



Estácio

Missão Prática | Nível 4 | Mundo 3

RPG0017 - Vamos integrar sistemas.

Aluna: Simone Ramos de Jesus.

Matricula: 202208290965.

Campus: Polo Prado – Belo Horizonte – MG.

Curso: Desenvolvimento Full Stack

Turma: 2023.3

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.

Analise:

Organização de Projeto no NetBeans:

- Projetos corporativos são organizados em módulos, pacotes e classes no NetBeans, geralmente envolvendo diferentes tipos de projetos Java EE.

Papel de JPA e EJB em Aplicações Web Java:

- JPA mapeia objetos Java para entidades de banco de dados, enquanto EJB fornece componentes empresariais para lógica de negócios, facilitando o desenvolvimento de aplicativos escaláveis.

Melhoria de Produtividade com NetBeans, JPA e EJB:

- O NetBeans oferece suporte integrado para JPA e EJB, melhorando a eficiência do desenvolvedor ao fornecer ferramentas de desenvolvimento, depuração e gerenciamento.

Servlets no NetBeans:

- O NetBeans facilita o desenvolvimento de Servlets em projetos Web, componentes Java que respondem a requisições HTTP. Comunicação entre

Servlets e Session Beans no NetBeans:

- A comunicação ocorre por chamadas de método aos Session Beans do pool de EJBs, utilizando injeção de dependência ou pesquisa no contexto JNDI.

Padrão Front Controller e MVC no NetBeans:

- O NetBeans suporta a implementação do padrão Front Controller usando Servlets como controladores, seguindo a arquitetura MVC para separação de preocupações.

Diferenças e Semelhanças entre Servlets e JSPs:

- Servlets respondem a solicitações HTTP, enquanto JSPs são usados para criar interfaces web dinâmicas. Servlets são mais programáticos, JSPs mais focados em markup.

Redirecionamento e Método Forward no NetBeans:

- O redirecionamento leva o cliente para outra página, enquanto o método forward transfere o controle para outra página no servidor. Parâmetros e atributos nos objetos HttpRequest transmitem dados entre partes do aplicativo.

Uso do Framework Bootstrap:

- Bootstrap é um framework front-end que simplifica o desenvolvimento de interfaces responsivas, oferecendo estilos predefinidos e componentes reutilizáveis.

Independência Estrutural do HTML com Bootstrap:

- Bootstrap garante independência estrutural do HTML ao fornecer estilos e componentes consistentes.

Relação entre Bootstrap e Responsividade da Página:

- Bootstrap é projetado para responsividade, adaptando-se automaticamente ao tamanho da tela com um sistema de grid e classes específicas para uma experiência consistente em dispositivos variados.

Conclusão:

Em conclusão, o NetBeans se destaca na organização eficiente de projetos corporativos Java, oferecendo suporte integrado para tecnologias fundamentais como JPA e EJB. A plataforma proporciona melhorias significativas na produtividade do desenvolvedor, simplificando o desenvolvimento de Servlets e facilitando a comunicação entre eles e os Session Beans. Além disso, a implementação do padrão Front Controller e a adoção da arquitetura MVC são suportadas, promovendo uma estrutura clara e modular. Ao incorporar o Bootstrap, o NetBeans possibilita a criação de interfaces web responsivas e visualmente atraentes, garantindo a independência estrutural do HTML e contribuindo para uma experiência consistente em diversos dispositivos.

