

Roteiro para aula prática 2 – Spring boot

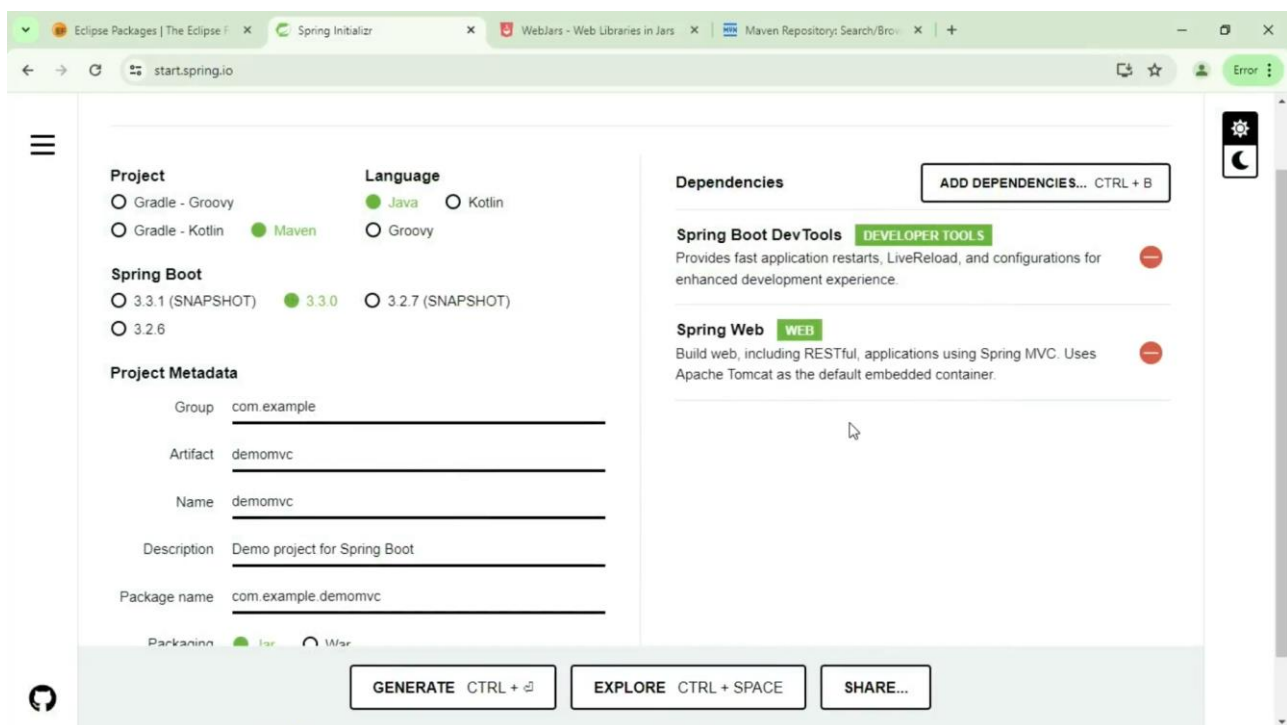
Tempo estimado para esta prática: 15 min

IDE utilizada: Eclipse JavaEE

Nessa aula iremos criar um projeto para gerenciar tarefas. Nas aulas posteriores continuaremos nesse mesmo projeto. Os arquivos necessário para essa aula estão disponíveis na rota de aprendizagem.

