isso foi crucial para garantir a facilidade na atualização de dados no futuro e para reduzir a redundância.

Em suma, essa abordagem de modelagem cuidadosa garantiu que a base de dados da Sigma Spy fosse bem estruturada, eficiente e capaz de se adaptar às necessidades da constante evolução da agência de detetives.

3.2 Identificação e Caracterização das Entidades

Para realizar o modelo conceptual foi necessário identificar e definir as várias entidades que irão existir, utilizando os requisitos levantados anteriormente. A primeira entidade que foi definida foi a entidade **Detetive**, a base da agência são os seus detetives, o que faz com que esta seja uma entidade fundamental.

A partir desta podemos definir a entidade **Caso** que é outra peça fundamental na agência que está associada aos detetives. Associado aos casos está a entidade **Prova**, que por sua vez tem **Análise** como entidade associada.

Associado aos casos está a entidade **Cliente** e **Pagamento**. Associado aos detetives estão ainda as entidades **Arma** e **Veículo**.

Requisito	Entidade	Descrição	Ocorrência
R.D1	Caso	Acontecimento que necessita de ser investigado e resolvido.	Existem vários casos que vão ter pagamento, cliente, prova e detetive associados.
R.D3	Cliente	Utilizador dos serviços da agência.	Existem clientes associados aos casos.
R.D4	Detetive	Trabalhador da agência.	Existem detetives que vão ter veículo, arma e caso associados.
R.D6	Prova	Evidência, informação ou documento.	Existem provas que estão associadas aos casos.
R.D8	Veículo	Meio de transporte.	Existem veículos que estão associados aos detetives.
R.D16	Arma	Arma portátil de fogo.	Existem armas que estão associadas

			aos detetives.
R.D17	Análise	Exame laboratorial com fins diagnósticos.	Existem análises que estão associadas às provas

Tabela 4 - Requisitos de descrição

3.3 Identificação e Caracterização dos Relacionamentos

Depois de definirmos as Entidades, 3.2, necessitamos de definir os relacionamentos utilizando os requisitos levantados anteriormente. O **Pagamento** está associado a um **Caso** e vice-versa. Cada **Caso** possui um **Cliente** e um **Cliente** pode estar relacionado a vários **Casos**. Cada **Caso** é investigado por um **Detetive** e um **Detetive** pode não estar a investigar nenhum **Caso**. Cada **Caso** tem um número n de **Provas**. Cada **Prova** pode ter um número n de **Análises** realizadas a essa **Prova**. Um detetive pode requisitar várias **Armas** e pode requisitar um só **Veículo**.

Requisito	Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
R.D21	Caso	(1,n)	Possui	(1,1)	Cliente
R.D7	Caso	(0,n)	Investigado	(1,1)	Detetive
R.D5	Caso	(1,1)	Possui	(0,n)	Prova
R.D22	Prova	(1,1)	Tem	(0,n)	Análises
R.D18	Detetive	(0,1)	Requisita	(0,n)	Arma
R.D8	Detetive	(1,1)	Requisita	(0,1)	Veículo

Tabela 5 - Identificação e Caracterização dos Relacionamentos

3.4 Identificação e Caracterização dos Atributos das Entidades e dos Relacionamentos

Após determinar quais são as diferentes entidades envolvidas num sistema e como elas se relacionam entre si, é importante identificar quais informações específicas estão associadas a cada uma dessas entidades. Essas informações são chamadas de atributos e podem ser facilmente descobertas através da análise dos requisitos do sistema.

Uma vez feita essa análise, percebe-se que os relacionamentos entre as entidades não têm informações associadas a eles, apenas as próprias entidades têm atributos. Em outras palavras, enquanto as entidades têm características ou propriedades relevantes para o sistema, os relacionamentos em si não carregam informações específicas além da própria ligação entre as entidades.

Requisito	Entidade	Atributos	Descrição	Tipo e tamanho dos dados	NULL(S/N)	Tipo de Atributo
R.D4	Detetive	ID detetive	Identificador do Detetive	INT	N	Chave primária
		Nome	Nome do detetive	VARCHAR(45)	N	Simple
		Especialidade	Qual a especialidade do detetive	VARCHAR(45)	N	Simple
		Estado	Se o detetive está ativo ou inativo num caso	VARCHAR(45)	N	Simple
		Data de Nascimento	Data de nascimento do detetive	DATETIME	N	Simple
		Morada	Morada do detetive	VARCHAR(45)	N	Simple
		Telefone	Número de telefone do detetive	VARCHAR(45)	N	Chave Candidata
		Email	E-mail do detetive	VARCHAR(45)	N	Chave Candidata
		NIF	NIF do detetive	VARCHAR(9)	N	Chave Candidata
R.D1	Caso	ID caso	Identificador do	INT	N	Chave primária

			caso			
		Estado do Processo	Se o caso já foi concluído ou se ainda está a ser investigado	BINARY(1)	N	Simple
		Data Fim Investigação	Data do fim da investigação	DATETIME	S	Simple
		Data Início Investigação	Data do início da investigação	DATETIME	N	Simple
		Nome	Nome do caso	VARCHAR(45)	N	Simple
		Categoria	Categoria em que é inserida esse caso	VARCHAR(45)	N	Simple
		Montante de Pagamento	Quantidade a ser paga	INT	N	Simple
		Forma de Pagamento	Forma de pagamento dinheiro vivo, cartão, check	VARCHAR(45)	S	Simple
		Estado do pagamento	Pago ou ainda por pagar	BINARY(1)	N	Simple

Requisito	Entidade	Atributos	Descrição	Tipo e tamanho dos dados	NULL(S/N)	Tipo de Atributo
R.D3	Cliente	NIF	Identificador do Cliente	INT	N	Chave primária
		Morada	Morada do cliente	VARCHAR(45)	N	Simple
		Contacto Telefónico	Contacto do cliente	INT	N	Chave Candidata
		Email	Email do cliente	VARCHAR(45)	N	Chave candidata
		Nome	Nome do cliente	VARCHAR(45)	N	Simple
R.D6	Prova	ID prova	Identificador de uma prova	INT	N	Chave primária
		Descrição da prova	Explicação do que é a prova	VARCHAR(45)	N	Simple
		Data de Obtenção	Data em que foi obtida a prova	DATETIME	N	Simple
R.D17	Análise	ID análise	Identificador	INT	N	Chave primária

			das análises			
		Data de análise	Data em que foi feita a análise de uma prova	DATETIME	N	Simple
		Data de resultados	Data em que saíram os resultados das análises	DATETIME	S	Simple
R.D15	Arma	ID arma	Identificador da Arma	INT	N	Chave primária
		Nº Licença	Número referente à licença da arma	INT	N	Chave Candidata
		Data de Fabrico	Data em que foi fabricada a arma	DATETIME	N	Simple
		Modelo	Modelo da arma, automática, semi automática etc.	VARCHAR(45)	N	Simple
		Nº Munições	Número de munições totais dessa arma	INT	N	Simple
R.D9	Veículo	ID veículo	Identificador do veículo	INT	N	Chave primária
		Data de Requisito	Data em que foi feito o pedido do veículo	DATETIME	N	Simple
		Data de devolução	Data em que foi devolvido o veículo	DATETIME	S	Simple
		Capacidade de veículo	Quantas pessoas consegue levar o veículo	INT	N	Simple
		Quantidade de Gasolina	Quantidade de gasolina presente no depósito	INT	N	Simple
		Estado do veículo	Se está a ser utilizado ou não	VARCHAR(45)	N	Simple

