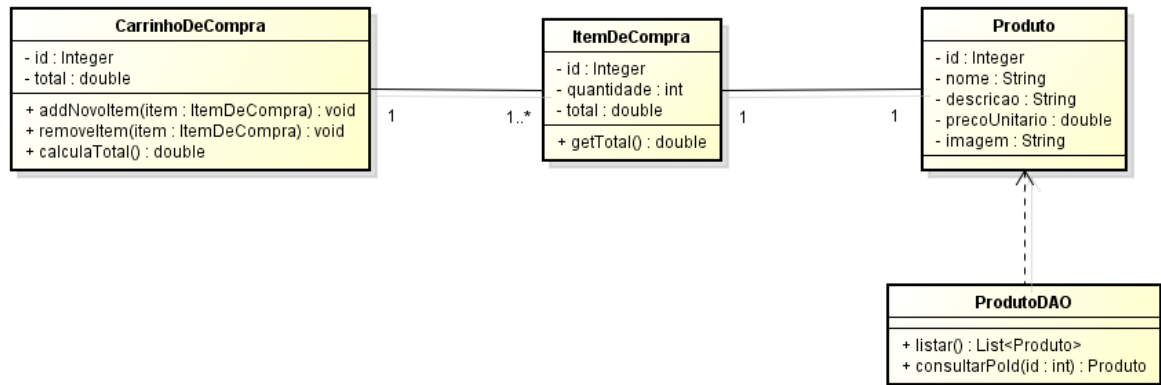


Aplicativos Web com Orientação a Objetos

Objetivo: Praticar o relacionamento entre classe “um para muitos” implementando um exemplo de carrinho de compras de produtos.

Diagrama de Classes - Model



1. Crie um banco de dados chamado produtos e uma tabela produto. Realize as inserções na tabela produto de acordo com os scripts abaixo.