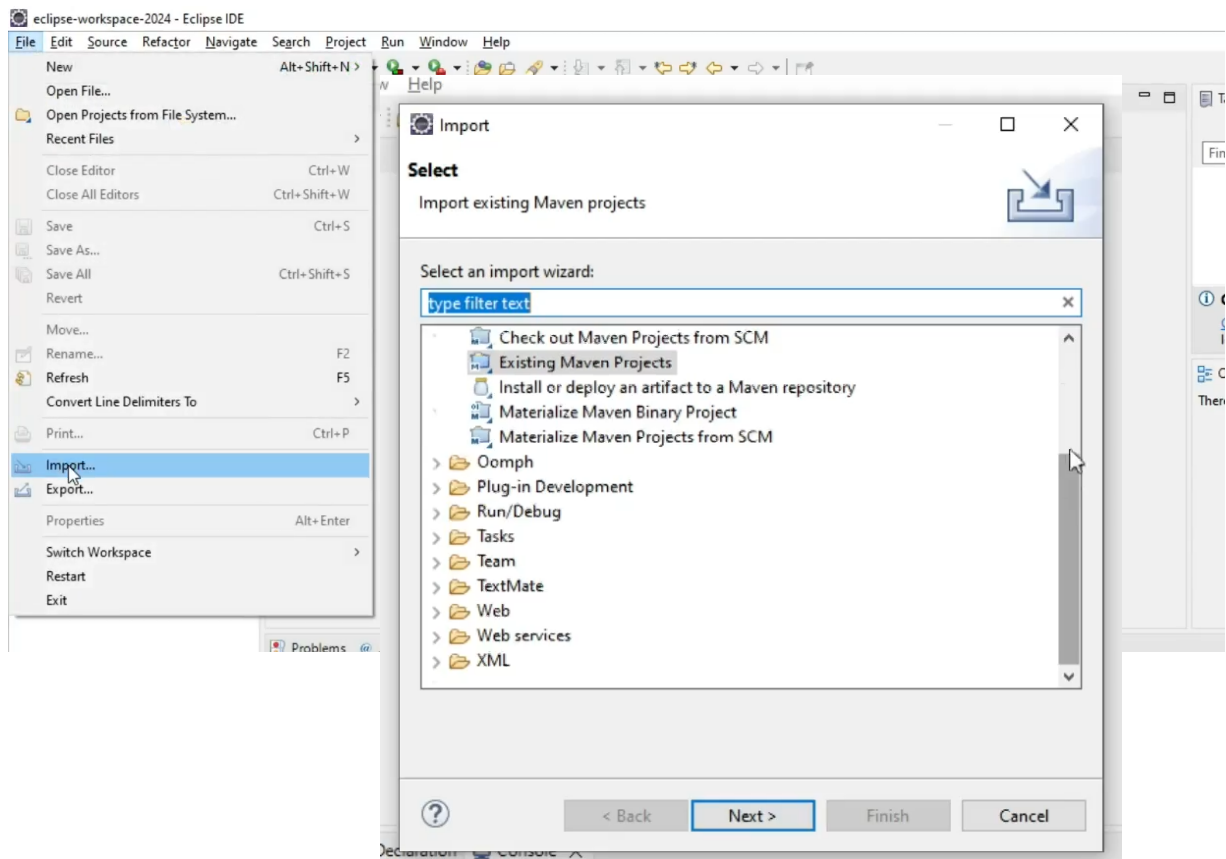
1. Vamos criar o projeto no inicializr, acessível no endereço: <https://start.spring.io/>

O projeto deverá ser criado de acordo com as configurações a seguir:

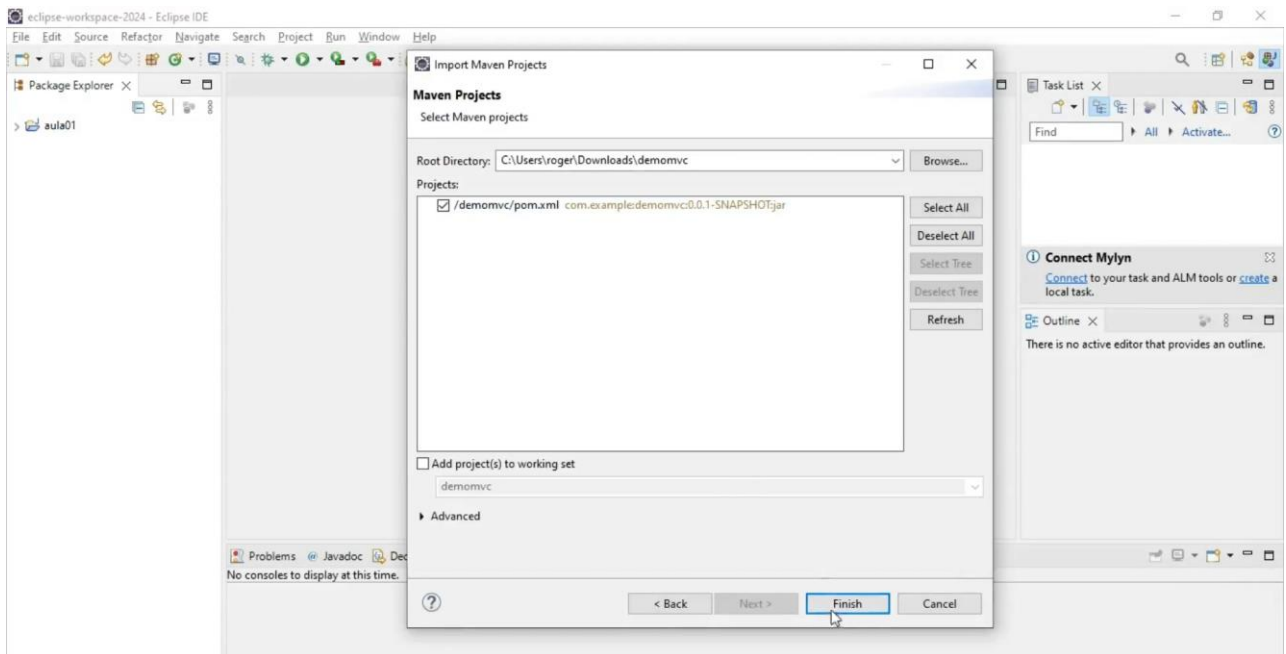


Para gerar o projeto clique em GENERATE - irá gerar e baixar o seu projeto compactado.

2. Descompactar o arquivo e abrir no eclipse (File > import>Maven>Existing Maven Projects).



Marque a caixa de seleção em Projects e clique em Finish.

