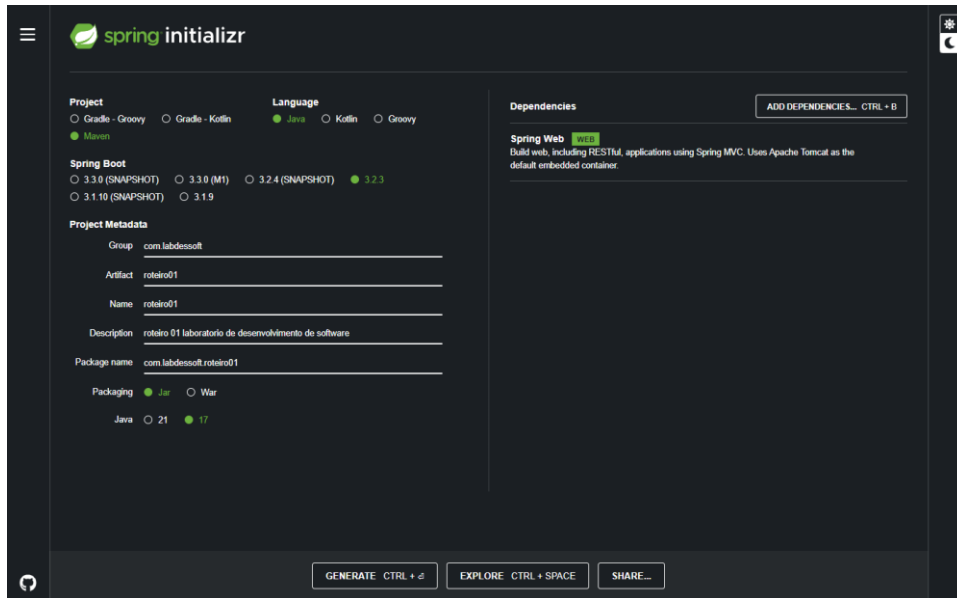


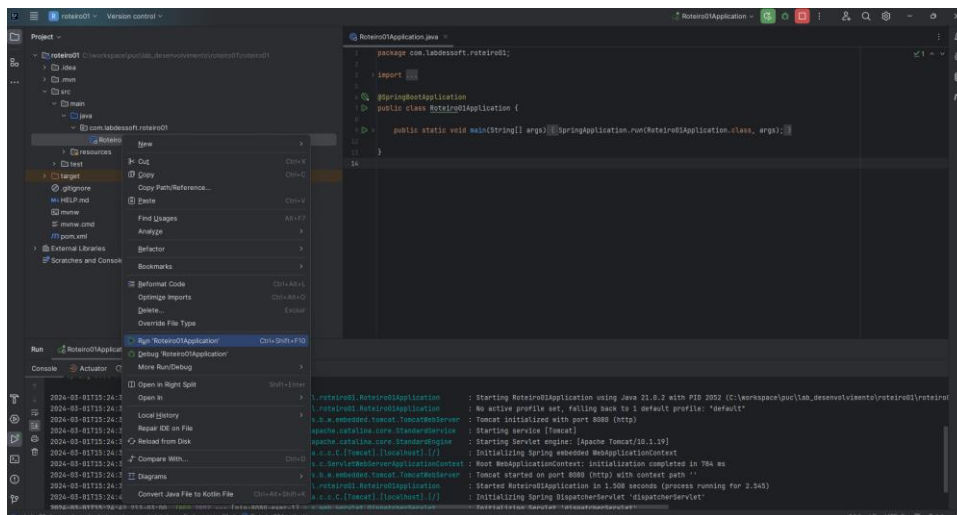
Criando nosso primeiro projeto

1. Acesse o site: <https://start.spring.io>
2. Preencher conforme figura a seguir:

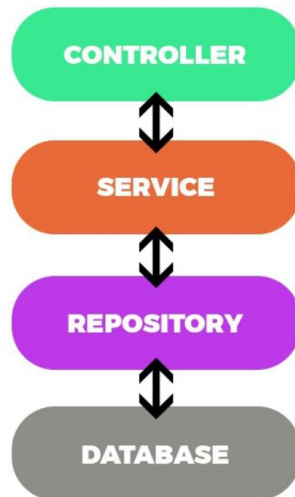


The image shows the Spring Initializr web form. The 'Project' section has 'Maven' selected. The 'Language' section has 'Java' selected. The 'Spring Boot' section has '3.2.3' selected. The 'Project Metadata' section has 'Group' as 'com.labdessoft', 'Artifact' as 'roteiro01', 'Name' as 'roteiro01', 'Description' as 'roteiro 01 laboratorio de desenvolvimento de software', 'Package name' as 'com.labdessoft.roteiro01', and 'Packaging' as 'Jar'. The 'Dependencies' section has 'Spring Web' selected. At the bottom, there are buttons for 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

