

## Faculdade Estácio - Campus - Monte Castelo

**Curso:** Desenvolvimento Full Stack

**Disciplina:** Vamos Integrar Sistemas

**Número da Turma:** RPG0017

**Semestre Letivo:** 3

**Integrante:** Lucas de Oliveira dos Santos

**Repositório:** <https://github.com/Lucasph3/mundo3-missao4>

### 1. Título da Prática:

#### **RPG0017 - Vamos integrar sistemas**

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

### 2. 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.

### 3. Códigos do roteiro:

## Arquivo: [model/Movimento.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 java.io.Serializable;
import jakarta.persistence.Basic;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;
import jakarta.persistence.Table;

/**
 *
 * @author Mari
 */
@Entity
@Table(name = "Movimento")
@NamedQueries({
    @NamedQuery(name = "Movimento.findAll", query = "SELECT m FROM Movimento m"),
    @NamedQuery(name = "Movimento.findByIdMovimento", query = "SELECT m FROM Movimento m WHERE m.idMovimento = :idMovimento"),
    @NamedQuery(name = "Movimento.findByQuantidade", query = "SELECT m FROM Movimento m WHERE m.quantidade = :quantidade"),
    @NamedQuery(name = "Movimento.findByTipo", query = "SELECT m FROM Movimento m WHERE m.tipo = :tipo"),
    @NamedQuery(name = "Movimento.findByValorUnitario", query = "SELECT m FROM Movimento m WHERE m.valorUnitario = :valorUnitario"))
public class Movimento implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idMovimento")
    private Integer idMovimento;
    @Basic(optional = false)
    @Column(name = "quantidade")
    private int quantidade;
    @Basic(optional = false)
    @Column(name = "tipo")
    private Character tipo;
    @Basic(optional = false)
    @Column(name = "valorUnitario")
```

```

private long valorUnitario;
@JoinColumn(name = "Pessoa_idPessoa", referencedColumnName = "idPessoa")
@ManyToOne(optional = false)
private Pessoa pessoaidPessoa;
@JoinColumn(name = "Produto_idProduto", referencedColumnName = "idProduto")
@ManyToOne(optional = false)
private Produto produtoidProduto;
@JoinColumn(name = "Usuario_idUsuario", referencedColumnName = "idUsuario")
@ManyToOne(optional = false)
private Usuario usuarioidUsuario;

public Movimento() {
}

public Movimento(Integer idMovimento) {
    this.idMovimento = idMovimento;
}

public Movimento(Integer idMovimento, int quantidade, Character tipo, long valorUnitario) {
    this.idMovimento = idMovimento;
    this.quantidade = quantidade;
    this.tipo = tipo;
    this.valorUnitario = valorUnitario;
}

public Integer getIdMovimento() {
    return idMovimento;
}

public void setIdMovimento(Integer idMovimento) {
    this.idMovimento = idMovimento;
}

public int getQuantidade() {
    return quantidade;
}

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

public Character getTipo() {
    return tipo;
}

public void setTipo(Character tipo) {
    this.tipo = tipo;
}

public long getValorUnitario() {
    return valorUnitario;
}

public void setValorUnitario(long valorUnitario) {
    this.valorUnitario = valorUnitario;
}

```

```

public Pessoa getPessoaidPessoa() {
    return pessoaidPessoa;
}

public void setPessoaidPessoa(Pessoa pessoaidPessoa) {
    this.pessoaidPessoa = pessoaidPessoa;
}

public Produto getProdutoidProduto() {
    return produtoidProduto;
}

public void setProdutoidProduto(Produto produtoidProduto) {
    this.produtoidProduto = produtoidProduto;
}

public Usuario getUsuarioidUsuario() {
    return usuarioidUsuario;
}

public void setUsuarioidUsuario(Usuario usuarioidUsuario) {
    this.usuarioidUsuario = usuarioidUsuario;
}

