



## Missão Prática | Nível 4 | Mundo 3

Camilla Rodrigues Alves Gomes - 202302631673

**Campus Nova América**

**Disciplina: RPG0017 Vamos integrar sistemas – Turma 9001 – 2024.1**

### Objetivo da Prática

Implementação de sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE.

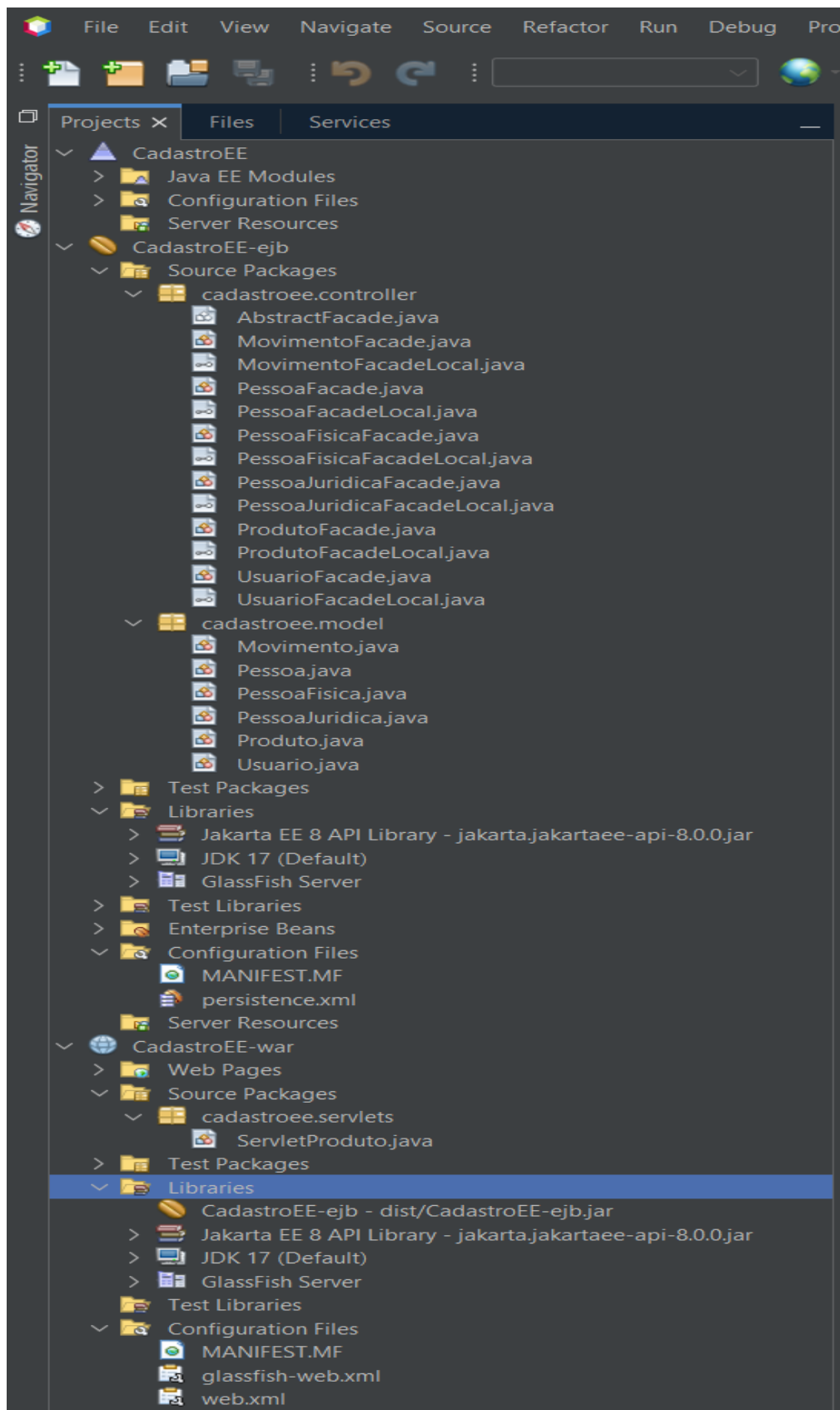
- Implementar persistência com base em JPA.
- Implementar regras de negócio na plataforma JEE, através de EJBs.
- Implementar sistema cadastral Web com base em Servlets e JSPs.
- Utilizar a biblioteca Bootstrap para melhoria do design.
- No final do exercício, o aluno terá criado todos os elementos necessários para exibição e entrada de dados na plataforma Java Web, tornando-se capacitado para lidar com contextos reais de aplicação.

