
Escola Secundária Ferreira Dias

**Curso Profissional Técnico de Gestão e
Programação de Sistemas Informáticos**

2023/ 2026

Projeto Final - M19

Relatório do Projeto

Danylo Duma

2ºGI

Agualva-Cacém, maio de 2025

Tema

Projeto Final - M19

Orientador

Prof.^a Fernanda Oliveira

(assinatura)

Siglas e Acrónimos

GUI - Interface Gráfica do Utilizador (Graphical User Interface)

POO - Programação Orientada a Objetos

SGBD - Sistema de Gestão de Base de Dados

CRUD - Create, Read, Update, Delete (Criar, Ler, Atualizar, Eliminar)

DAO - Data Access Object (Objeto de Acesso a Dados)

MVC - Model - View - Controller (Modelo - Visão - Controlador)

SQL - Structured Query Language

JDBC - Java Database Connectivity

FXML - JavaFX Markup Language

Glossário

Entidade - Representação de um objeto real no sistema (ex: Livro, Autor, Tema).

Interface Gráfica (GUI) - Área visual onde o utilizador interage com a aplicação.

CRUD - Conjunto de operações básicas para manipulação de dados em sistemas informáticos.

JavaFX - Biblioteca Java usada para construir interfaces gráficas modernas.

FXML - Ficheiro XML usado para descrever interfaces gráficas em JavaFX.

DAO - Classe responsável por comunicar com a base de dados e realizar operações como inserir, atualizar e eliminar dados.

SGBD - Software responsável por gerir, armazenar e recuperar dados de forma estruturada.

MVC - Padrão de arquitetura que separa a lógica de dados, apresentação e controlo de ações.

JDBC - API que permite a ligação entre a linguagem Java e uma base de dados.

Índice

Siglas e Acrónimos	2
Glossário	3
Índice	4
1. Introdução	5
1.1 Enquadramento	5
1.2 Apresentação do projeto	5
1.3 Fundamentação da escolha do projeto	5
1.4 Finalidades	5
1.5 Tecnologias utilizadas	6
2. Modelo lógico	7
3. Diagrama de Classes	8
4. Modelo Físico	9
5. Desenvolvimento	10
6. Conclusão	18

1. Introdução

1.1 Enquadramento

O projeto consiste em desenvolver uma aplicação GUI em Java com ligação a uma base de dados, cuja interface gráfica é programada visualmente com o JavaFX.

1.2 Apresentação do projeto

O projeto consiste em desenvolver uma aplicação com a ajuda da linguagem de programação Java, com uma interface gráfica acessível ao utilizador e com uma ligação estável à base de dados. A aplicação deve conter as operações básicas de um sistema CRUD (Create, Read, Update, Delete), acessíveis visualmente. Neste caso, o sistema permitirá a gestão de livros numa biblioteca.

1.3 Fundamentação da escolha do projeto

A escolha deste projeto baseia-se nos conhecimentos e na experiência adquiridos na área da programação orientada a objetos (POO), assim como na manipulação de Sistemas de Gestão de Base de Dados (SGBD) que funcionam em conjunto com esta abordagem. Além disso, o domínio desta área é essencial na maioria dos setores do mercado de trabalho ligados aos sistemas informáticos. Este conhecimento permite aos alunos evoluir em vários aspetos profissionais e pode abrir portas para um futuro mais promissor.

