



Jonathan Domingos Carneiro da Silva
matricula: 2023.04.12744-1
3274 POLO CENTRO - ITAITINGA – CE

Nível 4: Implementação de sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE.

– DESENVOLVIMENTO FULL STACK 2023.1 – 3º Semestre – 2024.2

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.
7. Lidar com contextos reais de aplicação.

Introdução

O Bootstrap é um framework CSS popular que facilita a criação de interfaces web responsivas e consistentes. Neste relatório, analisaremos como ele se encaixa no desenvolvimento de um sistema cadastral web, explorando seus benefícios e como ele se integra com as tecnologias Java utilizadas na prática.

Utilização do Bootstrap no Sistema Cadastral

O Bootstrap oferece diversos recursos que otimizam o desenvolvimento do sistema cadastral:

- **Componentes prontos:** Botões, menus, formulários, modais e outros elementos pré-construídos agilizam a criação da interface.
- **Grade responsiva:** Garante que o layout se adapte automaticamente a diferentes dispositivos, desde desktops até smartphones.
- **Estilos personalizáveis:** Classes CSS permitem personalizar a aparência dos componentes de acordo com a identidade visual do sistema.
- **JavaScript opcional:** Possibilita interações dinâmicas com os componentes do Bootstrap, como validação de formulários e animações.

Independência Estrutural do HTML

O Bootstrap promove a separação entre a estrutura (HTML) e a apresentação (CSS) da página, proporcionando:



- **HTML semântico:** O foco na estrutura e no conteúdo facilita a leitura e manutenção do código.
- **Flexibilidade:** Permite modificar a aparência sem afetar a estrutura da página.
- **Reutilização:** Componentes do Bootstrap podem ser reutilizados em diferentes partes do sistema.

Bootstrap e Responsividade

A responsividade é crucial para garantir uma boa experiência do usuário em diversos dispositivos. O Bootstrap oferece recursos para criar layouts responsivos:

- **Grade fluida:** A grade se ajusta automaticamente à largura da tela, garantindo que os elementos sejam exibidos corretamente.
- **Componentes responsivos:** Menus, botões e outros elementos se adaptam à diferentes tamanhos de tela.
- **Media queries:** Permitem definir estilos específicos para diferentes faixas de largura de tela, otimizando a visualização em cada dispositivo.

Integração com Tecnologias Java

O Bootstrap se integra perfeitamente com as tecnologias Java utilizadas na prática:

- **Servlets e JSPs:** A interface web desenvolvida com servlets e JSPs pode ser estilizada com o Bootstrap, criando um visual moderno e responsivo.
- **EJBs:** As regras de negócio encapsuladas em EJBs podem ser utilizadas para validar dados de formulários e realizar outras operações no back-end do sistema.
- **JPA:** A persistência de dados com JPA pode ser integrada com os componentes do Bootstrap para exibir e editar informações do banco de dados na interface web.

Conclusão

O Bootstrap é uma ferramenta valiosa para o desenvolvimento de sistemas cadastrais web em Java. Ele facilita a criação de interfaces responsivas, melhora a experiência do usuário e se integra perfeitamente com as tecnologias Java utilizadas na prática. Ao dominar o Bootstrap, os alunos estarão aptos a criar sistemas web modernos, escaláveis e com alta usabilidade.

Considerações Adicionais

- O Bootstrap oferece uma ampla comunidade e documentação extensa, facilitando o aprendizado e a resolução de problemas.
- Diversos tutoriais e cursos online estão disponíveis para auxiliar no aprendizado do Bootstrap.
- É importante manter o Bootstrap atualizado para aproveitar os novos recursos e correções de bugs.

Recomendações



- Explorar a documentação oficial do Bootstrap para conhecer todos os seus recursos: <https://getbootstrap.com/>
- Praticar a utilização do Bootstrap em projetos pessoais para desenvolver familiaridade com o framework.
- Participar da comunidade online do Bootstrap para obter ajuda e trocar experiências com outros desenvolvedores.

Conclusão

O Bootstrap é uma ferramenta essencial para o desenvolvimento de interfaces web modernas e responsivas. Ao integrar o Bootstrap com as tecnologias Java na prática proposta, os alunos estarão criando um sistema cadastral web completo e profissional, aprimorando suas habilidades para lidar com contextos reais de aplicação.

