

Guia De Desenvolvimento
Spring MVC
Thymeleaf
Spring Boot
Git
2023
Parte 1

Introdução

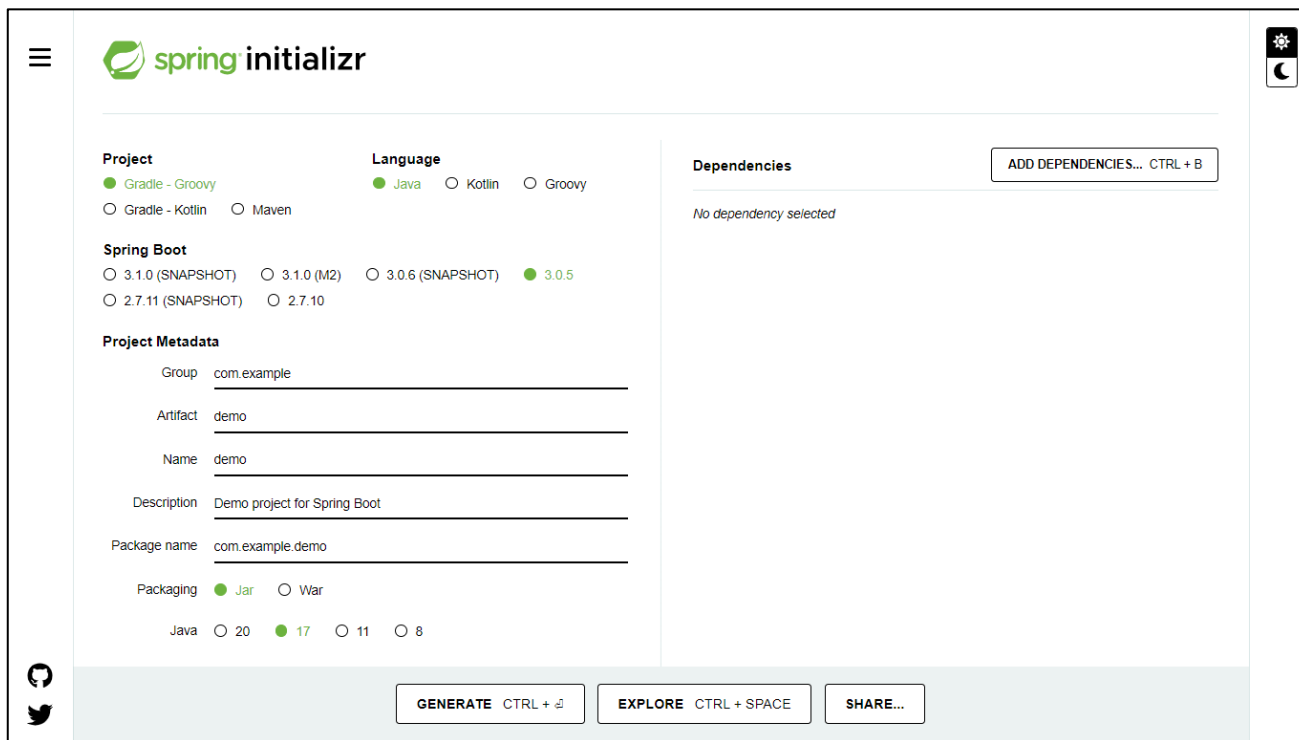
Neste documento teremos o passo-a-passo para a construção de uma aplicação web utilizando as ferramentas Spring MVC, Thymeleaf, Spring Boot e Git.

Este guia se destina a dar suporte as aulas de LIP1 do segundo módulo do curso de informática noturno do ITB Belval, portanto, algumas explicações mais detalhadas serão omitidas pois serão apresentadas pelo professor durante a aula.

Parte 1

Criar novo projeto

1 – Acesse a página <https://start.spring.io/>




The screenshot shows the Spring Initializr web application interface. The header includes the Spring logo and the text "spring initializr". The main content area is divided into three sections: "Project", "Language", and "Dependencies".

- Project:** Includes radio buttons for "Gradle - Groovy" (selected), "Gradle - Kotlin", and "Maven".
- Language:** Includes radio buttons for "Java" (selected), "Kotlin", and "Groovy".
- Spring Boot:** Includes radio buttons for "3.1.0 (SNAPSHOT)", "3.1.0 (M2)", "3.0.6 (SNAPSHOT)", "3.0.5" (selected), "2.7.11 (SNAPSHOT)", and "2.7.10".
- Project Metadata:** Includes text input fields for "Group" (com.example), "Artifact" (demo), "Name" (demo), "Description" (Demo project for Spring Boot), and "Package name" (com.example.demo). It also includes radio buttons for "Packaging" (Jar selected, War) and "Java" version (20, 17 selected, 11, 8).
- Dependencies:** Includes a button "ADD DEPENDENCIES... CTRL + B" and the text "No dependency selected".

At the bottom, there are three buttons: "GENERATE CTRL + G", "EXPLORE CTRL + SPACE", and "SHARE...".

2 – No campo “Project” selecione a opção “Maven”, no campo “Group” preencha com “br.com.belval”, em “Artifact” preencha com “crud”, em “Description” preencha com “Exemplo de CRUD desenvolvido com Spring MVC, Thymeleaf e Spring Boot”. Marque a opção “11” para a versão do java.



Project

☐ Gradle - Groovy
 ☒ Gradle - Kotlin
 ☒ Maven

Language

☒ Java
 ☐ Kotlin
 ☐ Groovy

Spring Boot

☐ 3.1.0 (SNAPSHOT)
 ☐ 3.1.0 (M2)
 ☐ 3.0.6 (SNAPSHOT)
 ☒ 3.0.5
 ☐ 2.7.11 (SNAPSHOT)
 ☐ 2.7.10

Project Metadata

Group

Artifact

Name


Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 20 ☐ 17 ☒ 11 ☐ 8

3 – Clique no botão “ADD DEPENDENCIES... CTRL + B”



Project

☐ Gradle - Groovy
 ☒ Gradle - Kotlin
 ☒ Maven

Language

☒ Java
 ☐ Kotlin
 ☐ Groovy

Spring Boot

☐ 3.1.0 (SNAPSHOT)
 ☐ 3.1.0 (M2)
 ☐ 3.0.6 (SNAPSHOT)
 ☒ 3.0.5
 ☐ 2.7.11 (SNAPSHOT)
 ☐ 2.7.10

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 20 ☐ 17 ☒ 11 ☐ 8

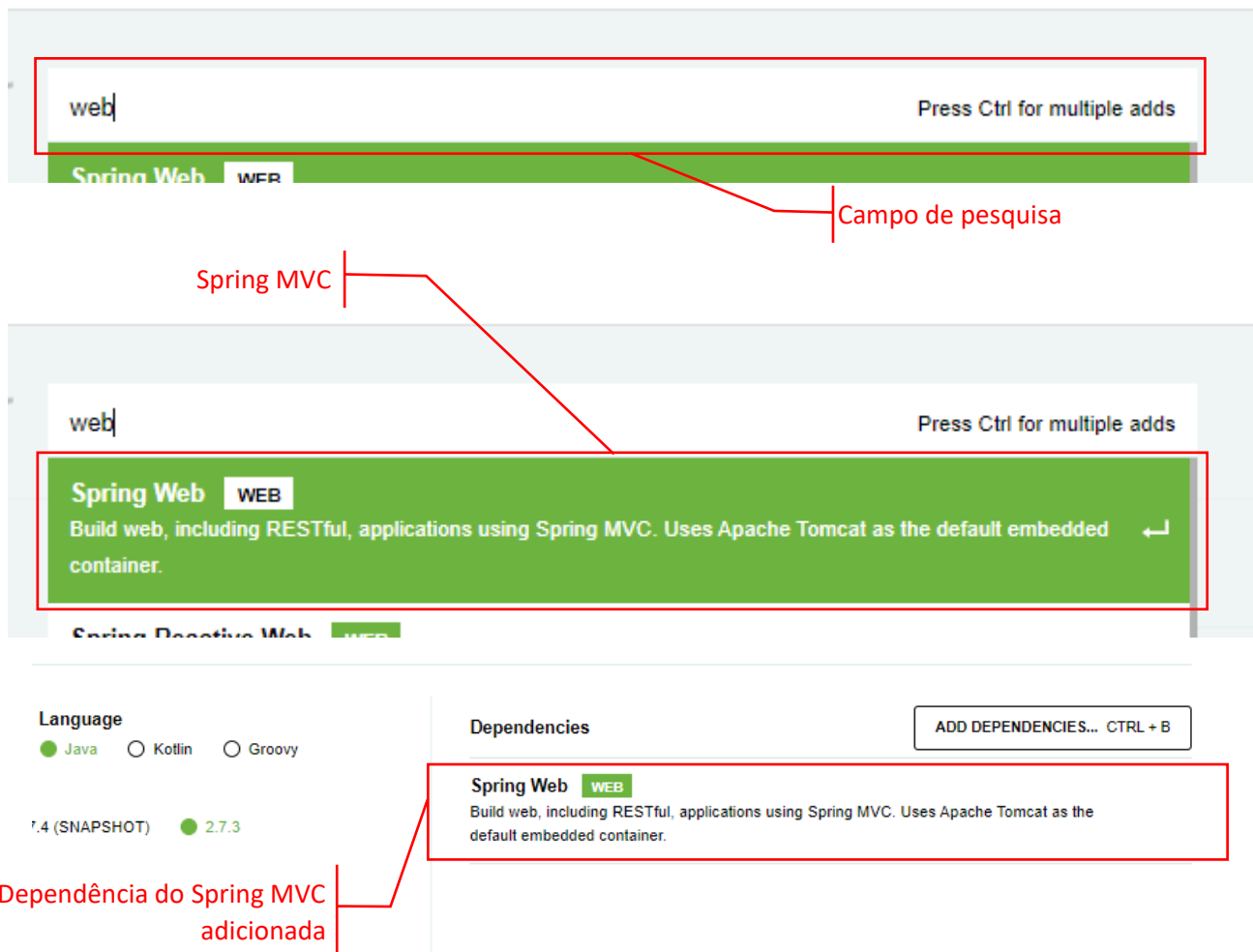
Dependencies

No dependency selected

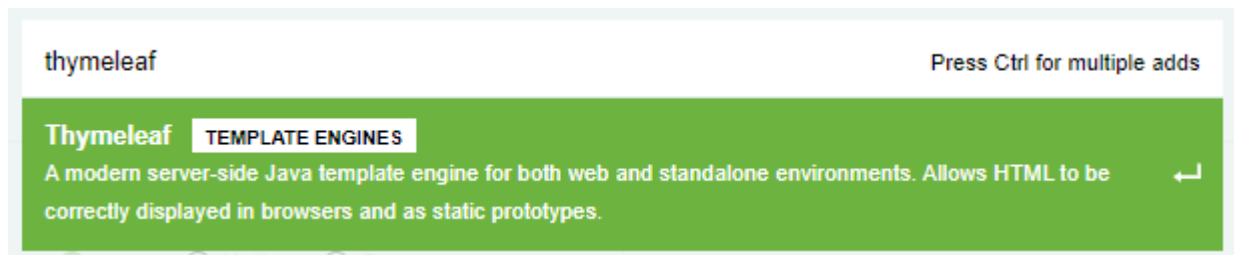
ADD DEPENDENCIES... CTRL + B

GENERATE CTRL + G **EXPLORE CTRL + SPACE** **SHARE...**

4 – Na próxima tela, preencha o campo de pesquisa com o termo “web”. Provavelmente surgirá uma opção “Spring Web” na primeira opção da lista. Clique nesse item para adicionar o framework Spring MVC ao projeto.