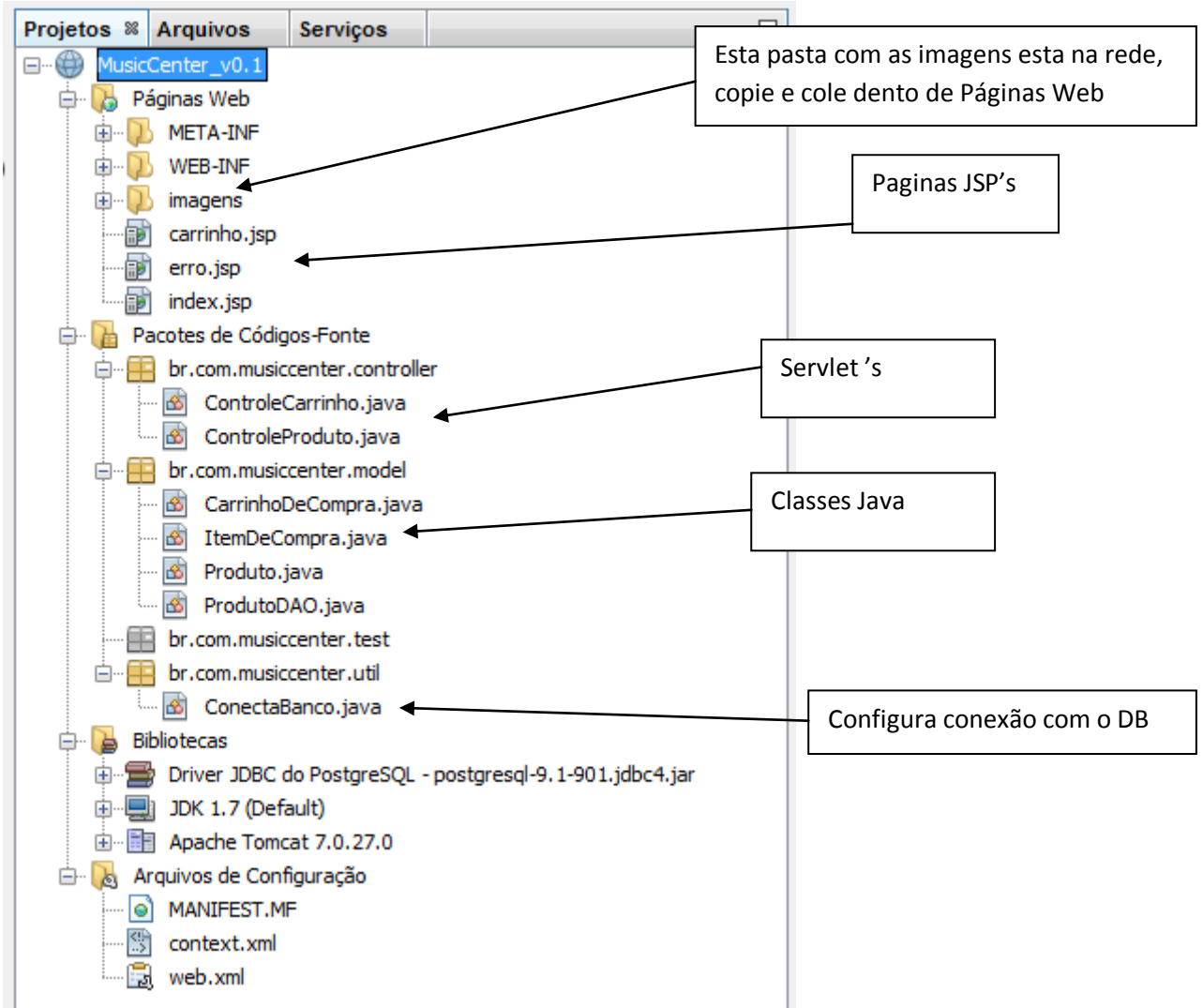
```
create database lojavirtual;
```

```
create table produto(
id serial,
nome varchar (100),
descricao varchar (255),
precoUnitario real,
imagem varchar (100),
Primary Key(id)
);
```

```
insert into produto (nome, descricao, precoUnitario, imagem) values ('afuche',
'prercursao', 1030, 'afuche.jpg');
insert into produto (nome, descricao, precoUnitario, imagem) values ('bateria',
'percursao', 1000, 'bateria.jpg');
insert into produto (nome, descricao, precoUnitario, imagem) values ('sax', 'sopro',
1500, 'sax.jpg');
insert into produto (nome, descricao, precoUnitario, imagem) values ('ektaratenor',
'sopro', 1200, 'ektaratenor.jpg');
insert into produto (nome, descricao, precoUnitario, imagem) values ('violao', 'cordas',
3200, 'violao.jpg');
```

Aplicativos Web com Orientação a Objetos

2. Crie um novo projeto web no NetBeans chamado CarrinhoProduto de acordo com a estrutura abaixo:



Aplicativos Web com Orientação a Objetos

3. Implemente a classe **ConectaBanco**, como o exemplo abaixo:

```
package br.com.musiccenter.util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConectaBanco {
    public static Connection getConexao() {
        Connection conexao = null;
        try {
            //driver que será utilizado
            Class.forName("org.postgresql.Driver");
            //cria um objeto de conexao com um banco especificado no caminho...
            conexao =
DriverManager.getConnection("jdbc:postgresql://127.0.0.1:5432/lojavirtual",
"postgres", "postgres");
        } catch (ClassNotFoundException erro1) {
            throw new RuntimeException(erro1);

        } catch (SQLException erro2) {
            throw new RuntimeException(erro2);

        }
        return conexao;
    }
}
```

4. Implemente a classe **Produto** com os atributos especificados no diagrama de classes e os métodos **get/set**.

Aplicativos Web com Orientação a Objetos

5. Implemente a classe ItemDeCompra como o exemplo abaixo:

```
6
7 public class ItemDeCompra {
8     private Integer id;
9     private Produto produto;
10    private int quantidade;
11    private double total;
12
13    public Integer getId() {
14        return id;
15    }
16
17    public Produto getProduto() {
18        return produto;
19    }
20
21    public void setProduto(Produto produto) {
22        this.produto = produto;
23    }
24
25    public int getQuantidade() {
26        return quantidade;
27    }
28
29    public void setQuantidade(int quantidade) {
30        this.quantidade = quantidade;
31    }
32    public double getTotal() {
33        this.total = this.quantidade * this.produto.getPrecoUnitario();
34        return total;
35    }
36 }
```

