



Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Base de Dados

Ano Letivo de 2023/2024

Brigada de Espionagem e Localização Operacional

Eduardo Faria
a104353

Hélder Gomes
a104100

Pedro Pereira
a104082

Nuno Silva
a104089

28 de maio de 2024

BD

Data da Receção	
Responsável	
Avaliação	
Observações	

**Brigada de Espionagem e
Localização Operacional**

Eduardo Faria a104353	Hélder Gomes a104100	Pedro Pereira a104082	Nuno Silva a104089
---------------------------------	--------------------------------	---------------------------------	------------------------------

28 de maio de 2024

Resumo

Ao longo deste projeto foi desenvolvido um sistema de gestão de base de dados, direcionado para a empresa *Lusium*, com o intuito de melhorar a segurança em Vizela. Este sistema seria aplicado especificamente na gestão dos roubos relacionados com minérios *edium* e *nunium*, onde seriam avaliadas informações sobre os terrenos e os seus trabalhadores, bem como os casos ativos.

Este projeto engloba várias etapas que, de forma sucinta, podem ser divididas em definição do sistema, levantamento e análise dos requisitos, modelação concetual, modelação lógica e implementação física. Estas etapas, e a sua posterior validação, tornam-se essenciais para a construção de um sistema de gestão de base de dados bem estruturado.

Após uma ponderação sobre as melhores ferramentas a utilizar, optamos pelo uso do “BrModelo” para a modelação concetual, do “MySQL Workbench” para a modelação lógica e implementação física e do “Relax” para abordar a componente de álgebra linear, dada a sua eficácia e capacidade de lidar com desafios específicos de um sistema de gestão de bases.

Área de Aplicação: Análise, planeamento, modelação, arquitetura e implementação de um sistema de gestão de base de dados.

Palavras-Chave: Sistema de gestão de base de dados, *Lusium*, Casos, Requisitos, Modelação Concetual, Modelação Lógica, Implementação Física.

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	6
2.1. Método de Levantamento e de Análise de Requisitos Adotado	6
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	8
2.3. Análise e Validação Geral dos Requisitos	9
3. Modelação Concetual	10
3.1. Apresentação da Abordagem de Modelação Realizada	10
3.2. Identificação e Caracterização das Entidades	10
3.3. Identificação e Caracterização dos Relacionamentos	11
3.4. Identificação e Caracterização dos Atributos das Entidades e dos Relacionamentos	13
3.5. Apresentação e Explicação do Diagrama ER Produzido	15
4. Modelação Lógica	17
4.1. Construção e Validação do Modelo de Dados Lógico	17
4.2. Apresentação e Explicação do Modelo Lógico Produzido	17
4.3. Normalização de Dados	20
4.4. Validação do Modelo com Interrogações do Utilizador	21
5. Implementação Física	25
5.1. Apresentação e explicação da base de dados implementada	25
5.2. Criação de utilizadores da base de dados	28
5.3. Povoamento da base de dados	29
5.4. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)	35
5.5. Definição e caracterização de vistas de utilização em SQL	38
5.6. Tradução das interrogações do utilizador para SQL	39
5.7. Indexação do Sistema de Dados	42
5.8. Implementação de procedimentos, funções e gatilhos	43
6. Conclusões e Trabalho Futuro	46
Referências	48
Lista de Siglas e Acrónimos	49
Anexos	50

Anexo 1: Ata da primeira reunião	50
Anexo 2: Ata da segunda reunião	51
Anexo 3: Ata da terceira reunião	52
Anexo 4: Ata da quarta reunião	54
Anexo 5: Logo da Universidade do Minho	56
Anexo 6: Ilustração do diagrama de Gantt inicial.	56
Anexo 7: Ilustração do Relacionamento Funcionário - Funcionário.	56
Anexo 8: Ilustração do Relacionamento Funcionário - Função.	57
Anexo 9: Ilustração do Relacionamento Funcionário - Terreno.	57
Anexo 10: Ilustração do Relacionamento Funcionário - Caso.	57
Anexo 11: Ilustração do Relacionamento Terreno - Caso.	57
Anexo 12: Ilustração do diagrama ER produzido.	58
Anexo 13: Ilustração do modelo lógico produzido.	58
Anexo 14: Ilustração do diagrama de Gantt real.	58
Anexo 15: Ilustração do Inquérito Estatístico de Opinião.	59
Anexo 16: Tabela lógica da Função.	59
Anexo 17: Tabela lógica do Funcionário.	59
Anexo 18: Tabela lógica do Número de Telemóvel.	59
Anexo 19: Tabela lógica do Gere.	59
Anexo 20: Tabela lógica do Terreno.	60
Anexo 21: Tabela lógica do Trabalha.	60
Anexo 22: Tabela lógica do Caso.	60
Anexo 23: Tabela lógica do Suspeito.	60
Anexo 24: Script de criação da base de dados em SQL.	60
Anexo 25: Script de povoamento da base de dados em SQL.	62
Anexo 26: Script de povoamento da base de dados em Pthon.	63
Anexo 27: Script de procedimentos, funções e gatilhos da base de dados em SQL.	65
Anexo 28: Script de índices em SQL.	67
Anexo 29: Script de vistas da base de dados em SQL.	67
Anexo 30: Script de interrogações da base de dados em SQL.	68

Lista de Figuras

Figura 1: Ilustração do diagrama de Gantt inicial.	4
Figura 2: Ilustração do Relacionamento Funcionário - Funcionário.	11
Figura 3: Ilustração do Relacionamento Funcionário - Função.	12
Figura 4: Ilustração do Relacionamento Funcionário - Terreno.	12
Figura 5: Ilustração do Relacionamento Funcionário - Caso.	13
Figura 6: Ilustração do Relacionamento Terreno - Caso.	13
Figura 7: Ilustração do diagrama ER produzido.	16
Figura 8: Ilustração do modelo lógico produzido.	20
Figura 9: Ilustração do primeiro diagrama de Álgebra Relacional.	21
Figura 10: Ilustração do segundo diagrama de Álgebra Relacional.	22
Figura 11: Ilustração do terceiro diagrama de Álgebra Relacional.	22
Figura 12: Ilustração do quarto diagrama de Álgebra Relacional.	23
Figura 13: Ilustração do quinto diagrama de Álgebra Relacional.	23
Figura 14: Ilustração do sexto diagrama de Álgebra Relacional.	24
Figura 15: Ilustração do sétimo diagrama de Álgebra Relacional.	24
Figura 16: Instruções SQL para criar e utilizar a base de dados.	25
Figura 17: Tabela lógica da Função.	25
Figura 18: Implementação correspondente à Função.	25
Figura 19: Tabela lógica do Funcionário.	26
Figura 20: Implementação correspondente ao Funcionário.	26
Figura 21: Tabela lógica do Número de Telemóvel.	26
Figura 22: Implementação correspondente ao Número de Telemóvel.	26
Figura 23: Tabela lógica do Gere.	26
Figura 24: Implementação correspondente ao Gere.	26
Figura 25: Tabela lógica do Terreno.	27
Figura 26: Implementação correspondente ao Terreno.	27
Figura 27: Tabela lógica do Trabalha.	27
Figura 28: Implementação correspondente ao Trabalha.	27
Figura 29: Tabela lógica do Caso.	27
Figura 30: Implementação correspondente ao Caso.	27
Figura 31: Tabela lógica do Suspeito.	28
Figura 32: Implementação correspondente ao Suspeito.	28
Figura 33: Criação de utilizadores “detetive” e devidas permissões.	28
Figura 34: Criação de utilizadores “representantes” e devidas permissões.	29
Figura 35: Inserção de dados na tabela Função com SQL.	29
Figura 36: Inserção de dados na tabela Funcionário com SQL.	30
Figura 37: Inserção de dados na tabela Número_de_tele móvel com SQL.	30
Figura 38: Inserção de dados na tabela Gere com SQL.	30
Figura 39: Inserção de dados na tabela Terreno com SQL.	30
Figura 40: Inserção de dados na tabela Trabalha com SQL.	31
Figura 41: Inserção de dados na tabela Caso com SQL.	31
Figura 42: Inserção de dados na tabela Suspeito com SQL.	31
Figura 43: Importação de bibliotecas e conexão à base de dados.	32
Figura 44: Inserção de dados na tabela Função com Python.	32
Figura 45: Inserção de dados na tabela Funcionário com Python.	33

Figura 46: Inserção de dados na tabela Número_de_telefone com Python.	33
Figura 47: Inserção de dados na tabela Gere com Python.	33
Figura 48: Inserção de dados na tabela Terreno com Python.	33
Figura 49: Inserção de dados na tabela Trabalha com Python.	34
Figura 50: Inserção de dados na tabela Caso com Python.	34
Figura 51: Inserção de dados na tabela Suspeito com Python.	34
Figura 52: Finalização das transações e conexão com a base de dados.	34
Figura 53: Vista que demonstra todos os funcionários e os seus casos associados.	38
Figura 54: Vista que demonstra todos os funcionários e os seus casos associados.	39
Figura 55: Exemplo de utilização da vista “FuncionariosEmTerrenos”.	39
Figura 56: Declaração dos IDs para utilização nas consultas.	39
Figura 57: Listar o prejuízo de um terreno com SQL.	40
Figura 58: Ver quando é que um funcionário se tornou suspeito de um determinado caso com SQL.	40
Figura 59: Listar os suspeitos de um determinado caso com SQL.	40
Figura 60: Ver a data do último caso de um determinado funcionário com SQL.	41
Figura 61: Listar os casos a que um determinado funcionário está associado com SQL.	41
Figura 62: Ver o dia em que mais casos foram abertos com SQL.	41
Figura 63: Listar os top 5 funcionários por quantidade de casos com SQL.	42
Figura 64: Aplicação de indexação na base de dados.	42
Figura 65: Função que calcula a estimativa de roubo de um caso.	43
Figura 66: Função que calcula a idade através de uma data de nascimento.	43
Figura 67: Gatilho responsável por atualizar a data de encerramento de um caso.	44
Figura 68: Procedimento que automatiza a atribuição de suspeitos a um novo caso.	45
Figura 69: Ilustração do diagrama de Gantt real da primeira fase.	46
Figura 70: Ilustração do diagrama de Gantt real da segunda fase.	46

Lista de Tabelas

Tabela 1: Requisitos de descrição.	8
Tabela 2: Requisitos de manipulação.	8
Tabela 3: Requisitos de controlo.	9
Tabela 4: Caracterização das entidades.	10
Tabela 5: Caracterização dos relacionamentos.	11
Tabela 6: Caracterização dos atributos da entidade Funcionário.	14
Tabela 7: Caracterização dos atributos da entidade Função.	14
Tabela 8: Caracterização dos atributos da entidade Terreno.	14
Tabela 9: Caracterização dos atributos da entidade Caso.	15
Tabela 10: Caracterização dos atributos do relacionamento Funcionário - Caso.	15
Tabela 11: Caracterização da tabela lógica do Funcionário.	18
Tabela 12: Caracterização da tabela lógica do Número de telemóvel.	18
Tabela 13: Caracterização da tabela lógica da Função.	18
Tabela 14: Caracterização da tabela lógica de Gere.	18
Tabela 15: Caracterização da tabela lógica do Terreno.	18
Tabela 16: Caracterização da tabela lógica do Caso.	19
Tabela 17: Caracterização da tabela lógica de Trabalha.	19
Tabela 18: Caracterização da tabela lógica do Suspeito.	19
Tabela 19: Tipos de dados utilizados e o seu respetivo tamanho em bytes.	35
Tabela 20: Espaço ocupado pela tabela Funcionário com um registo.	35
Tabela 21: Espaço ocupado pela tabela Número de Telemóvel com um registo.	36
Tabela 22: Espaço ocupado pela tabela Função com um registo.	36
Tabela 23: Espaço ocupado pela tabela Gere com um registo.	36
Tabela 24: Espaço ocupado pela tabela Terreno com um registo.	36
Tabela 25: Espaço ocupado pela tabela Caso com um registo.	36
Tabela 26: Espaço ocupado pela tabela Trabalha com um registo.	37
Tabela 27: Espaço ocupado pela tabela Suspeito com um registo.	37
Tabela 28: Espaço total ocupado pelas tabelas com o povoamento inicial.	37
Tabela 29: Variações do espaço ocupado pelas tabelas após um ano.	38

1. Definição do Sistema

1.1. Contexto de aplicação

Num passado não muito distante, Vizela mantinha-se como uma cidade pequena e remota no norte de Portugal. A vida da população seguia o seu curso tranquilamente e, devido à natureza pacífica da comunidade, a intervenção judicial era mínima. Os crimes esporádicos, que pontuavam a rotina da cidade, eram tratados com eficácia pela modesta “Casa do Xerife”, comandada pelo xerife Vítor Hugo Salgado, indicando uma realidade onde a simplicidade prevalecia na resolução de questões criminais.

Nesse cenário, a pequena “Casa do Xerife” desempenhava um papel vital no controlo do fluxo criminal da cidade. Localizada numa rua modesta, era o ponto central para a administração da justiça local. Os crimes eram resolvidos com relativa rapidez, muitas vezes através de julgamentos informais que refletiam os valores e tradições da comunidade.

Porém, o curso tranquilo da vida em Vizela estava prestes a ser alterado de maneira irreversível. A descoberta dos minerais raros, *nunium* e *edium*, transformou essa cidade remota num centro de atenção internacional. O que tornava essa descoberta ainda mais extraordinária era a sua capacidade única. A combinação desses minerais resultava na formação de diamantes, um tesouro valioso que despertou a cobiça e o interesse de investidores ao redor do globo.

O influxo económico resultante dessa descoberta não mudou apenas a paisagem de Vizela, mas também trouxe consigo a criação da maior empresa de mineração da Península Ibérica, *Lusium*. Os seus terrenos antes tranquilos e pouco explorados eram agora cobiçados locais de riqueza, gerando uma corrida frenética pela extração desses minérios preciosos.

A prosperidade económica não se limitou apenas à indústria de mineração. Inúmeros comerciantes de joalharias, atraídos pela matéria-prima única disponível em Vizela, estabeleceram-se pela cidade. O brilho dos diamantes criados pela combinação de *nunium* e *edium* adornava as vitrines das lojas, transformando Vizela num destino renomeado para os amantes de jóias exclusivas.

Contudo, nem tudo eram rosas na cidade agora florescente. Com a crescente demanda por *nunium* e *edium*, os roubos aos terrenos repletos de riquezas naturais, administrados pela *Lusium*, aumentaram significativamente. O xerife Vítor Hugo, que anteriormente conseguia controlar eficazmente os crimes esporádicos, viu-se agora sobrecarregado diante dos desafios impostos pela nova realidade.

Os casos de roubo e furto tornaram-se mais complexos e frequentes, ultrapassando a capacidade da antiga “Casa do Xerife”. A cidade remota de Vizela, outrora conhecida pela sua tranquilidade, agora enfrentava dilemas de segurança e ordem pública, demandando uma abordagem mais estruturada e sofisticada para lidar com os casos crescentes de criminalidade.

1.2. Motivação e Objetivos do Trabalho

Atormentada pelo crescimento exponencial da criminalidade, a pequena cidade necessitava agora, mais do que nunca, de um novo regime de segurança e organização que permitisse restabelecer a ordem local. Nesse contexto, os líderes comunitários, liderados pelo xerife Vítor Hugo, tomaram uma decisão crucial:

estabelecer um acordo com a empresa *Lusium*. Em troca de terrenos ricos em minérios preciosos e benefícios fiscais, comprometeram-se a reduzir drasticamente o roubo de *nunium* e *edium*.

Com a nova missão em mente, os representantes da *Lusium*, Ana Sofia, Maria Eduarda e João Guilherme, fundaram o departamento “Brigada de Espionagem e Localização Operacional” para enfrentar a árdua tarefa de combater o roubo de minérios.

À medida que a situação se desenrolava, surgiu a oportunidade de buscar aconselhamento junto da “Quatro em Linha”, uma empresa jovem composta por engenheiros de software da Universidade do Minho, especializada em desenvolver projetos íntegros e escaláveis de armazenamento, controlo e manipulação de informação.

Desta forma, contratualizou-se a aquisição dos serviços de prospeção e aplicação de um modelo de base de dados personalizado e simplificador da questão-problema inicial, tendo como objetivos:

- Controlar o fluxo criminal na cidade de Vizela, proporcionando uma melhoria na qualidade de vida dos residentes;
- Organizar o modelo de concretização da entidade de segurança, bem como melhorar a sua capacidade de gestão e registo de furtos, terrenos e funcionários que neles trabalham;
- Coordenar recursos materiais e humanos;
- Supervisionar, de forma eficiente, os recursos da empresa de minérios *Lusium*, tendo em conta os terrenos adquiridos e as joalharias, assim como os lucros das mesmas;
- Permitir a análise dos dados coletados, de modo a identificar tendências e padrões nos furtos, bem como traçar um perfil dos hábitos dos suspeitos;
- Melhorar a segurança de dados, ajudando a empresa de minérios a proteger informação privada de ameaças externas;
- Desenvolvimento de um modelo operacional estruturado, visando aprimorar a eficiência na resposta a furtos e crimes nos terrenos de mineração.
- Técnicas de análise de padrões para identificação de áreas de risco.

1.3. Análise da Viabilidade do processo

Antes de se decidir implementar um sistema de gestão de base de dados, é necessário fazer uma análise da sua viabilidade, de modo a perceber as vantagens que esta implementação traria para a *Lusium*.

A implementação deste sistema traria à agência um maior controlo e conhecimento sobre as relações entre casos, terrenos e funcionários que neles trabalham, o que levaria tanto a uma redução significativa dos frequentes furtos de *edium* e *nunium*, como a um aumento da segurança na cidade e dos lucros da empresa de mineração.

Parâmetros relativos a terrenos, elementos identificadores de um funcionário e ferramentas de trabalho de um detetive, seriam avaliados no contexto do problema em questão. Deste modo, seria promovida uma maior facilidade na gestão dos casos ativos e agilidade na resolução dos mesmos, proporcionada pela utilização do sistema de gestão de base de dados, sendo por isso prevista uma diminuição dos furtos dos minérios em 45%.

Por outro lado, é esperado um aumento das receitas da *Lusium* devido à redução prevista dos furtos de *edium* e *nunium*. As quantidades roubadas desses minérios têm contribuído para uma perda de cerca de 16% do rendimento anual da empresa. Entretanto, com a diminuição destes incidentes, espera-se recuperar parte, ou até mesmo a totalidade desse prejuízo, o que traria, para além de possíveis lucros, segurança e estabilidade financeira à mesma.

Para além disso, esta redução teria um efeito direto no reforço da segurança da cidade de, aproximadamente, 23%, pois é previsto que o escoamento deste tipo de crimes aliviaria a “Casa do Xerife”, permitindo redirecionar, novamente, o seu foco para os casos de pequena escala.

É certo que a base de dados traria grandes benefícios para a cidade e para a empresa e, tendo em conta tudo aquilo que foi referido, acreditamos que a implementação deste modelo é definitivamente viável e trará resultados significativos. Apesar disso, a agência terá de assumir os custos e desafios que estão implícitos na implementação do sistema, até porque haverão determinados recursos extra que serão mencionados no tópico posterior.

Devido à confirmação da sua viabilidade, damos como previsão de entrega do sistema de gestão de base de dados o dia 24 de maio de 2024. Para tornar esta data possível, mobilizamos os recursos da equipa da “Quatro em Linha” de forma a otimizar o período de tempo pré-estabelecido pela Lusium e executar o projeto proposto.

1.4. Recursos e Equipa de Trabalho

Para esta fase inicial do projeto, a equipa “Quatro em Linha” precisou de estudar o método de atuação da “Casa do Xerife”, para perceber o modelo de aplicação e aprofundar o conhecimento na área. Neste seguimento, organizou-se um plano de execução para tornar o objetivo possível, o qual depende tanto de recursos humanos como materiais.

A nível de recursos humanos, a empresa *Lusium* é orientada por três representantes, uma equipa especializada de espionagem pertencente ao departamento “Brigada de Espionagem e Localização Operacional” e contém mais de cem funcionários que conhecem e lidam no quotidiano com o funcionamento da mesma. Para concretizar o significado dos dados recorreu-se a diversas reuniões com os representantes e a inquéritos gerais sobre a experiência individual dos funcionários, que incidem no problema do elevado fluxo criminal.

Quanto a recursos materiais, a equipa “Quatro em Linha” adotou uma abordagem prática, utilizando papel e caneta para inquéritos e atas de reuniões, posteriormente transcritos para formato digital. Essa escolha simples, mas eficaz, permite a obtenção de dados detalhados sobre a experiência dos funcionários e a gestão do elevado fluxo criminal.

A equipa de trabalho é composta pelos membros da “Quatro em Linha”, cada um trazendo uma especialidade valiosa para o desenvolvimento do sistema de gestão de base de dados da Lusium. Eduardo Faria, especialista em análise de requisitos, lidera a identificação precisa das necessidades do cliente. Hélder Gomes, com vasta experiência em modelagem de sistemas, encarrega-se da criação de estruturas eficientes para a base de dados. Nuno Silva, especializado em implementação de sistemas, traduzirá as especificações em código funcional. Por fim, Pedro Pereira, especialista em otimização de desempenho, garantirá que o sistema opere de maneira eficaz e otimizada. Juntos, esta equipa diversificada assegura uma abordagem abrangente e bem-sucedida para o projeto em questão.

Além disso, os três representantes responsáveis pela empresa Lusium, Ana Sofia, Maria Eduarda e João Guilherme, desempenham um papel crucial na implementação do sistema de gestão da base de dados. Para além das suas funções habituais, atuam como porta-vozes dos funcionários, sendo fundamentais para uma comunicação eficaz. A sua participação ativa em reuniões estratégicas e sessões de discussão permite que as preocupações e sugestões dos funcionários sejam adequadamente transmitidas e consideradas no processo de implementação do sistema. Essa abordagem colaborativa reforça a coesão e a eficácia do projeto, garantindo que as necessidades e perspetivas de todos os membros da empresa sejam levadas em conta.

No âmbito desta equipa dedicada, é imprescindível destacar a significativa contribuição dos detetives do Departamento “Brigada de Espionagem e Localização Operacional”. Estes profissionais desempenham um papel vital na análise e resolução dos casos relacionados com o elevado fluxo criminal. A colaboração entre os detetives e os especialistas em tecnologia da “Quatro em Linha” fortalece a abordagem integral do projeto, assegurando uma implementação eficaz do sistema de gestão da base de dados da Lusium.

Por último, é imperativo destacar que a importância de todos os funcionários na implementação do sistema de gestão da base de dados não pode ser subestimada. Cada membro da equipa da *Lusium* desempenha um papel vital na colheita de dados e na adaptação às mudanças. Reconhecendo a valiosa participação de

cada indivíduo, o projeto ganha força através da diversidade de experiências e perspectivas, reforçando a união e promovendo um ambiente de trabalho mais colaborativo e eficiente.