Pessoa.java:

```
Pessoa.java 1 X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Pessoa.java
1 package cadastroee.model;
2
3 import java.io.Serializable;
4 import javax.persistence.Basic;
5 import javax.persistence.Column;
6 import javax.persistence.Entity;
7 import javax.persistence.GeneratedValue;
8 import javax.persistence.GenerationType;
9 import javax.persistence.Id;
10 import javax.persistence.NamedQueries;
11 import javax.persistence.NamedQuery;
12 import javax.persistence.Table;
13
14 @Entity
15 @Table(name = "Pessoa")
16
17 @NamedQueries({
18     @NamedQuery(name = "Pessoa.findAll", query = "SELECT pes FROM Pessoa pes"),
19     @NamedQuery(name = "Pessoa.findByIdPessoa", query = "SELECT pes FROM Pessoa pes WHERE pes.idPessoa = :idPessoa"),
20     @NamedQuery(name = "Pessoa.findByName", query = "SELECT pes FROM Pessoa pes WHERE pes.nome = :nome"),
21     @NamedQuery(name = "Pessoa.findByLogradouro", query = "SELECT pes FROM Pessoa pes WHERE pes.logradouro = :logradouro"),
22     @NamedQuery(name = "Pessoa.findByCidade", query = "SELECT pes FROM Pessoa pes WHERE pes.cidade = :cidade"),
23     @NamedQuery(name = "Pessoa.findByEstado", query = "SELECT pes FROM Pessoa pes WHERE pes.estado = :estado"),
24     @NamedQuery(name = "Pessoa.findByTelefone", query = "SELECT pes FROM Pessoa pes WHERE pes.telefone = :telefone"),
25     @NamedQuery(name = "Pessoa.findByEmail", query = "SELECT pes FROM Pessoa pes WHERE pes.email = :email"),
26     @NamedQuery(name = "Pessoa.findByTpPessoa", query = "SELECT pes FROM Pessoa pes WHERE pes.tpPessoa = :tpPessoa")
27 })
28 public class Pessoa implements Serializable {
29
30     private static final long serialVersionUID = 1L;
31     @Id
32     @GeneratedValue(strategy = GenerationType.IDENTITY)
33     @Basic(optional = false)
34     @Column(name = "idPessoa")
35     private Integer idPessoa;
36     @Column(name = "nome")
37     private String nome;
```

Pessoa.java 1 X

CadastroEE-ejb > cadastroee.controller > cadastroee.model > Pessoa.java

```
35     private Integer idPessoa;
36     @Column(name = "nome")
37     private String nome;
38     @Column(name = "logradouro")
39     private String logradouro;
40     @Column(name = "cidade")
41     private String cidade;
42     @Column(name = "estado")
43     private String estado;
44     @Column(name = "telefone")
45     private String telefone;
46     @Column(name = "email")
47     private String email;
48     @Column(name = "tpPessoa")
49     private String tpPessoa;
50
51     public Pessoa() {
52     }
53
54     public Pessoa(Integer idPessoa) {
55         this.idPessoa = idPessoa;
56     }
57
58     public Integer getIdPessoa() {
59         return idPessoa;
60     }
61
62     public void setIdPessoa(Integer idPessoa) {
63         this.idPessoa = idPessoa;
64     }
65
66     public String getNome() {
67         return nome;
68     }
69
70     public void setNome(String nome) {
71         this.nome = nome;
```

Pessoa.java 1 X

CadastroEE-ejb > cadastroee.controller > cadastroee.model > Pessoa.java

```
73
74     public String getLogradouro() {
75         return logradouro;
76     }
77
78     public void setLogradouro(String logradouro) {
79         this.logradouro = logradouro;
80     }
81
82     public String getCidade() {
83         return cidade;
84     }
85
86     public void setCidade(String cidade) {
87         this.cidade = cidade;
88     }
89
90     public String getEstado() {
91         return estado;
92     }
93
94     public void setEstado(String estado) {
95         this.estado = estado;
96     }
97
98     public String getTelefone() {
99         return telefone;
100    }
101
102    public void setTelefone(String telefone) {
103        this.telefone = telefone;
104    }
105
106    public String getEmail() {
107        return email;
108    }
109
```

Pessoa.java 1 X

CadastroEE-ejb > cadastroee.controller > cadastroee.model > Pessoa.java

```
110     public void setEmail(String email) {
111         this.email = email;
112     }
113
114     public String getTpPessoa() {
115         return tpPessoa;
116     }
117
118     public void setTpPessoa(String tpPessoa) {
119         this.tpPessoa = tpPessoa;
120     }
121
122     @Override
123     public int hashCode() {
124         int hash = 0;
125         hash += (idPessoa != null ? idPessoa.hashCode() : 0);
126         return hash;
127     }
128
129     @Override
130     public boolean equals(Object object) {
131         if (!(object instanceof Pessoa)) {
132             return false;
133         }
134         Pessoa other = (Pessoa) object;
135         if ((this.idPessoa == null && other.idPessoa != null)
136             || (this.idPessoa != null && !this.idPessoa.equals(other.idPessoa))) {
137             return false;
138         }
139         return true;
140     }
141
142     @Override
143     public String toString() {
144         return "cadastroee.model.Pessoa[ idPessoa=" + idPessoa + " ]";
145     }
146
```


