



Desenvolvimento de Aplicativo Web Corporativo utilizando Servlets, EJB, JPA e Bootstrap no NetBeans

Universidade Estácio de Sá Polo Araçatuba

Curso: Desenvolvedor Full Stack

Disciplina: Servlets-JPA-E-JEE

Turma: 2023/2

Semestre: 2024/4

Integrante: Felipe Nathan de Souza

Objetivo da Prática: O objetivo desta prática foi desenvolver um sistema web utilizando tecnologias Java, como Servlets, EJB (Enterprise Java Beans) e JPA (Java Persistence API), para criação de uma aplicação de cadastro de produtos. Além disso, o relatório também cobre a parte do front-end, onde foi utilizada a biblioteca Bootstrap para melhorar o design da interface.

Códigos Solicitados:

1. Código do ServletProduto

```
@WebServlet("/ServletProduto")
public class ServletProduto extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    private ProdutoFacadeLocal produtoFacade;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        List<Produto> listaProdutos = produtoFacade.listarProdutos();
        request.setAttribute("produtos", listaProdutos);
        RequestDispatcher dispatcher = request.getRequestDispatcher("ProdutoLista.jsp");
        dispatcher.forward(request, response);
    }
}
```

```
}
```

2. Código do ServletProdutoFC

```
@WebServlet("/ServletProdutoFC")
public class ServletProdutoFC extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    private ProdutoFacadeLocal produtoFacade;

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String acao = request.getParameter("acao");
        String destino = "ProdutoLista.jsp";

        switch (acao) {
            case "listar":
                List<Produto> produtos = produtoFacade.listarProdutos();
                request.setAttribute("produtos", produtos);
                break;
            case "formIncluir":
                destino = "ProdutoDados.jsp";
                break;
            case "incluir":
                Produto produto = new Produto();
                produto.setNome(request.getParameter("nome"));
                produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
                produto.setPreco(Double.parseDouble(request.getParameter("preco")));
                produtoFacade.incluirProduto(produto);
                break;
            // Outros casos: excluir, alterar, etc.
        }

        RequestDispatcher dispatcher = request.getRequestDispatcher(destino);
        dispatcher.forward(request, response);
    }
}
```

3. Código ProdutoFacadeLocal (EJB)

```
java
CopyEdit
@Local
public interface ProdutoFacadeLocal {
    void incluirProduto(Produto produto);
    List<Produto> listarProdutos();
    // Outros métodos para excluir, alterar produtos
}
```

```
        pfDAO.getPessoas().forEach(p -> p.exibir());  
pfDAO.excluir(pf.getId());    } catch (SQLException e) {  
    e.printStackTrace();  
}  
}}
```

Resultados da Execução dos Códigos

Durante a execução do código no ambiente NetBeans, a funcionalidade foi testada acessando as URLs configuradas no web.xml, com a comunicação entre Servlets e a exibição dos dados no front-end utilizando JSPs.

A aplicação foi capaz de:

1. Exibir a lista de produtos cadastrados.
2. Incluir, alterar e excluir produtos com os dados sendo persistidos no banco via JPA.
3. Realizar a navegação entre páginas de consulta e formulário com os links gerados dinamicamente.

Análise e Conclusão

1. Como é organizado um projeto corporativo no NetBeans?

No NetBeans, um projeto corporativo é organizado com a separação das responsabilidades. No caso da aplicação, temos a camada de **Modelo** (entidades), **Facade (EJB)**, **Servlets** para controle da lógica de negócios e a camada de **Visão** (JSP). O NetBeans facilita o gerenciamento das dependências e a configuração de servidor (ex: Tomcat ou Glassfish) diretamente pelo IDE.

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

O JPA (Java Persistence API) é utilizado para o mapeamento objeto-relacional, permitindo a persistência dos dados em um banco de dados relacional, como SQL Server. O EJB (Enterprise JavaBeans) fornece a lógica de negócios, e a separação entre as camadas facilita o desenvolvimento escalável e manutenível.

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

O NetBeans oferece uma integração nativa com as tecnologias JPA e EJB, possibilitando a criação e configuração de entidades, session beans e a execução da aplicação em servidores de aplicação, sem a necessidade de configurações manuais.

4. 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 processam requisições HTTP e gerenciam a lógica do lado do servidor. O NetBeans oferece ferramentas para a criação e configuração de Servlets automaticamente, permitindo uma fácil configuração de mapeamentos e roteamentos HTTP.

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

A comunicação é realizada via injeção de dependências, utilizando a anotação `@EJB`, o que permite que o Servlet tenha acesso aos métodos do EJB sem a necessidade de configuração adicional, e a comunicação entre as camadas é transparente.

Web Front

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

O padrão Front Controller atua como um controlador central que processa todas as requisições antes de passá-las para os componentes apropriados. No nosso caso, o `ServletProdutoFC` é o controlador principal que recebe as requisições, chama o EJB para manipulação dos dados e direciona a resposta para o JSP correspondente.

2. Quais as diferenças e semelhanças entre Servlets e JSPs?

- **Servlets** são usados para processar a lógica do servidor, manipular dados e gerar respostas dinâmicas.
- **JSPs (JavaServer Pages)** são usadas para a visualização, sendo a camada de apresentação. Servlets podem redirecionar ou encaminhar para JSPs para exibir o conteúdo. Ambos trabalham juntos para seguir o padrão MVC, onde o Servlet é o controlador e o JSP é a visão.

3. 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** é quando o servidor instrui o navegador a fazer uma nova requisição, mudando a URL.
- **Forward** é quando o servidor processa internamente uma nova requisição para outro recurso sem alterar a URL no navegador. Parâmetros e atributos no `HttpRequest` são usados para transmitir dados entre o controlador (Servlet) e a visão (JSP).

Design Front

1. Como o framework Bootstrap é utilizado?

O Bootstrap foi utilizado para estilizar as páginas de listagem e cadastro de produtos, oferecendo componentes prontos como botões, tabelas e formulários, que são facilmente integrados com as páginas JSP.

2. Por que o Bootstrap garante a independência estrutural do HTML?

O Bootstrap usa classes CSS que podem ser aplicadas em qualquer estrutura HTML, tornando o código flexível e modular. Isso permite que o design seja consistente sem interferir na lógica HTML.

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

O Bootstrap tem um sistema de grid flexível que ajusta automaticamente os elementos da página para diferentes tamanhos de tela, garantindo uma visualização otimizada em dispositivos móveis e desktops.

Observações Finais: O uso de Servlets, EJBs, JPA e Bootstrap permite a criação de uma aplicação web escalável, organizada e com um design agradável e responsivo, demonstrando as boas práticas de desenvolvimento Java para a plataforma Web.