3.5 Apresentação e Explicação do Diagrama ER Produzido

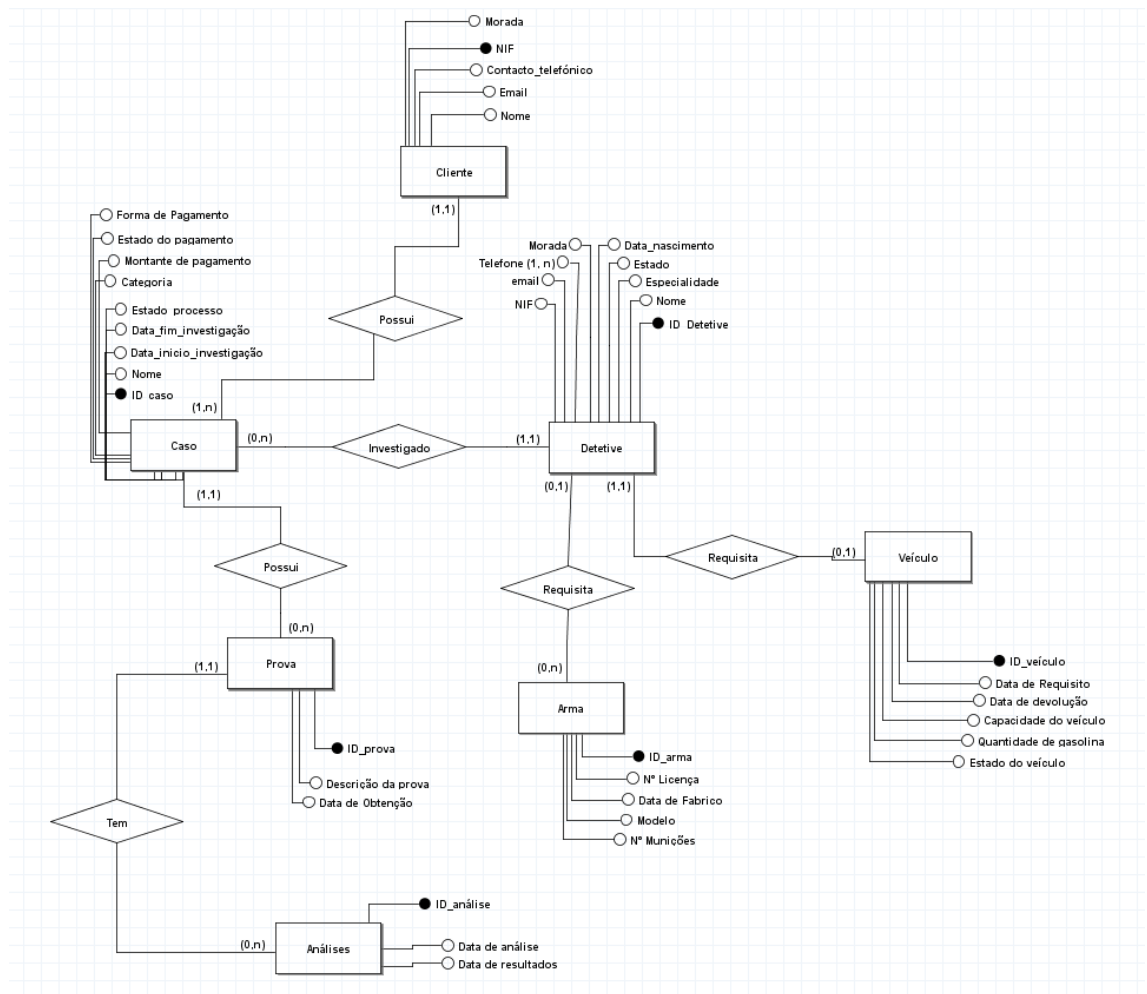


Figura 2 - Diagrama ER do modelo conceptual

Após definir e caracterizar as entidades, os relacionamentos e os atributos é possível elaborar um diagrama utilizando o software brModelo.

4. Modelação Lógica

4.1. Construção e Validação do Modelo de Dados Lógico

Para identificar e estabelecer as relações necessárias para que o modelo lógico possa representar as entidades, relacionamentos e atributos requeridos, analisamos o nosso modelo conceitual por fases, realizando a transição de uma modelagem para outra. O processo de derivação envolve a descrição de como as relações são derivadas para as estruturas que podem ocorrer em um modelo de dados conceitual. No nosso modelo, podemos ter as seguintes:

1. Entidades Fortes;
2. Entidades Fracas;

4.1.1 Entidades Fortes

Dado que uma entidade forte é aquela que possui atributos capazes de formar uma chave primária sem depender de outras entidades, chegamos à conclusão que as entidades **Caso**, **Detetives**, **Cliente**, **Arma**, **Veículo**, **Prova**, **Análises** e **Pagamento** são entidades fortes.

Caso(ID caso, Nome, Data_inicio_investigação, Data_fim_investigação, Categoria, Montante, Forma de Pagamento, Estado do Pagamento)

Detetives(ID Detetive, Nome, Especialidade, Estado, Data_Nascimento, Morada, Telefone (1,n), email, NIF).

Cliente(NIF, Morada, Contacto_telefónico, Email, Nome)

Veículo(ID_veículo, Data de Requisição, Data de devolução, Capacidade de veículo, Quantidade de gasolina, Estado do veículo)

Arma(ID_arma, Nº Licença, Data de Fabrico, Modelo, Nº Munições)

Para justificar a escolha de chaves primárias de cada entidade:

1. Nas entidades **Caso**, **Detetives**, **Arma**, **Veículo**, **Prova**, **Análises e Pagamento**, utiliza-se o id de cada um destes ("ID Caso", "ID Detetive"; "ID_Arma", "ID_veículo", "ID_prova", "ID_análise" e "id pagamento"), pois é o atributo, além de único que não se repete, mais simples e fixo destas entidades.
2. Nas entidades **Detetives**, **Cientes**, tínhamos ainda como chave candidatas os atributos "NIF", "Telefone (1,n)"/"Contacto_telefónico", "email"/"Email", poderiam ser chaves primárias, pois são dados especificamente únicos, no entanto, seria algo complexo e comprido, para localizar na base de dados (por exemplo: detectivatingEdna@SigmaSpy.com)[email] ou "xxxxxxxxx" [NIF] ou "913xxxxxx"[telefone]). Além disso, no caso do número de telemóvel/telefone pode haver alteração de operadora ao qual, o que torna este atributo não fixo e alterável.
3. Na entidade **Arma**, o "Nº Licença" é uma chave candidata pois é um atributo único, no entanto é algo complicado para procurar no sistema de dados.

