

CETEJ35 - Java Web - JAVA_XXX (2024_01)

[Meus cursos](#) / [CETEJ35 - Web \(2024_01\)](#) / [Semana 01: 02/09 a 08/09](#) / [Gerenciamento de Cidades](#)

Gerenciamento de Cidades

✓ Feito: Ver

✓ Feito: Gastar pelo menos 20 minutos na atividade

✓ Feito: Passar pela atividade até o fim

Fecha: segunda-feira, 2 dez. 2024, 00:00

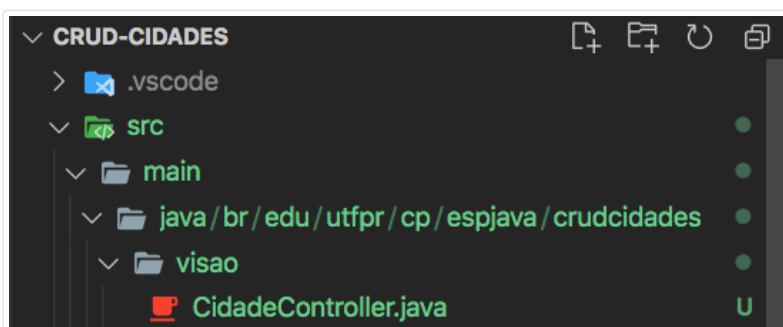
Nessa primeira aula prática, daremos início a um projeto do tipo CRUD. Um CRUD é um aplicativo que permite criar (*create*), ler (*read*), alterar (*update*) e excluir (*delete*) dados. Como referência, vamos usar os dados de cidades. Uma cidade está associada a um único estado. Portanto, para criar uma cidade precisamos informar o nome da cidade e o estado onde ela se encontra.

CRIANDO O ACESSO À PÁGINA

Agora que nossa página estática está pronta, precisamos criar uma classe Java que será responsável por receber as solicitações do navegador e enviar a página para o navegador. Para isso, vamos criar um pacote que vamos chamar de **visao**. No VS Code, basta criar uma pasta dentro da pasta **crudcidades**. Em seguida, crie a classe **CidadeController.java**, dentro da pasta **visao**. Se estiver usando outra IDE/editor, siga o procedimento adequado na sua IDE/editor para criar a classe o pacote.



Um pacote em Java é representado na estrutura de diretórios como uma pasta.



Na classe **CidadeController**, crie um método que retorne o nome da página estática como uma **String**, conforme a Figura abaixo.

```
6      public String listar() {  
7          return "crud.html";  
8      }
```

Por fim, adicione as anotações **org.springframework.stereotype.Controller** imediatamente antes da declaração da classe, e **org.springframework.web.bind.annotation.GetMapping** imediatamente antes da declaração do método.

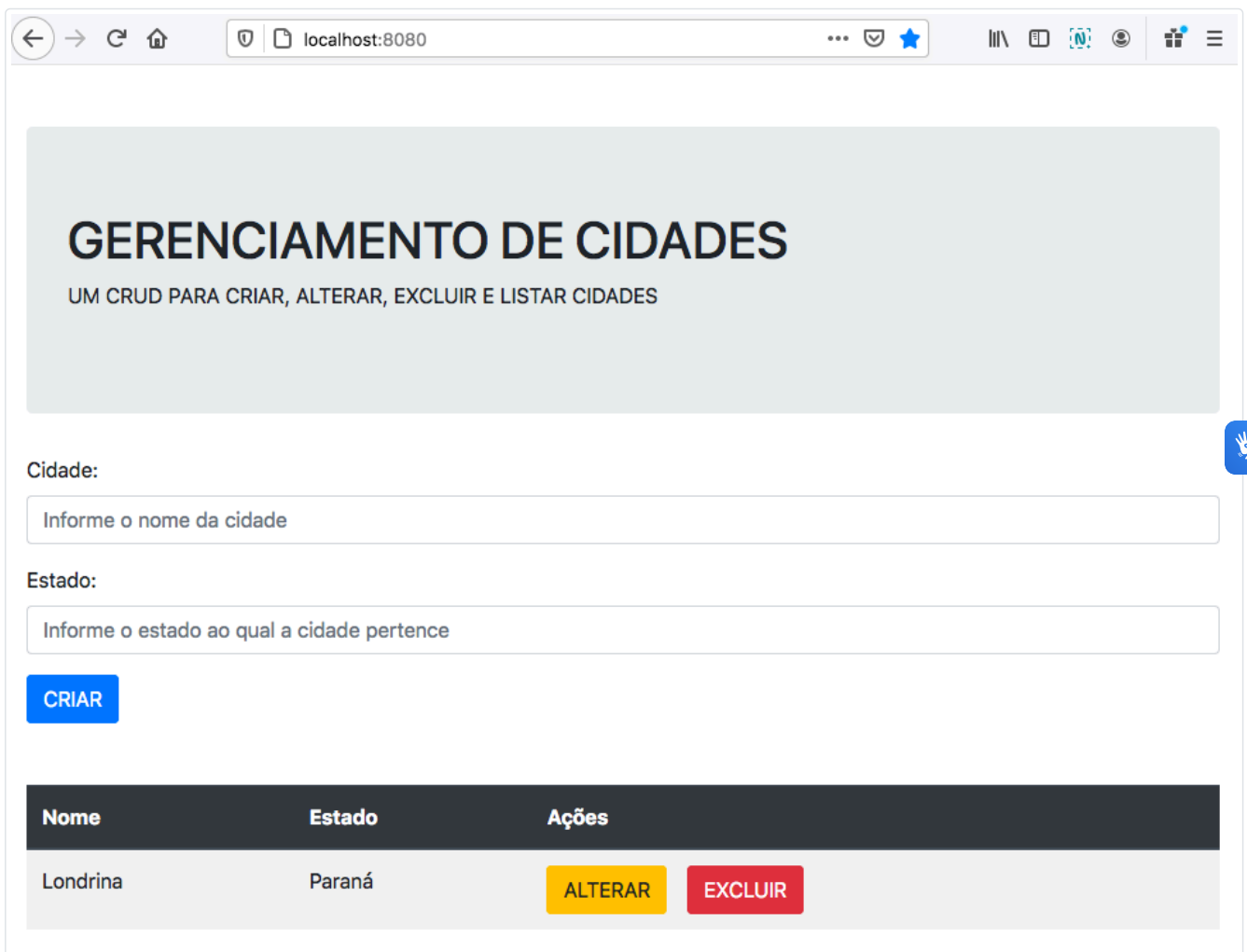
O VS Code também completa o código pra você, assim como as IDEs. Basta começar a escrever e ele já faz as sugestões. O VS Code também importa as classes usadas pelas bibliotecas referenciadas no **pom.xml**.

```

6  @Controller
7  public class CidadeController {
8
9      @GetMapping("/")
10     public String listar() {
11         return "crud.html";
12     }