5 – Em seguida, clique novamente em “ADD DEPENDENCIES...CTRL + B” e preencha o campo de pesquisa com o termo “thymeleaf”. Provavelmente surgirá uma opção “Thymeleaf” como única opção da lista. Clique nesse item para adicionar o motor de “templates”(modelos).





Project

☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ Maven

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.1.0 (SNAPSHOT) ☐ 3.1.0 (M2) ☐ 3.0.6 (SNAPSHOT) ☒ 3.0.5 ☐ 2.7.11 (SNAPSHOT) ☐ 2.7.10

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 20 ☐ 17 ☒ 11 ☐ 8

Dependencies

Spring Web WEB Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container. ☐

Thymeleaf TEMPLATE ENGINES A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes. ☐

ADD DEPENDENCIES... CTRL + B



GENERATE CTRL + ⌘

EXPLORE CTRL + SPACE

SHARE...

6 – Clique em “Generate CTRL + <ENTER>” para baixar o projeto.

Package name

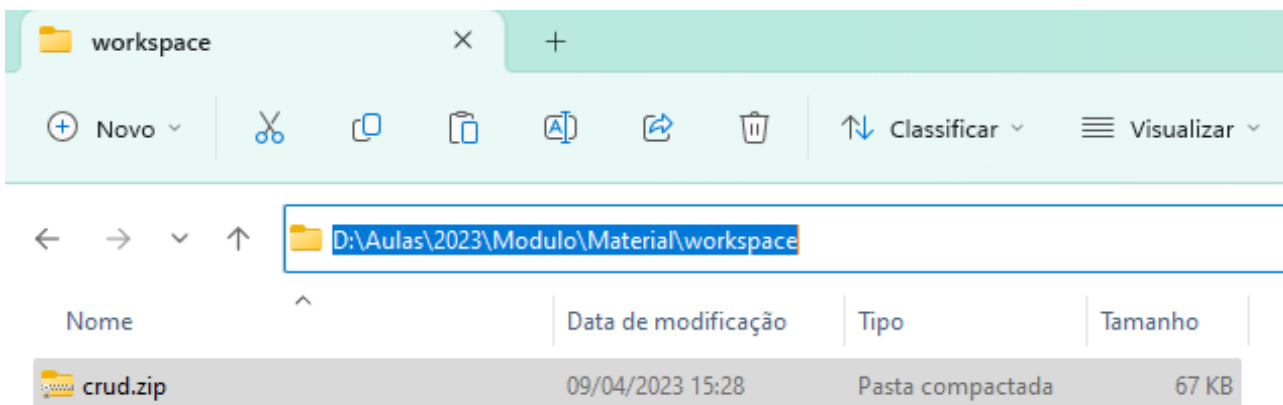
Packaging ☒ Jar ☐ War

Java ☐ 20 ☐ 17 ☒ 11 ☐ 8

GENERATE CTRL + ⌘

crud.zip

7 – Copie o *.zip baixado para sua pasta “.../workspace”



8 – Dentro do arquivo *.zip há uma pasta “crud”, extraia esta pasta para dentro do diretório workspace.

D:\Aulas\2023\Modulo\Material\workspace				
Nome		Data de modificação	Tipo	Tamanho
crud		09/04/2023 15:33	Pasta de arquivos	
crud.zip		09/04/2023 15:28	Pasta compactada	67 KB

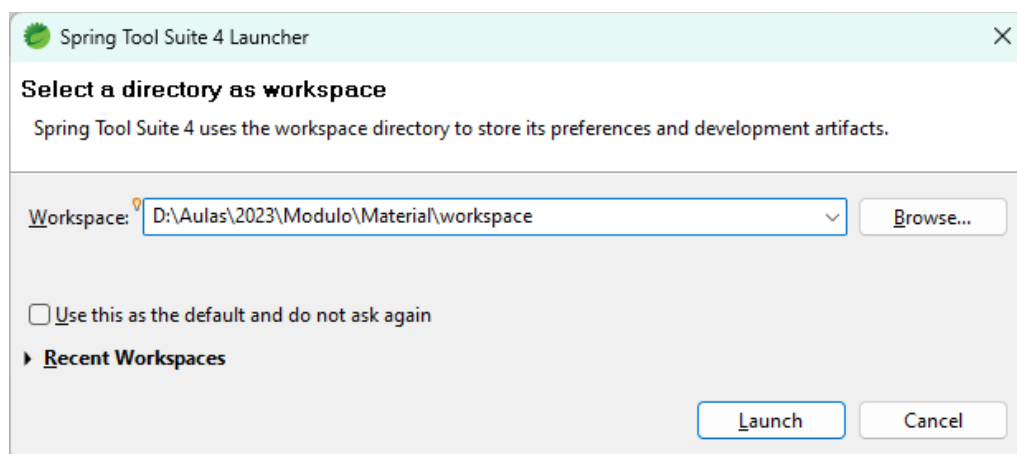
9 – Como efeito, teremos uma pasta e um arquivo *.zip “crud” dentro de workspace. Apague o arquivo *.zip ficando apenas com a pasta “crud” dentro do diretório workspace.

D:\Aulas\2023\Modulo\Material\workspace				
Nome		Data de modificação	Tipo	Tamanho
crud		09/04/2023 15:33	Pasta de arquivos	

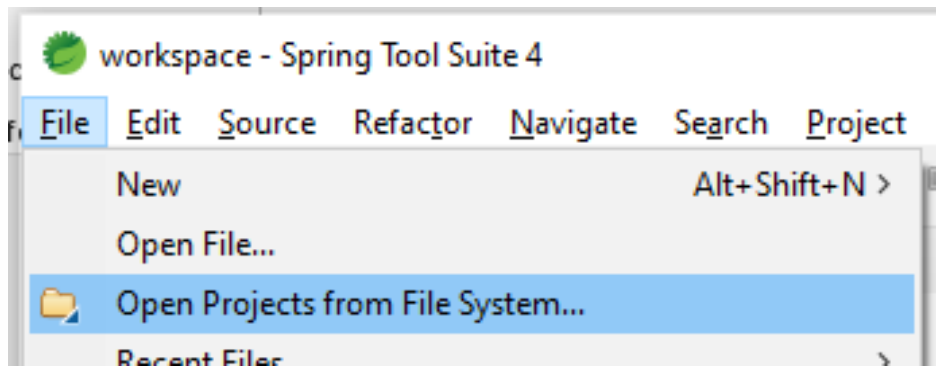
D:\Aulas\2023\Modulo\Material\workspace\crud				
Nome		Data de modificação	Tipo	Tamanho
.mvn		09/04/2023 15:33	Pasta de arquivos	
src		09/04/2023 15:33	Pasta de arquivos	
.gitignore		09/04/2023 15:33	Arquivo Fonte Git ...	1 KB
HELP.md		09/04/2023 15:33	Arquivo Fonte Ma...	2 KB
mvnw		09/04/2023 15:33	Arquivo	11 KB
mvnw.cmd		09/04/2023 15:33	Script de Comand...	7 KB
pom.xml		09/04/2023 15:33	Arquivo XML	2 KB

Abrir projeto no Spring Tool Suite

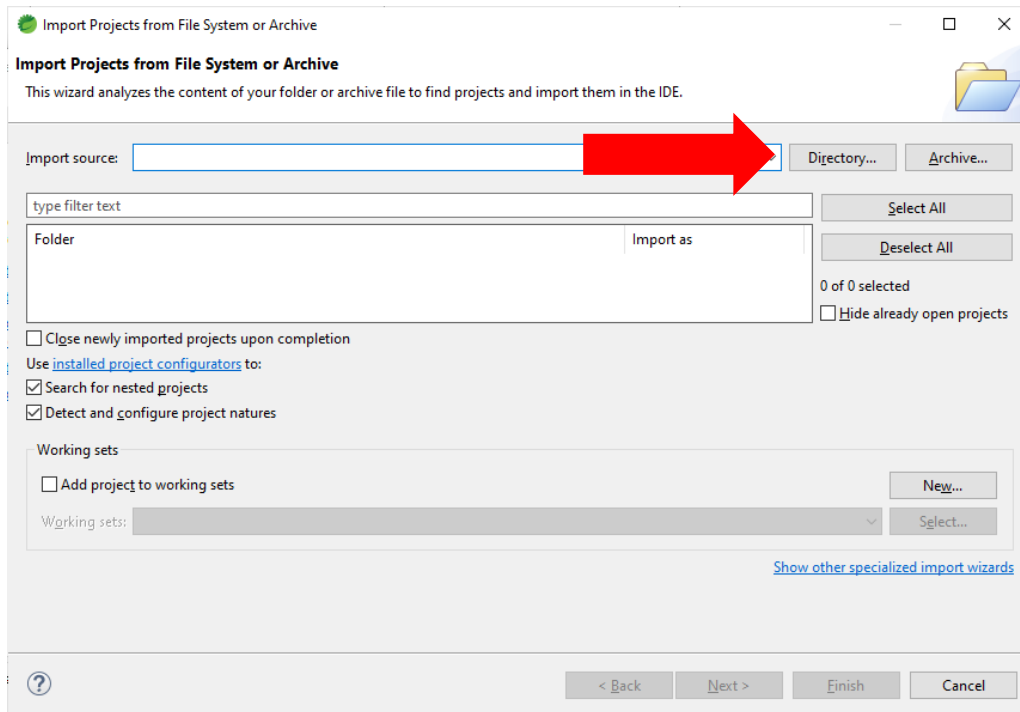
1 – Abra o Spring Tool Suite (STS) clicando no atalho na área de trabalho, preencha o campo “Workspace” com o caminho de seu diretório workspace e, por fim, clique em “Launch”.



