

WorkShop Curso Java para Web

Ambient Informática

July 31, 2008

1 Intrdução

2 O Modelo

O modelo de classes do sistema está na figura 2.1. Nesta figura é possível ver os relacionamentos entre as classes.

Um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. É uma modelagem muito útil para o sistema, define todas as classes que o sistema necessita possuir e é a base para a construção dos diagramas de comunicação, sequência e estados.

Um diagrama de classes possui:

- Classe
- Atributo: Definem características
- Métodos: Operações que o objeto é capaz de realizar
- Associação: Relacionamentos entre as classes como multiplicidade e navegação

2.1 O Mapeamento Objeto-Relacional

Para realizar o mapeamento objeto-relacional (converter os objetos em tabelas relacionais para o banco de dados) será utilizada a Java Persistence API (JPA).

Para configurar o mapeamento é necessário editar o arquivo persistence.xml apresentado na listagem 1. Neste arquivo estão as configurações da conexão com o Banco de Dados e o comportamento do Hibernate.

Após realizada a configuração deste arquivo é necessário editar as classes do modelo para definir como a JPA irá mapea-las para o Banco de Dados.

A anotação `@Entity` define uma entidade. Toda entidade necessita de um identificador. Este identificador é descrito utilizando a anotação `@Id`:

```
@Id
@SequenceGenerator
@GeneratedValue
```

As demais anotações configuram um Sequence para obtenção de id's.

A classe da listagem 10 apresenta como serão criados os EntityManagers.

A primeira atividade consiste em implementar as classes do modelo e realizar seu mapeamento objeto-relacional.

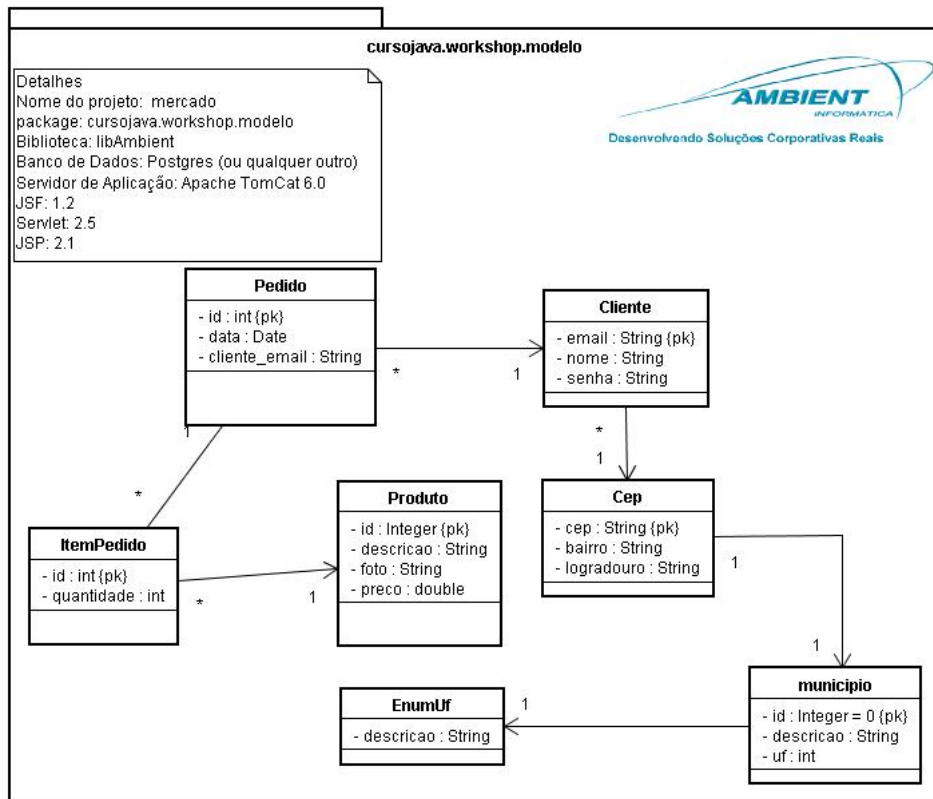


Figure 1: Diagrama de Classes do Modelo

Listing 1: O arquivo persistence.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="1.0"
3     xmlns="http://java.sun.com/xml/ns/persistence"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
6         http://java.sun.com/xml/ns/persistence/persistence_1_0.
7         xsd">
8     <persistence-unit name="mercado">
9         <properties>
10             <property name="hibernate.hbm2ddl.auto"
11                 value="update" />
12             <property name="hibernate.dialect"
13                 value="org.hibernate.dialect.
14                     PostgreSQLDialect" />
15             <!-- dados da conexao com o banco -->
16             <property name="hibernate.connection.
17                 driver_class"
18                 value="org.postgresql.Driver" />
19             <property name="hibernate.connection.
20                 username"
21                 value="postgres" />
22             <property name="hibernate.connection.

```

```

18         password"
19             value="123456" />
20     <property name="hibernate.connection.url"
21             value="jdbc:postgresql://localhost/
22             mercado" />
23
24     <property name="hibernate.max_fetch_depth"
25             value="3" />
26
27     <!-- mostra o sql no log -->
28     <property name="hibernate.show_sql" value="
29         true" />
30 </properties>
31 </persistence-unit>
32 </persistence>

```

Listing 2: Classe Municipio

