

Disciplina Regular 3

Desenvolvimento Web com Java EE

Graduação em Engenharia de Software - 2020

Etapa 3 Aula 2

Arquitetura MVC

Para acompanhar pelo Moodle você
deve estudar a etapa 4

Etapa 3 Aula 2 - Competências

- **Competências** Trabalhadas Nesta Etapa
 - **Construir aplicações Java Web usando Arquitetura MVC com Spring MVC.**
 - Compreender a importância dos Frameworks na construção de aplicações.
 - Conhecer os conceitos fundamentais do Spring Framework e Spring MVC.

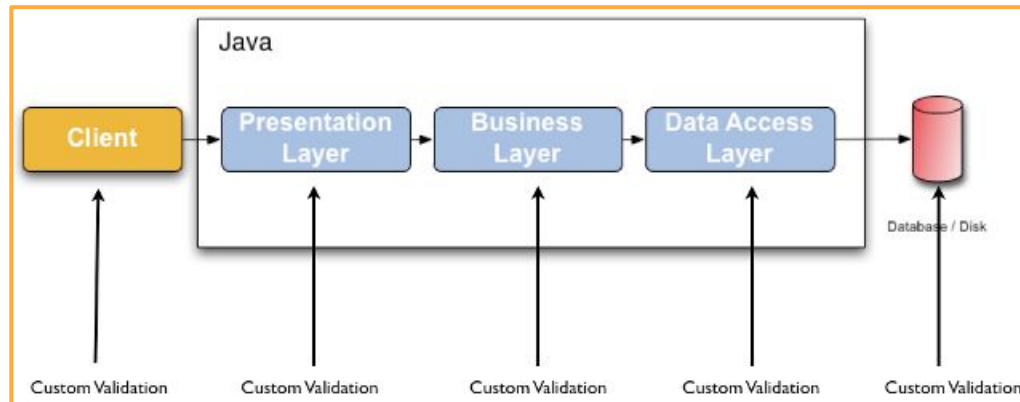
Etapa 3 Aula 2 - Checklist

- **Checklist DR3:**
 - Ter revisado os slides e **feito os exercícios** das aulas anteriores.
 - Ter iniciado o **TP1** que é para **04/05/2020**.
- **Checklist PB:**
 - Ter feito a reunião desta semana com o grupo para tratar do ASSESSMENT.
 - Ter clonado o repositório do projeto e subido a especificação do módulo (TP5).
 - Ter tratado na reunião sobre o **TP7** que é para entregar no dia **11/5/2020**.

Validação de Objetos

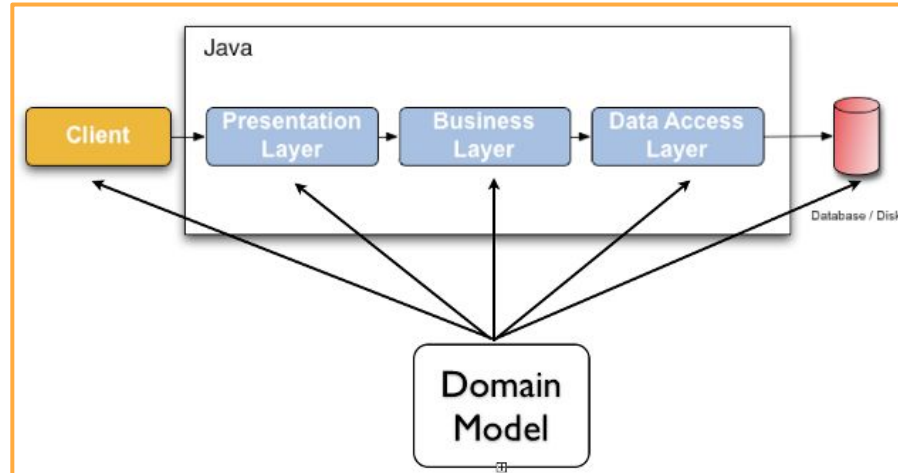
Fundamentos

- A validação de dados é uma tarefa comum que ocorre em todas as camadas do aplicativo, da apresentação à camada de persistência.
- Geralmente, a mesma lógica de validação é implementada em cada camada, que consome tempo e é propensa a erros.
- Para evitar a duplicação dessas validações, os desenvolvedores costumam agrupar a lógica de validação diretamente no **modelo de domínio**.