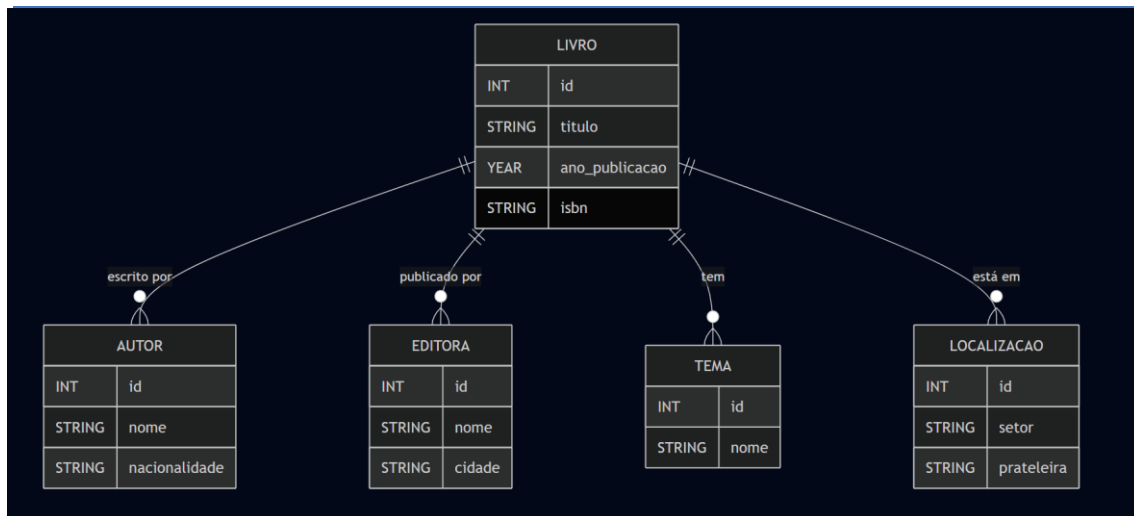
1.4 Finalidades

O principal objetivo deste projeto é demonstrar, de forma prática, o funcionamento de uma aplicação orientada a objetos com interface gráfica (GUI), interligada a um sistema de gestão de base de dados (SGBD). Além disso, pretende ser uma ferramenta de aprendizagem sólida que os alunos poderão aplicar em contextos reais no mundo profissional.

