Campus: POLO FSP - RO
Curso: Desenvolvimento Full Stack
Disciplina: Nível 4: Vamos integrar sistemas
Turma: POO-01
Semestre Letivo: 2025.1

Aluno: Eliton Rodrigues de Oliveira

Título da Prática

Desenvolvimento de uma aplicação corporativa Java EE com NetBeans, JPA, EJB e Servlets

Objetivo da Prática

Construir uma aplicação corporativa utilizando o NetBeans, empregando as tecnologias Java Persistence API (JPA), Enterprise Java Beans (EJB) e Servlets. O objetivo é integrar essas tecnologias para criar um sistema básico de cadastro e listagem de produtos, compreendendo como cada componente colabora no desenvolvimento de sistemas distribuídos para a plataforma Java EE.

📁 Códigos Fonte do Projeto

1. Produto.java (Entidade JPA)

```
package modelo;

import java.io.Serializable;

import jakarta.persistence.*;

@Entity

public class Produto implements Serializable {

    @Id

   @GeneratedValue(strategy = GenerationType.IDENTITY)

   private int id;
```

```
    private String nome;
private double precoVenda;
public Produto() {
}
public Produto(String nome, double precoVenda) {
    this.nome = nome;
    this.precoVenda = precoVenda;
}
public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public double getPrecoVenda() {
    return precoVenda;
}
public void setPrecoVenda(double precoVenda) {
    this.precoVenda = precoVenda;
}
```

```
}
```

2. ProdutoBean.java (Session Bean EJB)

```
package controle;

import jakarta.ejb.Stateless;

import jakarta.persistence.*;

import java.util.List;

import modelo.Produto;

@Stateless

public class ProdutoBean {

   @PersistenceContext(unitName = "MeuPU")

   private EntityManager em;

   public void inserir(Produto p) {

      em.persist(p);

   }

   public List<Produto> listar() {

      return em.createQuery("SELECT p FROM Produto p", Produto.class).getResultList();

   }

}
```

3. ServletProduto.java (Servlet)

```
package controle;

import jakarta.ejb.EJB;

import jakarta.servlet.*;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.*;

import java.io.IOException;

import java.io.PrintWriter;
```

```
import java.util.List;
import modelo.Produto;
@WebServlet("/produto")
public class ServletProduto extends HttpServlet {
   @EJB
   private ProdutoBean produtoBean;
   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try (PrintWriter out = response.getWriter()) {
         out.println("<html><head><title>Produtos</title></head><body>");
         out.println("<h1>Lista de Produtos</h1>");
         List<Produto> produtos = produtoBean.listar();
         if (produtos.isEmpty()) {
            out.println("<p>Nenhum produto cadastrado.</p>");
         } else {
            out.println("<ul>");
            for (Produto p : produtos) {
               out.println("<li>" + p.getNome() + " - R$ " + p.getPrecoVenda() + "</li>");
            }
            out.println("</ul>");
         }
         out.println("</body></html>");
      }
   }
```

```
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String nome = request.getParameter("nome");
        double preco = Double.parseDouble(request.getParameter("preco"));
        Produto novoProduto = new Produto(nome, preco);
        produtoBean.inserir(novoProduto);
        response.sendRedirect("produto");
    }
}
```

