

Missão Prática | Nível 4

Mundo 3

Objetivos da Prática:

- 1.Implementar persistência com base em JPA;
- 2.Implementar regras de negócio na plataforma JEE através de EJBs;
- 3.Implementar sistema cadastral Web com base em Servlets e JSPs;
- 4.Utilizar a biblioteca Bootstrap para melhoria do design;
- 5.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.

Todos os códigos solicitados neste roteiro de aula:

/*Nota:

Professor, sei que isso não é muito natural, mas não consegui implementar os procedimentos do sway. Segui todos os passos e passei por todos os erros que davam, mas na hora de rodar o projeto acontecia um erro que não consegui resolver.

Percebi que meus colegas de classe também encontraram muito bicos sem saída com o NetBeans, o glassfish e o SQLServer. Como sou iniciante na programação, a outra única IDE que eu estou familiarizada é o Eclipse. Tentei usar Eclipse com Tomcat e MySQL (prefiro esse servidor por ter mais materiais de apoio na internet), também tentei Eclipse + glassfish + MySQL, NetBeans + Tomcat + MySQL, NetBeans + glassfish + MySQL, e o NetBeans + glassfish e SQLServer. Desinstalei e reinstalei esses programas várias vezes para garantir que ia conseguir o caminho mais limpo possível, gastei várias horas ao passar dos dias pesquisando cada erro, mas como tudo isso foi apenas para o preparo do ambiente para o que aprendemos com essa disciplina, desisti dois dias antes do prazo final. Vou colocar no relatório os códigos que eu teria usado se tivesse conseguido rodar o projeto.

Sei com certeza isso não vai valer os pontos inteiros, mas eu realmente me esforcei muito, por isso, qualquer ponto que esse relatório valha ajudaria muito. Espero futuramente conseguir configurar um ambiente para uma aplicação web de outras formas quando tiver mais tempo, mas essa realmente não me ajudou e me desmotivou bastante.

Espero sua compreensão.

*/

ProdutoLista.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
YvpcrYf0tY3IHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcldslK1eN7N6jleHz"
crossorigin="anonymous"></script>
    <title>Listagem de Produtos</title>
</head>
<body class="container">
    <h1>Listagem de Produtos</h1>
    <a href="ServletProdutoFC?acao=formIncluir" class="btn btn-primary m-2">Novo
Produto</a>

    <table class="table table-striped">
```

```

<thead class="table-dark">
  <tr>
    <th>#</th>
    <th>Nome</th>
    <th>Quantidade</th>
    <th>Preço de Venda</th>
    <th>Opções</th>
  </tr>
</thead>
<tbody>
  <%
List<Produto> produtoList=(List<Produto>)request.getAttribute("produtoList");
    if (produtoList != null) {
      for (Produto produto : produtoList) {
  <%>
    <tr>
      <td><%= produto.getId() %></td>
      <td><%= produto.getNome() %></td>
      <td><%= produto.getQuantidade() %></td>
      <td><%= produto.getPreco() %></td>
      <td>
        <a href="ServletProdutoFC?acao=formAlterar&id=<%= produto.getId()
  <%>" class="btn btn-primary btn-sm mr-2">Alterar</a>
        <a href="ServletProdutoFC?acao=excluir&id=<%= produto.getId() %>"
class="btn btn-danger btn-sm">Excluir</a>
      </td>
    </tr>
  <%
    }
  }
  <%>
</tbody>

```

```
</table>
</body>
</html>
```

ProdutosDados.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcldslK1eN7N6jleHz"
crossorigin="anonymous"></script>

    <title>Dados do Produto</title>
</head>
<body class="container">
    <h1>Dados do Produto</h1>

    <form action="ServletProdutoFC" method="post">
        <%
            Produto produto = (Produto) request.getAttribute("produto");
            String acao = (produto == null) ? "incluir" : "alterar";
        %>
```

```

<input type="hidden" name="acao" value="<%= acao %>">

<%
    if ("alterar".equals(acao)) {
%>

        <input type="hidden" name="id" value="<%= produto.getId() %>">

<%
    }
%>

<div class="mb-3">
    <label for="nome" class="form-label">Nome:</label>
    <input type="text" id="nome" name="nome" class="form-control" value="<
%= produto != null ? produto.getNome() : "" %>">
</div>
<div class="mb-3">
    <label for="quantidade" class="form-label">Quantidade:</label>
    <input type="number" id="quantidade" name="quantidade" class="form-
control" value="<%= produto != null ? produto.getQuantidade() : "" %>">
</div>
<div class="mb-3">
    <label for="preco" class="form-label">Preço de Venda:</label>
    <input type="number" id="preco" name="preco" step="0.01" class="form-
control" value="<%= produto != null ? produto.getPreco() : "" %>">
</div>

<input type="submit" class="btn btn-primary" value="<%= produto == null ?
"Incluir" : "Alterar" %>">
</form>
</body>

