

# Generation

## **Boas práticas de construção de uma API rest.**

- 0 configure o applicationProperties para o seu banco de dados.
- 1 Crie uma model coloque todos os atributos necessários + get setters+ anotações
- 2 Teste sua aplicação (verifique se a tabela foi criada no mysql workbench)
- 3 Crie um repository e um controller com um método findAll
- 3 insira dados via o mysql workbench
- 4 teste o método findAll através do postman.
- 5 Crie um método findById no controller
- 6 Teste o método findById via postman.
- 7 crie um método post no controller
- 8 teste o método post via postman.
- 9 crie um método put no controller.
- 10 teste o método put no postman.
- 11 crie um método delete no controller
- 12 teste o método delete no postman.

## **Atividade CRUD Back-end blogPessoal.**

Construa um back-end para uma rede social onde ele tenha a capacidade de manipular os dados desta postagem dos usuários.

### **Parte 1 criação da tabela Tema.**

1. Camada de model com o nome de **Tema** com os atributos apresentados na video-aula:
2. Uma camada de repository com o nome **TemaRepository** (com a capacidade de se comunicar com o banco de dados mysql).
3. Uma camada de Controller com o nome de **TemaController** Com 5 endpoints:
  - findAllTema = um endPoint com a capacidade de trazer todas os temas.
  - findByIdTema = um endPoint com a função de trazer uma unico tema por id.
  - findByDescricaoPostagem = um endPoint com a função de trazer um unico tema por Descricao.
  - postTema = um endPoint com a função de gravar um novo tema no banco de dados.
  - putTema = um endPoint com a função de atualizar dados de um tema.
  - deleteTema = um endPoint com a função de apagar um tema do banco de dados).

## **Parte 2 relacionamento com a tabela Postagem.**

4. Camada de model com o nome de **Postagem** com os atributos apresentados na video-aula:
5. Crie um relacionamento de um para muitos/muitos para um, para essas models (um tema para muitas postagem e muitas postagem para um tema)

**IMPORTANTE: Não esqueça de colocar as anotações corretamente para o relacionamento e inibir recursividade através da anotação @JsonIgnoreProperties.**

5. Uma camada de repository com o nome **PostagemRepository** (com a capacidade de se comunicar com o banco de dados mysql).
6. Uma camada de Controller com o nome de **PostagemController** Com 5 endpoints:

- findAllPostagem = um endPoint com a capacidade de trazer todas as turmas.
- findByIdPostagem = um endPoint com a função de trazer uma unica turma por id.
- findByDescricaoTitulo = um endPoint com a função de trazer uma unica postagem por Titulo.
- postPostagem = um endPoint com a função de gravar uma nova turma no banco de dados.
- putPostagem = um endPoint com a função de atualizar dados de uma turma.
- deletePostagem = um endPoint com a função de apagar uma turma do banco de dados).