**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 seters+ 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

4. Insira dados via o mysql workbench

5. Teste o método findAll através do postman.

6. Crie um método findById no controller

7. Teste o método findByID via postman.

8. Crie um método post no controller

9. Teste o método post via postman.

10. Crie um método put no controller.

11. Teste o método put no postman.

12. Crie um método delete no controller

13. Teste o método delete no postman.

14. Crie um método getByTitulo no repository

12. Crie um método getByTitulo no controller

13. Teste o método getByTitulo no postman.

**Atividade CRUD Back-end blog-pessoal.**

O back-end deverá possuir 3 camadas:

1. Camada de model com o nome de **Postagem** com os seus respectivos atributos descritos nos video.
2. Uma camada de repository com o nome **PostagemRepository** (com a capacidade de se comunicar com o banco de dados mysql).
3. Uma camada de Controller com o nome de **PostagemController** Com 6 endpoints:

* findAllPostagem = um endPoint com a capacidade de trazer todas as Postagem.
* findByIDPostagem = um endPoint com a função de trazer uma única Postagem por id.
* postPostagem = um endPoint com a função de gravar uma nova Postagem no banco de dados.
* putPostagem = um endPoint com a função de atualizar dados de uma Postagem.
* deletePostagem = um endPoint com a função de apagar uma Postagem do banco de dados.
* Crie o método getByTitulo no repository.
* Crie um método getByTitulo no controller = um endPoint com a função de trazer uma única Postagem por Título.