1.5. Plano de Execução do Projeto

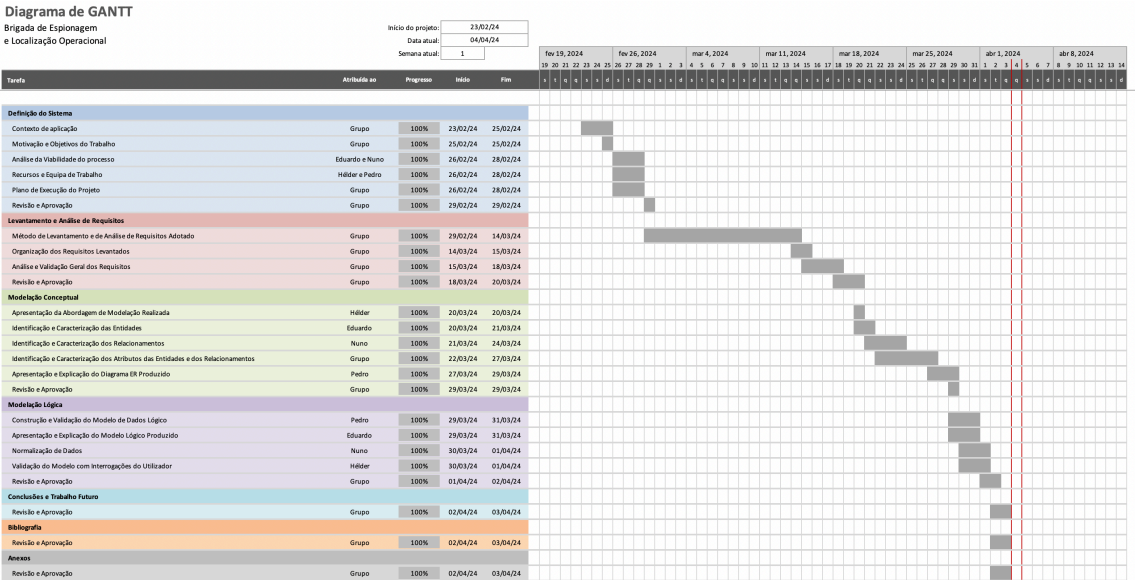


Figura 1: Ilustração do diagrama de Gantt inicial.

No decorrer da confirmação do projeto, a equipa da “Quatro em Linha” desenvolveu um diagrama de GANTT que permitisse, de forma pertinente e ponderada, distribuir todas as tarefas vitais para o desenvolvimento de um sistema de gestão de base de dados numa linha temporal otimizada e concreta.

Com o principal objetivo de entregar uma primeira fase íntegra do projeto no prazo previsto, optou-se pela aplicação de uma **estratégia monovista** de execução, de acordo com o qual separamos o projeto em cinco fragmentos fundamentais: Definição do Sistema, Levantamento e Análise de Requisitos, Modelação Concetual, Modelação Lógica e Conclusões e Trabalho Futuro.

Nesta perspetiva, ponderamos o dia três de abril como o objetivo do culminar de toda a primeira fase, o que permite à “Quatro em Linha” garantir uma margem de revisão e reavaliação dos conteúdos, se necessário, ao mesmo tempo que é demonstrado um trabalho atempado e eficaz.

Iniciaram-se os trabalhos dia vinte e três de fevereiro pela **Definição do Sistema**, este que é um ponto de trabalho com vista ao estudo do cliente e aprofundamento do conhecimento na área de desenvolvimento, dentro do qual se destacam tópicos cruciais evidenciados no diagrama proposto: Contexto de aplicação, Motivação e Objetivos do Trabalho, Análise da Viabilidade do processo, Recursos e Equipa de Trabalho e o Plano de Execução do Projeto. A previsão da divisão do tempo neste período introdutório foi muito equitativa, pelo que, ao longo de uma semana, distribuímos as diferentes tarefas, com prioridade à realização de pesquisa e construção associada em grupo, salvaguardando que todos os envolvidos adquiram as habilidades e conhecimentos indispensáveis.

De seguida, durante o planeamento, a “Quatro em Linha” deparou-se com a etapa mais preponderante do desenvolvimento, o **Levantamento de Requisitos**. Neste ponto, atribuímos vinte e um dias, dado a relevância do tópico, de forma a garantir bom sustento para as tarefas subsequentes e coerência do projeto. Prevemos o maior intervalo de tempo para a escolha e aplicação do Método de Levantamento e de Análise de Requisitos Adotado, onde optaremos por introduzir todos os membros da empresa de forma a diversificar as abordagens de recolha de informação, entre deslocações para reuniões presenciais, análise de validade de formulários estatísticos, escritas e aprovação de atas e outros. Dentro deste período destacaremos 7 dias para corporificar a informação obtida, este processo será dividido em três fases: Organização dos Requisitos Levantados, Análise e Validação Geral dos Requisitos e Revisão e Aprovação, respetivamente,

permitindo a estruturação dos conteúdos, posterior abonação do trabalho conseguido e consequente corroboração de resultados desta etapa.

Para a **Modelação Concetual**, pretendemos reservar 10 dias e uma abordagem diferente da tomada até ao momento. Dado a boa identificação e definição dos requisitos, é possível neste passo realizar uma distribuição equitativa de tarefas por todos os elementos da “Quatro em Linha”. Através da atribuição equilibrada de cargas de trabalho, ficará responsável o Hélder Gomes pela Apresentação da Abordagem de Modelação Realizada, Eduardo Faria pela Identificação e Caracterização das Entidades e Nuno Silva pela Identificação e Caracterização dos Relacionamentos. Excecionalmente, o grupo reunirá para Identificação e Caracterização dos Atributos das Entidades e dos Relacionamentos dada a influência e impacto decisivo deste tópico na atuação da equipa. De forma a concluir esta etapa, Pedro Pereira irá elaborar a Apresentação e Explicação do Diagrama ER Produzido, de forma a simplificar e concretizar a exposição do diagrama final efetuado.

Para a **Modelação Lógica**, procedeu-se à separação da equipa para otimizar os processos. Durante cinco dias, Pedro Pereira estará encarregue da Construção e Validação do Modelo de Dados Lógico, Eduardo Faria consubstanciará a Apresentação e Explicação do Modelo Lógico Produzido e, em simultâneo, Nuno Silva procederá à Normalização de Dados de forma a uniformizar e garantir a visualização intuitiva da informação existente. Posteriormente, Hélder Gomes prosseguirá com a Validação do Modelo com Interrogações do Utilizador. Finalmente, será realizada uma reunião de equipa para a revisão da etapa e confirmação geral da edificação deste tópico.

2. Levantamento e Análise de Requisitos

Na sequência do decorrer das etapas do desenvolvimento, surge a necessidade de um levantamento, organização e efetivação concreta dos requisitos que sustentarão a totalidade do sistema de gestão de base de dados.

2.1. Método de Levantamento e de Análise de Requisitos Adotado

De forma a garantir um eficiente desenvolvimento do sistema de gestão de base de dados para a *Lusium*, a equipa “Quatro em Linha” dedicou-se a registar e documentar todas as condições e determinações da empresa em relação à melhor forma de organizar todo o modelo que envolve o problema que é enfrentado. Para tal, esta recorreu aos seguintes métodos de levantamentos de requisitos:

- **Reuniões:** Foram feitas reuniões com os detetives da B.E.L.O e representantes da *Lusium* de forma a entender qual a melhor forma de gerir os dados relacionados com os funcionários e terrenos onde trabalham, bem como os dados relacionados com os casos e os suspeitos que neles se integram;
- **Inquérito:** Foi executado um questionário denominado “Inquérito Estatístico de Opinião”, orientado aos colaboradores da *Lusium*, com a principal meta de aprofundar o conhecimento e a transversalidade dos recursos humanos da “Quatro em Linha” de acordo com o contexto do caso em estudo.

O registo destes procedimentos é imprescindível para assegurar a consistência dos requisitos recolhidos a partir da intervenção dos diversos trabalhadores da *Lusium*, estando a documentação destes procedimentos visível no [anexo 1](#), [anexo 2](#), [anexo 3](#), [anexo 4](#) e [anexo 15](#). Numa fase final do tratamento dos requisitos, a estruturação dos mesmos foi apresentada ao representantes e detetives da *Lusium*, de forma a obter a sua validação.

2.2. Organização dos Requisitos Levantados

Após a definição do método de levantamento e análise de requisitos, e consequente aplicação do mesmo, procede-se à disposição tabelada e organizada de acordo com a informação explícita nos anexos ao presente relatório.

2.2.1. Requisitos de descrição

Tipo	Nº	Data	Descrição do requisito	Fonte	Analista
Descrição	1	04/03/2024	Um funcionário é identificado pelo seu identificador único e sequencial, nome, data de nascimento, salário, número de identificação fiscal, fotografia, se aplicável, e números de telemóvel.	Representantes Maria Eduarda e João Guilherme	Eduardo Faria e Nuno Silva
	2	04/03/2024	Um funcionário tem 1 ou mais números de telemóvel.	Representante Maria Eduarda	Eduardo Faria e Nuno Silva
	3	04/03/2024	Uma função é identificada pelo seu identificador único e designação, que pode tomar os valores de Operacional, Detetive e Representante.	Representante Ana Sofia	Eduardo Faria e Nuno Silva
	4	04/03/2024	Um funcionário desempenha uma única função e uma função é desempenhada por 1 ou mais funcionários.	Representante Ana Sofia	Eduardo Faria e Nuno Silva
	5	04/03/2024	Um funcionário Operacional trabalha em 1 ou mais terrenos e num terreno trabalham 1 ou mais funcionários.	Representante Ana Sofia	Eduardo Faria e Nuno Silva
	6	04/03/2024	Um terreno é identificado pelo seu identificador único e sequencial, minério previsto e minério coletado.	Representantes da Lusium	Eduardo Faria e Nuno Silva
	7	04/03/2024	O minério previsto corresponde à quantidade mínima estimada de minério a coletar por dia.	Representantes da Lusium	Eduardo Faria e Nuno Silva
	8	04/03/2024	O minério coletado corresponde à quantidade de minério efetivamente coletado por dia.	Representantes da Lusium	Eduardo Faria e Nuno Silva
	9	04/03/2024	Um funcionário que desempenha a função de Representante gere os funcionários que desempenham a função de Operacional.	Representantes da Lusium	Eduardo Faria e Nuno Silva
	10	08/03/2024	Um caso é identificado pelo seu identificador único e sequencial, data de abertura, data de encerramento, se aplicável, estado e estimativa de roubo.	Detetive Filipe Sousa	Hélder Gomes e Pedro Pereira
	11	08/03/2024	A estimativa de roubo de um caso corresponde à diferença entre a quantidade de minério prevista a ser coletada e a quantidade de minério efetivamente obtida.	Detetive Filipe Sousa	Hélder Gomes e Pedro Pereira
	12	08/03/2024	Um terreno pode ter casos e cada caso está associado a um só terreno.	Detetive Lara Machado	Hélder Gomes e Pedro Pereira
	13	08/03/2024	O estado de um caso pode ser aberto ou fechado.	Detetive Américo Costa	Hélder Gomes e Pedro Pereira
	14	08/03/2024	Caso a quantidade de minério coletada de um determinado dia seja inferior à quantidade de minério mínima prevista, um novo caso é aberto no terreno e todos os funcionários associados tornam-se suspeitos.	Representante Ana Sofia	Hélder Gomes e Pedro Pereira

Descrição	15	08/03/2024	Um funcionário pode ou não pertencer a um ou mais casos, enquanto que a um caso estão associados um ou mais funcionários, obrigatoriamente.	Representante Ana Sofia	Hélder Gomes e Pedro Pereira
	16	08/03/2024	Um funcionário associado a um caso é considerado suspeito.	Representante Ana Sofia	Hélder Gomes e Pedro Pereira
	17	08/03/2024	Um suspeito é caracterizado pelo seu estado, nível de envolvimento e notas, se aplicável.	Detetive Sandra Silva	Hélder Gomes e Pedro Pereira
	18	08/03/2024	O nível de envolvimento é caracterizado por um inteiro de 1 a 10, onde 1 corresponde a pouco envolvimento e 10 a extremamente envolvido.	Detetive Sandra Silva	Hélder Gomes e Pedro Pereira
	19	08/03/2024	O estado de um suspeito pode admitir os valores de inocente, em investigação ou culpado.	Detetive Sandra Silva	Hélder Gomes e Pedro Pereira

Tabela 1: Requisitos de descrição.

2.2.2. Requisitos de manipulação

Tipo	Nº	Data	Descrição do requisito	Fonte	Analista
Manipulação	1	04/03/2024	Listar o prejuízo de um terreno.	Representantes da Lusium	Eduardo Faria e Nuno Silva
	2	08/03/2024	Ver quando é que um funcionário se tornou suspeito de um determinado caso.	Representantes da Lusium	Hélder Gomes e Pedro Pereira
	3	08/03/2024	Listar os suspeitos de um determinado caso.	Representantes da Lusium	Hélder Gomes e Pedro Pereira
	4	08/03/2024	Ver a data do último caso de um determinado funcionário.	Representantes da Lusium	Hélder Gomes e Pedro Pereira
	5	08/03/2024	Listar os casos a que um determinado funcionário está associado.	Detetives	Hélder Gomes e Pedro Pereira
	6	08/03/2024	Ver o dia em que mais casos foram abertos.	Detetives	Hélder Gomes e Pedro Pereira
	7	08/03/2024	Listar os top 5 funcionários por quantidade de casos.	Detetive Américo Costa	Hélder Gomes e Pedro Pereira

Tabela 2: Requisitos de manipulação.

2.2.3. Requisitos de controlo

Tipo	Nº	Data	Descrição do requisito	Fonte	Analista
Controlo	1	04/03/2024	Só um funcionário Representante deve conseguir criar um terreno e alterar os campos de minério previsto e minério coletado de um terreno.	Representantes da Lusium	Eduardo Faria e Nuno Silva
	2	04/03/2024	Só um funcionário Representante deve conseguir criar um funcionário e alterar os campos nome, data de nascimento, função, salário, número de identificação fiscal, fotografia e telemóvel de um funcionário.	Representantes da Lusium	Eduardo Faria e Nuno Silva
	3	04/03/2024	Funcionários Operacionais não têm acesso a qualquer informação da base de dados.	Representante João Guilherme	Eduardo Faria e Nuno Silva
	4	08/03/2024	Só um funcionário Detetive deve conseguir criar um caso e alterar o seu estado e estimativa de roubo.	Detetives	Hélder Gomes e Pedro Pereira
	5	08/03/2024	Só um funcionário Detetive deve conseguir tornar um funcionário suspeito num caso e mudar o estado de um suspeito.	Detetives	Hélder Gomes e Pedro Pereira

Tabela 3: Requisitos de controlo.

2.3. Análise e Validação Geral dos Requisitos

Como foi previamente referido, nesta última fase, o levantamento e organização dos requisitos sofreu um processo rigoroso e crucial de validação da sua consistência e integridade. Deste modo, atendeu-se aos resultados do Inquérito Estatístico de Opinião e à reunião final de validação para se entender as alterações necessárias relativamente aos requisitos que mais discordâncias despertaram.

Durante a reunião final, cuja ata se encontra em anexo (anexo 4), todos os requisitos anteriormente levantados foram rigorosamente avaliados e aprovados pelos membros da equipa “Quatro em Linha”, representantes e detetives da *Lusium*. Os representantes da *Lusium* referiram a alteração de algumas das características de um funcionário, sendo esta a substituição dos números de identificação, nomeadamente, o número de identificação fiscal (NIF), o número de identificação bancária (NIB) e o número de segurança social (NSS) para apenas o número de identificação fiscal (NIF), dado o âmbito do projeto, o que foi aprovado. Pedro Pereira sugeriu a alteração da informação do minério previsto para minério mínimo previsto. João Guilherme, representante da *Lusium*, respondeu que seria uma alteração possível, apenas teriam de comunicar essa alteração aos técnicos que iriam avaliar os terrenos. Por último, Sandra Silva, detetive da B.E.L.O, pediu o acréscimo de um terceiro estado para o suspeito, sendo este “em investigação”, o que também foi unanimemente aprovado.

Assim sendo, a equipa “Quatro em Linha” tem agora a devida certeza que os requisitos se encontram em conformidade com as expectativas dos representantes da *Lusium* e dos detetives da B.E.L.O, conseguindo assegurar que a implementação futura do sistema de gestão de base de dados poderá ser corretamente efetuada sem quaisquer ambiguidades e defeitos. Após esta validação e correção, é então seguro passar para uma fase posterior onde será desenvolvido o modelo concetual, fundamentado pelos requisitos definidos.

3. Modelação Concetual

3.1. Apresentação da Abordagem de Modelação Realizada

Com a conclusão da definição dos requisitos, é então possível passar à organização da informação neles contida. Para isso, a equipa “Quatro em linha” submeteu-se ao desenvolvimento de um modelo concetual capaz de estruturar e representar coerentemente os dados essenciais para atender os requisitos definidos e, para isso, recorreu à criação de um diagrama ER.

A escolha da ferramenta usada na construção do modelo concetual revela-se uma tarefa significativa no desenvolvimento da base de dados, sendo que a escolha certa consegue agilizar todo processo, uma vez que a equipa desenvolvedora do sistema estaria totalmente integrada no mesmo. Para tal, decidiu-se utilizar uma ferramenta que se baseasse na notação Chen, visto ser a notação preferencial da equipa, sendo escolhida, para esse efeito, a ferramenta “brModelo”, fruto da sua clareza e facilidade na manipulação das entidades, atributos e relacionamentos.

Para se proceder à modelação concetual é necessário, numa fase inicial, identificar todas as entidades únicas e relevantes destacadas pelos requisitos levantados, os relacionamentos entre as mesmas e os atributos tanto das entidades, como, possivelmente, dos relacionamentos. Identificadas estas componentes, será então possível proceder às identificações e caracterizações detalhadas de cada uma destas.

3.2. Identificação e Caracterização das Entidades

Tendo em conta os requisitos recolhidos, foi possível identificar as entidades **Funcionário**, **Função**, **Terreno** e **Caso**.

Designação	Descrição	Sinónimos	Ocorrências
Funcionário	Entidade que representa os diferentes trabalhadores da Lusium.	Trabalhador	Um funcionário desempenha 1 função, gere 1 ou mais funcionários e trabalha em 1 ou mais terrenos.
Função	Entidade que define a função que os trabalhadores da Lusium desempenham.	Cargo	Um função é desempenhada por 1 ou mais funcionários.
Terreno	Entidade que representa o local de onde são extraídos os minérios num e dium.	Campo	Um terreno contém 1 ou mais casos e trabalham neste 1 ou mais funcionários.
Caso	Entidade que representa um possível furto de minérios, num dado terreno de extração, em investigação.	Furto/Roubo	Um caso está associado a 1 terreno e contém 1 ou mais funcionários.

Tabela 4: Caracterização das entidades.

Definidas as entidades, é importante passar à enumeração e descrição dos atributos únicos que caracterizam cada uma delas, tendo sempre por base os respetivos requisitos.

Funcionário

Com base nos requisitos 1 e 2 ([Tabela 1](#)), um funcionário é identificado pelo seu **ID** e caracterizado com base no seu **nome**, **data de nascimento**, **salário**, **número de identificação fiscal**, **fotografia** (não obrigatória) e **números de telemóvel**, podendo também apresentar apenas um.

Função

Segundo o requisito 3 ([Tabela 1](#)), uma função é identificada pelo seu **ID** e caracterizada por uma designação que pode tomar os valores de **Operacional**, **Detetive** ou **Representante**.

Terreno

Partindo dos requisitos 6, 7 e 8 ([Tabela 1](#)), um terreno é identificado pelo seu **ID** e caracterizado pelo **minério previsto**, que corresponde à quantidade mínima estimada de minério a coletar por dia, e pelo **minério coletado**, que corresponde à quantidade de minério efetivamente coletado por dia.

Caso

Tendo por base os requisitos 10, 11, 13 e 14 ([Tabela 1](#)), um caso é identificado pelo seu **ID** e definido pelo seu **estado** (aberto ou fechado), **data de abertura** do caso, **data de encerramento** do caso, se porventura estiver fechado, e **estimativa de roubo**, que corresponde à quantidade estimada de minério roubado. Um caso é aberto quando a quantidade de minério mínimo previsto num terreno, num dado dia, é superior à quantidade de minério coletado, cuja diferença origina o valor da estimativa de roubo.

3.3. Identificação e Caracterização dos Relacionamentos

Na modelação concetual surgirão vários relacionamentos entre entidades responsáveis por representar as diferentes interações entre os elementos definidos na base de dados.

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade (relacionada)
Funcionário	N (obrigatório)	gere	M (obrigatório)	Funcionário
Funcionário	N (obrigatório)	desempenha	1 (obrigatório)	Função
Funcionário	N (obrigatório)	trabalha	M (obrigatório)	Terreno
Funcionário	N (parcial)	pertence	M (obrigatório)	Caso
Terreno	1 (parcial)	tem	N (obrigatório)	Caso

Tabela 5: Caracterização dos relacionamentos.

Relacionamento Funcionário - Funcionário (Requisito 9, [Tabela 1](#))

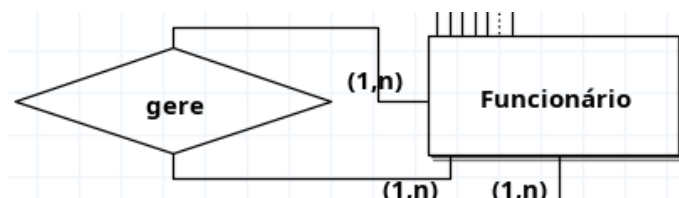


Figura 2: Ilustração do Relacionamento Funcionário - Funcionário.

Relacionamento: Funcionário gere Funcionário.

Descrição: No que toca aos funcionários da Lusium, existem, entre eles, funcionários qualificados para gerir certos grupos de trabalhadores.

Cardinalidade: Funcionário (1,n) - Funcionário (1,n). Um funcionário qualificado gere, obrigatoriamente, um ou mais funcionários. Por outro lado, um funcionário é, obrigatoriamente, gerido por um ou mais funcionários qualificados.

Atributos: Este relacionamento não apresenta atributos.

Relacionamento Funcionário - Função (Requisito 4, [Tabela 1](#))

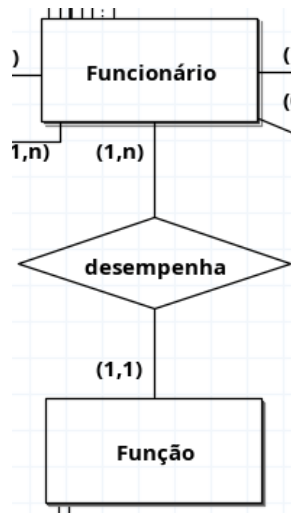


Figura 3: Ilustração do Relacionamento Funcionário - Função.

Relacionamento: Funcionário desempenha Função.

Descrição: Os funcionários da Lusium distinguem-se com base na função que desempenham.

Cardinalidade: Funcionário $(1,n)$ - Função $(1,1)$. Um funcionário desempenha, obrigatoriamente, uma única função. Por outro lado, uma função é, obrigatoriamente, desempenhada por um ou vários funcionários.

Atributos: Este relacionamento não apresenta atributos, até porque a sua cardinalidade não é N:M.

Relacionamento Funcionário - Terreno (Requisito 5, [Tabela 1](#))



Figura 4: Ilustração do Relacionamento Funcionário - Terreno.

Relacionamento: Funcionário trabalha em Terreno.

Descrição: Os funcionários operacionais da Lusium trabalham em terrenos monitorizados pela empresa.

Cardinalidade: Funcionário $(1,n)$ - Terreno $(1,n)$. Um funcionário trabalha, obrigatoriamente, em um ou mais terrenos. Por outro lado, num terreno trabalham, obrigatoriamente, um ou mais funcionários.

Atributos: Este relacionamento não apresenta atributos.

