SENAI GASPAR RICARDO JUNIOR

MANUELA BERNARDI DE LIMA

**MANUAL CALCULADORA E QUESTIONÁRIO**

APLICATIVO: ECLIPSE – PARTE 02

Logotipo

O conteúdo gerado por IA pode estar incorreto.

SOROCABA - SP

2025

**QUESTIONÁRIO – ANNOTATION**

1) Explique o papel da anotação @**PostMapping**em uma API REST desenvolvida com Spring. Como ela ajuda a mapear as requisições HTTP para métodos específicos na classe controladora? Dê exemplos de quando utilizar @PostMapping e qual a diferença entre essa anotação e outras como @GetMapping.

- O que faz: Mapeia requisições HTTP do tipo POST para um método específico na classe controladora.

- Como ajuda: Quando o cliente envia uma requisição POST para uma URL definida, o Spring chama o método anotado com @PostMapping para tratar essa requisição.

- Quando usar: Para operações que criam ou enviam dados ao servidor, como cadastro, envio de formulários etc.

- Diferença para @GetMapping:

@GetMapping mapeia requisições HTTP do tipo GET, geralmente usadas para obter ou consultar dados.

@PostMapping é para enviar dados e criar recursos, enquanto GET é só para leitura.

Exemplo:

@PostMapping("/usuario")

public void criarUsuario(@RequestBody Usuario usuario) {

// cria um novo usuário

}  
2) Descreva a anotação @**RequestParam**e explique seu papel no código fornecido. Como ela permite a obtenção dos parâmetros passados na requisição HTTP? Quais seriam as consequências de não utilizar **@RequestParam** em um método como o **calcularJSON**?

- O que faz: Captura parâmetros enviados na URL da requisição HTTP (query parameters) e os vincula a parâmetros do método.

- Como funciona: Se a requisição for, por exemplo, /calcular?valor=10, o valor 10 será passado para o parâmetro do método anotado com @RequestParam("valor").

- Consequências de não usar:

* O método não receberia os valores da URL automaticamente.
* Poderia causar erro ou o parâmetro ficaria null, impedindo o funcionamento correto do método.

3) Explique a anotação **@*ResponseBody***e como ela influencia o comportamento do método que a utiliza. No exemplo, o que acontece com o objeto retornado pelo método quando essa anotação está presente?

- O que faz: Indica que o retorno do método será convertido diretamente para o corpo da resposta HTTP (como JSON ou XML), e enviado para o cliente.

- Como influência: O Spring serializa o objeto retornado para JSON (ou outro formato) automaticamente.

- No exemplo: O objeto retornado pelo método será transformado em JSON e enviado como resposta da requisição.  
  
4) Explique o papel da anotação **@RestController**em um controlador Spring. Como ela é diferente de **@Controller** e qual é o impacto do uso de **@RestController** no comportamento da resposta da API? No contexto do código dado, o que aconteceria se @RestController fosse substituído por **@Controller**?

- O que faz: Combina @Controller + @ResponseBody, ou seja, já indica que todos os métodos da classe retornam dados que devem ser enviados no corpo da resposta (JSON por exemplo).

- Diferença para @Controller:

* @Controller normalmente retorna nomes de páginas (views), usados em aplicações web com front-end tradicional.
* @RestController é usado para APIs REST, retornando dados diretamente no corpo da resposta.

- Impacto ao trocar @RestController por @Controller:

* O método não enviaria o JSON automaticamente, e o Spring tentaria interpretar o retorno como nome de uma view, causando erro ou comportamento inesperado.

**MANUAL – PASSO A PASSO ECLIPSE**

Acesse o aplicativo: Eclipse

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Clique em File 🡪 New 🡪 Project

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Abra o Spring boot 🡪 Spring Starter Project 🡪 Next

Interface gráfica do usuário, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Altere o name (Calculadora) 🡪 Type (Maven) 🡪 Group (com. devsenai1A) 🡪 Description (Realizar cálculos matemáticos básicos) 🡪 Package (com. devsenai1a.Calculadora) 🡪 Next

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Inclua todas as dependências (Spring Boot DevTools e Spring Web) – Ao finalizar clique em Finish

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Acessar a programação (CalculadoraApplication)

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Aperte o botão direito do mouse, insira New e em seguida Package

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Ajuste o nome da sua paste e finalize

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Após a pasta criada, clica novamente com o botão direito, prossiga clicando em New e após isso você irá criar uma nova classe.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Coloque um nome para a classe e finalize.

**PROGRAMAÇÃO 01 – CALCULADORA**

Texto

O conteúdo gerado por IA pode estar incorreto.

Entre em CalculadoraController.java e desenvolva a programação.

Em seguida incremente a programação com os requisitos abaixo (não esqueça de importa o que for necessário).

Texto

O conteúdo gerado por IA pode estar incorreto.

Em seguida rode seu programa clicando com o botão direito em Calculadora.Application.java 🡪 Run as 🡪 9 Sprint Boot App;

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Por último teste sua programação no google.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Exemplo: Localhost:8080/multiplicar?a=8&b=5