

# Programação Web

## Módulo 5 - Introdução ao Spring MVC

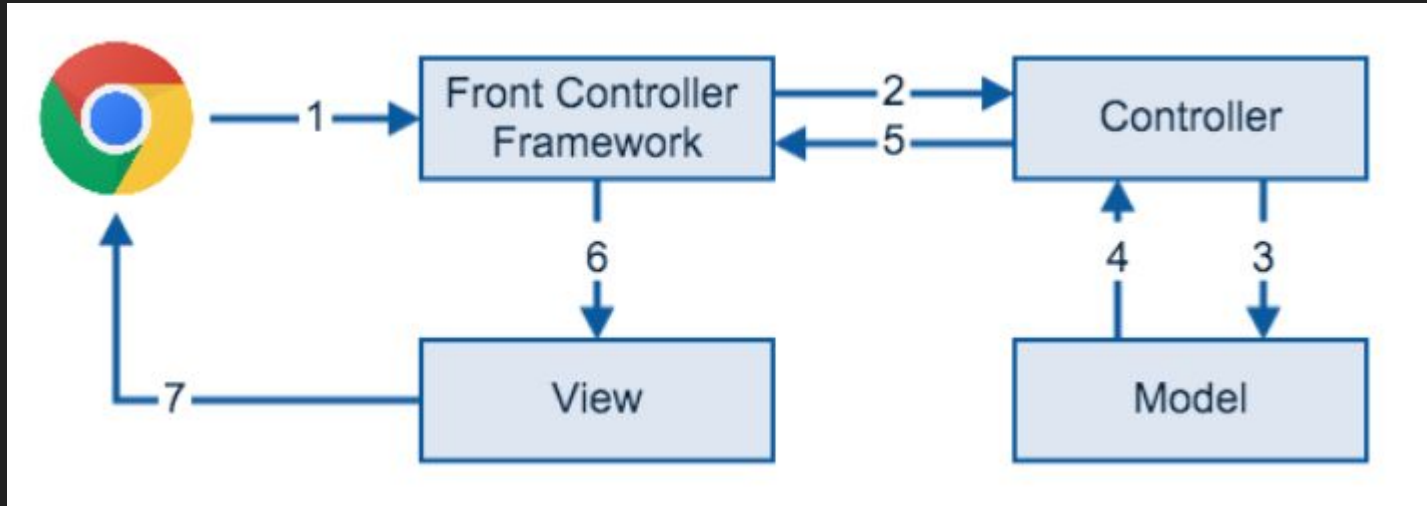
Daniel Lucrédio  
daniel.lucradio@ufscar.br

Este conteúdo foi baseado no conjunto de slides gentilmente cedido pelo prof. Delano Medeiros Beder  
(<https://github.com/delanobeder/DSW1>)

