



Universidade Federal do Ceará - Campus de Quixadá
Disciplina: Sistemas Distribuídos
Código: QXD0043
Professor: Rafael Braga

1. Instalar o Java: Você pode seguir os seguintes tutoriais a depender do seu sistema operacional:

Linux: https://youtu.be/QMeC_Ioin7g

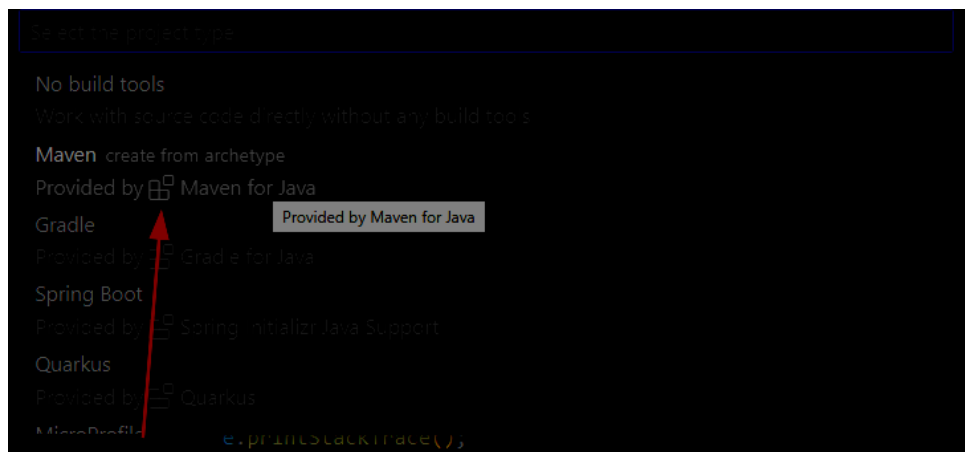
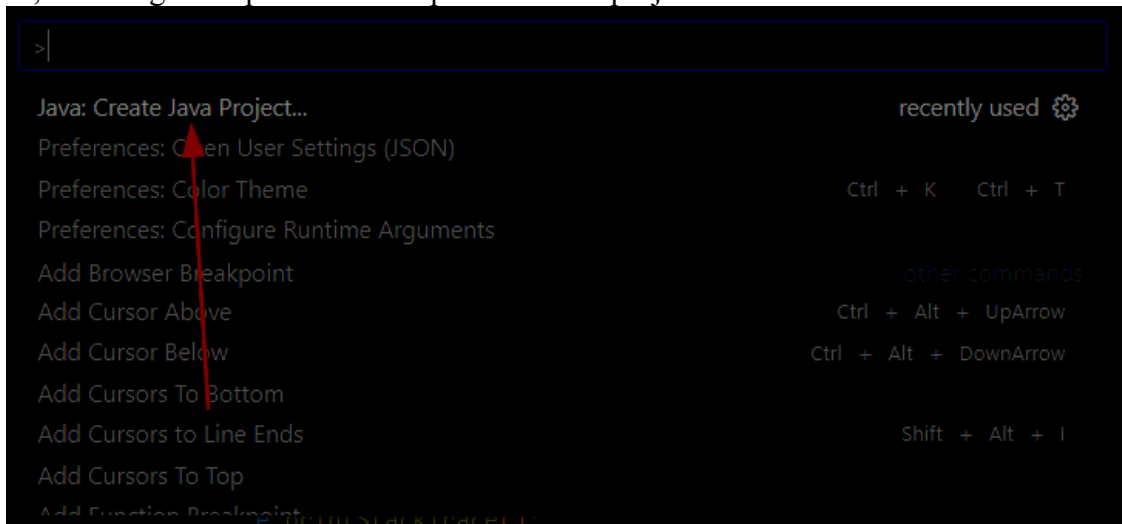
Windows: <https://youtu.be/sNFii-cvNz0>

