

## [Lab04] Repositório de Livros JPA

Você foi convidado por um amigo apaixonado por leitura para criar um pequeno sistema que ajude a organizar o seu acervo pessoal de livros. Ele deseja cadastrar os livros que possui, incluindo informações como título, autor, ano de publicação e editora. Além disso, ele quer poder realizar buscas rápidas, por exemplo, localizar todos os livros escritos por um determinado autor, verificar quais livros foram publicados em um determinado ano, ou até mesmo pesquisar livros que contenham um termo específico no título.

Para tornar essa tarefa eficiente, você decidiu utilizar **Java com Spring Boot e Spring Data JPA**. Por se tratar de um projeto simples, você optou por uma interface de console (sem interface gráfica ou API REST).

Seu desafio será criar uma aplicação que permita:

- Cadastrar novos livros no sistema;
- Visualizar a lista completa dos livros cadastrados;
- Realizar buscas específicas nos dados cadastrados.

## Funcionalidades

### 1. Cadastrar um Novo Livro

O usuário poderá cadastrar novos livros no acervo, informando os seguintes dados:

- **Título do livro** (ex.: *"A Revolução dos Bichos"*)
- **Autor** (ex.: *"George Orwell"*)
- **Ano de publicação** (ex.: *1945*)
- **Editora** (ex.: *"Companhia das Letras"*)

Exemplo:

```
[Cadastro de Livro]
Digite o título: A Revolução dos Bichos
Digite o autor: George Orwell
Digite o ano de publicação: 1945
Digite a editora: Companhia das Letras
```

```
Livro cadastrado com sucesso!
```

## 2. Listar Todos os Livros Cadastrados

O usuário poderá visualizar todos os livros cadastrados, exibindo suas informações organizadamente.

Exemplo:

[Listagem Completa do Acervo]

| ID  | Título                 | Autor            | Ano  | Editora              |
|-----|------------------------|------------------|------|----------------------|
| 1   | A Revolução dos Bichos | George Orwell    | 1945 | Companhia das Letras |
| 2   | Dom Casmurro           | Machado de Assis | 1899 | Ática                |
| ... |                        |                  |      |                      |

## 3. Buscar Livros por Autor

Permite ao usuário localizar rapidamente todos os livros de um determinado autor.

Exemplo:

[Busca por Autor]

Digite o nome do autor: Machado de Assis

Livros encontrados:

- Dom Casmurro (1899, Ática)
- Memórias Póstumas de Brás Cubas (1881, Tipografia Nacional)

## 4. Buscar Livros por Ano de Publicação

Permite ao usuário listar todos os livros lançados em determinado ano.

Exemplo:

[Busca por Ano de Publicação]

Digite o ano desejado: 1945

Livros publicados em 1945:

- A Revolução dos Bichos, por George Orwell (Companhia das Letras)
- Vidas Secas, por Graciliano Ramos (Record)

## 5. Buscar Livros por Termo no Título

Permite ao usuário encontrar livros que possuam um termo específico no título, ignorando maiúsculas e minúsculas.

Exemplo:

```
[Busca por Termo no Título]
```

```
Digite o termo desejado: revolução
```

```
Livros encontrados:
```

- A Revolução dos Bichos, por George Orwell (1945)
- A Revolução Francesa, por Eric Hobsbawm (1962)

Após cada operação, o usuário retornará automaticamente ao menu principal para continuar interagindo com o sistema ou poderá optar por encerrar a aplicação.

## Requisitos Técnicos

### Tecnologias e Ferramentas Necessárias:

- Java (JDK versão 17 ou superior)
- Spring Boot (versão estável mais recente)
- Spring Data JPA
- Banco de dados H2 (em memória) ou Supabase
- Maven (gerenciador de dependências)
- IDE de sua preferência (IntelliJ, Eclipse ou VS Code com plugins Java). Usamos github Codespace nas aulas

### Estrutura do Projeto:

O projeto deverá seguir a estrutura padrão do Spring Boot:

```
src
├── main
│   └── java
│       └── br
│           └── seu_nome
│               └── acervo
│                   ├── AcervoApplication.java
│                   ├── entidade
│                   │   └── Livro.java
│                   ├── repositorio
│                   │   └── LivroRepository.java
│                   └── aplicacao
│                       └── ConsoleApp.java
```

## Requisitos Técnicos Obrigatórios:

- Criar a classe `Livro` como uma entidade JPA com atributos:
  - `id` (Long, chave primária com auto incremento)
  - `titulo` (String)
  - `autor` (String)
  - `anoPublicacao` (int)
  - `editora` (String)

## Repositório:

- Criar uma interface `LivroRepository` que estenda a `JpaRepository`.
- Implementar métodos de busca personalizados usando a convenção do Spring Data:
  - Busca por autor (`findByAutor`)
  - Busca por ano de publicação (pesquisar como implementar)
  - Busca por termo no título, ignorando maiúsculas/minúsculas (pesquisar como implementar)
  - Método para verificar se existe um livro com mesmo título e autor (pesquisar como implementar)

## Aplicação Console:

- Criar uma classe `ConsoleApp` com método que implementa um menu interativo via console para que o usuário utilize as funcionalidades:
  - Cadastrar livros

- Listar livros
- Realizar buscas específicas (autor, ano e título)

### Banco de Dados:

- Utilizar o H2 (banco em memória) já configurado pelo Spring Boot, sem necessidade de configuração adicional pelo aluno, ou utilizar banco de dados Supabase online

### Código Base Inicial (apenas o essencial):

#### Classe principal ( `AcervoApplication.java` ):

```
package br.seu_nome.acervo;

import br.seu_nome.acervo.aplicacao.ConsoleApp;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class AcervoApplication implements CommandLineRunner {
    public static void main(String[] args) {
        SpringApplication.run(AcervoApplication.class, args);
    }

    @Override
    public void run(String... args) {
        ConsoleApp consoleApp
        consoleApp.iniciar();
    }
}
```

Exemplo simplificado da classe `ConsoleApp.java` (para referência):

```

package br.seu_nome.acervo.aplicacao;

import org.springframework.stereotype.Component;

public class ConsoleApp {

    public void iniciar() {
        System.out.println("Aplicação console iniciada!");
        // Lógica do menu aqui
    }
}

```

## Orientações para entrega no Moodle

### 1. Código-fonte completo

Inclua no arquivo ZIP o projeto completo (pasta raiz do projeto Spring Boot), com:

- Arquivos `.java` contendo a aplicação implementada (entidades, repositórios, aplicação console).
- Arquivo de configuração `pom.xml`.
- Demais arquivos gerados automaticamente pelo Spring Boot.

```

seu_nome_acervo.zip
├── seu_nome_acervo
│   ├── pom.xml
│   └── src
│       ├── main
│       │   └── java
│       │       ├── br
│       │       │   ├── seu_nome
│       │       │   │   └── acervo
│       │       │       ├── AcervoApplication.java
│       │       │       ├── aplicacao
│       │       │       │   └── ConsoleApp.java
│       │       ├── entidade
│       │       │   └── Livro.java
│       │       └── repositorio
│       │           └── LivroRepository.java

```

**Obs:** **Não enviar** arquivos compilados (ex.: `target/`). Envie apenas os arquivos fonte.