```

1 package cursojava.workshop.modelo;
2
3 import javax.persistence.Entity;
4 import javax.persistence.EnumType;
5 import javax.persistence.Enumerated;
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.GenerationType;
8 import javax.persistence.Id;
9 import javax.persistence.SequenceGenerator;
10
11 @Entity
12 public class Municipio {
13
14     @Id
15     @GeneratedValue(strategy=GenerationType.SEQUENCE, generator = "
16         geradorMunicipio")
17     @SequenceGenerator(name="geradorMunicipio", sequenceName = "
18         gerador_municipio", allocationSize=1, initialValue=1)
19     private Integer id = 0;
20
21     private String descricao;
22
23     @Enumerated(EnumType.STRING)
24     private EnumUf uf;
25
26     public Integer getId() {
27         return id;
28     }
29
30     public void setId(Integer id) {
31         this.id = id;
32     }
33
34     public String getDescricao() {
35         return descricao;
36     }
37
38     public void setDescricao(String descricao) {
39         this.descricao = descricao;
40     }
41
42     public EnumUf getUf() {
43         return uf;
44     }
45 }

```

```

43
44     public void setUf(EnumUf uf) {
45         this.uf = uf;
46     }
47 }

```

Listing 3: Classe Cep

```

1  package cursojava.workshop.modelo;
2
3  import javax.persistence.Entity;
4  import javax.persistence.Id;
5  import javax.persistence.OneToOne;
6
7  @Entity
8  public class Cep {
9
10     @Id
11     private String cep;
12
13     @OneToOne
14     private Municipio municipio;
15
16     private String bairro;
17     private String logradouro;
18
19     public String getCep() {
20         return cep;
21     }
22
23     public void setCep(String cep) {
24         this.cep = cep;
25     }
26
27     public Municipio getMunicipio() {
28         return municipio;
29     }
30
31     public void setMunicipio(Municipio municipio) {
32         this.municipio = municipio;
33     }
34
35     public String getBairro() {
36         return bairro;
37     }
38
39     public void setBairro(String bairro) {
40         this.bairro = bairro;
41     }
42
43     public String getLogradouro() {
44         return logradouro;
45     }
46
47     public void setLogradouro(String logradouro) {
48         this.logradouro = logradouro;
49     }
50 }

```

Listing 4: Classe Produto

```

1  package cursojava.workshop.modelo;

```

```

2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7 import javax.persistence.SequenceGenerator;
8
9 @Entity
10 public class Produto {
11
12     @Id
13     @SequenceGenerator(name="geradorProduto", sequenceName="
        gerador_produto", allocationSize=1, initialValue=1)
14     @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="
        geradorProduto")
15     private int id;
16     private String descricao;
17     private double preco;
18     private String foto;
19
20     public boolean equals(Object obj) {
21         try {
22             Produto p = (Produto) obj;
23             return p.getDescricao().equals(descricao) && p.getId() ==
                id;
24         } catch (Exception e) {
25             return false;
26         }
27     }
28
29     public String getDescricao() {
30         return descricao;
31     }
32     public void setDescricao(String descricao) {
33         this.descricao = descricao;
34     }
35     public double getPreco() {
36         return preco;
37     }
38     public void setPreco(double preco) {
39         this.preco = preco;
40     }
41     public String getFoto() {
42         return foto;
43     }
44     public void setFoto(String foto) {
45         this.foto = foto;
46     }
47     public int getId() {
48         return id;
49     }
50 }
51

```

Listing 5: Classe Pedido

```

1 package cursojava.workshop.modelo;
2
3 import java.util.ArrayList;
4 import java.util.Date;
5 import java.util.List;
6

```

```

7  import javax.persistence.CascadeType;
8  import javax.persistence.Entity;
9  import javax.persistence.GeneratedValue;
10 import javax.persistence.GenerationType;
11 import javax.persistence.Id;
12 import javax.persistence.ManyToOne;
13 import javax.persistence.OneToMany;
14 import javax.persistence.SequenceGenerator;
15
16 @Entity
17 public class Pedido {
18
19     @Id
20     @SequenceGenerator(name="geradorPedido", sequenceName="
        gerador_pedido", allocationSize=1, initialValue=1)
21     @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="
        geradorPedido")
22     private int id;
23     private Date data;
24
25     @ManyToOne
26     private Cliente cliente;
27
28     @OneToMany(cascade=CascadeType.ALL, mappedBy="pedido")
29     private List<ItemPedido> itens = new ArrayList<ItemPedido>();
30
31     public void addProduto(Produto produto){
32         ItemPedido novoItem = new ItemPedido(produto, this);
33         if(itens.contains(novoItem)){
34             ItemPedido item = itens.get(itens.indexOf(novoItem));
35             item.setQuantidade(item.getQuantidade()+1);
36         }else{
37             itens.add(novoItem);
38         }
39     }
40
41     public void removerItem(ItemPedido item){
42         itens.remove(item);
43     }
44
45     public double getPrecoTotal(){
46         double preco = 0;
47         for(ItemPedido item : itens){
48             preco += item.getPrecoTotal();
49         }
50         return preco;
51     }
52
53     public boolean equals(Object obj) {
54         try {
55             Pedido p = (Pedido) obj;
56             return p.getId() == id;
57         } catch (Exception e) {
58             return false;
59         }
60     }
61
62     public Date getData() {
63         return data;
64     }
65
66     public void setData(Date data) {

```

