



Universidade Federal do Ceará
Campus Quixadá
QXD0020 - Desenvolvimento de Software para Web
Prof. Francisco Victor da Silva Pinheiro

Exercício 3 - Plataforma de Comunidade com API REST

Enunciado: Implemente uma API RESTful utilizando FastAPI, SQLite e Auth0, responsável pelo backend de uma Plataforma de Comunidade Online. A API será consumida por uma aplicação web (frontend não implementado neste exercício) e deverá oferecer autenticação segura, controle de acesso por perfil, gestão de conteúdo e mecanismos de busca e paginação. O foco do exercício é integrar conceitos de desenvolvimento web, segurança com JWT, persistência de dados e boas práticas de backend.

1. Estrutura e Tecnologias

A aplicação deve utilizar obrigatoriamente:

- FastAPI para construção da API
- SQLite como banco de dados
- Auth0 para autenticação e autorização (JWT)
- Pydantic para validação de dados
- SQLAlchemy ou SQLModel para mapeamento objeto-relacional
- Organização do projeto em módulos (routers, models, schemas, auth, database)

2. Autenticação e Perfis de Usuário

2.1 Autenticação

- A API deve validar tokens JWT emitidos pelo Auth0
- O token deve ser enviado no header:

Authorization: Bearer <token>

- O identificador do usuário (sub) deve ser extraído do token e utilizado internamente

2.2 Perfis de Usuário

A aplicação deve trabalhar com os seguintes perfis:

- **USER**
 - Participa da comunidade
 - Cria, edita e exclui seu próprio conteúdo
- **MODERATOR**
 - Pode moderar conteúdo (posts e comentários)
- **ADMIN**
 - Possui acesso total à aplicação

O perfil do usuário deve ser obtido a partir dos claims do token JWT.

3. Criação e Gerenciamento de Postagens

3.1 Criação de Postagens

Usuários autenticados podem criar postagens contendo:

- Título
- Conteúdo
- Autor (vinculado ao sub do Auth0)
- Data de criação

Endpoint esperado:

POST /posts

3.2 Edição e Exclusão de Postagens

- Usuários podem editar e excluir apenas suas próprias postagens
- Moderadores e administradores podem atuar conforme permissões

Endpoints esperados:

PUT /posts/{id}

DELETE /posts/{id}

3.3 Listagem de Postagens

- Retornar lista de postagens com informações básicas
- Permitir paginação (ver seção 7)

GET /posts

4. Comentários em Postagens

4.1 Criação de Comentários

- Usuários autenticados podem comentar em postagens
- Cada comentário deve:
 - Pertencer a um post
 - Ter um autor
 - Possuir data de criação

POST /posts/{id}/comments

4.2 Moderação de Comentários

- Moderadores e administradores podem:
 - Ocultar comentários
 - Excluir comentários inadequados

5. Curtidas (Likes)

5.1 Curtidas em Postagens e Comentários

- Usuários autenticados podem:
 - Curtir postagens
 - Curtir comentários
- Um usuário só pode curtir uma única vez cada post ou comentário

POST /posts/{id}/like

POST /comments/{id}/like

5.2 Contagem de Curtidas

- Cada post e comentário deve retornar:
 - Quantidade total de curtidas

6. Categorias e Tags

6.1 Categorias

- Cada postagem deve possuir uma categoria principal
- Apenas ADMIN pode:
 - Criar
 - Editar

- Remover categorias

6.2 Tags

- Postagens podem possuir múltiplas tags
- Tags são criadas dinamicamente conforme uso

7. Busca, Filtros e Ordenação

A API deve permitir:

7.1 Busca

- Busca textual por:
 - Título
 - Conteúdo

GET /posts/search?q=texto

7.2 Filtros

Filtros combináveis por:

- Categoria
- Tag
- Autor

7.3 Ordenação

- Ordenar postagens por:
 - Data de criação
 - Popularidade (número de curtidas)

8. Paginação e Performance

A API deve implementar paginação baseada em limite e offset para:

- Listagem de postagens
- Listagem de comentários

Parâmetros esperados:

?limit=10&offset=0

9. Controle Fino de Acesso

A API deve respeitar rigorosamente as seguintes regras:

USER

- Criar, editar e excluir apenas seu próprio conteúdo
- Curtir posts e comentários
- Comentar postagens

MODERATOR

- Moderar postagens e comentários
- Não pode excluir conteúdo de administradores

ADMIN

- Gerenciar categorias
- Acessar e gerenciar qualquer recurso
- Acesso irrestrito à API

*"A melhor maneira de prever o futuro é inventá-lo."
Alan Kay*