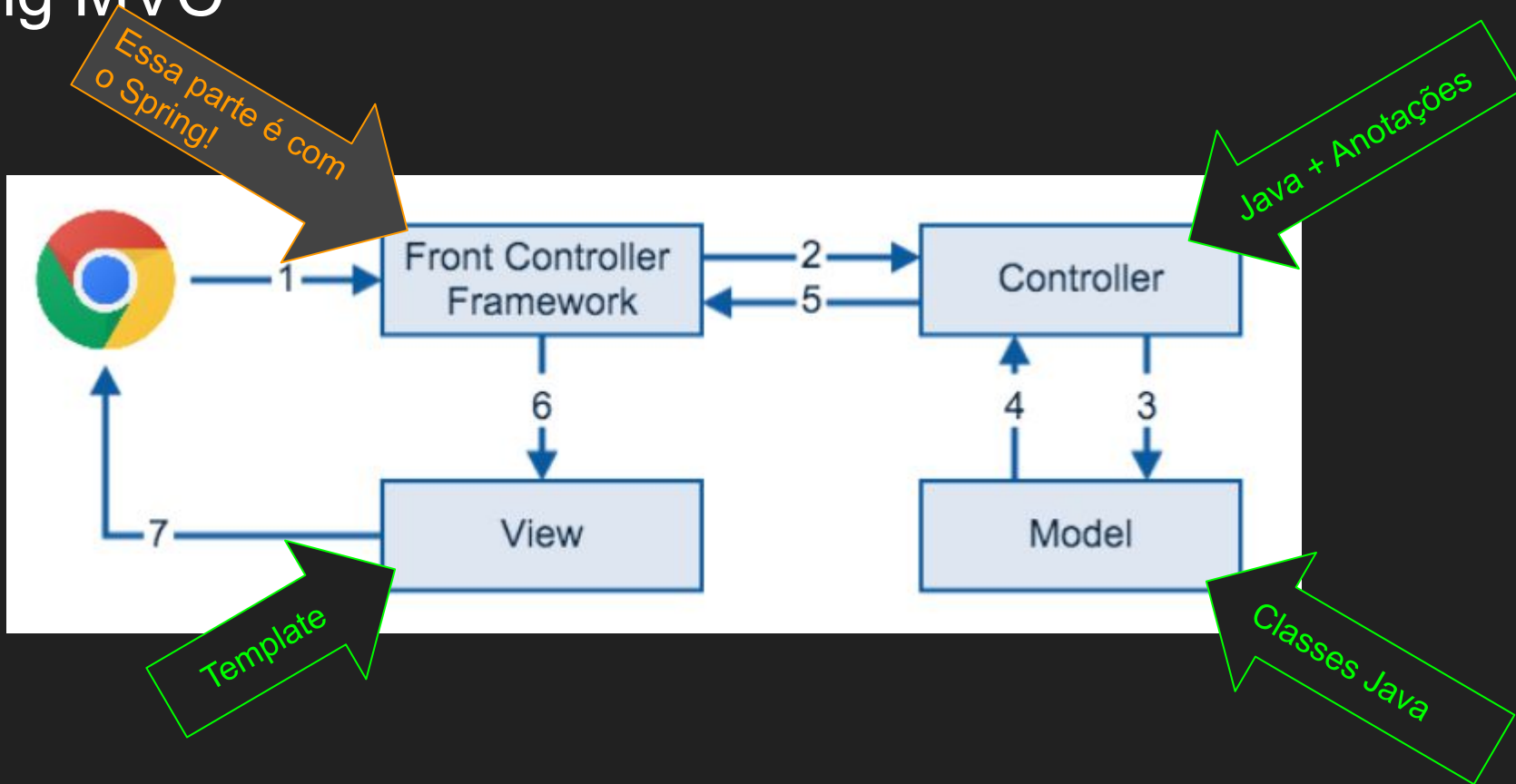
# Spring

- Nasceu em 2003 - inversão de Controle e Injeção de Dependência
  - Complexidade do padrão J2EE
- Atualmente tem inúmeros projetos
  - Spring Data
  - Spring Security
  - Spring Batch
  - ...
  - **Spring MVC**

# Spring MVC



# Spring MVC



# Anotações

@Controller

```
public class MeuControlador {
```

```
    @GetMapping("/")
```

```
    public String index() { ... }
```

```
    @PostMapping("/usuario")
```

```
    public String salvarUsuario(@Valid Usuario u) { ... }
```

```
}
```

# Template

- Camada de visão



```
-- HomeController.java
@RequestMapping("/")
public String home(Model model) {
```

```
    model.addAttribute("firstName","Fulano");
    model.addAttribute("lastName","Silva");
    return "home";
}
```

```
-- templates/home.html
```

```
<span th:text="${'Olá ' + firstName + ' ' + lastName}">>Oi</span>
```

http://localhost:8080/

Olá Fulano Silva

file:///templates/home.html

Oi

# “Alô mundo” com Spring MVC

Demonstração 18



# Internacionalização com ThymeLeaf

Demonstração 19

# Outras facilidades

## #dates

- #dates.format(date), #dates.day(date)

## #numbers

- #numbers.formatInteger(num,3), #numbers.sequence(from,to,step)

## #strings

- #strings.isEmpty(str), #strings.contains(str,"hi")

## #arrays

- #arrays.length(arr), #arrays.isEmpty(arr)

## #lists

- #lists.size(list), #lists.isEmpty(list)

## #sets

- #sets.contains(set,element), #sets.isEmpty(set)

# Condicionais

- Operador ternário **?:**
- Renderização condicional de conteúdo

```
<td th:text="{usuario.logado}?'SAIR':'ENTRAR'" />
```

# Condicionais

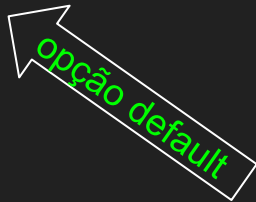
- **if** e **unless**
- Renderização condicional de todo elemento

```
<td>  
  <span th:if="{usuario.logado==true}">Sair</span>  
  <span th:unless="{usuario.logado==true}">Entrar</span>  
</td>
```

# Condicionais

- **switch** e **case**
- Teste de várias alternativas

```
<td th:switch="${#lists.size{aluno.disciplinas}}">  
  <span th:case="'0'">Sem disciplinas</span>  
  <span th:case="'1'" th:text="${aluno.disciplinas[0]}"></span>  
  <div th:case="*">Mais de uma disciplina</div>  
</td>
```



# Iterações

- **each**
- Percorre vários elementos de uma lista (java.util.Iterable)

```
<tr th:each="d: ${aluno.disciplinas}">  
  <td th:text="${d.codigo}" />  
  <td th:text="${d.nome}" />  
</tr>
```

# Iterações

- **each**
- Variável **status** ajuda a controlar a iteração
  - index: índice de iteração atual, começando com zero
  - count: número de elementos processados até agora
  - size: tamanho da lista
  - even/odd: se o índice atual é par ou ímpar
  - first: se estamos na primeira iteração
  - last: se estamos na última iteração

```
<tr th:each="d, stat: ${aluno.disciplinas}"  
    th:style="${stat.odd}?'font-weight:bold;'">  
    <td th:text="${d.codigo}" />  
    <td th:text="${d.nome}" />  
</td>
```

Fim