Para justificar a exclusão de chaves primárias de cada entidade:

1. As entidades **Caso**, **Detetive**, **Cliente**, apresentam o atributo "Nome", ao qual era descartada a hipótese de serem chaves primárias, uma vez que, pode haver mais de um caso com o mesmo nome, como por exemplo, "Caso Dona Laura". Adicionalmente, dois detetives podem ter o mesmo nome (Detetives: João Vieira e João Vieira), assim como, dois cliente podem possuir o mesmo nome (Clientes: Maria Lopes e Maria Lopes).
2. O atributo de "Morada" que pertença a qualquer entidade (Cliente e Detetive), será recusado, pois pode haver, mesmo que baixa, a possibilidade de dois detetives e/ou clientes morarem na mesma habitação.
3. Os atributos referentes a qualquer tipo de data ("Data_Nascimento", "Data_fim_investigação", "Data_inicio_investigação", "Data de Fabrico", "Data de Requisição", "Data de devolução", "Data de análise", "Data

de resultados”), pertencente a certa(s) entidade(s), será excluído, dado que é algo que facilmente se repete, como por exemplo: O Detetive João Vieira e a Detetive Maria Alves nasceram ambos a 4 de outubro de 1970; Dois casos podem ter a mesma data de iniciação e fim.

4. Os atributos referentes a qualquer tipo de estado (“Estado de veículo”, “Estado processo”, “Estado”), não será considerado já que um estado vai ser repetido para indicar a situação de uma certa entidade, seja do veículo, do detetive, do caso. Como se vê no exemplo de: o DJV e a DMA estão ambos ativos; Dois casos podem estar terminados.
5. Os atributos relacionados a quantidades (“Nº Munições”, “Capacidade de veículo”, “Quantidade de gasolina”) serão descartados pois são números que se repetem.

4.1.2 Entidades Fracas

Diferentemente das entidades fortes, as entidades fracas não possuem atributos suficientes para formar uma chave primária própria e, portanto, dependem de outras entidades para definir sua identidade. Não temos nenhuma entidade fraca na nossa base de dados.

4.2. Apresentação e Explicação do Modelo Lógico Produzido

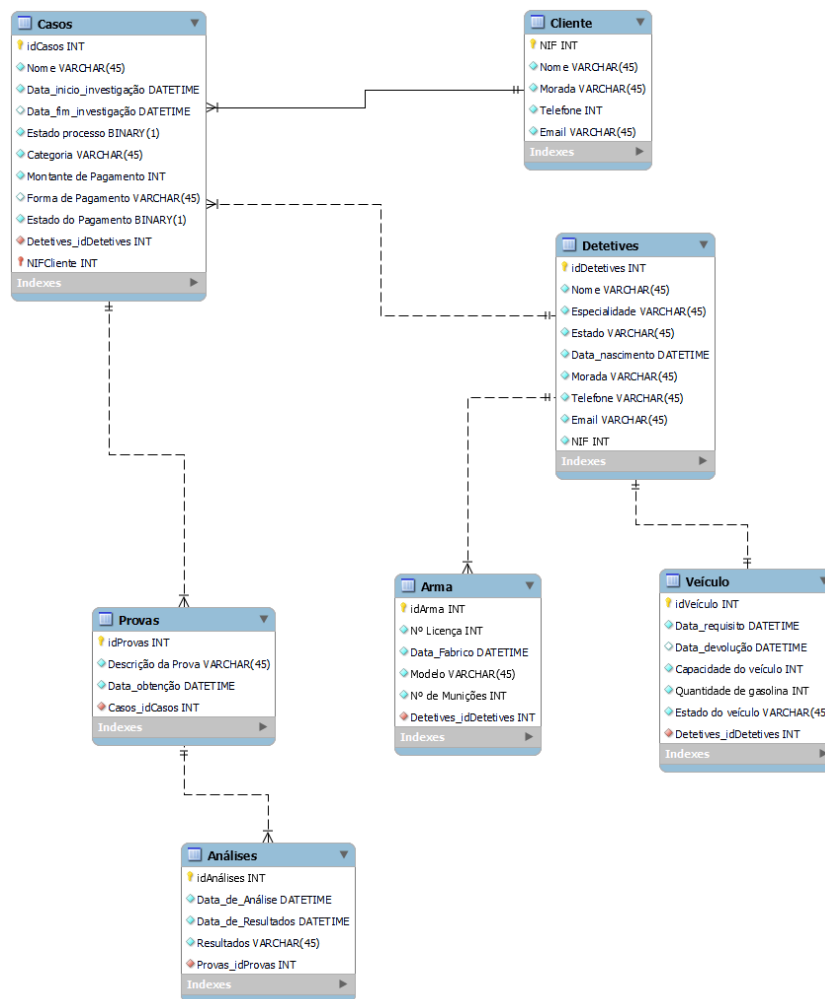


Figura 3 - Modelo Lógico

Após a derivação das várias entidades e relacionamentos para o modelo lógico, o próximo passo foi a criação das tabelas no software MySQLWorkbench.

As tabelas Pagamentos, Clientes, Detetives, Casos, Provas, Arma, Veículo e Análises representam as várias entidades.

Cada losango azul representa os atributos específicos associados a cada entidade ou relação, conforme definido no subcapítulo 3.4.

Cada losango vermelho representa uma chave estrangeira, assim como cada losango amarelo indica a chave primária.

4.3. Normalização de Dados

A normalização de um modelo relacional desempenha um papel fundamental, assegurando que as relações possuam um conjunto mínimo e essencial de atributos para atender a todos os requisitos de dados da agência. Além do mais, reduz a redundância de dados ao mínimo necessário.

Nesta fase final da modelação lógica da nossa base de dados, foi necessário rever e validar a mesma com o objetivo de garantir que toda a estrutura criada satisfaz os requisitos desejados para a mesma.

Como podemos ver nos pontos acima enunciados, através do modelo lógico apresentado fomos capazes de perceber os tipos de entidades existentes, a forma como os relacionamentos entre elas as afetam e demonstrámos ser capazes de responder às questões colocadas pelo utilizador.

Assim, após esta revisão, consideramos que o modelo lógico é válido e viável uma vez que cumpre todos os requisitos pré-estabelecidos, considerando-o assim validado.

4.4. Validação do Modelo com Interrogações do Utilizador

1. Quais os casos ativos de um detetive?
2. É possível contar o número de provas associadas a um caso ?
3. É possível saber quais as análises que estão associadas a uma prova?
4. É possível saber qual o histórico de casos de investigação de um detetive?
5. É possível saber quais os casos associados a um cliente?