Aplicativos Web com Orientação a Objetos

6. Implemente a classe CarrinhoDeCompra como o exemplo abaixo:

```
7  import java.util.ArrayList;
8  import java.util.Iterator;
9  import java.util.List;
10
11  public class CarrinhoDeCompra {
12      private Integer id;
13      private List<ItemDeCompra> itens;
14      private double total;
15
16      public void addNovoItem(ItemDeCompra item){
17          if(this.itens==null){
18              this.itens = new ArrayList<ItemDeCompra>();
19          }
20          this.itens.add(item);
21      }
22      public void removerItem(ItemDeCompra itemRemove){
23          for(Iterator i = itens.iterator();i.hasNext();){
24              ItemDeCompra item = (ItemDeCompra) i.next();
25              if(item.getProduto().getId() == itemRemove.getProduto().getId()){
26                  i.remove();
27              }
28          }
29      }
30      public double calculaTotal(){
31          double vttotal=0;
32          for(ItemDeCompra item : this.itens){
33              vttotal += item.getTotal();
34          }
35          this.total = vttotal;
36          return total;
37      }
38
39      public Integer getId() {
40          return id;
41      }
42
43      public List<ItemDeCompra> getItens() {
44          return itens;
45      }
46
47      public double getTotal() {
48          return total;
49      }
50
51  }
```

Aplicativos Web com Orientação a Objetos

7. Implemente a classe ProdutoDAO com os métodos especificados no diagrama de classe.

```
import br.com.musiccenter.util.ConectaBanco;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

public class ProdutoDAO {

    private static final String SELECTALL = "select * from produto";
    private static final String SELECTID = "select * from produto where id = ?";

    public ArrayList<Produto> listar() {
        Connection conexao = null;
        ArrayList<Produto> listaProduto = new ArrayList<Produto>();
        try {
            conexao = ConectaBanco.getConexao();

            PreparedStatement pstmt = conexao.prepareStatement(SELECTALL);
            ResultSet rs = pstmt.executeQuery();

            while (rs.next()) {
                Produto prod = new Produto();
                prod.setId(rs.getInt("id"));
                prod.setNome(rs.getString("nome"));
                prod.setDescricao(rs.getString("descricao"));
                prod.setPrecoUnitario(rs.getDouble("precoUnitario"));
                prod.setImagem(rs.getString("imagem"));

                listaProduto.add(prod);
            }
        } catch (SQLException ex1) {
            throw new RuntimeException(ex1);
        } finally {
            try {
                if (conexao != null) {
                    conexao.close();
                }
            } catch (SQLException ex1) {
                throw new RuntimeException(ex1);
            }
        }
    }
}
```

Aplicativos Web com Orientação a Objetos

```
    }

    //retorna a lista
    return listaProduto;
}

public Produto consultarPorId(int id) {
    Connection conexao = null;
    Produto produto = new Produto();
    try {
        conexao = ConectaBanco.getConexao();
        PreparedStatement pstmt = conexao.prepareStatement(SELECTID);
        pstmt.setInt(1, id);
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            produto.setId(rs.getInt("id"));
            produto.setNome(rs.getString("nome"));
            produto.setDescricao(rs.getString("descricao"));
            produto.setPrecoUnitario(rs.getDouble("precoUnitario"));
            produto.setImagem(rs.getString("imagem"));

        }

    } catch (SQLException ex1) {
        throw new RuntimeException(ex1);
    } finally {
        try{
            if(conexao!=null){
                conexao.close();
            }
        }catch(SQLException ex2){
            throw new RuntimeException(ex2);
        }
    }

    return produto;
}
}
```

