

Gestão de Projetos

Xavier Tacanho
Nº45930
*Departamento de Informática
Universidade da Beira Interior
Covilhã, Portugal
xavier.tacanho@ubi.pt*

Pedro Martins
Nº50076
*Departamento de Informática
Universidade da Beira Interior
Covilhã, Portugal
pv.martins@ubi.pt*

Abstract—No contexto da Unidade Curricular de Base de Dados, foi-nos proposto este projeto laboratorial, que consiste no desenvolvimento e implementação de uma base de dados e de uma aplicação "cliente" para interagir com a informação armazenada na base de dados desenvolvida.

Keywords: *SQL Server Management Studio 19, ERDPlus, Javafx*

I. INTRODUÇÃO

O presente projeto laboratorial intitulado de Gestão de Projetos tem como objetivo o desenvolvimento de um sistema abrangente que permite monitorizar os projetos internos e financiados, tanto como os contratos de prestação de serviço. O objetivo é conseguir construir esta ferramenta de modo a implementar tudo o que nos fora fornecido nas aulas práticas e teóricas da cadeira de Base de Dados lecionadas pelos Professores Rui Cardoso e o Luís Horta.

A implementação destes serviços será realizada utilizando as seguintes plataforma:

- **ERD Plus;**
- **SQL Server Management Studio 19;**
- **DBSchema;**
- **Java;**
- **Javafx.**

A base do projeto irá incluir a criação de funcionalidades para que seja possível a consulta e manipulação da informação inserida dentro da nossa base de dados.

Além desta nossa gestão de dados e tarefas iremos criar de raiz via ERD Plus os nossos diagramas de Entidade-Relacionamento (**ER**) e o nosso modelo Relacional, tendo em conta a 3ª forma normal e a criação das nossas chaves primárias (**PK**) e das nossas chaves estrangeiras (**FK**).

Após a criação destes modelos iremos fazer a implementação e criação da nossa Base de dados no SQL Server Management Studio 19 e a criação das queries para de modo a que seja possível verificar se na base de dados conseguimos demonstrar consultas, inserções, atualizações, remoções de dados e alertas.

Por fim iremos criar uma aplicação em Java/JavaFx de modo a termos uma interface que demonstre a interação com a base de dados desenvolvida com o fim de demonstrar a sua interação.

II. MODELO DE DADOS

Para o desenvolvimento desta aplicação, utilizámos a framework **ERDPlus** online de modo a desenvolver e percebermos

como quebrar este enigma da Gestão de Tarefas. Deste modo conseguimos subdividi-lo de maneira a que tivéssemos uma base de dados limpa e na 3ª Forma Normal na implementação e funcionamento do sistema.

A. **ERDPlus**

Para o desenvolvimento deste projeto, utilizamos o software ERDPlus para criar diagramas de relacionamento de entidades (ERD), facilitando a modelagem da base de dados. Esta ferramenta foi essencial para garantir a estrutura e integridade dos dados, permitindo uma visualização clara das relações entre as diferentes entidades do sistema.

1) **Modelo Relacional:** Com este modelo tentamos ver quais seriam as dependências gerais que seriam necessárias tais como as chaves primárias e as chaves secundárias.

No nosso caso acabamos por subdividir a nossa Base de Dados em 20 tabelas sendo estas:

- 1) Tabela Projeto;
- 2) Tabela Estado;
- 3) Tabela Tipo_estado;
- 4) Tabela Keywords;
- 5) Tabela Keywords_projeto;
- 6) Tabela Dominio;
- 7) Tabela Tipo_Dominio;
- 8) Tabela Area;
- 9) Tabela Tipo_Area_Cientifica;
- 10) Tabela Entidade;
- 11) Tabela País;
- 12) Tabela Programa;
- 13) Tabela Departamento_Investigação;
- 14) Tabela Departamento;
- 15) Tabela Membros_DIUBI;
- 16) Tabela Função_Membro;
- 17) Tabela Publicação;
- 18) Tabela Publicação_Identificadores;
- 19) Tabela Financiamento;
- 20) Tabela Tipo_Financiamento;
- 21) Tabela ProjetosEntidade.

Na Figura 1, apresentamos a solução proposta para a interação entre as nossas tabelas, demonstrando detalhadamente a organização estruturada que decidimos adotar. Esta representação ilustra como as diferentes tabelas se relacionam entre si, facilitando a compreensão das interações e a manutenção da integridade dos dados no sistema. Vale

destacar que a estrutura foi normalizada até a Terceira Forma Normal (3FN), garantindo a eliminação de redundâncias e a consistência dos dados.



