


| | |
|---|--|
|  Estácio | Universidade Estácio Campus Castelo Curso de Desenvolvimento Full Stack Relatório da Missão Prática 4 - Mundo 3 |
| Disciplina: | RPG0017 - Vamos Integrar Sistemas |
| Nome: | Breno Ambrosim Louzada |
| Turma: | 2023.1 |

Implementação de sistema cadastral com interface web,
baseado nas tecnologias de servlets, JPA e JEE

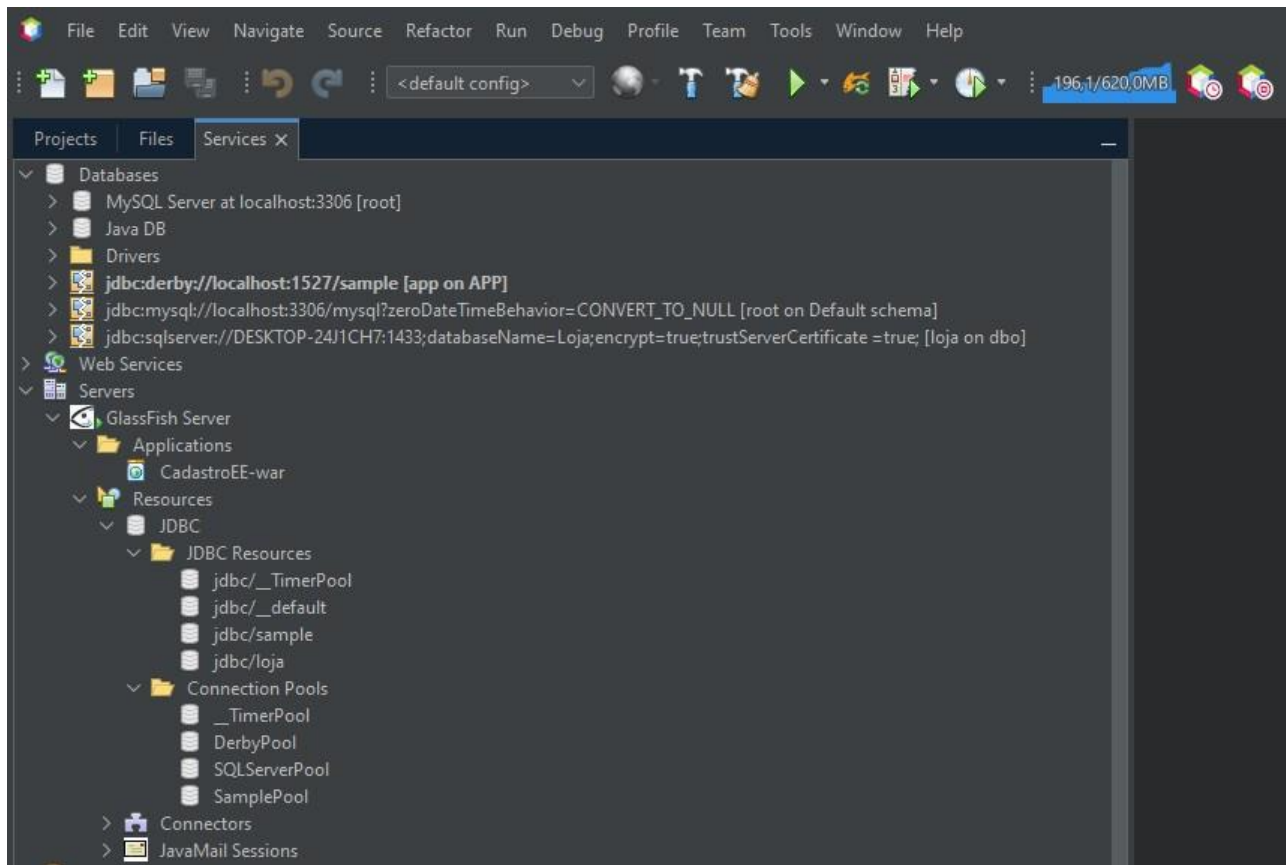
1. Título da Prática: “1º Procedimento | Camadas de Persistência e Controle”