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

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Movimento)) {
        return false;
    }
    Movimento other = (Movimento) object;
    if ((this.idMovimento == null && other.idMovimento != null) || (this.idMovimento != null &&
!this.idMovimento.equals(other.idMovimento))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "cadastroee.model.Movimento[ idMovimento=" + idMovimento + " ]";
}
}

```

**Arquivo:** [model/Pessoa.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 java.io.Serializable;
import java.util.Collection;
import jakarta.persistence.Basic;
import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;
import jakarta.persistence.OneToOne;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;

/**
 *
 * @author Mari
 */
@Entity
@Table(name = "Pessoa")
@NamedQueries({
    @NamedQuery(name = "Pessoa.findAll", query = "SELECT p FROM Pessoa p"),
    @NamedQuery(name = "Pessoa.findByIdPessoa", query = "SELECT p FROM Pessoa p WHERE p.idPessoa = :idPessoa"),
    @NamedQuery(name = "Pessoa.findByName", query = "SELECT p FROM Pessoa p WHERE p.nome = :nome"),
    @NamedQuery(name = "Pessoa.findByLogradouro", query = "SELECT p FROM Pessoa p WHERE p.logradouro = :logradouro"),

```

```

    @NamedQuery(name = "Pessoa.findByCidade", query = "SELECT p FROM Pessoa p WHERE p.cidade = :cidade"),
    @NamedQuery(name = "Pessoa.findByEstado", query = "SELECT p FROM Pessoa p WHERE p.estado = :estado"),
    @NamedQuery(name = "Pessoa.findByTelefone", query = "SELECT p FROM Pessoa p WHERE p.telefone = :telefone"),
    @NamedQuery(name = "Pessoa.findByEmail", query = "SELECT p FROM Pessoa p WHERE p.email = :email"))} public
class Pessoa implements Serializable {

```

```

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "idPessoa")
    private Integer idPessoa;
    @Basic(optional = false)
    @Column(name = "nome")
    private String nome;
    @Column(name = "logradouro")
    private String logradouro;
    @Column(name = "cidade")
    private String cidade;
    @Column(name = "estado")
    private String estado;
    @Column(name = "telefone")
    private String telefone;

```

```
@Column(name = "email")
private String email;
@OneToOne(cascade = CascadeType.ALL, mappedBy = "pessoa")
private PessoaJuridica pessoaJuridica;
@OneToOne(cascade = CascadeType.ALL, mappedBy = "pessoa")
private PessoaFisica pessoaFisica;
@OneToMany(cascade = CascadeType.ALL, mappedBy = "pessoaidPessoa")
private Collection<Movimento> movimentoCollection;

public Pessoa() {
}

public Pessoa(Integer idPessoa) {
    this.idPessoa = idPessoa;
}

public Pessoa(Integer idPessoa, String nome) {
    this.idPessoa = idPessoa;
    this.nome = nome;
}

public Integer getIdPessoa() {
    return idPessoa;
}

public void setIdPessoa(Integer idPessoa) {
    this.idPessoa = idPessoa;
}

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

public String getLogradouro() {
    return logradouro;
}

public void setLogradouro(String logradouro) {
    this.logradouro = logradouro;
}

public String getCidade() {
    return cidade;
}

public void setCidade(String cidade) {
    this.cidade = cidade;
}

public String getEstado() {
    return estado;
}
```

```
public void setEstado(String estado) {
    this.estado = estado;
}

public String getTelefone() {
    return telefone;
}

public void setTelefone(String telefone) {
    this.telefone = telefone;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public PessoaJuridica getPessoaJuridica() {
    return pessoaJuridica;
}

public void setPessoaJuridica(PessoaJuridica pessoaJuridica) {
    this.pessoaJuridica = pessoaJuridica;
}

public PessoaFisica getPessoaFisica() {
    return pessoaFisica;
}
```

```

public void setPessoaFisica(PessoaFisica pessoaFisica) {
    this.pessoaFisica = pessoaFisica;
}

public Collection<Movimento> getMovimentoCollection() {
    return movimentoCollection;
}

public void setMovimentoCollection(Collection<Movimento> movimentoCollection) {
    this.movimentoCollection = movimentoCollection;
}

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

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Pessoa)) {
        return false;
    }
    Pessoa other = (Pessoa) object;
    if ((this.idPessoa == null && other.idPessoa != null) || (this.idPessoa != null &&
!this.idPessoa.equals(other.idPessoa))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "cadastroee.model.Pessoa[ idPessoa=" + idPessoa + " ]";
}
}