Relacionamento Funcionário - Caso (Requisito 15, [Tabela 1](#))

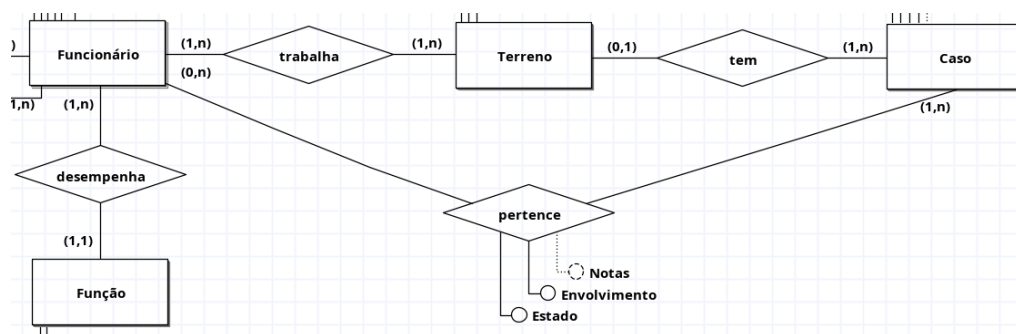


Figura 5: Ilustração do Relacionamento Funcionário - Caso.

Relacionamento: Funcionário pertence a Caso.

Descrição: Os funcionários da Lusium podem estar associados a um caso de furto relacionado com um determinado terreno.

Cardinalidade: Funcionário (0,n) - Caso (1,n). Um funcionário pode ou não pertencer a um ou mais casos. Por outro lado, a um caso pertencem, obrigatoriamente, um ou mais funcionários.

Atributos: Este relacionamento contém os atributos **estado**, **envolvimento** e **notas**. Deste modo, estas características podem ser atribuídas a um determinado suspeito, isto é, um funcionário associado a um caso. Desta forma, **estado** faz referência ao estado atual do suspeito (inocente, em investigação ou culpado), **envolvimento** traduz-se num nível de 1 a 10, capaz de indicar o grau de envolvimento do suspeito no caso e **notas** consistem em comentários adicionais não obrigatórios sobre o suspeito.

Relacionamento Terreno - Caso (Requisito 12, [Tabela 1](#))

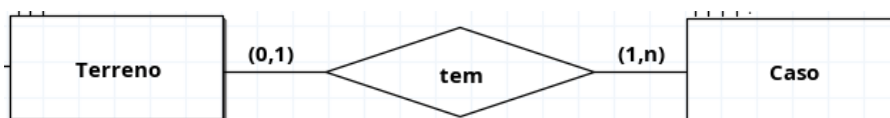


Figura 6: Ilustração do Relacionamento Terreno - Caso.

Relacionamento: Terreno tem Caso.

Descrição: Os terrenos da Lusium podem ter casos de furto de minérios associados.

Cardinalidade: Terreno (0,1) - Caso (1,n). Um terreno pode ou não ter um ou mais casos associados a si mesmo. Por outro lado, um caso está, obrigatoriamente, associado a um único terreno.

Atributos: Este relacionamento não apresenta atributos.

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

Nesta etapa, a caracterização dos atributos das entidades e, possivelmente, dos relacionamentos passa a ser o alvo principal.

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave Primária
Funcionário	ID	Identificador único do funcionário	Não	Não	Não	Não	Sim
	Nome	Nome do funcionário	Não	Não	Não	Não	Não
	Data de nascimento	Data de nascimento do funcionário	Não	Não	Não	Não	Não
	Salário	Salário do funcionário atribuído pela Lusium	Não	Não	Não	Não	Não
	NIF	Número de identificação fiscal do funcionário	Não	Não	Não	Não	Não
	Fotografia	Fotografia identificadora do funcionário	Sim	Não	Não	Não	Não
	Número de telemóvel	Número de telemóvel do funcionário	Não	Não	Sim	Não	Não

Tabela 6: Caracterização dos atributos da entidade Funcionário.

Entidade Funcionário

Nenhum dos atributos da entidade Funcionário (**ID** - chave primária, **Nome**, **Data de nascimento**, **Salário**, **NIF**, **Fotografia** e **Número de Telemóvel**) é **composto** ou **derivado**. Apenas o atributo **Fotografia** pode ser **nulo**, uma vez que um funcionário pode escolher ter ou não uma fotografia associada, sendo todos os outros atributos **não nulos**. Por fim, apenas o atributo **Número de telemóvel** é **multivalorado**, visto que um funcionário pode ter vários números de telemóvel.

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave Primária
Função	ID	Identificador único do função	Não	Não	Não	Não	Sim
	Designação	Designação da função	Não	Não	Não	Não	Não

Tabela 7: Caracterização dos atributos da entidade Função.

Entidade Função

Nenhum dos atributos da entidade Função (**ID** - chave primária e **designação**) é **composto**, **derivado**, **nulo** ou **multivalorado**.

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave Primária
Terreno	ID	Identificador único do terreno	Não	Não	Não	Não	Sim
	Minério previsto	Quantidade mínima estimada de minério a coletar por dia	Não	Não	Não	Não	Não
	Minério coletado	Quantidade de minério efetivamente coletado por dia	Não	Não	Não	Não	Não

Tabela 8: Caracterização dos atributos da entidade Terreno.

Entidade Terreno

Nenhum dos atributos da entidade Terreno (ID - chave primária, **Minério previsto** e **Minério coletado**) é **composto**, **derivado**, **nulo** ou **multivalorado**.

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave Primária
Caso	ID	Identificador único do caso	Não	Não	Não	Não	Sim
	Data de abertura	Data de abertura do caso	Não	Não	Não	Não	Não
	Estado	Estado atual do caso	Não	Não	Não	Não	Não
	Estimativa de roubo	Quantidade estimada de minério roubado	Não	Não	Não	Sim	Não
	Data de encerramento	Data de encerramento do caso	Sim	Não	Não	Não	Não

Tabela 9: Caracterização dos atributos da entidade Caso.

Entidade Caso

Nenhum dos atributos da entidade Caso (ID - chave primária, **Data de abertura**, **Estado**, **Estimativa de roubo** e **Data de encerramento**) é **composto** ou **multivalorado**. Apenas o atributo **Estimativa de roubo** pode ser **derivado**, uma vez que provém da diferença entre os valores dos atributos **Minério previsto** e **Minério coletado** associados à entidade **Terreno**. Por fim, apenas o atributo **Data de encerramento** pode ser **nulo**, pois um caso só apresenta uma data de encerramento a partir do momento em que o seu estado passa para “fechado”.

Relacionamento	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave Primária
Funcionário - Caso	Estado	Estado atual do suspeito no caso	Não	Não	Não	Não	Não
	Envolvimento	Grau de envolvimento do suspeito no caso	Não	Não	Não	Não	Não
	Notas	Observações importantes sobre o suspeito	Sim	Não	Não	Não	Não

Tabela 10: Caracterização dos atributos do relacionamento Funcionário - Caso.

Relacionamento Funcionário - Caso

Nenhum dos atributos do relacionamento Funcionário - Caso (**Estado**, **Envolvimento** e **Notas**) é **composto**, **derivado** ou **multivalorado**. Apenas o atributo **Notas** pode ser **nulo**, uma vez que podem não existir notas associadas ao relacionamento.

3.5. Apresentação e Explicação do Diagrama ER Produzido

Após a identificação e caracterização detalhada de cada entidade, atributo e relacionamento nos tópicos anteriores, prosseguimos agora para a apresentação de todo o diagrama ER que suporta o modelo da base de dados em questão.

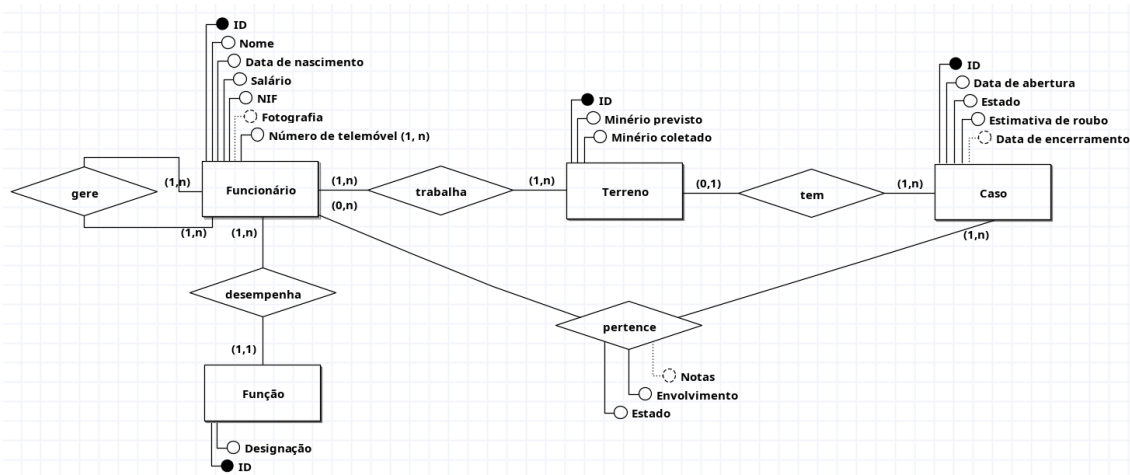


Figura 7: Ilustração do diagrama ER produzido.

Neste diagrama, é possível identificar de forma clara todas as entidades da base de dados, bem como os relacionamentos responsáveis por interligar as mesmas. Deste modo, o diagrama é interpretado da seguinte forma: um funcionário desempenha obrigatoriamente uma função; um funcionário qualificado gere outros funcionários; os funcionários operacionais trabalham em diversos terrenos; a um terreno podem estar associados vários casos; num caso podem estar contidos vários funcionários.

Por fim, podemos também identificar, em cada entidade e num dos relacionamentos, os atributos que os definem. Em todas as entidades destacam-se os atributos que constituem uma chave primária da mesma, visto serem representados por um círculo totalmente preenchido. Para além disso, no diagrama são também visíveis outros tipos de atributos, tais como multivalorados e opcionais.

Assim, com a conclusão desta fase, é possível passar para a construção de um novo modelo capaz de especificar a caracterização de cada um dos elementos, sendo este o modelo de dados lógico.

4. Modelação Lógica

4.1. Construção e Validação do Modelo de Dados Lógico

Após a conclusão e revisão da modelação concetual, a equipa da “Quatro em Linha” está agora a avançar para a elaboração do Modelo de Dados Lógico. Este tipo de modelo descreve os elementos de dados em detalhe e é utilizado para criar uma compreensão visual das entidades de dados, atributos, chaves e relacionamentos. O objetivo principal é estabelecer uma estrutura fundamental para os componentes da camada semântica em sistemas de gestão de dados, sem depender de um motor específico. Propôs-se a elaboração de um Modelo de Dados Lógico com o intuito de definir como o sistema em desenvolvimento deve ser implementado.

Em particular, o Modelo Lógico de Dados proporciona mais detalhes na representação das entidades e dados, oferecendo um contexto visual para os dados, atributos e relações entre eles. Para alcançar este objetivo, a equipa de desenvolvimento optou por utilizar a ferramenta MySQL WorkBench para estudar e construir o modelo esperado.

O MySQL WorkBench é uma plataforma de modelação de dados poderosa e versátil, projetada para auxiliar na criação, visualização e gestão de modelos de dados lógicos e físicos. Com uma interface intuitiva e diversas funcionalidades, o MySQL WorkBench permite aos utilizadores criar representações precisas e compreensíveis dos dados e das suas inter-relações, facilitando assim o desenvolvimento de sistemas de informação robustos e eficazes.

4.2. Apresentação e Explicação do Modelo Lógico Produzido

Para a correta construção do modelo lógico, a equipa da “Quatro em Linha” reuniu-se para considerar como proceder à execução do diagrama, apoiado, inicialmente, por uma ponderação sobre a quantidade de tabelas a utilizar. Procede-se à apresentação dos resultados:

- 4 tabelas, uma por entidade;
- 2 tabelas, uma por cada relacionamento N para M;
- 1 tabela de um atributo multivalorado;
- 1 tabela de um relacionamento unário.

A equipa iniciou a produção das 8 tabelas referidas acima de acordo com a ordem de formulação do modelo concetual. Para tal, inaugurou o diagrama com a criação da tabela de Funcionário ao herdar os atributos já definidos no diagrama de sustento, que se define pelas seguintes colunas:

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Funcionário	"Funcionário_ID"	INT	-
	"Nome"	VARCHAR(75)	-
	"Data_de_nascimento"	DATE	-
	"Salário"	INT	-
	"NIF"	VARCHAR(10)	-
	"Fotografia"	VARCHAR(150)	-
	"Função_ID"	INT	"Função_ID" (Função)

Tabela 11: Caracterização da tabela lógica do Funcionário.

De acordo com esta decisão, surgiu a necessidade de criar 2 tabelas com um relacionamento associado, sendo elas, a tabela de Número de telemóvel, derivada do atributo multivalorado "Número de telemóvel" da entidade Funcionário, e Função, constituída pela "Designação" correspondente ao cargo de um determinado Funcionário:

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Número de telemóvel	"Número_de_Telemóvel"	INT	-
	"Funcionário_ID"	INT	"Funcionário_ID" (Funcionário)

Tabela 12: Caracterização da tabela lógica do Número de telemóvel.

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Função	"Função_ID"	INT	-
	"Designação"	ENUM("Representante", "Detetive", "Operacional")	-

Tabela 13: Caracterização da tabela lógica da Função.

De acordo com os requisitos, torna-se necessário fazer surgir uma nova tabela denominada Gere, esta que deriva de um relacionamento unário da entidade Funcionário para a própria, permitindo assim a autogestão de informações:

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Gere	"Funcionário_Gestor_ID"	INT	"Funcionário_ID" (Funcionário)
	"Funcionário_ID"	INT	"Funcionário_ID" (Funcionário)

Tabela 14: Caracterização da tabela lógica de Gere.

Deu-se seguimento aos trabalhos com a conversão da entidade Terreno para uma tabela, à qual se associaram os atributos previamente definidos:

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Terreno	"Terreno_ID"	INT	-
	"Minério_previsto"	INT	-
	"Minério_coletado"	INT	-

Tabela 15: Caracterização da tabela lógica do Terreno.

Na sequência da referência da tabela anterior, cria-se então a nova tabela Caso, que mais uma vez reflete os atributos definidos na modelação concetual:

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Caso	"Caso_ID"	INT	-
	"Data_de_abertura"	DATE	-
	"Estado"	ENUM("Aberto", "Fechado")	-
	"Estimativa_de_roubo"	INT	-
	"Data_de_encerramento"	DATE	-
	"Terreno_ID"	INT	"Terreno_ID" (Terreno)

Tabela 16: Caracterização da tabela lógica do Caso.

Ao analisar o diagrama conceitual, repara-se no relacionamento N para M entre as entidades de Funcionário e Terreno, este que, no processo de conversão, culmina com o aparecimento de uma nova tabela que estabelece o relacionamento:

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Trabalha	"Funcionário_ID"	INT	"Funcionário_ID" (Funcionário)
	"Terreno_ID"	INT	"Terreno_ID" (Terreno)

Tabela 17: Caracterização da tabela lógica de Trabalha.

Finalmente, finda-se o processo de obtenção do modelo lógico com a transfiguração do relacionamento fulcral entre as entidades Funcionário e Caso que, ao representar um relacionamento N para M com atributos associados, concebe a última tabela, Suspeito.

Tabela	Nome da coluna	Tipo de dado	Referência (tabela)
Suspeito	"Funcionário_ID"	INT	"Funcionário_ID" (Funcionário)
	"Caso_ID"	INT	"Caso_ID" (Caso)
	"Estado"	ENUM("Inocente", "Em Investigação", "Culpado")	-
	"Envolvimento"	INT	-
	"Notas"	TEXT(256)	-

Tabela 18: Caracterização da tabela lógica do Suspeito.

Na modelação lógica, os relacionamentos entre entidades desempenham um papel crucial na representação da interação entre diferentes elementos de um sistema de gestão de base de dados:

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 Caso(N) e Terreno(1)
- "desempenha" entre Funcionário(N) e Função(1)

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:

- "trabalha" entre Funcionário e Terreno
- "pertence" entre Funcionário e Caso

Relacionamento unário:

Na ocorrência “gere”, entre a entidade Funcionário e a própria, surge um relacionamento unário, do grau N:M, pois um ou mais funcionários podem gerir outros funcionários. Isto representa uma estrutura hierárquica onde um funcionário é responsável pela gestão de outros, dentro da organização.

Com base nos parâmetros e nos relacionamentos estabelecidos anteriormente, foi desenvolvido um modelo lógico para representar de forma precisa e organizada as interações entre as entidades envolvidas. Este modelo visa proporcionar uma estrutura robusta para armazenar e gerir informações relacionadas aos diversos elementos do sistema de gestão de base de dados.

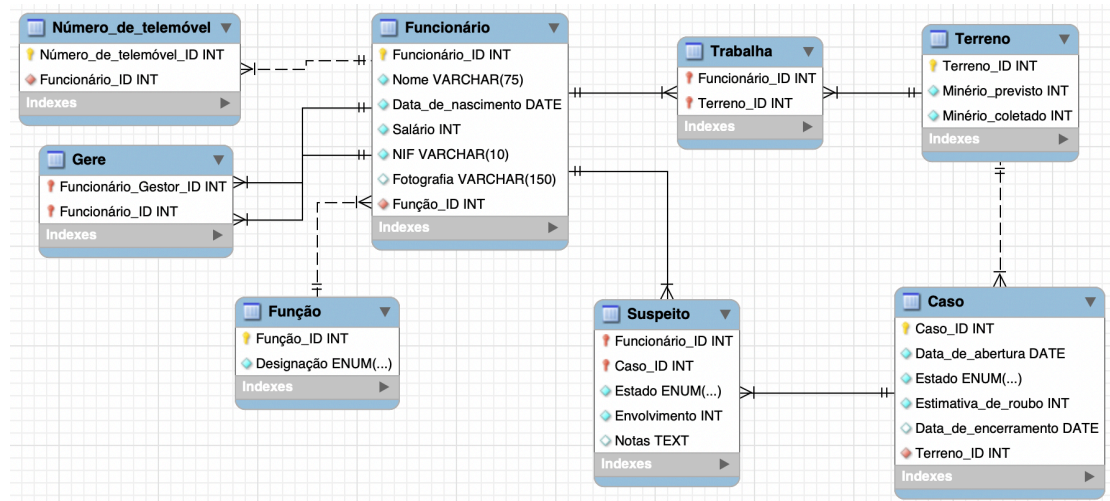


Figura 8: Ilustração do modelo lógico produzido.

4.3. Normalização de Dados

A normalização de uma base de dados é um processo crucial na execução de um sistema de gestão de base de dados relacional. O objetivo principal da normalização é organizar os dados de forma a minimizar a redundância, ambiguidade e inconsistência, garantindo, em simultâneo, a integridade da informação e facilitando as operações de consulta e manipulação da mesma.

Existem várias formas normais, habitualmente descritas pela nomenclatura 1NF(Primeira Forma Normal), 2NF(Segunda Forma Normal), 3NF(Terceira Forma Normal), entre outras. Cada uma destas formas normais aborda diferentes tipos de problemas de difusão e dependências funcionais entre os atributos das tabelas.

Na Primeira Forma Normal (1NF), as tabelas são organizadas de forma a que cada célula contenha apenas um valor indivisível, e não listas de valores separados por vírgulas, pontos, ou outros delimitadores. Adicionalmente, cada tabela deve ter uma chave primária única que a identifica de forma exclusiva. Por exemplo, na tabela de Caso verifica-se a garantia de valores indivisíveis e a unicidade da chave primária. Após a análise atenta do modelo lógico em produção, verifica-se o cumprimento desta regra de normalização.

Na Segunda Forma Normal (2NF), todas as dependências parciais são eliminadas, o que significa que cada atributo na tabela deve depender totalmente da chave primária e não apenas de parte dela. Para corroborar esta regra, exemplificando, na tabela de Número de telemóvel, o atributo “Número de telemóvel” subordina-se à chave primária correspondente. Valida-se então esta norma não obstante à confirmação da 1NF, de forma a respeitar a ordem de precedência estabelecida.

Na Terceira Forma Normal (3NF), as dependências transitivas são eliminadas, ou seja, não deve haver atributos que dependam de outros que não seja chave primária da tabela em que se encontram definidos. Constata-se a veracidade desta norma no nosso modelo lógico, por exemplo, na tabela de Função, o atributo “Designação” depende só e apenas da chave primária.

Ao considerar a homologação sequencial e progressiva destas três formas normais, confirma-se a validade do modelo lógico em produção e verificação. Complementarmente, existem mais fórmulas como a Forma Normal de Boyce-Codd (BCNF) ou Quarta e Quinta Formas Normais (4NF e 5NF) que impõem uma especificidade e detalhe superior na minimização de redundâncias e irregularidades, no entanto, descartámo-las dado a complexidade excessiva no âmbito do nosso desenvolvimento.

4.4. Validação do Modelo com Interrogações do Utilizador

No âmbito do projeto, é fundamental garantir que o modelo lógico concebido satisfaz os requisitos de manipulação estabelecidos. Uma das formas de validação do modelo ocorre através da execução de consultas ou queries, neste caso, representadas em Álgebra Relacional. Esta abordagem permite comprovar se as operações de manipulação de dados permitem fornecer os dados previstos.

Para cumprir este fim, apresentam-se de seguida sete expressões em Álgebra Relacional que refletem os sete requisitos de manipulação estabelecidos anteriormente:

Listar o prejuízo de um terreno: (Requisito 1, [Tabela 2](#))

Para o exemplo do terreno de ID número 1.

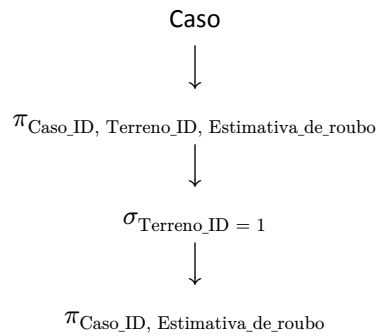


Figura 9: Ilustração do primeiro diagrama de Álgebra Relacional.

$\text{Casos_total} \leftarrow \pi_{\text{Caso_ID}, \text{Terreno_ID}, \text{Estimativa_de_roubo}} (\text{Casos})$

$\text{Caso_Terreno} \leftarrow \sigma_{\text{Terreno_ID} = 1} (\text{Casos_total})$

$\pi_{\text{Caso_ID}, \text{Estimativa_de_roubo}} (\text{Casos_Terreno})$

Ver quando é que um funcionário se tornou suspeito de um determinado caso: (Requisito 2, [Tabela 2](#))

Para o exemplo do caso de ID número 1.

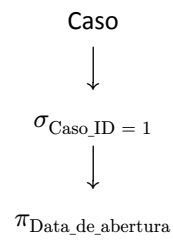


Figura 10: Ilustração do segundo diagrama de Álgebra Relacional.

$\pi_{\text{Data_de_abertura}} (\sigma_{\text{Caso_ID} = \text{ID_Caso_Pretendido}} (\text{Caso}))$

Listar os suspeitos de um determinado caso: (Requisito 3, [Tabela 2](#))

Para o exemplo do caso de ID número 1.

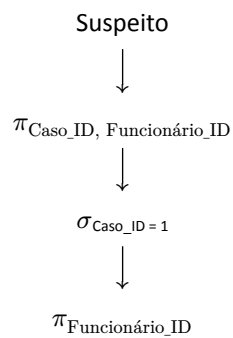


Figura 11: Ilustração do terceiro diagrama de Álgebra Relacional.

$\text{Suspeitos} \leftarrow \pi_{\text{Caso_ID}, \text{Funcionário_ID}} (\text{Suspeitos})$

$\text{Suspeitos_Caso} \leftarrow \sigma_{\text{Caso_ID} = 1} (\text{Suspeitos})$

$\pi_{\text{Funcionário_ID}} (\text{Suspeitos_Caso})$

Ver a data do último caso de um determinado funcionário: (Requisito 4, [Tabela 2](#))

Para o exemplo do funcionário de ID número 1.

