

CheckPoint 05 Java Advanced

Projeto Receitas

Recriando o projeto do meu checkpoint 03 de python do primeiro ano agora utilizando o thymeleaf e outras tecnologias



Minha Ideia

A ideia para este checkpoint foi reviver o primeiro projeto completo que desenvolvi na matéria de Python, durante o segundo semestre da FIAP. Na época, junto de duas amigas, criamos uma aplicação com front-end, back-end e banco de dados integrados, que consumia dados de uma API de receitas e realizava a tradução para o português

Checkpoint 05 de Python


Desenvolvemos um projeto completo com front-end e back-end integrados, consumindo dados de uma API externa de receitas com tradução automática

2024

2025

Checkpoint 05 de Java

Criei uma aplicação em Spring MVC com páginas dinâmicas, CRUD completo, validações, duas entidades e navegação com Thymeleaf



Transformando a ideia em ação

Voltei no projeto original de Python para entender como ele funcionava. A partir disso, consegui identificar as classes, funcionalidades e estrutura que precisaria adaptar ou reconstruir para essa nova versão

Criando o novo projeto

Decidimos começar o novo projeto em Java seguindo a estrutura que o professor apresentou em sala, com Spring MVC

Versão do Professor

```
1  2tdspc-spring-mvc-master/
2  └─ java/
3     └─ br/
4        └─ com/
5           └─ fiap/
6              └─ spring_mvc/
7                 SpringMvcApplication.java
8                 configuration/
9                    InternacionalizacaoConfiguration.java
10                controller/
11                   LivroController.java
12                dto/
13                   LivroRequest.java
14                model/
15                   Categoria.java
16                   Livro.java
17                repository/
18                   LivroRepository.java
19                security/
20                   SecurityConfig.java
21                service/
22                   LivroService.java
23
24  resources/
25  └─ application.properties
26  └─ internationalization/
27     └─ messages.properties
28     └─ messages_en.properties
29  └─ static/
30     └─ css/
31        style.css
32     └─ js/
33        script.js
34  └─ templates/
35     livroCadastro.html
36     livroEdicao.html
37     livroLista.html
38     fragments/
39        footer.html
40        header.html
```

Minha Versão

```
1  catalogo-receitas/
2  └─ java/
3     └─ br/
4        └─ com/
5           └─ fiap/
6              └─ springmvc/
7                 SpringMvcApplication.java
8                 controller/
9                    IngredienteController.java
10                   ReceitaController.java
11                model/
12                   Dificuldade.java
13                   Ingrediente.java
14                   Receita.java
15                   SpringMvcApplication.java
16                repository/
17                   IngredienteRepository.java
18                   ReceitaRepository.java
19
20  resources/
21  └─ application.properties
22  └─ internationalization/
23     └─ messages.properties
24     └─ messages_en.properties
25  └─ static/
26     └─ css/
27        style.css
28     └─ img/
29        grandma.png
30  └─ templates/
31     index.html
32     fragments/
33        footer.html
34        header.html
35     ingrediente/
36        form.html
37        list.html
38     receita/
39        form.html
40        list.html
```

Reaproveitamento

A estrutura de pastas ensinada pelo professor serviu como base para o projeto. Também mantive o uso do Thymeleaf para criação das páginas dinâmicas, além da organização com controllers, services e repositories.

Adaptações

Mesmo com a estrutura base, precisei adaptar algumas partes para encaixar melhor a proposta do meu projeto, como a criação de novas entidades, ajustes nas rotas e personalização das páginas para refletirem a identidade da aplicação.