Posteriormente à construção do modelo lógico e da respectiva normalização, é necessário verificar se este era capaz de responder aos requisitos do utilizador. Assim, testamos o modelo através de interrogações sobre ele:

1. Quais os casos ativos de um detetive?

É necessário aceder à tabela dos detetives e procurar o id do detetive que queremos analisar. Seguidamente, selecionamos da tabela dos casos, associados a esse detetive, e verificamos quais deles têm definido como estado ativo.

2. É possível contar o número de provas associadas a um caso?

É necessário aceder à tabela das provas e procurar pelo id de caso que pretendemos analisar. Selecionamos e contamos todas as provas que tenham esse id de caso associado.

3. É possível saber quais as análises que estão associadas a uma prova?

Começamos por ir à tabela das análises e pesquisamos em quais dessas tem associado um id_prova igual ao pretendido. De seguida, filtramos os resultados e apresentamos as análises que obtivemos.

4. É possível saber qual o histórico de casos de investigação de um detetive?

A partir das tabelas dos casos, filtramos a mesma pelo id do detetive, e apresentamos todos os casos que estejam associados ao respetivo detetive, independentemente do estado do caso.

5. É possível saber quais os casos associados a um cliente?

A partir da tabela de casos pesquisamos pelo NIF do cliente, e verificamos os casos que pretendemos associados a esse.

```
pi idCasos (sigma NIF = 309875622 (Cliente)) join Casos;
```

5. Implementação física

Para finalizar a construção do sistema de base de dados, passamos para a fase de implementação física, seguida pelo povoamento da base de dados e pela exploração e monitoramento de todas as informações armazenadas. Primeiramente, convertemos o modelo lógico, apresentado no capítulo 4, para o SGBD (Sistema de Gestão de Bases de Dados) escolhido. Após a implementação e o povoamento, definimos em SQL algumas ações para explorar e monitorar os dados armazenados.

Neste capítulo, além de detalhar os passos já mencionados, também estimamos o espaço em disco que será ocupado pela nossa base de dados.

5.1 Apresentação e explicação da base de dados implementada

Para traduzir o esquema lógico para o MySQL, utilizamos a ferramenta MySQL Workbench. Com essa ferramenta, convertemos o modelo lógico, que desenvolvemos anteriormente, em código SQL.

Começamos por criar as tabelas, de acordo com o modelo lógico, tendo em conta todas as chaves primárias e estrangeiras definidas no mesmo.

- Tabela Detetive

O detetive possui chave primária *idDetetive* que se incrementa automaticamente e o NIF é um valor único (UNIQUE INDEX)

```
CREATE TABLE IF NOT EXISTS Detetive (  
    idDetetive INT NOT NULL AUTO_INCREMENT,  
    Nome VARCHAR(45) NOT NULL,  
    Especialidade VARCHAR(45) NOT NULL,  
    Estado VARCHAR(45) NOT NULL,  
    Data_nascimento DATETIME NOT NULL,  
    Morada VARCHAR(45) NOT NULL,  
    Telefone VARCHAR(45) NOT NULL,  
    Email VARCHAR(45) NOT NULL,  
    NIF INT NOT NULL,  
    PRIMARY KEY (idDetetive),  
    UNIQUE INDEX NIF_UNIQUE (NIF ASC) VISIBLE)
```

Figura 4 - Código SQL para a criação da tabela Detetive

- Tabela Cliente

O cliente possui a chave primária NIF

```
CREATE TABLE IF NOT EXISTS Cliente (  
    NIF INT NOT NULL,  
    Nome VARCHAR(45) NOT NULL,  
    Morada VARCHAR(45) NOT NULL,  
    Telefone INT NOT NULL,  
    Email VARCHAR(45) NOT NULL,  
    PRIMARY KEY (NIF))
```

Figura 5 - Código SQL para criação da tabela Cliente

- Tabela Prova

A tabela prova possui a sua chave primária *idProva* que se incrementa automaticamente e possui uma chave estrangeira chamada *Id_Caso* que está associada ao caso de onde essa prova está associada

```
CREATE TABLE IF NOT EXISTS Prova (  
    idProva INT NOT NULL AUTO_INCREMENT,  
    Descrição_da_Prova VARCHAR(45) NOT NULL,  
    Data_obtenção DATETIME NOT NULL,  
    Id_Caso INT NOT NULL,  
    PRIMARY KEY (idProva),  
    FOREIGN KEY (Id_Caso)  
    REFERENCES Caso (idCaso))
```

Figura 6 - Código SQL para criação da tabela Prova

- Tabela Caso

A tabela Caso possui uma chave primária *idCaso* que se incrementa automaticamente e a chave primária estrangeira *NIF_Cliente* que está associado ao cliente a que pertence este caso. Além disso, também possui uma chave estrangeira *Id_Detetive* que está associada ao detetive que está a tomar conta deste caso.

```

CREATE TABLE IF NOT EXISTS Caso (
    idCaso INT NOT NULL AUTO_INCREMENT,
    Nome VARCHAR(45) NOT NULL,
    Data_inicio_investigação DATETIME NOT NULL,
    Data_fim_investigação DATETIME NULL,
    Estado_processo binary(1) NOT NULL,
    Categoria VARCHAR(45) NOT NULL,
    Montante_de_Pagamento INT NOT NULL,
    Forma_de_Pagamento VARCHAR(45) NULL,
    Estado_do_Pagamento BINARY(1) NOT NULL,
    Id_Detetive INT NOT NULL,
    NIF_Cliente INT NOT NULL,
    PRIMARY KEY (idCaso, NIF_Cliente),
    FOREIGN KEY (Id_Detetive)
    REFERENCES Detetive (idDetetive),
    FOREIGN KEY (NIF_Cliente)
    REFERENCES Cliente (NIF))

```

Figura 7 - Código SQL para criação da tabela Caso

- Tabela Análise

A tabela Análise possui um *idAnálise* que é a sua chave Primária e é incrementada automaticamente, possui também uma chave estrangeira *Id_Prova* que diz a que prova é que a análise está a ser feita.

```

CREATE TABLE IF NOT EXISTS Análise (
    idAnálise INT NOT NULL AUTO_INCREMENT,
    Resultados VARCHAR(45) NOT NULL,
    Data_de_Análise DATETIME NOT NULL,
    Data_de_Resultados DATETIME NOT NULL,
    Id_Prova INT NOT NULL,
    PRIMARY KEY (idAnálise),
    FOREIGN KEY (Id_Prova)
    REFERENCES Prova (idProva))

