

## 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.

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

## EXERCÍCIOS

### Boas práticas:

- 1) Insira todas as Dependências necessárias para a **Spring Security**
- 2) Na Camada Model, crie as duas Classes Model, necessárias para a Spring Security e coloque todos os atributos necessários, incluindo os Métodos **Get** e **Set** e as **Anotações**. Não é necessário criar um Método Construtor.
- 3) Na Camada Model, crie os Relacionamentos necessários para a Spring Security. Não esqueça de acrescentar todas as Anotações e Métodos Get e Set necessários nas 2 Classes.
- 4) Execute a sua aplicação no STS e verifique se a tabela foi criada no **MySQL Workbench**.
- 5) Na Camada Repository, crie a terceira Interface Repository.
- 6) Na nova Interface Repository, crie o Método de Busca Específica necessário para a Spring Security
- 7) Crie a Camada Security com as Classes necessárias para a Spring Security.
- 8) Crie a Camada Service com as Classe de Serviço necessária para a Spring Security.
- 9) Na Camada Controller, crie a segunda Classe Controller.
- 10) Na Classe Controller, crie todos os Métodos do **CRUD**.

11) Teste todos os métodos do CRUD através do **Insomnia**.

12) Ao concluir a atividade, envie as atualizações do projeto para o repositório remoto.

## ATIVIDADE SECURITY BACKEND LOJA DE GAMES

Nesta etapa, vamos implementar a segurança no Projeto Loja de Games:

- 1) Insira as Dependências: **Spring Security** e **Commons Codec**.
- 2) Na Camada Model será criada as Classes **Usuario** e **UsuarioLogin** com os seus respectivos atributos.
- 3) A Classe **UsuarioLogin** não irá gerar tabela, logo não precisa de Anotações.
- 4) A Classe **Usuario** criará a Tabela **tb\_usuarios** no Banco de dados da aplicação.
- 5) Na Camada Repository será criada a Interface **UsuarioRepository** (com a capacidade de se comunicar com o banco de dados MySQL).
- 6) Na Interface **UsuarioRepository** vamos adicionar o Método de busca específica:
  - **findByUsuario()** com a função de trazer um usuário específico pelo e-mail (conteúdo do atributo usuário, da Model **Usuario**).
- 7) Na Camada Security serão criadas as Classes:
  - **BasicSecurityConfig**
  - **UserDetailsImpl**
  - **UserDetailsServiceImpl**
- 8) Na Camada Service será criada a Classe **UsuarioService**
- 9) Implemente na Classe **UsuarioService** os Métodos:
  - **cadastrarUsuario()**
  - **atualizarUsuario()**
  - **autenticarUsuario()**
- 10) Na Camada Controller será criada a Classe **UsuarioController**.
- 11) Na Classe **UsuarioController** serão criados os seguintes métodos:
  - **getAll()** com a capacidade de listar todos os Usuários.
  - **getById()** com a função de trazer um único Usuário identificada pelo id.
  - **post()** com a função de gravar (persistir) um novo Usuário no banco de dados.
  - **put()** com a função de atualizar os dados de um Usuário.
  - **login()** com a função de efetuar login na API.