

Aluno: Altair Silva Filho

Matrícula: 202403254409

Unidade: Palhoça – SC

Curso: Desenvolvimento Full Stack

Disciplina: Vamos integrar sistemas

Período: 2025.1

Relatório da Missão Prática – Nível 4 – Mundo 3

Vamos integrar sistemas

1º Procedimento | Camadas de persistência e controle

Objetivo

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

Neste procedimento, foram criadas as camadas de persistência e controle

Códigos do Projeto

Arquivos da camada de persistência e controle

Persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="3.0" 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">
  <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>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="5.0" xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-
app_5_0.xsd">
```

```

<servlet>
  <servlet-name>ServletProduto</servlet-name>
  <servlet-class>cadastroee.servlets.ServletProduto</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>ServletProduto</servlet-name>
  <url-pattern>/ServletProduto</url-pattern>
</servlet-mapping>

<session-config>
  <session-timeout>30</session-timeout>
</session-config>

</web-app>

```

ServletProduto.java

```

package cadastroee.servlets;

import cadastroee.controller.ProdutoFacadeLocal;
import cadastroee.model.Produto;
import jakarta.ejb.EJB;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.util.List;

public class ServletProduto extends HttpServlet {

    @EJB
    private ProdutoFacadeLocal facade;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

            List<Produto> produtos = facade.findAll();

            out.println("<html><body>");
            out.println("<h2>Servlet ServletProduto at /Cadastro-war</h2>");
            out.println("<ul>");
            for (Produto p : produtos) {
                out.printf("<li>%s", p.getNome());
            }
            out.println("</ul>");

```

