

Programação de Aplicações Corporativas

Arquitetura MVC sem a utilização de Frameworks

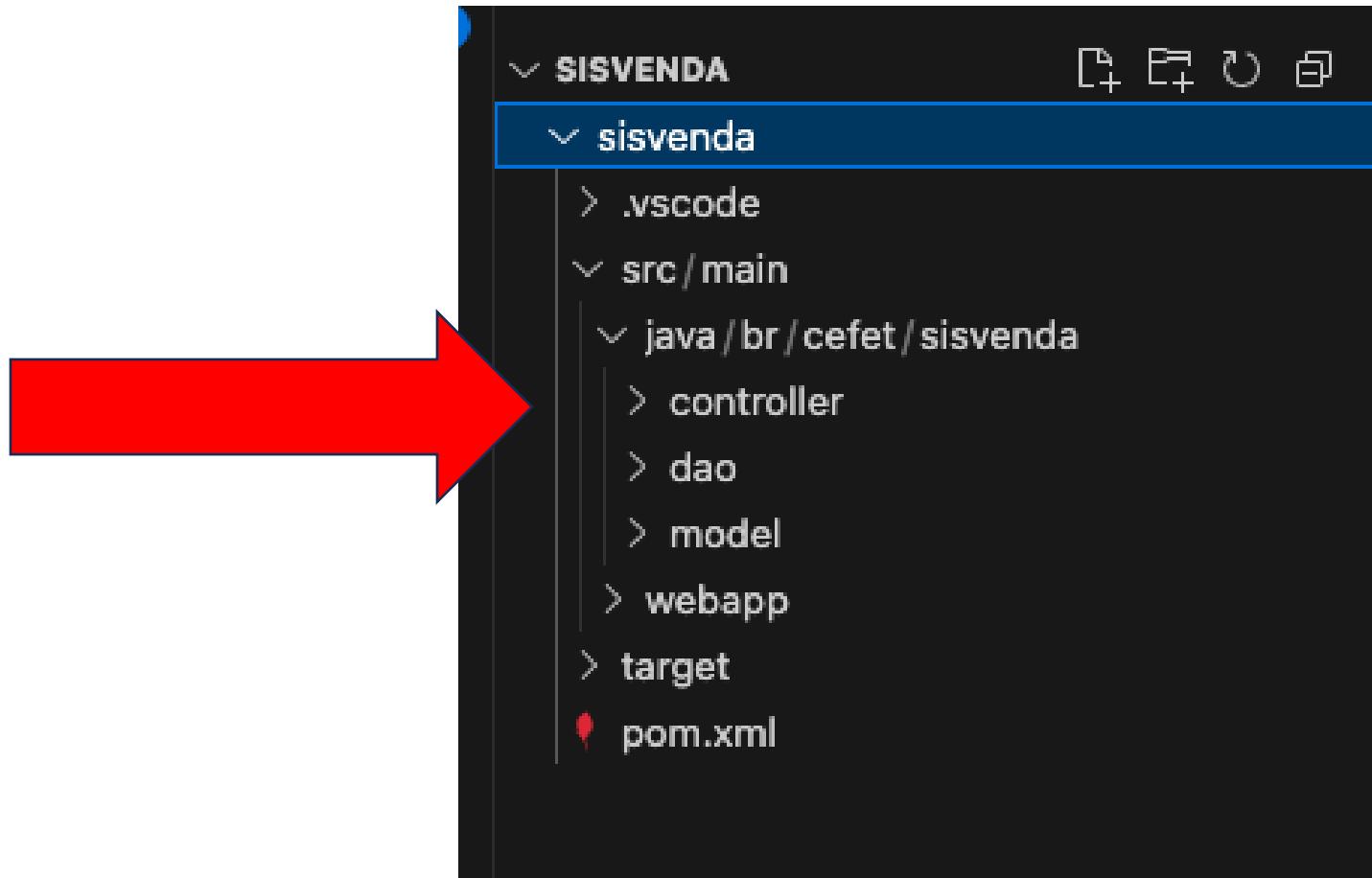
View: HTML e Java Script

Controller: Servlet (LEGADO) despachando JSON

DAO: Implementação de classes de acesso aos dados relacionais usando SQL

Model: Java Bean (atributos privados, construtores, gest e sets)

Camada Controller



Servlet CategoriaServlet2

- Despachará um JSON por meio de uma resposta HTTP;
- Para trabalhar com JSON utilizaremos a Biblioteca *Gson* do Google.