```

67         this.data = data;
68     }
69
70     public Cliente getCliente() {
71         return cliente;
72     }
73
74     public void setCliente(Cliente cliente) {
75         this.cliente = cliente;
76     }
77
78     public List<ItemPedido> getItens() {
79         return itens;
80     }
81
82     public void setItens(List<ItemPedido> itens) {
83         this.itens = itens;
84     }
85
86     public int getId() {
87         return id;
88     }
89 }
90

```

Listing 6: Classe ItemPedido

```

1 package cursojava.workshop.modelo;
2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7 import javax.persistence.ManyToOne;
8 import javax.persistence.SequenceGenerator;
9
10 @Entity
11 public class ItemPedido {
12
13     @Id
14     @SequenceGenerator(name="geradorItemPedido", sequenceName="
        gerador_itemPedido", allocationSize=1, initialValue=1)
15     @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="
        geradorItemPedido")
16     private int id;
17
18     private int quantidade = 1;
19
20     @ManyToOne
21     private Produto produto;
22
23     @ManyToOne
24     private Pedido pedido;
25
26     ItemPedido() {
27     }
28
29     public ItemPedido(Produto produto, Pedido pedido) {
30         this.produto = produto;
31         this.pedido = pedido;
32     }
33

```

```

34     public double getPrecoTotal() {
35         return quantidade * produto.getPreco();
36     }
37
38
39     public boolean equals(Object obj) {
40         try {
41             ItemPedido item = (ItemPedido) obj;
42             return item.getProduto().equals(produto)
43                 && item.getPedido().equals(pedido);
44         } catch (Exception e) {
45             return false;
46         }
47     }
48
49     public int getQuantidade() {
50         return quantidade;
51     }
52
53     public void setQuantidade(int quantidade) {
54         this.quantidade = quantidade;
55     }
56
57     public Produto getProduto() {
58         return produto;
59     }
60
61     public void setProduto(Produto produto) {
62         this.produto = produto;
63     }
64
65     public Pedido getPedido() {
66         return pedido;
67     }
68
69     public void setPedido(Pedido pedido) {
70         this.pedido = pedido;
71     }
72
73     public int getId() {
74         return id;
75     }
76 }

```

Listing 7: Classe Cliente

```

1  package cursojava.workshop.modelo;
2
3  import javax.persistence.Entity;
4  import javax.persistence.Id;
5  import javax.persistence.ManyToOne;
6
7  @Entity
8  public class Cliente {
9
10     @Id
11     private String email;
12
13     private String nome;
14
15     private String senha;
16

```



```

17     @ManyToOne
18     private Cep cep;
19
20     public String getNome() {
21         return nome;
22     }
23
24     public void setNome(String nome) {
25         this.nome = nome;
26     }
27
28     public String getSenha() {
29         return senha;
30     }
31
32     public void setSenha(String senha) {
33         this.senha = senha;
34     }
35
36     public Cep getCep() {
37         return cep;
38     }
39
40     public void setCep(Cep cep) {
41         this.cep = cep;
42     }
43
44     public String getEmail() {
45         return email;
46     }
47
48     public void setEmail(String email) {
49         this.email = email;
50     }
51 }
52

```

Listing 8: Classe EnumUf

```

1 package cursojava.workshop.modelo;
2 public enum EnumUf{
3
4     AC("Acre"),
5     AL("Alagoas"),
6     AP("Amapa"),
7     AM("Amazonas"),
8     BA("Bahia"),
9     CE("Ceara"),
10    DF("Distrito Federal"),
11    ES("Espirito Santo"),
12    GO("Goiias"),
13    MA("Maranhao"),
14    MT("Mato Grosso"),
15    MS("Mato Grosso do Sul"),
16    MG("Minas Gerais"),
17    PA("Para"),
18    PB("Paraiba"),
19    PE("Pernambuco"),
20    PR("Parana"),
21    PI("Piaui"),
22    RJ("Rio De Janeiro"),
23    RN("Rio Grande do Norte"),

```

```

24     RS("Rio Grande do Sul"),
25     RO("Rondonia"),
26     RR("Roraima"),
27     SC("Santa Catarina"),
28     SE("Sergipe"),
29     SP("Sao Paulo"),
30     TO("Tocantins");
31
32     private String descricao;
33
34     EnumUf(String descricao) {
35         this.descricao = descricao;
36     }
37
38     public String getDescricao() {
39         return descricao;
40     }
41
42     public void setDescricao(String descricao) {
43         this.descricao = descricao;
44     }
45 }

```

Listing 9: Classe Mercado Exception

```

1 package cursojava.workshop.modelo;
2
3 public class MercadoException extends Exception{
4
5     private static final long serialVersionUID = 1L;
6
7     public MercadoException(String message) {
8         super(message);
9     }
10
11 }

```

3 A Camada de Persistência

Conceitualmente, uma Camada de Persistência de Objetos é uma biblioteca que permite a realização do processo de persistência (isto é, o armazenamento e manutenção do estado de objetos em algum meio não-volátil, como um banco de dados) de forma transparente. Graças à independência entre a camada de persistência e o repositório (backend) utilizado, também é possível gerenciar a persistência de um modelo de objetos em diversos tipos de repositórios, teoricamente com pouco ou nenhum esforço extra. A utilização deste conceito permite ao desenvolvedor trabalhar como se estivesse em um sistema completamente orientado a objetos – utilizando métodos para incluir, alterar e remover objetos e uma linguagem de consulta para SGBDs Orientados a Objetos – comumente a linguagem OQL – para realizar consultas que retornam coleções de objetos instanciados.

A classe `ConexaoJpa` da listagem 10 é responsável por criar os entity managers que gerenciarão as entidades do sistema.

Observe a classe `Persistencia`, descrita na listagem 11. Esta classe contém os métodos comuns a qualquer persistência: incluir, excluir, alterar e consultar.

As demais classes herdam de Persistencia. Desta forma estas classes também terão os métodos de Persistencia.

Listing 10: Classe ConexaoJpa