Fundamentos

- **Bean Validation** define um modelo de metadados e API para validação de entidade e método. A fonte de metadados padrão são anotações, com a capacidade de substituir e estender os metadados através do uso de XML.
- A API não está vinculada a uma camada de aplicativo específica nem a um modelo de programação.



Fundamentos

- O **Hibernate Validator** é a implementação de referência da especificação Bean Validation do Java EE.
- A implementação em si, bem como a API de validação de beans são fornecidos e distribuídos sob a Apache Software License 2.0.
- O Hibernate Validator 6 e Bean Validation 2.0 requerem Java 8 ou posterior.

```
<dependency>  
  <groupId>org.hibernate.validator</groupId>  
  <artifactId>hibernate-validator</artifactId>  
  <version>6.1.4.Final</version>  
</dependency>
```



```
1 package br.edu.infnet.fornecedores;
2
3 import javax.validation.constraints.NotEmpty;
4 import javax.validation.constraints.Size;
5
6 public class Fornecedor {
7
8     //-----
9     @NotEmpty(message = "O campo Nome de Fantasia é obrigatório")
10    @Size(min = 1, max = 20, message = "O campo Nome de Fantasia requer no máximo 20 caracteres")
11    private String nomeFantasia;
12    //-----
13    @NotEmpty(message = "O campo Razão Social é obrigatório")
14    @Size(min = 1, max = 150, message = "O campo Razão Social requer no máximo 150 caracteres")
15    private String razaoSocial;
16    //-----
17
18    public String getNomeFantasia() {
19        return nomeFantasia;
20    }
21
22    public void setNomeFantasia(String nomeFantasia) {
23        this.nomeFantasia = nomeFantasia;
24    }
25
```

FornecedorController.java x

Source History

```
1 package br.edu.infnet.fornecedores;
2
3 import ...8 lines
4
5
6
7
8
9
10
11
12 @Controller
13 public class FornecedorController {
14
15     @RequestMapping("/fornecedores/cadastrarFornecedor")
16     public ModelAndView cadastrarFornecedor(@Valid Fornecedor fornecedor, BindingResult br) {
17
18         ModelAndView retorno = new ModelAndView("fornecedores/manter");
19         if(br.hasErrors()) {
20
21             retorno.addObject("erros", br.getFieldErrors());
22
23         } else {
24
25             //-----
26             //Grava no banco de dados usando um DAO
27             //FornecedorDAO dao = new FornecedorDAO();
28             //dao.inserir(fornecedor);
29             //-----
30             retorno.addObject("sucesso", "Fornecedor " + fornecedor.getNomeFantasia()
31                             + " gravado com sucesso.");
32         }
33         return retorno;
34     }
35 }
```

```

manter.jsp x
Source History
25 <td>
26     Razão Social
27 </td>
28 <td>
29     <input type="text" name="razaoSocial" />
30 </td>
31 </tr>
32 <tr>
33 <td colspan="2" style="text-align: center">
34     <input type="submit" value="Cadastrar" />
35 </td>
36 </tr>
37 </table>
38 </form>
39 <div style="color: red">
40     <c:forEach var="erro" items="${erros}">
41         <h3>${erro.defaultMessage}</h3>
42     </c:forEach>
43 </div>
44 <div style="color: green">
45     <h3>${sucesso}</h3>
46 </div>
47 </body>
48 </html>

```

Sistema de Compras - Forn x +

localhost:8080/SistemaCompras/fornecedores/cadastrarFornecedor

Sistema de Compras - Fornecedores

[Voltar](#)

Nome Fantasia

Razão Social

O campo Nome de Fantasia é obrigatório

O campo Razão Social é obrigatório

O campo Razão Social requer no máximo 150 caracteres

O campo Nome de Fantasia requer no máximo 20 caracteres