“DEθ (Deteta)”

Implementação de um Sistema de Gestão de Base de Dados

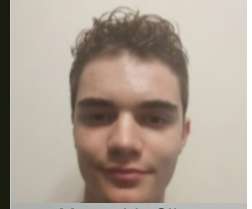
Grupo 25

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

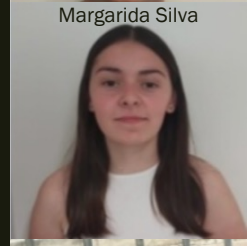
Universidade do Minho
Escola de Engenharia

Equipa de Trabalho e Tema do Projeto

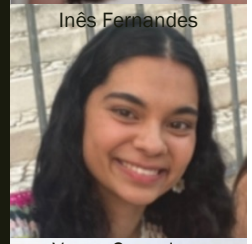
Alexandre Monsanto



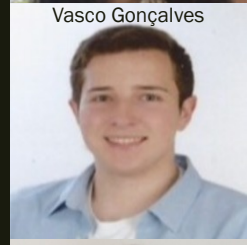
Margarida Silva



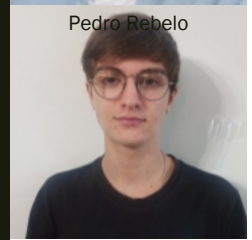
Inês Fernandes

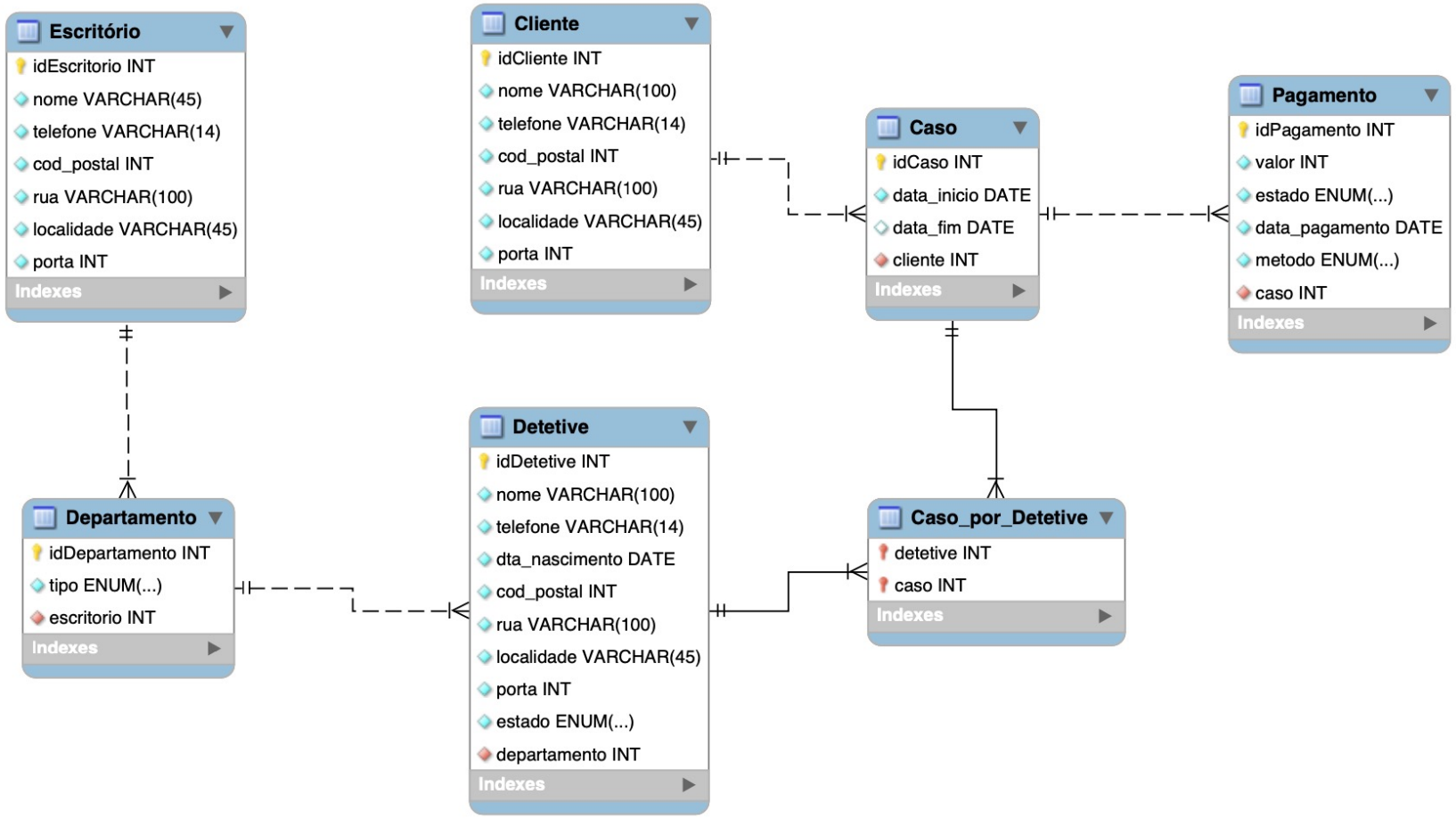


Vasco Gonçalves



Pedro Rebelo





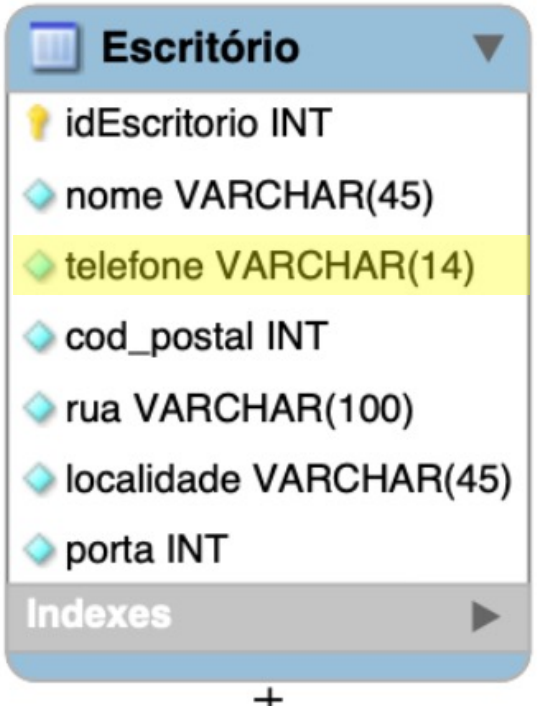
Estrutura da apresentação

- Correções feitas à primeira fase;
- Processo de implementação física da base de dados;
- Caracterização dos utilizadores;
- Povoamento da base de dados;
- Cálculo do espaço da base de dados;
- Definição e caracterização de vistas de utilização em SQL;
- Interrogações do utilizador (tradução em SQL);
- Indexação do sistema de dados;
- Implementação de procedimentos, funções e gatilhos;
- Conclusões e trabalho futuro.

Correções feitas à primeira fase

- Alteração do tipo de dados do atributo “telefone” nas entidades “Escritório”, “Detetive” e “Cliente” de INT para VARCHAR(14) para permitir o registo de nºs de telefone estrangeiros.

- Por exemplo:
- Um nº português será registado com o indicativo 00351 e os 9 dígitos que o constituem.
- O tipo VARCHAR garante que os dois primeiros zeros não são eliminados pelo sistema, o que aconteceria se os dados de “telefone” fossem caracterizados com, por exemplo, INT.



Correções feitas à primeira fase

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

- Por exemplo:
- O ‘metodo’ tem o tipo ENUM(‘Numerário’, ‘Cartão’, ‘MBWay’, ‘Transferência’)
- Não pode ser introduzido o método ‘numerario’.
- Este *data type* garante que só existem estes 4 métodos de pagamento, ao contrário de VARCHAR.

Correções feitas à primeira fase