- PessoaFisica.java:

```
PessoaFisica.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > PessoaFisica.java > PessoaFisica
1  import java.io.Serializable;
2  import java.util.Collection;
3
4  import jakarta.persistence.Id;
5  import jakarta.persistence.Basic;
6  import jakarta.persistence.Table;
7  import jakarta.persistence.OneToMany;
8  import jakarta.persistence.Entity;
9  import jakarta.persistence.NamedQueries;
10 import jakarta.persistence.NamedQuery;
11 import jakarta.persistence.Column;
12
13 @Entity
14 @Table(name = "PessoaFisica")
15 @NamedQueries({
16     @NamedQuery(name = "PessoaFisica.findAll", query = "SELECT pf FROM Pessoa_Fisica pf"),
17     @NamedQuery(name = "PessoaFisica.findByIdPessoaFisica", query = "SELECT pf FROM PessoaFisica pf WHERE pf.id = :id"),
18     @NamedQuery(name = "PessoaFisica.findByCpf", query = "SELECT p FROM PessoaFisica p WHERE p.cpf = :cpf")
19 })
19 public class PessoaFisica implements Serializable {
20
21     private static final Long serialVersionUID = 1L;
22     @Id
23     @Basic(optional = false)
24     @Column(name = "id_PessoaFisica")
25     private Integer id_PessoaFisica;
26     @Column(name = "cpf")
27     private String cpf;
28     @OneToMany(mappedBy = "id_Comprador")
29     private Collection<MovimentoVenda> movimentoVendaCollection;
30
31     public PessoaFisica() {
32     }
33
34     public PessoaFisica(Integer id_PessoaFisica) {
35         this.id_PessoaFisica = id_PessoaFisica;
36     }
37 }
```

```

PessoaFisica.java 9+
CadastroEE-ejb > cadastroee.controller > cadastroee.model > PessoaFisica.java > PessoaFisica<movimentoVendaCollection> > equals(Object)
37
38     public Integer getId_PessoaFisica() {
39         return id_PessoaFisica;
40     }
41
42     public void setId_PessoaFisica(Integer id_PessoaFisica) {
43         this.id_PessoaFisica = id_PessoaFisica;
44     }
45
46     public String getCpf() {
47         return cpf;
48     }
49
50     public void setCpf(String cpf) {
51         this.cpf = cpf;
52     }
53
54
55     public <MovimentoVenda> Collection <MovimentoVenda> getMovimentoVendaCollection() {
56         return movimentoVendaCollection;
57     }
58
59     public void setMovimentoVendaCollection(Collection<MovimentoVenda> movimentoVendaCollection) {
60         this.movimentoVendaCollection = movimentoVendaCollection;
61     }
62
63     @Override
64     public int hashCode() {
65         int hash = 0;
66         hash += (id_PessoaFisica != null ? id_PessoaFisica.hashCode() : 0);
67         return hash;
68     }
69
70     @Override
71     public boolean equals(Object object) {
72

```

```

PessoaFisica.java 9+
CadastroEE-ejb > cadastroee.controller > cadastroee.model > PessoaFisica.java > PessoaFisica<movimentoVendaCollection> > equals(Object)
69
70     @Override
71     public boolean equals(Object object) {
72
73         if (!(object instanceof PessoaFisica)) {
74             return false;
75         }
76         PessoaFisica other = (PessoaFisica) object;
77         if ((this.id_PessoaFisica == null && other.id_PessoaFisica != null)
78             || (this.id_PessoaFisica != null && !this.id_PessoaFisica.equals(other.id_PessoaFisica))) {
79             return false;
80         }
81         return true;
82     }
83
84     @Override
85     public String toString() {
86         return "cadastroee.model.PessoaFisica[ id_PessoaFisica=" + id_PessoaFisica + " ]";
87     }
88
89 }
90