```

Figura 8 - Código SQL para criação da tabela Análise

- Tabela Arma

A tabela Arma possui uma chave primária *idArma* que é incrementada automaticamente e uma chave estrangeira *Id_Detetive* que está associado ao detetive que possui a arma.

```
CREATE TABLE IF NOT EXISTS Arma (  
    idArma INT NOT NULL AUTO_INCREMENT,  
    N_Licença INT NOT NULL,  
    Data_Fabrico DATETIME NOT NULL,  
    Modelo VARCHAR(45) NOT NULL,  
    N_de_Munições INT NOT NULL,  
    Id_Detetive INT NOT NULL,  
    PRIMARY KEY (idArma),  
    FOREIGN KEY (Id_Detetive)  
    REFERENCES Detetive (idDetetive))
```

Figura 9 - Código SQL para criação da tabela Arma

- Tabela Veículo

A Tabela Veículo possui uma chave primária *idVeículo* que se incrementa automaticamente e possui uma chave estrangeira *Id_Detetive* que está associado ao detetive que requisitou esse veículo.

```
CREATE TABLE IF NOT EXISTS Veículo (  
    idVeículo INT NOT NULL AUTO_INCREMENT,  
    Data_requisito DATETIME NOT NULL,  
    Data_devolução DATETIME NULL,  
    Capacidade_do_veículo INT NOT NULL,  
    Quantidade_de_gasolina INT NOT NULL,  
    Estado_do_veículo VARCHAR(45) NOT NULL,  
    Id_Detetive INT NOT NULL,  
    PRIMARY KEY (idVeículo),  
    FOREIGN KEY (Id_Detetive)  
    REFERENCES Detetive (idDetetive))
```

Figura 10 - Código SQL para criação da tabela Veículo

5.2 . Criação de utilizadores da base de dados

Nesta fase iremos criar quais os utilizadores que vão aceder à nossa base de dados e respectivas permissões. Neste caso criamos um utilizador chamado 'detetive' e um chamado 'Cliente'.

```
-- Criação do utilizador 'detetive'
CREATE USER 'detetive'@'localhost';

-- Definição de password do detetive
SET PASSWORD FOR 'detetive'@'localhost' = 'detetive';

-- Criação do utilizador 'cliente'
CREATE USER 'cliente'@'localhost';

-- Definição de password de cliente
SET PASSWORD FOR 'cliente'@'localhost' = 'cliente';
```

Figura 11 - Código SQL para criação dos utilizadores

O 'Detetive' poderá aceder e modificar a base de dados da forma que quiser, o 'Cliente' só poderá visualizar o Nome, Especialidade, Telefone e Email de um detetive.

```
-- Permissões
-- Cliente só pode ver o nome dos detetives a sua especialidade e o sei email

GRANT SELECT (Nome, Especialidade, Telefone, Email)
ON Tricornio.Detetive TO 'cliente'@'localhost';

GRANT ALL ON tricornio.* TO 'detetive'@'localhost';
-- SELECT * FROM mysql.user WHERE User = 'detetive';
```

Figura 12 - Código SQL para as permissões dos utilizadores

5.3 . Povoamento da base de dados

Para o povoamento da nossa base de dados decidimos criar 20 Detetives, 17 Clientes, 17 Casos, 10 Armas, 15 Veículos, 45 Provas e 14 Análises.

Para isso utilizamos o comando INSERT e tivemos o cuidado de dar os valores das chaves estrangeiras consoante as chaves primárias existentes, a seguir iremos mostrar excertos do código SQL utilizado.

- Povoamento Tabela Detetive

```
INSERT INTO Detetive
(Nome, Especialidade, Estado, Data_nascimento, Morada, Telefone, Email, NIF)
VALUES
('João Silva', 'Investigação Criminal', 'Ativo', '1985-04-23', 'Rua das Flores, 123, Lisboa', '912345678', 'joao.silva@gmail.com', '123456789'),
('Maria Oliveira', 'Perícia Forense', 'Ativo', '1978-07-12', 'Avenida da Liberdade, 456, Porto', '919876543', 'maria.oliveira@gmail.com', '987654321'),
('Pedro Santos', 'Cybersegurança', 'Ativo', '1990-09-15', 'Travessa do Mar, 789, Faro', '915678234', 'pedro.santos@gmail.com', '456789123'),
('Ana Ferreira', 'Inteligência e Contraespionagem', 'Inativo', '1982-11-30', 'Rua do Sol, 321, Coimbra', '917654321', 'ana.ferreira@gmail.com', '321654987'),
('Luís Costa', 'Análise de Dados', 'Ativo', '1988-01-05', 'Largo da Praça, 654, Braga', '918273645', 'luis.costa@gmail.com', '654123789'),
('Rita Nunes', 'Investigação Financeira', 'Ativo', '1975-03-22', 'Rua do Mercado, 12, Lisboa', '912345679', 'rita.nunes@gmail.com', '223456789'),
('Carlos Martins', 'Investigação de Fraudes', 'Ativo', '1981-08-17', 'Avenida dos Aliados, 34, Porto', '919876544', 'carlos.martins@gmail.com', '887654321'),
('Sofia Carvalho', 'Investigação de Homicídios', 'Ativo', '1987-12-29', 'Rua das Estrelas, 56, Faro', '915678235', 'sofia.carvalho@gmail.com', '556789123'),
```

Figura 13 - Código SQL para o povoamento da tabela Detetive

- Povoamento Tabela Cliente

```
INSERT INTO Cliente
(NIF, Nome, Morada, Telefone, Email)
VALUES
('111222333', 'Afonso Matos', 'Rua das Flores, 123, Lisboa', '912345678', 'afonso.matos@gmail.com'),
('222333444', 'Beatriz Gonçalves', 'Avenida da Liberdade, 456, Porto', '919876543', 'beatriz.goncalves@gmail.com'),
('333444555', 'Carlos Silva', 'Travessa do Mar, 789, Faro', '915678234', 'carlos.silva@gmail.com'),
```

Figura 14 - Código SQL para o povoamento da tabela Cliente

- Povoamento Tabela Caso

```
INSERT INTO Caso
(Nome, Data_inicio_investigação, Data_fim_investigação, Estado_processo, Categoria, Montante_de_Pagamento,
Forma_de_Pagamento, Estado_do_Pagamento, Id_Detetive, NIF_Cliente)
VALUES
('Caso 1', '2024-01-01', '2024-03-01', '1', 'Investigação Criminal', 1500.00, 'Cartão de Crédito', '1', 1, '111222333'),
('Caso 2', '2024-02-15', '2024-04-15', '1', 'Perícia Forense', 2000.00, 'Transferência Bancária', '1', 2, '222333444'),
('Caso 3', '2024-03-01', '2024-05-01', '0', 'Cybersegurança', 2500.00, 'Dinheiro', 0, 3, '333444555'),
('Caso 4', '2024-04-10', '2024-06-10', '0', 'Inteligência e Contraespionagem', 3000.00, 'Cartão de Crédito', '0', 4, '444555666'),
('Caso 5', '2024-05-05', '2024-07-05', '1', 'Análise de Dados', 1800.00, 'Transferência Bancária', '1', 5, '555666777'),
```