```

**Arquivo:** [model/PessoaFisica.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 java.io.Serializable;
import jakarta.persistence.Basic;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
```

```
import jakarta.persistence.JoinColumn;
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;
```

```
/**
```

```
*
```

```
* @author Mari
```

```
*/
```

```
@Entity
```

```
@Table(name = "PessoaFisica")
```

```
@NamedQueries({
```

```
    @NamedQuery(name = "PessoaFisica.findAll", query = "SELECT p FROM PessoaFisica p"), @NamedQuery(name =
    "PessoaFisica.findByIdPessoa", query = "SELECT p FROM PessoaFisica p WHERE p.idPessoa = :idPessoa"),
```

```
    @NamedQuery(name = "PessoaFisica.findByCpf", query = "SELECT p FROM PessoaFisica p WHERE p.cpf = :cpf"))})
```

```
public class PessoaFisica implements Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    @Id
```

```
    @Basic(optional = false)
```

```
    @Column(name = "idPessoa")
```

```
    private Integer idPessoa;
```

```
    @Basic(optional = false)
```

```
    @Column(name = "CPF")
```

```
    private String cpf;
```

```
    @JoinColumn(name = "idPessoa", referencedColumnName = "idPessoa", insertable = false, updatable = false)
```

```
    @OneToOne(optional = false)
```

```
    private Pessoa pessoa;
```

```
    public PessoaFisica() {
```

```
    }
```

```
    public PessoaFisica(Integer idPessoa) {
```

```
        this.idPessoa = idPessoa;
```

```
    }
```

```
    public PessoaFisica(Integer idPessoa, String cpf) {
```

```
        this.idPessoa = idPessoa;
```

```
        this.cpf = cpf;
```

```
    }
```

```

public Integer getIdPessoa() {
    return idPessoa;
}

public void setIdPessoa(Integer idPessoa) {
    this.idPessoa = idPessoa;
}

public String getCpf() {
    return cpf;
}

public void setCpf(String cpf) {
    this.cpf = cpf;
}

public Pessoa getPessoa() {
    return pessoa;
}

public void setPessoa(Pessoa pessoa) {
    this.pessoa = pessoa;
}

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

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof PessoaFisica)) {
        return false;
    }
    PessoaFisica other = (PessoaFisica) object;
    if ((this.idPessoa == null && other.idPessoa != null) || (this.idPessoa != null &&
!this.idPessoa.equals(other.idPessoa))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "cadastroee.model.PessoaFisica[ idPessoa=" + idPessoa + " ]";
}
}

```

## Arquivo: [model/PessoaJuridica.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 java.io.Serializable;
import jakarta.persistence.Basic;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;

/**
 *
 * @author Mari
 */
@Entity
@Table(name = "PessoaJuridica")
@NamedQueries({
    @NamedQuery(name = "PessoaJuridica.findAll", query = "SELECT p FROM PessoaJuridica p"),
    @NamedQuery(name = "PessoaJuridica.findByIdPessoa", query = "SELECT p FROM PessoaJuridica p WHERE p.idPessoa = :idPessoa"),
    @NamedQuery(name = "PessoaJuridica.findByCnpj", query = "SELECT p FROM PessoaJuridica p WHERE p.cnpj = :cnpj")})
public class PessoaJuridica implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Basic(optional = false)
    @Column(name = "idPessoa")
    private Integer idPessoa;
    @Basic(optional = false)
    @Column(name = "CNPJ")
    private String cnpj;
    @JoinColumn(name = "idPessoa", referencedColumnName = "idPessoa", insertable = false, updatable = false)
    @OneToOne(optional = false)
    private Pessoa pessoa;

    public PessoaJuridica() {
    }

    public PessoaJuridica(Integer idPessoa) {
        this.idPessoa = idPessoa;
    }

    public PessoaJuridica(Integer idPessoa, String cnpj) {
        this.idPessoa = idPessoa;
    }
}
```

```

        this.cnpj = cnpj;
    }

    public Integer getIdPessoa() {
        return idPessoa;
    }

    public void setIdPessoa(Integer idPessoa) {
        this.idPessoa = idPessoa;
    }

    public String getCnpj() {
        return cnpj;
    }

    public void setCnpj(String cnpj) {
        this.cnpj = cnpj;
    }

    public Pessoa getPessoa() {
        return pessoa;
    }

    public void setPessoa(Pessoa pessoa) {
        this.pessoa = pessoa;
    }

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

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof PessoaJuridica)) {
            return false;
        }
        PessoaJuridica other = (PessoaJuridica) object;
        if ((this.idPessoa == null && other.idPessoa != null) || (this.idPessoa != null &&
!this.idPessoa.equals(other.idPessoa))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "cadastroee.model.PessoaJuridica[ idPessoa=" + idPessoa + " ]";
    }
}