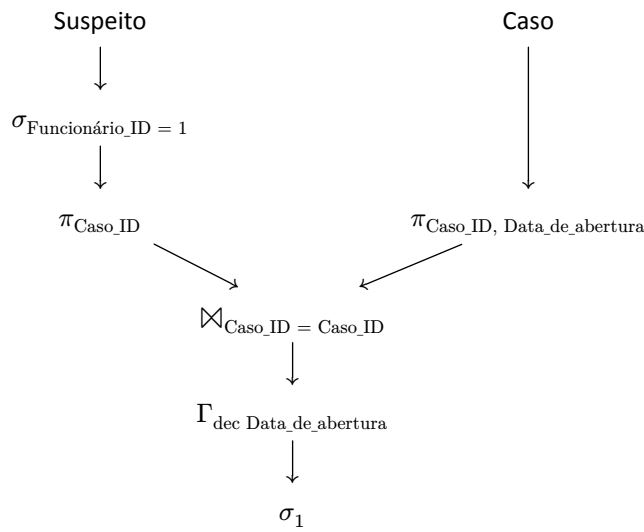


Figura 12: Ilustração do quarto diagrama de Álgebra Relacional.

$\text{Suspeito} \leftarrow \pi_{\text{Caso_ID}} (\sigma_{\text{Funcionário_ID} = 1} (\text{Suspeito}))$

$\text{Caso} \leftarrow \pi_{\text{Caso_ID}, \text{Data_de_abertura}}$

$\text{Casos_Suspeito} \leftarrow (\text{Suspeito}) \bowtie_{\text{Caso_ID} = \text{Caso_ID}} (\text{Caso})$

$\sigma(1) \tau_{\text{dec Data_de_abertura}} (\text{Casos_Suspeitos})$

Listar os casos a que um determinado funcionário está associado: (Requisito 5, [Tabela 2](#))

Para o exemplo do funcionário de ID número 1.

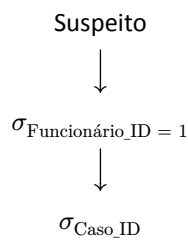


Figura 13: Ilustração do quinto diagrama de Álgebra Relacional.

$\pi_{\text{Caso_ID}} (\sigma_{\text{Funcionário_ID} = 1} (\text{Suspeito}))$

Ver o dia em que mais casos foram abertos: (Requisito 6, [Tabela 2](#))

Quantidade_de_casos $F_{COUNT(Data_de_abertura), \pi(Data_de_abertura)}$ (Caso)

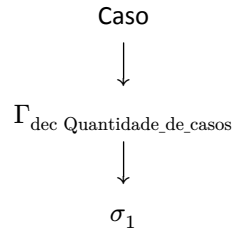


Figura 14: Ilustração do sexto diagrama de Álgebra Relacional.

Quantidade_Casos $F_{COUNT Data_de_abertura}$ (Caso)

$\pi_{Data\ de\ abertura} (\sigma(1) \tau_{Quantidade_Casos_COUNT} (Quantidade_Casos))$

Listar os top 5 funcionários por quantidade de casos: (Requisito 7, [Tabela 2](#))

Quantidade_de_casos $F_{COUNT(Funcionário_ID), \pi(Funcionário_ID)}$ (Suspeito)

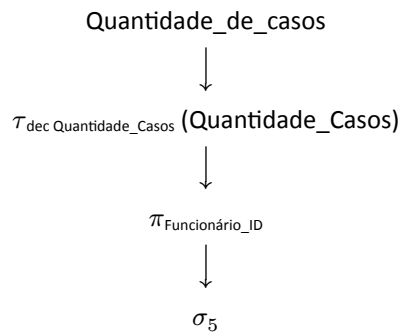


Figura 15: Ilustração do sétimo diagrama de Álgebra Relacional.

Quantidade_Casos $F_{COUNT Funcionário_ID}$ (Suspeito)

$\pi_{Funcionário_ID} (\sigma(5) \tau_{dec\ Quantidade_Casos} (Quantidade_Casos))$

5. Implementação Física

5.1. Apresentação e explicação da base de dados implementada

De forma a iniciar o desenvolvimento do sistema de gestão de base de dados, decidimos criar a implementação MySQL de cada tabela apresentada anteriormente no modelo lógico.

Dá-se início ao processo com duas instruções SQL essenciais: a primeira é responsável pela criação da base de dados com o nome atribuído *Lusium*, a segunda indica ao sistema qual a base de dados que deve ser utilizada para as operações executadas.

```
CREATE DATABASE IF NOT EXISTS Lusium;  
  
USE Lusium;
```

Figura 16: Instruções SQL para criar e utilizar a base de dados.

A primeira tabela a ser implementada no modelo físico é a Função, esta que, ao seguir a estrutura habitual de SQL, é inaugurada com duas colunas. Possui o identificador único “Função_ID”, que é definido pelos parâmetros “INT”, que define o tipo de dados, “PRIMARY KEY” que garante a atomicidade e capacidade identificativa, “AUTO_INCREMENT” para que o sistema atribua automaticamente um identificador sequencial e “NOT NULL”, para assegurar que o valor existe. Representa-se também a “Designação”, do tipo “ENUM()”, para restringir os valores possíveis na coluna Operacional, Detetive ou Representante, apoiada pelo “NOT NULL”, de forma a certificar uma atribuição de valor.

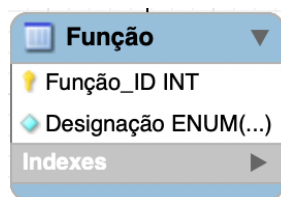


Figura 17: Tabela lógica da Função.

```
-- Criação da tabela Função  
CREATE TABLE Função (  
    Função_ID INT AUTO_INCREMENT NOT NULL,  
    Designação ENUM('Operacional',  
                    'Detetive', 'Representante') NOT NULL,  
    PRIMARY KEY (Função_ID)  
);
```

Figura 18: Implementação correspondente à Função.

A tabela Funcionário armazena dados essenciais dos colaboradores. A coluna “Funcionário_ID” é um “INT” que serve como “PRIMARY KEY” e é “NOT NULL”, garantindo que cada funcionário tenha um identificador único. “Nome” é um “VARCHAR(75)” e também “NOT NULL”, permitindo nomes com até 75 caracteres. “Data_de_nascimento” é um campo “DATE” e “NOT NULL”, registrando a data de nascimento de cada funcionário. “Salário” é um “INT” e “NOT NULL”, refletindo o salário do funcionário. “NIF” é um “VARCHAR(10)” e “NOT NULL”, contendo o número fiscal do funcionário. “Fotografia” é um “VARCHAR(150)” e pode ser “NULL”, indicando que a fotografia é opcional. “Função_ID” é um “INT” e “NOT NULL”, atuando como uma chave estrangeira que referencia a tabela Função.

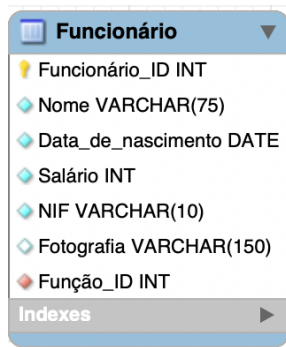


Figura 19: Tabela lógica do Funcionário.

```
-- Criação da tabela Funcionário
CREATE TABLE Funcionário (
    Funcionário_ID INT AUTO_INCREMENT NOT NULL,
    Nome VARCHAR(75) NOT NULL,
    Data_de_nascimento DATE NOT NULL,
    Salário INT NOT NULL,
    NIF VARCHAR(10) NOT NULL,
    Fotografia VARCHAR(150) NULL,
    Função_ID INT NOT NULL,
    PRIMARY KEY (Funcionário_ID),
    FOREIGN KEY (Função_ID) REFERENCES
    Função(Função_ID)
);
```

Figura 20: Implementação correspondente ao Funcionário.

A tabela Número de telemóvel é dedicada aos contactos telefónicos dos funcionários. “Número_de_tele-móvel_ID” é um “INT”, definido como “PRIMARY KEY” e “NOT NULL”, assegurando a unicidade de cada número. “Funcionário_ID” é um “INT” e “NOT NULL”, estabelecendo uma chave estrangeira que liga o número de telemóvel ao seu respectivo funcionário na tabela Funcionário.

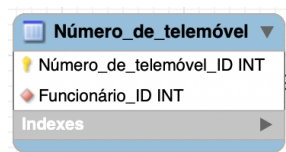


Figura 21: Tabela lógica do Número de Telemóvel.

```
-- Criação da tabela Número de telemóvel
CREATE TABLE Número_de_tele-móvel (
    Número_de_tele-móvel_ID INT NOT NULL,
    Funcionário_ID INT NOT NULL,
    PRIMARY KEY (Número_de_tele-móvel_ID),
    FOREIGN KEY (Funcionário_ID) REFERENCES
    Funcionário(Funcionário_ID)
);
```

Figura 22: Implementação correspondente ao Número de Telemóvel.

A tabela Gere representa a hierarquia de gestão. “Funcionário_Gestor_ID” e “Funcionário_ID” são ambos “INT” e “NOT NULL”, formando uma “PRIMARY KEY” composta que identifica cada relação de gestão. Ambos são também chaves estrangeiras que apontam para “Funcionário_ID” na tabela Funcionário, delineando quem gere quem.

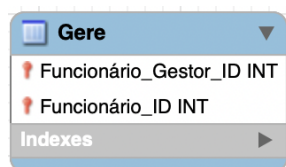


Figura 23: Tabela lógica do Gere.

```
-- Criação da tabela Gere
CREATE TABLE Gere (
    Funcionário_Gestor_ID INT NOT NULL,
    Funcionário_ID INT NOT NULL,
    PRIMARY KEY (Funcionário_Gestor_ID,
    Funcionário_ID),
    FOREIGN KEY (Funcionário_Gestor_ID)
    REFERENCES Funcionário(Funcionário_ID),
    FOREIGN KEY (Funcionário_ID) REFERENCES
    Funcionário(Funcionário_ID)
);
```

Figura 24: Implementação correspondente ao Gere.

A tabela Terreno regista os terrenos de mineração. “Terreno_ID” é um “INT”, marcado como “PRIMARY KEY”, “AUTO_INCREMENT” para que o sistema atribua automaticamente um identificador sequencial e

“NOT NULL”, identificando cada terreno de forma única. “Minério_previsto” e “Minério_coletado” são ambos “INT” e “NOT NULL”, quantificando o minério esperado e extraído, respectivamente.



Figura 25: Tabela lógica do Terreno.

```
-- Criação da tabela Terreno
CREATE TABLE Terreno (
    Terreno_ID INT AUTO_INCREMENT NOT NULL,
    Minério_previsto INT NOT NULL,
    Minério_coletado INT NOT NULL,
    PRIMARY KEY (Terreno_ID)
);
```

Figura 26: Implementação correspondente ao Terreno.

A tabela Trabalha associa funcionários aos terrenos onde operam. “Funcionário_ID” e “Terreno_ID” são “INT” e “NOT NULL”, constituindo uma “PRIMARY KEY” composta que garante a singularidade da relação. Estas colunas são chaves estrangeiras que referenciam as tabelas “Funcionário” e “Terreno”, respectivamente.



Figura 27: Tabela lógica do Trabalha.

```
-- Criação da tabela Trabalha
CREATE TABLE Trabalha (
    Funcionário_ID INT NOT NULL,
    Terreno_ID INT NOT NULL,
    PRIMARY KEY (Funcionário_ID,
    Terreno_ID),
    FOREIGN KEY (Funcionário_ID) REFERENCES
    Funcionário(Funcionário_ID),
    FOREIGN KEY (Terreno_ID) REFERENCES
    Terreno(Terreno_ID)
);
```

Figura 28: Implementação correspondente ao Trabalha.

A tabela Caso é usada para o acompanhamento de investigações. “Caso_ID” é um “INT”, servindo como “PRIMARY KEY”, “AUTO_INCREMENT” para que o sistema atribua automaticamente um identificador sequencial e “NOT NULL”. “Data_de_abertura” é um campo “DATE” e “NOT NULL”, marcando o início do caso. “Estado” é um “ENUM(‘Aberto’, ‘Fechado’)” e “NOT NULL”, indicando o estado atual do caso. “Estimativa_de_roubo” é um “INT” e “NOT NULL”, estimando o valor subtraído. “Data_de_encerramento” pode ser um campo “DATE” e “NULL”, representando a data de conclusão do caso. “Terreno_ID” é um “INT” e “NOT NULL”, funcionando como uma chave estrangeira que liga o caso ao terreno relevante.

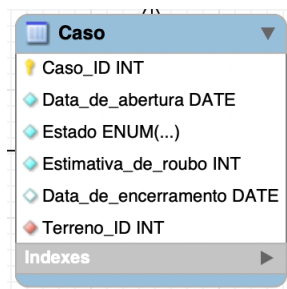


Figura 29: Tabela lógica do Caso.

```
-- Criação da tabela Caso
CREATE TABLE Caso (
    Caso_ID INT AUTO_INCREMENT NOT NULL,
    Data_de_abertura DATE NOT NULL,
    Estado ENUM('Aberto', 'Fechado') NOT
    NULL,
    Estimativa_de_roubo INT NOT NULL,
    Data_de_encerramento DATE NULL,
    Terreno_ID INT NOT NULL,
    PRIMARY KEY (Caso_ID),
    FOREIGN KEY (Terreno_ID) REFERENCES
    Terreno(Terreno_ID)
);
```

Figura 30: Implementação correspondente ao Caso.

A tabela Suspeito mantém o registo dos suspeitos nos casos. “Funcionário_ID” e “Caso_ID” são “INT” e “NOT NULL”, formando uma “PRIMARY KEY” composta. “Estado” é um “ENUM(‘Inocente’, ‘Em investigação’, ‘Culpado’)” e “NOT NULL”, refletindo a condição do suspeito. “Envolvimento” é um “INT” e “NOT NULL”, medindo o grau de envolvimento do suspeito no caso. “Notas” é um campo “TEXT” e pode ser “NULL”, permitindo anotações adicionais sobre o suspeito.

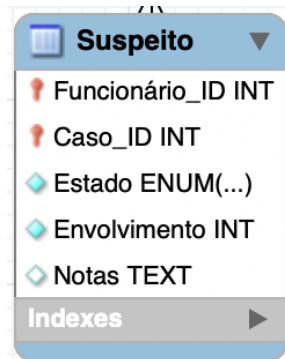


Figura 31: Tabela lógica do Suspeito.

```
-- Criação da tabela Suspeito
CREATE TABLE Suspeito (
    Funcionário_ID INT NOT NULL,
    Caso_ID INT NOT NULL,
    Estado ENUM('Inocente', 'Em
investigação', 'Culpado') NOT NULL,
    Envolvimento INT NOT NULL,
    Notas TEXT NULL,
    PRIMARY KEY (Funcionário_ID, Caso_ID),
    FOREIGN KEY (Funcionário_ID) REFERENCES
Funcionário(Funcionário_ID),
    FOREIGN KEY (Caso_ID) REFERENCES
Caso(Caso_ID)
);
```

Figura 32: Implementação correspondente ao Suspeito.

5.2. Criação de utilizadores da base de dados

Beneficiando agora dos requisitos de controlo recolhidos inicialmente, a equipa passou a desenvolver e definir toda a caracterização relativa aos utilizadores da base de dados em questão. Neste caso, os únicos utilizadores a serem criados no sistema seriam os representantes da *Lusium* e os detetives do departamento de espionagem. Seguida da criação dos mesmos, são definidas as suas permissões de trabalho na base de dados, de forma a garantir a segurança e integridade da mesma.

Na base de dados, o utilizador “detetive” foi criado com permissões limitadas ao controlo de casos e suspeitos adjacentes. Para tal, a este utilizador são atribuídas permissões de “SELECT”, “INSERT” e “UPDATE” nas tabelas Caso e Suspeito e de, unicamente, “SELECT” na tabela Funcionário e na view FuncionariosEmCasos. Para além disso, também conta com permissões de execução do procedimento CriarCasosETornarSuspeitos e da função CalcularEstimativaDeRoubo.

```
CREATE USER 'detetive'@'localhost' IDENTIFIED BY 'detetive1234';

GRANT SELECT, INSERT, UPDATE ON Caso TO 'detetive'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Suspeito TO 'detetive'@'localhost';
GRANT SELECT ON Funcionário TO 'detetive'@'localhost';
GRANT SELECT ON FuncionariosEmCasos TO 'detetive'@'localhost';
GRANT EXECUTE ON PROCEDURE CriarCasoETornarSuspeitos TO 'detetive'@'localhost';
GRANT EXECUTE ON FUNCTION CalcularEstimativaDeRoubo TO 'detetive'@'localhost';
```

Figura 33: Criação de utilizadores “detetive” e devidas permissões.

Foi também criado o utilizador “representante” com permissões limitadas ao controlo de funcionários e terrenos. Assim, o mesmo tem acesso às tabelas Gere, Função, Terreno, Trabalha, Funcionário e Número de telemóvel, onde está restrito às operações de “SELECT”, “INSERT” e “UPDATE” nas mesmas. Adicionalmente, também dispõe da permissão de “SELECT” na vista FuncionariosEmTerrenos.

```
CREATE USER 'representante'@'localhost' IDENTIFIED BY 'representante1234';

GRANT SELECT, INSERT, UPDATE ON Gere TO 'representante'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Função TO 'representante'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Terreno TO 'representante'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Trabalha TO 'representante'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Funcionário TO 'representante'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Número de telemóvel TO 'representante'@'localhost';
GRANT SELECT ON FuncionariosEmTerrenos TO 'representante'@'localhost';
```

Figura 34: Criação de utilizadores “representantes” e devidas permissões.

Desta forma, ficam definidos os utilizadores da base de dados, assim como as suas permissões. Os detetives apenas têm autorização para manipular os casos e os seus suspeitos, enquanto que os representantes têm permissões mais abrangentes que lhes permitem manipular todas as tabelas, com a exceção das tabelas caso e suspeito.

5.3. Povoamento da base de dados

Para o povoamento da base de dados, implementámos duas abordagens distintas. A primeira abordagem utilizou um ficheiro SQL, onde, através de instruções INSERT, inserimos um número reduzido de entradas para facilitar a visualização e a execução de queries simples. Esta abordagem permite uma compreensão mais clara das operações básicas da base de dados.

A segunda abordagem foi realizada utilizando Python, que permitiu a inserção de um volume maior de dados na base de dados. Utilizando a biblioteca *mysql.connector*, estabelecemos a conexão com o servidor MySQL e gerámos dados aleatórios para povoar a base de dados de forma programática. Esta técnica é particularmente útil para simular um ambiente mais próximo do real, possibilitando testes de desempenho e escalabilidade, bem como a validação de queries mais complexas e robustas.

Inicia-se o processo pela introdução de dados na tabela “Função”, que contém informações sobre as funções que os funcionários podem desempenhar na *Lusium*. Adicionam-se três funções: “Operacional”, “Detetive” e “Representante”. Cada função tem um ID único atribuído automaticamente e uma designação correspondente.

```
-- Inserção de dados na tabela Função
INSERT INTO Função (Designação) VALUES
  ('Operacional'),
  ('Detetive'),
  ('Representante');
```

Figura 35: Inserção de dados na tabela Função com SQL.

Em seguida, procedemos à inserção de dados na tabela “Funcionário”, que armazena informações detalhadas sobre os funcionários, incluindo o seu ID, nome, data de nascimento, salário, NIF, fotografia e o ID da função desempenhada. Este passo é essencial para registar os dados pessoais e profissionais dos funcionários na base de dados.

```
-- Inserção de dados na tabela Funcionário
INSERT INTO Funcionário (Nome, Data_de_nascimento, Salário, NIF, Fotografia, Função_ID)
VALUES
    ('Miguel Silva', '1985-03-15', 1200, '123456789', 'foto_miguel.jpg', 1),
    ('Ana Costa', '1990-07-21', 2800, '987654321', 'foto_ana.jpg', 2),
    ('Pedro Santos', '1982-12-10', 3000, '456789123', 'foto_pedro.jpg', 3),
    ('Sofia Pereira', '1988-05-02', 1000, '789123456', 'foto_sofia.jpg', 1),
    ('Rui Oliveira', '1995-09-18', 2600, '321654987', 'foto_rui.jpg', 2),
    ('Inês Rodrigues', '1980-11-30', 850, '654987321', 'foto_ines.jpg', 1),
    ('Tiago Fernandes', '1993-04-25', 2900, '987321654', 'foto_tiago.jpg', 3);
```

Figura 36: Inserção de dados na tabela Funcionário com SQL.

A tabela “Número_de_telefone” regista os números de telefone associados a cada funcionário, estabelecendo uma ligação direta entre o ID do número de telefone e o ID do funcionário correspondente.

```
-- Inserção de dados na tabela Número_de_telefone
INSERT INTO Número_de_telefone (Número_de_telefone_ID, Funcionário_ID) VALUES
    (934678592, 1),
    (966492873, 2),
    (922245762, 3),
    (936457856, 4),
    (926486516, 5),
    (964884547, 6),
    (964554178, 7);
```

Figura 37: Inserção de dados na tabela Número_de_telefone com SQL.

Na tabela “Gere”, registamos a relação de gestão entre funcionários, onde um funcionário gestor é associado a outros funcionários que estão sob sua supervisão. Este relacionamento é representado pelos IDs do gestor e do funcionário.

```
-- Inserção de dados na tabela Gere
INSERT INTO Gere (Funcionário_Gestor_ID, Funcionário_ID) VALUES
    (3, 1),
    (3, 2),
    (3, 4),
    (6, 5),
    (6, 6),
    (6, 7);
```

Figura 38: Inserção de dados na tabela Gere com SQL.

A tabela “Terreno” armazena informações sobre os terrenos geridos pela empresa, incluindo o ID do terreno, a quantidade de minério prevista e a quantidade de minério coletado. Esta tabela é crucial para a gestão e monitorização das operações de mineração.

```
-- Inserção de dados na tabela Terreno
INSERT INTO Terreno (Minério_previsto, Minério_coletado) VALUES
    (1000, 800),
    (1500, 1200),
    (800, 600),
    (2000, 1800),
    (1200, 1000),
    (1800, 1600),
    (900, 700);
```

Figura 39: Inserção de dados na tabela Terreno com SQL.

A tabela “Trabalha” regista quais funcionários estão alocados a quais terrenos, estabelecendo uma ligação direta entre o ID do funcionário e o ID do terreno.

```
-- Inserção de dados na tabela Trabalha
INSERT INTO Trabalha (Funcionário_ID, Terreno_ID) VALUES
    (1, 1),
    (2, 2),
    (3, 3),
    (4, 4),
    (5, 5),
    (6, 6),
    (7, 7);
```

Figura 40: Inserção de dados na tabela Trabalha com SQL.

A tabela “Caso” regista os casos de investigação, incluindo informações sobre a data de abertura, estado atual, estimativa de roubo, data de encerramento e o ID do terreno associado ao caso. Estes dados são essenciais para o acompanhamento das investigações em curso.

```
-- Inserção de dados na tabela Caso
INSERT INTO Caso (Data_de_abertura, Estado, Estimativa_de_roubo, Data_de_encerramento,
Terreno_ID) VALUES
    ('2024-01-10', 'Aberto', 500, NULL, 1),
    ('2024-02-15', 'Fechado', 1000, '2024-05-05', 2),
    ('2024-03-20', 'Aberto', 700, NULL, 3),
    ('2024-04-25', 'Aberto', 800, NULL, 4),
    ('2024-05-01', 'Aberto', 300, NULL, 5),
    ('2024-05-08', 'Aberto', 900, NULL, 6),
    ('2024-05-10', 'Aberto', 600, NULL, 7);
```

Figura 41: Inserção de dados na tabela Caso com SQL.