```
1 package cursojava.workshop.persistencia;
2
3 import javax.persistence.EntityManager;
4 import javax.persistence.EntityManagerFactory;
5 import javax.persistence.Persistence;
6
7 public class ConexaoJpa {
8
9     private static EntityManagerFactory emf;
10
11     private static EntityManagerFactory criarEntityManagerFactory() {
12         if(emf == null){
13             emf = Persistence.createEntityManagerFactory("mercado");
14         }
15         return emf;
16     }
17
18     public static EntityManager criarEntityManager() {
19         return criarEntityManagerFactory().createEntityManager();
20     }
21
22 }
```

Listing 11: Classe Persistencia

```
1 package cursojava.workshop.persistencia;
2
3 import javax.persistence.EntityManager;
4
5 import cursojava.workshop.modelo.MercadoException;
6
7
8 public class Persistencia <E>{
9
10     private Class<E> classe;
11     protected EntityManager em;
12
13     public Persistencia(Class<E> classe) {
14         this.classe = classe;
15         em = ConexaoJpa.criarEntityManager();
16     }
17
18     public void abrirTransacao() {
19         em.getTransaction().begin();
20     }
21
22     public void commitTransacao() {
23         em.getTransaction().commit();
24     }
25
26     public void fechar() {
27         em.close();
28     }
29
30
31     public void incluir(E v) throws MercadoException{
32         try {
```

```

33         abrirTransacao();
34         em.persist(v);
35         commitTransacao();
36     } catch (Exception e) {
37         throw new MercadoException("Erro ao realizar a inclusao");
38     }
39 }
40
41 public void excluir(Object id) throws MercadoException{
42     try {
43         abrirTransacao();
44         E obj = em.find(classe, id);
45         em.remove(obj);
46         commitTransacao();
47     } catch (Exception e) {
48         throw new MercadoException("Erro ao realizar a exclusao");
49     }
50 }
51
52 public E alterar(E v) throws MercadoException{
53     try {
54         abrirTransacao();
55         E obj = em.merge(v);
56         commitTransacao();
57         return obj;
58     } catch (Exception e) {
59         throw new MercadoException("Erro ao realizar a alteracao")
60         ;
61     }
62 }
63
64 public E consultar(Object id) throws MercadoException{
65     try {
66         return em.find(classe, id);
67     } catch (Exception e) {
68         throw new MercadoException("Erro ao realizar a consulta");
69     }
70 }
71 }

```

Listing 12: Classe ProdutoDao

```

1 package cursojava.workshop.persistencia;
2
3 import java.util.List;
4
5 import javax.persistence.Query;
6
7 import cursojava.workshop.modelo.Produto;
8
9 public class ProdutoDao extends Persistencia<Produto>{
10
11     public ProdutoDao() {
12         super(Produto.class);
13     }
14
15     public List<Produto> listarPorDescricao(String descricao){
16         Query query = em.createQuery("select produto from Produto
17         produto where upper(produto.descricao) like :descricao");
18         query.setParameter("descricao", "%" + descricao.toUpperCase() +
19         "%");

```

```

18         return query.getResultList();
19     }
20 }

```

Listing 13: Classe ClienteDao

```

1 package cursojava.workshop.persistencia;
2
3 import javax.persistence.Query;
4
5 import cursojava.workshop.modelo.Cliente;
6 import cursojava.workshop.modelo.MercadoException;
7
8 public class ClienteDao extends Persistencia<Cliente> {
9
10     public ClienteDao() {
11         super(Cliente.class);
12     }
13
14     public void incluir(Cliente cliente) throws MercadoException {
15         Cliente c = consultar(cliente.getEmail());
16         if(c == null){
17             super.incluir(cliente);
18         }else{
19             throw new MercadoException("Ja existe um cliente com este
20                 email cadastrado.");
21         }
22     }
23
24     public Cliente autenticar(String email, String senha)
25     throws MercadoException {
26         try {
27             Query query = em
28                 .createQuery("select c from Cliente c where c.email = :
29                     email and c.senha = :senha");
30             query.setParameter("email", email);
31             query.setParameter("senha", senha);
32             return (Cliente) query.getSingleResult();
33         } catch (Exception e) {
34             throw new MercadoException("Usuario ou senha incorretos");
35         }
36     }
37 }

```

Listing 14: Classe CepDao

```

1 package cursojava.workshop.persistencia;
2
3 import cursojava.workshop.modelo.Cep;
4
5 public class CepDao extends Persistencia<Cep>{
6
7     public CepDao() {
8         super(Cep.class);
9     }
10
11
12
13 }

```

Listing 15: Classe PedidoDao

