DGT2811 Desenvolvimento Back-End Corporativo com Java E Cloud

Thalita P. Alexandre Laurentino - 202401412341- Terceiro Semestre
Campus Estácio São José - SC

Procedimento 1 - Camadas de persistência e Controle

1. Organização de um projeto corporativo no NetBeans

O NetBeans organiza o projeto em camadas separadas:

- Persistência (JPA) para lidar com o banco.
- Negócio (EJB) para regras da aplicação.

Apresentação/Controle (Servlets/JSP) para interação com o usuário.

- Essa divisão deixa o sistema mais limpo e fácil de manter.

2. Papel do JPA e EJB em um aplicativo Web Java

O JPA cuida da comunicação com o banco de dados de forma simples, usando entidades. Os EJBs tratam as regras de negócio, garantindo segurança, transações e organização do código.

3. Como o NetBeans melhora a produtividade com JPA e EJB

O NetBeans ajuda gerando código automaticamente (entidades, Beans), configurando o servidor mais facilmente e permitindo testes rápidos. Isso reduz erros e acelera o desenvolvimento.

4. O que são Servlets e como o NetBeans ajuda

Servlets são classes que recebem as requisições Web e enviam respostas. O NetBeans facilita criando a estrutura básica automaticamente e integrando o Servlet com JPA e EJB sem dificuldade.

5. Comunicação entre Servlets e Session Beans

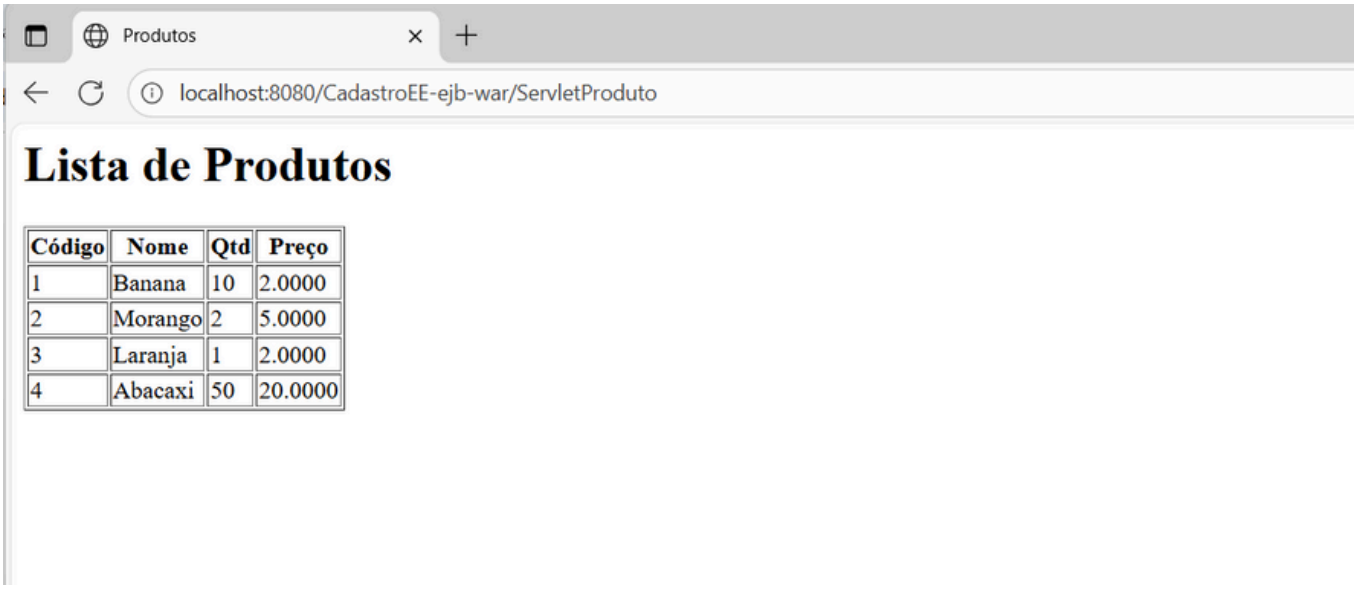
A comunicação é feita por injeção de dependência. O Servlet usa @EJB para acessar um Session Bean direto do pool e chamar seus métodos.