```

</html>

Análise e Conclusão:

Como é organizado um projeto corporativo no NetBeans?

Projetos corporativos no NetBeans possuem divisões bem definidas e que seguem o padrão MVC. A divisão EJB do projeto representa o Model (a lógica de negócio); a divisão Web representa a View (a interface do usuário, abrangendo HTML, CSS, Javascript, páginas JSP e JSF), além de ser responsável pelos Servlets, que atuam como o Controller do projeto (responsável pelo processamento de requisições).

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

A tecnologia JPA é responsável por gerenciar a persistência de dados, “conversando” com o banco de dados e fazer a mapeação objeto-relacional (ORM). Já a tecnologia EJB é responsável pela lógica negócio, além de se integrar com o JPA, usando-o para gerenciar a persistência de dados.

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

O NetBeans provê a criação de projeto EJB e entidades JPA através de assistentes, o que diminui o tempo de escrita de códigos repetitivos e erros que podem ocorrer na escrita manual. Nesse trabalho, por exemplo, vê-se a eficiência desses assistentes ao lidar com o ORM, gerando todas as classes das tabelas do banco de dados escolhidos, e na geração dos Enterprise Beans, usando o padrão Facade.

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

São os responsáveis por gerenciar requisições e respostas HTTP da aplicação ao servidor. Isso também pode gerar páginas dinâmicas de acordo com a ação dos usuários. O NetBeans possui um assistente de criação para Servlets, criando automaticamente um após a inserção dos dados.

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

Através da anotação `@EJB` (esta própria anotação foi utilizada no trabalho como `@EJB ProdutoFacadeLocal facade`;

Essa especificação ocorreu pois o Servlet especifica qual método será requisitado ao EJB, que depois retornará o resultado ao Servlet. Este por fim gerará a resposta HTTP para o usuário.

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

Sendo um padrão de design, ele centraliza o processo de solicitações de entrada, atuando como um “controlador principal” para recebê-las e distribuí-las aos outros controladores. Em um aplicativo Web Java, ele seria o servlet, captando as solicitações HTTP e encaminhando-as para os controladores apropriados.

Quais as diferenças e semelhanças entre Servlets e JSPs?

Servlets são classes java, enquanto que páginas JSP são páginas HTML que permitem as marcações de java. Os servlets, apesar de mais complexos, permitem uma organização de código melhor, mas ambas as tecnologias funcionam no lado do servidor.

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?

No redirecionamento simples, quem faz a solicitação é o usuário (seu navegador), ou seja, é um processamento cliente-servidor. No método `RequestDispatcher.forward`, essa solicitação é tratada no lado servidor (entre ele e o servlet). Os parâmetros `HttpRequest` servem para transmitir dados adquiridos nos formulários HTML para o servlet. Já os atributos transmitem informações entre servlets.

Como o framework Bootstrap é utilizado?

Como um estilizador para as páginas de forma mais simples, através de classes específicas atribuídas às tags. Sua utilização pode ser tanto via local (baixando seus arquivos), como online via CDN (que é o caso desse trabalho).

Por que o Bootstrap garante a independência estrutural do HTML?

Porque as classes Bootstrap nada mais são que classes CSS padronizadas, dessa forma, a estilização poderá ocorrer independente da reestruturação das tags HTML. O

sistema de grid do Bootstrap ajuda nessa flexibilidade, já que boa parte de suas áreas afetadas ocorrerão dentro de contêineres, como `divs`.

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

As estilizações responsivas do Bootstrap podem ser vistas na adição de linhas e botões da tabela corretamente formatados.