Verbo	Rota	Explicação
GET	/categorias2	Lista todas as categorias.

```
@WebServlet("/categorias2")
public class CategoriaServlet2 extends HttpServlet {
}
```

```
<dependency>
<groupId>com.google.code.gson</groupId>
<artifactId>gson</artifactId>
<version>2.10.1</version>
</dependency>
```

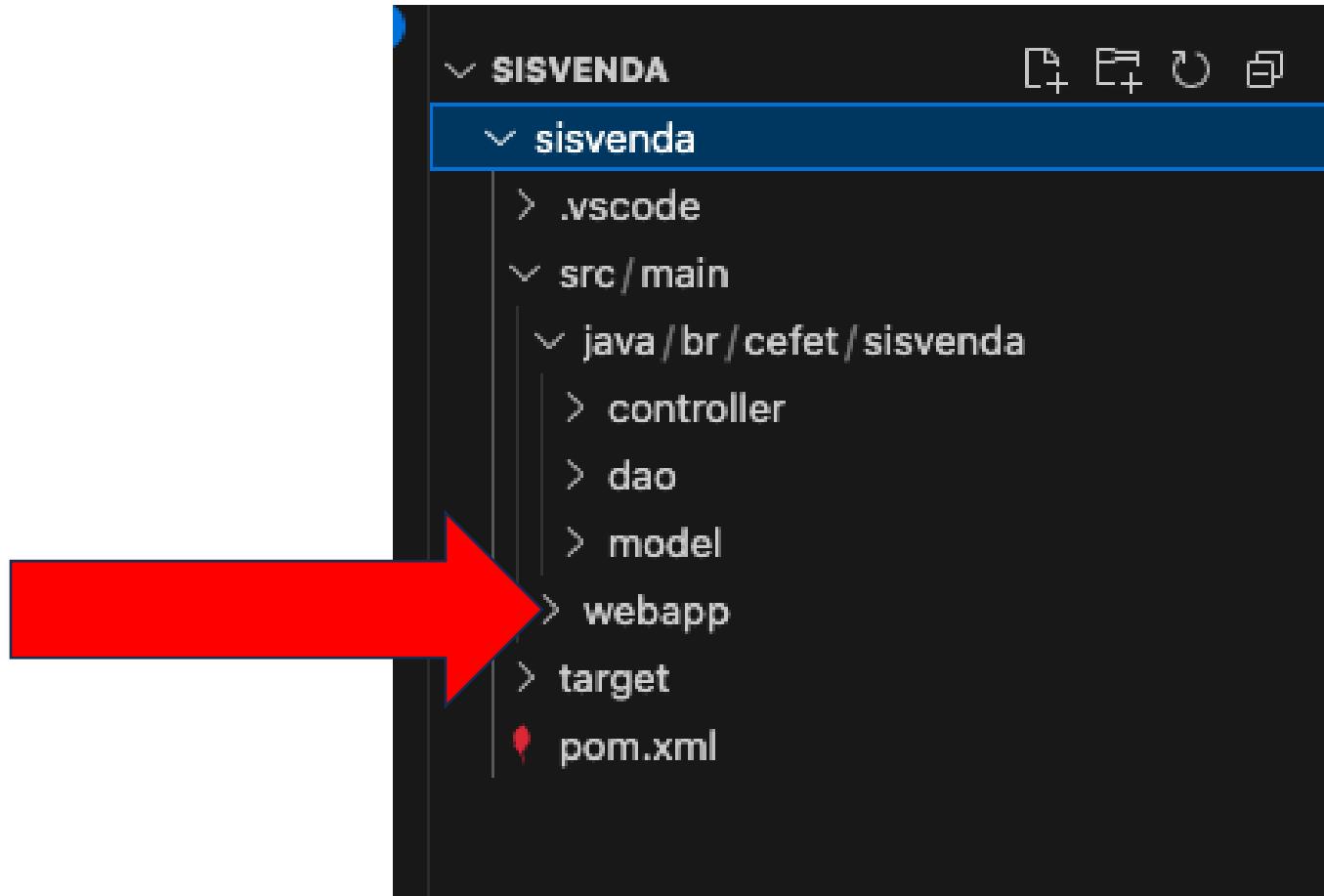
CategoriaServlet2 (doGet)

```
CategoriaDao catDao = new CategoriaDao();
List<Categoria> categorias = null;
try {
    categorias = catDao.listarTodos();
} catch (SQLException e) {
    e.printStackTrace();
}
```

```
// Converte a lista em JSON usando Gson
Gson gson = new Gson();
String json = gson.toJson(categorias);
```

```
// Configura a resposta HTTP
response.setContentType("application/json");
response.setCharacterEncoding("UTF-8");
PrintWriter out = response.getWriter();
out.print(json);
out.flush();
```

