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



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

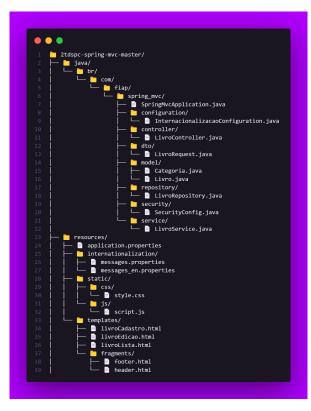
## Transformando a ideia em ação

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

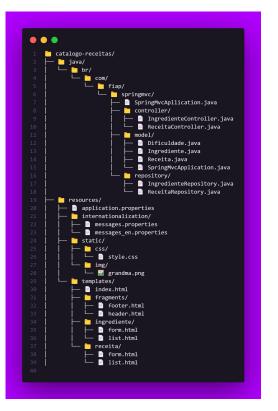
#### Versão do Professor

## Criando o novo projeto

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



#### Minha Versão



#### 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.

```
<!DOCTYPE html>
  <html xmLns:th="http://www.thymeleaf.org">
  <head th:replace="~{fragments/header :: head}"></head>
  <th:block th:replace="fragments/header :: header"></th:block>
  <h2 th:text="#{titulo.receitas}">Receitas</h2>
  <a th:href="@{/receitas/nova}" th:text="#{botao.novaReceita}">Nova Receita</a>
     <thead>
       Nome
       Descrição
       Tempo de Preparo
       Dificuldade
       Acoes
     </thead>
     Nome
       Descricao
       Tempo
       Dificuldade
         <a th:href="@{'/receitas/editar/' + ${receita.id}}" th:text="#{botao.editar}">Editar</a> |
          <a th:href="@{'/receitas/excluir/' + ${receita.id}}" th:text="#{botao.excluir}">Excluir</a>
     <th:block th:replace="fragments/footer :: footer"></th:block>
38 </body>
```

### Listagem de Receitas

	redientes			
Receitas				
Iova Receita				
Nome	Descricao	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	2	MEDIA	Editar   Excluir
110001100				

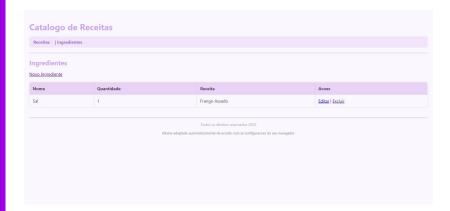
```
<!DOCTYPE htmL>
    <html xmLns:th="http://www.thymeleaf.org"</pre>
          xmLns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity6"
          xmlns:form="http://www.thymeleaf.org/thymeleaf-spring">
    <head th:replace="~{fragments/header :: head}"></head>
    <th:block th:replace="~{fragments/header :: header}"></th:block>
    <form th:action="@{/receitas/salvar}" th:object="${receita}" method="post">
        <h2 th:text="${#fields.hasErrors()} ? 'Corrija os erros abaixo' : #{botao.novaReceita}"></h2>
        <input type="hidden" th:field="*{id}"/>
        <div th:cLassappend="${#fields.hasErrors('nome')} ? 'has-error'">
            <label th:text="#{label.nome}"></label>
            <input type="text" th:field="*{nome}" />
            <span th:if="${#fields.hasErrors('nome')}" th:errors="*{nome}"></span>
        <div th:classappend="${#fields.hasErrors('descricao')} ? 'has-error'">
            <label th:text="#{label.descricao}"></label>
            <textarea th:field="*{descricao}"></textarea>
            <span th:if="${#fields.hasErrors('descricao')}" th:errors="*{descricao}"></span>
        <div th:classappend="${#fields.hasErrors('tempoPreparo')} ? 'has-error'">
            <label th:text="#{label.tempoPreparo}"></label>
            <input type="number" th:field="*{tempoPreparo}" />
            <span th:if="${#fields.hasErrors('tempoPreparo')}" th:errors="*{tempoPreparo}"></span>
        <div th:classappend="${#fields.hasErrors('dificuldade')} ? 'has-error'">
            <label th:text="#{label.dificuldade}"></label>
            <select th:field="*{dificuldade}">
                <option value="">-- Selectione --</option>
                <option th:each="dif: ${T(br.com.fiap.springmvc.model.Dificuldade).values()}"</pre>
                        th:value="${dif}"
                        th:text="#{ ${dif.labelKey} }"></option>
            <span th:if="${#fields.hasErrors('dificuldade')}" th:errors="*{dificuldade}"></span>
```

### Adição de Receitas

Receitas   Ingredientes	
Nova Receita	
Nome	
Descricao	
Tempo de Preparo	
Dificuldade	
- Selecione -	
Salvar <u>Cancelar</u>	

```
<!DOCTYPE htmL>
  <html xmLns:th="http://www.thymeleaf.org">
  <head th:replace="~{fragments/header :: head}"></head>
  <th:block th:replace="fragments/header :: header"></th:block>
  <h2 th:text="#{titulo.ingredientes}"></h2>
  <a th:href="@{/ingredientes/novo}" th:text="#{botao.novoIngrediente}"></a>
     <thead>
       Nome
       Quantidade
       Receita
       Acoes
     Nome
       Quantidade
       Receita
          <a th:href="@{'/ingredientes/editar/' + ${ingrediente.id}}" th:text="#{botao.editar}">Editar</a> |
          <a th:href="@{'/ingredientes/excluir/' + ${ingrediente.id}}" th:text="#{botao.excluir}">Excluir</a>
     <th:block th:replace="fragments/footer :: footer"></th:block>
  </body>
```

#### Listagem de Ingredientes



```
<!DOCTYPE htmL>
   <html xmlns:th="http://www.thymeleaf.org">
    <head th:replace="~{fragments/header :: head}"></head>
   <th:block th:replace="fragments/header :: header"></th:block>
8 <form th:action="@{/ingredientes/salvar}" th:object="${ingrediente}" method="post">
        <h2 th:text="${#fields.hasErrors()} ? 'Corrija os erros abaixo' : #{botao.novoIngrediente}"></h2>
        <input type="hidden" th:field="*{id}"/>
           <label th:text="#{label.nome}"></label>
           <input type="text" th:field="*{nome}" />
           <span th:if="${#fields.hasErrors('nome')}" th:errors="*{nome}"></span>
           <label th:text="#{label.quantidade}"></label>
           <input type="text" th:field="*{quantidade}" />
           <span th:if="${#fields.hasErrors('quantidade')}" th:errors="*{quantidade}"></span>
           <label th:text="#{label.receita}"></label>
           <select th:field="*{receita}">
               <option value="" th:text="#{label.selecione}">-- Selecione --</option>
               <option th:each="receita : ${receitas}" th:value="${receita.id}" th:text="${receita.nome}"></option>
            <span th:if="${#fields.hasErrors('receita')}" th:errors="*{receita}"></span>
           <button type="submit" th:text="#{botao.salvar}"></button>
           <a th:href="@{/ingredientes}" th:text="#{botao.cancelar}"></a>
42 <th:block th:replace="fragments/footer :: footer"></th:block>
```

#### Adição de Ingredientes

eceitas   Ingredientes	
Novo Ingrediente	
Nome	
Quantidade	
Receita	
Selecione	
Salvar Cancelar	
Soft Section.	

## 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