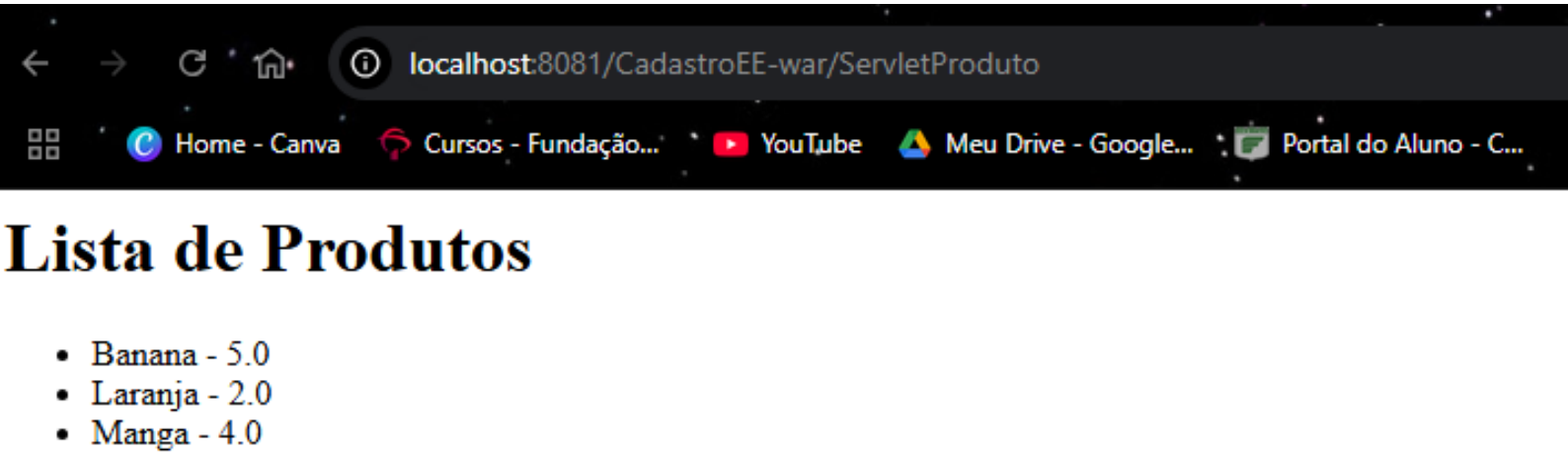
4. persistence.xml (Configuração de Persistência JPA)

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="https://jakarta.ee/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="https://jakarta.ee/xml/ns/persistence
             https://jakarta.ee/xml/ns/persistence/persistence_3_0.xsd"
             version="3.0">
  <persistence-unit name="MeuPU" transaction-type="JTA">
    <class>modelo.Produto</class>
    <properties>
      <property name="jakarta.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver"/>
      <property name="jakarta.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/seubanco"/>
      <property name="jakarta.persistence.jdbc.user" value="root"/>
      <property name="jakarta.persistence.jdbc.password" value="senha"/>
      <property name="jakarta.persistence.schema-generation.database.action"
value="create"/>
```

```
    </properties>
  </persistence-unit>
</persistence>
```

Resultados da Execução

- O projeto foi compilado e executado com sucesso após correção do nome do método.
- O Servlet apresentou a listagem dos produtos cadastrados no banco de dados, exibindo nome e preço.
- Foi possível observar a integração entre JPA (persistência), EJB (lógica de negócio) e Servlet (apresentação).



Análise e Conclusão

1. **Como é organizado um projeto corporativo no NetBeans?**

No NetBeans, um projeto corporativo Java EE é organizado em múltiplos módulos, geralmente incluindo:

- Um módulo *Enterprise Application* (EAR)
- Um módulo *EJB* (lógica de negócio e persistência com JPA)
- Um módulo *WAR* (web application – interface com Servlets, JSPs)

Essa estrutura favorece a separação de responsabilidades e facilita a manutenção e escalabilidade da aplicação.

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

- **JPA**: Responsável pelo mapeamento objeto-relacional (ORM), permite interagir com o banco de dados de forma orientada a objetos.
- **EJB**: Fornece componentes reutilizáveis com transações, segurança e persistência automatizadas. Os *Session Beans* encapsulam a lógica de negócio da aplicação.

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

O NetBeans oferece:

- Geração automática de entidades a partir de tabelas do banco
- Assistentes para criação de EJBs e Servlets
- Integração com servidores como GlassFish ou Payara, permitindo testes e deploy facilitados
- Ambiente visual para persistência, injeção de dependências e mapeamentos

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 geram respostas dinâmicas. Eles atuam na camada de controle em uma aplicação Web.
O NetBeans oferece suporte completo à criação, configuração e execução de Servlets com:

- Templates prontos
- Suporte à anotação @WebServlet
- Debugging integrado

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

A comunicação é feita via **injeção de dependência**, utilizando a anotação @EJB no Servlet. Isso permite que o Servlet invoque métodos dos Session Beans, que realizam a lógica de negócio e interações com o banco de dados.

@EJB

private ProdutoBean produtoBean;

Repositório GIT do Projeto

🔗 **GitHub**: https://github.com/Elitonr65/CadastroEE