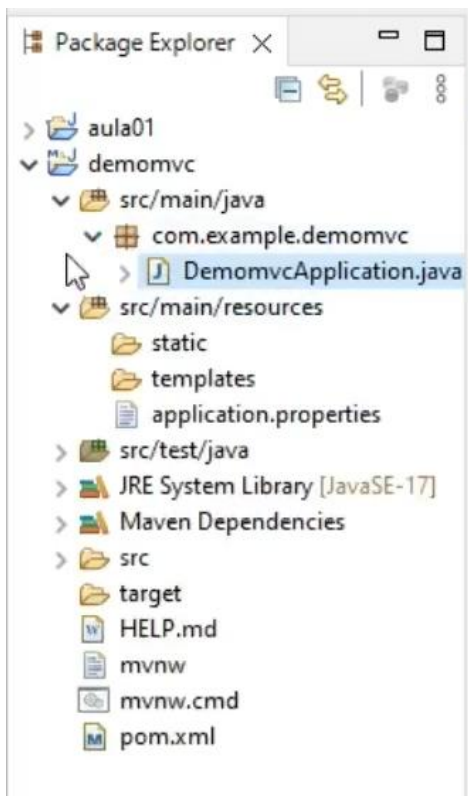


3. Observe a estrutura de pastas do seu projeto. A pasta raiz de seu projeto é a demomvc.

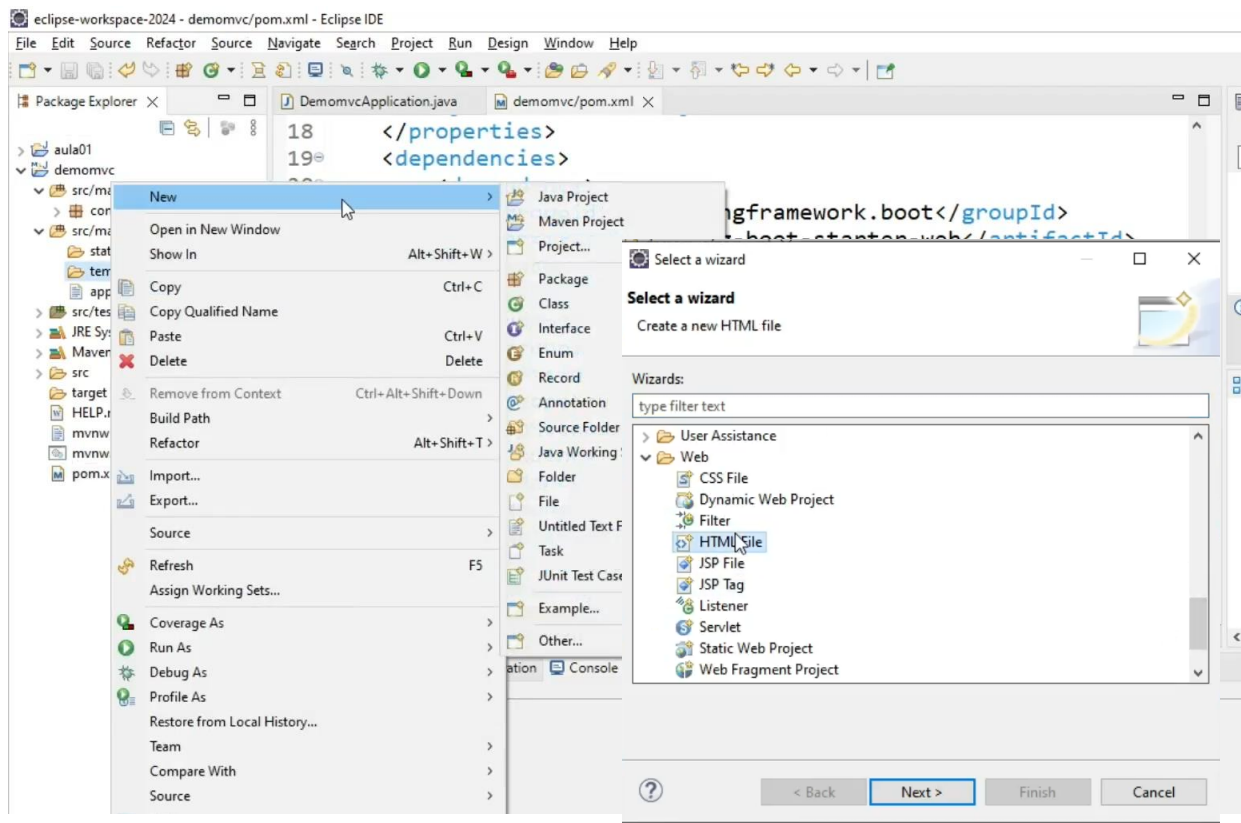
Visível em com.example.demomvc

O arquivo que executa a aplicação é o DemomvcApplication.java

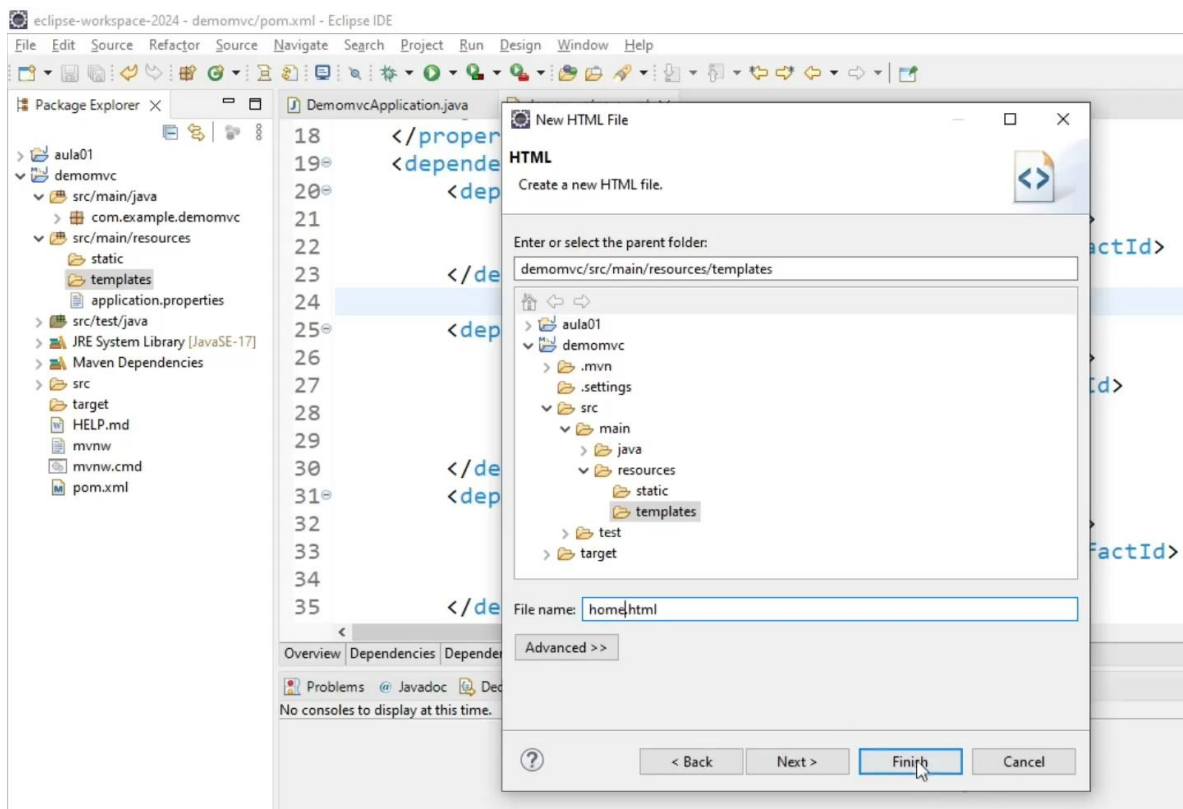
As páginas htmls serão colocadas na pasta templates.