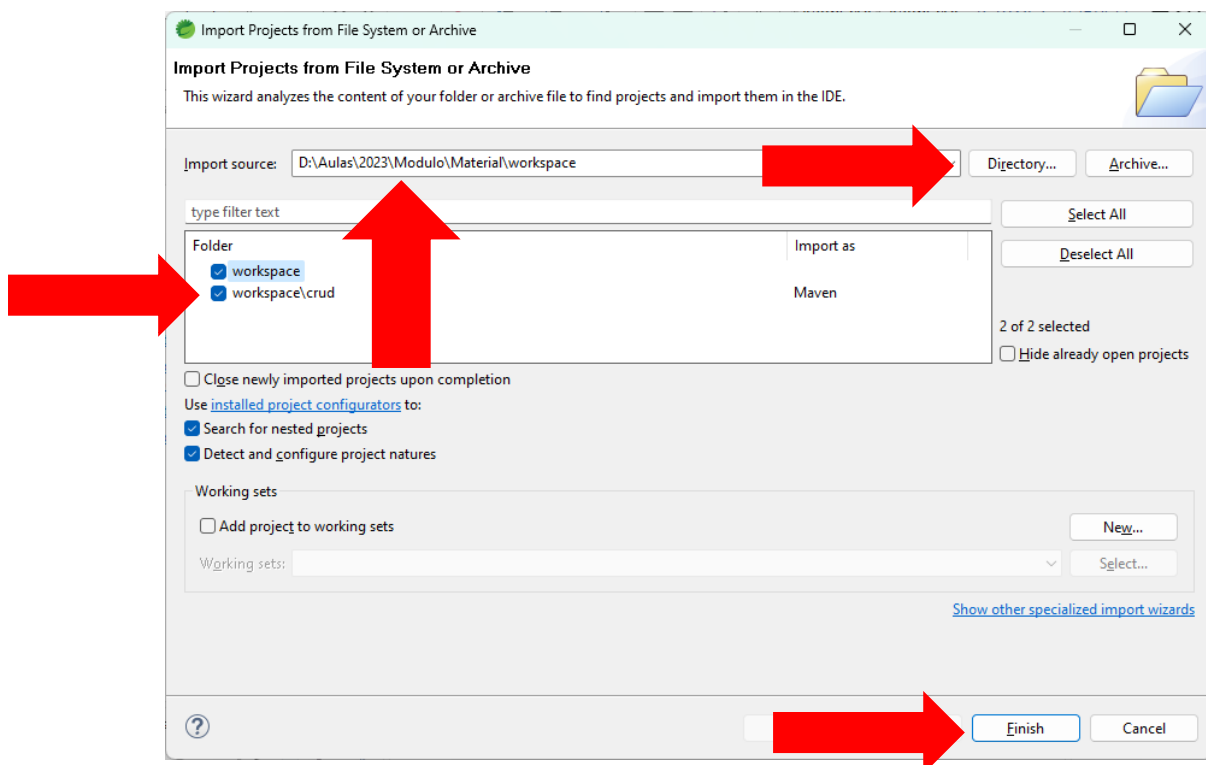
2 – Clique File > Open Projects from File System...



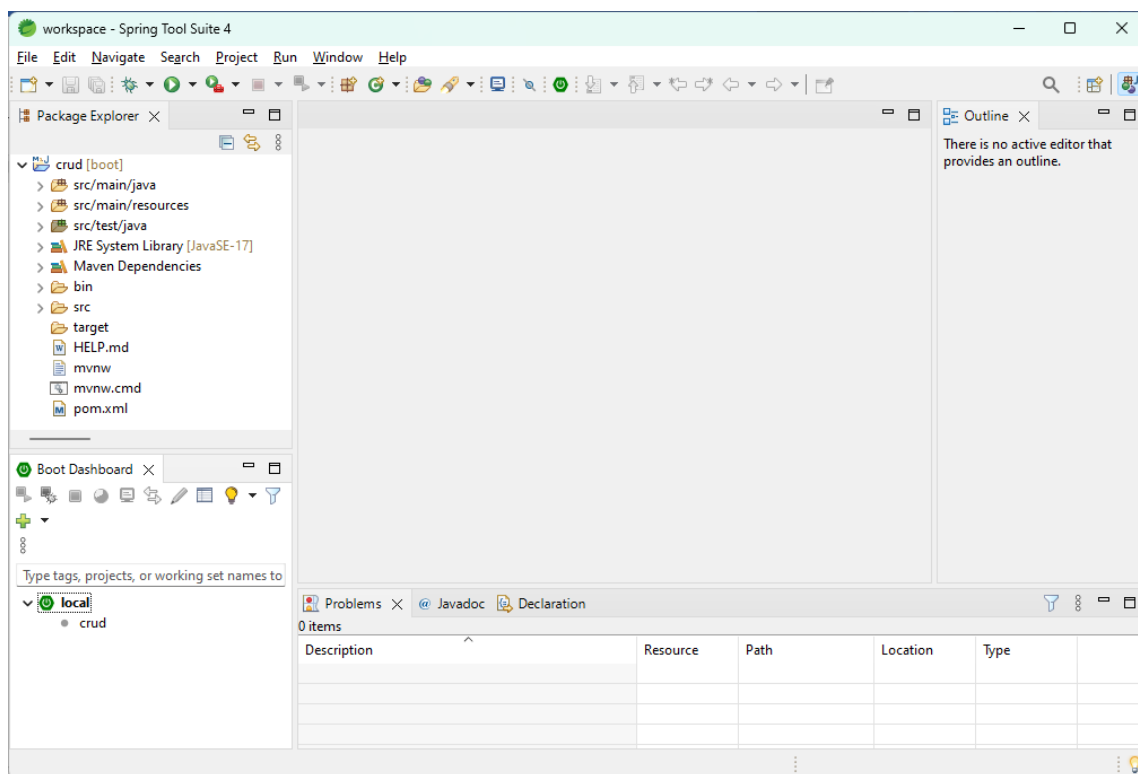
3 – Na tela seguinte clique em “Directory...” e selecione a pasta “.../workspace”



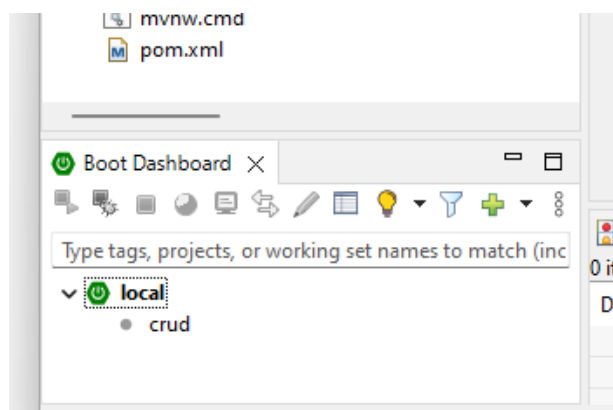
4 – Uma vez que o projeto crud esteja selecionado, clique em “Finish”.



Aguarde o Maven (gerenciador de dependências) buscar as dependências do projeto e o projeto deverá aparecer como a figura abaixo, sem nenhuma indicação de erro.

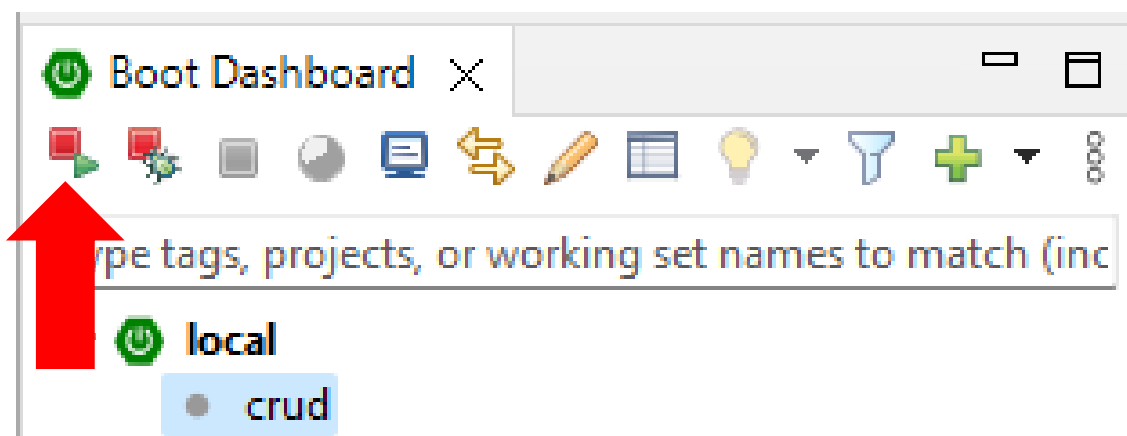


Observe que, caso a importação das dependências tenha sido bem sucedida, nosso projeto crud, deverá aparecer “dentro” de “local” na View “Boot Dashboard”.