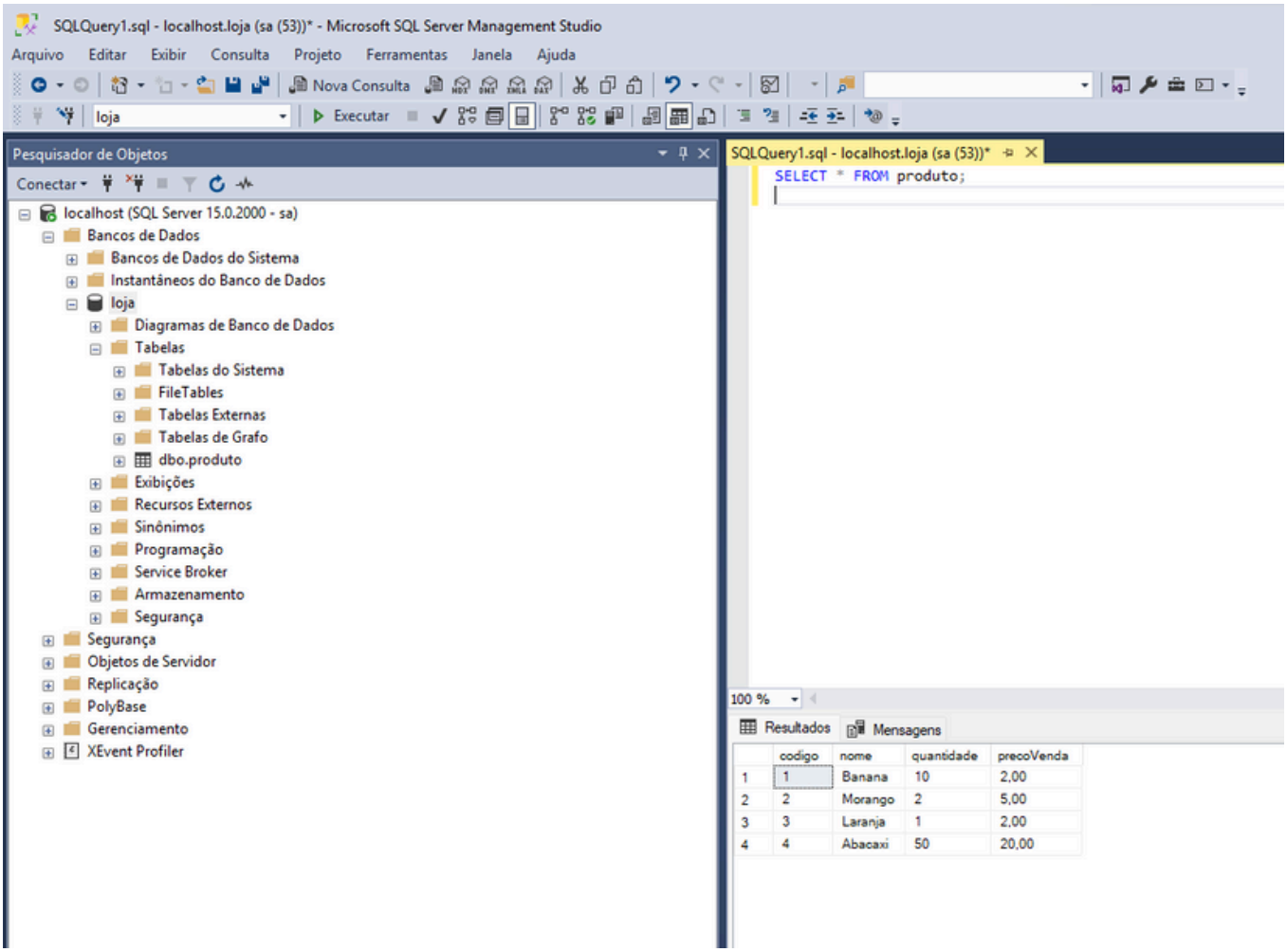
Conclusão

Para realizar essa prática, primeiro fiz um banco de dados no SQL Server Studio, onde coloquei todas as informações necessárias para o sistema. Esse banco foi utilizado diretamente no projeto, integrado ao NetBeans por meio do JPA. A partir daí, os EJBs aplicaram as regras de negócio e os Servlets fizeram a parte de controle. Trabalhar com essas tecnologias juntas deixou tudo mais organizado e ajudou a entender como cada camada se conecta. No final, o ambiente do NetBeans facilitou muito todo o processo e deixou o desenvolvimento mais simples e claro.

