

## **MINI-PROJETO U1:**

Sistema web backend para administrar empréstimos em uma biblioteca.

Universidade Federal do Rio Grande do Norte

Disciplina: Desenvolvimento de Sistemas WEB II

Professor: Jean Mário Moreira de Lima

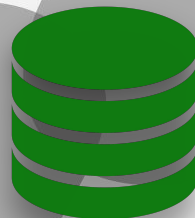
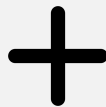
Aluno: André Augusto Fernandes

# 1. Proposta



Desenvolver um sistema web backend para administrar empréstimos em uma biblioteca, utilizando:

- Framework Spring
- Thymeleaf
- PostgreSQL
- Spring Data JPA com JpaRepository

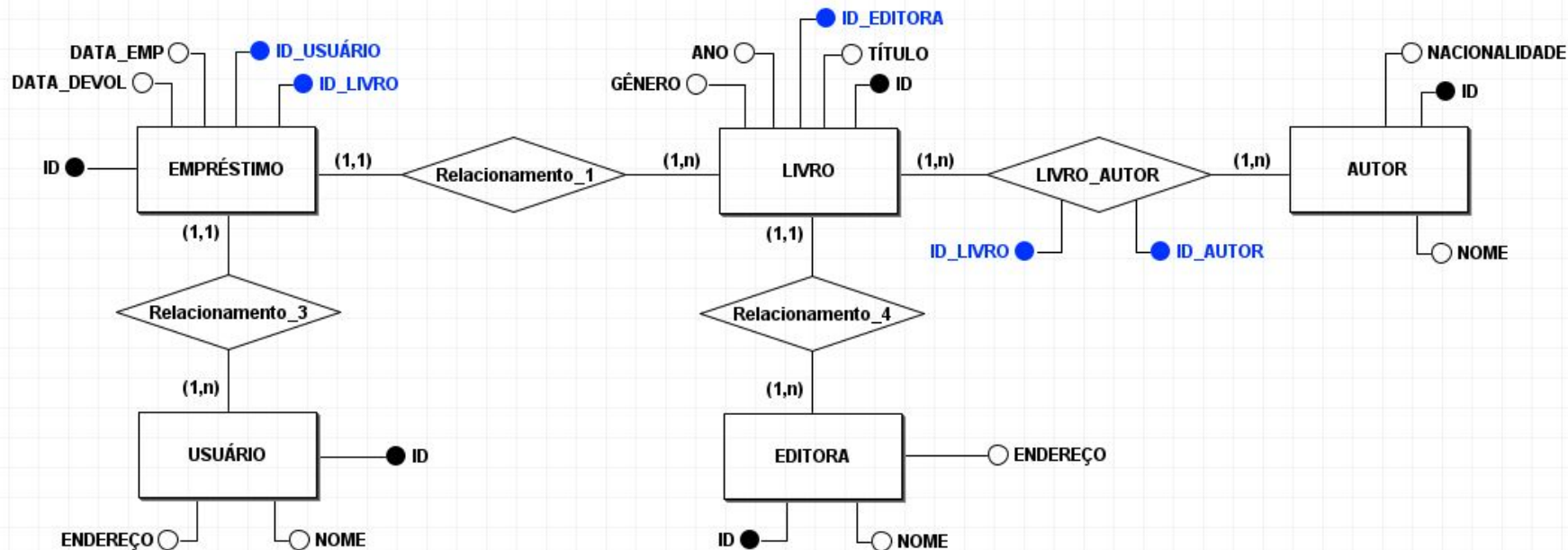


## 2. Mapeamento Objeto-Relacional



Para o mapeamento objeto-relacional, foram criadas as seguintes entidades:

### SISTEMA DE BIBLIOTECA



### 3. Arquitetura MVC

Para a arquitetura MVC, foram criados os seguintes pacotes:

#### Model:

- **Entidades:** Responsável por representar as entidades do banco de dados.
- **Service:** Responsável por desacoplar a aplicação das classes repositórios ou de regras de negócio.
- **Repository:** Responsável por fazer a persistência dos dados.