Por fim, a tabela “Suspeito” regista os funcionários suspeitos em cada caso, incluindo o seu estado de envolvimento, nível de envolvimento e notas adicionais sobre a suspeita. Estes dados são fundamentais para a análise e resolução dos casos de investigação.

```
-- Inserção de dados na tabela Suspeito
INSERT INTO Suspeito (Funcionário_ID, Caso_ID, Estado, Envolvimento, Notas) VALUES
    (1, 1, 'Inocente', 1, 'Nenhuma informação adicional.'),
    (2, 2, 'Culpado', 9, 'Suspeito estava presente na cena do crime.'),
    (3, 3, 'Em investigação', 5, 'Algumas pistas indicam possível envolvimento.'),
    (4, 4, 'Em investigação', 3, 'Não há provas concretas.'),
    (5, 5, 'Inocente', 2, 'Sem evidências contra o suspeito.'),
    (6, 6, 'Culpado', 8, 'Fortes indícios de participação.'),
    (7, 7, 'Em investigação', 4, 'Possível conexão com outras investigações.');
```

Figura 42: Inserção de dados na tabela Suspeito com SQL.

Para iniciar o processo com Python, primeiro importamos as bibliotecas necessárias, entre elas, *mysql-connector* para estabelecer conexão com a base de dados, *faker* para criação de dados fictícios, *random* de forma a induzir aleatoriedade nas informações criada e, finalmente, *datetime* para facilitar a manipulação de datas em formatos pré-concebidos. Neste exemplo, solicita-se ao utilizador do programa que insira as suas credenciais e procede-se ao estabelecimento da ligação ao sistema e definição do cursor para execução de comandos SQL. Por último, inicializa-se o *faker* para criar dados em “Português de Portugal”.

```

import mysql.connector
from faker import Faker
import random
from datetime import timedelta

# Solicita ao utilizador o nome de utilizador e a senha para se conectar ao SQL
user = input("Insira o nome de utilizador da base de dados SQL: ")
password = input("Insira a senha da base de dados SQL: ")

# Estabelece a conexão com a base de dados SQL
conn = mysql.connector.connect(
    host="localhost",
    user=user,
    password=password,
    database="Lusium"
)

# Verifica se a conexão foi bem-sucedida e imprime a versão do servidor
if conn.is_connected():
    print("Conectado ao SQL Server versão ", conn.get_server_info())

# Cria um cursor para executar comandos SQL
cursor = conn.cursor()

# Instancia o Faker configurado para gerar dados em português de Portugal
fake = Faker('pt_PT')

```

Figura 43: Importação de bibliotecas e conexão à base de dados.

Inicialmente, utiliza-se uma lista com as funções previstas e, em seguida, inserimos cada uma dessas funções na tabela “Função”. Através de um ciclo *for*, itera-se sobre a lista e o método *cursor.execute()* para executar a inserção, passando o identificador e a designação de cada função.

```

# Lista de funções para inserir na tabela Função
funcoes = ['Operacional', 'Detetive', 'Representante']

# Insere as 3 funções na tabela Função
for i, funcao in enumerate(funcoes, start=1):
    cursor.execute("INSERT INTO Função (Função_ID, Designação) VALUES (%s, %s)", (i, funcao))

```

Figura 44: Inserção de dados na tabela Função com Python.

De seguida, geram-se e inserem-se dados fictícios para 100 funcionários na tabela “Funcionário”. Utiliza-se a biblioteca *fake* para gerar nomes, datas de nascimento, NIFs e URLs de fotografias, e a biblioteca *random* para definir salários e IDs de função. Cada funcionário é inserido na tabela através de um ciclo *for* que utiliza o método *cursor.execute()*.


```
# Gera e insere dados fictícios para 100 funcionários na tabela Funcionário
for i in range(1, 101):
    nome = fake.name()
    data_nascimento = fake.date_of_birth(minimum_age=18, maximum_age=65)
    salario = random.randint(1000, 5000)
    nif = fake.random_number(digits=9)
    fotografia = fake.image_url()
    funcao_id = random.randint(1, len(funcoes))
    cursor.execute("INSERT INTO Funcionário (Funcionário_ID, Nome, Data_de_nascimento,
    Salário, NIF, Fotografia, Função_ID) VALUES (%s, %s, %s, %s, %s, %s, %s)",
    (i, nome, data_nascimento, salario, nif, fotografia, funcao_id))
```

Figura 45: Inserção de dados na tabela Funcionário com Python.

O bloco de código seguinte é responsável por inserir 100 números de telemóvel na tabela “Número_de_tele móvel”. Geram-se números sequenciais a partir de 900000000 e associa-se cada número a um funcionário. Utiliza-se um ciclo *for* para a inserção sequencial dos dados na tabela com o método *cursor.execute()*.

```
# Gera e insere 100 números de telefone sequenciais na tabela Número_de_tele móvel
for i in range(1, 101):
    num_telefone = 900000000 + i
    funcionario_id = i
    cursor.execute("INSERT INTO Número_de_tele móvel (Número_de_tele móvel_ID,
    Funcionário_ID) VALUES (%s, %s)", (num_telefone, funcionario_id))
```

Figura 46: Inserção de dados na tabela Número_de_tele móvel com Python.

Neste bloco de código, inserem-se 50 registos na tabela “Gere”. Estes registos representam relações de gestão entre funcionários. Para cada gestor, escolhe-se aleatoriamente um funcionário que ele gere, utilizando *random.randint* para determinar o ID do funcionário gerido. As inserções são realizadas num ciclo *for*.

```
# Gera e insere 50 dados de gestão aleatória na tabela Gere
for i in range(1, 51):
    funcionario_gestor_id = i
    funcionario_id = random.randint(1, 100)
    cursor.execute("INSERT INTO Gere (Funcionário_Gestor_ID, Funcionário_ID) VALUES
    (%s, %s)", (funcionario_gestor_id, funcionario_id))
```

Figura 47: Inserção de dados na tabela Gere com Python.

Prossegue-se com a geração e inserção de dados para 20 terrenos na tabela “Terreno”. Cada terreno tem um valor previsto e coletado de minério, ambos gerados aleatoriamente. Utiliza-se um ciclo *for* e o método *cursor.execute()* para realizar as inserções na tabela.

```
# Gera e insere dados fictícios para 20 terrenos na tabela Terreno
for i in range(1, 21):
    minério_previsto = random.randint(1000, 10000)
    minério_coletado = random.randint(0, minério_previsto)
    cursor.execute("INSERT INTO Terreno (Terreno_ID, Minério_previsto,
    Minério_coletado) VALUES (%s, %s, %s)", (i, minério_previsto, minério_coletado))
```

Figura 48: Inserção de dados na tabela Terreno com Python.

Na inserção de dados de trabalho para 100 funcionários na tabela “Trabalha”, cada funcionário é associado aleatoriamente a um terreno. Utiliza-se um ciclo *for* e a função *random.randint* para selecionar o ID do terreno. As inserções são feitas novamente com o método *cursor.execute()*.

```
# Gera e insere 100 dados de trabalho na tabela Trabalha
for i in range(1, 101):
    funcionario_id = i
    terreno_id = random.randint(1, 20)
    cursor.execute("INSERT INTO Trabalha (Funcionário_ID, Terreno_ID) VALUES (%s, %s)",
        (funcionario_id, terreno_id))
```

Figura 49: Inserção de dados na tabela Trabalha com Python.

No presente bloco de código, geram-se e inserem-se dados fictícios para 50 casos na tabela “Caso”. Geram-se datas de abertura e encerramento, estados dos casos, estimativas de roubo e associam-se cada caso a um terreno, aplicando as estratégias já anteriormente descritas.

```
# Gera e insere dados fictícios para 50 casos na tabela Caso
for i in range(1, 51):
    data_abertura = fake.date_between(start_date='-1y', end_date='today')
    estado = random.choice(['Aberto', 'Fechado'])
    estimativa_roubo = random.randint(1000, 100000)
    data_encerramento = data_abertura + timedelta(days=random.randint(10, 365)) if
    estado == 'Fechado' else None
    terreno_id = random.randint(1, 20)
    cursor.execute("INSERT INTO Caso (Caso_ID, Data_de_abertura, Estado,
        Estimativa_de_roubo, Data_de_encerramento, Terreno_ID) VALUES (%s, %s, %s, %s, %s,
        %s)", (i, data_abertura, estado, estimativa_roubo, data_encerramento, terreno_id))
```

Figura 50: Inserção de dados na tabela Caso com Python.

Neste bloco de código, inserem-se dados fictícios para 100 suspeitos na tabela “Suspeito”. Cada suspeito é associado a um funcionário e a um caso, com estados, níveis de envolvimento e notas gerados aleatoriamente.

```
# Gera e insere dados fictícios de 100 suspeitos na tabela Suspeito
for i in range(1, 101):
    funcionario_id = i
    caso_id = random.randint(1, 50)
    estado = random.choice(['Inocente', 'Em investigação', 'Culpado'])
    envolvimento = random.randint(1, 10)
    notas = fake.text()
    cursor.execute("INSERT INTO Suspeito (Funcionário_ID, Caso_ID, Estado, Envolvimento,
        Notas) VALUES (%s, %s, %s, %s, %s)", (funcionario_id, caso_id, estado, envolvimento,
        notas))
```

Figura 51: Inserção de dados na tabela Suspeito com Python.

Por fim, no último bloco de código, imprime-se uma mensagem de sucesso após a inserção correta dos dados. Em seguida, confirmam-se todas as transações pendentes na base de dados ao utilizar *conn.commit()* e fecha-se a conexão com a base de dados ao utilizador *conn.close()*.

```
# Imprime a mensagem de sucesso
print("Dados inseridos com sucesso!")

# Confirma todas as transações e fecha a conexão com a base de dados
conn.commit()
conn.close()
```

Figura 52: Finalização das transações e conexão com a base de dados.

As duas abordagens complementam-se eficazmente no povoamento da base de dados. A abordagem SQL, com menos registos, facilita a exemplificação de operações mais simples, enquanto a utilização de Python permitiu um povoamento extensivo e variado, essencial para testes avançados e validação de queries complexas. Desta forma, assegura-se um ambiente de desenvolvimento robusto e versátil.

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

De forma a perceber o espaço que a base de dados pode vir a ocupar, calculamos inicialmente o mesmo quando existe apenas um registo por tabela e, numa fase final, estimou-se o espaço total necessário para armazenar toda a informação contida na base de dados, assim como a informação que surgirá com um determinado crescimento anual esperado.

Assim, representa-se, a seguir, uma tabela que evidencia os tipos de dados usados no sistema, acompanhados pelos seus tamanhos respetivos em bytes, de acordo com o manual de referência do MySQL 8.0 .

Tipo de dados	Tamanho (bytes)
INT	4
DATE	3
TEXT	L + 2 bytes, onde $L < 2^{16}$
VARCHAR(M)	L + 1 bytes caso a coluna necessite 0 – 255 bytes
ENUM('valor1','valor2',...)	1 ou 2 bytes, dependendo no número de valores enumerados (máximo de 65,535 valores)

Tabela 19: Tipos de dados utilizados e o seu respetivo tamanho em bytes.

Seguidamente, representam-se os espaços ocupados por cada tabela, em bytes, para quando existe apenas um registo na mesma.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Funcionário	"Funcionário_ID"	INT	4
	"Nome"	VARCHAR(75)	76
	"Data_de_nascimento"	DATE	3
	"Salário"	INT	4
	"NIF"	VARCHAR(10)	11
	"Fotografia"	VARCHAR(150)	151
	"Função_ID"	INT	4
Total	-	-	253

Tabela 20: Espaço ocupado pela tabela Funcionário com um registo.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Número de Telemóvel	"Número_de_Telemóvel"	INT	4
	"Funcionário_ID"	INT	4
Total	-	-	8

Tabela 21: Espaço ocupado pela tabela Número de Telemóvel com um registo.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Função	"Função_ID"	INT	4
	"Designação"	ENUM("Representante", "Detetive", "Operacional")	1
Total	-	-	5

Tabela 22: Espaço ocupado pela tabela Função com um registo.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Gere	"Funcionário_ID"	INT	4
	"Funcionário_Gestor_ID"	INT	4
Total	-	-	8

Tabela 23: Espaço ocupado pela tabela Gere com um registo.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Terreno	"Terreno_ID"	INT	4
	"Minério_previsto"	INT	4
	"Minério_coletado"	INT	4
Total	-	-	12

Tabela 24: Espaço ocupado pela tabela Terreno com um registo.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Caso	"Caso_ID"	INT	4
	"Data_de_abertura"	DATE	3
	"Estado"	ENUM("Aberto", "Fechado")	1
	"Estimativa_de_roubo"	INT	4
	"Data_de_encerramento"	DATE	3
	"Terreno_ID"	INT	4
Total	-	-	19

Tabela 25: Espaço ocupado pela tabela Caso com um registo.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Trabalha	"Funcionário_ID"	INT	4
	"Terreno_ID"	INT	4
Total	-	-	8

Tabela 26: Espaço ocupado pela tabela Trabalha com um registo.

Tabela	Nome da coluna	Tipo de dado	Tamanho (bytes)
Suspeito	"Funcionário_ID"	INT	4
	"Caso_ID"	INT	4
	"Estado"	ENUM("Inocente", "Em Investigação", "Culpado")	1
	"Envolvimento"	INT	4
	"Notas"	TEXT(256)	258
Total	-	-	271

Tabela 27: Espaço ocupado pela tabela Suspeito com um registo.

Com esta configuração, o espaço total ocupado pelas tabelas, preenchidas com um registo cada, seria igual a 584 bytes.

Após o povoamento inicial, a base de dados conta com 100 funcionários registados, dos quais 3 são representantes e 10 são detetives. Cada funcionário tem, em média, 2 números de telemóvel e, geralmente, trabalha em 2 terrenos. Neste caso, existem 11 terrenos distintos e um total de 13 casos ativos, dos quais são suspeitos, em média, 18 funcionários. Para além disso, cada representante é responsável por gerir uma parte dos restantes 97 funcionários, neste caso, um dos representantes gere os 10 detetives e os outros dois gerem, respetivamente, 40 e 47 funcionários operacionais. Deste modo, é possível calcular o espaço total ocupado pelas tabelas na primeira implementação do sistema de gestão de base de dados.

Tabela	Quantidade de registos	Tamanho da tabela (bytes)
Funcionário	100	25300
Número de Telemóvel	200	1600
Função	100	500
Gere	97	776
Terreno	11	132
Caso	13	247
Trabalha	200	1600
Suspeito	234	63414
Total	-	93569

Tabela 28: Espaço total ocupado pelas tabelas com o povoamento inicial.

Com base nas estatísticas anuais fornecidas pela empresa *Lusium*, é esperado que o número de funcionários e o número de terrenos aumente em 20% e 27%, respetivamente. Por outro lado, é expectável que

surjam, em média, 7 novos casos por ano. Com isto, prevemos que no primeiro ano as tabelas vão ter, respetivamente, as seguintes variações:

Tabela	Quantidade de registos	Diferença da tabela (bytes)
Funcionário	+20	+5060
Número de Telemóvel	+40	+320
Função	+20	+100
Gere	+20	+160
Terreno	+3	+36
Caso	+7	+133
Trabalha	+40	+320
Suspeito	+119	+32249
Total	-	+38378

Tabela 29: Variações do espaço ocupado pelas tabelas após um ano.

Como foi evidenciado nas tabelas, o espaço necessário mínimo para a primeira implementação da base de dados seria de **93.6 KBytes**, com um crescimento anual de **41%**, em que no primeiro ano é traduzido por um aumento de **38.4 KBytes**.

5.5. Definição e caracterização de vistas de utilização em SQL

Nesta secção, apresenta-se a definição de um conjunto de vistas essenciais para uma utilização eficiente da base de dados implementada. As vistas foram criadas para fornecer informações relevantes sobre os funcionários e suas características, bem como sobre casos e terrenos, facilitando o trabalho dos detetives e representantes. Para a caracterização destas vistas foram usadas um conjunto de seleções e junções.

A primeira vista cria uma tabela virtual que contém informações vitais para a identificação de um funcionário, incluindo o nome e o número de telemóvel, além de dados cruciais para uma compreensão completa dos casos associados a cada funcionário.

```
-- View das informações dos funcionários e dos casos associados
CREATE VIEW FuncionariosEmCasos AS
SELECT
    Funcionário.Nome,
    Número_de_tele móvel.Número_de_tele móvel_ID,
    Caso.Caso_ID,
    Caso.Data_de_abertura,
    Caso.Estado,
    Caso.Estimativa_de_roubo,
    Caso.Data_de_encerramento,
    Caso.Terreno_ID
FROM
    Funcionário
INNER JOIN
    Suspeito ON Funcionário.Funcionário_ID = Suspeito.Funcionário_ID
INNER JOIN
    Caso ON Suspeito.Caso_ID = Caso.Caso_ID
LEFT JOIN
    Número_de_tele móvel ON Funcionário.Funcionário_ID =
    Número_de_tele móvel.Funcionário_ID;
```

Figura 53: Vista que demonstra todos os funcionários e os seus casos associados.

A segunda vista foca-se na utilização pelos representantes, permitindo-lhes acessar de forma exclusiva as informações de identificação de um funcionário, entre elas, o nome, o número de telemóvel, função desempenhada e através de uma função, a idade, assim como os dados relativos aos terrenos aos quais cada funcionário está associado.

```
-- View das informações dos funcionários e dos terrenos associados
CREATE VIEW FuncionariosEmTerrenos AS
SELECT
    Funcionário.Nome,
    Função.Designação,
    Número_de_tele móvel.Número_de_tele móvel_ID,
    CalcularIdade(Funcionário.Data_de_nascimento) AS Idade,
    Funcionário.Salário,
    Terreno.Terreno_ID,
    Terreno.Minério_previsto,
    Terreno.Minério_coletado
FROM
    Funcionário
INNER JOIN
    Trabalha ON Funcionário.Funcionário_ID = Trabalha.Funcionário_ID
INNER JOIN
    Terreno ON Trabalha.Terreno_ID = Terreno.Terreno_ID
INNER JOIN
    Função ON Funcionário.Função_ID = Função.Função_ID
LEFT JOIN
    Número_de_tele móvel ON Funcionário.Funcionário_ID =
    Número_de_tele móvel.Funcionário_ID;
```

Figura 54: Vista que demonstra todos os funcionários e os seus casos associados.

Segue abaixo, um exemplo de formato de utilização de uma das vistas criadas:

```
-- Exemplo de utilização da vista FuncionariosEmTerrenos
SELECT *
FROM FuncionariosEmTerrenos;
```

Figura 55: Exemplo de utilização da vista “FuncionariosEmTerrenos”.

5.6. Tradução das interrogações do utilizador para SQL

Nesta secção, apresenta-se uma análise detalhada das consultas SQL, convertidas dos requisitos de manipulação, utilizadas para extrair informações relevantes da base de dados da *Lusium*. Cada consulta aos dados foi cuidadosamente projetada para fornecer informações específicos sobre os dados relacionados a funcionários, terrenos e casos. Ao examinar cada interrogação, é possível entender melhor como os dados são manipulados e analisados de forma a auxiliar na investigação e tomada de decisões.

No início do ficheiro, declaram-se três variáveis para permitir indicar qual o funcionário, caso ou terreno a ser utilizado na resolução das consultas, se necessário. Neste exemplo, propõe-se a utilização dos IDs 3, 3 e 1, para o funcionário, o caso e o terreno, respetivamente.

```
-- Definir variáveis para utilizar nas queries
SET @Funcionario_ID = 3;
SET @Caso_ID = 3;
SET @Terreno_ID = 1;
```

Figura 56: Declaração dos IDs para utilização nas consultas.

A primeira consulta calcula o prejuízo de um terreno. Para isso, subtrai a quantidade de minério coletado da quantidade de minério previsto.

```
-- Listar o prejuízo de um terreno
SELECT
    Terreno_ID,
    (Minério_previsto - Minério_coletado) AS Prejuízo
FROM
    Terreno
WHERE Terreno_ID = @Terreno_ID;
```

Figura 57: Listar o prejuízo de um terreno com SQL.

A segunda consulta verifica quando um funcionário se tornou suspeito de um determinado caso. Ela une as tabelas “Suspeito” e “Caso” para obter a data de abertura do caso correspondente ao funcionário e caso especificados.

```
-- Ver quando é que um funcionário se tornou suspeito de um determinado caso
SELECT
    s.Funcionário_ID,
    s.Caso_ID,
    c.Data_de_abertura
FROM
    Suspeito s
INNER JOIN
    Caso c ON s.Caso_ID = c.Caso_ID
WHERE
    s.Funcionário_ID = @Funcionário_ID
    AND s.Caso_ID = @Caso_ID;
```

Figura 58: Ver quando é que um funcionário se tornou suspeito de um determinado caso com SQL.

A terceira consulta lista todos os suspeitos de um determinado caso. Une as tabelas “Suspeito” e “Funcionário” para exibir informações detalhadas sobre os suspeitos, incluindo nome, estado, envolvimento e notas.

```
-- Listar os suspeitos de um determinado caso
SELECT
    s.Funcionário_ID,
    f.Nome,
    s.Estado,
    s.Envolvimento,
    s.Notas
FROM
    Suspeito s
INNER JOIN
    Funcionário f ON s.Funcionário_ID = f.Funcionário_ID
WHERE
    s.Caso_ID = @Caso_ID;
```

Figura 59: Listar os suspeitos de um determinado caso com SQL.

A quarta consulta mostra a data do último caso associado a um determinado funcionário. Junta as tabelas “Caso” e “Suspeito” e ordena os resultados por data de abertura em ordem decrescente, limitando a um único resultado.


```
-- Ver a data do último caso de um determinado funcionário
SELECT
    c.Caso_ID,
    c.Data_de_abertura
FROM
    Caso c
INNER JOIN
    Suspeito s ON c.Caso_ID = s.Caso_ID
WHERE
    s.Funcionário_ID = @Funcionário_ID
ORDER BY
    c.Data_de_abertura DESC
LIMIT 1;
```

Figura 60: Ver a data do último caso de um determinado funcionário com SQL.

A quinta consulta lista todos os casos a que um determinado funcionário está associado. Une as tabelas “Caso” e “Suspeito” para exibir informações detalhadas sobre os casos, incluindo estado, estimativa de roubo, data de encerramento e ID do terreno.

```
-- Listar os casos a que um determinado funcionário está associado
SELECT
    c.Caso_ID,
    c.Data_de_abertura,
    c.Estado,
    c.Estimativa_de_roubo,
    c.Data_de_encerramento,
    c.Terreno_ID
FROM
    Caso c
INNER JOIN
    Suspeito s ON c.Caso_ID = s.Caso_ID
WHERE
    s.Funcionário_ID = @Funcionário_ID;
```

Figura 61: Listar os casos a que um determinado funcionário está associado com SQL.

A sexta consulta determina o dia em que mais casos foram abertos. Agrupa os casos pela data de abertura, conta quantos casos foram abertos em cada data, e ordena em ordem decrescente, limitando a um único resultado, ou seja, o que contém mais casos.

```
-- Ver o dia em que mais casos foram abertos
SELECT
    Data_de_abertura,
    COUNT(*) AS Total_de_Casos
FROM
    Caso
GROUP BY
    Data_de_abertura
ORDER BY
    Total_de_Casos DESC
LIMIT 1;
```

Figura 62: Ver o dia em que mais casos foram abertos com SQL.