Repositório git:
https://github.com/ThalitaLaurentino/trabalho-DGT2811-JAVA/tree/procedimento-01

Resultado em tela:





Códigos:

... CadastroEE-ejb-ejb > SourcePackages > cadastroee.model > produto.java

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package cadastroee.model;

import jakarta.persistence.Basic;
import jakarta.persistence.Column;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import jakarta.xml.bind.annotation.XmlRootElement;
import java.io.Serializable;
import java.math.BigDecimal;
import jakarta.persistence.Entity;

/**
 *
 * @author Thalita
 */


@Entity
@Table(name = "produto")
@XmlRootElement
public class Produto implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @NotNull
    @Column(name = "codigo")
    private Integer codigo;
    @Size(max = 50)
    @Column(name = "nome")
    private String nome;
    @Column(name = "quantidade")
    private Integer quantidade;
    @Column(name = "precoVenda")
    private BigDecimal precoVenda;

    public Produto() {
    }

    public Produto(Integer codigo) {
        this.codigo = codigo;
    }

    public Integer getCodigo() {
        return codigo;
    }

    public void setCodigo(Integer codigo) {
        this.codigo = codigo;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public Integer getQuantidade() {
        return quantidade;
    }

    public void setQuantidade(Integer quantidade) {
        this.quantidade = quantidade;
    }

    public BigDecimal getPrecoVenda() {
        return precoVenda;
    }

    public void setPrecoVenda(BigDecimal precoVenda) {
        this.precoVenda = precoVenda;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (codigo != null ? codigo.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        if (!(object instanceof Produto)) {
            return false;
        }
        Produto other = (Produto) object;
        if ((this.codigo == null && other.codigo != null) || (this.codigo != null && !this.codigo.equals(other.codigo))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "cadastroee.model.Produto[ codigo=" + codigo + " ]";
    }
```

... CadastroEE-ejb-ejb > SourcePackages > cadastroee.controller > AbstractFacade.java

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package cadastroee.controller;

import jakarta.persistence.EntityManager;
import java.util.List;

/**
 *
 * @author Thalita
 */

/**
 * Classe abstrata para operações CRUD de qualquer entidade.
 *
 * @param <T> O tipo da entidade que esta fachada vai gerenciar
 */

public abstract class AbstractFacade<T> {

    private final Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public void create(T entity) {
        getEntityManager().persist(entity);
    }

    public void edit(T entity) {
        getEntityManager().merge(entity);
    }

    public void remove(T entity) {
        getEntityManager().remove(getEntityManager().merge(entity));
    }

    public T find(Object id) {
        return getEntityManager().find(entityClass, id);
    }

    public List<T> findAll() {
        jakarta.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        return getEntityManager().createQuery(cq).getResultList();
    }

    public List<T> findRange(int[] range) {
        jakarta.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        jakarta.persistence.Query q = getEntityManager().createQuery(cq);
        q.setMaxResults(range[1] - range[0] + 1);
        q.setFirstResult(range[0]);
        return q.getResultList();
    }

    public int count() {
        jakarta.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        jakarta.persistence.criteria.Root<T> rt = cq.from(entityClass);
        cq.select(getEntityManager().getCriteriaBuilder().count(rt));
        jakarta.persistence.Query q = getEntityManager().createQuery(cq);
        return ((Long) q.getSingleResult()).intValue();
    }
}
```

... CadastroEE-ejb-ejb > SourcePackages > cadastroee.controller > ProdutoFacade.java

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this licens
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package cadastroee.controller;

import cadastroee.model.Produto;
import jakarta.ejb.Stateless;
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;

/**
 *
 * @author Thalita
 */
@Stateless
public class ProdutoFacade extends AbstractFacade<Produto> implements ProdutoFacadeLocal {

    @PersistenceContext(unitName = "CadastroEE-ejb-ejbPU")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public ProdutoFacade() {
        super(Produto.class);
    }

}
```

... CadastroEE-ejb-ejb > SourcePackages > cadastroee.controller > ProdutoFacadeLocal.java

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this licens
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
 */
package cadastroee.controller;

import cadastroee.model.Produto;
import jakarta.ejb.Local;
import java.util.List;

/**
 *
 * @author Thalita
 */
@Local
public interface ProdutoFacadeLocal {

    void create(Produto produto);

    void edit(Produto produto);

    void remove(Produto produto);

    Produto find(Object id);

    List<Produto> findAll();

    List<Produto> findRange(int[] range);

    int count();

}
```

... CadastroEE-ejb-war > WebPages >WEB-INF > web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="4.0"
         xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                             http://xmlns.jcp.org/xml/ns/javaee/web-app_4_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>

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

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

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

</web-app>
```