```

1 package cursojava.workshop.persistencia;
2
3 import cursojava.workshop.modelo.Pedido;
4
5 public class PedidoDao extends Persistencia<Pedido>{
6
7     public PedidoDao() {
8         super(Pedido.class);
9     }
10 }

```

4 A Camada de Controle

Para a camada de controle e visão será utilizada o JavaServer Faces.

As classes das listagens 17 e 18 são os managed beans.

A classe da listagem 16 possui utilidades para trabalhar com mensagens no JSF.

Listing 16: Classe Utilitario

```

1 package cursojava.workshop.util;
2
3 import javax.faces.application.FacesMessage;
4 import javax.faces.context.FacesContext;
5
6 public class Utilitario {
7
8     static public void addMensagemFaces(String msg) {
9         FacesContext context = FacesContext.getCurrentInstance();
10        FacesMessage mensagem = new FacesMessage(msg);
11        context.addMessage(null, mensagem);
12    }
13
14 }

```

Listing 17: Classe ClienteControl

```

1 package cursojava.workshop.controle;
2
3 import javax.faces.context.FacesContext;
4 import javax.faces.event.ActionEvent;
5 import javax.servlet.http.HttpServletRequest;
6
7 import cursojava.workshop.modelo.Cep;
8 import cursojava.workshop.modelo.Cliente;
9 import cursojava.workshop.modelo.MercadoException;
10 import cursojava.workshop.persistencia.CepDao;
11 import cursojava.workshop.persistencia.ClienteDao;
12 import cursojava.workshop.util.Utilitario;
13
14 public class ClienteControl {
15
16     private Cliente cliente = new Cliente();
17
18     private String email;
19     private String senha;
20     private String cepString;
21     private boolean autenticado;

```

```

22
23 public void autenticar(ActionEvent evento) {
24     ClienteDao clienteDao = new ClienteDao();
25     try {
26         cliente = clienteDao.autenticar(email, senha);
27         autenticado = true;
28     } catch (MercadoException e) {
29         Utilitario.addMensagemFaces(e.getMessage());
30         autenticado = false;
31     }
32 }
33
34 public void logout(ActionEvent evento) {
35     HttpServletRequest request = (HttpServletRequest)
36         FacesContext
37             .getCurrentInstance().getExternalContext().getRequest()
38             ;
39     request.getSession().invalidate();
40 }
41
42 public void consultarCep(ActionEvent evento) {
43     CepDao cepDao = new CepDao();
44     try {
45         Cep cep = cepDao.consultar(cepString);
46         cliente.setCep(cep);
47     } catch (MercadoException e) {
48         Utilitario.addMensagemFaces(e.getMessage());
49     }
50 }
51
52 public String confirmarCliente() {
53     if (!senha.equals(cliente.getSenha())) {
54         Utilitario.addMensagemFaces("As senhas devem ser iguais");
55         return null;
56     }
57
58     ClienteDao cdao = new ClienteDao();
59     try {
60         cdao.alterar(cliente);
61         Utilitario.addMensagemFaces("Confirmacao realizada com
62             sucesso");
63         autenticado = true;
64         return "confirmado";
65     } catch (MercadoException e) {
66         Utilitario.addMensagemFaces(e.getMessage());
67         return null;
68     }
69 }
70
71 public Cliente getCliente() {
72     return cliente;
73 }
74
75 public void setCliente(Cliente cliente) {
76     this.cliente = cliente;
77 }
78
79 public String getEmail() {
80     return email;
81 }
82
83 public void setEmail(String email) {

```

```

81         this.email = email;
82     }
83
84     public String getSenha() {
85         return senha;
86     }
87
88     public void setSenha(String senha) {
89         this.senha = senha;
90     }
91
92     public boolean isAutenticado() {
93         return autenticado;
94     }
95
96     public void setAutenticado(boolean autenticado) {
97         this.autenticado = autenticado;
98     }
99
100    public String getCepString() {
101        return cepString;
102    }
103
104    public void setCepString(String cepString) {
105        this.cepString = cepString;
106    }
107
108 }

```

Listing 18: Classe PedidoControl

```

1 package cursojava.workshop.controle;
2
3 import java.util.List;
4
5 import javax.faces.context.FacesContext;
6 import javax.faces.event.ActionEvent;
7 import javax.faces.model.DataModel;
8 import javax.faces.model.ListDataModel;
9 import javax.servlet.http.HttpServletRequest;
10
11 import cursojava.workshop.modelo.ItemPedido;
12 import cursojava.workshop.modelo.MercadoException;
13 import cursojava.workshop.modelo.Pedido;
14 import cursojava.workshop.modelo.Produto;
15 import cursojava.workshop.persistencia.PedidoDao;
16 import cursojava.workshop.persistencia.ProdutoDao;
17 import cursojava.workshop.util.Utilitario;
18
19 public class PedidoControl {
20
21     private String descricao;
22     private List<Produto> listaProdutos;
23     private DataModel produtos;
24     private Produto produto;
25     private Pedido pedido = new Pedido();
26     private DataModel itens;
27
28     public void listarProdutos(ActionEvent evento) {
29         ProdutoDao pdao = new ProdutoDao();
30         listaProdutos = pdao.listarPorDescricao(descricao);
31     }

```