A última consulta lista os top 5 funcionários por quantidade de casos. Une as tabelas “Suspeito” e “Funcionário”, agrupa por funcionário, conta os casos associados a cada um e ordena em ordem decrescente, limitando os resultados aos cinco principais.

```
-- Listar os top 5 funcionários por quantidade de casos
SELECT
    s.Funcionário_ID,
    f.Nome,
    COUNT(s.Caso_ID) AS Total_de_Casos
FROM
    Suspeito s
INNER JOIN
    Funcionário f ON s.Funcionário_ID = f.Funcionário_ID
GROUP BY
    s.Funcionário_ID, f.Nome
ORDER BY
    Total_de_Casos DESC
LIMIT 5;
```

Figura 63: Listar os top 5 funcionários por quantidade de casos com SQL.

5.7. Indexação do Sistema de Dados

Nesta secção aborda-se a necessidade de aplicar uma metodologia de indexação à base de dados, com o objetivo de otimizar a eficiência das operações de pesquisa e manipulação de dados. O primeiro passo é a avaliação e previsão dos pontos de incidência resultantes da utilização da base de dados.

O processo de avaliação inicia-se com a identificação das tabelas e colunas que necessitarão de um índice. Após estudo detalhado da situação, concluiu-se que dado que os utilizadores principais da base de dados serão os detetives, prevê-se que as tabelas Caso e Suspeito sejam frequentemente acedidas, tornando assim os seus identificadores (chaves primárias e estrangeiras) pontos críticos de acesso.

Recorre-se à utilização de indexação por três motivos principais:

1. **Melhoria do desempenho das operações de consulta:** A indexação permite um acesso mais rápido aos registos, uma vez que os índices facilitam a localização eficiente dos dados. Isto é particularmente útil para operações de leitura, que são frequentes e exigem tempos de resposta rápidos.
2. **Redução da necessidade de leitura de dados repetitivos:** Com a utilização de índices, o sistema pode aceder diretamente aos elementos específicos de uma tabela sem necessidade de percorrer todos os registos. Isto reduz a carga de trabalho no sistema de armazenamento e melhora a eficiência geral.
3. **Facilitação da escalabilidade do sistema:** À medida que o número de registos aumenta, os índices são projetados para lidar com pesquisas simultâneas de forma eficiente, sem degradar o desempenho geral do sistema de gestão de base de dados. Isto é essencial para manter a performance do sistema em ambientes de alto tráfego e grande volume de dados.

Na situação da *Lusium*, após identificar os pontos de maior incidência, prossegue-se a criação dos índices nas chaves estrangeiras das duas tabelas com maior frequência. Note-se que não há necessidade de criar índices para chaves primárias, uma vez que naturalmente já são constituídas num índice no momento da sua criação.

```
-- Criação de um índice para a coluna Terreno_ID na tabela Caso
CREATE INDEX index_caso_terreno_id ON Caso(Terreno_ID);

-- Criação de um índice para a coluna Caso_ID na tabela Suspeito
CREATE INDEX idx_suspeito_caso_id ON Suspeito(Caso_ID);

-- Criação de um índice para a coluna Funcionário_ID na tabela Suspeito
CREATE INDEX idx_suspeito_funcionario_id ON Suspeito(Funcionário_ID);
```

Figura 64: Aplicação de indexação na base de dados.

5.8. Implementação de procedimentos, funções e gatilhos

Como forma de automatizar e simplificar processos dentro do contexto do sistema de gestão da base de dados, foram definidos alguns gatilhos, funções e procedimentos. De seguida, evidenciam-se os mesmos, assim como a explicação para a sua implementação.

A função “CalcularEstimativaDeRoubo” permite ao sistema calcular rapidamente quantos quilogramas de minério foram roubados de um terreno, recebendo, para o efeito, apenas o ID do terreno em questão. Desta forma, a função devolve a subtração do minério previsto pelo minério realmente coletado. Esta função revela-se bastante útil uma vez que simplifica todo o processo de seleção e cálculo envolvido neste contexto e é usada, por exemplo, pelo procedimento “CriarCasoETornarSuspeitos” (evidenciado posteriormente).

```
DELIMITER $$
CREATE FUNCTION CalcularEstimativaDeRoubo(p_Terreno_ID INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE v_Estimativa_De_Roubo INT;

    -- Obter o minério previsto e coletado para o terreno especificado
    SELECT (Minério_previsto - Minério_coletado) INTO v_Estimativa_De_Roubo
    FROM Terreno WHERE Terreno_ID = p_Terreno_ID;

    RETURN v_Estimativa_De_Roubo;
END $$
DELIMITER ;
```

Figura 65: Função que calcula a estimativa de roubo de um caso.

Para facilitar a visualização da idade dos utilizadores, decidimos criar a função “CalcularIdade”. Esta função calcula a idade atual a partir da data de nascimento fornecida. A função é utilizada na vista “Funcionario-sEmTerrenos” para apresentar a idade de um funcionário de forma clara.

```
DELIMITER $$
CREATE FUNCTION CalcularIdade(Data_de_nascimento DATE)
RETURNS INT DETERMINISTIC
BEGIN
    DECLARE idade INT;

    -- Calcula a idade considerando apenas o ano
    SET idade = YEAR(CURDATE()) - YEAR(Data_de_nascimento);

    -- Ajusta a idade se o aniversário ainda não ocorreu este ano
    IF (MONTH(CURDATE()) < MONTH(Data_de_nascimento)) OR
        (MONTH(CURDATE()) = MONTH(Data_de_nascimento) AND DAY(CURDATE()) <
        DAY(Data_de_nascimento)) THEN
        SET idade = idade - 1;
    END IF;

    RETURN idade;
END $$
DELIMITER ;
```

Figura 66: Função que calcula a idade através de uma data de nascimento.

O gatilho “AtualizarDataEncerramento” é responsável por atribuir uma data de encerramento a um determinado caso que tenha encerrado recentemente. Assim sendo, a partir das suas tabelas OLD e NEW, é possível perceber, registo a registo, se houve uma atualização do estado de um certo caso para “Fechado”

e, em caso afirmativo, atribui-se a data atual, usando a função CURDATE(), à data de encerramento do mesmo. Desta forma, este gatilho acaba por eliminar a necessidade da atribuição manual de uma data de encerramento a um determinado caso, sempre que o mesmo se dá como concluído. Isto promove uma maior abstração no contexto de manipulação de casos no que toca à interação dos utilizadores com a base de dados.

```
DELIMITER $$
CREATE TRIGGER AtualizarDataEncerramento
BEFORE UPDATE ON Caso
FOR EACH ROW
BEGIN
    IF NEW.Estado = 'Fechado' AND OLD.Estado != 'Fechado' THEN
        SET NEW.Data_de_encerramento = CURDATE();
    END IF;
END $$
DELIMITER ;
```

Figura 67: Gatilho responsável por atualizar a data de encerramento de um caso.

O procedimento “CriarCasoETornarSuspeitos” tem como propósito automatizar o processo de criação de um caso relativo a um terreno e aos seus trabalhadores. O mesmo recebe o ID de um terreno e abre um caso, marcando todos os funcionários do mesmo como suspeitos. Para tal, faz uso da tabela trabalha, da qual consegue recolher todos os funcionários que lá trabalham, e cria um novo suspeito, por cada funcionário, com os atributos estado, envolvimento e notas já pré-definidos.

```

DELIMITER $$
CREATE PROCEDURE CriarCasoETornarSuspeitos(
    IN p_Terreno_ID INT,
    IN p_Data_de_abertura DATE
)
BEGIN
    DECLARE v_Caso_ID INT;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        -- Em caso de erro, reverter todas as operações
        ROLLBACK;
    END;

    START TRANSACTION;

    -- Insere um novo caso
    INSERT INTO Caso (Data_de_abertura, Estado, Estimativa_de_roubo,
Data_de_encerramento, Terreno_ID)
VALUES (p_Data_de_abertura, 'Aberto', CalcularEstimativaDeRoubo(p_Terreno_ID),
NULL, p_Terreno_ID);

    -- Obter o último ID do caso inserido
    SET v_Caso_ID = LAST_INSERT_ID();

    -- Seleciona todos os funcionários que trabalham no terreno especificado
    INSERT INTO Suspeito (Funcionário_ID, Caso_ID, Estado, Envolvimento, Notas)
SELECT t.Funcionário_ID, v_Caso_ID, 'Em Investigação', 3, 'Funcionário estava
presente no terreno no dia do roubo'
FROM Trabalha t
WHERE t.Terreno_ID = p_Terreno_ID;

    COMMIT;
END $$
DELIMITER ;

```

Figura 68: Procedimento que automatiza a atribuição de suspeitos a um novo caso.

6. Conclusões e Trabalho Futuro

Com a implementação do sistema de gestão de base de dados, tanto Vizela como a empresa *Lusium* aprimoraram a sua segurança em diversos aspetos, trazendo um aumento exponencial tanto nos lucros como na qualidade de vida de todos os envolvidos. A adoção deste sistema representa um passo fundamental para garantir a integridade, confidencialidade e disponibilidade de informações críticas para a resolução de casos.

Durante a execução deste projeto, tivemos o auxílio de um diagrama de Gantt para delimitar prazos a cumprir durante o seu desenvolvimento. Pontualmente, demonstramos alguma inexperiência na demarcação do tempo necessário para concluir as distintas fases do projeto. Estas inconveniências causaram algumas irregularidades no diagrama de Gantt, porém foram superadas, sem causar muitos impedimentos ao cumprimento da data final de entrega:

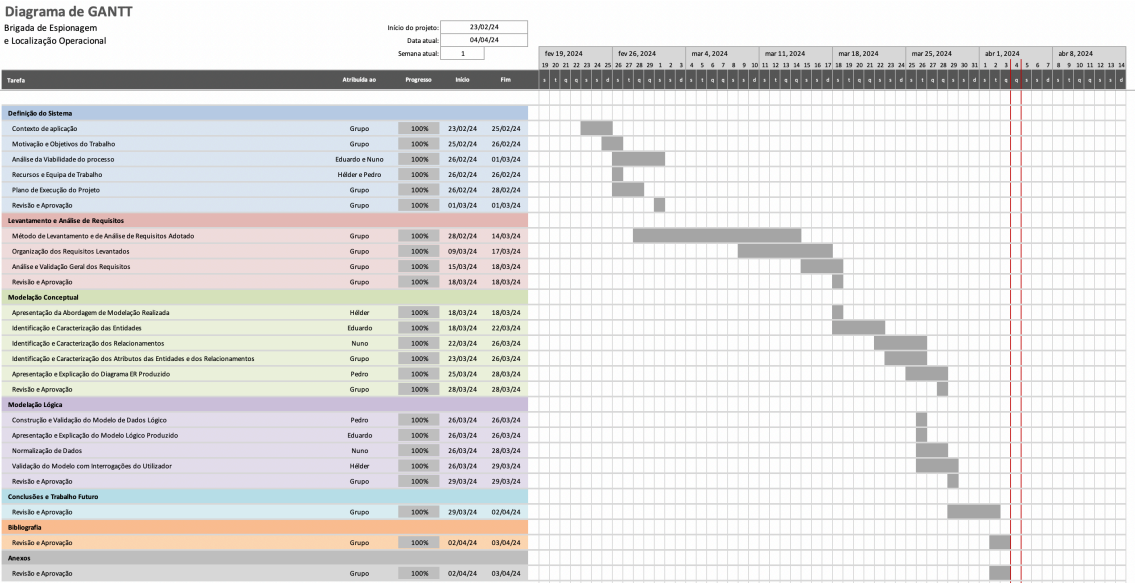


Figura 69: Ilustração do diagrama de Gantt real da primeira fase.

De forma semelhante à execução da primeira fase do projeto, foi criado um diagrama de Gantt para planejar e estipular prazos para a conclusão da implementação física. Com a nossa pequena experiência em planeamento, conseguimos produzir um diagrama de Gantt realista e exequível:

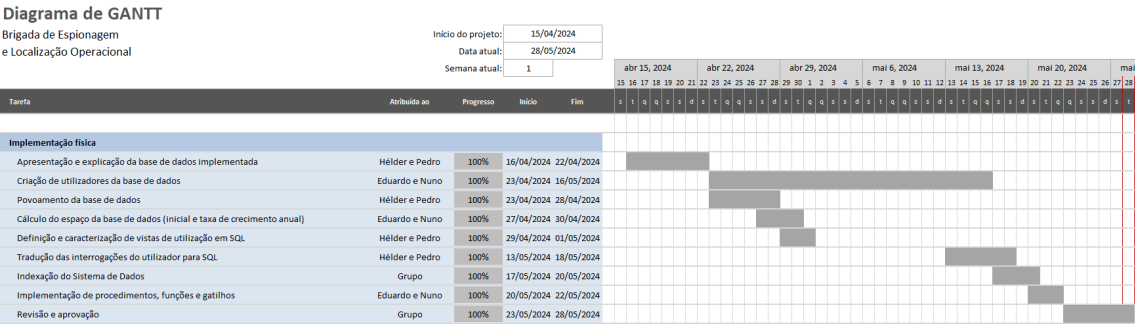


Figura 70: Ilustração do diagrama de Gantt real da segunda fase.

Manteremos continuamente contacto com a *Lusium* para prestar serviços de manutenção e auxílio com eventuais problemas que possam surgir na base de dados. A formulação íntegra e consolidada do caso de estudo, apresentado no decorrer da produção da solução proposta, permitiu a solidificação da composição de todo o sistema, que teve por base o estrito cumprimento de todas as etapas do “ciclo de vida de uma base de dados”.

Referências

- Connolly, T. M., & Begg, C. E. (2005). Database systems: a practical approach to design, implementation, and management. Pearson Education.
- MySQL 8.0 Reference Manual. MySQL. (2024). <https://dev.mysql.com/doc/refman/8.0/en/>

Lista de Siglas e Acrónimos

B.E.L.O. Brigada de Espionagem e Localização Operacional

SQL Structured Query Language

Anexos

Anexo 1: Ata da primeira reunião

Data de realização: 1 de março de 2024

Hora de início: 09:37h

Hora de término: 11:56h

Participantes

- Hélder Gomes - Membro da “Quatro em Linha”
- Pedro Pereira - Membro da “Quatro em Linha”
- Ana Sofia - Representante da empresa
- Maria Eduarda - Representante da empresa
- João Guilherme - Representante da empresa

Contextualização

Ao primeiro dia do mês de março de dois mil e vinte e quatro, com início às nove horas e trinta e sete minutos da manhã, na sala de reuniões número doze, localizada na sede da Lusium, realizou-se uma primeira reunião com os representantes da empresa, Ana Sofia, Maria Eduarda e João Guilherme, com a principal meta de apresentar uma solução para a questão-problema apresentada à empresa “Quatro em Linha”.

Levantamentos

Iniciaram-se os trabalhos previstos com a apresentação dos representantes de ambas as valências, Hélder Gomes começou por fazer uma **descrição da história da empresa**, como as motivações de fundação e os princípios laborais inerentes. Após a intervenção, Pedro Pereira enumerou as **competências e habilidades** dos intervenientes associados aos vários projetos que a empresa já desenvolveu até ao momento com sucesso.

No decorrer da ordem de trabalhos, Maria Eduarda, na qualidade de representante da Lusium, procedeu à **descrição e qualificação da questão-problema**, principiando com o enunciar dos inúmeros casos de furtos decorrentes da criação da empresa de minério e triunfo consequente. João Guilherme completou a intervenção de Maria Eduarda com uma preocupação pessoal sobre a empresa, mas principalmente sobre Vizela, a sua cidade natal.

Após as apresentações iniciais e discussão sobre os problemas enfrentados pela empresa e pela comunidade local, Hélder Gomes e Pedro Pereira **propuseram a implementação de um sistema de gestão de base de dados** como forma de controlar os novos furtos. A ideia foi bem recebida pelos participantes, que demonstraram entusiasmo com a proposta.

Ana Sofia agradeceu as contribuições de todos os presentes e destacou a **importância do diálogo** e da **colaboração** para encontrar soluções eficazes. Em resposta à proposta, foi decidido marcar outras reuniões para avaliar e desenvolver a viabilidade da ideia.

Em adição, ficou acordado formar um grupo de trabalho dedicado a abordar os desafios identificados e a elaborar estratégias para **melhorar a segurança e o bem-estar da comunidade**. Hélder Gomes e Pedro Pereira enfatizaram a eficácia de tal sistema de monitorização e registo para identificar padrões e **agir preventivamente contra os furtos**.

A proposta foi recebida com entusiasmo pelos participantes, que concordaram em participar ativamente no grupo de trabalho. Ficou agendada uma próxima reunião, com data prevista de 4 de março de 2024,

às nove horas e trinta minutos da manhã, para dar continuidade às discussões e recolher as medidas e requisitos para a base de dados.

Para sumarizar, a reunião foi encerrada com a promessa de um **esforço conjunto** para enfrentar os desafios e promover o **crescimento saudável** da empresa e da comunidade.

Assinaturas

Helder Gomes

Ana Sofia

João Guilherme

Pedro Pereira

Maria Eduarda

Anexo 2: Ata da segunda reunião

Data de realização: 4 de março de 2024

Hora de início: 09:30h

Hora de término: 12:00h

Contextualização

Ao quarto dia do mês de março de dois mil e vinte e quatro, com início às nove horas e trinta minutos da manhã, na sala de reuniões número cinco, localizada na sede da Lusium, realizou-se uma segunda reunião com os representantes da empresa, Ana Sofia, Maria Eduarda e João Guilherme, com o objetivo de idealizar os requisitos mandatórios, relacionados com a gestão dos funcionários e dos terrenos onde trabalham, para o correto funcionamento do sistema de gestão de base de dados.

Participantes

- Eduardo Faria - Membro da “Quatro em Linha”
- Nuno Silva - Membro da “Quatro em Linha”
- Ana Sofia - Representante da empresa
- Maria Eduarda - Representante da empresa
- João Guilherme - Representante da empresa

Tópicos abordados

No decorrer da reunião, foram esclarecidos quais os aspetos a ter em conta no domínio da gestão dos funcionários da empresa e dos terrenos de extração de minérios. Estes aspetos revelam-se cruciais e são dependentes de consistência e de uma boa ponderação.

Levantamentos

Com o desenrolar da reunião, foi possível recolher diversas propostas e sugestões relacionadas com o modelo de funcionamento dos funcionários e terrenos.

Maria Eduarda, numa fase inicial, referiu que cada **funcionário** deveria ser **identificado de forma exclusiva**, acrescentando ainda que o **nome, data de nascimento, função, salário, números de identificação e números de telemóvel** seriam características importantes a associar ao mesmo. Além disso, mencionou que deveriam ter uma **fotografia** associada, ao passo que João Guilherme manifestou a ideia de que a fotografia deveria ser algo **opcional** e nunca de carácter obrigatório.

A partir de uma primeira intervenção por parte da representante Ana Sofia, ficou definido que os funcionários deveriam ser designados para diferentes funções, como **operacional, detetive** ou **representante**, esclarecendo ainda que **nos terrenos** da empresa **trabalham vários operacionais** e que cada um destes pode estar atribuído a **mais do que um terreno**.

Maria Eduarda, numa outra intervenção, destacou que cada **terreno** da empresa deveria ser **identificado de forma única**, tornando possível a distinção entre cada um deles. Complementarmente, os representantes reforçaram a ideia de que são levantadas suspeitas num terreno sempre que a quantidade de minério

coletado não corresponde à quantidade prevista de minério a coletar no dia. Como resposta, Nuno Silva e Eduardo Faria propuseram a ideia de que um terreno deveria conter informações sobre o **minério previsto** e o **minério coletado**, ideia esta que foi diretamente aprovada pelos representantes. O minério previsto representaria a **quantidade estimada de minério** que se espera coletar **diariamente**, enquanto que o **minério coletado** refletiria a **quantidade real de minério coletada por dia**. Deste modo, todos os intervenientes chegaram à conclusão que seria importante monitorizar todos esses dados para, posteriormente, ser possível **calcular e listar o prejuízo** de cada terreno, o que ajudaria na tomada de decisões estratégicas e no planeamento futuro das operações.

Ana Sofia, Maria Eduarda e João Guilherme, em conformidade, salientaram o facto de serem os únicos **responsáveis pela gestão dos funcionários operacionais**, coordenando todas as suas atividades e garantindo eficiência operacional. Assim sendo, por unanimidade de todos os participantes, chegou-se à conclusão que deveria ser possível fornecer permissões aos **representantes** na base de dados, de modo a que estes **consigam criar novos funcionários e, adicionalmente, terrenos**, bem como **modificar as informações adjacentes aos mesmos**. Em contrapartida, João Guilherme enfatizou o facto de que os **funcionários operacionais não devem ter acesso direto às informações sensíveis** desta base de dados, de forma a manter a correta gestão da mesma e, consequentemente, da empresa.

Deste modo, após a formalização de todas estas condições, salientou-se a necessidade de agendar uma próxima reunião que garantisse, agora, a declaração das imprescindibilidades do modelo que abrange a gestão dos casos e dos suspeitos adjacentes. Esta nova reunião ficou marcada para o próximo dia 8 de março de 2024, às nove horas da manhã, da qual também fariam parte os detetives da B.E.L.O, visto serem os mais qualificados para descrever o modo como se deve correlacionar suspeitos e casos com funcionários e terrenos, perante o problema em questão.

Assinaturas

Nuno Silva

Ana Sofia

João Guilherme

Eduardo Faria

Maria Eduarda

Anexo 3: Ata da terceira reunião

Data de realização: 8 de março de 2024

Hora de início: 09:04h

Hora de término: 11:29h

Contextualização

Ao oitavo dia do mês de março de dois mil e vinte e quatro, com início às nove horas e quatro minutos da manhã, na sala de reuniões número sete, localizada no departamento B.E.L.O da Lusium, realizou-se uma terceira reunião com os detetives da empresa e um dos representantes da mesma, Ana Sofia. Nesta reunião, Ana, para além de trazer consigo ideias capazes de completar medidas estabelecidas na reunião anterior, teria, agora também, o cargo de autorizar os pedidos e decisões dos detetives, quando se apresentassem admissíveis. Deste modo, a reunião seguiria com o objetivo de elaborar os requisitos relacionados com a gestão dos casos e suspeitos adjacentes, necessários para um adequado desenvolvimento e consequente funcionamento do sistema de gestão de base de dados.

Participantes

- Pedro Pereira - Membro da “Quatro em Linha”
- Hélder Gomes - Membro da “Quatro em Linha”
- Ana Sofia - Representante da empresa
- Filipe Sousa - Detetive da B.E.L.O.
- Sandra Silva - Detetive da B.E.L.O.
- João Silva - Detetive da B.E.L.O.

- Américo Costa - Detetive da B.E.L.O.
- Henrique Mendes - Detetive da B.E.L.O.
- Lara Machado - Detetive da B.E.L.O.

Tópicos abordados

Ao longo desta reunião, foram desenvolvidos vários pontos que têm por base todo o funcionamento da gestão dos casos com que lida a agência, bem como os suspeitos envolvidos nos mesmos. Estes pontos devem ser bem definidos e alvos de uma forte especificação, visando assegurar uma abordagem sólida que garanta a eficácia e integridade da base de dados.

Levantamentos

Com o decorrer da reunião, uma pluralidade de propostas e sugestões foram apresentadas, todas elas orientadas para o modelo de funcionamento dos casos e dos suspeitos relacionados com os mesmos.