Figura 15 - Código SQL para o povoamento da tabela Caso

- Povoamento Tabela Prova

```
INSERT INTO Prova
(Descrição_da_Prova, Data_obtenção, Id_Caso)
VALUES
('Fibra Encontrada', '2024-01-20 14:45:00', 1),
('Entrevista com Testemunha', '2024-01-25 11:30:00', 1),
('Análise de Documentos', '2024-01-30 09:20:00', 1),
('Análise de Vídeo', '2024-02-20 10:00:00', 2),
```

Figura 16 - Código SQL para o povoamento da tabela Prova

- Povoamento da Tabela Veículo

```
INSERT INTO Veículo
(Data_requisito, Data_devolução, Capacidade_do_veículo, Quantidade_de_gasolina,
Estado_do_veículo, Id_Detetive)
VALUES
('2024-01-10 09:00:00', '2024-01-20 17:00:00', 5, 50, 'Bom', 1),
('2024-02-05 08:30:00', '2024-02-15 16:30:00', 4, 60, 'Excelente', 2),
('2024-03-12 10:00:00', '2024-03-22 18:00:00', 7, 45, 'Regular', 3),
```

Figura 17 - Código SQL para o povoamento da tabela Veículo

- Povoamento da Tabela Análise

```
INSERT INTO Análise
(Resultados, Data_de_Análise, Data_de_Resultados, Id_Prova)
VALUES
('Análise concluída com sucesso.', '2024-01-20 09:00:00', '2024-01-25 14:00:00', 1),
('Resultados não conclusivos.', '2024-01-25 10:00:00', '2024-01-30 15:00:00', 2),
('Sem resultados significativos.', '2024-01-30 11:00:00', '2024-02-05 16:00:00', 3),
```

Figura 18 - Código SQL para o povoamento da tabela Análise

- Povoamento da Tabela Arma

```
INSERT INTO Arma
(N_Licença, Data_Fabrico, Modelo, N_de_Munições, Id_Detetive)
VALUES
(123456, '2023-01-15 10:00:00', 'Glock 19', 15, 1),
(234567, '2022-05-20 14:30:00', 'Beretta 92FS', 10, 2),
```

Figura 19 - Código SQL para o povoamento da tabela Arma

O outro método utilizado foi um programa em python que efetua o povoamento da tabela Cliente e da tabela Arma. Este programa conecta-se à base de dados e insere os dados especificados no código abaixo.


```

import mysql.connector

# dados do servidor sql
mydb = mysql.connector.connect(
    host = "localhost",
    user = "root",
    password = "palavrapasse",
    database = "Tricornio"
)

mycursor = mydb.cursor()

# Dados que se pretende inserir
sql = "INSERT INTO Cliente (NIF, Nome, Morada, Telefone, Email) VALUES (%s, %s, %s, %s, %s)"
val = [
    ("179825878", "António Silva", "Rua Eusébio, 66, Lisboa", "934457689", "otamendinobolso@benfica.pt"),
    ("273634538", "Ricardo Quaresma", "Rua da Trivela, 7, Porto", "968033543", "triveladourada7@gmail.com"),
    ("360123767", "João Capital", "Rua da Fragata, 1, Guimarães", "934407689", "joacapital123@hotmail.com"),
    ("156789123", "Maria Zita", "Rua da Chuva, 12, Faro", "934457689", "asviagensdamaria@gmail.com"),
    ("176309821", "José Saramago", "Rua do Livro, 55, Lisboa", "914957789", "ensaiosobreacegueira@saramago.com")
]

sql2 = "INSERT INTO Arma (idArma, N_licença, Data_Fabrico, Modelo, N_de_Munições, Id_Detetive) VALUES (%s, %s, %s, %s, %s, %s)"
val2 = [
    ("11", "345654", "2021-03-05 16:02:47", "Glock 22", "17", "11"),
    ("12", "123679", "2001-09-11 08:46:00", "Dandelion 21", "12", "12"),
    ("13", "456789", "2010-05-12 12:45:00", "M1911", "7", "13"),
    ("14", "108726", "2015-07-23 14:23:00", "Glock 17", "14", "12"),
    ("15", "987654", "2019-11-30 18:00:00", "S1935", "15", "10")
]

mycursor.executemany(sql, val)
mycursor.executemany(sql2, val2)
mydb.commit()

print(mycursor.rowcount, "detalhes inseridos")

# desconectar do servidor
mydb.close()

```

Figura 20 - Código python para o povoamento das tabelas Cliente e Arma

No código de python, começamos por indicar o nome da base de dados ao qual se conectar “Tricórnio” (no *database*=“Tricornio”). Seguidamente cria-se um objeto cursor, que será usado para executar comandos SQL na base de dados (*mycursor* = *mydb.cursor()*).

Após esta introdução, preparamos a instrução SQL (*sql* = “*Insert INTO ...*”) e os valores a serem inseridos numa nova linha da tabela Cliente. (*val* = [(...)]). Imediatamente, repete-se o mesmo raciocínio para a tabela Arma.

Finalmente, a *executemany()* executa as instruções SQL (*sql* e *sql2*) para cada conjunto de valores fornecido na lista (*val* e *val2*). E com *mydb.commit()* confirma-se as alterações guardando-as na base de dados.

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

Seguidamente, iremos apresentar uma estimativa do espaço em disco que a nossa base de dados irá ocupar. Para termos uma estimativa mais segura, e de maneira a precaver qualquer cenário possível, os atributos VARCHAR terão o tamanho máximo declarado.

Entidade	Atributo	Data Type	Tamanho	Total
Detetive	idDetetive	INT	4 Bytes	286 Bytes
	Nome	VARCHAR(45)	45 Bytes	
	Especialidade	VARCHAR(45)	45 Bytes	
	Estado	VARCHAR(45)	45 Bytes	
	Data_nascimento	DATETIME	8 Bytes	
	Morada	VARCHAR(45)	45 Bytes	
	Telefone	VARCHAR(45)	45 Bytes	
	Email	VARCHAR(45)	45 Bytes	
	NIF	INT	4 Bytes	
Cliente	NIF	INT	4 Bytes	143 Bytes
	Nome	VARCHAR(45)	45 Bytes	
	Morada	VARCHAR(45)	45 Bytes	
	Telefone	INT	4 Bytes	
	Email	VARCHAR(45)	45 Bytes	
Caso	idCaso	INT	4 Bytes	169 Bytes
	Nome	VARCHAR(45)	45 Bytes	
	Data_inicio_investigação	DATETIME	8 Bytes	
	Data_fim_investigação	DATETIME	8 Bytes	
	Estado_processo	BINARY(1)	1 Bytes	
	Categoria	VARCHAR(45)	45 Bytes	
	Montante_Pagamento	INT	4 Bytes	
	Forma_de_Pagamento	VARCHAR(45)	45 Bytes	
	Estado_do_pagamento	BINARY(1)	1 Bytes	
	Id_Detetive	INT	4 Bytes	
	NIF_Cliente	INT	4 Bytes	