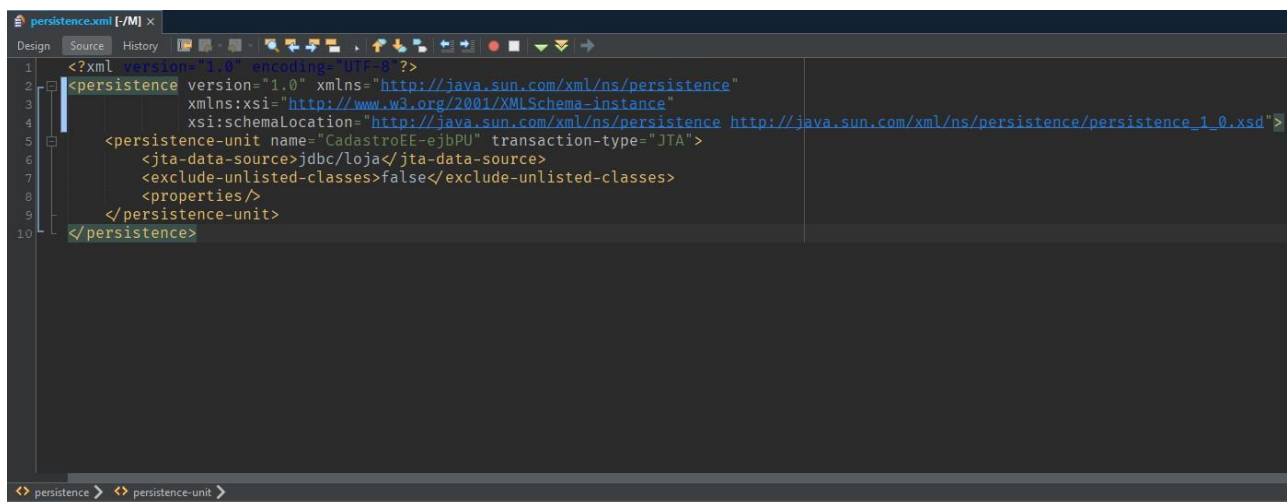
2. Objetivo da Prática

- 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.

3. Códigos solicitados: estão no repositório <https://github.com/BrenoAmbrosim/CadastroEE>

4. Resultados da execução dos códigos





```
<?xml version="1.0" encoding="UTF-8" ?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
  <persistence-unit name="CadastroEE-ejbPU" transaction-type="JTA">
    <jta-data-source>jdbc/loja</jta-data-source>
    <exclude-unlisted-classes>>false</exclude-unlisted-classes>
    <properties/>
  </persistence-unit>
</persistence>
```



5. Análise e Conclusão

1. Como é organizado um projeto corporativo no NetBeans? A organização de um projeto corporativo no NetBeans geralmente segue uma estrutura modular e hierárquica. No início, você cria um projeto corporativo, que serve como um contêiner para os módulos do projeto. Dentro deste contêiner, podem ser adicionados diferentes tipos de módulos, como projetos EJB para a lógica de negócios, projetos web para a interface do usuário, e projetos de aplicativo cliente para a interação do usuário. Cada módulo é gerenciado separadamente, mas integrado dentro do projeto corporativo para garantir a coesão e a funcionalidade completa do sistema. O ambiente do NetBeans facilita esta organização fornecendo ferramentas e assistentes para configurar, construir e implantar cada parte do projeto de maneira eficiente.

2. Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para plataforma web no ambiente java? As tecnologias JPA (Java Persistence API) e EJB (Enterprise JavaBeans) desempenham papéis cruciais na construção de aplicativos web em ambiente Java. JPA é responsável pela camada de persistência, facilitando a interação com bancos de dados através de uma abordagem orientada a objetos, permitindo que os desenvolvedores manipulem entidades como objetos Java comuns, sem se preocupar com

as complexidades das operações de banco de dados. Por outro lado, EJB é usado para a lógica de negócios, proporcionando um framework robusto para construir componentes de servidor reutilizáveis e escaláveis, gerenciando aspectos como transações, segurança e concorrência. Juntas, essas tecnologias oferecem uma plataforma poderosa para o desenvolvimento de aplicações web empresariais, permitindo uma divisão clara entre lógica de negócios e persistência de dados, além de fornecer um conjunto de serviços que facilitam o desenvolvimento e a manutenção de aplicações complexas.

3. 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 funcionam no lado do servidor e são usados para estender as capacidades de um servidor web. Eles são frequentemente empregados para processar ou armazenar dados que foram submetidos por um formulário HTML, interagir com bancos de dados e outras aplicações de servidor, e produzir conteúdo web dinâmico. O NetBeans, como um ambiente de desenvolvimento integrado (IDE), oferece suporte à criação e gestão de Servlets em projetos web, facilitando o processo com recursos como assistentes de criação, modelos de código, e ferramentas para depurar e testar os Servlets dentro do contexto de um aplicativo web. Isso torna o NetBeans uma escolha popular para desenvolvedores que trabalham com tecnologias Java EE, incluindo Servlets.