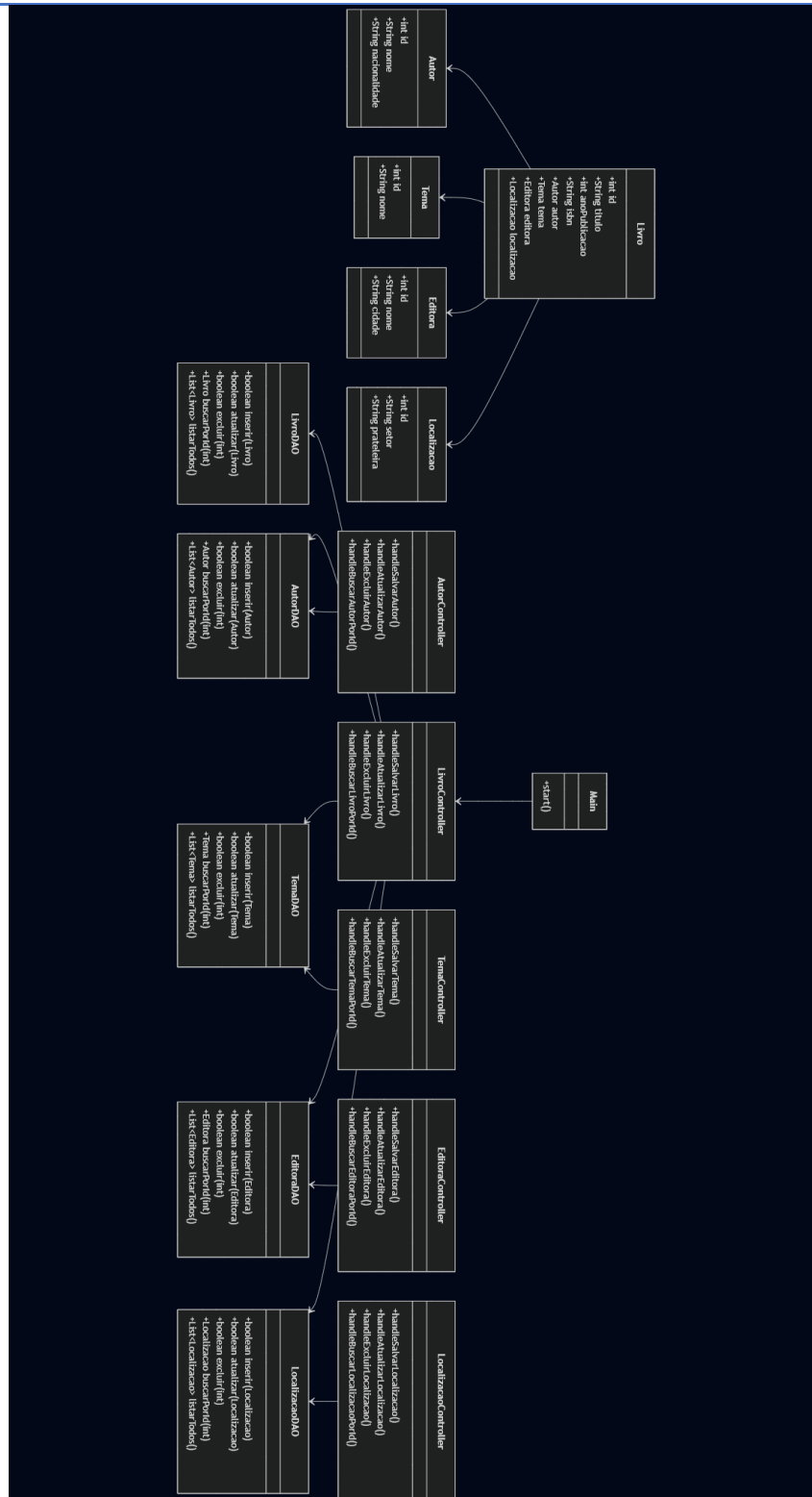
1.5 Tecnologias utilizadas

As tecnologias utilizadas neste projeto foram: a linguagem de programação Java; o MySQL Workbench como sistema de gestão de base de dados (SGBD); o driver JDBC para estabelecer a ligação entre o Java e a base de dados; e a biblioteca JavaFX para a criação da interface gráfica.

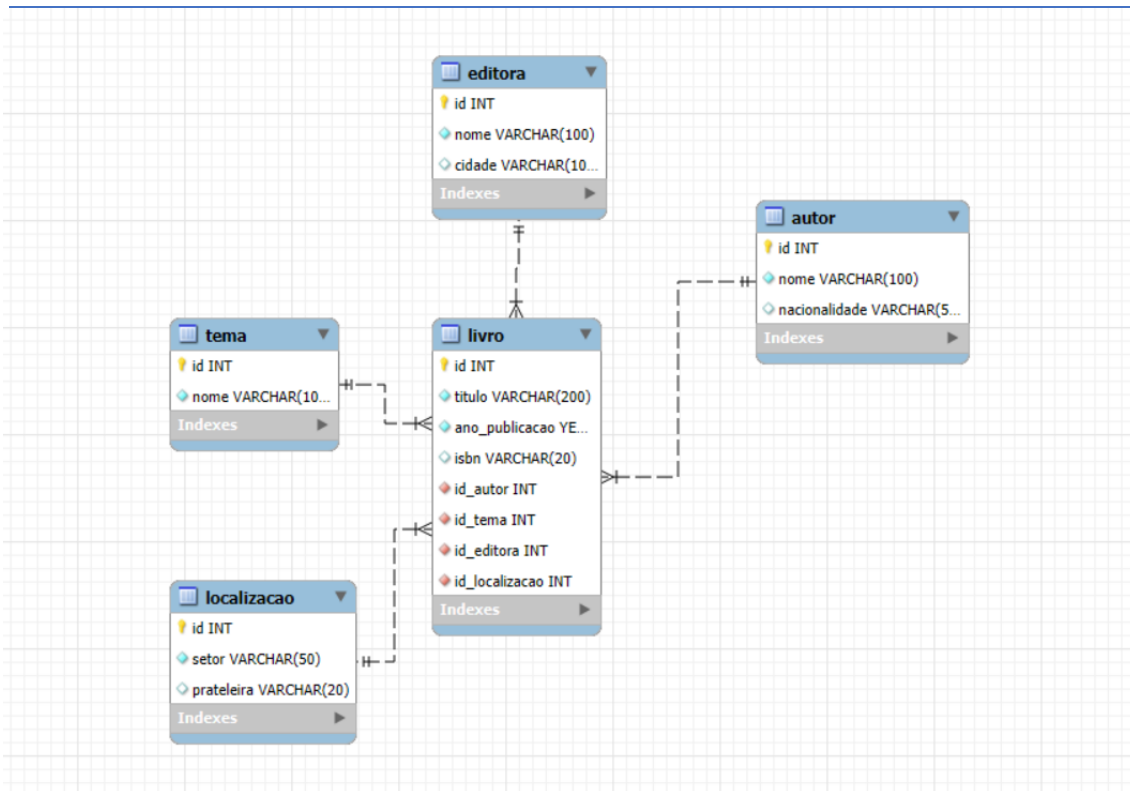
2. Modelo lógico



3. Diagrama de Classes



4. Modelo Físico



5. Desenvolvimento

O seguinte projeto apresenta o seguinte sistema: CRUD de gerenciamento de livros de uma Biblioteca, ou Verbox.