Pessoa.java

```
package cadastroee.model;
```

```
import java.io.Serializable;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;
```

```
@Entity
```

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

```
@NamedQueries({
```

```
    @NamedQuery(name = "Pessoa.findAll", query = "SELECT pes FROM Pessoa pes"),
```

```
    @NamedQuery(name = "Pessoa.findByIdPessoa", query = "SELECT pes FROM Pessoa pes WHERE  
pes.id_Pessoa = :id_Pessoa"),
```

```
    @NamedQuery(name = "Pessoa.findByName", query = "SELECT pes FROM Pessoa pes WHERE pes.nome  
= :nome"),
```

```
    @NamedQuery(name = "Pessoa.findByLogradouro", query = "SELECT pes FROM Pessoa pes WHERE  
pes.logradouro = :logradouro"),
```

```
    @NamedQuery(name = "Pessoa.findByCidade", query = "SELECT pes FROM Pessoa pes WHERE  
pes.cidade = :cidade"),
```



```
@NamedQuery(name = "Pessoa.findByEstado", query = "SELECT pes FROM Pessoa pes WHERE pes.estado  
= :estado"),  
@NamedQuery(name = "Pessoa.findByTelefone", query = "SELECT pes FROM Pessoa pes WHERE  
pes.telefone = :telefone"),  
@NamedQuery(name = "Pessoa.findByEmail", query = "SELECT pes FROM Pessoa pes WHERE pes.email  
= :email"),  
@NamedQuery(name = "Pessoa.findByTpPessoa", query = "SELECT pes FROM Pessoa pes WHERE  
pes.tpPessoa = :tpPessoa"))}
```

```
public class Pessoa implements Serializable {  
  
    private static final long serialVersionUID = 1L;  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Basic(optional = false)  
    @Column(name = "id_Pessoa")  
    private Integer id_Pessoa;  
    @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;  
    @Column(name = "tpPessoa")  
    private String tpPessoa;  
  
    public Pessoa() {  
    }  
  
    public Pessoa(Integer id_Pessoa) {  
        this.id_Pessoa = id_Pessoa;  
    }  
  
    public Integer getId_Pessoa() {  
        return id_Pessoa;  
    }  
  
    public void setIdPessoa(Integer idPessoa) {  
        this.id_Pessoa = id_Pessoa;  
    }  
}
```



```
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 String getTpPessoa() {
    return tpPessoa;
}

public void setTpPessoa(String tpPessoa) {
    this.tpPessoa = tpPessoa;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (id_Pessoa != null ? id_Pessoa.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;
    return !((this.id_Pessoa == null && other.id_Pessoa != null) || (this.id_Pessoa != null &&
!this.id_Pessoa.equals(other.id_Pessoa)));
}

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

PessoaFisica.java

```
package cadastroee.model;

import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.Column;
```



```
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name = "PessoaFisica")
@NamedQueries({
    @NamedQuery(name = "PessoaFisica.findAll", query = "SELECT pf FROM Pessoa_Fisica pf"),
    @NamedQuery(name = "PessoaFisica.findByIdPessoaFisica", query = "SELECT pf FROM PessoaFisica pf
WHERE pf.id_PessoaFisica = :id_PessoaFisica"),
    @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 = "id_PessoaFisica")
    private Integer id_PessoaFisica;
    @Column(name = "cpf")
    private String cpf;
    @OneToMany(mappedBy = "id_Comprador")
    private Collection<MovimentoVenda> movimentoVendaCollection;

    public PessoaFisica() {
    }

    public PessoaFisica(Integer id_PessoaFisica) {
        this.id_PessoaFisica = id_PessoaFisica;
    }

    public Integer getId_PessoaFisica() {
        return id_PessoaFisica;
    }

    public void setId_PessoaFisica(Integer id_PessoaFisica) {
        this.id_PessoaFisica = id_PessoaFisica;
    }

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
```



```
        this.cpf = cpf;
    }

    public Collection<MovimentoVenda> getMovimentoVendaCollection() {
        return movimentoVendaCollection;
    }

    public void setMovimentoVendaCollection(Collection<MovimentoVenda> movimentoVendaCollection) {
        this.movimentoVendaCollection = movimentoVendaCollection;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id_PessoaFisica != null ? id_PessoaFisica.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;
        return !((this.id_PessoaFisica == null && other.id_PessoaFisica != null) || (this.id_PessoaFisica != null &&
!this.id_PessoaFisica.equals(other.id_PessoaFisica)));
    }

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

PessoaJuridica.java

```
package cadastroee.model;

import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Basic;
```




```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "PessoaJuridica")
@NamedQueries({
    @NamedQuery(name = "PessoaJuridica.findAll", query = "SELECT p FROM PessoaJuridica p"),
    @NamedQuery(name = "PessoaJuridica.findById_PessoaJuridica", query = "SELECT p FROM
PessoaJuridica p WHERE p.id_PessoaJuridica = :id_PessoaJuridica"),
    @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 = "id_PessoaJuridica")
    private Integer id_PessoaJuridica;
    @Column(name = "cnpj")
    private String cnpj;
    @OneToMany(mappedBy = "id_Fornecedor")
    private Collection<MovimentoCompra> movimentoCompraCollection;

    public PessoaJuridica() {
    }

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

    public Integer getId_PessoaJuridica(Integer id_PessoaJuridica) {
        return id_PessoaJuridica;
    }

    public void setId_PessoaJuridica(Integer id_PessoaJuridica) {
        this.id_PessoaJuridica = id_PessoaJuridica;
    }

    public String getCnpj() {
```



```
        return cnpj;
    }

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

    public Collection<MovimentoCompra> getMovimentoCompraCollection() {
        return movimentoCompraCollection;
    }

    public void setMovimentoCompraCollection(Collection<MovimentoCompra> movimentoCompraCollection)
    {
        this.movimentoCompraCollection = movimentoCompraCollection;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id_PessoaJuridica != null ? id_PessoaJuridica.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;
        } else {
            PessoaJuridica other = (PessoaJuridica) object;
            return !((this.id_PessoaJuridica == null && other.id_PessoaJuridica != null) || (this.id_PessoaJuridica !=
            null && !this.id_PessoaJuridica.equals(other.id_PessoaJuridica)));
        }
    }

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

Produto.java