```

## Arquivo: [model/Produto.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 java.io.Serializable;
import java.util.Collection;
import jakarta.persistence.Basic;
import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;
import jakarta.persistence.OneToMany;
import jakarta.persistence.Table;

/**
 *
 * @author Mari
 */
@Entity
@Table(name = "Produto")
@NamedQueries({
    @NamedQuery(name = "Produto.findAll", query = "SELECT p FROM Produto p"),
    @NamedQuery(name = "Produto.findByIdProduto", query = "SELECT p FROM Produto p WHERE p.idProduto = :idProduto"),
    @NamedQuery(name = "Produto.findByName", query = "SELECT p FROM Produto p WHERE p.nome = :nome"),
    @NamedQuery(name = "Produto.findByQuantidade", query = "SELECT p FROM Produto p WHERE p.quantidade = :quantidade"),
    @NamedQuery(name = "Produto.findByPrecoVenda", query = "SELECT p FROM Produto p WHERE p.precoVenda = :precoVenda")})
public class Produto implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idProduto")
    private Integer idProduto;
    @Basic(optional = false)
    @Column(name = "nome")
    private String nome;
    @Basic(optional = false)
    @Column(name = "quantidade")
    private int quantidade;
```

```
@Basic(optional = false)
@Column(name = "precoVenda")
private float precoVenda;
@OneToMany(cascade = CascadeType.ALL, mappedBy = "produtoIdProduto")
private Collection<Movimento> movimentoCollection;

public Produto() {
}

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

public Produto(Integer idProduto, String nome, int quantidade, long precoVenda) {
    this.idProduto = idProduto;
    this.nome = nome;
    this.quantidade = quantidade;
    this.precoVenda = precoVenda;
}

public Integer getIdProduto() {
    return idProduto;
}

public void setIdProduto(Integer idProduto) {
    this.idProduto = idProduto;
}

public String getNome() {
    return nome;
}

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

public int getQuantidade() {
    return quantidade;
}

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

public float getPrecoVenda() {
    return precoVenda;
}

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

public Collection<Movimento> getMovimentoCollection() {
    return movimentoCollection;
}
```

```

public void setMovimentoCollection(Collection<Movimento> movimentoCollection) {
    this.movimentoCollection = movimentoCollection;
}

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

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not
    set if (!(object instanceof Produto)) {
        return false;
    }
    Produto other = (Produto) object;
    if ((this.idProduto == null && other.idProduto != null) || (this.idProduto != null
    && !this.idProduto.equals(other.idProduto))) {
        return false;
    }
    return true;
}

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

```

**Arquivo:** [model/Usuario.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 java.io.Serializable;
import java.util.Collection;
import jakarta.persistence.Basic;
import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;
import jakarta.persistence.OneToMany;
import jakarta.persistence.Table;

/**
 *
 * @author Mari
 */
@Entity
@Table(name = "Usuario")
@NamedQueries({
    @NamedQuery(name = "Usuario.findAll", query = "SELECT u FROM Usuario u"),
    @NamedQuery(name = "Usuario.findByIdUsuario", query = "SELECT u FROM Usuario u WHERE u.idUsuario = :idUsuario"),
    @NamedQuery(name = "Usuario.findByLogin", query = "SELECT u FROM Usuario u WHERE u.login = :login"),
    @NamedQuery(name = "Usuario.findBySenha", query = "SELECT u FROM Usuario u WHERE u.senha = :senha")})
public class Usuario implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id

