



**Certified Tech
Developer**

The Ultimate Degree

Back end I

Exercício para as Mesas de Trabalho: padrão MVC (View Template)

Objetivo

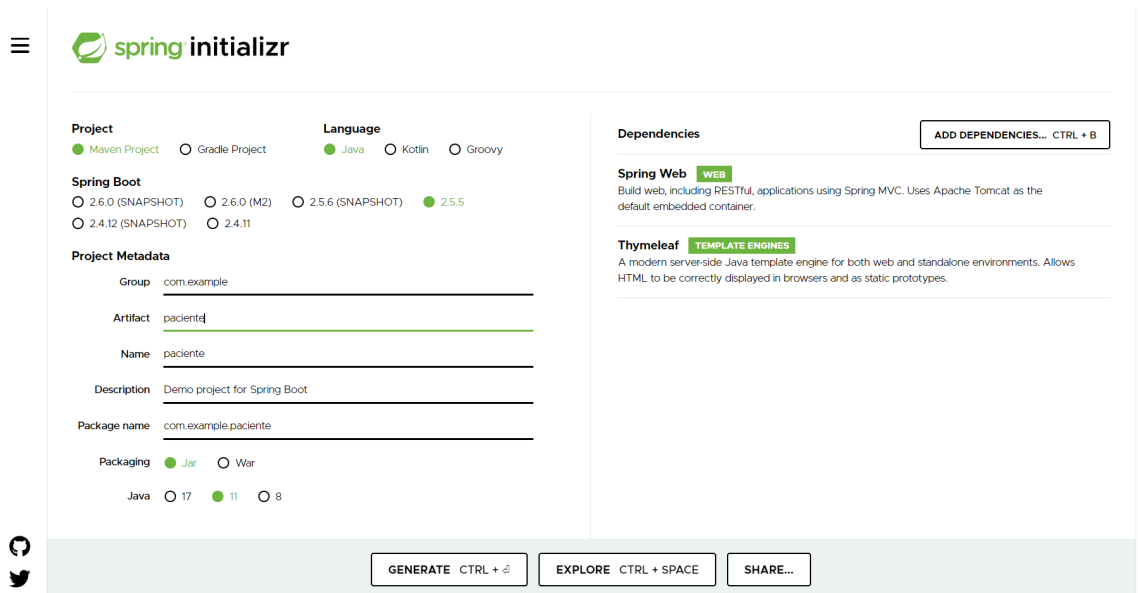
Assim como vimos na aula virtual, agora vamos criar outro projeto com Spring Boot com o nome Paciente, considere:

- Exercício individual
- Nível de complexidade: baixo 🔥

Instruções

1. Primeiro, devemos acessar o seguinte link: <https://start.spring.io>.

Lembre-se de atribuir um nome ao projeto e adicionar as dependências: Spring Web e Thymeleaf.

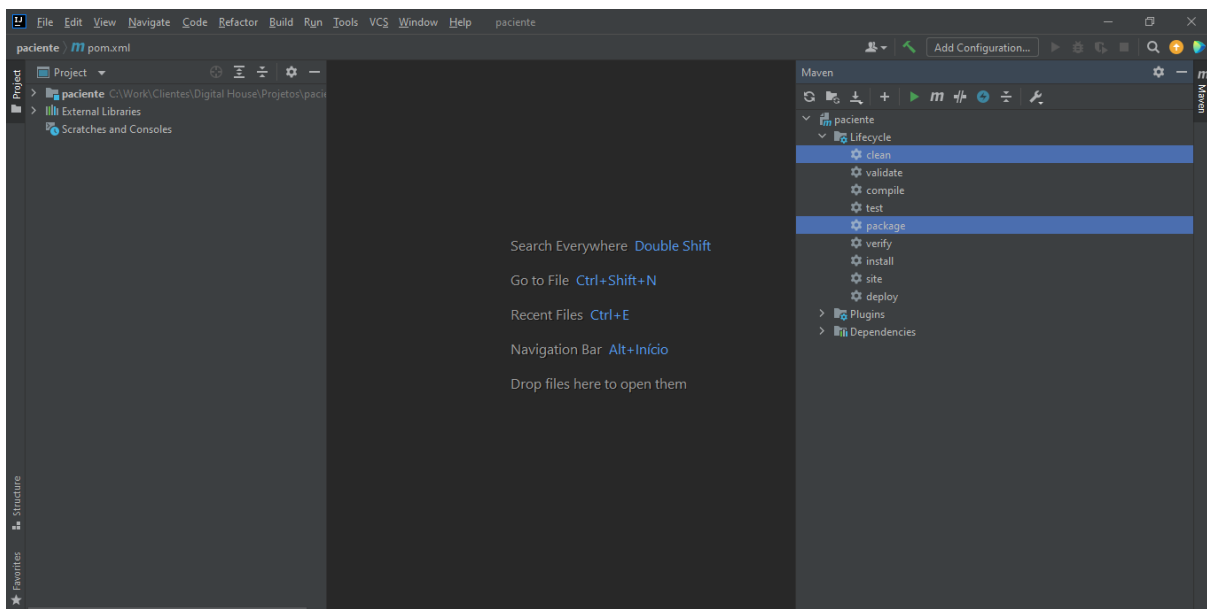


The image shows the Spring Initializr web interface. It is a form for generating a Spring project. The 'Project' section has 'Maven Project' selected. The 'Language' section has 'Java' selected. The 'Spring Boot' section has '2.5.5' selected. The 'Project Metadata' section has 'Group' as 'com.example', 'Artifact' as 'paciente', 'Name' as 'paciente', 'Description' as 'Demo project for Spring Boot', and 'Package name' as 'com.example.paciente'. The 'Packaging' section has 'Jar' selected. The 'Dependencies' section has 'Spring Web' and 'Thymeleaf' selected. At the bottom, there are buttons for 'GENERATE', 'EXPLORE', and 'SHARE...'. There are also social media icons for GitHub and Twitter on the left.

2. Gerar o projeto e descompactar o arquivo .zip para abri-lo no IntelliJ IDEA.

No IntelliJ, acesse o menu “File” > “New” > “Project from Existing Sources”;

3. Na aba Maven, selecione os itens “clean” e “package”, e pressione o “play” na aba superior do Maven.



4. Crie um controller no package Controller.

```

@Controller
public class PacienteController {

    @GetMapping("/index")
    public String welcome(Model model) {
        model.addAttribute("nome", "Felipe");
        return "index";
    }
}

```

Como podemos ver, o método welcome tem a Model como parâmetro. É desta forma que devemos enviar os dados do Controller para a View.

Na última linha, retornamos "index", mencionando a um template HTML.

5. Crie um arquivo index.html com o conteúdo abaixo, dentro da estrutura "resources/templates".

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <title>Users</title>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstr
ap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkF
OJwJ8ERdknLPMO" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.4.1/css/all.css"
integrity="sha384-5sAR7xN1Nv6T6+dT2mhtzEpVJvfS3NScPQTrOxhwjIuvcA67KV
2R5Jz6kr4abQsz" crossorigin="anonymous">

```

```
<link rel="stylesheet" href="../css/shards.min.css">
</head>
<body>
<td th:text="${nome}"></td>

</body>
</html>
```

6. Execute o método main da classe PacienteApplication. Em seguida acesse no navegador <http://localhost:8080/paciente>. Então você deverá ver na View o nome que definimos na Model.