O Verbox tem a seguinte estrutura:

```
Verbox/
├── src/
│   ├── application/
│   │   └── Main.java - Classe principal que inicia a aplicação JavaFX.
│   ├── controller/
│   │   ├── AutorController.java
│   │   ├── LivroController.java
│   │   ├── TemaController.java
│   │   ├── EditoraController.java
│   │   └── LocalizacaoController.java
│   │       → Controladores da lógica da interface gráfica (FXML).
│   ├── dao/
│   │   ├── AutorDAO.java
│   │   ├── LivroDAO.java
│   │   ├── TemaDAO.java
│   │   ├── EditoraDAO.java
│   │   └── LocalizacaoDAO.java
│   │       → Camada de acesso a dados (CRUD com MySQL).
│   ├── model/
│   │   ├── Autor.java
│   │   ├── Livro.java
│   │   ├── Tema.java
│   │   ├── Editora.java
│   │   └── Localizacao.java
│   │       → Entidades com atributos e métodos.
│   ├── util/
│   │   └── DBConnection.java - Ligação com a base de dados MySQL.
│   └── view/
│       ├── AutorView.fxml
│       ├── LivroView.fxml
│       ├── TemaView.fxml
│       ├── EditoraView.fxml
│       └── LocalizacaoView.fxml
│           → Interfaces gráficas criadas com Scene Builder.
```

Esta estrutura segue o padrão MVC (Model - View - Controller), organizado de forma limpa e profissional.

MVC significa Model-View-Controller e é um padrão de arquitetura de software muito utilizado no desenvolvimento de aplicações como esta.

O seu objetivo é organizar o código separando responsabilidades em três camadas principais:

Model (Modelo):

- Representa os dados e a lógica de negócio da aplicação.
- Contém as classes de entidade, como Livro, Autor, Editora, etc.
- É responsável por:
 - A estrutura dos dados
 - As regras de negócio
 - Eventualmente a ligação com a base de dados (através dos DAOs)

Exemplo no Verbox:

Livro.java, Autor.java, Tema.java

View (Visão ou Interface):

- É a interface gráfica (GUI) da aplicação, com a qual o utilizador interage.
- Mostra dados ao utilizador e recebe as suas ações.
- No JavaFX, a View é definida em ficheiros .fxml criados com Scene Builder.

Exemplo no Verbox:

LivroView.fxml, AutorView.fxml, TemaView.fxml

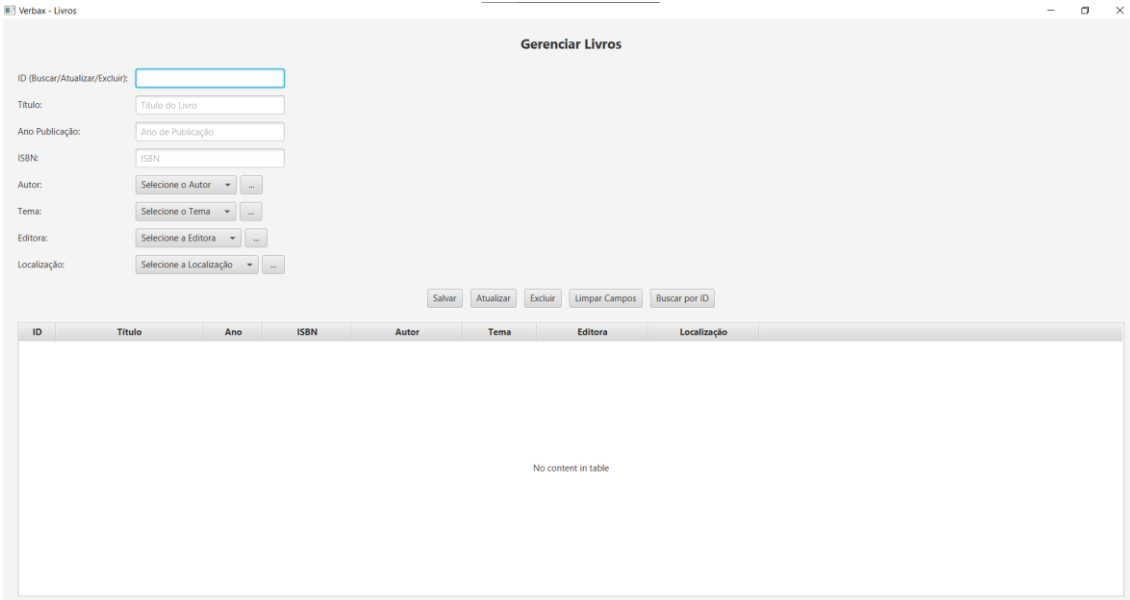
Controller (Controlador):