### 1º Procedimento | Camadas de Persistência e Controle

**Códigos:** <https://github.com/Camilla-Alves/CadastroEE>

**Resultado da execução do projeto:**





*Estrutura do projeto depois dos procedimentos pedidos.*

## Análise e Conclusão:

a) Como é organizado um projeto corporativo no NetBeans?

- Um projeto corporativo no NetBeans é organizado em módulos. Cada módulo representa um componente lógico da aplicação, como uma camada de negócios, uma camada de apresentação ou uma camada de acesso a dados. Os módulos são agrupados em um projeto principal que gerencia as dependências e a construção da aplicação.

b) Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para a plataforma Web no ambiente Java?

- JPA (Java Persistence API): Fornece uma interface padrão para acesso e gerenciamento de dados relacionais. Ele mapeia objetos Java para tabelas de banco de dados e vice-versa, facilitando o acesso e a persistência de dados.
- EJB (Enterprise JavaBeans): Uma especificação que define componentes reutilizáveis que podem ser usados em aplicativos corporativos distribuídos. Os EJBs fornecem serviços como gerenciamento de transações, segurança e acesso a recursos.

c) Como o NetBeans viabiliza a melhoria de produtividade ao lidar com as tecnologias JPA e EJB?

- ✓ Modelos e preenchimento automático de código para classes JPA e EJB.
- ✓ Navegação e validação de código fáceis.
- ✓ Integração com servidores de aplicativos para implantação e depuração.
- ✓ Assistente de persistência para geração automática de classes de entidade JPA.
- ✓ Editor avançado com recursos como destaque de sintaxe e refatoração de código.

d) O que são Servlets, e como o NetBeans oferece suporte à construção desse tipo de componentes em um projeto Web?

- Servlets são componentes Java que respondem a requisições HTTP e são usados para criar aplicativos web dinâmicos.
- O NetBeans oferece suporte completo para a construção de Servlets em projetos web. Ele fornece modelos e assistentes para criar Servlets, facilitando a configuração do ambiente de desenvolvimento. Além disso, o NetBeans inclui um servidor web embutido para testar e depurar Servlets localmente, o que simplifica o processo de desenvolvimento.

e) Como é feita a comunicação entre os Servlets e os Session Beans do pool de EJBs?

- Através da injeção de dependência, usando anotações como @EJB. Isso permite que os servlets acessem métodos dos Session Beans para executar operações de negócio de forma simplificada, como acesso a banco de dados ou processamento de dados. Esta abordagem facilita o desenvolvimento de aplicativos, pois os servlets podem delegar tarefas complexas aos Session Beans, que são gerenciados pelo container EJB.

## **2º Procedimento | Interface Cadastral com Servlet e JSPs**

**Códigos:** <https://github.com/Camilla-Alves/CadastroEE>

### **Resultados:**

*Primeira imagem: resultado ao listar todos os produtos.*

*Segunda imagem: incluindo um novo produto chamado Pera.*

*Terceira imagem: lista após a inclusão do novo produto.*

*Quarta imagem: alteração da quantidade e preço do produto Banana.*

*Quinta imagem: lista após as alterações.*

Lista de Produtos

localhost:8080/CadastroEE-war/ServletProdutoFC?acao=listar

## Listagem de Produtos

ID	Nome	Quantidade	Preço	Opções
1	Banana	100	5.0	<a href="#">Alterar</a>   <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a>   <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a>   <a href="#">Excluir</a>

[Novo Produto](#)

Cadastro de Produtos

localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formIncluir

## Cadastro de Produto

[Voltar](#)

ID:  Nome:  Quantidade:  Preço de Venda:

Lista de Produtos

localhost:8080/CadastroEE-war/ServletProdutoFC

## Listagem de Produtos

ID	Nome	Quantidade	Preço	Opções
1	Banana	100	5.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
2	Pera	300	3.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

[Novo Produto](#)

Cadastro de Produtos

localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formAlterar&id=1

## Alteração de Produto

[Voltar](#)

ID:  Nome:  Quantidade:  Preço de Venda:

Lista de Produtos

localhost:8080/CadastroEE-war/ServletProdutoFC

## Listagem de Produtos

ID	Nome	Quantidade	Preço	Opções
1	Banana	650	4.5	<a href="#">Alterar</a> <a href="#">Excluir</a>
2	Pera	300	3.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

[Novo Produto](#)

Sexta imagem: Lista após a exclusão do produto Pera.