```

32
33 public void confirmarCompra(ActionEvent evento) {
34     HttpServletRequest request = (HttpServletRequest)
35         FacesContext
36             .getCurrentInstance().getExternalContext().getRequest()
37             ;
38     ClienteControl clienteControl = (ClienteControl) request.
39         getSession()
40         .getAttribute("clienteControl");
41     pedido.setCliente(clienteControl.getCliente());
42     PedidoDao pedidoDao = new PedidoDao();
43     try {
44         pedidoDao.incluir(pedido);
45         Utilitario.addMensagemFaces("Pedido incluído com sucesso")
46             ;
47     } catch (MercadoException e) {
48         Utilitario.addMensagemFaces(e.getMessage());
49     }
50 }
51
52 public void excluirProdutoCarrinho(ActionEvent evento) {
53     ItemPedido item = (ItemPedido) itens.getRowData();
54     pedido.removerItem(item);
55 }
56
57 public void selecionarProduto(ActionEvent evento) {
58     produto = (Produto) produtos.getRowData();
59 }
60
61 public void comprar(ActionEvent evento) {
62     pedido.addProduto(produto);
63 }
64
65 public DataModel getProdutos() {
66     return produtos = new ListDataModel(listaProdutos);
67 }
68
69 public DataModel getItens() {
70     return itens = new ListDataModel(pedido.getItens());
71 }
72
73 public String getDescricao() {
74     return descricao;
75 }
76
77 public void setDescricao(String descricao) {
78     this.descricao = descricao;
79 }
80
81 public Produto getProduto() {
82     return produto;
83 }
84
85 public void setProduto(Produto produto) {
86     this.produto = produto;
87 }
88
89 public Pedido getPedido() {
90     return pedido;
91 }
92
93 public void setPedido(Pedido pedido) {

```

```

90         this.pedido = pedido;
91     }
92 }
93 }

```

5 A Camada de Visão

Listing 19: Tela do carrinho

```

1  <%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>
2  <%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
3  <%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
4  <%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
5
6  <html>
7
8  <head>
9  <link href="mercado.css" rel="stylesheet" type="text/css" />
10 </head>
11
12 <body>
13 <%@include file="menu.html"%>
14 <f:view>
15     <h:form>
16         <h:messages/>
17         <rich:dataTable value="#{pedidoControl.itens}" var="item"
18             width="700px">
19             <f:facet name="header">
20                 <h:outputText value="Seu Carrinho de Compras"/>
21             </f:facet>
22             <rich:column>
23                 <f:facet name="header">
24                     <h:outputText value="Foto"/>
25                 </f:facet>
26                 <h:graphicImage value="figuras/#{item.produto.foto}"
27                     styleClass="foto"/>
28             </rich:column>
29             <rich:column>
30                 <f:facet name="header">
31                     <h:outputText value="Descricao"/>
32                 </f:facet>
33                 <h:outputText value="#{item.produto.descricao}" />
34             </rich:column>
35             <rich:column>
36                 <f:facet name="header">
37                     <h:outputText value="Preco Unitario"/>
38                 </f:facet>
39                 <h:outputText value="#{item.produto.preco}">
40                     <f:convertNumber pattern="###,##0.00" />
41                 </h:outputText>
42             </rich:column>
43             <rich:column>
44                 <f:facet name="header">
45                     <h:outputText value="Quantidade"/>
46                 </f:facet>
47                 <h:inputText value="#{item.quantidade}" />
48             </rich:column>
49             <rich:column>
50                 <f:facet name="header">

```

```

51         <h:outputText value="Preco Total"></h:outputText>
52     </f:facet>
53     <h:outputText value="#{item.precoTotal}">
54         <f:convertNumber pattern="###,##0.00" />
55     </h:outputText>
56     <f:facet name="footer">
57         <h:outputText value="#{pedidoControl.pedido.precoTotal}">
58             <f:convertNumber pattern="###,##0.00" />
59         </h:outputText>
60     </f:facet>
61 </rich:column>
62 <rich:column>
63     <f:facet name="header">
64         <h:outputText value="Alterar"></h:outputText>
65     </f:facet>
66     <h:commandButton value="Alterar Quantidade"></h:
        commandButton>
67 </rich:column>
68 <rich:column>
69     <f:facet name="header">
70         <h:outputText value="Excluir"></h:outputText>
71     </f:facet>
72     <h:commandButton value="Excluir"
73         actionListener="#{pedidoControl.excluirProdutoCarrinho}">
        </h:commandButton>
74 </rich:column>
75 </rich:dataTable>
76 <h:commandButton action="comprar" value="Finalizar a compra"></
    h:commandButton>
77
78 </h:form>
79 </f:view>
80 </body>
81 </html>

```

Listing 20: Tela de Dados do cliente

```

1 <%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>
2 <%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
3 <%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
4 <%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
5
6 <html>
7
8 <head>
9 <link href="mercado.css" rel="stylesheet" type="text/css" />
10 </head>
11
12 <body>
13 <%@include file="menu.html"%>
14 <f:view>
15     <h:form>
16         <h:messages></h:messages>
17         <rich:panel>
18             <f:facet name="header">
19                 <h:outputText value="Cadastro de Cliente" />
20             </f:facet>
21             <h:panelGrid columns="2">
22                 <h:outputText value="Nome" />
23                 <h:inputText value="#{clienteControl.cliente.nome}" />
24                 <h:outputText value="Email" />
25                 <h:inputText value="#{clienteControl.cliente.email}" />

```

