

2TDSPC – Checkpoint 1 – 2º Semestre – 07/03/2025

Orientações: Responder sempre com as próprias palavras (parafrasear).

- Quanto mais detalhado, melhor. Mas seja objetivo.
- Responder num documento do word e entregar no Teams.
- Consultar o projeto, internet, IA, Stackoverflow, mas não pergunte para outro aluno.
- Respostas iguais ou muito parecidas com a de outro aluno – ou com o conteúdo da internet, ou que sugira que foi gerado por IA – serão descartadas.
- Evite conversar durante a prova para não ser desclassificado.

Projeto: <https://github.com/lgsreal/2tdspc-spring-mvc/tree/aula-2802-2025>

Aula: <http://bit.ly/AulaThymeleaf>

Questões (2 pontos cada):

1. O que é o padrão MVC?
2. O que é o Thymeleaf e o que ele tem a ver como Spring?
3. O que é ModelAndView e como o utilizamos no contexto do Spring MVC?
4. O que são dialetos do Thymeleaf?
5. Quais são os 5 tipos de expressões padrão do Thymeleaf? Explique cada um.

Pergunta extra (1 ponto extra, mas a prova continua valendo 10):

Explique o código abaixo. Para que serve? Como funciona?

```
<script th:href="@{/js/script.js}" defer></script>
```

Respondendo as Questões:

1. O que é o padrão MVC?

R): O padrão MVC (Model-View-Controller) é um padrão que é utilizado no desenvolvimento de aplicações que separa a aplicação em três camadas principais sendo elas a camada “model”, “view” e a camada “controller” cada uma dessas camadas possui uma função específica:

Model: É a camada responsável pela lógica e manipulação de dados. Representa as entidades e também se comunica com o banco de dados.

View: A camada responsável pela interface gráfica com a qual o usuário interage. No Spring MVC, o Thymeleaf é utilizado para renderizar as páginas HTML.

Controller: Esta camada gerencia as requisições do usuário, processa as informações e define qual View será renderizada com quais dados do Model.

No Spring MVC, esse padrão é implementado através dos Controllers, que utilizam a annotation `@Controller` para gerenciar requisições e nos retornar dados ou páginas HTML

2. O que é o Thymeleaf e o que ele tem a ver como Spring?

R) O Thymeleaf é uma ferramenta que permite criar páginas HTML dinâmicas em aplicações Java. Ele é usado para preencher o HTML com informações vindas do backend, tornando as páginas mais interativas e adaptáveis aos dados da aplicação.

No Spring MVC, o Thymeleaf já vem integrado e facilita a exibição das informações na interface da aplicação. Com ele, é possível exibir textos, listas, links dinâmicos e até controlar a exibição de elementos na página de acordo com certas condições, sem precisar usar JavaScript (embora seja possível, como em qualquer projeto HTML). Além disso, também é possível aplicar estilos CSS diretamente nos templates.

Um dos pontos negativos do Thymeleaf é que ele não é a melhor opção para projetos muito grandes ou que exigem muitas interações em tempo real. Nesses casos seria melhor utilizarmos frameworks como React.

3. O que é ModelAndView e como o utilizamos no contexto do Spring MVC?

R) O ModelAndView é uma classe do Spring MVC que combina dados (Model) e a página HTML (View) que será exibida para o usuário. Ele é uma alternativa ao Model,

o que permite passarmos informações para a página junto com o nome da view que deve ser carregada.

Em projetos com Spring MVC, usamos o ModelAndView para enviar dados do backend para a página HTML, ajudando na organização do código. Ele é útil quando queremos definir tanto os dados quanto a página de destino em um único objeto.

Uma das vantagens do ModelAndView é que ele permite adicionar múltiplos atributos ao modelo e definir a view dinamicamente, o que pode ser útil em aplicações mais complexas.

4. O que são dialetos do Thymeleaf?

R) No Thymeleaf, os dialetos são “recursos” que adicionam novas funcionalidades e facilitam o desenvolvimento de páginas dinâmicas, permitindo incluir novos atributos e comportamentos dentro do HTML.

O Thymeleaf já possui um conjunto de dialetos nativos, que incluem comandos básicos como th:text, th:if e th:each, utilizados para exibir valores, aplicar condições e criar repetições dentro do HTML.

Além dos dialetos que já vêm no Thymeleaf, também é possível criar dialetos personalizados para atender necessidades específicas de um projeto, ampliando ainda mais as possibilidades da ferramenta.

5. Quais são os 5 tipos de expressões padrão do Thymeleaf? Explique cada um.

R): No Thymeleaf, existem cinco tipos principais de expressões que permitem manipular os dados dentro dos templates HTML de forma dinâmica.

Variable Expressions (\${...})

São utilizadas para acessar variáveis enviadas pelo backend e exibir seus valores dentro da página.

Selection Expressions (*{...})

Funcionam de maneira semelhante às expressões de variável, mas são usadas dentro de elementos que já possuem um objeto definido, facilitando a organização do código.

Message Expressions (#{...})

Permitem buscar mensagens armazenadas em arquivos de configuração da aplicação, sendo muito úteis para internacionalização e reutilização de textos dentro do sistema.

Link Expressions (@{...})

São utilizadas para gerar URLs dinâmicas dentro da aplicação, tornando os links flexíveis e evitando a necessidade de escrever caminhos manualmente.

Fragment Expressions (~{...})

Servem para reutilizar partes de um template dentro de outros arquivos, permitindo a criação de componentes reaproveitáveis, como cabeçalhos e rodapés.

6. Explique o código abaixo. Para que serve? Como funciona?

<script th:href="@{/js/script.js}" defer></script>

R): O código serve para incluir um arquivo JavaScript dentro da página HTML usando Thymeleaf.

O trecho `th:href="@{/js/script.js}"` faz com que o Thymeleaf crie automaticamente o caminho correto para o arquivo de script. Isso evita que o desenvolvedor tenha que escrever manualmente o endereço do arquivo, o que ajuda quando a aplicação está rodando em diferentes ambientes.

Já o atributo `defer` indica que o navegador só vai carregar e executar o JavaScript depois que todo o HTML da página estiver pronto. Isso melhora o desempenho e evita erros ao tentar acessar elementos que ainda não foram carregados.

Esse tipo de configuração é útil para garantir que a página funcione corretamente e para facilitar a organização dos arquivos dentro da aplicação.