Fig. 1. Modelo Relacional proposto e utilizado

B. Modelo Entidade-Relação

Na Figura 2, apresentamos o modelo de relacionamento de entidades (ERD), detalhando a interação entre as nossas tabelas e a organização estruturada que decidimos adotar. Esta representação ilustra como as diferentes entidades se relacionam entre si, facilitando a compreensão das interações e a manutenção da integridade dos dados no sistema. Destacando ainda que neste modelo também se encontra já na Terceira Forma Normal (3FN) para garantir a eliminação de redundâncias.

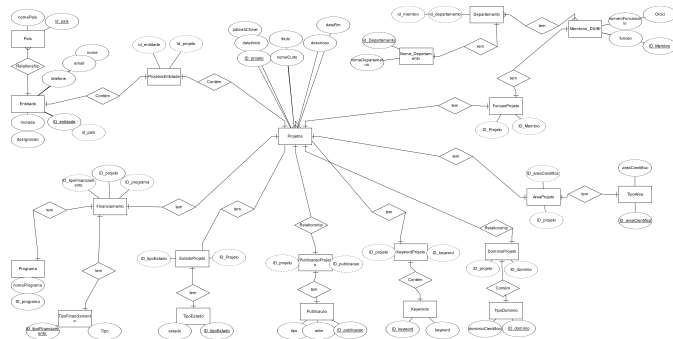


Fig. 2. Modelo Entidade-Relação proposto e utilizado

C. SQL Server Management Studio 19

Para o nosso projeto, seguimos a recomendação do professor e optamos por utilizar o SQL Server Management Studio 19

(SSMS 19). Escolhemos essa ferramenta devido à sua gama de recursos e funcionalidades que simplificam a administração e gerenciamento de bancos de dados SQL Server. Com uma interface intuitiva e recursos avançados de otimização de desempenho, o SSMS 19 se mostrou a melhor opção para atender às demandas do nosso trabalho, conforme orientado pelo professor.

• Criação da Base de Dados:

```
--Criação da Base de Dados--

USE master

IF ( EXISTS( SELECT * FROM [sys].[Databases] WHERE name = 'PROJETO'))
BEGIN
    DROP DATABASE PROJETO
END

IF (NOT EXISTS( SELECT * FROM [sys].[Databases] WHERE name = 'PROJETO'))
BEGIN
    CREATE DATABASE PROJETO
END
```

Fig. 3. Código do uso e criação da base de dados

• Inserts de teste inseridos na base de dados:

```
--Inserts de teste na base de dados--

USE PROJETO;

-- Insert statements for TipoEstado
INSERT INTO TipoEstado (estado) VALUES
('Iniciado'),
('Em Progresso'),
('Concluído');

-- Insert statements for TipoDominio
INSERT INTO TipoDominio (dominioCientifico) VALUES
('Ciências Naturais'),
('Engenharia e Tecnologia'),
('Ciências Médicas e da Saúde');

-- Insert statements for TipoArea
INSERT INTO TipoArea (areaCientifica) VALUES
('Biologia'),
('Física'),
('Química');

-- Insert statements for Membros_DIUBI
INSERT INTO Membros_DIUBI (numeroFuncionario, orcid, funcao) VALUES
('001', 123456789, 1),
('002', 987654321, 2),
('003', 112233445, 3);
```

Fig. 4. Inserção de dados na Base de dados

```

-- Insert statements for TipoFinanciamento
INSERT INTO TipoFinanciamento (Tipo) VALUES
(1),
(0);

-- Insert statements for NomeDepartamento
INSERT INTO NomeDepartamento (nomeDepartamento) VALUES
('Departamento de Física'),
('Departamento de Química'),
('Departamento de Biologia');

-- Insert statements for Programa
INSERT INTO Programa (nomePrograma) VALUES
('Programa de Investigação Científica'),
('Programa de Desenvolvimento Tecnológico'),
('Programa de Saúde Pública');

-- Insert statements for Publicacao
INSERT INTO Publicacao (tipo, valor) VALUES
('Artigo Científico', 'Nature'),
('Livro', 'Springer'),
('Conferência', 'IEEE');

-- Insert statements for Keywords
INSERT INTO Keywords (keyword) VALUES
('Biotecnologia'),
('Nanotecnologia'),
('Inteligência Artificial');