```

```

@GeneratedValue(strategy = GenerationType.IDENTITY)
@Basic(optional = false)
@Column(name = "idUsuario")
private Integer idUsuario;
@Column(name = "login")
private String login;
@Column(name = "senha")
private String senha;
@OneToMany(cascade = CascadeType.ALL, mappedBy = "usuarioidUsuario")
private Collection<Movimento> movimentoCollection;

public Usuario() {
}

public Usuario(Integer idUsuario) {
    this.idUsuario = idUsuario;
}

```



```
public Integer getIdUsuario() {
    return idUsuario;
}

public void setIdUsuario(Integer idUsuario) {
    this.idUsuario = idUsuario;
}

public String getLogin() {
    return login;
}

public void setLogin(String login) {
    this.login = login;
}

public String getSenha() {
    return senha;
}

public void setSenha(String senha) {
    this.senha = senha;
}

public Collection<Movimento> getMovimentoCollection() {
    return movimentoCollection;
}

public void setMovimentoCollection(Collection<Movimento> movimentoCollection) {
    this.movimentoCollection = movimentoCollection;
}

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

```

    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Usuario)) {
            return false;
        }
        Usuario other = (Usuario) object;
        if ((this.idUsuario == null && other.idUsuario != null) || (this.idUsuario != null
        && !this.idUsuario.equals(other.idUsuario))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "cadastroee.model.Usuario[ idUsuario=" + idUsuario + " ]";
    }
}

```

**Arquivo:** [controller/AbstractFacade.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 java.util.List;
import jakarta.persistence.EntityManager;

/**
 *
 * @author Mari
 * @param <T>
 */
public abstract class AbstractFacade<T> {

    private 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();
}
}

```

**Arquivo:** [controller/MovimentoFacade.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 cadastroee.model.Movimento;
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;

/**
 *
 */

```

```
* @author Mari
*/

@jakarta.ejb.Stateless
public class MovimentoFacade extends AbstractFacade<Movimento> implements MovimentoFacadeLocal {

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

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

    public MovimentoFacade() {
        super(Movimento.class);
    }

}
```

**Arquivo:** [controller/MovimentoFacadeLocal.java](#)

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

import cadastroee.model.Movimento;
import java.util.List;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Local
public interface MovimentoFacadeLocal {

    void create(Movimento movimento);

    void edit(Movimento movimento);

    void remove(Movimento movimento);

    Movimento find(Object id);

    List<Movimento> findAll();

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

    int count();

}
```

**Arquivo:** [controller/PessoaFacade.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 cadastroee.model.Pessoa;
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Stateless
public class PessoaFacade extends AbstractFacade<Pessoa> implements PessoaFacadeLocal {

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

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

    public PessoaFacade() {
        super(Pessoa.class);
    }

}

```

**Arquivo:** [controller/PessoaFacadeLocal.java](#)

```

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

import cadastroee.model.Pessoa;
import java.util.List;

/**
 *
 * @author Mari
 */

```

```

@jakarta.ejb.Local
public interface PessoaFacadeLocal {

    void create(Pessoa pessoa);

    void edit(Pessoa pessoa);

    void remove(Pessoa pessoa);

    Pessoa find(Object id);

    List<Pessoa> findAll();

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

    int count();

}

```

## Arquivo: [controller/PessoaFisicaFacade.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 cadastroee.model.PessoaFisica;
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Stateless
public class PessoaFisicaFacade extends AbstractFacade<PessoaFisica> implements PessoaFisicaFacadeLocal {

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

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

    public PessoaFisicaFacade() {
        super(PessoaFisica.class);
    }

}

```



```
}
```

## Arquivo: [controller/PessoaFisicaFacadeLocal.java](#)

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

import cadastroee.model.PessoaFisica;
import java.util.List;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Local
public interface PessoaFisicaFacadeLocal {

    void create(PessoaFisica pessoaFisica);

    void edit(PessoaFisica pessoaFisica);

    void remove(PessoaFisica pessoaFisica);