O detetive Filipe Sousa fez a primeira intervenção na reunião, mencionando que um caso deveria ser **identificado de forma exclusiva** e, simultaneamente, incluir informações extra, como a sua **data de abertura, data de encerramento, se aplicável, o seu estado e a quantidade estimada de minério roubado**.

A detetive Lara Machado acabou por reforçar a ideia de que cada **caso** estaria **somente ligado a um terreno**, porém, cada **terreno** da empresa poderia **contar com vários casos**. De forma a complementar a intervenção de Lara Machado, Hélder Gomes e Pedro Pereira questionaram os detetives sobre de que forma é que um caso deveria ser **classificado** quanto ao seu **estado**, ao que Américo Costa respondeu que apenas seria necessário classificar o mesmo como **“aberto” ou “fechado”**.

Não deixando de parte o que foi referido na reunião anterior sobre o levantamento de suspeitas num terreno, a representante Ana Sofia incluiu agora que um **novo caso** deveria ser automaticamente **aberto** caso a **quantidade de minério coletada num terreno, num dado dia**, fosse **inferior à quantidade mínima prevista**. Consequentemente, **todos os funcionários** associados a esse terreno **tornar-se-iam suspeitos**, iniciando assim uma investigação para determinar qualquer possível atividade criminosa.

Todos os **detetives** concordaram, de forma unânime, que deveriam ser os únicos com permissão para **criar casos** e **alterar algumas informações** dos mesmos na base de dados, o que foi aceite pela representante presente. Ficou também confirmado que deveriam ser os únicos autorizados a **atribuir suspeitos a um caso e atualizar o estado desses suspeitos** conforme necessário.

Dentro do mesmo contexto, Sandra Silva achou por bem explicar que cada suspeito deveria ser bem caracterizado pelo seu **estado, nível de envolvimento e possíveis notas adicionais** sobre o mesmo. Esta aprimorou a sua intervenção esclarecendo que o **nível de envolvimento** deveria **variar de 1 a 10**, indicando o grau de participação do suspeito no caso, enquanto que o **estado** poderia variar entre **“inocente” ou “culpado”**, de forma a facilitar a análise de cada um deles.

Todos os intervenientes concordaram que o acompanhamento da atividade dos casos seria algo imprescindível e, para tal, os dois membros da “Quatro em Linha” propuseram a ideia de se poder **verificar quando é que um funcionário se tornou suspeito** de um determinado caso, **visualizar os suspeitos de um caso específico** e verificar a **data do último caso associado a um funcionário**, o que foi aprovado por todos os integrantes. Complementarmente, os detetives também consideraram relevante haver a possibilidade de se fazer uma **listagem dos casos associados a um determinado funcionário**, bem como uma verificação do **dia em que mais casos foram abertos**, o que poderia fornecer esclarecimentos valiosos para a gestão de segurança e operações da empresa.

Américo Costa, numa fase final, ainda salientou que deveria ser possível organizar os diferentes suspeitos com base na quantidade de casos a que se associam, visto ser um informação/dado que auxiliaria na priorização das investigações e na identificação de padrões de comportamentos suspeitos. Assim, de modo a simplificar o processo descrito, foi acordado que deveria haver a possibilidade de se fazer uma **listagem específica dos cinco suspeitos com mais casos** de roubo de minérios associados.

Ao terminar a 3ª reunião, ficou acordado por Ana Sofia, que participou em nome de todos os representantes, e por todos os detetives presentes que seria pertinente agendar uma próxima reunião, que contaria com a informação referente à experiência dos trabalhadores da Lusium através de um inquérito estatístico

de opinião. O propósito da mesma seria verificar a integridade de todas as ideias e sugestões recolhidas até ao momento, antes da implementação dos requisitos gerados pelas mesmas no sistema de gestão de base de dados.

Assinaturas

Hélder Gomes

Ana Sofia

Sandra Silva

Pedro Pereira

Filipe Sousa

João Silva

Américo Costa

Henrique Mendes

Lara Machado

Anexo 4: Ata da quarta reunião

Data de realização: 14 de março de 2024

Hora de início: 09:11h

Hora de término: 11:42h

Contextualização

Ao décimo quarto dia do mês de março de dois mil e vinte e quatro, com início às nove horas e onze minutos da manhã, na sala de reuniões número um, localizada no departamento B.E.L.O da Lusium, realizou-se uma quarta e última reunião com todos os detetives da empresa e todos os representantes da mesma, Ana Sofia, Maria Eduarda e João Guilherme. Nesta reunião, todos os requisitos anteriormente levantados foram reavaliados por todos os envolvidos com o intuito de os concluir.

Participantes

- Eduardo Faria - Membro da “Quatro em Linha”
- Hélder Gomes - Membro da “Quatro em Linha”
- Nuno Silva - Membro da “Quatro em Linha”
- Pedro Pereira - Membro da “Quatro em Linha”
- Ana Sofia - Representante da empresa
- Maria Eduarda - Representante da empresa
- João Guilherme - Representante da empresa
- Filipe Sousa - Detetive da B.E.L.O
- Sandra Silva - Detetive da B.E.L.O
- João Silva - Detetive da B.E.L.O
- Américo Costa - Detetive da B.E.L.O
- Henrique Mendes - Detetive da B.E.L.O
- Lara Machado - Detetive da B.E.L.O

Tópicos abordados

Durante o decorrer desta reunião, foram abordados todos os pontos anteriormente discutidos, para assim dar por concluída toda a base do funcionamento da base de dados. Estes pontos devem ser bem definidos e alvos de uma forte especificação, visando assegurar uma abordagem sólida que garanta a eficácia e integridade da base de dados.

Levantamentos Hélder Gomes deu início à reunião pedindo alguns esclarecimentos sobre requisitos levantados anteriormente. A primeira elucidação abordaria o que foi referido, inicialmente, por Maria Eduarda, ao pedir que cada funcionário seja caracterizado por **números de identificação**, ao qual foi respondido que, depois de uma reunião onde apenas estiveram presentes os representantes, decidiram alterar esse pedido para apenas ter em conta o **número de identificação fiscal (NIF)**, dado o âmbito do projeto. As restantes características mantinham-se inalteradas, pelo que tanto os membros da “Quatro em Linha”, como da agência B.E.L.O concordaram.

Eduardo Faria interpelou logo de seguida, pedindo aos detetives a confirmação sobre o funcionamento dos funcionários e como eles se relacionam com os terrenos, à qual Henrique Mendes e Lara Machado concordaram prontamente.

De seguida, Pedro Pereira descreveu todas as características de um terreno, finalizando com um apelo à melhoria da eficiência da base de dados, sendo este a mudança da informação de **minério previsto** para **minério mínimo previsto**, ao qual João Guilherme respondeu que seria uma alteração possível, apenas teriam de comunicar essa alteração aos técnicos que iriam avaliar os terrenos. As restantes características foram aprovadas sem nenhum acréscimo.

Quanto às permissões da base de dados, Nuno Silva expôs os pedidos feitos pelos representantes. Os detetives demonstraram compreensão e concordaram que essa logística é a ideal para a integridade dos dados.

Pedro Pereira comunicou as características de um caso, sendo que, depois de uma pequena discussão entre representantes e detetives, decidiram manter como anteriormente definido.

Maria Eduarda questionou se, caso o **minério coletado** fosse **igual** ao **minério mínimo** previsto, seria possível abrir um **caso para investigação**. Esse comentário proporcionou um momento de reflexão entre os presentes. Alguns expressaram a sua indignação de forma leve, mas a ponderação prevaleceu na decisão final. Ficou estabelecido que apenas o que for **coletado abaixo do mínimo estipulado será alvo de investigação**.

Sandra Silva solicitou uma modificação num requisito previamente estabelecido. Essa solicitação surgiu após uma breve reunião informal entre os membros da agência, na qual se chegou unanimemente à conclusão de que classificar o estado de um suspeito apenas como **“inocente”** ou **“culpado”** seria **insuficiente**. Propôs-se então a **adição de um terceiro estado, “em investigação”**. Todos os presentes concordaram com essa decisão.

No restante da reunião prevaleceu a **exposição dos requisitos** e a **aprovação** por parte de todos os presentes.

A reunião foi encerrada pelas onze horas e quarenta e dois minutos com todos os envolvidos a demonstrarem, de forma não verbal, a gratificação pelo início da implementação do sistema de gestão de base de dados.

Assinaturas

Helder Gomes

Pedro Pereira

Nuno Silva

Sandra Silva

Eduardo Faria

Lara Machado

Filipe Sousa

João Silva

Henrique Mendes

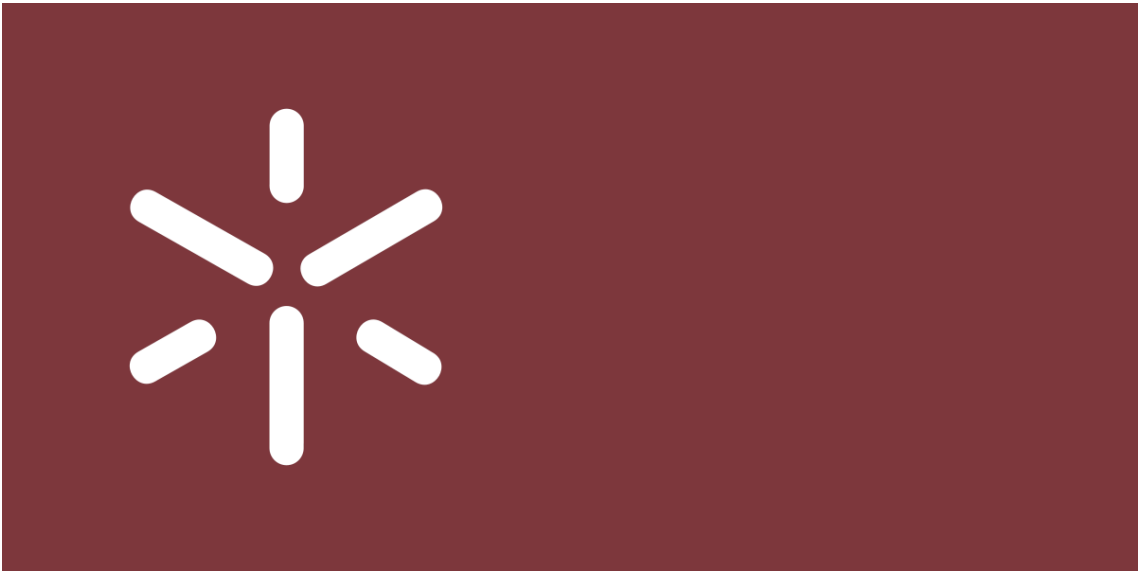
Américo Costa

Ana Sofia

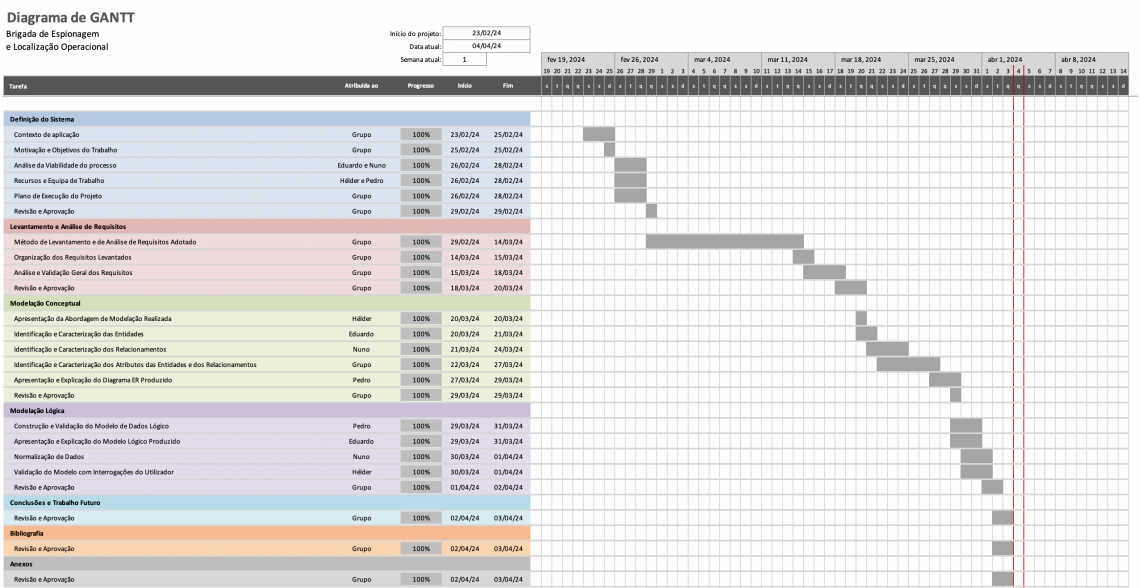
João Guilherme

Maria Eduarda

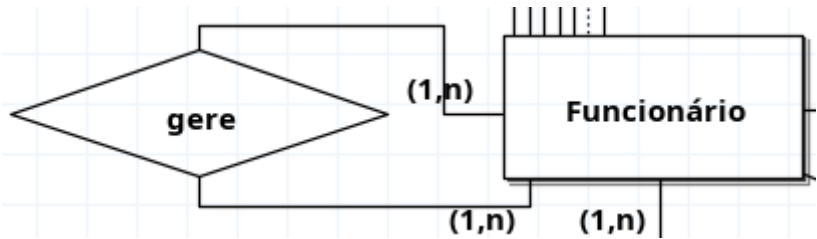
Anexo 5: Logo da Universidade do Minho



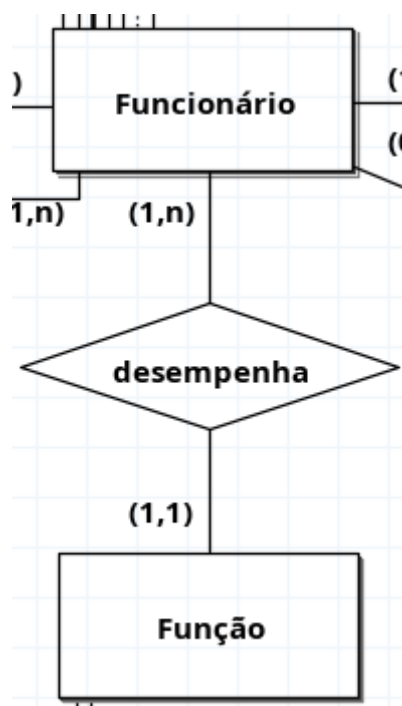
Anexo 6: Ilustração do diagrama de Gantt inicial.



Anexo 7: Ilustração do Relacionamento Funcionário - Funcionário.



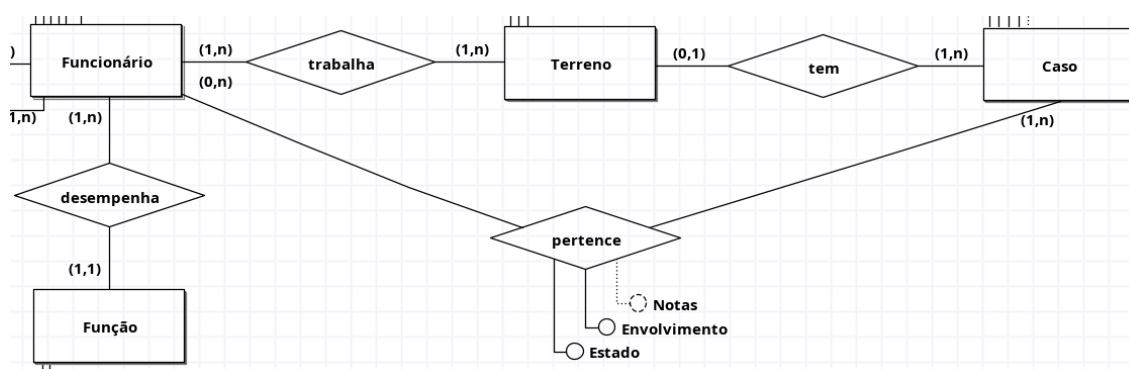
Anexo 8: Ilustração do Relacionamento Funcionário - Função.



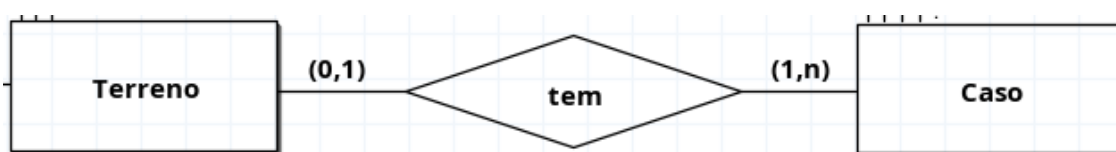
Anexo 9: Ilustração do Relacionamento Funcionário - Terreno.



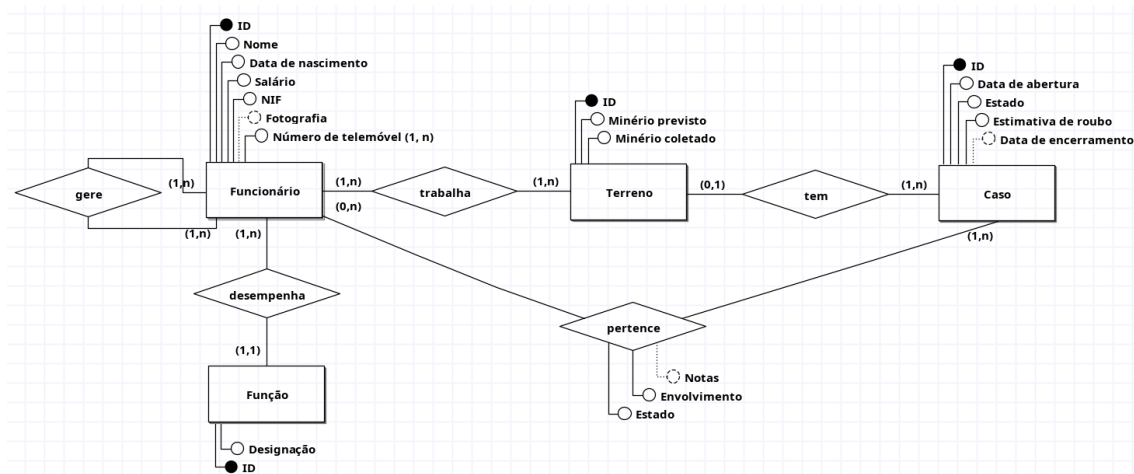
Anexo 10: Ilustração do Relacionamento Funcionário - Caso.



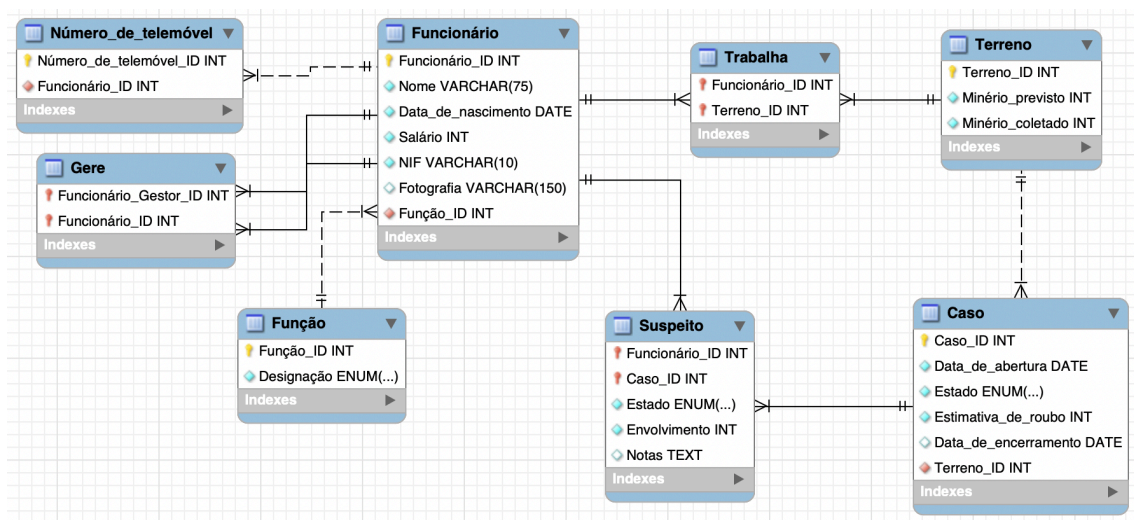
Anexo 11: Ilustração do Relacionamento Terreno - Caso.



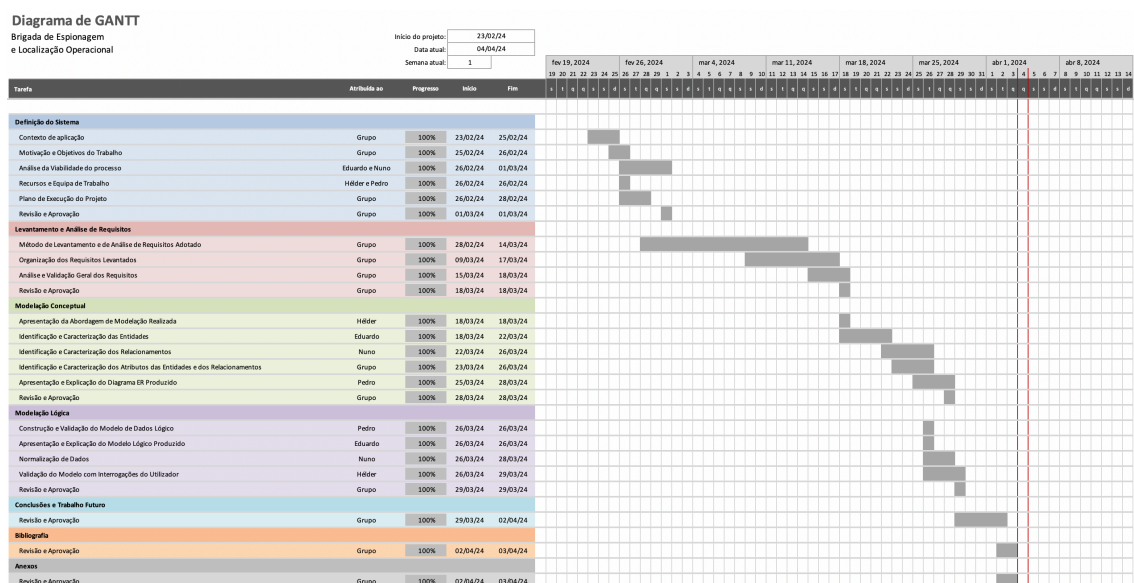
Anexo 12: Ilustração do diagrama ER produzido.



Anexo 13: Ilustração do modelo lógico produzido.



Anexo 14: Ilustração do diagrama de Gantt real.



Anexo 15: Ilustração do Inquérito Estatístico de Opinião.