```

- PessoaJuridica.java:

```
PessoaJuridica.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > PessoaJuridica.java > PessoaJuridica

1  import java.io.Serializable;
2  import java.util.Collection;
3
4  import jakarta.persistence.Entity;
5  import jakarta.persistence.NamedQueries;
6  import jakarta.persistence.NamedQuery;
7  import jakarta.persistence.Table;
8  import jakarta.persistence.Basic;
9  import jakarta.persistence.Column;
10 import jakarta.persistence.Id;
11 import jakarta.persistence.OneToMany;
12
13 @Entity
14 @Table(name = "PessoaJuridica")
15 @NamedQueries({
16     @NamedQuery(name = "PessoaJuridica.findAll", query = "SELECT p FROM PessoaJuridica p"),
17     @NamedQuery(name = "PessoaJuridica.findByIdPessoaJuridica", query = "SELECT p FROM PessoaJuridica p w
18     @NamedQuery(name = "PessoaJuridica.findByCnpj", query = "SELECT p FROM PessoaJuridica p WHERE p.cnpj
19 public class PessoaJuridica implements Serializable {
20
21     private static final Long serialVersionUID = 1L;
22     @Id
23     @Basic(optional = false)
24     @Column(name = "id_PessoaJuridica")
25     private Integer id_PessoaJuridica;
26     @Column(name = "cnpj")
27     private String cnpj;
28     @OneToMany(mappedBy = "id_Forneecedor")
29     private Collection<MovimentoCompra> movimentoCompraCollection;
30
31     public PessoaJuridica() {
32     }
33
34     public PessoaJuridica(Integer id_PessoaJuridica) {
35         this.id_PessoaJuridica = id_PessoaJuridica;
36     }
37
```

PessoaJuridica.java 9+ X

CadastroEE-ejb > cadastroee.controller > cadastroee.model > PessoaJuridica.java > PessoaJuridica

```
37
38     public Integer getId_PessoaJuridica() {
39         return id_PessoaJuridica;
40     }
41
42     public void setId_PessoaJuridica(Integer id_PessoaJuridica) {
43         this.id_PessoaJuridica = id_PessoaJuridica;
44     }
45
46     public String getCnpj() {
47         return cnpj;
48     }
49
50     public void setCnpj(String cnpj) {
51         this.cnpj = cnpj;
52     }
53
54     public Collection<MovimentoCompra> getMovimentoCompraCollection() {
55         return movimentoCompraCollection;
56     }
57
58     public void setMovimentoCompraCollection(Collection<MovimentoCompra> movimentoCompraCollection) {
59         this.movimentoCompraCollection = movimentoCompraCollection;
60     }
61
62     @Override
63     public int hashCode() {
64         int hash = 0;
65         hash += (id_PessoaJuridica != null ? id_PessoaJuridica.hashCode() : 0);
66         return hash;
67     }
68
69     @Override
70     public boolean equals(Object object) {
71         // TODO: Warning - this method won't work in the case the id fields are not set
72         if (!(object instanceof PessoaJuridica)) {
73             return false;
74
75         }
76
77         PessoaJuridica other = (PessoaJuridica) object;
78         if ((this.id_PessoaJuridica == null && other.id_PessoaJuridica != null)
79             || (this.id_PessoaJuridica != null && !this.id_PessoaJuridica.equals(other.id_PessoaJuridica))
80         ) {
81             return false;
82         }
83         return true;
84     }
85
86     @Override
87     public String toString() {
88         return "cadastroee.model.PessoaJuridica[ id_PessoaJuridica=" + id_PessoaJuridica + " ]";
89     }
90 }
```