```

        out.println("</body></html>");
    }
}
}

```

Resultados da execução dos códigos

As janelas com os resultados dos procedimentos e execuções do código são mostradas a seguir:

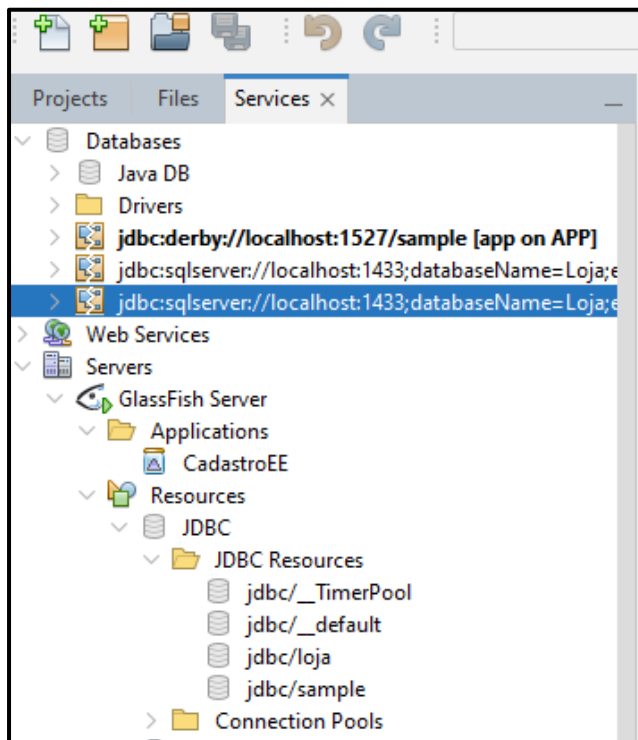


Figura 1 – Criação dos Drivers e Servidores

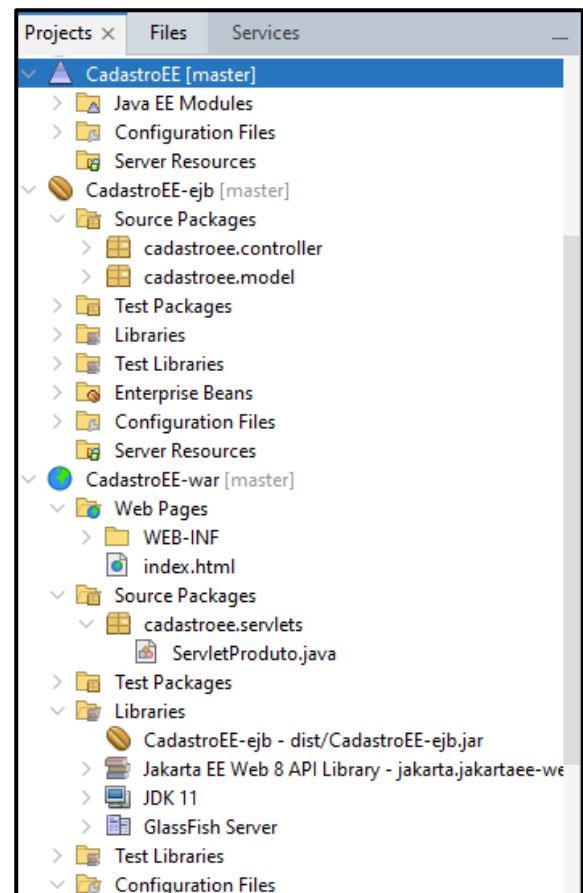


Figura 2 – Criação do projeto principal e demais arquivos

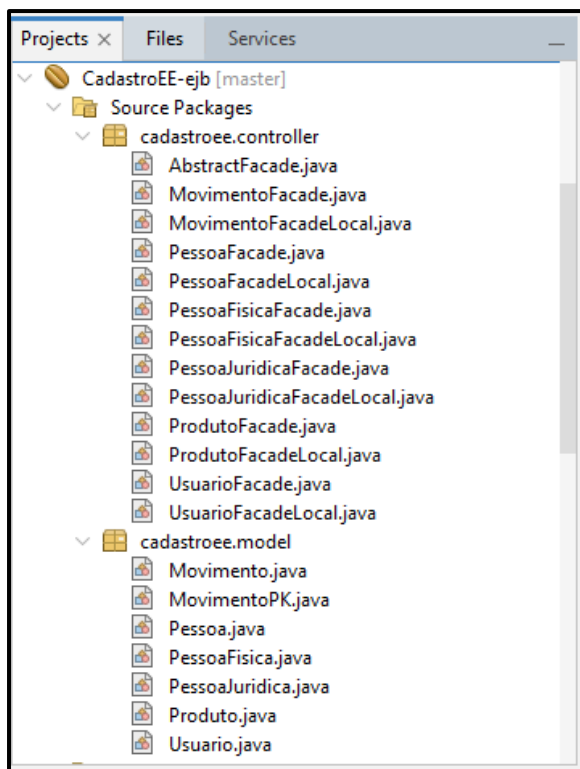


Figura 3 – Componentes EJB



Figura 4 – Resultado da execução do projeto

Análise e Conclusão

A missão mostrou como usar o NetBeans para a criação de um projeto corporativo com interface Web, no qual é possível reunir recursos necessários para desenvolver e implementar um sistema de grande porte. Isto é feito com o uso de elementos como EJBs (Enterprise Java Beans), Servlets e JSPs. Geralmente, os objetos de negócio e as classes de acesso ao banco de dados são compactados em um arquivo *jar*, enquanto a interface web é organizada em um arquivo *war*, sendo os dois arquivos compactados em um terceiro, com extensão *ear*.

Um dos desafios dos projetos com interface Web é evitar que os problemas sejam percebidos apenas no momento da execução. Nesse sentido, um dos maiores avanços do Java foi a criação do JPA (Java Persistence API), que permitiu padronizar a arquitetura dos frameworks de persistência e concentrou as configurações no arquivo *persistence.xml*. Essa tecnologia simplifica a persistência de dados em aplicações Java, padroniza o desenvolvimento e torna o código mais legível.

Outra tecnologia essencial para o desenvolvimento deste tipo de interface são as EJB's (Enterprise JavaBeans). Eles permitem a implementação da lógica de negócios no lado do servidor, encapsulando e as operações que devem ser realizadas na aplicação. Com isto é possível separar claramente as responsabilidades.

A fim de usarmos as tecnologias mencionadas acima, o NetBeans é uma ferramenta importante, pois oferece recursos que otimizam o desenvolvimento, automatizando tarefas, gerando código e proporcionando uma experiência mais eficiente, evitando a necessidade de se escrever alguns códigos manualmente.

Finalmente, ainda no campo do desenvolvimento de aplicação em Web, a missão mostrou a grande importância de outros componentes, a saber, os *Servlets*. Eles são pequenos programas em Java que agem como intermediários entre o servidor Web e as classes responsáveis pelas regras e entidades de negócios. O NetBeans oferece suporte completo para a criação e desenvolvimento de *Servlets*, facilitando a construção de aplicações Web com Java. No projeto, foi criada a classe

ServelProduto.Java e a comunicação entre este *Servlet* e o Session Beans do pool EJB foi feita através da anotação @EJB no arquivo, que permite a injeção de dependências.

Armazenado no Github: <https://github.com/altairsf/CadastroEE.git>