

UniFecaf – Taboão da Serra

Estudo de Caso – Object Oriented Programming

ADS 3º Semestre

Luciano Boaventura Bastos

73399

2025

Solução para o Problema

O sistema foi projetado utilizando os princípios de Programação Orientada a Objetos (POO) e segue o modelo CRUD (Create, Read, Update, Delete). A solução abrange as seguintes áreas principais:

- 1. Diagrama de Classes: Representação das entidades do sistema e suas relações.
- 2. Justificativa das Escolhas: Explicação das decisões de design adotadas.

Diagrama de Classes do Sistema de Biblioteca

Classe: Livro

Atributos Tipo

id	Long
titulo	String
autor	String
genero	String
ano	Integer

Métodos	Retorno
getId()	Long
getTitulo()	String
getAutor()	String
getGenero()	String
getAno()	Integer
setTitulo(String titulo)	void
setAutor(String autor)	void
setGenero(String genero)	void
setAno(Integer ano)	void

Classe: LivroService

Métodos	Retorno
listarTodos()	List
buscarPorId(Long id)	Livro
adicionarLivro(Livro livro)	Livro
atualizarLivro(Long id, Livro)	Livro
deletarLivro(Long id)	void

Classe: LivroController

Endpoint	Verbo HTTP	Ação
/livros	GET	Lista todos os livros
/livros/{id}	GET	Retorna os detalhes de um livro
/livros	POST	Adiciona um novo livro
/livros/{id}	PUT	Atualiza os dados de um livro
/livros/{id}	DELETE	Remove um livro

Relacionamentos

1. LivroController usa LivroService para delegar operações.
2. LivroService manipula dados da classe Livro.
3. Livro é a classe de modelo conectada ao banco de dados.

Justificativa das Escolhas

1. Encapsulamento:
 - Todos os atributos são privados e acessíveis apenas por métodos getters e setters para proteger os dados.
2. Modularização:

- A separação entre LivroController, LivroService e Livro segue o padrão MVC (Model-View-Controller), garantindo um código organizado e escalável.

3. Reutilização:

- Os métodos CRUD do LivroService e do LivroController foram implementados para reutilização em diferentes partes do sistema.

4. Tecnologias Utilizadas:

- **Spring Boot:** Framework para facilitar a criação de aplicações Java.
- **React:** Biblioteca JavaScript para criação de interfaces de usuário dinâmicas.
- **MySQL:** Banco de dados relacional para armazenamento seguro e eficiente dos dados.

5. Padrões de Projeto:

- Seguiu-se o padrão MVC para garantir uma separação clara entre as camadas de lógica de negócios, controle e modelo.

Conclusão

Este sistema de gerenciamento de biblioteca atende ao desafio proposto, oferecendo funcionalidades completas de CRUD, integração com banco de dados e uma interface amigável para o usuário. Todas as escolhas de design e tecnologia foram feitas para garantir escalabilidade, manutenção e usabilidade do sistema.