

PROGRAMAÇÃO ORIENTADA A OBJETOS

DAO - Cadastrar

```
// método para cadastrar um empregado
public void cadastrar(Empregado empregado) {
    Conexao conexao = new Conexao();
    connection = conexao.conectar();
    sql = "insert into java_empregado values(empregado_sequence.nextval"
        + ", ?, ?, ?)";

    try {
        ps = connection.prepareStatement(sql);
        ps.setInt(1, empregado.getDepartamento().getId());
        ps.setString(2, empregado.getNome());
        ps.setString(3, empregado.getCpf());
        ps.execute();
        ps.close();
        conexao.desconectar();
    } catch (SQLException e) {
        System.out.println("erro ao cadastrar empregado\n"+e);
    }
}
```

JSP - Cadastrar

```
<body>
<% List<Departamento> lista = new DepartamentoDAO().listar(); %>
<div class="container">
<div class="row">
<form class="col s12" action="cadastro">
<div class="row">
<div class="input-field col s6">
<input id="nome" name="nome" type="text">
<label for="nome">Nome</label>
</div>
</div>
<div class="row">
<div class="input-field col s6">
<input id="cpf" name="cpf" type="text">
<label for="cpf">CPF</label>
</div>
</div>
<div class="row">
<label>Departamento</label>
<select name="departamento">
<option value="" disabled selected>Escolha o departamento</option>
<% for(Departamento dep : lista) { %>
<option value="<%= dep.getId() %>"><%= dep.getNome() %></option>
<% } %>
</select>
</div>
<div class="row">
<button class="btn waves-effect waves-teal" type="submit">Cadastrar</button>
</div>
</form>
</div>
```

DAO – Listar

```
public class EmpregadoDAO extends DAO {

// método para retornar os dados dos empregados
public List<Empregado> listar() {
    List<Empregado> lista = new ArrayList<Empregado>();
    Conexao conexao = new Conexao();
    connection = conexao.conectar();
    Empregado empregado;
    Departamento departamento;

    sql = "select E.nome, E.cpf, D.nome as dnome from java_empregado E, "
        + "java_departamento D where E.id_departamento = D.id order by E.nome";

    try {
        ps = connection.prepareStatement(sql);
        rs = ps.executeQuery();
        while(rs.next()) {
            departamento = new Departamento();
            departamento.setNome(rs.getString("dnome"));
            empregado = new Empregado();
            empregado.setCpf(rs.getString("cpf"));
            empregado.setNome(rs.getString("nome"));
            empregado.setDepartamento(departamento);
            lista.add(empregado);
        }
        ps.close();
        conexao.desconectar();
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return lista;
}
```

Servelet - Cadastrar

```
public CadastroServlet() {
    super();
    // TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
 */
protected void service(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String nome = request.getParameter("nome");
    String cpf = request.getParameter("cpf");
    Integer id = Integer.parseInt(request.getParameter("departamento"));
    Departamento departamento = new Departamento();
    departamento.setId(id);
    Empregado empregado = new Empregado();
    empregado.setCpf(cpf);
    empregado.setNome(nome);
    empregado.setDepartamento(departamento);
    new EmpregadoDAO().cadastrar(empregado);

    //redireciona para a página index.jsp
    response.sendRedirect("index.jsp");
}
```

```
DAO.java X
1 package br.fiap.dao;
2
3 import java.sql.Connection;
4
5 public class DAO {
6     protected static Connection connection;
7     protected ResultSet rs;
8     protected PreparedStatement ps;
9     protected String sql;
10
11     public DAO() {
12         if (connection == null) {
13             connection = new Conexao().conectar();
14         }
15     }
16
17     protected PreparedStatement prepareStatement(String sql) throws SQLException {
18         return connection.prepareStatement(sql);
19     }
20 }
21
22
23
24
25
26
```

Imports jsp

```
<%@page import="br.fiap.modelo.entidade.Departamento"%>
<%@page import="br.fiap.dao.conexao.DepartamentoDAO"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.util.List"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

Imports html/css

```
<%@page import="br.fiap.modelo.entidade.Departamento"%>
<%@page import="br.fiap.dao.conexao.DepartamentoDAO"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.util.List"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
<body>
<% List<Empregado> lista = new EmpregadoDAO().listar(); %>
<div class="container">
<table class="striped">
<thead>
<tr>
<th>CPF</th>
<th>Nome</th>
<th>Departamento</th>
</tr>
</thead>
<tbody>
<% for(Empregado empregado : lista) { %>
<tr>
<td><%= empregado.getCpf() %></td>
<td><%= empregado.getNome() %></td>
<td><%= empregado.getDepartamento().getNome() %></td>
</tr>
<% } %>
</tbody>
</table>
</div>
</body>
</html>
```

Alterar – JSP

```
<div class="container">
  <div class="content">

    <!--FORMULÁRIO DE ALTERAÇÃO-->
    <div id="cadastro">
      <form method="post" action="../alterar">
        <h1>Alteração de Dados</h1>
        <% Integer id = Integer.parseInt(request.getParameter("id"));
           Contato contato = new ContatoDAO().pesquisarId(id);

           %>
        <p>
          <input id="id" name="id" type="hidden" value="<%= contato.getId() %>" />
        </p>
        <p>
          <label for="nome">Nome</label>
          <input id="nome" name="nome" type="text" value="<%= contato.getNome() %>" />
        </p>
        <p>
          <label for="email">Email</label>
          <input id="email" name="email" type="email" value="<%= contato.getEmail() %>" />
        </p>
        <p>
          <label for="senha">Senha</label>
          <input id="senha" name="senha" type="password" value="" />
        </p>

