

Luan G. F. Ricardo

## Documentação da API REST - Node.js, Express e MongoDB

### 1. Visão Geral

Este projeto consiste em uma API REST desenvolvida com Node.js, Express e MongoDB, estruturada com os princípios de separação de responsabilidades, utilizando os padrões de camadas: Controllers, Services, Repositories, Models e Middlewares. A API oferece recursos de autenticação JWT, controle de usuários, gerenciamento de autores e postagens, além de documentação técnica via Swagger.

### 2. Requisitos do Ambiente

- Node.js 18.x ou superior
- MongoDB 6.x ou superior (local ou Atlas)
- npm ou yarn

### 3. Instalação

Clonar o repositório e instalar as dependências:

```
git clone https://github.com/LuanGFRicardo/kanban
```

```
npm install
```

### 4. Execução

Para iniciar o servidor localmente:

```
npm start
```

A API estará disponível por padrão em:  
`http://localhost:3000`

A documentação Swagger estará disponível em:  
`http://localhost:3000/api-docs`

### 6. Endpoints Disponíveis

## 6.1 Autenticação

Método	Rota	Descrição
POST	/auth/login	Autentica usuário via e-mail e senha e retorna um token JWT

## 6.2 Usuários

Método	Rota	Descrição
GET	/users	Lista todos os usuários
POST	/users	Cria um novo usuário
GET	/users/:id	Busca usuário pelo ID
PUT	/users/:id	Atualiza dados de um usuário
DELETE	/users/:id	Remove usuário
GET	/users/search	Busca usuários por nome

## 6.3 Autores

Método	Rota	Descrição
GET	/authors	Lista todos os autores
POST	/authors	Cria um novo autor
GET	/authors/:id	Busca autor pelo ID
PUT	/authors/:id	Atualiza dados de um autor
DELETE	/authors/:id	Remove autor
GET	/authors/search	Busca autores por nome

## 6.4 Posts

Método	Rota	Descrição
GET	/posts	Lista todas os posts

POST	/posts	Cria uma novo post
GET	/posts/:id	Busca post pelo ID
PUT	/posts/:id	Atualiza dados de um post
DELETE	/posts/:id	Remove post
GET	/posts/search	Busca postagens por palavras-chave (título ou descrição)

## 8. Documentação da API (Swagger)

A documentação Swagger está disponível no endpoint:

GET /api-docs

Este recurso fornece detalhes de todos os endpoints da API, parâmetros, respostas e exemplos, seguindo o padrão OpenAPI 3.0.

O arquivo de configuração encontra-se em:

src/config/swagger.js

## 9. Considerações Técnicas

- O padrão adotado para arquitetura da aplicação é baseado na separação de responsabilidades: Controllers, Services, Repositories.
- Os dados recebidos e retornados são tratados por meio de DTOs.
- A autenticação é baseada em tokens JWT e middleware de proteção para rotas privadas.
- A camada de repositório faz uso do Mongoose para interação com o banco MongoDB.
- A API foi projetada para ser escalável e manter baixo acoplamento entre componentes.