- Interpreta as ações do utilizador (ex: clicar num botão).
- Comunica com o Model para atualizar dados.
- Atualiza a View com os resultados.

Exemplo no Verbox:

LivroController.java, AutorController.java

Interface da aplicação é vista da seguinte forma:



Gerenciar Livros

ID (Buscar/Atualizar/Excluir):

Título:

Ano Publicação:

ISBN:

Autor:

Tema:

Editora:

Localização:

ID	Título	Ano	ISBN	Autor	Tema	Editora	Localização
No content in table							

Na aplicação é possível adicionar diretamente os livros a partir dos modelos de Autor, Tema, Editora e Localização. Estes modelos têm a sua própria exclusividade em adicionar os seus registos.

Antes de adicionar um livro, é necessário ter, pelo menos, um registo em cada modelo referente anteriormente. Se não, retorna mensagem de erro.

Verbase - Livros

Gerenciar Livros

ID (Buscar/Atualizar/Excluir):

Título:

Ano Publicação:

ISBN:

Autor:

Selecione o Autor

...

Tema:

Selecione o Tema

...

Editora:

Selecione a Editora

...

Localização:

Selecione a Localização

...

Campos Incompletos

Por favor, preencha todos os campos e selecione as opções.

OK

Salvar

Atualizar

Excluir

Limpar Campos

Buscar por ID

ID	Título	Ano	ISBN	Autor	Tema	Editora	Localização
No content in table							

[illegible]

[illegible]

[illegible]

Ao inserir cada registo em cada modelo, agora sim é possível inserir registo de livro.

Verbaex - Livros

Gerenciar Livros

ID (Buscar/Atualizar/Excluir):

ID do Livro

Título:

Star Wars

Ano Publicação:

1999

ISBN:

123456789

Autor:

Michael

...

Tema:

Fantasia

...

Editora:

Porto Editora

...

Localização:

Fantasia - 10E

...

Sucesso

Livro salvo com sucesso!

OK

Salvar

Atualizar

Excluir

Limpar Campos

Buscar por ID

ID	Título	Ano	ISBN	Autor	Tema	Editora	Localização
No content in table							

[illegible]

As implementações de CRUD também estão em funcionamento.

[illegible]

Verbas - Livros

Gerenciar Livros

ID (Buscar/Atualizar/Excluir):

ID do Livro

Título:

Título do Livro

Ano Publicação:

Ano de Publicação

ISBN:

ISBN

Autor:

Selecione o Autor

Tema:

Selecione o Tema

Editora:

Selecione a Editora

Localização:

Selecione a Localização

Salvar

Atualizar

Excluir

Limpar Campos

Buscar por ID

ID	Título	Ano	ISBN	Autor	Tema	Editora	Localização
No content in table							

Todos estes registos aparecem dinamicamente na base de dados via MySQL Workbench.

6. Conclusão

O desenvolvimento do Verbax mostrou-se numa experiência interessante. Ao longo do processo, foi possível aplicar conhecimentos nas áreas de programação orientada a objetos (POO), ligação a bases de dados, construção de interfaces gráficas com JavaFX e adoção do padrão MVC, em que promove uma estrutura de código limpa e estruturada.

A aplicação desenvolvida atinge os objetivos estabelecidos, nomeadamente a gestão completa de livros, autores, editoras, temas e localizações, através de operações CRUD acessíveis ao utilizador por meio de uma interface gráfica.

De forma geral, este projeto proporcionou uma excelente base para perceber o funcionamento de uma aplicação real, em que reforça competências essenciais tanto para o percurso académico como para o mundo profissional.

NOTA: O relatório poderia estar melhor.