```

26
27     <h:outputText value="Cep:" />
28     <h:panelGroup>
29         <h:inputText value="#{clienteControl.cepString}" />
30         <h:commandButton
31             actionListener="#{clienteControl.consultarCep}"
32             value="Consultar" />
33     </h:panelGroup>
34
35     <h:outputText value="Senha:" />
36     <h:inputSecret value="#{clienteControl.cliente.senha}" />
37
38     <h:outputText value="Senha Novamente:" />
39     <h:inputSecret value="#{clienteControl.senha}" />
40
41     <h:outputText value="Logradouro" />
42     <h:outputText value="#{clienteControl.cliente.cep.
43         logradouro}" />
44     <h:outputText value="Bairro" />
45     <h:outputText value="#{clienteControl.cliente.cep.bairro}"
46     />
47     <h:outputText value="Município" />
48     <h:outputText
49         value="#{clienteControl.cliente.cep.municipio.descricao}"
50     />
51     <h:commandButton action="#{clienteControl.confirmarCliente}"
52     value="Confirmar" />
53 </h:panelGrid>
54 </rich:panel>
55
56 </h:form>
57 </f:view>
58 </body>
59 </html>

```

Listing 21: Tela de Detalhes do produto

```

1 <%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>
2 <%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
3 <%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
4 <%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
5
6 <html>
7
8 <head>
9 <link href="mercado.css" rel="stylesheet" type="text/css" />
10 </head>
11
12 <body>
13 <f:view>
14 <h:form>
15 <h:panelGrid columns="2">
16 <h:graphicImage value="figuras/#{pedidoControl.produto.foto}"
17 styleClass="fotoMaior"></h:graphicImage>
18 <h:panelGrid>
19 <h:outputText value="Codigo:#{pedidoControl.produto.id}" />
20 <h:outputText value="#{pedidoControl.produto.descricao}" />
21 <h:outputText value="#{pedidoControl.produto.preco}">
22 <f:convertNumber pattern="###,##0.00" />
23 </h:outputText>
24 <h:commandButton action="carrinho"

```

```

25         actionListener="#{pedidoControl.comprar}" value="Comprar"
26         </h:commandButton>
27     </h:panelGrid>
28 </h:panelGrid>
29 </h:form>
30 </f:view>
31 </body>
32 </html>

```

Listing 22: Tela de Login

```

1  <%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>
2  <%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
3  <%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
4  <%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
5
6  <html>
7
8  <head>
9  <link href="mercado.css" rel="stylesheet" type="text/css" />
10 </head>
11
12 <body>
13 <%@include file="menu.html"%>
14 <f:view>
15     <h:form>
16         <h:messages</h:messages>
17         <h:panelGrid columns="2" rendered="#{!clienteControl.
18             autenticado}">
19             <h:outputText value="Email:"</h:outputText>
20             <h:inputText value="#{clienteControl.email}"</h:inputText>
21             <h:outputText value="Senha:"</h:outputText>
22             <h:inputSecret value="#{clienteControl.senha}"</h:
23                 inputSecret>
24             <h:commandButton actionListener="#{clienteControl.autenticar}
25                 "
26                 value="Entrar"</h:commandButton>
27             <h:commandButton action="formularioCliente" value="Fazer
28                 Cadastro" />
29         </h:panelGrid>
30         <h:panelGroup rendered="#{clienteControl.autenticado}">
31             <rich:panel>
32                 <f:facet name="header">
33                     <h:outputText value="Seus Dados Pessoais" />
34                 </f:facet>
35                 <h:panelGrid columns="2">
36                     <h:outputText value="Nome" />
37                     <h:outputText value="#{clienteControl.cliente.nome}" />
38                     <h:outputText value="Email" />
39                     <h:outputText value="#{clienteControl.cliente.email}" />
40                     <h:outputText value="Logradouro" />
41                     <h:outputText value="#{clienteControl.cliente.cep.
42                         logradouro}" />
43                     <h:outputText value="Bairro" />
44                     <h:outputText value="#{clienteControl.cliente.cep.bairro}
45                         " />
46                     <h:outputText value="Município" />
47                     <h:outputText
48                         value="#{clienteControl.cliente.cep.municipio.descricao}
49                         " />
50                     <h:commandButton

```

```

44         actionListener="#{pedidoControl.confirmarCompra}"
45         value="Confirmar Compra" />
46     <h:commandButton action="formularioCliente"
47         value="Alterar Dados" />
48 </h:panelGrid>
49     <h:commandButton actionListener="#{clienteControl.logout}"
50         value="Sair"></h:commandButton>
51 </rich:panel>
52 </h:panelGroup>
53
54 </h:form>
55 </f:view>
56 </body>
57 </html>

```

Listing 23: Menu

```

1 <a href="pesquisa.faces">Pesquisa</a>
2 <a href="carrinho.faces">Carrinho</a>
3 <a href="login.faces">Login</a>

```

Listing 24: Tela da Pesquisa

```

1 <%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>
2 <%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
3 <%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
4 <%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
5
6 <html>
7
8 <head>
9 <link href="mercado.css" rel="stylesheet" type="text/css" />
10 </head>
11
12 <body>
13 <%@include file="menu.html"%>
14 <f:view>
15     <h:form>
16         <h:messages/>
17         Descricao: <h:inputText value="#{pedidoControl.
18             descricao}" />
19         <h:commandButton actionListener="#{pedidoControl.listarProdutos
20             }"
21             value="listar" />
22
23     <rich:dataTable value="#{pedidoControl.produtos}" var="produto"
24         width="700px">
25         <f:facet name="header">
26             <h:outputText value="Produtos Encontrados" />
27         </f:facet>
28         <rich:column>
29             <f:facet name="header">
30                 <h:outputText value="Foto" />
31             </f:facet>
32             <h:graphicImage value="figuras/#{produto.foto}"
33                 styleClass="foto" />
34         </rich:column>
35         <rich:column>
36             <f:facet name="header">
37                 <h:outputText value="Descricao" />
38             </f:facet>
39             <h:outputText value="#{produto.descricao}" />

```