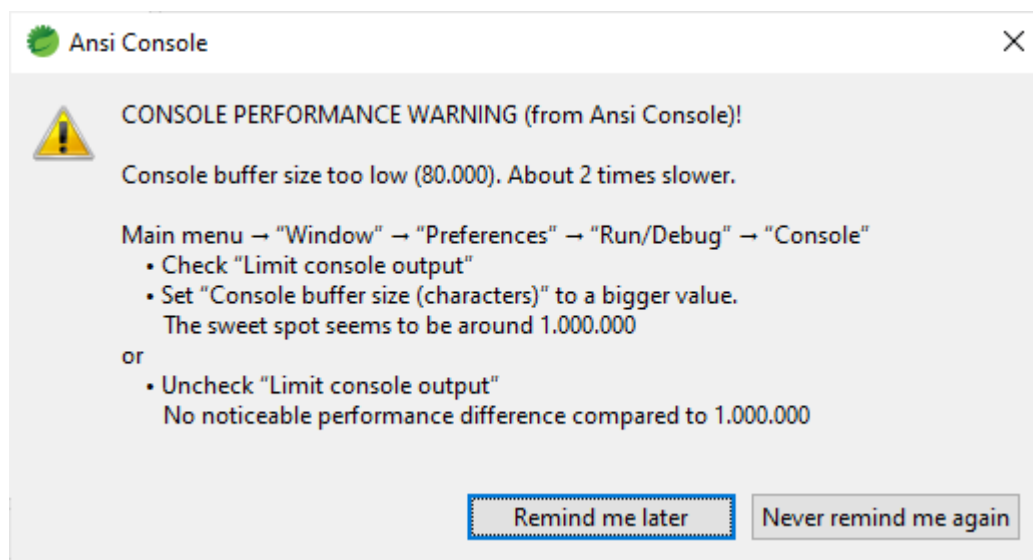


Executar o projeto

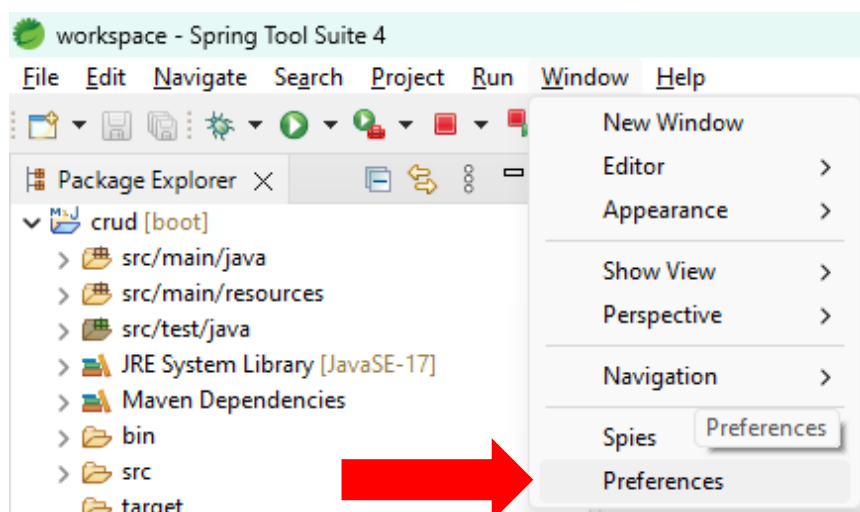
1 – Uma das formas para executar um projeto baseado no Spring Boot é, através da IDE STS, na View “Boot Dashboard”, selecionar o projeto e clicar no botão “Start or Restart”.

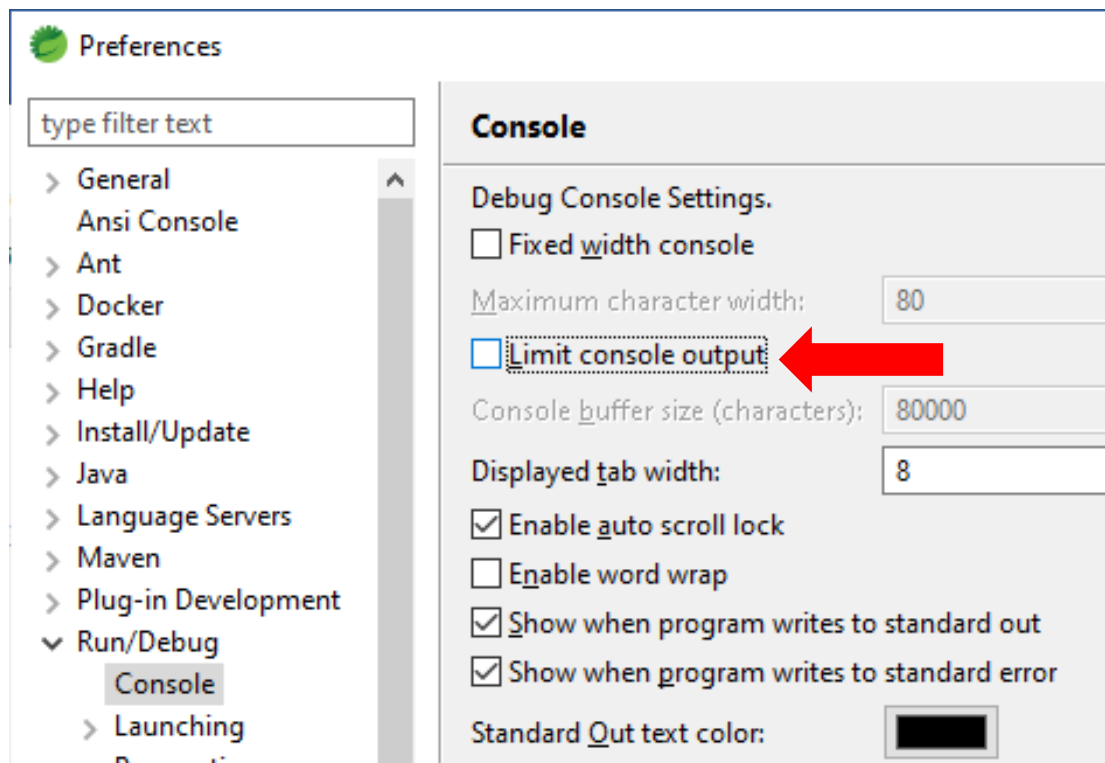


2 - Na primeira vez que você tentar executar seu projeto pode aparecer a tela a seguir, que alerta sobre problemas de performance em função do espaço reduzido do console. Clique em “Never remind me again”.

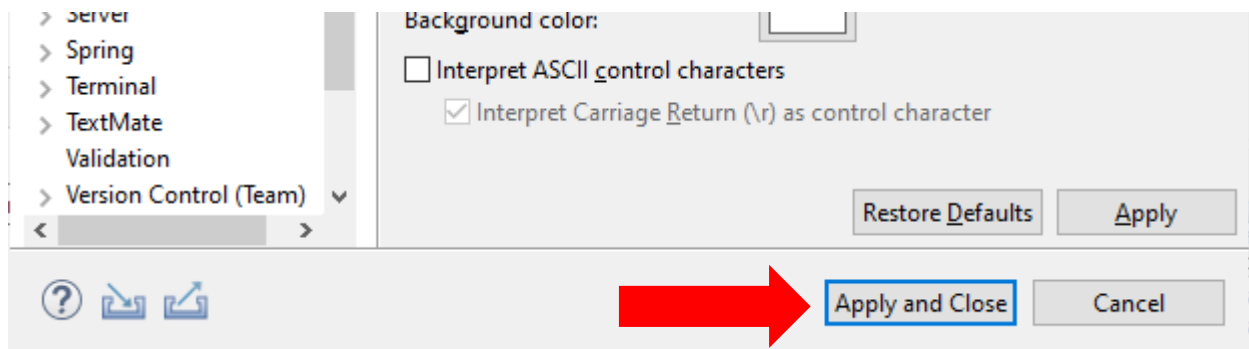


3 – Caso seu ambiente esteja muito lento você pode seguir a recomendação dessa tela e ir no menu Window > Preferences > Run/Debug > Console e desmarcar a opção “Limit console output”.





4 – Então é só clicar em “Apply and Close”



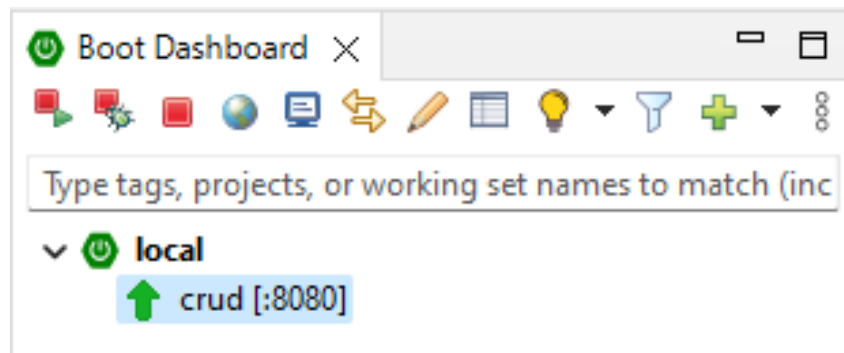
Após executar o projeto, aparecerá, no console, o registro dessa execução com algumas informações, entre elas, que o servidor web tomcat foi executado e que ele e, por extensão, a nossa aplicação também, pois esta roda dentro do tamcat, estão “ouvindo” a porta 8080 da máquina. É através dessa porta que acessaremos nossa aplicação.

```
workspace - Spring Tool Suite 4
File Edit Navigate Search Project Run Window Help
crud - CrudApplication [Spring Boot App] D:\Aulas\2023\LIP1\ambiente\sts-4.15.3.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220805-1047\jre\bin\javaw.exe (9 de abr. de 2023 15:56:56) [pid: 23308]

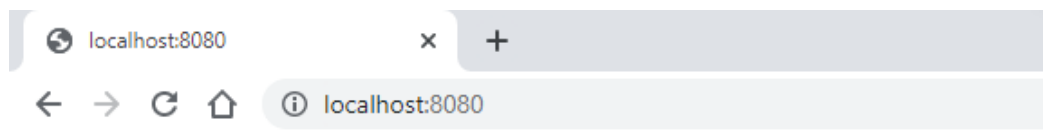
:: Spring Boot ::
(v3.0.5)

2023-04-09T15:56:58.472-03:00 INFO 23308 --- [main] br.com.belval.crud.CrudApplication : Starting CrudApplication using Java 17.0.4 with PID
23308 (D:\Aulas\2023\Modulo\Material\workspace\crud\target\classes started by takol in D:\Aulas\2023\Modulo\Material\workspace\crud)
2023-04-09T15:56:58.472-03:00 INFO 23308 --- [main] br.com.belval.crud.CrudApplication : No active profile set, falling back to 1 default
profile: "default"
2023-04-09T15:56:59.565-03:00 INFO 23308 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-04-09T15:56:59.575-03:00 INFO 23308 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-04-09T15:56:59.575-03:00 INFO 23308 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.7]
2023-04-09T15:56:59.667-03:00 INFO 23308 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-04-09T15:56:59.667-03:00 INFO 23308 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1144 ms
2023-04-09T15:56:59.924-03:00 WARN 23308 --- [main] ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:/templates/
(please add some templates, check your Thymeleaf configuration, or set spring.thymeleaf.check-template-location=false)
2023-04-09T15:57:00.112-03:00 INFO 23308 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context
path ''
2023-04-09T15:57:00.121-03:00 INFO 23308 --- [main] br.com.belval.crud.CrudApplication : Started CrudApplication in 2.01 seconds (process
running for 3.056)
```

No “Boot Dashboard”, aparecerá



5 – Abra o browser e digite na barra de endereço <http://localhost:8080>



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sun Apr 09 18:30:15 BRT 2023

There was an unexpected error (type=Not Found, status=404).