    PessoaFisica find(Object id);

    List<PessoaFisica> findAll();

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

    int count();

}
```

## Arquivo: [controller/PessoaJuridicaFacade.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 cadastroee.model.PessoaJuridica;
```

```

import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Stateless
public class PessoaJuridicaFacade extends AbstractFacade<PessoaJuridica> implements
PessoaJuridicaFacadeLocal {

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

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

    public PessoaJuridicaFacade() {
        super(PessoaJuridica.class);
    }

}

```

**Arquivo:** [controller/PessoaJuridicaFacadeLocal.java](#)

```

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

import cadastroee.model.PessoaJuridica;
import java.util.List;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Local
public interface PessoaJuridicaFacadeLocal {

    void create(PessoaJuridica pessoaJuridica);

    void edit(PessoaJuridica pessoaJuridica);

    void remove(PessoaJuridica pessoaJuridica);

    PessoaJuridica find(Object id);
}

```

```
List<PessoaJuridica> findAll();

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

int count();

}
```

## Arquivo: [controller/ProdutoFacade.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 cadastroee.model.Produto;
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Stateless
public class ProdutoFacade extends AbstractFacade<Produto> implements ProdutoFacadeLocal {

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

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

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

}
```

## Arquivo: [controller/ProdutoFacadeLocal.java](#)

```
/*
```

```

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

import cadastroee.model.Produto;
import java.util.List;

/**
 *
 * @author Mari
 */
@jakarta.ejb.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();

}

```

## Arquivo: [controller/UsuarioFacade.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 cadastroee.model.Usuario;
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Stateless
public class UsuarioFacade extends AbstractFacade<Usuario> implements UsuarioFacadeLocal {

```

```

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

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

public UsuarioFacade() {
    super(Usuario.class);
}
}
}

```

## Arquivo: [controller/UsuarioFacadeLocal.java](#)

```

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

import cadastroee.model.Usuario;
import java.util.List;

/**
 *
 * @author Mari
 */
@jakarta.ejb.Local
public interface UsuarioFacadeLocal {

    void create(Usuario usuario);

    void edit(Usuario usuario);

    void remove(Usuario usuario);

    Usuario find(Object id);

    List<Usuario> findAll();

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

    int count();

}

```

## Arquivo: [conf/persistence.xml](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/ persistence
http://java.sun.com/xml/ns/persistence/ persistence_1_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>
```

## Arquivo: [servlets/ServletProduto.java](#)

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

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;

import jakarta.ejb.EJB;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import cadastroee.controller.ProdutoFacadeLocal;
import cadastroee.model.Produto;

/**
 *
 * @author Mari
 */
public class ServletProduto extends HttpServlet {
    @EJB
    private ProdutoFacadeLocal facade;

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
}
```

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet ServletProduto</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet ServletProduto at " + request.getContextPath() + "</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

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

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h1>Lista de Produtos</h1>");
    out.println("<ul>");

    for (Produto produto : produtos) {
        out.println("<li>" + produto.getNome() + "</li>");
    }