Produto.java:

```
Produto.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Produto.java > Produto
1  import java.io.Serializable;
2  import java.util.Collection;
3
4  import jakarta.persistence.NamedQueries;
5  import jakarta.persistence.NamedQuery;
6  import jakarta.persistence.Basic;
7  import jakarta.persistence.OneToOne;
8  import jakarta.persistence.Column;
9  import jakarta.persistence.Entity;
10 import jakarta.persistence.Table;
11 import jakarta.persistence.Id;
12
13 @Entity
14 @Table(name = "Produto")
15 @NamedQueries({
16     @NamedQuery(name = "Produto.findAll", query = "SELECT p FROM Produto p"),
17     @NamedQuery(name = "Produto.findByIdProduto", query = "SELECT p FROM Produto p WHERE p.id_Produto = :id_Produto"),
18     @NamedQuery(name = "Produto.findByName", query = "SELECT p FROM Produto p WHERE p.nome = :nome"),
19     @NamedQuery(name = "Produto.findByQuantidade", query = "SELECT p FROM Produto p WHERE p.quantidade = :quantidade"),
20     @NamedQuery(name = "Produto.findByPrecoVenda", query = "SELECT p FROM Produto p WHERE p.precoVenda = :precoVenda")
21 })
22
23 public class Produto implements Serializable {
24
25     private static final Long serialVersionUID = 1L;
26     @Id
27     @Basic(optional = false)
28     @Column(name = "id_Produto")
29     private Integer id_Produto;
30     @Column(name = "nome")
31     private String nome;
32     @Column(name = "quantidade")
33     private Integer quantidade;
34     @Column(name = "precoVenda")
35     private Float precoVenda;
36     @OneToOne(mappedBy = "id_Produto")
37     private Collection<MovimentoVenda> movimentoVendaCollection;
```

```
Produto.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Produto.java > Produto
38 @OneToMany(mappedBy = "id_Produto")
39 private Collection<MovimentoCompra> movimentoCompraCollection;
40
41 public Produto() {
42 }
43
44 public Produto(Integer id_Produto) {
45     this.id_Produto = id_Produto;
46 }
47
48 public Integer getId_Produto() {
49     return id_Produto;
50 }
51
52 public void setId_Produto(Integer id_Produto) {
53     this.id_Produto = id_Produto;
54 }
55
56 public String getNome() {
57     return nome;
58 }
59
60 public void setNome(String nome) {
61     this.nome = nome;
62 }
63
64 public Integer getQuantidade() {
65     return quantidade;
66 }
67
68 public void setQuantidade(Integer quantidade) {
69     this.quantidade = quantidade;
70 }
71
72 public Float getPrecoVenda() {
73     return precoVenda;
74 }
```

```

Produto.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Produto.java > Produto
74     }
75
76     public void setPrecoVenda(Float precoVenda) {
77         this.precoVenda = precoVenda;
78     }
79
80     public Collection<MovimentoVenda> getMovimentoVendaCollection() {
81         return movimentoVendaCollection;
82     }
83
84     public void setMovimentoVendaCollection(Collection<MovimentoVenda> movimentoVendaCollection) {
85         this.movimentoVendaCollection = movimentoVendaCollection;
86     }
87
88     public Collection<MovimentoCompra> getMovimentoCompraCollection() {
89         return movimentoCompraCollection;
90     }
91
92     public void setMovimentoCompraCollection(Collection<MovimentoCompra> movimentoCompraCollection) {
93         this.movimentoCompraCollection = movimentoCompraCollection;
94     }
95
96     @Override
97     public int hashCode() {
98         int hash = 0;
99         hash += (id_Produto != null ? id_Produto.hashCode() : 0);
100        return hash;
101    }
102
103    @Override
104    public boolean equals(Object object) {
105        // TODO: Warning - this method won't work in the case the id fields are not set
106        if (!(object instanceof Produto)) {
107            return false;
108        }
109        Produto other = (Produto) object;
110        if ((this.id_Produto == null && other.id_Produto != null)

```

```

Produto.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Produto.java > Produto
102
103    @Override
104    public boolean equals(Object object) {
105        // TODO: Warning - this method won't work in the case the id fields are not set
106        if (!(object instanceof Produto)) {
107            return false;
108        }
109        Produto other = (Produto) object;
110        if ((this.id_Produto == null && other.id_Produto != null)
111            || (this.id_Produto != null && !this.id_Produto.equals(other.id_Produto))) {
112            return false;
113        }
114        return true;
115    }
116
117    @Override
118    public String toString() {
119        return "cadastroee.model.Produto[ id_Produto=" + id_Produto + " ]";
120    }
121
122 }
123

