

## ATIVIDADE – Spring

Atividade prática – Desenvolvendo o Backend com Spring

### Instruções gerais:

1. Utilize o Spring Tool Suite (STS) para desenvolver os projetos Spring.
2. Crie um repositório no Github para cada projeto Spring
3. Ao concluir uma nova etapa do projeto, envie todas as alterações para o Repositório do projeto criado na sua conta pessoal do Github (a pasta inteira)
4. Envie o link do repositório no Github através da Plataforma da Generation na data indicada
5. Caso seja solicitado, adicione os links individuais dos arquivos .JAVA indicados, no item:  
**Adicione um dos links da sua entrega**, localizada depois do link do Repositório, na tela de entrega da atividade na plataforma, para validação da atividade.
6. **Validar:** *Postagem.java*, *PostagemRepository.java* e *PostagemController.java*

**Mantenha as entregas das Atividades em dia na Plataforma da Generation**

## EXERCÍCIOS

### Boas práticas:

- 1) Crie o Repositório do projeto no Github.
- 2) Crie o projeto no **Spring Initializr**, seguindo as boas práticas de projeto.
- 3) Configure o arquivo **application.properties** para criar e acessar o Banco de dados MySQL no seu computador.
- 4) Execute a sua aplicação no STS e verifique se o Banco de dados foi criado no **MySQL Workbench**.
- 5) Crie a Classe Model e coloque todos os atributos necessários, incluindo os Métodos **Get e Set** e as **Anotações**. Não é necessário criar um Método Construtor.
- 6) Execute a sua aplicação no STS e verifique se a tabela foi criada no **MySQL Workbench**.
- 7) Crie a Interface Repository.
- 8) Crie a Classe Controller.
- 9) Na Classe Controller, crie o Método **getAll()**.

10) Insira dados na tabela através do **MySQL Workbench**.

11) Teste o método `getAll()` através do **Insomnia**.

12) Ao final desta atividade, inicialize a pasta do projeto como um repositório Git e faça a conexão com o repositório remoto no Github.

13) Envie o projeto para o repositório remoto.

## **ATIVIDADE CRUD BACKEND BLOG-PESSOAL – PARTE 01**

O Projeto Blog Pessoal possui as seguintes características:

- 1) O Banco de dados da aplicação será o **db\_blogpessoal**.
- 2) Na Camada Model será criada a Classe **Postagem** com os seus respectivos atributos.
- 3) A Classe Postagem criará a Tabela **tb\_postagens** no Banco de dados **db\_blogpessoal**.
- 4) Na Camada Repository será criada a Interface **PostagemRepository** (com a capacidade de se comunicar com o banco de dados MySQL).
- 5) Na Camada de Controller será criada a Classe **PostagemController**.
- 6) Na Camada Controller será criado um endpoint:
  - **getAll()** com a capacidade de listar todas as Postagens.