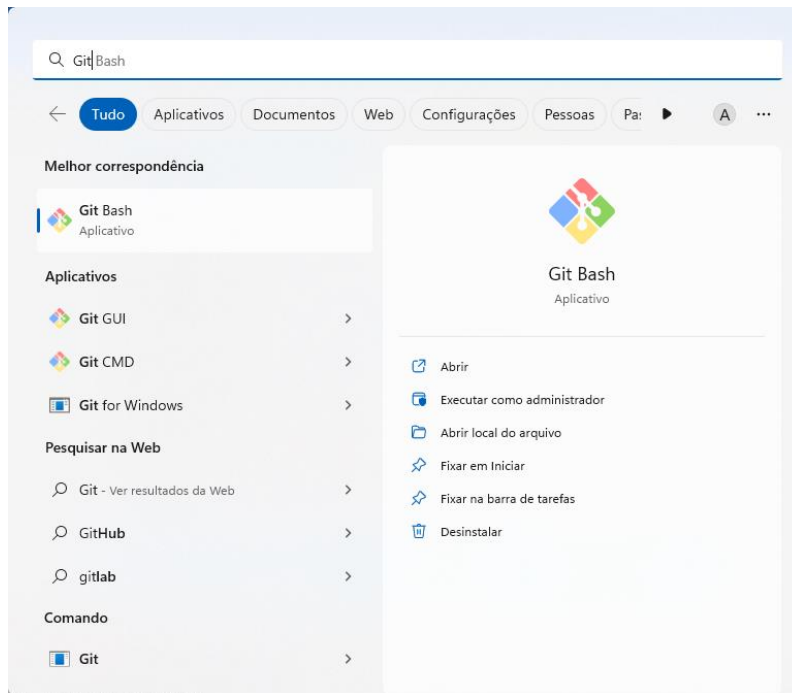
Caso apareça o texto “Whitelabel Error Page”, isso significa que nossa aplicação está funcionando.

Salvando uma fotografia do projeto

Nesse ponto temos um projeto corretamente configurado e é importante salvarmos uma fotografia desse ponto exato do projeto para que, caso venhamos a “quebrar” nosso projeto ao tentar adicionar uma nova funcionalidade, possamos voltar a ele depois.

Podemos fazer isso através do uso de ferramentas de controle de versão e a ferramenta mais utilizada hoje em dia é o Git. Ele pode ser um pouco complicado às vezes, mas se tratando de uma ferramenta tão utilizada na indústria de desenvolvimento de software é bom começarmos a aprender como utilizá-lo.

1 – Abra o Git Bash



2 – Digite o comando “cd <caminho do diretório do projeto>”

```
MINGW64;D:/Aulas/2023/Modulo/Material/workspace/crud
tako1@DESKTOP-8THFTQM MINGW64 ~
$ cd /D/Aulas/2023/Modulo/Material/workspace/crud
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$
```

3 – Configure o seu usuário e seu e-mail digitando o comando abaixo

```
git config --global user.name "Seu Nome"
git config --global user.email "seu_email@qualquercoisa.com"
```

```
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$ git config --global user.name "Alessandro"
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$ git config --global user.email "professornpc@gmail.com"
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$
```

4 – Configure os delimitadores de linha para o bom funcionamento da ferramenta diff

No Windows digite

```
git config --global core.autocrlf true
git config --global core.safecrlf warn
```

```
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$ git config --global core.autocrlf true

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$ git config --global core.safecrlf warn

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$
```

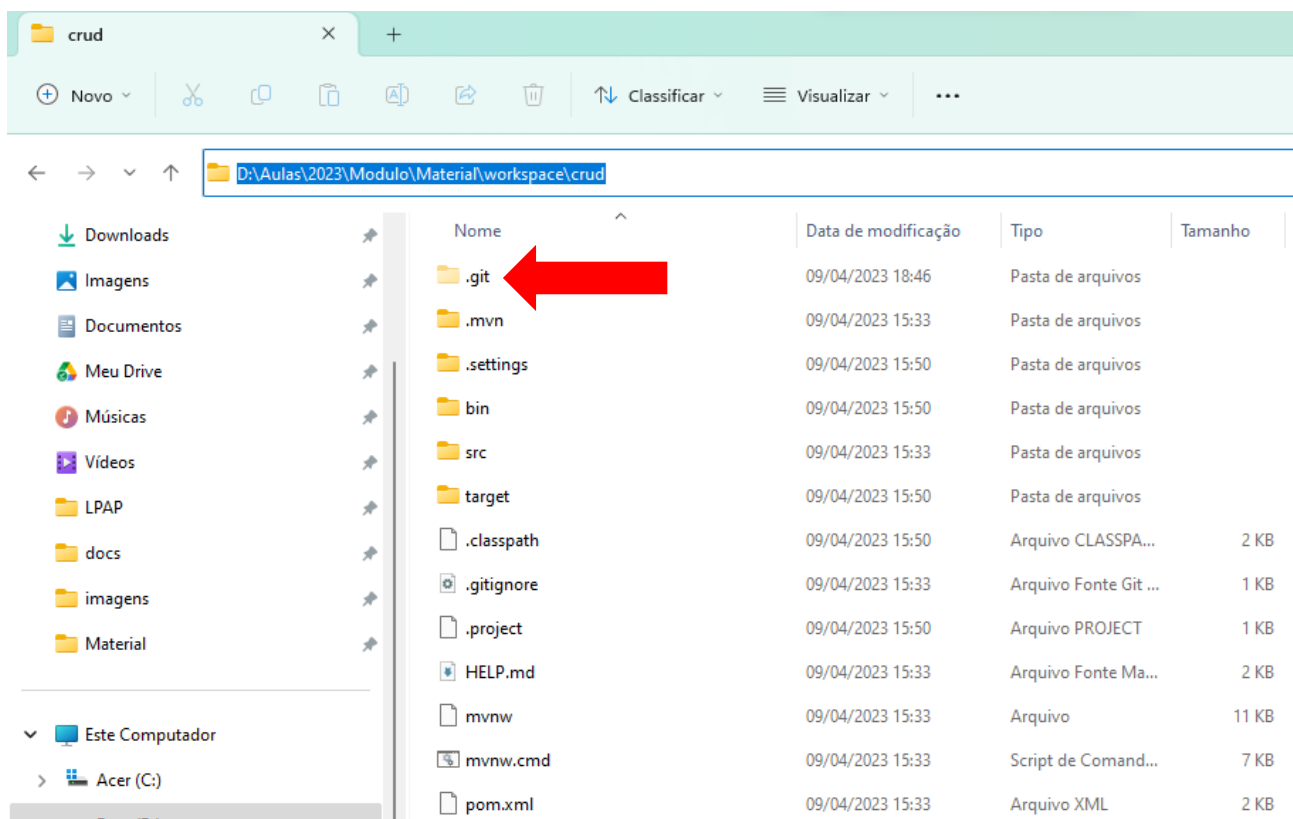
5 – Crie um novo repositório com o comando `git init`

```
MINGW64;D:/Aulas/2023/Modulo/Material/workspace/crud

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud
$ git init
Initialized empty Git repository in D:/Aulas/2023/Modulo/Material/workspace/crud/.git/

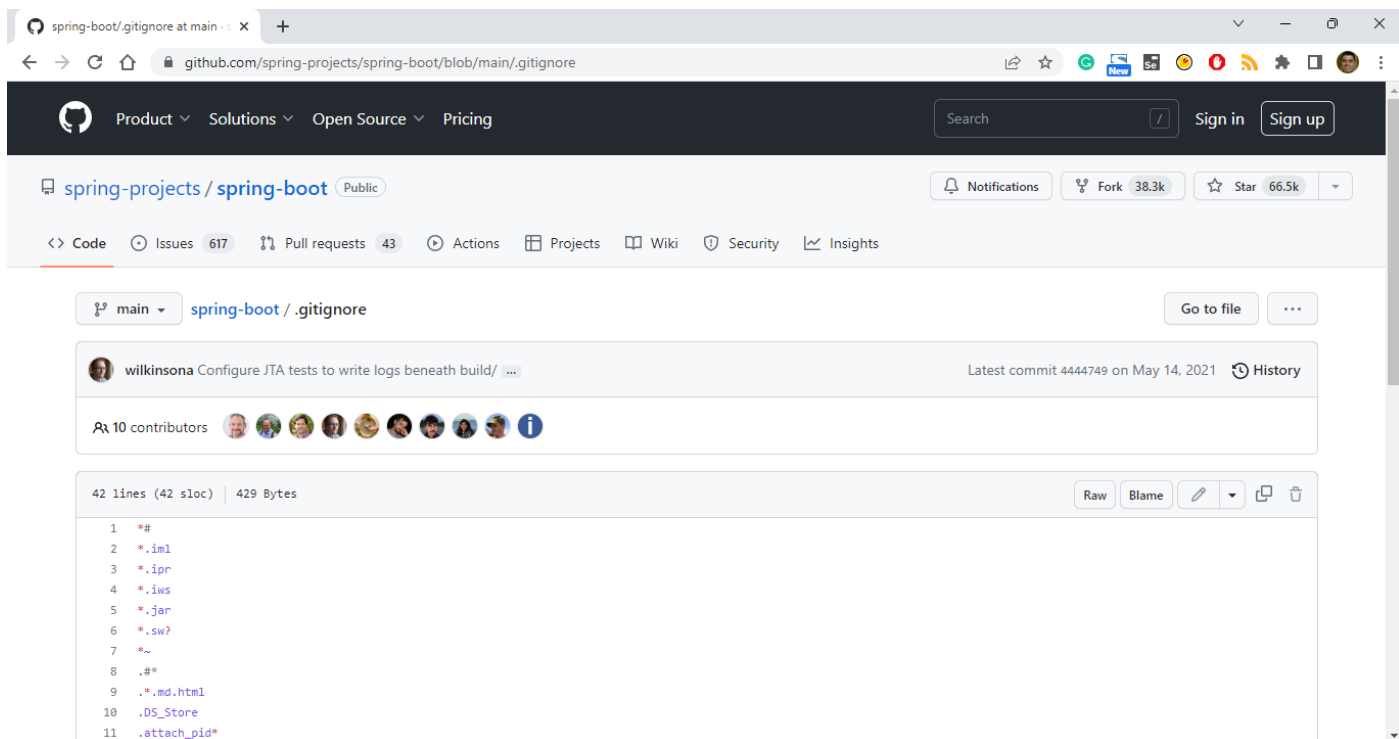
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$
```

Ao executar esse comando, um diretório “.git”, oculto, é criado dentro do diretório app1

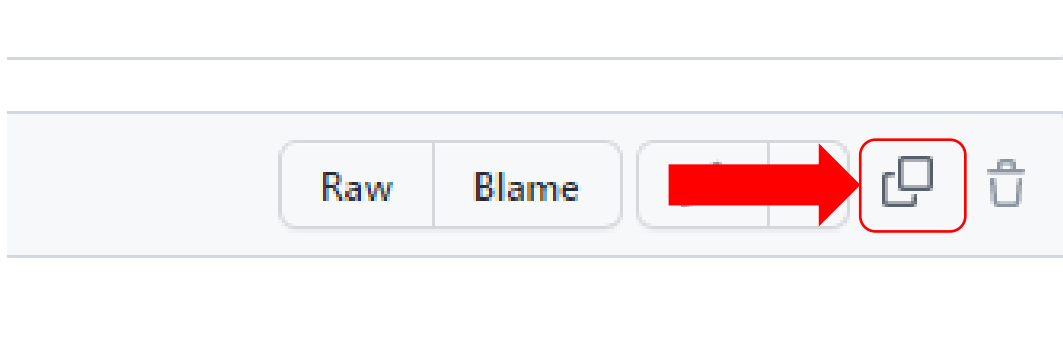


Esse diretório será o repositório que conterà todas as diferentes versões do nosso software nos permitindo recuperar versões antigas caso necessário. NÃO APAGUE ESSE DIRETÓRIO.