Aplicativos Web com Orientação a Objetos

8. Implemente o Servlet ControleProduto como o exemplo abaixo:

```
package br.com.musiccenter.controller;

import br.com.musiccenter.model.Produto;
import br.com.musiccenter.model.ProdutoDAO;
import java.io.IOException;
import java.util.ArrayList;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ControleProduto extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        try {
            String acao = request.getParameter("acao");
            if (acao.equals("listaProdutos")) {

                //monta uma lista para exibição na pagina principal
                ArrayList<Produto> produtos = new ProdutoDAO().listar();

                //armazena os produto na requisição
                request.setAttribute("produtos", produtos);

                //invia para index.jsp
                request.getRequestDispatcher("/index.jsp").forward(request, response);

            }

        } catch (Exception erro) {
            request.setAttribute("erro", erro);
            request.getRequestDispatcher("erro.jsp").forward(request, response);
        }

    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}
```


Aplicativos Web com Orientação a Objetos

```
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
}
```

9. Implemente a Servlet ControleCarrinho como o exemplo a baixo:

```
package br.com.musiccenter.controller;

import br.com.musiccenter.model.CarrinhoDeCompra;
import br.com.musiccenter.model.ItemDeCompra;
import br.com.musiccenter.model.Produto;
import br.com.musiccenter.model.ProdutoDAO;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class ControleCarrinho extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        try {
            String acao = request.getParameter("acao");

            if (acao.equals("addProduto")) {
                //recupera o id do produto que deve ser add no carrinho
                int idProduto = Integer.parseInt(request.getParameter("idProduto"));
                //flag para controle de inserção de novos produtos no carrinho
                boolean existe = false;

                //recupera a sessão pertencente ao request
                HttpSession sessao = request.getSession();

                //recupera um carrinho de produtos da sessão
                //se não existe um carrinho na sessão o valor será igual a null
                CarrinhoDeCompra carrinho = (CarrinhoDeCompra) sessao.getAttribute("carrinho");

                //verifica se já exista um carrinho na sessão
                if (carrinho == null) {
                    //cria um carrinho
                }
            }
        }
    }
}
```

Aplicativos Web com Orientação a Objetos

```
carrinho = new CarrinhoDeCompra();
sessao.setAttribute("carrinho", carrinho);
}
//verifica se o produto existe no carrinho
if (carrinho.getItens() != null) {
    for (ItemDeCompra item : carrinho.getItens()) {
        if (item.getProduto().getId() == idProduto) {
            //incrementa a quantidade
            item.setQuantidade(item.getQuantidade() + 1);
            existe = true;
        }
    }
}
//se não existe o item ou produto, cria um novo
if (existe == false) {
    //encontra o produto no banco
    Produto produto = new ProdutoDAO().consultarPorId(idProduto);

    //cria um novo item
    ItemDeCompra novotem = new ItemDeCompra();
    novotem.setProduto(produto);
    novotem.setQuantidade(1);
    //adiciona novo item
    carrinho.addNovotem(novotem);
}

//carrega a pagina do carrinho de compras
request.getRequestDispatcher("/carrinho.jsp").forward(request, response);

} //fim addProduto
else if (acao.equals("removeProduto")) {

    //recupera a sessão pertencente ao request
    HttpSession sessao = request.getSession();

    //recupera um carrinho de produtos da sessão
    CarrinhoDeCompra carrinho = (CarrinhoDeCompra) sessao.getAttribute("carrinho");

    //recupera o id do produto
    int idProduto = Integer.parseInt(request.getParameter("idProduto"));

    ItemDeCompra itemRemove = new ItemDeCompra();
    Produto prodRemove = new Produto();
    prodRemove.setId(idProduto);
    itemRemove.setProduto(prodRemove);

    carrinho.removeItem(itemRemove);
}
```