Camada View



Registrar Categoria (categorialistarjson.html)

```
<body><h2>Lista de Categorias</h2>
```

```
<table id="tabela-categorias">
```

```
<thead>
```

```
<tr>
```

```
    <th>ID</th>
```

```
    <th>Nome</th>
```

```
</tr>
```

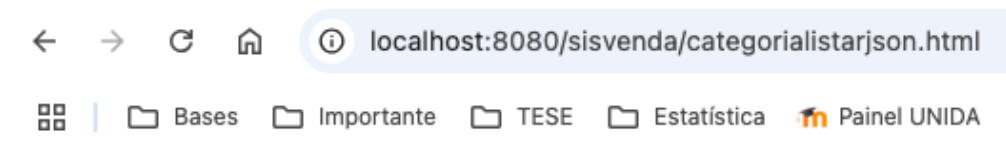
```
</thead>
```

```
<tbody>
```

```
<!-- Categorias serão inseridas aqui pelo JavaScript -->
```

```
</tbody>
```

```
</table>
```



Lista de Categorias

ID	Nome
43	Açaougue
44	Padaria
45	Limpeza
46	teste22

Registrar Categoria (categorialistarjson.html)

```
<script>
fetch('/sisvenda/categorias2')
.then(response => {
if (!response.ok) {
    throw new Error("Erro ao buscar categorias");
}
return response.json();
})
.then(categorias => {
const tbody = document.querySelector("#tabela-categorias tbody");
```

Registrar Categoria (categorialistarjson.html)

```
categorias.forEach(categoria => {
  const tr = document.createElement("tr");

  tr.innerHTML = `
    <td>${categoria.id}</td>
    <td>${categoria.nome}</td>`;

  tbody.appendChild(tr);
});

}).catch(error => {
  console.error("Erro:", error);
  alert("Erro ao carregar categorias.");
});
</script>
```

Menu (menu.jsp)

A screenshot of a web browser window. The title bar shows the URL: localhost:8080/sisvenda/menu.jsp. The page content includes a navigation bar with icons for back, forward, search, and home, followed by links to 'Bases', 'Importante', 'TESE', 'Estatística', and a help icon. Below this is a horizontal line. Underneath the line, there are four purple underlined links: 'Registrar Categoria', 'Listar Categoria', 'Registrar Produto', and 'Buscar Produtos'.