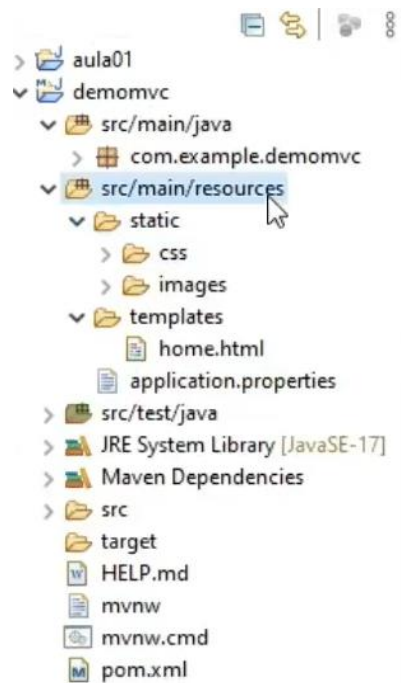
4. Para criar uma página html: selecionar a pasta templates, clicar com botão direito do mouse e escolher New > Other > Web > HTML File



Clicar em Next, abrirá a seguinte janela, dê um nome home ao arquivo html e clique em Finish.



5. Copiar para o projeto as pastas css e images (copiar para dentro da pasta static). Copiar e o conteúdo do arquivo home.html para sua página home.html.



6. Agora vamos colocar as dependências dos webjars necessárias para o desenvolvimento do projeto.

O endereço para os sites de onde estão vindo esses webjars e dependências estão colocados abaixo:

<https://www.webjars.org/>

<https://mvnrepository.com/>

A seguir estão as dependências que devem ser colocadas no pom.xml. Essas dependências estão disponíveis também na rota de aprendizagem. Basta copiá-las e colá-las no arquivo pom.xml do seu projeto.

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>webjars-locator-core</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>bootstrap</artifactId>
  <version>4.5.2</version>
</dependency>
```

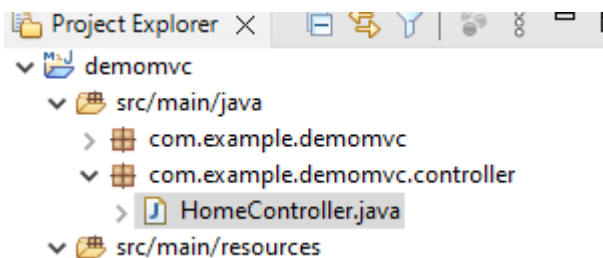
```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>jquery</artifactId>
  <version>3.5.1</version>
```

</dependency>

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>jquery-maskedinput</artifactId>
  <version>1.4.0</version>
</dependency>
```

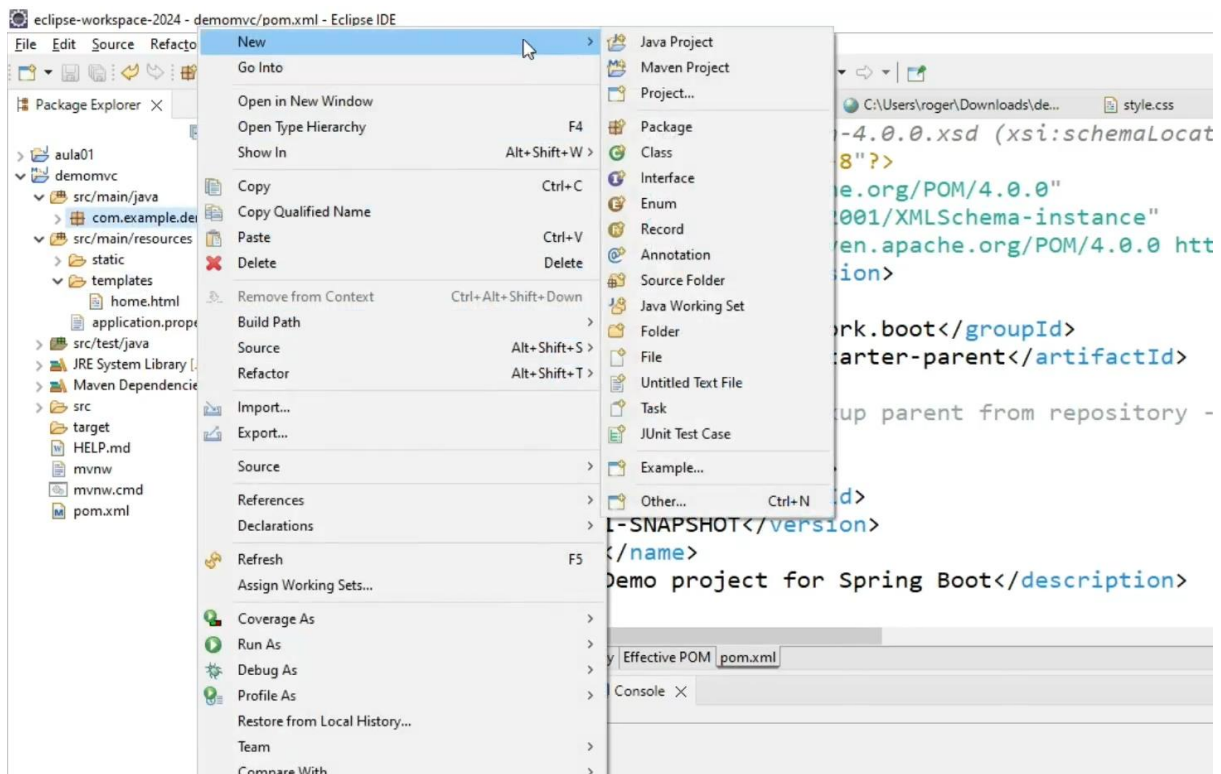
7. Criar um Controller simples para testar a página html.