4. Como é feita a comunicação entre os Servlets e os Session Beans do pool de EJBs? A comunicação entre Servlets e Session Beans em um pool de Enterprise JavaBeans (EJBs) é realizada através de um mecanismo chamado JNDI (Java Naming and Directory Interface). Os Servlets utilizam o JNDI para localizar e referenciar os Session Beans disponíveis no pool de EJBs. Após a localização, os Servlets podem invocar métodos nos Session Beans como se estivessem chamando métodos em objetos locais. Essa abordagem permite que os Servlets, que geralmente gerenciam solicitações de usuários via HTTP, interajam com lógicas de negócios complexas encapsuladas nos Session Beans, promovendo assim uma separação eficiente entre a camada de apresentação e a lógica de negócios.

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java with Maven
- Java with Gradle
- ✓ Java with Ant
 - JavaFX
 - Java Web
 - Java Enterprise**
 - NetBeans Modules
- HTML5/JavaScript
- C/C++
- PHP
- > Samples

Projects:

- Enterprise Application
- Enterprise Application with Existing Sources**
- EJB Module
- EJB Module with Existing Sources
- Enterprise Application Client
- Enterprise Application Client with Existing Sources

Description:

Imports an existing enterprise application into a standard IDE project. The enterprise application must conform to the J2EE BluePrints Project Conventions for Enterprise Applications. This wizard also imports components that are nested in the enterprise application. Standard projects use an IDE-generated Ant build script to build and run

< Back

Next >

Finish

Cancel

Help

1. Título da Prática: “2º Procedimento | Interface Cadastral com Servlet e JSP”

2. Objetivo da Prática

- 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.

3. Códigos solicitados: estão no repositório <https://github.com/BrenoAmbrosim/CadastroEE>

4. Resultados da execução dos códigos

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
CadastroEE-war - Apache NetBeans IDE 21
799.3/892.0MB

Projects Services Files
AbstractFacade.java
MovimentoFacadeLocal.java
MovimentoFacadeLocal.java
PessoaFacadeLocal.java
PessoaFacadeLocal.java
PessoaFisicaFacadeLocal.java
PessoaFisicaFacadeLocal.java
PessoaJuridicaFacadeLocal.java
PessoaJuridicaFacadeLocal.java
ProdutoFacadeLocal.java
ProdutoFacadeLocal.java
UsuarioFacadeLocal.java
UsuarioFacadeLocal.java
CadastroEE-war [master]
Web Pages
WEB-INF
glassfish-web.xml
web.xml
DbLista.jsp
ProdutoDados.jsp
estacio-logo.png
index.html
Remote Files
bootstrap.bundle.min.js
bootstrap.min.js
bootstrap.min.css
bootstrap.min.css
bootstrap.min.css
bootstrap.min.css
Source Packages
cadastroee.servlets
ServletProdutoFC.java
ServletProdutoFC.java
Test Packages
Libraries
Test Libraries
Configuration Files