Arma	idArma	INT	4 Bytes	69 Bytes
	N_licenca	INT	4 Bytes	
	Data_Fabrico	DATETIME	8 Bytes	
	Modelo	VARCHAR(45)	45 Bytes	
	N_de_munições	INT	4 Bytes	
	Id_Detetive	INT	4 Bytes	
Veículo	idVeículo	INT	4 Bytes	77 Bytes
	Data_requisito	DATETIME	8 Bytes	
	Data_devolução	DATETIME	8 Bytes	
	Capacidade_do_veículo	INT	4 Bytes	
	Quantidade_de_galosina	INT	4 Bytes	
	Estado_do_veículo	VARCHAR(45)	45 Bytes	
	Id_Detetive	INT	4 Bytes	
Prova	idProva	INT	4 Bytes	61 Bytes
	Descrição_da_Prova	VARCHAR(45)	45 Bytes	
	Data_obtenção	DATETIME	8 Bytes	
	Id_Caso	INT	4 Bytes	
Análise	idAnálise	INT	4 Bytes	69 Bytes
	Resultados	VARCHAR(45)	45 Bytes	
	Data_de_Análise	DATETIME	8 Bytes	
	Data_de_Resultados	DATETIME	8 Bytes	
	Id_Prova	INT	4 Bytes	

Tabela 7 - Espaço ocupado pela Base de Dados

Após somarmos os valores ocupados por todas as tabelas o tamanho da nossa base de dados seria de **874 Bytes**, isto seria para uma estimativa se considerássemos que cada tabela só possui 1 atributo de cada tipo.

Considerando a nossa povoação onde temos 20 Detetives, 17 Clientes, 17 Casos, 10 Armas, 15 Veículos, 45 Provas e 14 Análises isso daria um total de: $20 \times 286 + 17 \times 143 + 17 \times 169 + 10 \times 69 + 15 \times 77 + 45 \times 61 + 14 \times 69 = \mathbf{16580 \text{ Bytes}}$

Se considerarmos que o número de detetives só irá aumentar até aos 50 e que mensalmente existem 17 clientes e 17 casos, e com 10 provas no total em que cada

prova possui uma análise e que o número de armas e veículos não varia isso significa que anualmente teremos um total de $50 \times 296 + 17 \times 12 \times 143 + 17 \times 12 \times 169 + 10 \times 69 + 15 \times 77 + 10 \times 12 \times 61 + 10 \times 12 \times 69 = \mathbf{95893 \text{ Bytes/ANO}}$.

5.5 . Definição e caracterização de vistas de utilização em SQL

- View que dá o contacto dos clientes(Telefone e Email)

```
CREATE VIEW Clientes_contactos
AS
SELECT Nome AS "Nome", Telefone AS "Telefone", Email AS "Email"
FROM Cliente;
--
```

Figura 21 - Código SQL para a vista Clientes_contactos

- View que dá o contacto dos detetives(Telefone e Email)

```
CREATE VIEW Detetives_contactos
AS
SELECT Nome AS "Nome", Telefone AS "Telefone", Email AS "Email"
FROM Detetive;
```

Figura 22 - Código SQL para a vista Detetives_contactos

- View que dá os casos ativos de um detetive

```
CREATE VIEW CasosAtivos_detetives
As
SELECT Caso.Nome AS "Nome do Caso", Detetive.Nome AS "Nome do Detetive"
FROM Caso INNER JOIN Detetive
ON caso.Id_Detetive = detetive.idDetetive
where caso.Estado_processo = 1 ;
```

Figura 23 - Código SQL para a vista CasosAtivos_detetives

- View que dá a folha de pagamento de um cliente possui o nome do cliente o montante de pagamento a forma de pagamento e o seu estado

```
CREATE VIEW Folha_Pagamento
As
SELECT Cliente.Nome AS "Nome", Montante_de_Pagamento AS "Montante", Forma_de_Pagamento AS "Forma de Pagamento", Estado_do_Pagamento AS "Estado"
FROM Cliente INNER JOIN Caso
ON Cliente.NIF = Caso.NIF_Cliente;
```

Figura 24 - Código SQL para a vista Folha_Pagamento

5.6 . Tradução das interrogações do utilizador para SQL

Nesta secção, serão apresentados exemplos de código SQL que permitem obter as respostas às interrogações ao utilizador colocadas no ponto 4.4.

1. Quais os casos ativos de um detetive?

```
Create procedure casos_detetives(IN id_detetive INT)
Begin
  Select *
  From Caso INNER JOIN Detetive
  ON Caso.Id_Detetive = Detetive.idDetetive
  where Caso.Estado_processo = 1 AND Detetive.idDetetive = id_detetive;
END $$
```

Figura 25 - Código SQL para o procedimento casos_detetives

2. É possível contar o número de provas associada a um caso?

```
DELIMITER $$  
Create procedure numerodeprovas_caso(IN id_caso INT)  
Begin  
    Select count(*)  
    From Prova INNER JOIN Caso  
    ON Prova.Id_Caso = Caso.idCaso  
    where caso.idCaso = id_caso;  
END $$
```

Figura 26 - Código SQL para o procedimento numerodeprovas_caso

3. É possível saber quais as análises que estão associadas a uma prova?

```
DELIMITER $$  
Create procedure analises_prova(IN id_prova INT)  
Begin  
    Select *  
    From Análise  
    Where Análise.Id_Prova = id_prova;  
END $$
```

Figura 27 - Código SQL para o procedimento analises_prova

4. É possível saber qual o histórico de casos de investigação de um detetive?

```
DELIMITER $$  
Create procedure casos_detetive(IN id_detetive INT)  
Begin  
    Select *  
    From Caso  
    Where Caso.Id_Detetive = id_detetive;  
END $$
```

Figura 28 - Código SQL para o procedimento casos_detetive

5. É possível saber quais os casos associados a um cliente?

```
DELIMITER $$  
Create procedure casos_cliente(IN id_cliente INT)  
Begin  
    Select *  
    From Caso  
    Where Caso.NIF_Cliente = id_cliente;  
END $$
```

Figura 29 - Código SQL para o procedimento casos_cliente

5.7 . Indexação do Sistema de Dados

De forma a minimizar o tempo de resposta às queries feitas pelo cliente, utilizamos índices, que são estruturas de dados que nos irão permitir aceder a um ficheiro/arquivo com maior facilidade e rapidez.