Vamos agora criar uma classe HomeController.java dentro da pasta controller

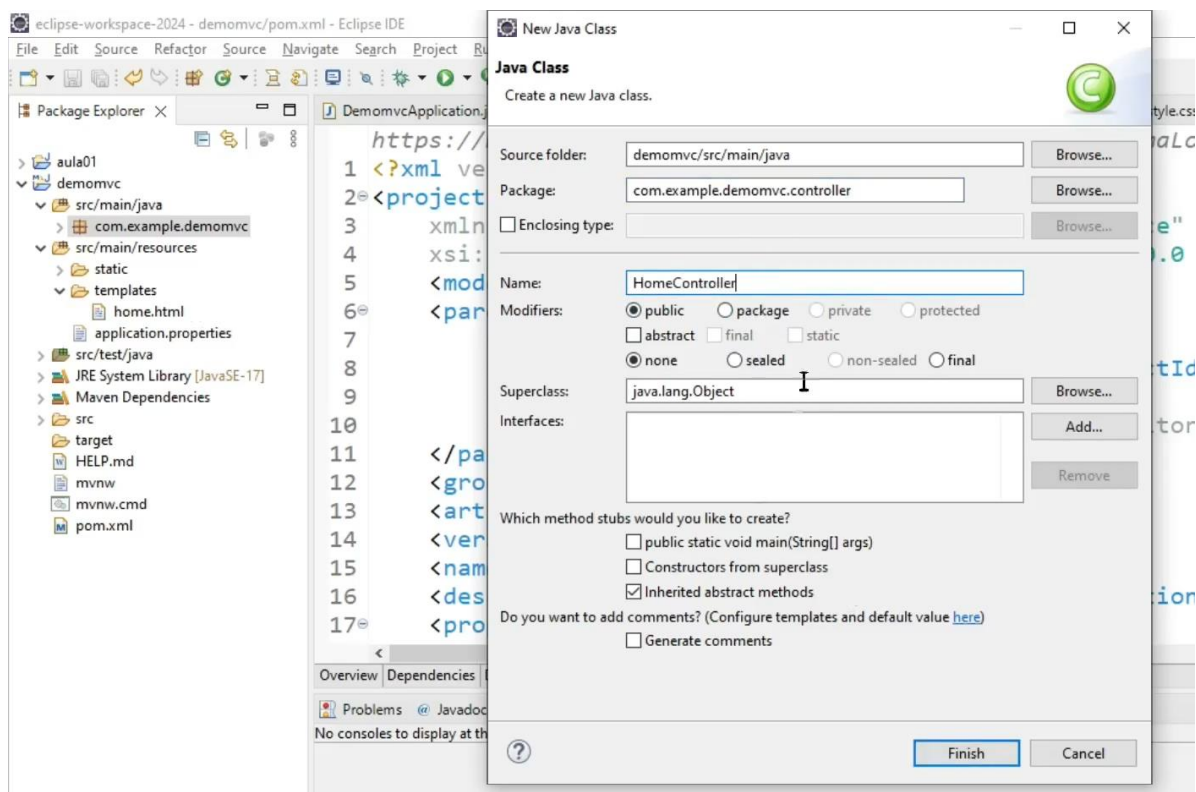


Relembrando como cria classe e pasta.

Selecione a pasta raiz, botão direito do mouse > New > Class



Daremos um nome a nossa classe e já criaremos também a pasta com nome de controller.