```
package cadastroee.model;
```

```
import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
```

```
@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.id_Produto = :id_Produto"),
```

```
    @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
```

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

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

```
    private Integer id_Produto;
```

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

```
    private String nome;
```

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

```
    private Integer quantidade;
```

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

```
    private Float precoVenda;
```

```
    @OneToOne(mappedBy = "id_Produto")
```

```
    private Collection<MovimentoVenda> movimentoVendaCollection;
```



```
public Produto() {  
}  
  
public Produto(Integer id_Produto) {  
    this.id_Produto = id_Produto;  
}  
  
public Integer getId_Produto() {  
    return id_Produto;  
}  
  
public void setId_Produto(Integer id_Produto) {  
    this.id_Produto = id_Produto;  
}  
  
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 Float getPrecoVenda() {  
    return precoVenda;  
}  
  
public void setPrecoVenda(Float precoVenda) {  
    this.precoVenda = precoVenda;  
}  
  
public Collection<MovimentoVenda> getMovimentoVendaCollection() {  
    return movimentoVendaCollection;  
}  
  
public void setMovimentoVendaCollection(Collection<MovimentoVenda> movimentoVendaCollection) {  
    this.movimentoVendaCollection = movimentoVendaCollection;  
}
```



```
public Collection<MovimentoCompra> getMovimentoCompraCollection(Collection<MovimentoCompra>
movimentoCompraCollection) {
    return movimentoCompraCollection;
}

public void setMovimentoCompraCollection(Collection<MovimentoCompra> movimentoCompraCollection)
{
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (id_Produto != null ? id_Produto.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;
    return !((this.id_Produto == null && other.id_Produto != null) || (this.id_Produto != null &&
!this.id_Produto.equals(other.id_Produto)));
}

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

Usuario.java

```
package cadastroee.model;

import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
```



```
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;
```

```
@Entity
```

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

```
@NamedQueries({
```

```
    @NamedQuery(name = "Usuario.findAll", query = "SELECT u FROM Usuario u"),
```

```
    @NamedQuery(name = "Usuario.findById_Usuario", query = "SELECT u FROM Usuario u WHERE  
u.id_Usuario = :id_Usuario"),
```

```
    @NamedQuery(name = "Usuario.findByName", query = "SELECT u FROM Usuario u WHERE u.nome =  
:nome"),
```

```
    @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
```

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

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

```
    private Integer id_Usuario;
```

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

```
    private String nome;
```

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

```
    private String senha;
```

```
    @OneToMany(mappedBy = "id_Usuario")
```

```
    private Collection<MovimentoVenda> movimentoVendaCollection;
```

```
    @OneToMany(mappedBy = "id_Usuario")
```

```
    private Collection<MovimentoCompra> movimentoCompraCollection;
```

```
    public Usuario() {
```

```
    }
```

```
    public Usuario(Integer id_Usuario) {
```

```
        this.id_Usuario = id_Usuario;
```

```
    }
```

```
    public Integer getId_Usuario() {
```

```
        return id_Usuario;
```

```
    }
```

```
    public void setId_Usuario(Integer id_Usuario) {
```



```
this.id_Usuario = id_Usuario;
}

public String getNome() {
    return nome;
}

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

public String getSenha() {
    return senha;
}

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

public Collection<MovimentoVenda> getMovimentoVendaCollection() {
    return movimentoVendaCollection;
}

public void setMovimentoVendaCollection(Collection<MovimentoVenda> movimentoVendaCollection) {
    this.movimentoVendaCollection = movimentoVendaCollection;
}

public Collection<MovimentoCompra> getMovimentoCompraCollection() {
    return movimentoCompraCollection;
}

public void setMovimentoCompraCollection(Collection<MovimentoCompra> movimentoCompraCollection)
{
    this.movimentoCompraCollection = movimentoCompraCollection;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (id_Usuario != null ? id_Usuario.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;
    return !((this.id_Usuario == null && other.id_Usuario != null) || (this.id_Usuario != null &&
!this.id_Usuario.equals(other.id_Usuario)));
}

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