3. Pressionar na opção Generate e utilizar o arquivo gerado.
4. Agora basta descompactar o arquivo e abri-la no seu editor predileto.
5. Testando o projeto

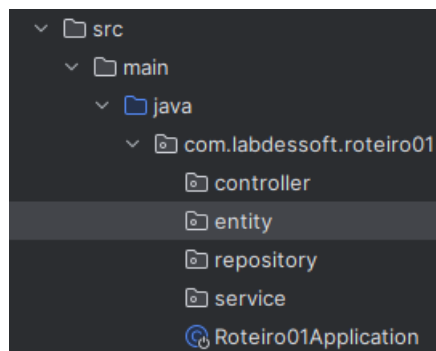


6. Arquitetura do nosso projeto será Controller-Service-Repository
 - É um padrão muito adotado no Spring Boot.
 - Faz uma importante separação de interesses.

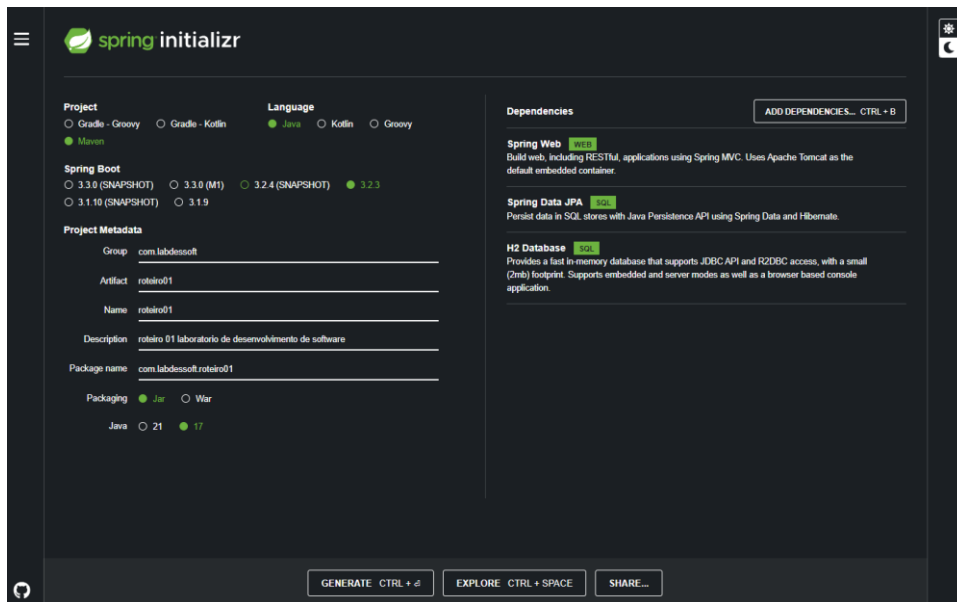


- a. **Controller:** é a única camada responsável por expor a funcionalidade para que ela possa ser consumida por entidades externas.
- b. **Service:** é responsável pela lógica do negócio.
- c. **Repository:** é responsável por armazenar e recuperar qualquer conjunto de dados.
- d. Leitura adicional:
 - i. <https://tom-collings.medium.com/controller-service-repository-16e29a4684e5>
 - ii. <https://wiki-coatic.saude.gov.br/backend-java>

7. Vamos criar a estrutura de organização de código, conforme a imagem abaixo.



8. E se eu não tiver colocado todas as dependências? Basta entrar novamente no <https://start.spring.io/>, adicionar as dependências e utilizar o botão **Explore**.



9. Agora é hora de criar nosso Controller.

```
package com.labdessoft.roteiro01.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class TaskController {
    @GetMapping("/task")
    public String listAll() {
        return "Listar todas as tasks";
    }
}
```

10. Vamos testar?

- a. <http://localhost:8080/>
- b. <http://localhost:8080/api/>
- c. <http://localhost:8080/api/task>

11. Agora é com você:

- a. Faça os endpoints que julgar necessário
- b. Desenvolva a classe de Entidade da forma que julgar necessária.