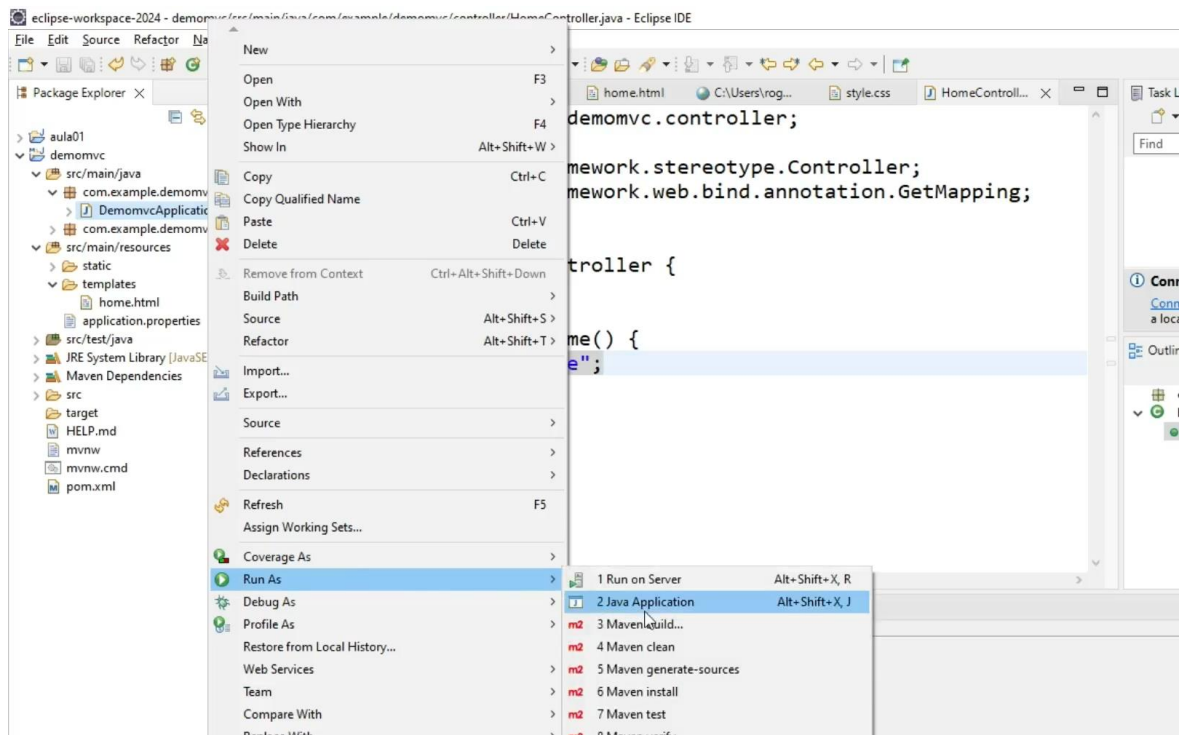


Código do Controller

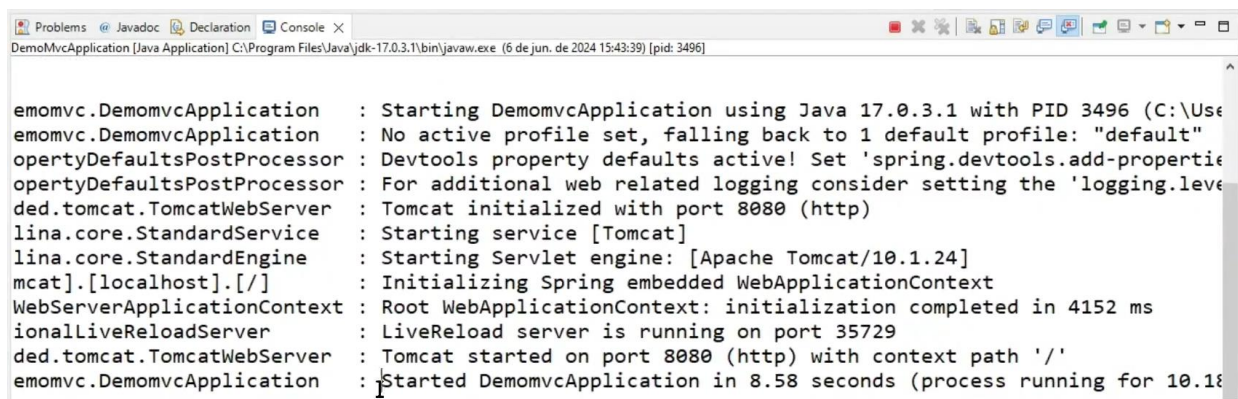
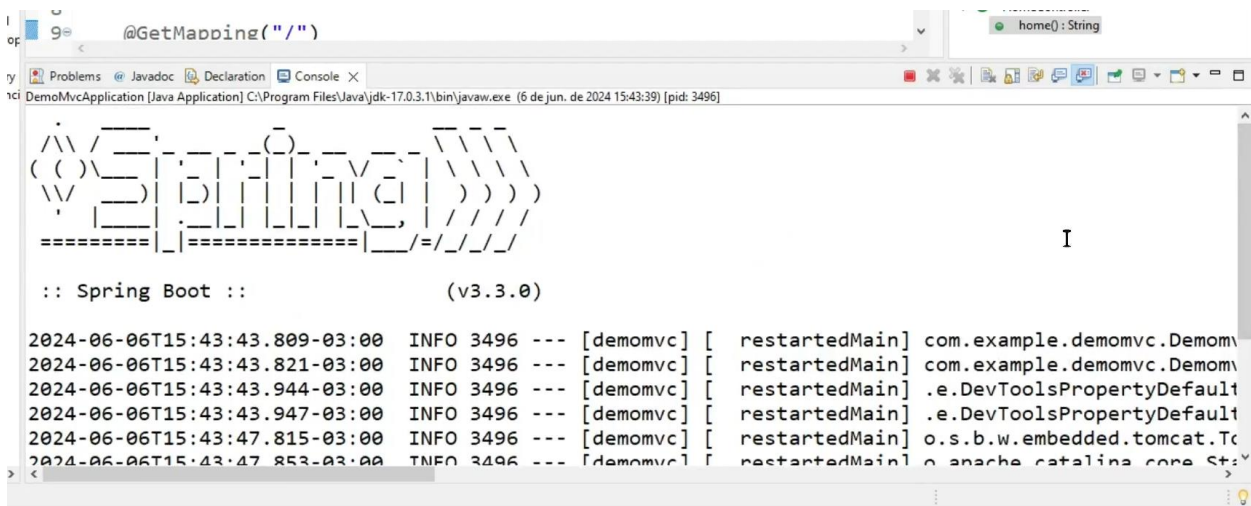
```
package com.example.demomvc.controller;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
@Controller
public class HomeController {
    @GetMapping("/")
    public String home() {
        return "/home";
    }
}
```

8. Rodar o projeto para testar