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

- Por não termos esclarecido dúvidas com os docentes, foi decidido pela equipa alterar o “estado” do Detetive para ENUM('No Ativo', 'Reformado') e o do Pagamento para ENUM(‘Pago', 'Não Pago’).
- Este *data type* garante que só existem estes estados.

Implementação física da base de dados

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

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

Implementação física da base de dados

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

```sql
CREATE TABLE IF NOT EXISTS Escritório (
    idEscritorio INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(45) NOT NULL,
    telefone VARCHAR(14) NOT NULL,
    cod_postal INT NOT NULL,
    rua VARCHAR(100) NOT NULL,
    localidade VARCHAR(45) NOT NULL,
    porta INT NOT NULL,
    PRIMARY KEY (idEscritorio)
    );
```

```sql
CREATE TABLE IF NOT EXISTS Departamento (
    idDepartamento INT NOT NULL,
    tipo ENUM('Homicídios', 'Roubos', 'Sequestros') NOT NULL,
    escritorio INT NOT NULL,
    PRIMARY KEY (idDepartamento),
    FOREIGN KEY (escritorio) REFERENCES Escritório(idEscritorio)
    );
```

Caracterização dos utilizadores

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

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

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

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

GRANT SELECT ON ClienteRestr TO cliente1;
GRANT SELECT ON CasosSolucionadosCliente1 TO cliente1;

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

Povoamento da base de dados

O próximo passo a seguir é o povoamento da base de dados, que é feito de duas formas. Na primeira, inserimos informações diretamente na base de dados através da instrução INSERT em SQL. Na segunda forma, criamos um pograma, na linguagem Python, que acede à base de dados e nos permite inserir dados em duas tabelas distintas (Cliente e Caso).

