

# - Caraduação





#### I Agenda



- Continuar com o projeto produto-mvc
- Camada de Serviço Categoria

#### Objetivos



- Continuar com o projeto produto-mvc
- Entender o funcionamento da camada de serviços.
- Delegar e implementar as reponsabilidades do controller Categoria para a classe de serviço.



#### **Exercício**

### **I Alterações**



- Vamos trabalhar com o perfil test.
- Baixar o projeto da aula 13.
- Incluir as dependências para Lombok.
- Alterar classes Categoria e Produto para usar Lombok.
- · Alterar navbar.

```
FIAP Home Categoria → Produto →
```

#### Responsabilidades



#### O Design Pattern Front Controller

#### Definição de MVC

- Model: Define o modelo ou domínio da aplicação
  - Regras de negócios
  - Persistência de dados
- View: Interação com usuário
- Controller: Componente intermediário entre View e Model
  - Recebe requisições (requests)
  - Envia respostas (responses)
  - o Interage com a camada model

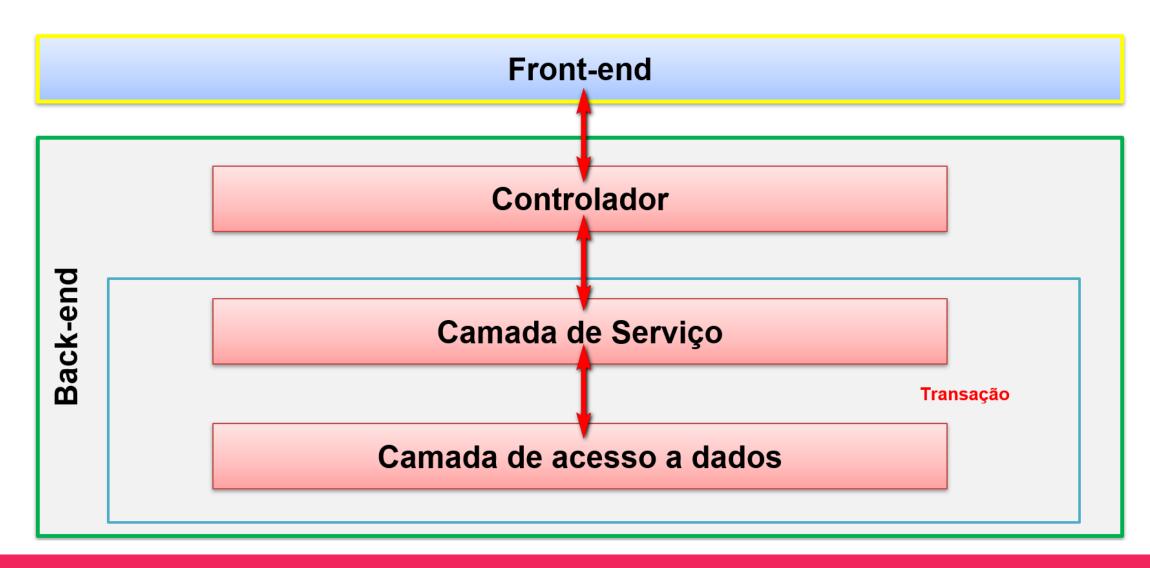
#### Responsabilidades

- Controller: responder interações do usuário
  - o No caso do MVC: recebe requisições (requests); enviar respostas (responses); Interage com a camada model.
- Service: realizar operações de negócio.
  - Um método da camada de serviços deve ter um significado relacionado ao negócio, podendo executar várias operações.
- Repository:
  - o Realizar operações "individuais" de acesso ao banco de dados.