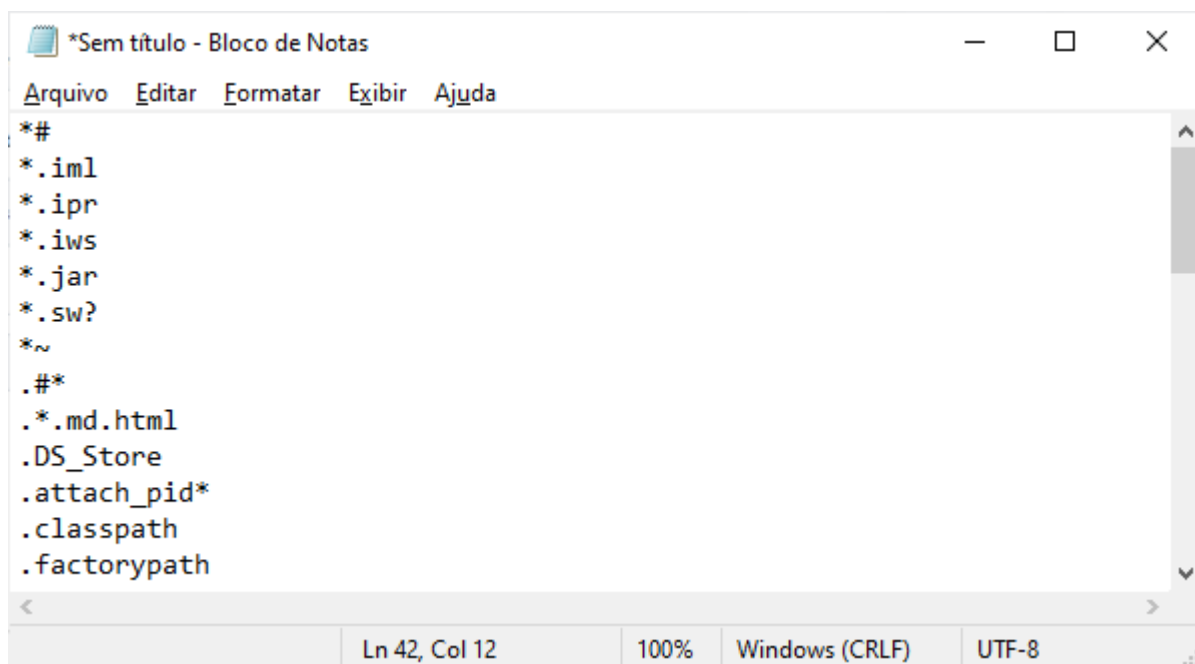
6 – Acesso o endereço <https://github.com/spring-projects/spring-boot/blob/main/.gitignore>



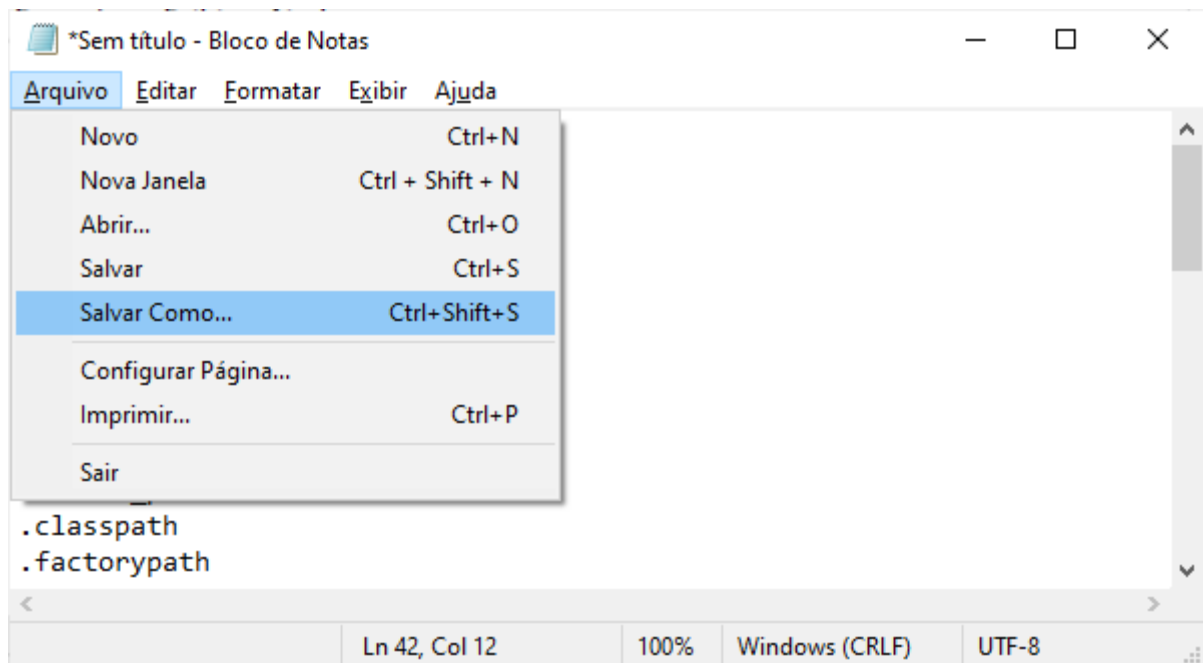
7 – Clique no botão copiar



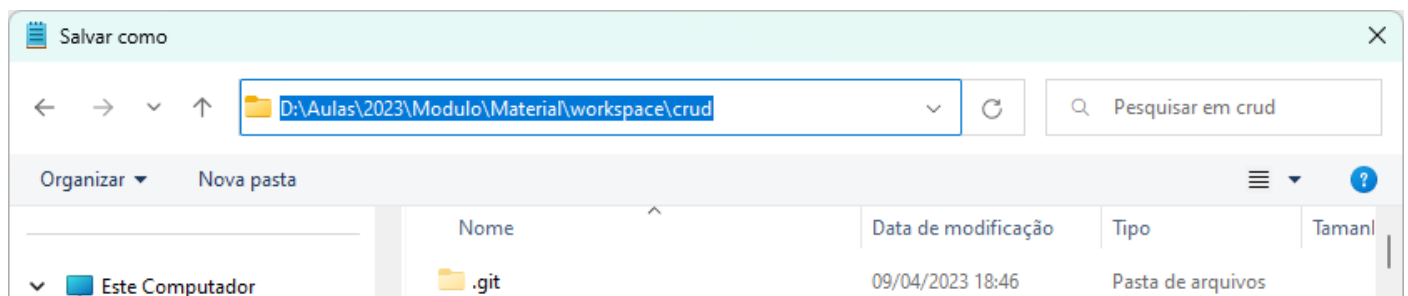
8 – Abra o notepad e cole o conteúdo copiado



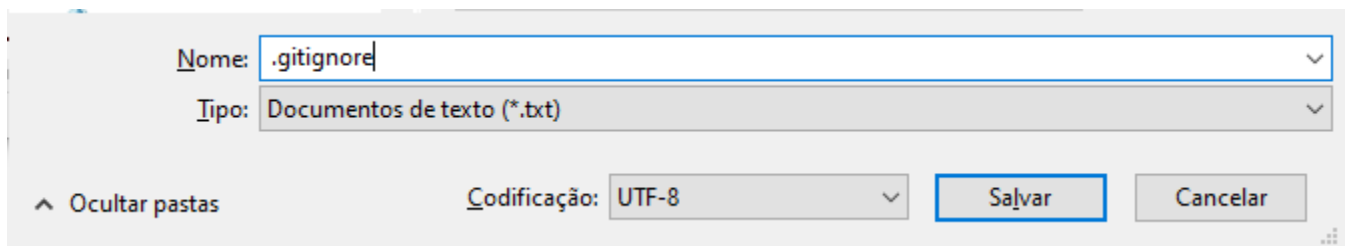
9 – Clique em Arquivo > Salvar como...



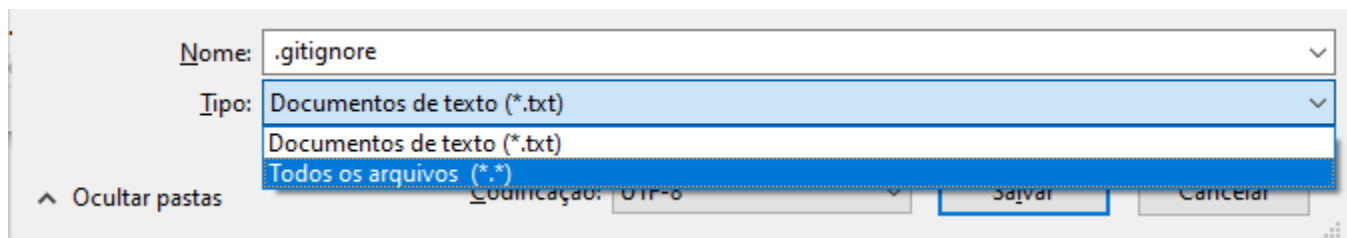
10 – Selecione a pasta do projeto (.../workspace/crud)



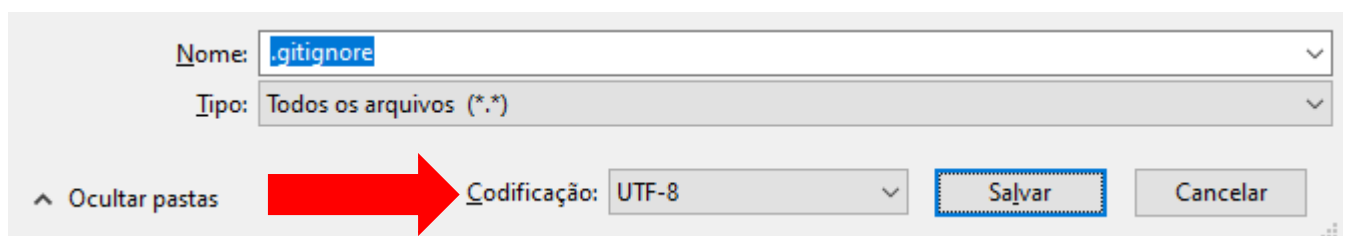
11 – Preencha o nome do arquivo como “.gitignore” (observe o ponto inicial)