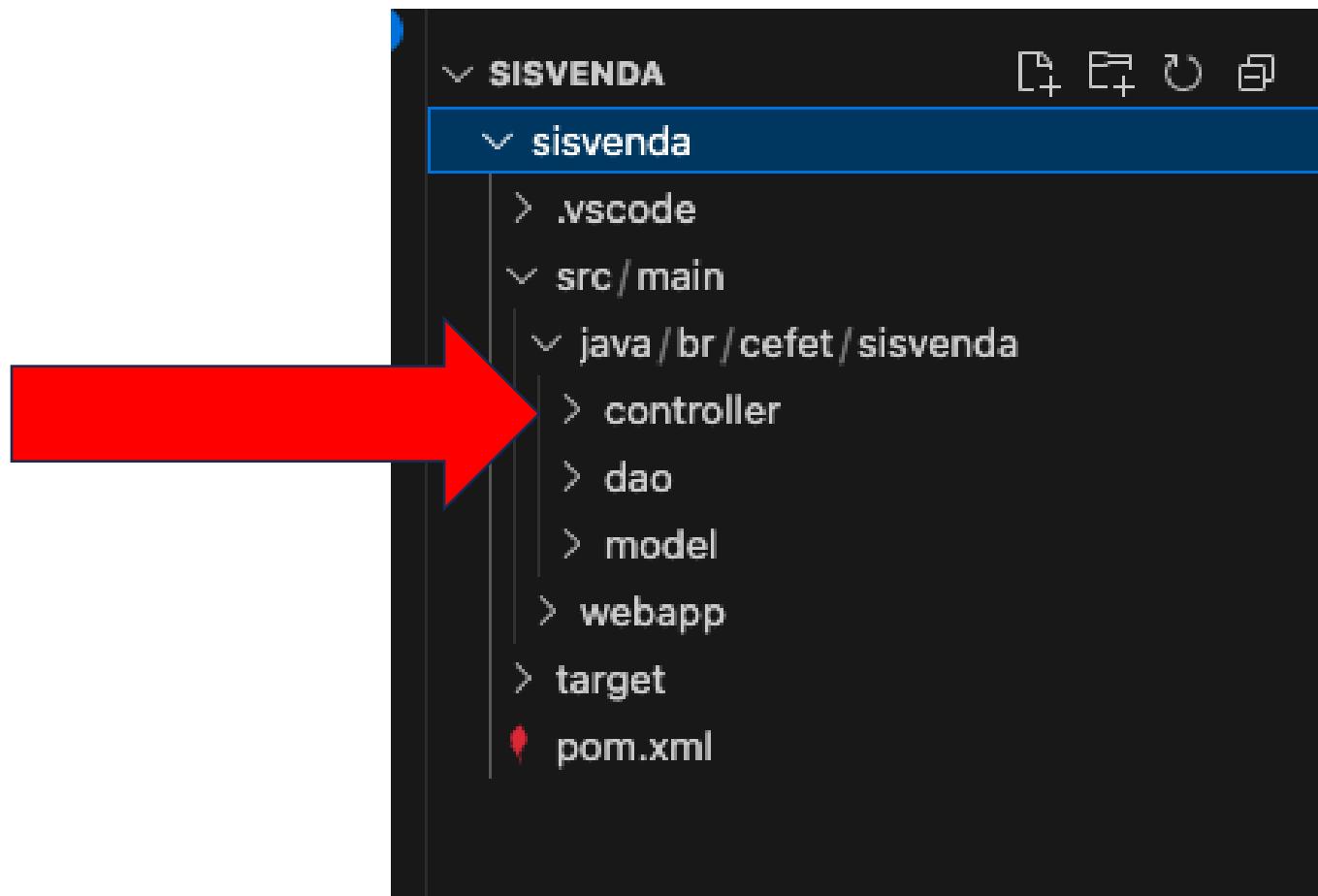
JSON

[Listar Categorias](#)
[Buscar Produtos](#)

```
<p> JSON </p>
<a href="/sisvenda/categorialistarjson.html"> Listar Categorias </a> <br>
<a href="/sisvenda/produtolistarjson.html"> Buscar Produtos</a> <br>
```

Camada Controller

ServletProduto2



Servlet Produto2

- Controller que propõe o listar um, listar vários, atualizar e excluir um produto;
- Para incluir e listar, a view executará o método HTTP apropriado e passará o JSON na requisição HTTP.
- A exclusão será por meio do ID;

Verbo	Rota	Explicação
GET	/produtos2?q=texto	Listar produtos de acordo com uma string fornecida.
GET	/produtos2/1	Lista apenas um produto (ex. id = 1).
PUT	/produtos2/	Recebe um JSON e atualiza o produto.
DELETE	/produtos2/1	Exclui um produto (ex. id = 1).

```
@WebServlet("/produtos2")
public class ProdutoServlet2 extends HttpServlet {
}
```