```

Fig. 5. Continuação da inserção de dados

```

-- Insert statements for Pais
INSERT INTO Pais (nomePais) VALUES
('Portugal'),
('Espanha'),
('Brasil');

-- Insert statements for NomeDepartamento
INSERT INTO NomeDepartamento (nomeDepartamento) VALUES
('Departamento de Ciências da Computação'),
('Departamento de Matemática'),
('Departamento de Engenharia Civil');

-- Insert statements for Projetos
INSERT INTO Projetos (nomeCurto, titulo, descricao, palavraChave, dataInicio, dataFim, ID_entidade) VALUES
('ProjBio', 'Projeto de Biotecnologia', 'Desenvolvimento de novas técnicas biotecnológicas', 'Biotecnologia', '2023-01-01', '2024-01-01', 1),
('ProjNano', 'Projeto de Nanotecnologia', 'Pesquisa em nanomateriais', 'Nanotecnologia', '2023-02-01', '2024-02-01', 2),
('ProjIA', 'Projeto de Inteligência Artificial', 'Desenvolvimento de sistemas de IA', 'Inteligência Artificial', '2023-03-01', '2024-03-01', 3);

-- Insert statements for Entidade
INSERT INTO Entidade (nome, email, telefone, designacao, morada, ID_projeto, ID_pais) VALUES
('Universidade de Lisboa', 'contact@ulb.pt', '351123456789', 'Instituição de Ensino Superior', 'Avenida da Universidade, Lisboa', 1, 1),
('Universidade do Porto', 'info@up.pt', '351987654321', 'Instituição de Ensino Superior', 'Rua da Universidade, Porto', 2, 2),
('Universidade de Coimbra', 'mail@uc.pt', '351456789123', 'Instituição de Ensino Superior', 'Praça da Universidade, Coimbra', 3, 3);

-- Insert statements for Departamento
INSERT INTO Departamento (ID_departamento, ID_membro) VALUES
(1, 1),
(2, 2),
(3, 3);

```

Fig. 6. Continuação da inserção de dados

D. Java/Javafx

Utilizamos Java ,mais concretamente a Framework JavaFX, para criar a interação entre a base de dados e a nossa aplicação. Com Java, implementamos a lógica necessária para acessar, obter e modificar dados no SQL Server, garantindo que todas as operações fossem realizadas de forma eficiente e segura. A interface gráfica da aplicação foi desenvolvida

com JavaFX, proporcionando uma experiência de usuário intuitiva e amigável. Esta combinação permitiu não apenas a manipulação de dados, mas também a exibição de resultados e a realização de diversas operações na base de dados, atendendo perfeitamente aos requisitos do projeto.