**View:** É a camada de apresentação do sistema, onde o usuário interage com a aplicação.

**Controller:** Responsável por fazer a ponte entre a camada view e a camada model.

#### Model

model

- Autor.java
- Editora.java
- Emprestimo.java
- Livro.java
- Usuario.java

service

- AutorService.java
- EditoraService.java
- EmprestimoService.java
- LivroService.java
- UsuarioService.java

repository

- AutorRepository.java
- EditoraRepository.java
- EmprestimoRepository.java
- LivroRepository.java
- UsuarioRepository.java

#### Controller

controller

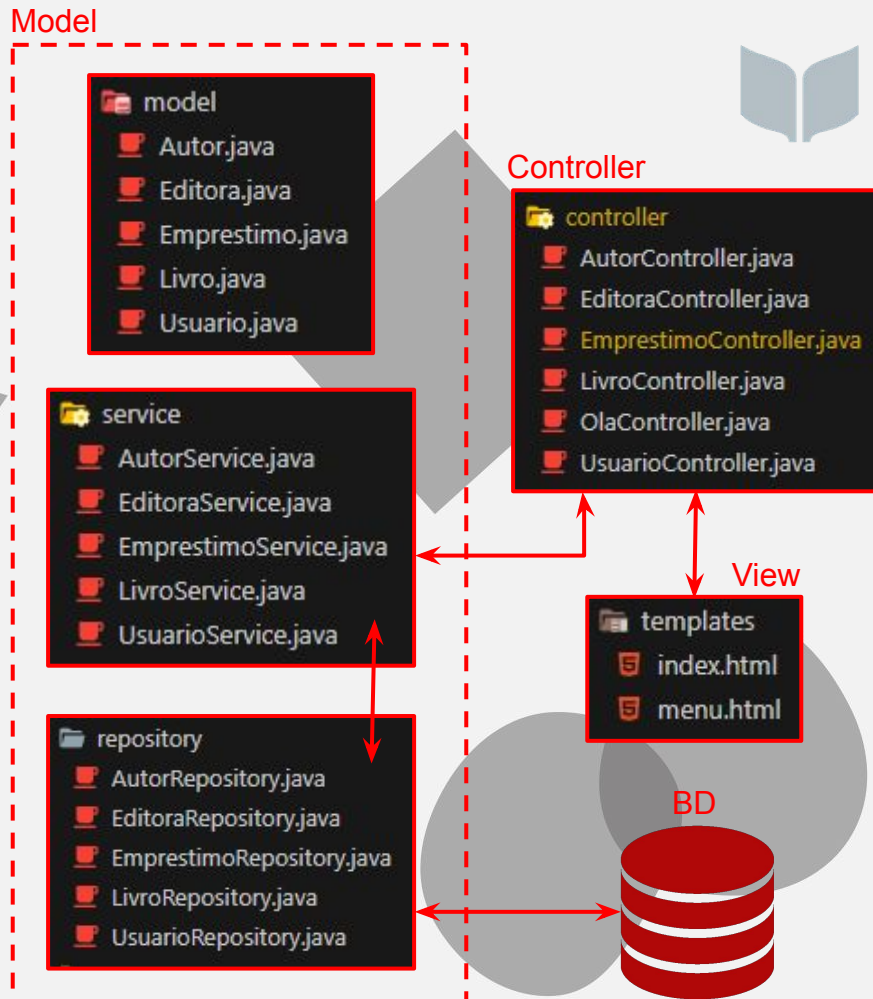
- AutorController.java
- EditoraController.java
- EmprestimoController.java
- LivroController.java
- OlaController.java
- UsuarioController.java

#### View

templates

- index.html
- menu.html

#### BD



## 4. Frontend

The logo for BiblioTech, featuring the text "BiblioTech" in a white sans-serif font, followed by a stylized icon of an open book with two pages.

### menu.html

Esta página (imagem acima) é criada por padrão pelo Spring Boot e é uma espécie de porta de entrada para a aplicação.

Ao clicar no logo da aplicação, o usuário será direcionado para index.html

### index.html

Para exibir as informações do sistema foi utilizada uma abordagem de página única, na qual o trecho de html a ser exibido é determinado por uma informação passada pelo controller e manipulada pelo Thymeleaf, que utilizando a diretiva `th:switch`, exibe o código desejado.