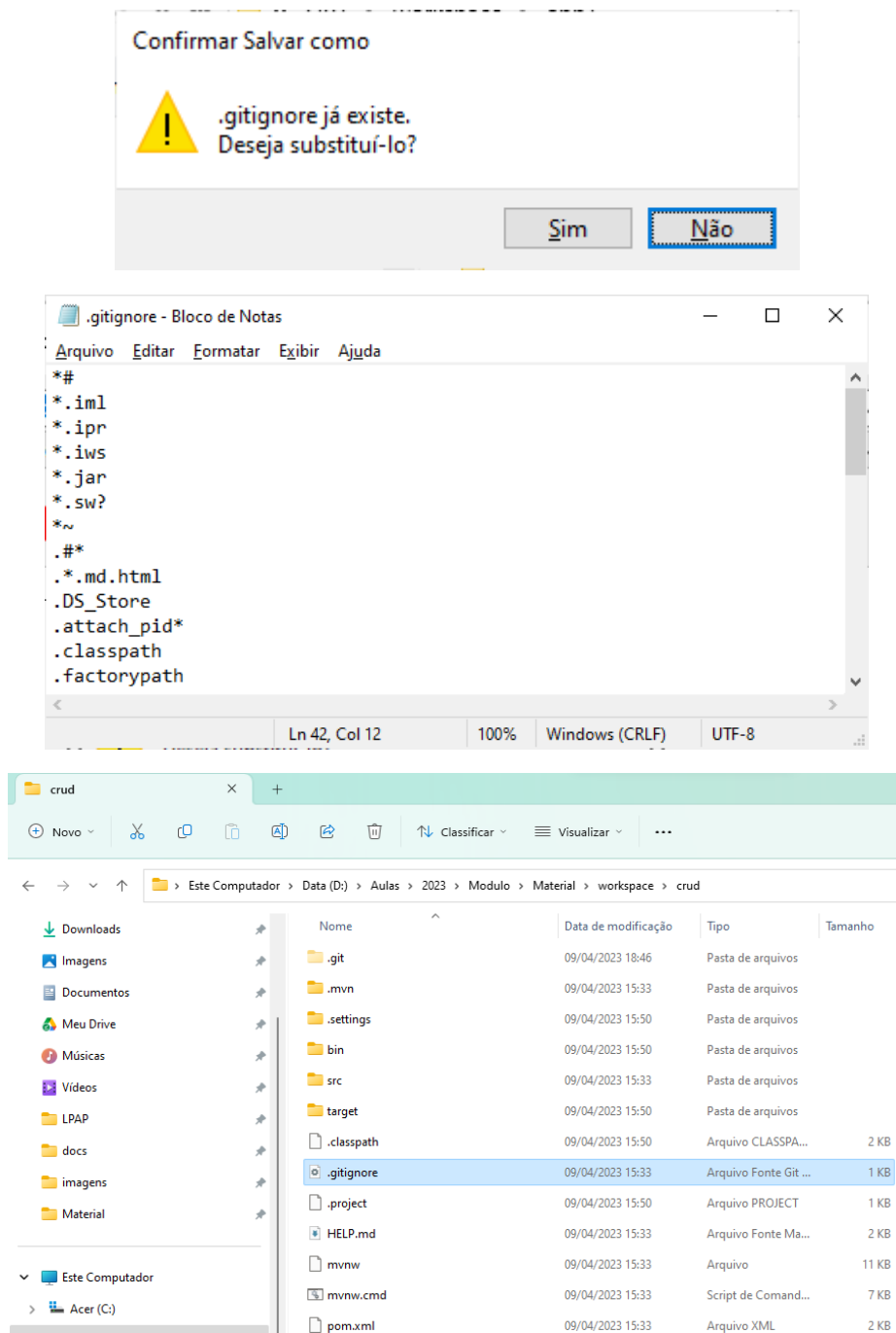
12 – Selecione o tipo “Todos os arquivos (*.*)”



13 – Deixe a codificação como “UTF-8” e clique em Salvar



14 – Se o arquivo já existir e for perguntado se deseja substituí-lo, clique em Sim.



15 – Volte ao git bash e digite os comandos abaixo

```
git add .gitignore
```

```
git status
```

```
tako1@DESKTOP-8THFTQM MINGW64 /D:/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git add .gitignore

tako1@DESKTOP-8THFTQM MINGW64 /D:/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .mvn/
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
    src/

tako1@DESKTOP-8THFTQM MINGW64 /D:/Aulas/2023/Modulo/Material/workspace/crud (main)
$ |
```

16 – Em seguida digite o comando

```
git commit -m "Add .gitignore"
```

```
tako1@DESKTOP-8THFTQM MINGW64 /D:/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git commit -m "Add .gitignore"
[main (root-commit) cf20477] Add .gitignore
1 file changed, 42 insertions(+)
create mode 100644 .gitignore

tako1@DESKTOP-8THFTQM MINGW64 /D:/Aulas/2023/Modulo/Material/workspace/crud (main)
$ |
```

Isso fará com que o git exclua os arquivos compilados deixando apenas o código fonte dentro do repositório e com isso deixando o repositório mais leve.

16 – Execute o comando abaixo para ver o status do “ambiente de trabalho” do git

```
git status
```

```
tako1@DESKTOP-8THFTQM MINGW64 /D:/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .mvn/
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
    src/

nothing added to commit but untracked files present (use "git add" to track)

tako1@DESKTOP-8THFTQM MINGW64 /D:/Aulas/2023/Modulo/Material/workspace/crud (main)
$
```

17 – Agora, vamos adicionar os fontes do projeto para “stage” do git, digite o comando abaixo

```
git add .
```

```
FTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
Working copy of '.mvn/wrapper/maven-wrapper.properties', LF will be replaced by CRLF the
Working copy of 'HELP.md', LF will be replaced by CRLF the next time Git touches it
Working copy of 'mvnw', LF will be replaced by CRLF the next time Git touches it
Working copy of 'mvnw.cmd', LF will be replaced by CRLF the next time Git touches it
Working copy of 'pom.xml', LF will be replaced by CRLF the next time Git touches it
Working copy of 'src/main/java/br/com/belval/crud/CrudApplication.java', LF will be repla
nes it
Working copy of 'src/main/resources/application.properties', LF will be replaced by CRLF
Working copy of 'src/test/java/br/com/belval/crud/CrudApplicationTests.java', LF will be
touches it

FTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
```

18 – Execute um comando “git status” para ver o que será adicionado ao repositório

```
MINGW64:/D/Aulas/2023/Modulo/Material/workspace/crud
takol@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .mvn/wrapper/maven-wrapper.properties
    new file:   HELP.md
    new file:   mvnw
    new file:   mvnw.cmd
    new file:   pom.xml
    new file:   src/main/java/br/com/belval/crud/CrudApplication.java
    new file:   src/main/resources/application.properties
    new file:   src/test/java/br/com/belval/crud/CrudApplicationTests.java

takol@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ |
```

19 – Execute o comando abaixo para salvar a versão atual dos fontes no repositório.

```
git commit -m "Versão inicial do projeto crud"
```

```
takol@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git commit -m "Versão inicial do projeto crud"
[main 5ed491a] Versão inicial do projeto crud
 8 files changed, 618 insertions(+)
 create mode 100644 .mvn/wrapper/maven-wrapper.properties
 create mode 100644 HELP.md
 create mode 100644 mvnw
 create mode 100644 mvnw.cmd
 create mode 100644 pom.xml
 create mode 100644 src/main/java/br/com/belval/crud/CrudApplication.java
 create mode 100644 src/main/resources/application.properties
 create mode 100644 src/test/java/br/com/belval/crud/CrudApplicationTests.java

takol@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$
```

20 – Verifique o histórico das versões do seu software digitando o comando abaixo

```
git log
```

```
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git log
commit 5ed491a3ae059f49599727d989f929caad26f84c (HEAD -> main)
Author: Alessandro <professornpc@gmail.com>
Date: Sun Apr 9 19:54:05 2023 -0300

    Versão inicial do projeto crud

commit cf20477fd86b19f67060519142259521cc297305
Author: Alessandro <professornpc@gmail.com>
Date: Sun Apr 9 19:47:11 2023 -0300

    Add .gitignore

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$
```

Fazendo a aplicação responder algo

1 – Abra o arquivo CrudApplication.java

2 – Para que nossa classe possa responder alguma coisa, a primeira coisa a fazer é “anotar” a classe como sendo um “Controller”. Talvez a versão mais simples de controller que podemos utilizar são os “RestControllers” que permitirão que nossos métodos respondam um texto diretamente para os clientes.

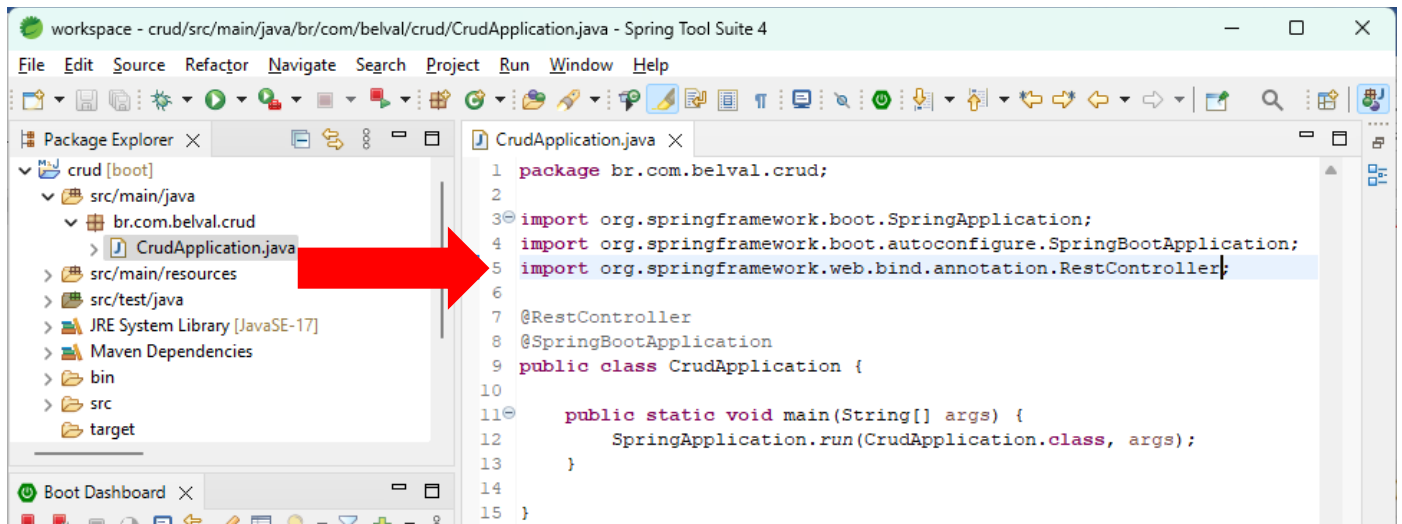
Coloque “@RestController” antes de “@SpringBootApplication”

Como você pode ver, o nosso fonte apresenta um erro de compilação.