```

Usuario.java:

```
Usuario.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Usuario.java > Usuario
1  import java.io.Serializable;
2  import java.util.Collection;
3
4  import jakarta.persistence.NamedQueries;
5  import jakarta.persistence.NamedQuery;
6  import jakarta.persistence.Entity;
7  import jakarta.persistence.Table;
8  import jakarta.persistence.Basic;
9  import jakarta.persistence.Id;
10 import jakarta.persistence.Column;
11 import jakarta.persistence.OneToMany;
12
13 @Entity
14 @Table(name = "Usuario")
15 @NamedQueries({
16     @NamedQuery(name = "Usuario.findAll", query = "SELECT u FROM Usuario u"),
17     @NamedQuery(name = "Usuario.findById_Usuario", query = "SELECT u FROM Usuario u WHERE u.id_Usuario = :id_Usuario"),
18     @NamedQuery(name = "Usuario.findByName", query = "SELECT u FROM Usuario u WHERE u.nome = :nome"),
19     @NamedQuery(name = "Usuario.findBySenha", query = "SELECT u FROM Usuario u WHERE u.senha = :senha")
20 })
21 public class Usuario implements Serializable {
22
23     private static final Long serialVersionUID = 1L;
24     @Id
25     @Basic(optional = false)
26     @Column(name = "id_")
27     private Integer id_Usuario;
28     @Column(name = "nome")
29     private String nome;
30     @Column(name = "senha")
31     private String senha;
32     @OneToMany(mappedBy = "id_Usuario")
33     private Collection<MovimentoVenda> movimentoVendaCollection;
34     @OneToMany(mappedBy = "id_Usuario")
35     private Collection<MovimentoCompra> movimentoCompraCollection;
36 }
```


Usuario.java 9+ X

CadastroEE-ejb > cadastroee.controller > cadastroee.model > Usuario.java > Usuario

```
37  public Usuario() {
38  }
39
40  public Usuario(Integer id_Usuario) {
41      this.id_Operador = id_Usuario;
42  }
43
44  public Integer getId_Usuario() {
45      return id_Usuario;
46  }
47
48  public void setId_Usuario(Integer id_Usuario) {
49      this.id_Usuario = id_Usuario;
50  }
51
52  public String getNome() {
53      return nome;
54  }
55
56  public void setNome(String nome) {
57      this.nome = nome;
58  }
59
60  public String getSenha() {
61      return senha;
62  }
63
64  public void setSenha(String senha) {
65      this.senha = senha;
66  }
67
68  public Collection<MovimentoVenda> getMovimentoVendaCollection() {
69      return movimentoVendaCollection;
70  }
```

```

Usuario.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Usuario.java > Usuario

72     public void setMovimentoVendaCollection(Collection<MovimentoVenda> movimentoVendaCollection) {
73         this.movimentoVendaCollection = movimentoVendaCollection;
74     }
75
76     public Collection<MovimentoCompra> getMovimentoCompraCollection() {
77         return movimentoCompraCollection;
78     }
79
80     public void setMovimentoCompraCollection(Collection<MovimentoCompra> movimentoCompraCollection) {
81         this.movimentoCompraCollection = movimentoCompraCollection;
82     }
83
84     @Override
85     public int hashCode() {
86         int hash = 0;
87         hash += (idOperador != null ? idOperador.hashCode() : 0);
88         return hash;
89     }
90
91     @Override
92     public boolean equals(Object object) {
93         // TODO: Warning - this method won't work in the case the id fields are not set
94         if (!(object instanceof Usuario)) {
95             return false;
96         }
97         Usuario other = (Usuario) object;
98         if ((this.id_Usuario == null && other.id_Usuario != null)
99             || (this.id_Usuario != null && !this.id_Usuario.equals(other.id_Usuario))) {
100             return false;
101         }
102         return true;
103     }
104
105     @Override
106     public String toString() {
107         return "cadastroee.model.Usuario[ id_Usuario=" + id_Usuario + " ]";
108     }

```

ProdutoFacade.java:

```

ProdutoFacade.java 9+ X
CadastroEE-ejb > cadastroee.controller > ProdutoFacade.java > ...

1  import java.util.List;
2
3  import cadastroee.model.Produto;
4
5  import jakarta.persistence.PersistenceContext;
6  import jakarta.ejb.Stateless;
7  import jakarta.persistence.EntityManager;
8
9  @Stateless
10 public class ProdutoFacade implements ProdutoFacadeLocal {
11
12     @PersistenceContext(unitName = "CadastroEE-ejb")
13     private EntityManager em;
14
15     protected EntityManager getEntityManager() {
16         return em;
17     }
18
19     @Override
20     public List<Produto> findAll() {
21         jakarta.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery()
22         cq.select(cq.from(Produto.class));
23         return getEntityManager().createQuery(cq).getResultList();
24     }
25
26 }
27