Inquérito Estatístico de Opinião (Respostas)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Carimbo de data/hora	Nome	E-mail de contacto	Número de Identificação Fiscal	Data de nascimento	Número de anos	Salário líquido	O que se O que se O que se O que se O que se O que se O que se O que se	Num panoram
2024-04-07 17:40:19				09/10/1974	Menos de um ano	Mais de 1200	0 2 3 5 5 1 3	
2024-04-07 17:40:19				19/02/1999	Dois a quatro anos	901 a 1000	0 2 0 1 2 1 1	
2024-04-07 17:40:19				19/04/1990	Dois a quatro anos	800€ a 900€	2 4 3 1 4 0 5	
2024-04-07 17:40:19				18/05/2000	Menos de um ano	901 a 1000	1 0 1 3 2 0 2	
2024-04-07 17:40:19				07/04/1968	Menos de um ano	901 a 1000	2 5 5 4 5 1 0	
2024-04-07 17:40:19				06/06/1978	Dois a quatro anos	901 a 1000	1 4 5 2 1 0 1	
2024-04-07 17:40:19				01/01/1990	Dois a quatro anos	Mais de 1200	1 1 5 0 1 4 2	
2024-04-07 17:40:19				06/12/1988	Dois a quatro anos	901 a 1000	4 1 4 4 3 1 5	
2024-04-07 17:40:19				01/01/1990	Menos de um ano	1001 a 1200	0 1 0 3 0 1 5	
2024-04-07 17:40:19				03/08/2003	Dois a quatro anos	800€ a 900€	3 0 5 5 5 0 5	
2024-04-07 17:40:19				06/05/1976	Menos de um ano	1001 a 1200	1 3 3 2 2 3 0	
2024-04-07 17:40:19				07/08/1967	Mais de 4 anos	1001 a 1200	5 3 1 0 4 0 4	
2024-04-07 17:40:19				04/01/1969	Um a dois anos	Mais de 1200	1 4 3 0 5 2 3	
2024-04-07 17:40:19				28/12/1991	Um a dois anos	Mais de 1200	0 0 2 3 2 5 4	
2024-04-07 17:40:19				19/05/1984	Menos de um ano	901 a 1000	5 4 3 1 4 2 0	
2024-04-07 17:40:19				17/09/1987	Mais de 4 anos	1001 a 1200	1 2 5 2 0 4 2	
2024-04-07 17:40:19				24/04/1974	Menos de um ano	Mais de 1200	5 4 3 1 3 4 4	
2024-04-07 17:40:19				25/04/1974	Um a dois anos	Mais de 1200	4 4 3 1 0 2 2	

Respostas do Formulário 1

Anexo 16: Tabela lógica da Função.

Função
Função_ID INT
Designação ENUM(...)
Indexes

Anexo 17: Tabela lógica do Funcionário.

Funcionário
Funcionário_ID INT
Nome VARCHAR(75)
Data_de_nascimento DATE
Salário INT
NIF VARCHAR(10)
Fotografia VARCHAR(150)
Função_ID INT
Indexes

Anexo 18: Tabela lógica do Número de Telemóvel.

Número_de_telefone
Número_de_telefone_ID INT
Funcionário_ID INT
Indexes

Anexo 19: Tabela lógica do Gere.

Gere
Funcionário_Gestor_ID INT
Funcionário_ID INT
Indexes

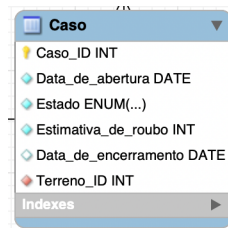
Anexo 20: Tabela lógica do Terreno.



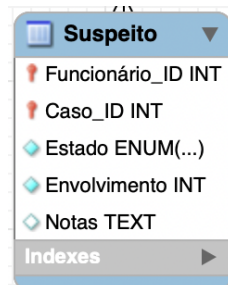
Anexo 21: Tabela lógica do Trabalho.



Anexo 22: Tabela lógica do Caso.



Anexo 23: Tabela lógica do Suspeito.



Anexo 24: Script de criação da base de dados em SQL.

```
CREATE DATABASE IF NOT EXISTS Lusium;

USE Lusium;

-- Criação da tabela Função
CREATE TABLE Função (
    Função_ID INT AUTO_INCREMENT NOT NULL,
    Designação ENUM('Operacional', 'Detetive', 'Representante') NOT NULL,
    PRIMARY KEY (Função_ID)
);

-- Criação da tabela Funcionário
CREATE TABLE Funcionário (
```

```

Funcionário_ID INT AUTO_INCREMENT NOT NULL,
Nome VARCHAR(75) NOT NULL,
Data_de_nascimento DATE NOT NULL,
Salário INT NOT NULL,
NIF VARCHAR(10) NOT NULL,
Fotografia VARCHAR(150) NULL,
Função_ID INT NOT NULL,
PRIMARY KEY (Funcionário_ID),
FOREIGN KEY (Função_ID) REFERENCES Função(Função_ID)
);

-- Criação da tabela Número de telemóvel
CREATE TABLE Número_de_telefone (
Número_de_telefone_ID INT AUTO_INCREMENT NOT NULL,
Funcionário_ID INT NOT NULL,
PRIMARY KEY (Número_de_telefone_ID),
FOREIGN KEY (Funcionário_ID) REFERENCES Funcionário(Funcionário_ID)
);

-- Criação da tabela Gere
CREATE TABLE Gere (
Funcionário_Gestor_ID INT NOT NULL,
Funcionário_ID INT NOT NULL,
PRIMARY KEY (Funcionário_Gestor_ID, Funcionário_ID),
FOREIGN KEY (Funcionário_Gestor_ID) REFERENCES Funcionário(Funcionário_ID),
FOREIGN KEY (Funcionário_ID) REFERENCES Funcionário(Funcionário_ID)
);

-- Criação da tabela Terreno
CREATE TABLE Terreno (
Terreno_ID INT AUTO_INCREMENT NOT NULL,
Minério_previsto INT NOT NULL,
Minério_coletado INT NOT NULL,
PRIMARY KEY (Terreno_ID)
);

-- Criação da tabela Trabalha
CREATE TABLE Trabalha (
Funcionário_ID INT NOT NULL,
Terreno_ID INT NOT NULL,
PRIMARY KEY (Funcionário_ID, Terreno_ID),
FOREIGN KEY (Funcionário_ID) REFERENCES Funcionário(Funcionário_ID),
FOREIGN KEY (Terreno_ID) REFERENCES Terreno(Terreno_ID)
);

-- Criação da tabela Caso
CREATE TABLE Caso (
Caso_ID INT AUTO_INCREMENT NOT NULL,
Data_de_abertura DATE NOT NULL,
Estado ENUM('Aberto', 'Fechado') NOT NULL,
Estimativa_de_roubo INT NOT NULL,
Data_de_encerramento DATE NULL,
Terreno_ID INT NOT NULL,
PRIMARY KEY (Caso_ID),
FOREIGN KEY (Terreno_ID) REFERENCES Terreno(Terreno_ID)
);

-- Criação da tabela Suspeito
CREATE TABLE Suspeito (
Funcionário_ID INT NOT NULL,
Caso_ID INT NOT NULL,

```

```

Estado ENUM('Inocente', 'Em investigação', 'Culpado') NOT NULL,
Envolvimento INT NOT NULL,
Notas TEXT NULL,
PRIMARY KEY (Funcionário_ID, Caso_ID),
FOREIGN KEY (Funcionário_ID) REFERENCES Funcionário(Funcionário_ID),
FOREIGN KEY (Caso_ID) REFERENCES Caso(Caso_ID)
);

```

Anexo 25: Script de povoamento da base de dados em SQL.

```

-- Inserção de dados na tabela Função
INSERT INTO Função (Designação) VALUES
    ('Operacional'),
    ('Detetive'),
    ('Representante');

-- Inserção de dados na tabela Funcionário
INSERT INTO Funcionário (Nome, Data_de_nascimento, Salário, NIF, Fotografia, Função_ID)
VALUES
    ('Miguel Silva', '1985-03-15', 1200, '123456789', 'foto_miguel.jpg', 1),
    ('Ana Costa', '1990-07-21', 2800, '987654321', 'foto_ana.jpg', 2),
    ('Pedro Santos', '1982-12-10', 3000, '456789123', 'foto_pedro.jpg', 3),
    ('Sofia Pereira', '1988-05-02', 1000, '789123456', 'foto_sofia.jpg', 1),
    ('Rui Oliveira', '1995-09-18', 2600, '321654987', 'foto_rui.jpg', 2),
    ('Inês Rodrigues', '1980-11-30', 850, '654987321', 'foto_ines.jpg', 1),
    ('Tiago Fernandes', '1993-04-25', 2900, '987321654', 'foto_tiago.jpg', 3);

-- Inserção de dados na tabela Número_de_telefone
INSERT INTO Número_de_telefone (Número_de_telefone_ID, Funcionário_ID) VALUES
    (934678592, 1),
    (966492873, 2),
    (922245762, 3),
    (936457856, 4),
    (926486516, 5),
    (964884547, 6),
    (964554178, 7);

-- Inserção de dados na tabela Gere
INSERT INTO Gere (Funcionário_Gestor_ID, Funcionário_ID) VALUES
    (3, 1),
    (3, 2),
    (3, 4),
    (6, 5),
    (6, 6),
    (6, 7);

-- Inserção de dados na tabela Terreno
INSERT INTO Terreno (Minério_previsto, Minério_coletado) VALUES
    (1000, 800),
    (1500, 1200),
    (800, 600),
    (2000, 1800),
    (1200, 1000),
    (1800, 1600),
    (900, 700);

```

```

-- Inserção de dados na tabela Trabalha
INSERT INTO Trabalha (Funcionário_ID, Terreno_ID) VALUES
    (1, 1),
    (2, 2),
    (3, 3),
    (4, 4),
    (5, 5),
    (6, 6),
    (7, 7);

-- Inserção de dados na tabela Caso
INSERT INTO Caso (Data_de_abertura, Estado, Estimativa_de_roubo, Data_de_encerramento,
Terreno_ID) VALUES
    ('2024-01-10', 'Aberto', 500, NULL, 1),
    ('2024-02-15', 'Fechado', 1000, '2024-05-05', 2),
    ('2024-03-20', 'Aberto', 700, NULL, 3),
    ('2024-04-25', 'Aberto', 800, NULL, 4),
    ('2024-05-01', 'Aberto', 300, NULL, 5),
    ('2024-05-08', 'Aberto', 900, NULL, 6),
    ('2024-05-10', 'Aberto', 600, NULL, 7);

-- Inserção de dados na tabela Suspeito
INSERT INTO Suspeito (Funcionário_ID, Caso_ID, Estado, Envolvimento, Notas) VALUES
    (1, 1, 'Inocente', 1, 'Nenhuma informação adicional.'),
    (2, 2, 'Culpado', 9, 'Suspeito estava presente na cena do crime.'),
    (3, 3, 'Em investigação', 5, 'Algumas pistas indicam possível envolvimento.'),
    (4, 4, 'Em investigação', 3, 'Não há provas concretas.'),
    (5, 5, 'Inocente', 2, 'Sem evidências contra o suspeito.'),
    (6, 6, 'Culpado', 8, 'Fortes indícios de participação.'),
    (7, 7, 'Em investigação', 4, 'Possível conexão com outras investigações.');
```

Anexo 26: Script de povoamento da base de dados em Pthon.

```

import mysql.connector
from faker import Faker
import random
from datetime import timedelta

# Solicita ao utilizador o nome de utilizador e a senha para se conectar ao MySQL
user = input("Insira o nome de utilizador da base de dados MySQL: ")
password = input("Insira a senha da base de dados MySQL: ")

# Estabelece a conexão com a base de dados MySQL
conn = mysql.connector.connect(
    host="localhost",
    user=user,
    password=password,
    database="Lusium"
)

# Verifica se a conexão foi bem-sucedida e imprime a versão do servidor
if conn.is_connected():
    print("Conectado ao MySQL Server versão ", conn.get_server_info())

# Cria um cursor para executar comandos SQL
cursor = conn.cursor()
```

```

# Instancia o Faker configurado para gerar dados em português de Portugal
fake = Faker('pt_PT')

# Lista de funções para inserir na tabela Função
funcoes = ['Operacional', 'Detetive', 'Representante']

# Insere as 3 funções na tabela Função
for i, funcao in enumerate(funcoes, start=1):
    cursor.execute("INSERT INTO Função (Função_ID, Designação) VALUES (%s, %s)", (i,
funcao))

# Gera e insere dados fictícios para 100 funcionários na tabela Funcionário
for i in range(1, 101):
    nome = fake.name()
    data_nascimento = fake.date_of_birth(minimum_age=18, maximum_age=65)
    salario = random.randint(1000, 5000)
    nif = fake.random_number(digits=9)
    fotografia = fake.image_url()
    funcao_id = random.randint(1, len(funcoes))
    cursor.execute("INSERT INTO Funcionário (Funcionário_ID, Nome, Data_de_nascimento,
Salário, NIF, Fotografia, Função_ID) VALUES (%s, %s, %s, %s, %s, %s, %s)",
(i, nome, data_nascimento, salario, nif, fotografia, funcao_id))

# Gera e insere 100 números de telefone sequenciais na tabela Número_de_tele móvel
for i in range(1, 101):
    num_telefone = 900000000 + i
    funcionario_id = i
    cursor.execute("INSERT INTO Número_de_tele móvel (Número_de_tele móvel_ID,
Funcionário_ID) VALUES (%s, %s)", (num_telefone, funcionario_id))

# Gera e insere 50 dados de gestão aleatória na tabela Gere
for i in range(1, 51):
    funcionario_gestor_id = i
    funcionario_id = random.randint(1, 100)
    cursor.execute("INSERT INTO Gere (Funcionário_Gestor_ID, Funcionário_ID) VALUES
(%s, %s)", (funcionario_gestor_id, funcionario_id))

# Gera e insere dados fictícios para 20 terrenos na tabela Terreno
for i in range(1, 21):
    minério_previsto = random.randint(1000, 10000)
    minério_coletado = random.randint(0, minério_previsto)
    cursor.execute("INSERT INTO Terreno (Terreno_ID, Minério_previsto,
Minério_coletado) VALUES (%s, %s, %s)", (i, minério_previsto, minério_coletado))

# Gera e insere 100 dados de trabalho na tabela Trabalha
for i in range(1, 101):
    funcionario_id = i
    terreno_id = random.randint(1, 20)
    cursor.execute("INSERT INTO Trabalha (Funcionário_ID, Terreno_ID) VALUES (%s, %s)",
(funcionario_id, terreno_id))

# Gera e insere dados fictícios para 50 casos na tabela Caso
for i in range(1, 51):
    data_abertura = fake.date_between(start_date='-1y', end_date='today')
    estado = random.choice(['Aberto', 'Fechado'])
    estimativa_roubo = random.randint(1000, 100000)
    data_encerramento = data_abertura + timedelta(days=random.randint(10, 365)) if
estado == 'Fechado' else None
    terreno_id = random.randint(1, 20)
    cursor.execute("INSERT INTO Caso (Caso_ID, Data_de_abertura, Estado,
Estimativa_de_roubo, Data_de_encerramento, Terreno_ID) VALUES (%s, %s, %s, %s, %s,

```



```

%s)",
                (i, data_abertura, estado, estimativa_roubo, data_encerramento,
                terreno_id))

# Gera e insere dados fictícios de 100 suspeitos na tabela Suspeito
for i in range(1, 101):
    funcionário_id = i
    caso_id = random.randint(1, 50)
    estado = random.choice(['Inocente', 'Em investigação', 'Culpado'])
    envolvimento = random.randint(1, 10)
    notas = fake.text()
    cursor.execute("INSERT INTO Suspeito (Funcionário_ID, Caso_ID, Estado, Envolvimento,
    Notas) VALUES (%s, %s, %s, %s, %s)",
                    (funcionário_id, caso_id, estado, envolvimento, notas))

# Imprime a mensagem de sucesso
print("Dados inseridos com sucesso!")

# Confirma todas as transações e fecha a conexão com a base de dados
conn.commit()
conn.close()

```

Anexo 27: Script de procedimentos, funções e gatilhos da base de dados em SQL.

```

-- Trigger: Atualizar Data de Encerramento do Caso ao Fechar
DELIMITER $$
CREATE TRIGGER AtualizarDataEncerramento
BEFORE UPDATE ON Caso
FOR EACH ROW
BEGIN
    IF NEW.Estado = 'Fechado' AND OLD.Estado != 'Fechado' THEN
        SET NEW.Data_de_encerramento = CURDATE();
    END IF;
END $$
DELIMITER ;

-- Function: Calcular Média Salarial por Função
DELIMITER $$

CREATE FUNCTION MediaSalarialPorFuncao(p_Função_ID INT) RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
    DECLARE media_salarial DECIMAL(10, 2);

    SELECT AVG(Salário) INTO media_salarial
    FROM Funcionário
    WHERE Função_ID = p_Função_ID;

    RETURN media_salarial;
END $$

DELIMITER ;

DELIMITER $$

```

```

CREATE FUNCTION CalcularIdade(Data_de_nascimento DATE)
RETURNS INT DETERMINISTIC
BEGIN
    DECLARE idade INT;

    -- Calcula a idade considerando apenas o ano
    SET idade = YEAR(CURDATE()) - YEAR(Data_de_nascimento);

    -- Ajusta a idade se o aniversário ainda não ocorreu este ano
    IF (MONTH(CURDATE()) < MONTH(Data_de_nascimento)) OR
        (MONTH(CURDATE()) = MONTH(Data_de_nascimento) AND DAY(CURDATE()) <
DAY(Data_de_nascimento)) THEN
        SET idade = idade - 1;
    END IF;

    RETURN idade;
END $$

DELIMITER ;

DELIMITER $$
CREATE FUNCTION CalcularEstimativaDeRoubo(p_Terreno_ID INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE v_Estimativa_De_Roubo INT;

    -- Obter o minério previsto e coletado para o terreno especificado
    SELECT (Minério_previsto - Minério_coletado) INTO v_Estimativa_De_Roubo
    FROM Terreno WHERE Terreno_ID = p_Terreno_ID;

    RETURN v_Estimativa_De_Roubo;
END $$
DELIMITER ;

-- CALL CriarCasoETornarSuspeitos(TERRENO_ID, CURDATE());
DELIMITER $$
CREATE PROCEDURE CriarCasoETornarSuspeitos(
    IN p_Terreno_ID INT,
    IN p_Data_de_abertura DATE
)
BEGIN
    DECLARE v_Caso_ID INT;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        -- Em caso de erro, reverter todas as operações
        ROLLBACK;
    END;

    START TRANSACTION;

    -- Insere um novo caso
    INSERT INTO Caso (Data_de_abertura, Estado, Estimativa_de_roubo,
Data_de_encerramento, Terreno_ID)
    VALUES (p_Data_de_abertura, 'Aberto', CalcularEstimativaDeRoubo(p_Terreno_ID),
NULL, p_Terreno_ID);

    -- Obter o último ID do caso inserido
    SET v_Caso_ID = LAST_INSERT_ID();

    -- Seleciona todos os funcionários que trabalham no terreno especificado

```

```

INSERT INTO Suspeito (Funcionário_ID, Caso_ID, Estado, Envolvimento, Notas)
SELECT t.Funcionário_ID, v_Caso_ID, 'Em Investigação', 3, 'Funcionário estava
presente no terreno no dia do roubo'
FROM Trabalha t
WHERE t.Terreno_ID = p_Terreno_ID;

COMMIT;
END $$

DELIMITER ;

```

Anexo 28: Script de índices em SQL.

```

-- Criação de um índice para a coluna Terreno_ID na tabela Caso
CREATE INDEX index_caso_terreno_id ON Caso(Terreno_ID);

-- Criação de um índice para a coluna Caso_ID na tabela Suspeito
CREATE INDEX idx_suspeito_caso_id ON Suspeito(Caso_ID);

-- Criação de um índice para a coluna Funcionário_ID na tabela Suspeito
CREATE INDEX idx_suspeito_funcionario_id ON Suspeito(Funcionário_ID);

```

Anexo 29: Script de vistas da base de dados em SQL.

```

-- View das informações dos funcionários e dos casos associados.
CREATE VIEW FuncionariosEmCasos AS
SELECT
    Funcionário.Nome,
    Número_de_telefone.Número_de_telefone_ID,
    Caso.Caso_ID,
    Caso.Data_de_abertura,
    Caso.Estado,
    Caso.Estimativa_de_roubo,
    Caso.Data_de_encerramento,
    Caso.Terreno_ID
FROM
    Funcionário
INNER JOIN
    Suspeito ON Funcionário.Funcionário_ID = Suspeito.Funcionário_ID
INNER JOIN
    Caso ON Suspeito.Caso_ID = Caso.Caso_ID
LEFT JOIN
    Número_de_telefone ON Funcionário.Funcionário_ID =
Número_de_telefone.Funcionário_ID;

-- View das informações dos funcionários e dos terrenos associados.
CREATE VIEW FuncionariosEmTerrenos AS
SELECT
    Funcionário.Nome,
    Função.Designação,
    Número_de_telefone.Número_de_telefone_ID,
    CalcularIdade(Funcionário.Data_de_nascimento) AS Idade,
    Funcionário.Salário,

```

```

        Terreno.Terreno_ID,
        Terreno.Minério_previsto,
        Terreno.Minério_coletado
FROM
    Funcionário
INNER JOIN
    Trabalha ON Funcionário.Funcionário_ID = Trabalha.Funcionário_ID
INNER JOIN
    Terreno ON Trabalha.Terreno_ID = Terreno.Terreno_ID
INNER JOIN
    Função ON Funcionário.Função_ID = Função.Função_ID
LEFT JOIN
        Número_de_telefone ON Funcionário.Funcionário_ID =
        Número_de_telefone.Funcionário_ID;

```

Anexo 30: Script de interrogações da base de dados em SQL.

```

-- Definir variáveis para utilizar nas queries
SET @Funcionario_ID = 3;
SET @Caso_ID = 3;
SET @Terreno_ID = 1;

-- Listar o prejuízo de um terreno
SELECT
    Terreno_ID,
    (Minério_previsto - Minério_coletado) AS Prejuízo
FROM
    Terreno
WHERE Terreno_ID = @Terreno_ID;

-- Ver quando é que um funcionário se tornou suspeito de um determinado caso
SELECT
    s.Funcionário_ID,
    s.Caso_ID,
    c.Data_de_abertura
FROM
    Suspeito s
INNER JOIN
    Caso c ON s.Caso_ID = c.Caso_ID
WHERE
    s.Funcionário_ID = @Funcionario_ID
    AND s.Caso_ID = @Caso_ID;

-- Listar os suspeitos de um determinado caso
SELECT
    s.Funcionário_ID,
    f.Nome,
    s.Estado,
    s.Envolvimento,
    s.Notas
FROM
    Suspeito s
INNER JOIN
    Funcionário f ON s.Funcionário_ID = f.Funcionário_ID
WHERE
    s.Caso_ID = @Caso_ID;

```

```

-- Ver a data do último caso de um determinado funcionário
SELECT
    c.Caso_ID,
    c.Data_de_abertura
FROM
    Caso c
INNER JOIN
    Suspeito s ON c.Caso_ID = s.Caso_ID
WHERE
    s.Funcionário_ID = @Funcionario_ID
ORDER BY
    c.Data_de_abertura DESC
LIMIT 1;

-- Listar os casos a que um determinado funcionário está associado
SELECT
    c.Caso_ID,
    c.Data_de_abertura,
    c.Estado,
    c.Estimativa_de_roubo,
    c.Data_de_encerramento,
    c.Terreno_ID
FROM
    Caso c
INNER JOIN
    Suspeito s ON c.Caso_ID = s.Caso_ID
WHERE
    s.Funcionário_ID = @Funcionario_ID;

-- Ver o dia em que mais casos foram abertos
SELECT
    Data_de_abertura,
    COUNT(*) AS Total_de_Casos
FROM
    Caso
GROUP BY
    Data_de_abertura
ORDER BY
    Total_de_Casos DESC
LIMIT 1;

-- Listar os top 5 funcionários por quantidade de casos
SELECT
    s.Funcionário_ID,
    f.Nome,
    COUNT(s.Caso_ID) AS Total_de_Casos
FROM
    Suspeito s
INNER JOIN
    Funcionário f ON s.Funcionário_ID = f.Funcionário_ID
GROUP BY
    s.Funcionário_ID, f.Nome
ORDER BY
    Total_de_Casos DESC
LIMIT 5;

```