ServletProdutoFC.java X
Source History
16 import jakarta.servlet.http.HttpServletRequest;
17 import jakarta.servlet.http.HttpServletResponse;
18
19 @WebServlet(name = "ServletProdutoFC", urlPatterns = {"/ServletProdutoFC"})
20 public class ServletProdutoFC extends HttpServlet {
21
22     @EJB
23     private ProdutoFacadeLocal facade;
24
25     @Override
26     protected void doGet(HttpServletRequest request, HttpServletResponse response)
27         throws ServletException, IOException {
28         String acao = request.getParameter(string:"acao");
29         String destino = handleGetAction(acao, request);
30         dispatchRequest(request, response, destino);
31     }
32
33     @Override
34     protected void doPost(HttpServletRequest request, HttpServletResponse response)
35         throws ServletException, IOException {
36         String acao = request.getParameter(string:"acao");
37         acao = acao == null || acao.isEmpty() ? " " : acao;
38         String destino = handlePostAction(acao, request);
39         dispatchRequest(request, response, destino);
40     }
41
42     private String handleGetAction(String acao, HttpServletRequest request) {
43         switch (acao) {
44             case "formIncluir":
45                 return "ProdutoDados.jsp";
46             case "excluir":
47                 return handleExcluir(request);
48             case "formAlterar":
49                 return handleFormAlterar(request);
50             default:
51                 return handleListarProdutos(request);
52         }
53     }
54
55     private String handlePostAction(String acao, HttpServletRequest request) {
56         switch (acao) {
57             case "incluir":
58                 return handleIncluir(request);
59         }
60     }
61 }
```

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
CadastroEE-war - Apache NetBeans IDE 21
799.3/892.0MB

Projects Services Files
AbstractFacade.java
MovimentoFacadeLocal.java
MovimentoFacadeLocal.java
PessoaFacadeLocal.java
PessoaFacadeLocal.java
PessoaFisicaFacadeLocal.java
PessoaFisicaFacadeLocal.java
PessoaJuridicaFacadeLocal.java
PessoaJuridicaFacadeLocal.java
ProdutoFacadeLocal.java
ProdutoFacadeLocal.java
UsuarioFacadeLocal.java
UsuarioFacadeLocal.java
CadastroEE-war [master]
Web Pages
WEB-INF
glassfish-web.xml
web.xml
DbLista.jsp
ProdutoDados.jsp
estacio-logo.png
index.html
Remote Files
bootstrap.bundle.min.js
bootstrap.min.js
bootstrap.min.css
bootstrap.min.css
bootstrap.min.css
bootstrap.min.css
Source Packages
cadastroee.servlets
ServletProdutoFC.java
ServletProdutoFC.java
Test Packages
Libraries
Test Libraries
Configuration Files

DbLista.jsp X
Source History
64 <table class="table table-striped table-bordered table-responsive">
65 <thead>
66 <tr class="table-dark">
67 <th>ID</th>
68 <th>Produto</th>
69 <th>Quantidade</th>
70 <th>Preço</th>
71 <th>Ações</th>
72 </tr>
73 </thead>
74
75 <%
76     DecimalFormat df = new DecimalFormat("#,##0.00");
77     List<Produto> produtos = (List<Produto>) request.getAttribute("produtos");
78
79     if (produtos != null && !produtos.isEmpty()) {
80         for (Produto produto : produtos) {
81
82             %>
83             <tr>
84                 <td class="text-center"><%=produto.getIdProduto()%></td>
85                 <td class="text-center"><%=produto.getNome()%></td>
86                 <td class="text-center"><%=produto.getQuantidade()%></td>
87                 <td class="text-center"><%=df.format(produto.getPrecoVenda())%></td>
88                 <td>
89                     <a class="btn btn-primary btn-sm" href="ServletProdutoFC?acao=formAlterar&id=<%=produto.getIdProduto()%>">Alterar</a>
90                     <a class="btn btn-danger btn-sm" href="ServletProdutoFC?acao=excluir&id=<%=produto.getIdProduto()%>">Excluir</a>
91                 </td>
92             </tr>
93             %>
94         } else {
95             %>
96             <tr>
97                 <td colspan="5">Nenhum produto encontrado.</td>
98             </tr>
99             %>
100         }
101     }
102 </table>
103
104 <div class="text-end mb-3">
105     <a class="btn btn-primary" href="ServletProdutoFC?acao=formIncluir">Cadastrar Produto</a>
106 </div>
107 </div>
```

| ID | Produto | Quantidade | Preço | Ações |
|----|-----------|------------|----------|---|
| 1 | Banana | 100 | R\$ 5,00 | Alterar Excluir |
| 2 | Laranja | 500 | R\$ 2,00 | Alterar Excluir |
| 3 | Manga | 800 | R\$ 4,00 | Alterar Excluir |
| 4 | Tangerina | 600 | R\$ 7,00 | Alterar Excluir |

[Cadastrar Produto](#)

5. Análise e Conclusão

(a) Como funciona o padrão Front Controller, e como ele é implementado em um aplicativo Web Java, na arquitetura MVC?

O padrão Front Controller (FC) em aplicações web Java, especialmente na arquitetura MVC (Model-View-Controller), funciona como um controlador centralizado que gerencia todas as solicitações do cliente.

Ao invés de utilizar vários controladores para diferentes tipos de solicitações, o FC atua como um ponto único de entrada, que interpreta as solicitações, delega as tarefas apropriadas para modelos específicos (model); em seguida, seleciona a visualização correta (view) para responder ao cliente.

É comum esse fluxo de interações ser implementado através de um servlet, o qual intercepta todas as solicitações, realiza o processamento necessário ou lógica de negócios e, finalmente, encaminha a resposta para a página JSP ou outra tecnologia de visualização de renderização de páginas. Este padrão auxilia na manutenção e gerenciamento centralizado das solicitações, de modo a promover a reutilização de código em uma estrutura mais organizada.

(b) Quais as diferenças e semelhanças entre Servlets e JSPs?

Servlets e JSP são tecnologias usadas para criar aplicações web em Java, mas possuem abordagens bem distintas.

Servlets são classes Java que permitem gerar código HTML a partir de instruções e comandos em Java; assim, os servlets são mais adequados para a lógica de negócios e processamento de dados.

Por outro lado, JSP são páginas HTML com capacidade de incorporar código Java; logo são mais adequados para a apresentação da interface com o usuário.