• Interface Gráfica



Fig. 7. Parte inicial da aplicação

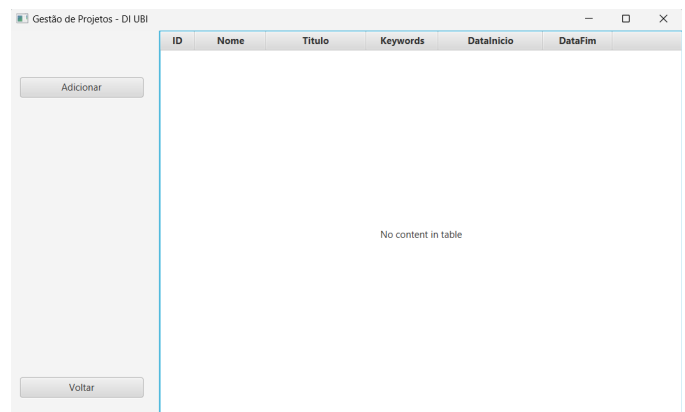


Fig. 8. Consulta de projetos na aplicação

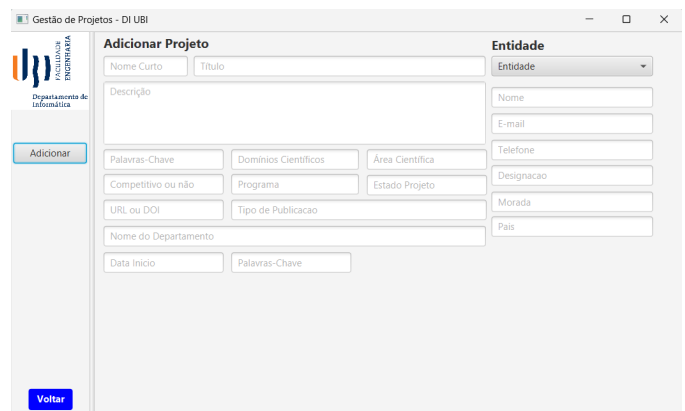


Fig. 9. Adição de mais projetos na base de dados

• Código gerado:

```
private void loadEntidades() {
    try {
        // Connect to your database
        Connection conn = Database.getConnection();
        Statement stat = conn.createStatement();
        ResultSet rs = stat.executeQuery("SELECT E.ID_entidade, E.nome, E.email, E.telefone, E.designacao, E.morada, p.nomePais " +
            "FROM Entidade E, Pais p " +
            "WHERE E.ID_pais = p.ID_pais");

        while (rs.next()) {
            MenuItem menuItem = new MenuItem(rs.getString("nome"));
            Entidade entidade = new Entidade(
                rs.getInt("ID_entidade"),
                rs.getString("nome"),
                rs.getString("email"),
                rs.getString("telefone"),
                rs.getString("designacao"),
                rs.getString("morada"),
                rs.getString("nomePais")
            );
            menuItem.setData(entidade);
            menuItem.setOnAction(event -> selectEntidade(entidade));
            menuButtonEntidade.getItems().add(menuItem);
        }
        rs.close();
        stat.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Fig. 10. Adiciona valores a base de dados

```
// Insert into other related tables
insertIntoDominiosCientificos(conn, projetoId);
insertIntoAreasCientificas(conn, projetoId);
insertIntoPublicacoes(conn, projetoId);
insertIntoDepartamentos(conn, projetoId);

conn.commit(); // Commit transaction
conn.setAutoCommit(true);

// Optionally, clear the fields after insertion
clearFields();

} catch (Exception e) {
    e.printStackTrace();
}
```

Fig. 11. Adiciona valores a base de dados

```
private void loadEntidades() {
    try {
        // Connect to your database
        Connection conn = Database.getConnection();
        Statement stat = conn.createStatement();
        ResultSet rs = stat.executeQuery("SELECT E.ID_entidade, E.nome, E.email, E.telefone, E.designacao, E.morada, p.nomePais " +
            "FROM Entidade E, Pais p " +
            "WHERE E.ID_pais = p.ID_pais");

        while (rs.next()) {
            MenuItem menuItem = new MenuItem(rs.getString("nome"));
            Entidade entidade = new Entidade(
                rs.getInt("ID_entidade"),
                rs.getString("nome"),
                rs.getString("email"),
                rs.getString("telefone"),
                rs.getString("designacao"),
                rs.getString("morada"),
                rs.getString("nomePais")
            );
            menuItem.setData(entidade);
            menuItem.setOnAction(event -> selectEntidade(entidade));
            menuButtonEntidade.getItems().add(menuItem);
        }
        rs.close();
        stat.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Fig. 12. Mostra as entidades

III. CONCLUSÃO

Neste projeto, seguimos a recomendação do professor e utilizamos o SQL Server Management Studio 19 (SSMS 19) para gerenciar a base de dados SQL Server. Optamos pelo SSMS 19 devido à sua vasta gama de ferramentas e funcionalidades, que simplificam a administração e o gerenciamento dos dados.

Para a interface e a interação com a base de dados, utilizamos Java e JavaFX. O Java foi fundamental para implementar a lógica de acesso, obtenção e modificação dos dados no SQL Server, garantindo que todas as operações fossem realizadas de maneira eficiente e segura. Com o JavaFX, desenvolvemos uma interface gráfica amigável e intuitiva, proporcionando uma boa experiência de usuário.

A combinação dessas tecnologias permitiu a manipulação eficaz dos dados, a exibição de resultados e a realização de várias operações necessárias ao projeto. Em resumo, a utilização do SSMS 19 juntamente com Java e JavaFX se mostrou uma escolha acertada, atendendo plenamente às necessidades e exigências do nosso trabalho.

Tivemos alguns problemas a nível de implementação de funcionalidades. Como trabalho futuro temos como objetivo a melhoria dos mesmos.

REFERENCES

- [1] <https://erdplus.com>
- [2] <https://www.w3schools.com/sql/>
- [3] <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16#download-ssms>
- [4] https://www.w3schools.com/java/java_examples.asp