```

A anotação na linha 6 indica pro Spring Boot que essa classe é uma classe que recebe requisições vindas do navegador. Já a anotação na linha 9 indica pro Spring Boot que qualquer solicitação feita para o endereço indicado, nesse caso, a raiz da URL, deve ser entregue para o método `listar()`. Quando o método retorna o nome da página estática, o Spring Boot entende que ele deve carregar essa página. Execute sua aplicação de dentro da sua IDE/editor ou usando `mvn spring-boot:run` em linha de comando. O resultado deve ser parecido com o da Figura a seguir.



← → ↺ 🏠 localhost:8080

GERENCIAMENTO DE CIDADES

UM CRUD PARA CRIAR, ALTERAR, EXCLUIR E LISTAR CIDADES

Cidade:

Estado:

CRIAR

| Nome | Estado | Ações |
|----------|--------|-------------------------------|
| Londrina | Paraná | ALTERAR EXCLUIR |

O código desse projeto está disponível no [Github](#), na branch `semana01-03-projeto-serve-estatica`.



Claro, essa página ainda não faz nada. Contudo, esses são os princípios fundamentais para criar uma aplicação Web com Java e Spring Boot. Agora, você pode comparar o código que você desenvolveu nessa aula com o código do **Petclinic**, apresentado na aula anterior. Na nossa próxima aula, vamos evoluir esse código adicionando dinamismo na página e criando um CRUD completo.

Retroceder

Avançar

Seguir para...

[Verificação de aprendizado - Gerenciamento de Cidades ▶](#)



 [Contate o suporte do site](#) 

Você acessou como RAFAEL ROCHA DA SILVA PROENCA (Sair)
CETEJ35 - Web (2024_01)

[Tema](#)
[Adaptable](#)
[Boost](#)
[Clássico](#)
[Campus](#)
[Apucarana](#)
[Campo Mourão](#)
[Cornélio Procópio](#)
[Curitiba](#)
[Dois Vizinhos](#)
[Francisco Beltrão](#)
[Guarapuava](#)
[Londrina](#)
[Medianeira](#)
[Pato Branco](#)
[Ponta Grossa](#)
[Reitoria](#)
[Santa Helena](#)
[Toledo](#)
[UTFPR](#)
[Ajuda](#)
[Chat UTFPR](#)
[Calendário Acadêmico](#)
[Biblioteca](#)
[e-Mail](#)
[Nuvem \(OwnCloud \)](#)
[Produção Acadêmica](#)
[Secretaria Acadêmica](#)
[Sistemas Corporativos](#)
[Sistema Eletrônico de Informação - SEI](#)
[Suporte ao usuário](#)
[Criação de curso](#)
[Comunidade](#)
[Português - Brasil \(pt_br\)](#)
[Deutsch \(de\)](#)
[English \(en\)](#)
[Português - Brasil \(pt_br\)](#)

[Resumo de retenção de dados](#)

[Baixar o aplicativo móvel.](#)

 [Dê um feedback sobre este software](#) 

Universidade Tecnológica Federal do Paraná - UTFPR
Suporte ao usuário