Lista de Produtos

localhost:8080/CadastroEE-war/ServletProdutoFC?acao=excluir&id=2

## Listagem de Produtos

ID	Nome	Quantidade	Preço	Opções
1	Banana	650	4.5	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

[Novo Produto](#)

## Análise e Conclusão:

- a) Como funciona o padrão Front Controller, e como ele é implementado em um aplicativo Web Java, na arquitetura MVC?
- O Front Controller é um padrão de design que centraliza o processamento de solicitações em um único componente, chamado de controlador. Ele gerencia o roteamento das solicitações, processa as requisições e coordena as ações do sistema.
  - No contexto da arquitetura MVC (Model-View-Controller), o Front Controller é geralmente implementado como um Servlet. Ele recebe todas as solicitações do cliente e decide qual ação tomar com base nas informações contidas na solicitação. O controlador então interage com o modelo (dados) e a visão (interface do usuário) conforme necessário, antes de enviar uma resposta de volta ao cliente.
- b) Quais as diferenças e semelhanças entre Servlets e JSPs?
- Semelhanças: Ambos são componentes Java usados para desenvolvimento web, são executados no servidor, interagem com o cliente através do protocolo HTTP, têm acesso aos objetos de requisição e resposta e são altamente extensíveis.
  - Diferenças: Servlets são predominantemente escritos em Java puro e são mais adequados para manipulação de lógica de negócios e processamento de solicitações. JSPs permitem a mistura de código Java e HTML, facilitando a criação de páginas web dinâmicas e são mais adequados para a apresentação de dados e interação com o usuário final.
- c) Qual a diferença entre um redirecionamento simples e o uso do método forward, a partir do RequestDispatcher? Para que servem parâmetros e atributos nos objetos HttpRequest?

- Redirecionamento simples: Envia uma resposta de redirecionamento para o navegador do cliente, instruindo-o a fazer uma nova solicitação para uma URL diferente. O navegador faz uma nova solicitação, e o cliente percebe que foi redirecionado.
- Forward usando RequestDispatcher: Permite que o servlet encaminhe a solicitação e a resposta para outro componente no servidor, como outro servlet, JSP ou arquivo estático, sem a participação do cliente. O encaminhamento é interno e transparente para o cliente, que não percebe que está acontecendo.
- Os parâmetros em objetos HttpRequest são dados enviados pelo cliente para o servidor como parte da solicitação HTTP. O servidor os utiliza para processar a solicitação e gerar a resposta. Eles podem incluir informações de formulários HTML ou parâmetros de URL.
- Os atributos em objetos HttpRequest são objetos associados ao escopo da solicitação (HttpServletRequest) pelo servidor. São úteis para compartilhar dados entre diferentes componentes durante o processamento da solicitação. Exemplos de atributos incluem objetos de sessão ou atributos definidos pelo servlet durante o processamento da solicitação.

### **3º Procedimento | Melhorando o Design da Interface**

**Códigos:** <https://github.com/Camilla-Alves/CadastroEE>

**Resultados:**



## Listagem de Produtos

Novo Produto

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana	650	4.5	<button>Alterar</button> <button>Excluir</button>
3	Laranja	500	2.0	<button>Alterar</button> <button>Excluir</button>
4	Manga	800	4.0	<button>Alterar</button> <button>Excluir</button>

## Alteração de Produto

Voltar

ID:

Nome:

Banana

Quantidade:

650

Preço de Venda:

4.5

Alterar

# Cadastro de Produto

[Voltar](#)

ID:

Nome:

Quantidade:

Preço de Venda:

[Adicionar](#)

## Análise e Conclusão:

a) Como o framework Bootstrap é utilizado?

- O Bootstrap é usado para criar sites responsivos rapidamente, com componentes e estilos CSS e JavaScript prontos para uso. Para incluir: use links CDN no <head>, baixe do site oficial e adicione os arquivos ao projeto, ou instale via npm/yarn e importe no JavaScript.

b) Por que o Bootstrap garante a independência estrutural do HTML?

- O Bootstrap garante a independência estrutural ao separar a apresentação (CSS) da estrutura (HTML). Suas classes predefinidas permitem aplicar estilos e comportamentos uniformes sem modificar o código HTML original.

c) Qual a relação entre o Bootstrap e a responsividade da página?

- O Bootstrap usa um grid flexível baseado em colunas que se ajusta automaticamente ao tamanho da tela, permitindo que os layouts se adaptem a diferentes dispositivos e resoluções de forma eficiente.