#### **AULA 15 – Spring MVC - Camada Service - Categoria**

# Responsabilidades





**AULA 15 – Spring MVC - Camada Service - Categoria** 



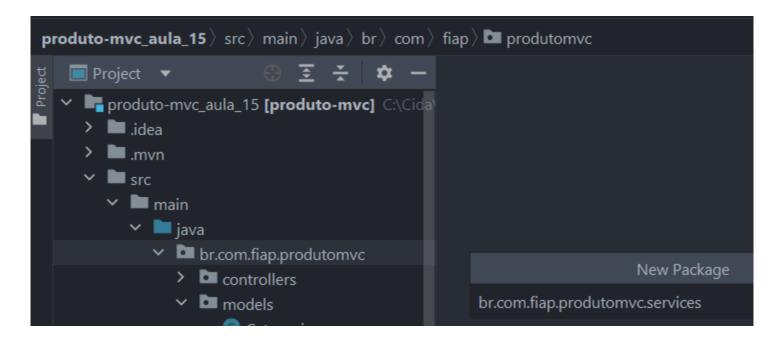
#### Camada de Serviços para Categoria

**Exercício** 

#### Camada de Serviços



Criar o package services



#### I Criar class CategoriaService



```
CategoriaService.java
     package br.com.fiap.produtomvc.services;
     import org.springframework.stereotype.Service;
     @Service
     public class CategoriaService {
```





```
😊 CategoriaService.java 🔀 😊 CategoriaController.java 🤇
       import java.util.List;
       @Service
       public class CategoriaService {
           @Autowired
           private CategoriaRepository repository;
           @Transactional(readOnly = true)
           public List<Categoria> findAll(){
                return repository.findAll();
```

#### I Alterar CategoriaController



```
CategoriaService.java ×
                © CategoriaController.java
          @Controller
          @RequestMapping("/categorias")
          public class CategoriaController {
              @Autowired
              private CategoriaService service;
              @Autowired
              private CategoriaRepository repository;
```





```
CategoriaService.java ×
                CategoriaController.java
             @GetMapping()
                                                                        Responsabilida do
                                                                        Service
             public String findAll(Model model) {
    ര
                  List<Categoria> categorias = service.findAll();
                 model.addAttribute("categorias", categorias);
                 return "/categoria/listar-categorias";
```





FIAP Home Categoria → Produto →							
Lista de Categorias							
	Nome	Ações					
	Smartphone	Excluir	Editar				
	Smart TV	Excluir	Editar				
	Notebook	Excluir	Editar				
	Tablet	Excluir	Editar				
	Mouse	Excluir	Editar				
	Teclado	Excluir	Editar				





```
CategoriaService.java ×

CategoriaController.java ×

//Método para inserir Categoria

A return repository.save(categoria);

Categoria

Categori
```

### I Alterar insert de CategoriaController



```
🕝 CategoriaService.java 🛚 🖰
                CategoriaController.java
              @PostMapping()
              public String insert(@Valid Categoria categoria,
     @
                                    BindingResult result,
                                    RedirectAttributes attributes) {
                  if (result.hasErrors()) {
                      return "categoria/nova-categoria";
                  categoria = service.insert(categoria);
                  attributes.addFlashAttribute("mensagem", "Categoria salva com sucesso!");
                  return "redirect:/categorias";
```

## I Testar aplicação



Home Categoria ▼ Produto ▼			FIAP Home Categoria	Produto ▼
Cadastro de Categoria		o requerido	Cadastro de Categoria	
Nome:		ne deve ter no mínimo 3 carateres	Nome: Tes	te
Salvar				
	FIAD. Harra Catagoria a Bradu	A		
	FIAP Home Categoria Produ			
	Nome	Ações		
	Smartphone	Excluir	Editar	
	Smart TV	Excluir	Editar	
	Notebook	Excluir	Editar	
	Tablet	Excluir	Editar	
	Mouse	Excluir	Editar	
	Teclado	Excluir	Editar	

**AULA 15 – Spring MVC - Camada Service - Categoria** 

# Implementar findById class CategoriaService [-|^\]

```
CategoriaService.java ×
                CategoriaController.java >
           @Transactional(readOnly = true)
           public Categoria findById(Long id) {
                Categoria categoria = repository.findById(id).orElseThrow(
                           -> new IllegalArgumentException("Recurso inválido - " + id)
               return categoria;
```

#### I Alterar findByld de CategoriaController



```
CategoriaService.java
                 CategoriaController.java
              @GetMapping("/{id}")
              public String findById(@PathVariable("id") Long id, Model model) {
     @
                  Categoria categoria = service.findById(id);
                  model.addAttribute("categoria", categoria);
                  return "/categoria/editar-categoria";
```





```
CategoriaController.java
      CategoriaService.java
                 @Transactional
                 public Categoria update(Long id, Categoria entity) {
                     try {
                         Categoria categoria = repository.getReferenceById(id);
                         copyToCategoria(entity, categoria);
                         categoria = repository.save(categoria);
                         return categoria;
                      catch (EntityNotFoundException e) {
                         throw new IllegalArgumentException("Recurso não encontrado");
                private void copyToCategoria(Categoria entity, Categoria categoria) {
     55 @
                     categoria.setNome(entity.getNome());
AULA 57
```

#### I Alterar update de CategoriaController

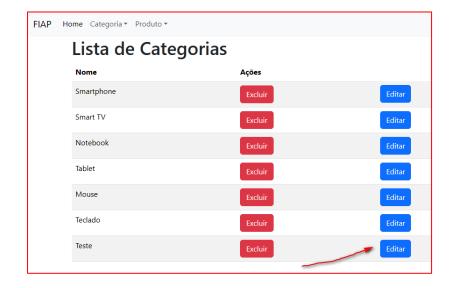


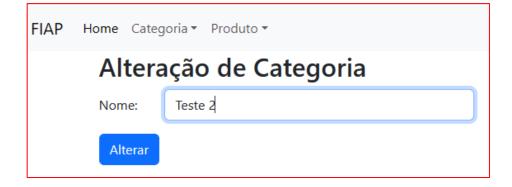
```
CategoriaService.java × ContegoriaController.java
              @PutMapping("/{id}")
     @
              public String update(@PathVariable("id") Long id,
                                    @Valid Categoria categoria,
                                     BindingResult result) {
                  if (result.hasErrors()) {
                      categoria.setId(id);
                      return "/categoria/editar-categoria";
                  service.update(id, categoria);
                  return "redirect:/categorias";
```

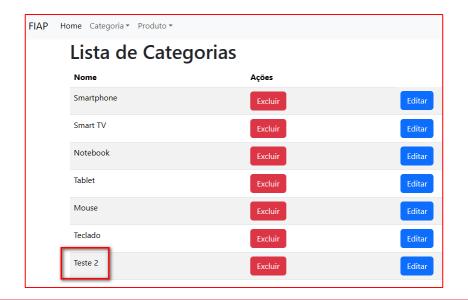
**AULA 15 – Spring MVC - Camada Service - Categoria** 

## **I Testar Aplicação**













```
😊 CategoriaService.java 🔀
               CategoriaController.java
          @Transactional
           public void delete(Long id) {
               if (!repository.existsById(id)) {
                   throw new IllegalArgumentException("Recurso inválido - id: " + id);
               trv {
                   repository.deleteById(id);
               } catch (DataIntegrityViolationException e) {
                   throw new IllegalArgumentException("Falha de integridade referencial - id: " + id);
```

### I Alterar delete de CategoriaController



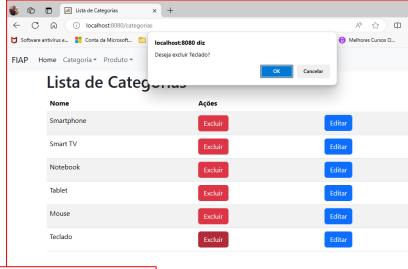
```
CategoriaService.java >
                 CategoriaController.java
              @DeleteMapping("/{id}")
               public String delete(@PathVariable("id") Long id, Model model) {
                   service.delete(id);
                   return "redirect:/categorias";
```

**AULA 15 – Spring MVC - Camada Service - Categoria** 

# **I** Testar Aplicação









### I Finalizando class CategoriaController



- Organizar os imports.
- Não estamos utilizando atributo repository então podemos excluir ou deixar comentado.

#### Validação



 Testar todas as funcionalidades da aplicação, inclusive de produto.

#### **I** Dúvidas







#### Copyright © 2024 Prof<sup>a</sup>. Aparecida de Fátima Castello Rosa

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).