The logo for BiblioTech, featuring the text "BiblioTech" in a white sans-serif font, followed by a stylized icon of an open book with two pages.

[Autor](#)   [Editora](#)   [Empréstimo](#)   [Livro](#)   [Usuario](#)   [Sobre](#)

### Bem vindo ao BiblioTech 📖

Sistema para administração de bibliotecas.

Escolha uma opção no menu acima para começar.

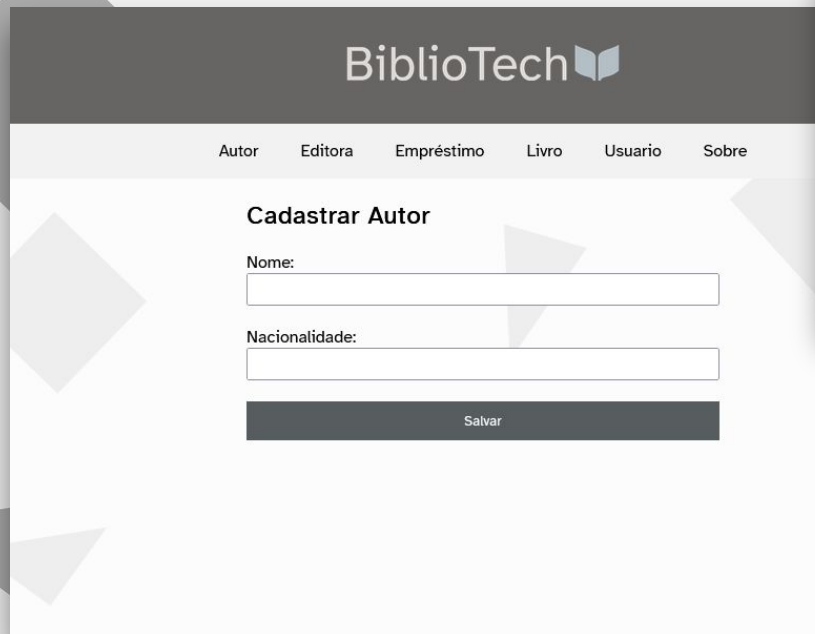
Orientações:

- O autor precisa ser/estar cadastrado antes que seu livro o seja.
- O usuário precisa ser/estar cadastrado antes de realizar o empréstimo.

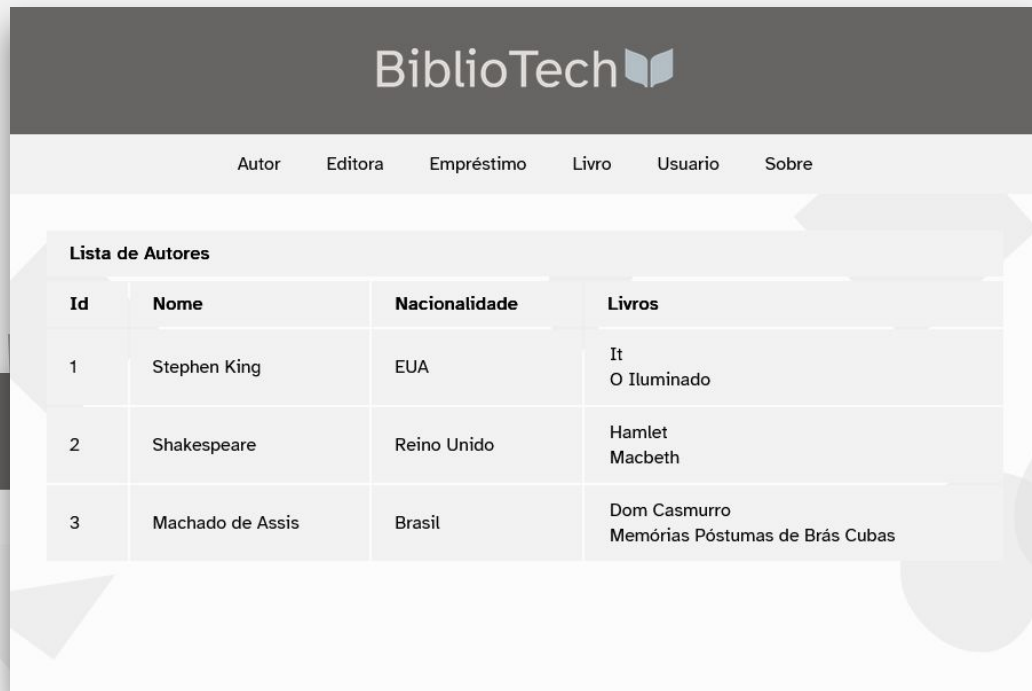
## 5. Operações CRUD

Para as operações CRUD, foram implementadas as seguintes funcionalidades:

**Create:** Cadastrar livro, autor (imagem abaixo), editora, empréstimo ou usuário.



The screenshot shows the 'Cadastrar Autor' (Register Author) form in the BiblioTech application. The form is located below a navigation bar with links: Autor, Editora, Empréstimo, Livro, Usuario, and Sobre. The form has two input fields: 'Nome:' and 'Nacionalidade:'. Below these fields is a dark gray button labeled 'Salvar'.



The screenshot shows the 'Lista de Autores' (List of Authors) table in the BiblioTech application. The table is located below a navigation bar with links: Autor, Editora, Empréstimo, Livro, Usuario, and Sobre. The table has four columns: Id, Nome, Nacionalidade, and Livros. It contains three rows of data.

Id	Nome	Nacionalidade	Livros
1	Stephen King	EUA	It O Iluminado
2	Shakespeare	Reino Unido	Hamlet Macbeth
3	Machado de Assis	Brasil	Dom Casmurro Memórias Póstumas de Brás Cubas

**Read:** Listar livros, autores (imagem acima), editoras, empréstimos ou usuários.

## 5. Operações CRUD

**Update:** Devolver empréstimo.

Ao clicar o botão **Devolver** (imagem abaixo), o sistema atualiza o estado do atributo “devolvido” da entidade “empréstimo” para TRUE.

BiblioTech

Autor Editora Empréstimo Livro Usuario Sobre

### Lista de Empréstimos

Id	Usuario	Livro	Data de Empréstimo	Data de Devolução	Devolvido
2	Maria	Hamlet	2024-04-08		Não <button>Devolver</button>
3	José	Dom Casmurro	2024-04-08		Não <button>Devolver</button>
1	João	It	2024-04-08	2024-04-08	Sim
4	João	It	2024-04-08		Não <button>Devolver</button>

BiblioTech

Autor Editora Empréstimo Livro Usuario Sobre

### Lista de Usuários

Id	Nome	Email
	João	<button>Excluir</button> joao@email.com
	Maria	<button>Excluir</button> maria@email.com
	José	<button>Excluir</button> jose@email.com
	Ana	<button>Excluir</button> ana@email.com
	Pedro	<button>Excluir</button> pedro@email.com

**Delete:** Excluir usuário.

Ao clicar em **Excluir** (imagem acima), o sistema excluirá o usuário do banco de dados e todos os empréstimos a ele associados.



## 6. Queries personalizadas

```
// recupera lista de autores do banco de dados  
@Query(value = "select a from Autor a")  
List<Autor> listarAutores();
```

```
// recupera lista de empréstimos ativos  
@Query(value = "select e from Empréstimo e where e.devolido = false")  
List<Empréstimo> listarEmpréstimosAtivos();
```





## 7. Personalização e Usabilidade

**CSS:** CSS puro foi utilizado para estilizar as páginas.

**JavaScript:** JavaScript foi utilizado no frontend apenas para exigir a confirmação do usuário ao realizar operações de Update e Delete.

**Thymeleaf:** esta ferramenta foi bastante utilizada para a manipulação de dados, comunicação com o controller e renderização de páginas.



## 8. Conclusões

- **spring boot** se mostrou uma ferramenta poderosa para o desenvolvimento de sistemas web backend, pois facilita a configuração e o desenvolvimento das aplicações.
- **Thymeleaf** também se mostrou uma ferramenta bastante útil, pois diminui significativamente a quantidade de código que precisa ser escrito para manipular os dados e renderizar as páginas.
- **spring data jpa** facilitou a comunicação com o banco de dados, pois introduz uma camada de abstração que dispensa a necessidade de escrever queries SQL.



**FIM**