Funcionalidades implementadas

No projeto atual, o usuário pode cadastrar receitas, adicionar ingredientes e vinculá-los às receitas. Além disso, o site está preparado para exibir o conteúdo em português ou inglês, de acordo com a linguagem definida no navegador do usuário.

```

1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  <head th:replace="~{fragments/header :: header}"></head>
4  <body>
5
6  <th:block th:replace="fragments/header :: header"></th:block>
7
8  <h2 th:text="#{titulo.receitas}">Receitas</h2>
9
10 <a th:href="@{/receitas/nova}" th:text="#{botao.novaReceita}">Nova Receita</a>
11
12 <table>
13   <thead>
14     <tr>
15       <th th:text="#{label.nome}">Nome</th>
16       <th th:text="#{label.descricao}">Descrição</th>
17       <th th:text="#{label.tempoPreparo}">Tempo de Preparo</th>
18       <th th:text="#{label.dificuldade}">Dificuldade</th>
19       <th th:text="#{label.acoes}">Acoes</th>
20     </tr>
21   </thead>
22   <tbody>
23     <tr th:each="receita : ${receitas}">
24       <td th:text="${receita.nome}">Nome</td>
25       <td th:text="${receita.descricao}">Descrição</td>
26       <td th:text="${receita.tempoPreparo}">Tempo</td>
27       <td th:text="${receita.dificuldade}">Dificuldade</td>
28       <td>
29         <a th:href="@{'/receitas/editar/' + ${receita.id}}" th:text="#{botao.editar}">Editar</a> |
30         <a th:href="@{'/receitas/excluir/' + ${receita.id}}" th:text="#{botao.excluir}">Excluir</a>
31       </td>
32     </tr>
33   </tbody>
34 </table>
35
36 <th:block th:replace="fragments/footer :: footer"></th:block>
37
38 </body>
39 </html>
40

```

Listagem de Receitas

Catalogo de Receitas

Receitas | Ingredientes

Receitas

Nova Receita

Nome	Descrição	Tempo de Preparo	Dificuldade	Acoes
Receita 1	lorem ipsun it dolor amet	1	FACIL	Editar Excluir
Receita 2	teste teste teste teste	4	DIFICIL	Editar Excluir
Receita 3	teste1 teste1 teste1 teste1	2	MEDIA	Editar Excluir

Todos os direitos reservados 2025
Idioma adaptado automaticamente de acordo com as configuracoes do seu navegador.

Adição de Receitas

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org"
3       xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity6"
4       xmlns:form="http://www.thymeleaf.org/thymeleaf-spring">
5 <head th:replace="~{fragments/header :: header}"></head>
6 <body>
7
8 <th:block th:replace="~{fragments/header :: header}"></th:block>
9
10 <form th:action="@{/receitas/salvar}" th:object="${receita}" method="post">
11
12     <h2 th:text="${#fields.hasErrors()} ? 'Corrija os erros abaixo' : #{botao.novaReceita}}"></h2>
13     <input type="hidden" th:field="*{id}"/>
14
15     <div th:classappend="${#fields.hasErrors('nome')} ? 'has-error'">
16         <label th:text="#{label.nome}"></label>
17         <input type="text" th:field="*{nome}" />
18         <span th:if="${#fields.hasErrors('nome')}" th:errors="*{nome}"></span>
19     </div>
20
21     <div th:classappend="${#fields.hasErrors('descricao')} ? 'has-error'">
22         <label th:text="#{label.descricao}"></label>
23         <textarea th:field="*{descricao}"></textarea>
24         <span th:if="${#fields.hasErrors('descricao')}" th:errors="*{descricao}"></span>
25     </div>
26
27     <div th:classappend="${#fields.hasErrors('tempoPreparo')} ? 'has-error'">
28         <label th:text="#{label.tempoPreparo}"></label>
29         <input type="number" th:field="*{tempoPreparo}" />
30         <span th:if="${#fields.hasErrors('tempoPreparo')}" th:errors="*{tempoPreparo}"></span>
31     </div>
32
33     <div th:classappend="${#fields.hasErrors('dificuldade')} ? 'has-error'">
34         <label th:text="#{label.dificuldade}"></label>
35         <select th:field="*{dificuldade}">
36             <option value="">-- Selecione --</option>
37             <option th:each="dif : ${T(br.com.fiap.springmvc.model.Dificuldade).values()}"
38                   th:value="${dif}"
39                   th:text="#{__${dif.labelKey}_}"></option>
40         </select>
41         <span th:if="${#fields.hasErrors('dificuldade')}" th:errors="*{dificuldade}"></span>
42     </div>
```

Catalogo de Receitas

[Receitas](#) | [Ingredientes](#)

Nova Receita

Nome

Descricao

Tempo de Preparo

Dificuldade

-- Selecione --

[Salvar](#) [Cancelar](#)

Todos os direitos reservados 2025

Idioma adaptado automaticamente de acordo com as configurações do seu navegador.