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

/**
 * Handles the HTTP <code>POST</code> method.
 *

```

```

    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

## Arquivo: [WEB-INF/web.xml](#)

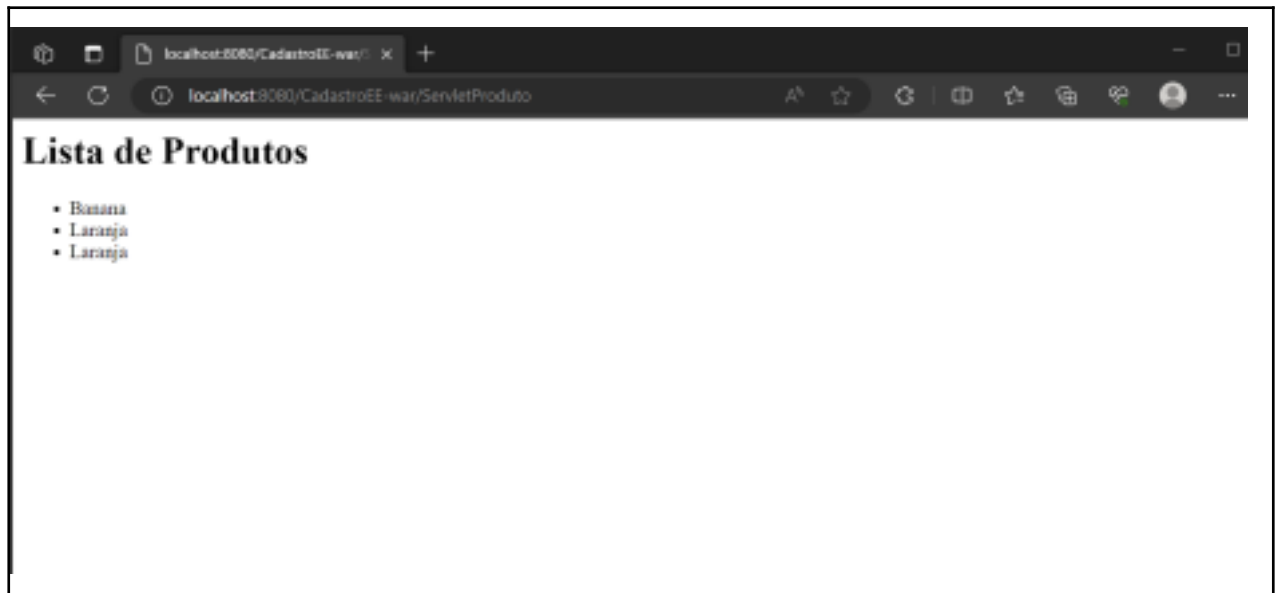
```

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

```

## 4. Resultados da execução dos códigos





## 5. Análise e Conclusão

a) Como é organizado um projeto corporativo no NetBeans?

**Resposta:** O projeto é dividido em 3 sub-projetos: CadastroEE, CadastroEE-ejb e CadastroEE-war, sendo o CadastroEE contendo referência aos demais projetos, o CadastroEE-ejb, responsável pela lógica de negócios, contendo os pacotes **cadastroee.model** e **cadastroee.controller** e sem dependências de outros projetos e o projeto CadastroEE-war, responsável pela parte web, contendo os pacotes **cadastroee.servlets** e com referência ao projeto CadastroEE-ejb.

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

**Resposta:** O JPA (Java Persistence API) é usado para persistir objetos Java em um banco de dados, permitindo a utilização de objetos para representar os dados do banco de dados. O EJB (Enterprise JavaBeans) fornece um modelo de componentes para a construção de aplicativos empresariais distribuídos, permitindo a criação de componentes reutilizáveis que podem ser implantados em diferentes servidores. Juntos eles fornecem uma estrutura para a construção de aplicativos empresariais escaláveis, permitindo os desenvolvedores concentrarem na lógica do aplicativo, em vez de se preocupar com detalhes de baixo nível, como gerenciamento de transações e conexões com o banco de dados.

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

**Resposta:** O NetBeans oferece várias ferramentas que agilizam o desenvolvimento de JPA e EJB, como por exemplo a Criação automática de entidades JPA e o Session Beans for Entity Classes, que gera códigos baseado na estrutura da base de dados de forma automatizada.

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

**Resposta:** Os Servlets são componentes Java usados na criação de aplicativos Web dinâmicos. Eles são executados no lado do servidor e são responsáveis por processar solicitações HTTP. Através das ferramentas que o NetBeans oferecem, como por exemplo, conexão com o Banco de Dados e gerenciamento de servidores como o GlassFish, e geração de código, o desenvolvimento de projetos Web acaba sendo mais eficiente.

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

**Resposta:** Os Session Beans são gerenciados pelo container EJB e podem ser configurados para serem armazenados em um pool de instâncias. Quando um Servlet chama um método em um Session Bean, o container EJB fornece uma instância do Session Bean a partir do pool, se disponível. Depois que o método é concluído, a instância do Session Bean é devolvida ao pool para ser reutilizada por outros Servlets.