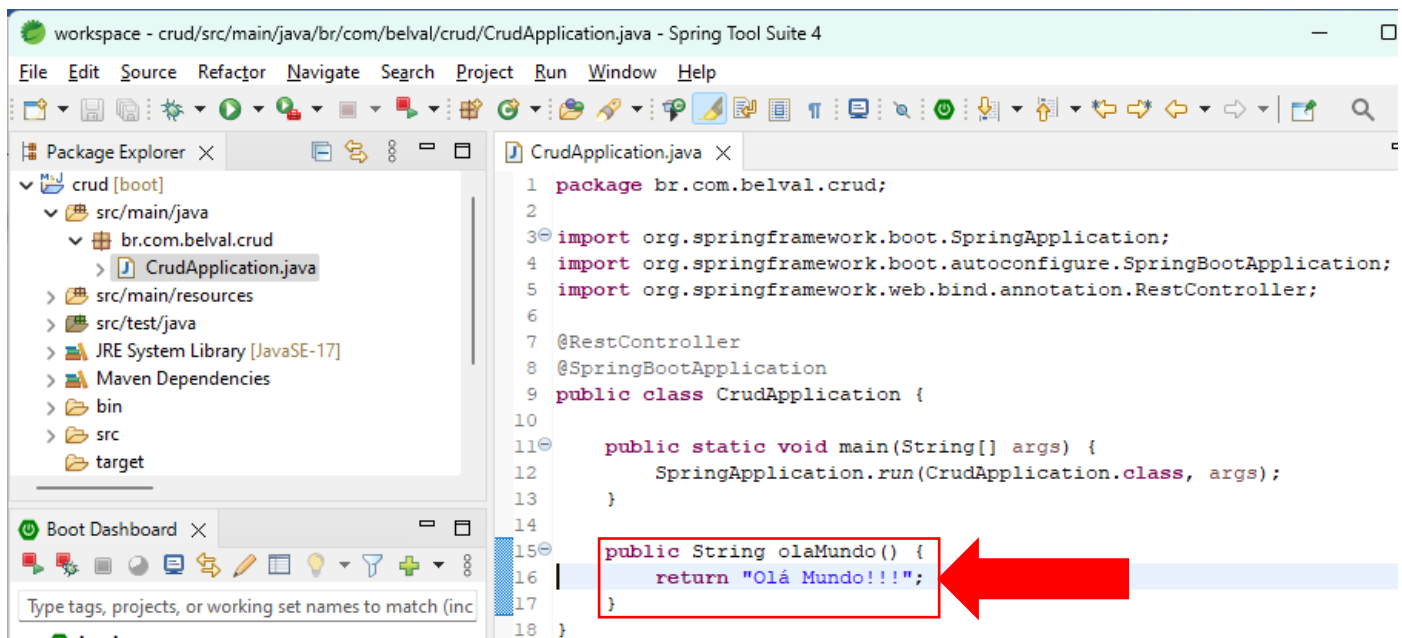
O problema aqui é que não adicionamos a importação da classe correspondente à anotação “@RestController”.

Para resolver isso podemos usar um atalho.

3 – Pressione “CTRL + SHIFT + O” que o ambiente automaticamente adiciona o *import* necessário.



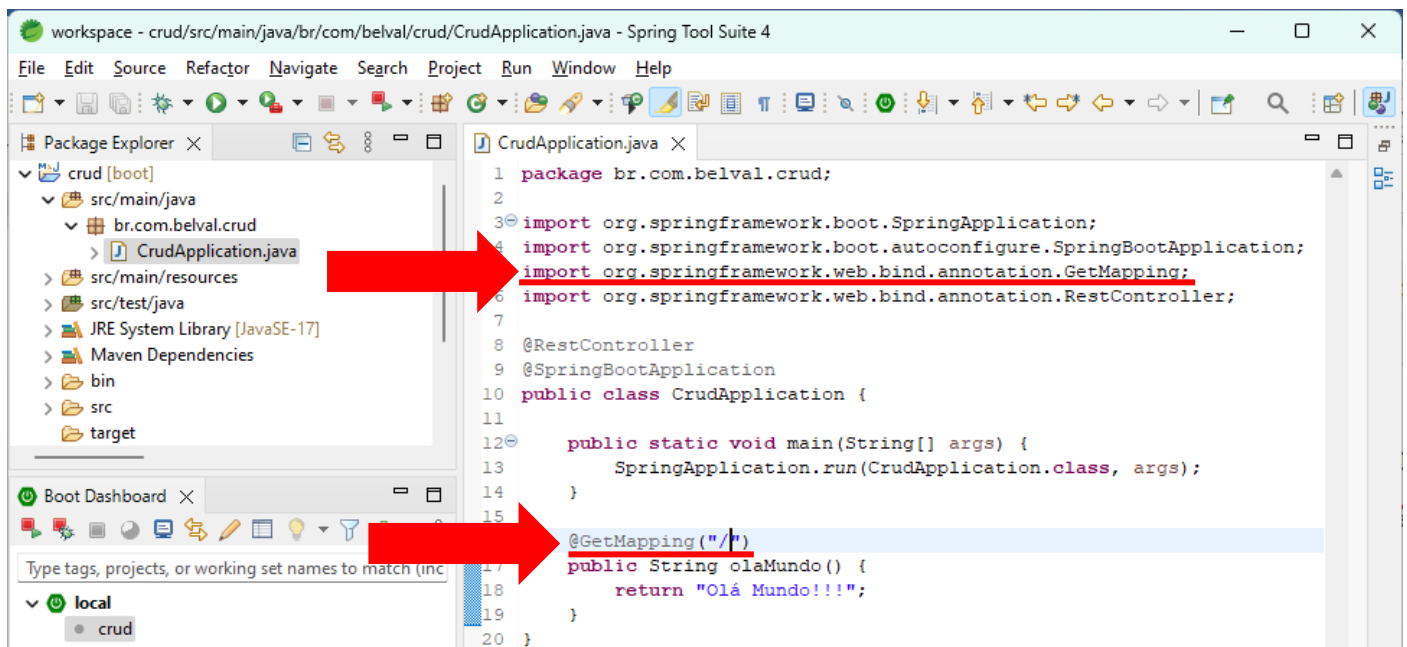
4 – Crie um método olaMundo() como abaixo



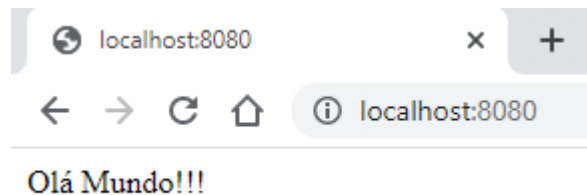
5 – Anote esse método com

@GetMapping("/")

E na sequência use o atalho “CTRL + SHIFT + O” para adicionar o *import* necessário.



6 – Execute a aplicação novamente e acesse o endereço “localhost:8080/”



7 – Pare a aplicação, e no git bash, execute o comando “git diff”

```
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git diff
warning: in the working copy of 'src/main/java/br.com.belval.crud/CrudApplication.java', LF will be replaced by CRLF the n
ext time Git touches it
diff --git a/src/main/java/br.com.belval.crud/CrudApplication.java b/src/main/java/br.com.belval.crud/CrudApplication.java
index 5366afe..5eea002 100644
--- a/src/main/java/br.com.belval.crud/CrudApplication.java
+++ b/src/main/java/br.com.belval.crud/CrudApplication.java
@@ -2,7 +2,10 @@ package br.com.belval.crud;

 import org.springframework.boot.SpringApplication;
 import org.springframework.boot.autoconfigure.SpringBootApplication;
+import org.springframework.web.bind.annotation.GetMapping;
+import org.springframework.web.bind.annotation.RestController;
+
+@RestController
+@SpringBootApplication
 public class CrudApplication {
@@ -10,4 +13,8 @@ public class CrudApplication {
     SpringApplication.run(CrudApplication.class, args);
 }
+
+    @GetMapping("/")
+    public String olaMundo() {
+        return "Olá Mundo!!!";
+    }
+}
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$
```

Observe que, as linhas que começam com “+” e que aqui também estão verdes, são aquelas que foram adicionadas em relação à última versão do nosso software.

8 – Se estiver tudo certo, execute os comandos abaixo e salve uma nova fotografia do nosso projeto.

```
git add .
git commit -m "Add método olaMundo()"
```

```
tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git add .
warning: in the working copy of 'src/main/java/br/com/belval/crud/CrudApplication.java', LF will be replaced by CRLF the next time Git touches it

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git commit -m "Add método olaMundo()"
[main d7daee7] Add método olaMundo()
1 file changed, 7 insertions(+)

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$
```

9 – Para verificar o histórico do nosso projeto, execute o comando “git log”

```
MINGW64:/D/Aulas/2023/Modulo/Material/workspace/crud

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$ git log
commit d7daee76a8e42698962837e6d341422ee1d49421 (HEAD -> main)
Author: Alessandro <professornpc@gmail.com>
Date: Sun Apr 9 22:36:56 2023 -0300

    Add método olaMundo()

commit 5ed491a3ae059f49599727d989f929caad26f84c
Author: Alessandro <professornpc@gmail.com>
Date: Sun Apr 9 19:54:05 2023 -0300

    Versão inicial do projeto crud

commit cf20477fd86b19f67060519142259521cc297305
Author: Alessandro <professornpc@gmail.com>
Date: Sun Apr 9 19:47:11 2023 -0300

    Add .gitignore

tako1@DESKTOP-8THFTQM MINGW64 /D/Aulas/2023/Modulo/Material/workspace/crud (main)
$
```