```

ProdutoFacadeLocal.java:

```
ProdutoFacadeLocal.java 4 X
CadastroEE-ejb > cadastroee.controller > ProdutoFacadeLocal.java > ProdutoFacadeLocal
1 package cadastroee.controller;
2
3 import java.util.List;
4 import cadastroee.model.Produto;
5 import jakarta.ejb.Local;
6
7 @Local
8 public interface ProdutoFacadeLocal {
9     List<Produto> findAll();
10 }
```

ServletsProduto.java:

```
Usuario.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Usuario.java > Usuario
1 import java.io.Serializable;
2 import java.util.Collection;
3
4 import jakarta.persistence.NamedQueries;
5 import jakarta.persistence.NamedQuery;
6 import jakarta.persistence.Entity;
7 import jakarta.persistence.Table;
8 import jakarta.persistence.Basic;
9 import jakarta.persistence.Id;
10 import jakarta.persistence.Column;
11 import jakarta.persistence.OneToMany;
12
13 @Entity
14 @Table(name = "Usuario")
15 @NamedQueries({
16     @NamedQuery(name = "Usuario.findAll", query = "SELECT u FROM Usuario u"),
17     @NamedQuery(name = "Usuario.findById_Usuario", query = "SELECT u FROM Usuario u WHERE u.id_Usuario = :id_Usuario"),
18     @NamedQuery(name = "Usuario.findByName", query = "SELECT u FROM Usuario u WHERE u.nome = :nome"),
19     @NamedQuery(name = "Usuario.findBySenha", query = "SELECT u FROM Usuario u WHERE u.senha = :senha")
20 })
21 public class Usuario implements Serializable {
22
23     private static final long serialVersionUID = 1L;
24
25     @Id
26     @Basic(optional = false)
27     @Column(name = "id_")
28     private Integer id_Usuario;
29     @Column(name = "nome")
30     private String nome;
31     @Column(name = "senha")
32     private String senha;
33     @OneToMany(mappedBy = "id_Usuario")
34     private Collection<MovimentoVenda> movimentoVendaCollection;
35     @OneToMany(mappedBy = "id_Usuario")
36     private Collection<MovimentoCompra> movimentoCompraCollection;
```

Usuario.java 9+ X

CadastroEE-ejb > cadastroee.controller > cadastroee.model > Usuario.java > Usuario

```
37  public Usuario() {
38  }
39
40  public Usuario(Integer id_Usuario) {
41      this.id_Usuario = id_Usuario;
42  }
43
44  public Integer getId_Usuario() {
45      return id_Usuario;
46  }
47
48  public void setId_Usuario(Integer id_Usuario) {
49      this.id_Usuario = id_Usuario;
50  }
51
52  public String getNome() {
53      return nome;
54  }
55
56  public void setNome(String nome) {
57      this.nome = nome;
58  }
59
60  public String getSenha() {
61      return senha;
62  }
63
64  public void setSenha(String senha) {
65      this.senha = senha;
66  }
67
68  public Collection<MovimentoVenda> getMovimentoVendaCollection() {
69      return movimentoVendaCollection;
70  }
```

```
Usuario.java 9+ X
CadastroEE-ejb > cadastroee.controller > cadastroee.model > Usuario.java > Usuario

72     public void setMovimentoVendaCollection(Collection<MovimentoVenda> movimentoVendaCollection) {
73         this.movimentoVendaCollection = movimentoVendaCollection;
74     }
75
76     public Collection<MovimentoCompra> getMovimentoCompraCollection() {
77         return movimentoCompraCollection;
78     }
79
80     public void setMovimentoCompraCollection(Collection<MovimentoCompra> movimentoCompraCollection) {
81         this.movimentoCompraCollection = movimentoCompraCollection;
82     }
83
84     @Override
85     public int hashCode() {
86         int hash = 0;
87         hash += (idOperador != null ? idOperador.hashCode() : 0);
88         return hash;
89     }
90
91     @Override
92     public boolean equals(Object object) {
93         // TODO: Warning - this method won't work in the case the id fields are not set
94         if (!(object instanceof Usuario)) {
95             return false;
96         }
97         Usuario other = (Usuario) object;
98         if ((this.id_Usuario == null && other.id_Usuario != null)
99             || (this.id_Usuario != null && !this.id_Usuario.equals(other.id_Usuario))) {
100             return false;
101         }
102         return true;
103     }
104
105     @Override
106     public String toString() {
107         return "cadastroee.model.Usuario[ id_Usuario=" + id_Usuario + " ]";
108     }
}
```