```

38     </rich:column>
39     <rich:column>
40         <f:facet name="header">
41             <h:outputText value="Preço"></h:outputText>
42         </f:facet>
43         <h:outputText value="#{produto.preço}">
44             <f:convertNumber pattern="###,##0.00" />
45         </h:outputText>
46     </rich:column>
47     <rich:column>
48         <f:facet name="header">
49             <h:outputText value="Detalhes"></h:outputText>
50         </f:facet>
51         <h:commandButton value="Detalhes" action="detalhes"
52             actionListener="#{pedidoControl.selecionarProduto}"></h:
53             commandButton>
54     </rich:column>
55 </rich:dataTable>
56 </h:form>
57 </f:view>
58 </body>
59 </html>

```

Listing 25: Arquivo de estilos

```

1 .foto{
2
3     width: 100px;
4
5 }
6 .fotoMaior{
7
8     width: 300px;
9
10 }

```

Listing 26: Descritor de Implantação

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://
4     java.sun.com/xml/ns/javaee/web-app_2_5.xsd" xsi:schemaLocation=
5     "http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/
6     javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
7     <display-name>mercado</display-name>
8
9     <context-param>
10         <param-name>org.richfaces.SKIN</param-name>
11         <param-value>blueSky</param-value>
12     </context-param>
13
14     <filter>
15         <display-name>RichFaces Filter</display-name>
16         <filter-name>richfaces</filter-name>
17         <filter-class>org.ajax4jsf.Filter</filter-class>
18     </filter>
19
20     <filter-mapping>
21         <filter-name>richfaces</filter-name>
22         <servlet-name>Faces Servlet</servlet-name>
23         <dispatcher>REQUEST</dispatcher>

```

```

20         <dispatcher>FORWARD</dispatcher>
21         <dispatcher>INCLUDE</dispatcher>
22     </filter-mapping>
23
24
25
26     <welcome-file-list>
27         <welcome-file>index.html</welcome-file>
28         <welcome-file>index.htm</welcome-file>
29         <welcome-file>index.jsp</welcome-file>
30         <welcome-file>default.html</welcome-file>
31         <welcome-file>default.htm</welcome-file>
32         <welcome-file>default.jsp</welcome-file>
33     </welcome-file-list>
34     <servlet>
35         <servlet-name>Faces Servlet</servlet-name>
36         <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
37         <load-on-startup>1</load-on-startup>
38     </servlet>
39     <servlet-mapping>
40         <servlet-name>Faces Servlet</servlet-name>
41         <url-pattern>*.faces</url-pattern>
42     </servlet-mapping>
43 </web-app>

```

Listing 27: faces-config.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <faces-config xmlns="http://java.sun.com/xml/ns/javaee"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http
6         ://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd"
7     version="1.2">
8     <managed-bean>
9         <managed-bean-name>pedidoControl</managed-bean-name>
10         <managed-bean-class>
11             cursojava.workshop.controle.PedidoControl
12         </managed-bean-class>
13         <managed-bean-scope>session</managed-bean-scope>
14     </managed-bean>
15     <managed-bean>
16         <managed-bean-name>clienteControl</managed-bean-
17             name>
18         <managed-bean-class>
19             cursojava.workshop.controle.ClienteControl
20         </managed-bean-class>
21         <managed-bean-scope>session</managed-bean-scope>
22     </managed-bean>
23     <navigation-rule>
24         <display-name>pesquisa</display-name>
25         <from-view-id>/pesquisa.jsp</from-view-id>
26         <navigation-case>
27             <from-outcome>detalhes</from-outcome>
28             <to-view-id>/detalhe.jsp</to-view-id>
29         </navigation-case>
30     </navigation-rule>
31     <navigation-rule>
32         <display-name>detalhe</display-name>
33         <from-view-id>/detalhe.jsp</from-view-id>
34         <navigation-case>

```



```

33         <from-outcome>carrinho</from-outcome>
34         <to-view-id>/carrinho.jsp</to-view-id>
35     </navigation-case>
36 </navigation-rule>
37 <navigation-rule>
38     <display-name>login</display-name>
39     <from-view-id>/login.jsp</from-view-id>
40     <navigation-case>
41         <from-outcome>formularioCliente</from-
42             outcome>
43         <to-view-id>/cliente.jsp</to-view-id>
44     </navigation-case>
45 </navigation-rule>
46 <navigation-rule>
47     <display-name>cliente</display-name>
48     <from-view-id>/cliente.jsp</from-view-id>
49     <navigation-case>
50         <from-outcome>
51             confirmado
52         </from-outcome>
53         <to-view-id>/login.jsp</to-view-id>
54     </navigation-case>
55 </navigation-rule>
56 <navigation-rule>
57     <display-name>carrinho</display-name>
58     <from-view-id>/carrinho.jsp</from-view-id>
59     <navigation-case>
60         <from-outcome>comprar</from-outcome>
61         <to-view-id>/login.jsp</to-view-id>
62     </navigation-case>
63 </navigation-rule>
64 </faces-config>

```