Como as tabelas que são mais acedidas e procuradas nas queries são as tabelas Detetive, Caso e Cliente, criamos os seguintes índices:

```
-- Índice para consulta "Detetives" dá o id do detetive
CREATE INDEX idx_detetive ON Detetive (idDetetive);

-- Índice para consulta "Caso"
CREATE INDEX idx_caso ON Caso (idCaso);

-- Índice para consulta "Cliente"
CREATE INDEX idx_cliente ON Cliente (NIF);
```

Figura 30 - Código SQL para os índices

5.8 . Implementação de funções

De forma a conseguirmos saber o rendimento que cada detetive produziu para a agência, criamos uma função que recebe um número, o id do detetive, e vai somar o valor pago em todos os casos que esse detetive trabalhou ou trabalha ainda atualmente. Assim, com esta informação, conseguimos, mais facilmente, perceber o quão importante cada detetive tem sido para a empresa.


```

DELIMITER $$
CREATE FUNCTION dinheirogerado_detetive(num INT)
RETURNS INT
DETERMINISTIC
BEGIN
    RETURN(SELECT SUM(Caso.Montante_de_Pagamento) AS 'Dinheiro Gerado' FROM Caso INNER JOIN
    Detetive ON Caso.Id_Detetive = Detetive.idDetetive
    WHERE Detetive.idDetetive = num);

END $$
DELIMITER ;

```

Figura 31 - Código SQL para a função dinheirogerado_detetive

6 . Conclusão

Em suma, o trabalho foi realizado de forma bastante homogênea pelos elementos do grupo, tendo sido bem distribuído e havido bastante entreaajuda.

Acreditamos que a realização e organização dos requisitos, assim como a atribuição de atributos às entidades, foi um dos pontos fortes do trabalho como também a modelação conceptual. Além disso, a realização e organização da implementação física, nomeadamente a criação das tabelas, os povoamentos e a criação de utilizadores da base de dados, foram dos pontos fortes do trabalho.

Por outro lado, a realização do relatório e a implementação de funções foi onde sentimos mais dificuldades.

Concluimos que, com a segunda parte, conseguimos desenvolver uma boa base de dados, uma vez que incluímos consultas SQL eficientes que minimizem o tempo de execução. Aliás usamos índices apropriados para acelerar consultas e operações de busca e definimos permissões e roles adequados para diferentes usuários.

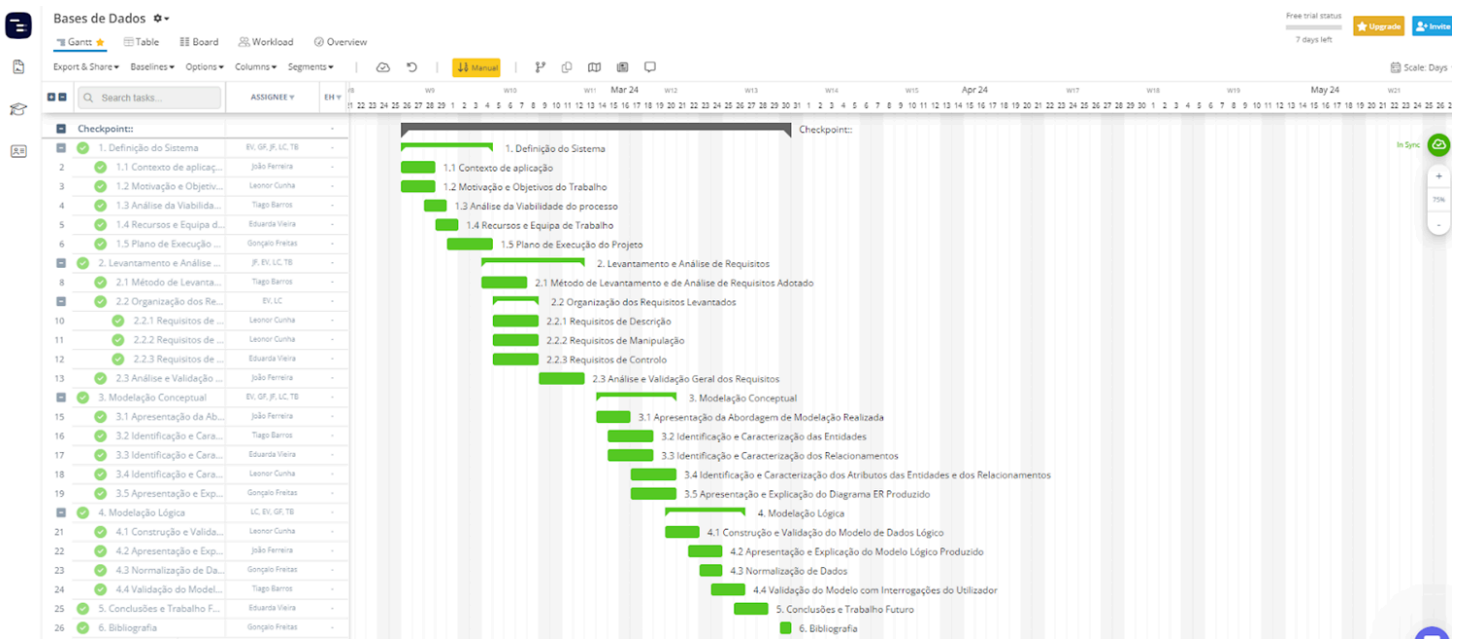


Figura 4 - Diagrama de Gantt (parte 1) final

Referências

Connolly, T., Begg, C., Database Systems: A Practical Approach to Design, Implementation, and Management, Addison- Wesley, Global Edition, 26 Sep 2014. ISBN-10: 1292061189, ISBN-13: 978-1292061184

Belo, O. -

https://elearning.uminho.pt/bbcswebdav/pid-1428103-dt-content-rid7870421_1/xid-7870421_1 ("O Ciclo de Vida do Desenvolvimento de uma Base de Dados");

Belo, O. -

https://elearning.uminho.pt/bbcswebdav/pid-1428103-dt-content-rid7870424_1/xid-7870424_1 ("A Mercearia da D. Acácia");

Belo, O. -

https://elearning.uminho.pt/bbcswebdav/pid-1428103-dt-content-rid8376411_1/xid-8376411_1 ("Normalização de Dados");

Ferreira, D. -

https://elearning.uminho.pt/bbcswebdav/pid-1441475-dt-content-rid8134933_1/xid-8134933_1 ("PL04 - Modelação Conceptual");

Ferreira, D. -

https://elearning.uminho.pt/bbcswebdav/pid-1446084-dt-content-rid8184990_1/xid-8184990_1 ("PL05 - Modelação Lógica")

Ferreira, D. -

https://elearning.uminho.pt/bbcswebdav/pid-1455084-dt-content-rid8273227_1/xid-8273227_1 ("PL06 - Modelação Lógica e Normalização")

Ferreira, D. -

https://elearning.uminho.pt/bbcswebdav/pid-1457450-dt-content-rid8367912_1/xid-8367912_1 ("PL07 – Normalização e Álgebra Relacional")

Lista de Siglas e Acrónimos

BD	Base de Dados
DW	Data Warehouse
OLTP	<i>On-Line Analytical Processing</i>
ER	<i>Entidade-Relacionamento</i>
DJV	<i>Detetive João Vieira</i>
DMA	<i>Detetive Maria Alves</i>