Aplicativos Web com Orientação a Objetos

```
//carrega a pagina do carrinho de compras
request.getRequestDispatcher("/carrinho.jsp").forward(request, response);

} else if (acao.equals("cancelaCompra")) {
    //recupera a sessão pertencente ao request
    HttpSession sessao = request.getSession();

    //remove o carrinho da sessão
    sessao.removeAttribute("carrinho");

    //redireciona para pagina principal
    response.sendRedirect("index.jsp");

}

} catch (Exception erro) {
    request.setAttribute("erro", erro);
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
}
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
}
```

10. Implemente o JSP index.jsp como o exemplo abaixo

```
<%@page import="br.com.musiccenter.model.Produto"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
</head>
<body>
```

Aplicativos Web com Orientação a Objetos

```
<%
// Recupera os produtos.
java.util.List<Produto> produtos = (java.util.List<Produto>)
request.getAttribute("produtos");
if(produtos == null) {
    request.getRequestDispatcher("/ControleProduto?acao=listaProdutos").forward(request,
response);
}
%>

<h1>Lista de Produtos</h1>
<table border="0" cellpadding="5" align="center">
    <%
        int contadorColuna=1;
        for(Produto produto : produtos){
            //se é o primeiro produto, cria o inicio da linha
            if(contadorColuna == 1) {
                out.println("<tr>");
            }
            %>
            <td align="center" valign="bottom"> "/> <br/>
                <%=produto.getNome()%><br/>
                <a href="ControleCarrinho?acao=addProduto&idProduto=<%=produto.getId()%>">
Comprar</a></td>
            <%
                //se é o último produto, exibir o término da linha
                if(contadorColuna == 3){
                    out.println("</tr>");
                    contadorColuna =0;
                }
                //atualiza o contador de colunas
                contadorColuna ++;
            %>
        }//fim do for

    %>
</table>

</body>
</html>
```

Aplicativos Web com Orientação a Objetos

11. Implemente o JSP carrinho.jsp como o exemplo abaixo:

```
<%@page import="br.com.musiccenter.model.ItemDeCompra"%>
<%@page import="br.com.musiccenter.model.CarrinhoDeCompra"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Carrinho de Compras</title>
  </head>
  <body>
    <h1>Carrinho de Compras!</h1>
    <table border="1" cellpadding="2" >
      <tr>
        <td bgcolor="#000088"><font color="white">Excluir</font></td>
        <td bgcolor="#000088"><font color="white">Item</font></td>
        <td bgcolor="#000088"><font color="white">QTD</font></td>
        <td bgcolor="#000088"><font color="white">Preço Unitário</font></td>
        <td bgcolor="#000088"><font color="white">Total Item</font></td>
        <td bgcolor="#000088"><font color="white">+1</font></td>
      </tr>
      <%
        //recupera os produtos do carrinho da sessao
        CarrinhoDeCompra carrinho = (CarrinhoDeCompra) session.getAttribute("carrinho");
        for(ItemDeCompra item : carrinho.getItems()){

          %>
          <tr>
            <td><a
href="ControleCarrinho?acao=removeProduto&idProduto=<%=item.getProduto().getId()%>">
X</td>
            <td><%=item.getProduto().getNome() %></td>
            <td><%=item.getQuantidade() %></td>
            <td><%=item.getProduto().getPrecoUnitario() %></td>
            <td><%=item.getTotal()%></td>
            <td><a
href="ControleCarrinho?acao=addProduto&idProduto=<%=item.getProduto().getId()%>">+</a>
</td>

          </tr>

          <%
            }
          %>

        </table>

        <strong>Valor Total: <%=carrinho.calculaTotal() %></strong><br/>

        <a href="index.jsp"> Continue comprando</a><br/>
```

Aplicativos Web com Orientação a Objetos

```
        <a href="ControleCarrinho?acao=cancelaCompra">Cancelar comprar</a>
    </body>
</html>
```

12. Implemente o JSP erro.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Erro!</title>
    </head>
    <%
        Exception erro = (Exception) request.getAttribute("erro");

    %>

    <body>
        <h1>erro! <%=erro%></h1>
    </body>
</html>
```

13. Execute o teste a aplicação !!!