```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="~{fragments/header :: head}"></head>
4 <body>
5
6 <th:block th:replace="fragments/header :: header"></th:block>
7
8 <h2 th:text="#{titulo.ingredientes}"></h2>
9
10 <a th:href="@{/ingredientes/novo}" th:text="#{botao.novoIngrediente}"></a>
11
12 <table>
13   <thead>
14     <tr>
15       <th th:text="#{label.nome}">Nome</th>
16       <th th:text="#{label.quantidade}">Quantidade</th>
17       <th th:text="#{label.receita}">Receita</th>
18       <th th:text="#{label.acoes}">Acoes</th>
19     </tr>
20   </thead>
21   <tbody>
22     <tr th:each="ingrediente : ${ingredientes}">
23       <td th:text="${ingrediente.nome}">Nome</td>
24       <td th:text="${ingrediente.quantidade}">Quantidade</td>
25       <td th:text="${ingrediente.receita.nome}">Receita</td>
26       <td>
27         <a th:href="@{'/ingredientes/editar/' + ${ingrediente.id}}" th:text="#{botao.editar}">Editar</a> |
28         <a th:href="@{'/ingredientes/excluir/' + ${ingrediente.id}}" th:text="#{botao.excluir}">Excluir</a>
29       </td>
30     </tr>
31   </tbody>
32 </table>
33
34 <th:block th:replace="fragments/footer :: footer"></th:block>
35
36 </body>
37 </html>
38

```

Listagem de Ingredientes

Catalogo de Receitas

Receitas | Ingredientes

Ingredientes

Novo ingrediente

Nome	Quantidade	Receita	Acoes
Sal	1	Frango Assado	Editar Excluir

Todos os direitos reservados 2025
Idioma adaptado automaticamente de acordo com as configurações de seu navegador.

Adição de Ingredientes

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head th:replace=~{fragments/header :: head}></head>
4 <body>
5
6 <th:block th:replace="fragments/header :: header"></th:block>
7
8 <form th:action="@{/ingredientes/salvar}" th:object="${ingrediente}" method="post">
9
10 <h2 th:text="${#fields.hasErrors()} ? 'Corrija os erros abaixo' : #[botao.novoIngrediente]></h2>
11
12 <input type="hidden" th:field="**{id}"/>
13
14 <div>
15 <label th:text="#{label.nome}></label>
16 <input type="text" th:field="**{nome}" />
17 <span th:if="${#fields.hasErrors('nome')}" th:errors="**{nome}></span>
18 </div>
19
20 <div>
21 <label th:text="#{label.quantidade}></label>
22 <input type="text" th:field="**{quantidade}" />
23 <span th:if="${#fields.hasErrors('quantidade')}" th:errors="**{quantidade}></span>
24 </div>
25
26 <div>
27 <label th:text="#{label.receita}></label>
28 <select th:field="**{receita}>
29 <option value="" th:text="#{label.selecao}>-- Seleccione --</option>
30 <option th:each="receita : ${receitas}" th:value="${receita.id}" th:text="${receita.nome}></option>
31 </select>
32 <span th:if="${#fields.hasErrors('receita')}" th:errors="**{receita}></span>
33 </div>
34
35 <div>
36 <button type="submit" th:text="#{botao.salvar}></button>
37 <a th:href="@{/ingredientes}" th:text="#{botao.cancelar}></a>
38 </div>
39
40 </form>
41
42 <th:block th:replace="fragments/footer :: footer"></th:block>
43
44 </body>
45 </html>
46
```

Catalogo de Receitas

[Receitas](#) | [Ingredientes](#)

Novo Ingrediente

Nome

Quantidade

Receita

→ Seleccione →

Salvar

Cancelar

Todos os direitos reservados 2025

Idioma adaptado automaticamente de acordo com as configurações do seu navegador.

Referências do projeto de Python

Para esses dois projetos pensamos em trazer elementos visuais e afetivos do projeto original. A estética fofa, inspirada em uma "vibe de vó", continua presente, assim como a mascote que criamos na primeira versão. Esses detalhes ajudam a manter a identidade do projeto e trazem um sentimento de nostalgia





Agora que apresentei a ideia e o funcionamento do projeto, vamos ver o código na prática no IntelliJ. Acompanhem a implementação