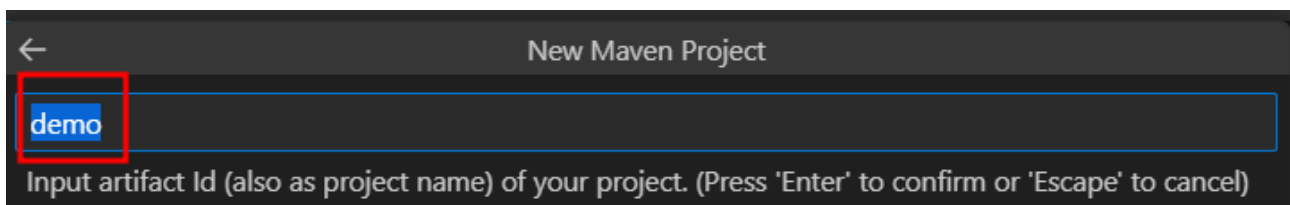
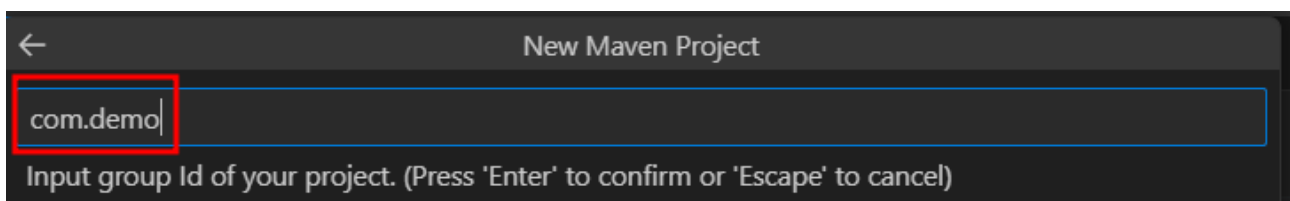
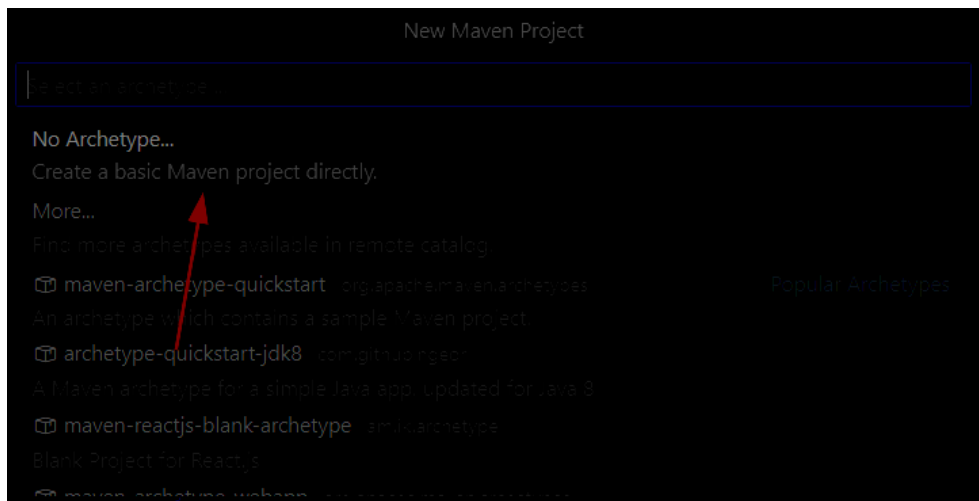
2. Instalar o Maven: Você pode seguir os seguintes tutoriais a depender do seu sistema operacional:

Linux: <https://www.digitalocean.com/community/tutorials/install-maven-linux-ubuntu>

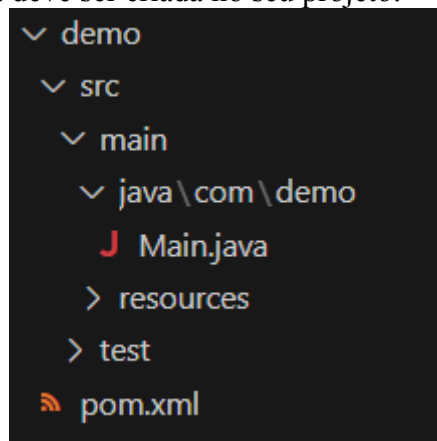
Windows: <https://youtu.be/x-VtjGWc94>

3. Criando o projeto: No VSCode, aperte Ctrl + Shift + P para abrir a barra de comandos. A partir de agora, basta seguir os passos abaixo para criar seu projeto:





Uma estrutura semelhante a essa deve ser criada no seu projeto:



4. Configure o pom.xml: Navegue até o diretório do projeto e abra o arquivo pom.xml.

O pom.xml será bem básico:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.6.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.demo</groupId>
  <artifactId>demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>demo</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
    <maven-jar-plugin.version>3.1.1</maven-jar-plugin.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

5. Criação do Código

No diretório `src/main/java/com/demo`, adicione o seguinte código em `Main.java`:

```
package com.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@SpringBootApplication
public class Main {

    @RequestMapping("/")
    String home() {
        return "Olá, seja bem vindo ao teste de Spring Boot para disciplina de Sistemas Distribuídos";
    }

    @RequestMapping("/nome/{nome}")
    String piada(@PathVariable String nome) {
        return "Olá " + nome + ", você vai realmente conseguir passar nessa disciplina?";
    }

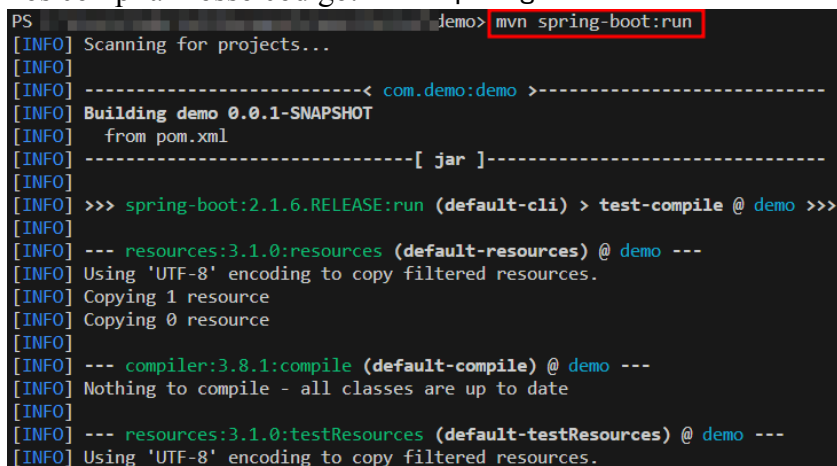
    public static void main(String[] args) {
        SpringApplication.run(Main.class, args);
    }
}
```

6. Executando o código:

Abra o cmd ou terminal na pasta do seu projeto e siga os seguintes passos:

Recomendo antes de iniciar o comando abaixo, usar este também: `mvn clean install`

Agora podemos compilar nosso código: `mvn spring-boot:run`

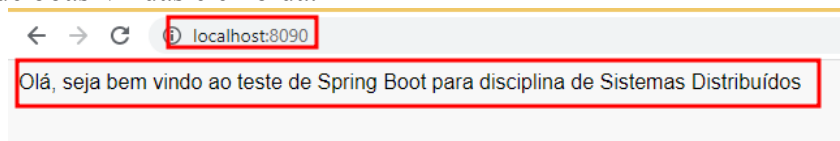
A terminal window with a dark background. The prompt is 'PS' followed by a directory path 'demo>'. The command 'mvn spring-boot:run' is entered and highlighted with a red box. The output shows Maven scanning for projects, building demo 0.0.1-SNAPSHOT from pom.xml, and running the jar. It then shows the execution of spring-boot:2.1.6.RELEASE:run (default-cli) > test-compile @ demo >>>. This is followed by resource copying and compilation steps, all of which complete successfully, indicating that the application is ready to run.

```
PS demo> mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.demo:demo >-----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot:2.1.6.RELEASE:run (default-cli) > test-compile @ demo >>>
[INFO]
[INFO] --- resources:3.1.0:resources (default-resources) @ demo ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ demo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.1.0:testResources (default-testResources) @ demo ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
```

Caso tudo tenha ocorrido bem, você verá essa tela:

[illegible]

Ao acessar a URL `http://localhost:8090/` (caso você siga o passo a passo, a porta será 8080) uma mensagem de boas-vindas é exibida.



Você também pode passar valores para o endpoint `http://localhost:8090/nome/` como no exemplo abaixo:

