

## ## 🎯 Especificação do Exercício — API REST: **Biblioteca Escolar**

### ### 🏠 Objetivo

Desenvolver uma API REST em **Spring Boot** para gerenciar uma **Biblioteca Escolar**, permitindo o cadastro e a consulta de **Livros** e seus respectivos **Autores**. A API deverá implementar validações nos dados de entrada, utilizando anotações de ``javax.validation``.

---

## ## 📁 Entidades

### ### 📖 **Livro**

- \* ``id`` (Long) — Gerado automaticamente
- \* ``titulo`` (String) — Obrigatório, mínimo 2 e máximo 150 caracteres
- \* ``isbn`` (String) — Obrigatório, deve ter exatamente 13 caracteres (validação de tamanho)
- \* ``anoPublicacao`` (Integer) — Obrigatório, valor mínimo 1500 e máximo o ano atual
- \* ``autor`` (Autor) — Obrigatório (relacionamento muitos-para-um)

### ### 👤 **Autor**

- \* ``id`` (Long) — Gerado automaticamente
- \* ``nome`` (String) — Obrigatório, mínimo 2 e máximo 100 caracteres
- \* ``email`` (String) — Obrigatório, formato de e-mail válido
- \* ``nacionalidade`` (String) — Opcional, máximo 50 caracteres

---

## ## 🔗 Relacionamento

- \* Um **Autor** pode ter vários **Livros** (1:N).
- \* Cada **Livro** está associado a exatamente um **Autor**.

---

## ## 🔥 Funcionalidades obrigatórias da API

### ### 📄 **Endpoints para Autores**

- \* ``POST /autores`` — Cadastrar um autor
- \* ``GET /autores`` — Listar todos os autores
- \* ``GET /autores/{id}`` — Buscar um autor por ID
- \* ``PUT /autores/{id}`` — Atualizar dados do autor
- \* ``DELETE /autores/{id}`` — Remover um autor (só permitido se não houver livros associados)

### ### 📖 \*\*Endpoints para Livros\*\*

- \* `POST /livros` — Cadastrar um livro
- \* `GET /livros` — Listar todos os livros
- \* `GET /livros/{id}` — Buscar um livro por ID
- \* `PUT /livros/{id}` — Atualizar dados do livro
- \* `DELETE /livros/{id}` — Remover um livro

---

### ## ✅ \*\*Regras de Validação\*\*

- \* Todos os campos obrigatórios devem ser validados utilizando `@NotNull`, `@NotBlank`, `@Size`, `@Email` e `@Min`/`@Max` conforme necessário.
- \* O campo `anoPublicacao` não pode ser anterior a 1500 e nem superior ao ano atual (validado com `@Max` dinamicamente ou lógica no service).
- \* O `isbn` deve ter exatamente 13 caracteres (`@Size(min = 13, max = 13)`).
- \* O nome do autor e título do livro devem ter tamanho mínimo e máximo.
- \* O autor não pode ser removido se houver livros vinculados a ele (validação no serviço).

---

### ## 🛠️ Requisitos Técnicos

- \* Framework: **Spring Boot**
- \* Dependências:
  - \* `spring-boot-starter-web`
  - \* `spring-boot-starter-validation`
  - \* `spring-boot-starter-data-jpa`
  - \* Banco de dados H2 (ou PostgreSQL)
- \* Utilizar DTOs para entrada de dados, aplicando validações nas classes DTO.
- \* Documentação dos endpoints com Swagger (opcional, recomendado).

---

### ## 💡 Desafio Extra (opcional)

- \* Adicionar endpoint `GET /autores/{id}/livros` para listar todos os livros de um autor específico.
- \* Implementar paginação no endpoint de listagem de livros.
- \* Implementar busca de livros por título (`GET /livros/search?titulo=`)