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

```python
import mysql.connector

basedados = mysql.connector.connect(
    host="127.0.0.1",
    database="AgenciaDeteta",
    user="root",
    password="pedro2004")

mycursor = basedados.cursor()

sql1 = "INSERT INTO Cliente(idCliente,nome,telefone,cod_postal,rua,localidade,porta) VALUES(%s,%s,%s,%s,%s,%s,%s)"
valores1 = [("15813902 4","Rui Jorge Loureiro","927734210","4830502","Rua da Grila","Taíde","38"),
("2502140128","Cátia Vanessa","926803555","4700432","Rua Daculá","Nogueira","321"),
("199132845","Rui Redes","913914256","4720418","Avenida Mordômo Cunha","Leiria","2"),
("331882003","Mário Guedes","913432991","4512512","Rua de Lima","Ponte de Lima","501"),
("255621023","Juliana Pinha","969655445","4101995","Avenida da Republica","Lamações","132"),]

sql2 = "INSERT INTO Caso(idCaso, data_inicio,data_fim, cliente) VALUES(%s,%s,%s,%s)"
valores2=[("116","2025-04-27","2025-05-15","158139024"),
("117","2025-04-28","2025-06-06","250140128"),
("118","2025-04-28","2025-06-16","199132845"),
("119","2025-04-30","2025-07-01","331882003"),
("120","2025-05-02","2025-07-01","255621023"),]

mycursor.executemany(sql1,valores1)
mycursor.executemany(sql2,valores2)

basedados.commit()

basedados.close()
```

Cálculo do espaço da base de dados (inicial)

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

Tabela	Atributos	Tipo de Dados	Tamanho (bytes)	Ocorrências	Subtotal
Escritório	idEscritorio	INT	4	9	36
	nome	VARCHAR(45)	46		414
	telefone	VARCHAR(14)	15		135
	cod_postal	INT	4		36
	rua	VARCHAR(100)	101		909
	localidade	VARCHAR(45)	46		414
	porta	INT	4		36
Departamento	idDepartamento	INT	4	27	108
	tipo	ENUM('Homicídios', 'Roubos', 'Sequestros')	1		27
	escritorio	INT	4		108
					(...)
				TOTAL	**54804**

Cálculo do espaço da base de dados (após 1 ano)

- Nos 3 anos anteriores à implementação da base de dados (2022, 2023 e 2024), foram resolvidos entre 8 e 9 casos em cada ano. Assim, tendo em conta que cada caso está associado a um cliente, a, no máximo, 2 pagamentos e a, em média, dois detetives, a equipa pôde calcular uma previsão de crescimento para a base de dados da agência.
- Assumiu-se que os valores das tabelas Escritório, Departamento e Detetive não se alterariam (Theta não está a planear expandir a agência nos próximos anos).
- Estima-se que a base de dados ocupe 57855 bytes após o primeiro ano, o que corresponde a um crescimento de, aproximadamente, **5,57%.**

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

Vistas de utilização em SQL

View 1: Detalhes de Casos e Detetives

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

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

Vistas de utilização em SQL

View 2: Pagamentos por Caso

Uma view que mostra os pagamentos realizados para cada caso.

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

Interrogações do utilizador

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

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

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

Interrogações do utilizador

Query3: Qual foi a faturação do escritório 2 no mês de fevereiro?

Criamos uma function para obter a faturação de um escritório específico num certo mês.

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

Indexação do sistema de dados

- A indexação desempenha um papel fundamental na otimização do desempenho e da eficiência das consultas num sistema de base de dados. Como duas das coisas mais procuradas são os contactos dos clientes e os id's dos casos decidimos conceder-lhes índices:

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

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

Procedimentos

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

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

Funções

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

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

Gatilhos

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

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

Exemplo de aplicação do gatilho

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Conclusões e Trabalho Futuro

- A implementação física do sistema de bases de dados, foi desenvolvida no MySQLWorkbench. Explicando e justificando a criação de cada tabela, apresentámos o esquema físico resultante, assim como a caracterização de utilizadores, o seu povoamento, o cálculo do espaço ocupado, as vistas de utilização, mas interrogações do utilizador traduzidas para SQL, a indexação do sistema de dados e a implementação de procedimentos, funções e gatilhos.
- Para o futuro, será importante realizar uma revisão contínua do sistema de base de dados para garantir que continue a atender às necessidades em constante evolução da agência.
- Posteriormente, é necessário desenvolver programas para recolha de dados e criação de painéis de análise. Ao garantir a disponibilidade e competência adequadas desses recursos, a DEθ estará bem posicionada para implementar uma base de dados eficaz que atenda às suas necessidades operacionais e estratégicas.

“DEθ (Deteta)”

Implementação de um Sistema de Gestão de Base de Dados

Grupo 25

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

Universidade do Minho
Escola de Engenharia