



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

Roteiro de Laboratório: WebServices com Spring Boot e Interface Gráfica HTML

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

Siga os tutoriais abaixo conforme seu sistema operacional:

- Linux: <https://www.digitalocean.com/community/tutorials/install-maven-linux-ubuntu>
- Windows: <https://youtu.be/x-VtjfGWc94>

3. Criando o Projeto

No VSCode, pressione Ctrl + Shift + P para abrir a barra de comandos e siga os passos abaixo:

1. Selecione "New Maven Project".
2. Escolha "No Archetype" para criar um projeto básico.
3. Insira os seguintes dados quando solicitado:
 - Group Id: com.calculadora
 - Artifact Id: calculadora

Uma estrutura semelhante a esta será criada:

- ✓ calculadora
 - ✓ src
 - ✓ main
 - ✓ java/com/calculadora
 - J Main.java
 - ✓ resources
 - static
 - templates
 - ✓ test

✓ pom.xml

4. Configurar o pom.xml

Abra o arquivo pom.xml e adicione o seguinte conteúdo:

```
<?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/>
  </parent>
  <groupId>com.calculadora</groupId>
  <artifactId>calculadora</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>calculadora</name>
  <description>Projeto de Calculadora com Spring Boot</description>

  <properties>
    <java.version>1.8</java.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-thymeleaf</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 WebService (API REST)

No diretório src/main/java/com/calculadora, crie o arquivo Main.java com o seguinte código:

```

package com.calculadora;

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

@RestController
@SpringBootApplication
public class Main {

    @GetMapping("/somar/{a}/{b}")
    public String somar(@PathVariable double a, @PathVariable double b) {
        return "Resultado: " + (a + b);
    }

    @GetMapping("/subtrair/{a}/{b}")
    public String subtrair(@PathVariable double a, @PathVariable double b) {
        return "Resultado: " + (a - b);
    }

    @GetMapping("/multiplicar/{a}/{b}")
    public String multiplicar(@PathVariable double a, @PathVariable double b) {
        return "Resultado: " + (a * b);
    }

    @GetMapping("/dividir/{a}/{b}")
    public String dividir(@PathVariable double a, @PathVariable double b) {
        if (b == 0) {
            return "Erro: Divisão por zero!";
        }
        return "Resultado: " + (a / b);
    }
}

```

```

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

```

Interface Gráfica HTML

Crie um arquivo index.html em src/main/resources/static:

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Calculadora Web</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 20px; }
    .container { max-width: 400px; margin: auto; }
    input, button { margin: 5px 0; padding: 8px; width: 100%; }
    button { background-color: #4CAF50; color: white; border: none; cursor:
pointer; }
    button:hover { background-color: #45a049; }
  </style>
</head>
<body>
  <div class="container">
    <h2>Calculadora Web</h2>
    <input type="number" id="num1" placeholder="Número 1">
    <input type="number" id="num2" placeholder="Número 2">
    <button onclick="calcular('somar')">Somar</button>
    <button onclick="calcular('subtrair')">Subtrair</button>
    <button onclick="calcular('multiplicar')">Multiplicar</button>
    <button onclick="calcular('dividir')">Dividir</button>
    <p id="resultado"></p>
  </div>

  <script>
    function calcular(operacao) {
      const num1 = parseFloat(document.getElementById("num1").value);
      const num2 = parseFloat(document.getElementById("num2").value);
      fetch(`http://localhost:8080/${operacao}/${num1}/${num2}`)
        .then(response => response.text())
        .then(data => document.getElementById("resultado").innerText = data)
        .catch(error => console.error("Erro:", error));
    }
  </script>

```

```
</script>  
</body>  
</html>
```

6. Executando o Código

- Abra o terminal na pasta do projeto e execute:

```
mvn clean install  
mvn spring-boot:run
```

- Acesse a interface gráfica em:

<http://localhost:8080/index.html>

- Teste os endpoints da API diretamente:

Somar: <http://localhost:8080/somar/5/3>

Subtrair: <http://localhost:8080/subtrair/5/3>

7. Resultados Esperados

A interface gráfica permitirá realizar cálculos básicos.
Os endpoints retornarão os resultados em formato de texto.