Ambos, servlets e JSP, são executados no servidor de aplicações (GlassFish, Tomcat), com a possibilidade de interagir com bancos de dados, assim como outras tecnologias Java Enterprise.

Em resumo, enquanto servlets são mais voltados para controle, JSPs são mais adequados para a visualização e, frequentemente, ambos são usados em conjunto para separar a lógica de negócios da interface de usuário.

(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?

O redirecionamento simples utiliza o método `response.sendRedirect(url)`, enquanto que o RequestDispatcher utiliza o método `forward`. A principal diferença entre eles está no tratamento da requisição. O redirecionamento simples envia uma resposta ao navegador, para indicar que este deve fazer uma nova requisição para outra URL. Já o forward encaminha a requisição atual para outro recurso no servidor sem informar ao cliente, e mantém a URL original.

Os parâmetros e atributos em objetos HttpRequest são fundamentais para passar informações entre cliente e servidor, ou entre diferentes partes do servidor. Parâmetros são tipicamente usados para enviar dados de formulários ou de solicitações de URL, enquanto atributos são usados para manter dados durante a vida útil de uma requisição ou sessão, de modo a permitir a comunicação entre diferentes componentes do servidor.

1. Título da Prática: “3º Procedimento | Melhorando o Design da Interface”

2. Objetivo da Prática

- 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.

3. Códigos solicitados: estão no repositório <https://github.com/BrenoAmbrosim/CadastroEE>

4. Resultados da execução dos códigos

O 3º procedimento utiliza a biblioteca do framework Bootstrap, o que torna a interface gráfica muito mais bonita e agradável.

A fig. 9 mostra a mesma listagem de produtos do procedimento anterior, mas com uso de Bootstrap, conforme solicitado no enunciado da Missão.

← ↻ 🏠 ⓘ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=listar 🔊 ☆ ⚙️ | 📄 ☆ 🗂️ 🔄 ⌵ ...

Listagem de Produtos

| ID | Produto | Quantidade | Preço | Ações |
|----|-----------|------------|----------|----------------------------|
| 1 | Banana | 100 | R\$ 5,00 | <div>Alterar Excluir</div> |
| 2 | Laranja | 500 | R\$ 2,00 | <div>Alterar Excluir</div> |
| 3 | Manga | 800 | R\$ 4,00 | <div>Alterar Excluir</div> |
| 4 | Tangerina | 600 | R\$ 7,00 | <div>Alterar Excluir</div> |

Cadastrar Produto

← ↻ 🏠 ⓘ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formIncluir 🔊 ☆ ⚙️ | 📄 ☆ 🗂️ 🔄 ⌵ ...

Cadastro de Produto

Voltar

Nome

Abacaxi

Quantidade

200

Preço de Venda

8.00

Cadastrar

Alteração de Produto

Voltar

Nome

Tangerina

Quantidade

600

Preço de Venda

7.0

Alterar

(a) Como o framework Bootstrap é utilizado?

O framework Bootstrap é amplamente utilizado para desenvolver interfaces de usuário responsivas, *mobile first*, para websites e aplicações web. Oferece um conjunto robusto de ferramentas baseadas em HTML, CSS e JavaScript, que incluem templates pré desenvolvidos para botões, formulários, navegação e outros elementos de interface, além de um sistema de grid flexível para layout. Isso permite aos desenvolvedores rapidamente construir sites esteticamente agradáveis, funcionais, e que se adaptem automaticamente a diferentes tamanhos de tela e dispositivos, sem a necessidade de escrever código de estilização CSS desde o início.

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

O Bootstrap assegura a independência estrutural do HTML, ao fornecer um conjunto de classes CSS pré-definidas, assim como componentes de interface que podem ser facilmente integrados ao HTML. Em razão disso, ao invés de escrever e ajustar diferentes estilos CSS personalizados para cada elemento, os desenvolvedores podem simplesmente usar as classes do Bootstrap para alcançar um design consistente e responsivo.

Portanto, a estrutura do HTML permanece limpa e desacoplada de estilos específicos, o que facilita a manutenção e a escalabilidade do código.

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

O Bootstrap é um framework de desenvolvimento web que facilita a criação de páginas responsivas. Oferece um sistema de grid flexível, componentes pré-desenvolvidos e classes CSS que se ajustam automaticamente ao tamanho da tela do dispositivo, de maneira a garantir que o layout da página seja otimizado para desktops, tablets e smartphones. Isso simplifica o processo de design responsivo, o que permite que desenvolvedores criem sites que proporcionam uma experiência de usuário consistente em diversos dispositivos com menos esforço.