        <p>
          <label for="fone">Telefone</label>
          <input id="fone" name="fone" type="text" value="<%= contato.getTelefone() %>" />
        </p>

        <p>
          <label for="data">Data Nascimento</label>
          <input id="data" name="data" type="date" value="<%= contato.getDataNascimento() %>" />
        </p>

        <p>
          <input type="submit" value="Alterar" />
        </p>
      </form>
    </div>
  </div>
</div>
```

Listar outra forma

```
<div class="container">
  <table class="bordered striped centered highlight responsive-table">
    <thead>
      <tr>
        <th>ID</th>
        <th>Nome</th>
        <th>Email</th>
        <th>Telefone</th>
        <th>Data de Nascimento</th>
        <th>Ação</th>
      </tr>
    </thead>
    <tbody>
      <%
        List<Contato> lista = new ContatoDAO().listar();
        for (Contato contato : lista) {
          %>
          <tr>
            <td><%=contato.getId()%></td>
            <td><%=contato.getNome()%></td>
            <td><%=contato.getEmail()%></td>
            <td><%=contato.getTelefone()%></td>
            <td><%=contato.getDataNascimento()%></td>
            <td><a href="form/formAlterar.jsp?id=<%=contato.getId()%>">
              <button class="btn waves-effect waves-teal" type="submit">Alterar</button>
              <button class="material-icons right">send</button></a>
            </td>
          </tr>
          <%
        }
      <%>
    </tbody>
  </table>
</div>
```

```
public class Util {
  public LocalDate formatarData(String dataString) {
    String[] dataAux = dataString.split("-");
    dataString = "";
    dataString = dataAux[2] + "/" + dataAux[1] + "/" + dataAux[0];

    DateTimeFormatter formato = DateTimeFormatter.ofPattern("dd/MM/yyyy");
    LocalDate data = LocalDate.parse(dataString, formato);
    return data;
  }
}
```

Alterar - DAO

```
// método para alterar os dados de um contato
public void alterar(Contato contato) {
  Conexao conexao = new Conexao();
  connection = conexao.conectar();
  sql = "update java_contato set nome = ?, email = ?, "
    + "telefone = ?, "
    + "dataNascimento = ? where id = ?";

  try {
    ps = connection.prepareStatement(sql);
    ps.setString(1, contato.getNome());
    ps.setString(2, contato.getEmail());
    ps.setString(3, contato.getTelefone());
    ps.setDate(4, Date.valueOf(contato.getDataNascimento()));
    ps.setInt(5, contato.getId());
    ps.execute();
    ps.close();
    conexao.desconectar();
  }
  catch (SQLException e) {
    System.out.println("Erro ao alterar dados do contato na "
      + "base de dados\n" + e);
  }
}
```

Consultar - DAO

```
public Integer obterPedidoId() {
  Integer id = 0;
  Conexao conexao = new Conexao();
  connection = conexao.conectar();
  sql = "select max(pedido_id) from java_pedido";

  try {
    ps = connection.prepareStatement(sql);
    rs = ps.executeQuery();
    if (rs.next()) {
      id = rs.getInt("max(pedido_id)");
    }
    rs.close();
    ps.close();
    connection.close();
    conexao.desconectar();
  } catch (SQLException e) {
    System.out.println("erro ao pesquisar o último id inserido\n" + e);
  }

  return id;
}
```

```
// método para retornar uma lista de contatos
public List<Contato> listar() {
  List<Contato> lista = new ArrayList<Contato>();
  Contato contato;
  Conexao conexao = new Conexao();
  connection = conexao.conectar();
  sql = "select * from java_contato";

  try {
    ps = connection.prepareStatement(sql);
    rs = ps.executeQuery();
    while (rs.next()) {
      contato = new Contato();
      contato.setId(rs.getInt("id"));
      contato.setNome(rs.getString("nome"));
      contato.setEmail(rs.getString("email"));
      contato.setSenha(rs.getString("senha"));
      contato.setTelefone(rs.getString("telefone"));
      contato.setDataNascimento(rs.getDate(
        "dataNascimento").toLocalDate());
      lista.add(contato);
    }
    ps.close();
    conexao.desconectar();
  }
  catch (SQLException e) {
    System.out.println("Erro ao inserir na base de dados\n" + e);
  }

  return lista;
}
```

```
11 // método para inserir um pedido na base de dados. O método retorna true para saber
12 * se pedido foi inserido ou não. Se o pedido não foi inserido, seus detalhes não serão
13 * inseridos na tabela detalhe pedido
14 */
15 public boolean inserir(Pedido pedido) {
16   boolean inseriu = false;
17   Conexao conexao = new Conexao();
18   connection = conexao.conectar();
19   sql = "insert into java_pedido values(pedido_sequence.nextval, ?, ?, ?)";
20
21   try {
22     ps = connection.prepareStatement(sql);
23     ps.setString(1, pedido.getNomeContato());
24     ps.setString(2, pedido.getEnderecoContato());
25     ps.setDate(3, Date.valueOf(pedido.getData())); // converte a data para Data (sql)
26     ps.execute();
27     inseriu = true;
28     ps.close();
29     connection.close();
30     conexao.desconectar();
31   } catch (SQLException e) {
32     System.out.println("Erro ao inserir produto\n" + e);
33   }
34
35   return inseriu;
36 }
37
```