Selecionar a classe DemomvcApplication > Botão direito do mouse > Run as>JavaApplication

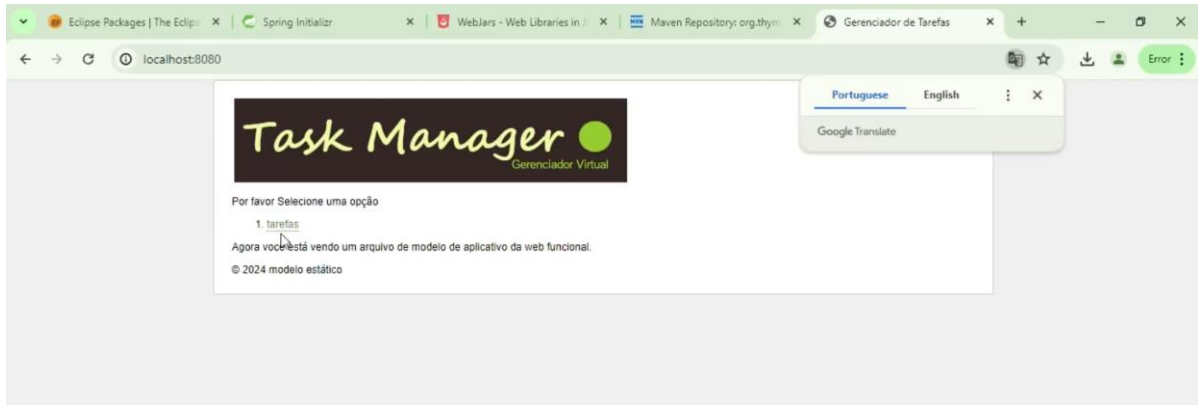


Se tudo der certo, na saída do console vai aparecer algo semelhante a imagem a seguir:



9. Agora vamos testar a aplicação no navegador

No seu navegador digite <http://localhost:8080>, a página deve carregar semelhante ao da imagem abaixo.



Finalizamos a primeira parte do nosso projeto.