

EducaCiência FastCode

Fala Galera,

- Artigo: 30/2020 Data: Outubro/2020
- Público Alvo: Desenvolvedores – Iniciantes
- Tecnologia: Java
- Tema: JSP – criando Método Update
- Link: <https://github.com/perucello/DevFP>

Neste artigo, daremos continuidade no Projeto CRUD JSP que iniciamos no artigo 27/2020.

Lembrando que criaremos uma série de artigos para explanarmos o CRUD em um projeto Java Server Page ou JSP.

Neste artigo 30/2020 chegamos a reta final do nosso CRUD e traremos uma maneira simples de criar o Método Atualizar em um projeto JSP.

Para este ambiente , já temos criado nosso Banco de Dados onde chamamos de EducaJSP.

Nosso ambiente consiste em:

⇒ Banco de Dados MySql

```
1
2 • create database EducaJSP;
3 • use EducaJSP;
4 • CREATE TABLE pessoa (
5     id INT auto_increment PRIMARY KEY not null,
6     nome VARCHAR(50),
7     endereco VARCHAR(50),
8     cidade VARCHAR(50),
9     estado char(2),
10    email VARCHAR(50));
11
12 • insert into pessoa value(
13     '1',
14     'Fulano de Tal',
15     'Rua das Hortas',
16     'Sao Paulo',
17     'SP',
18     'fulano@detal.com.br'
19 );
20
21 • select * from pessoa;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	nome	endereco	cidade	estado	email
▶	1	Fulano de Tal	Rua das Hortas	Sao Paulo	SP	fulano@detal.com.br
•		NULL	NULL	NULL	NULL	NULL

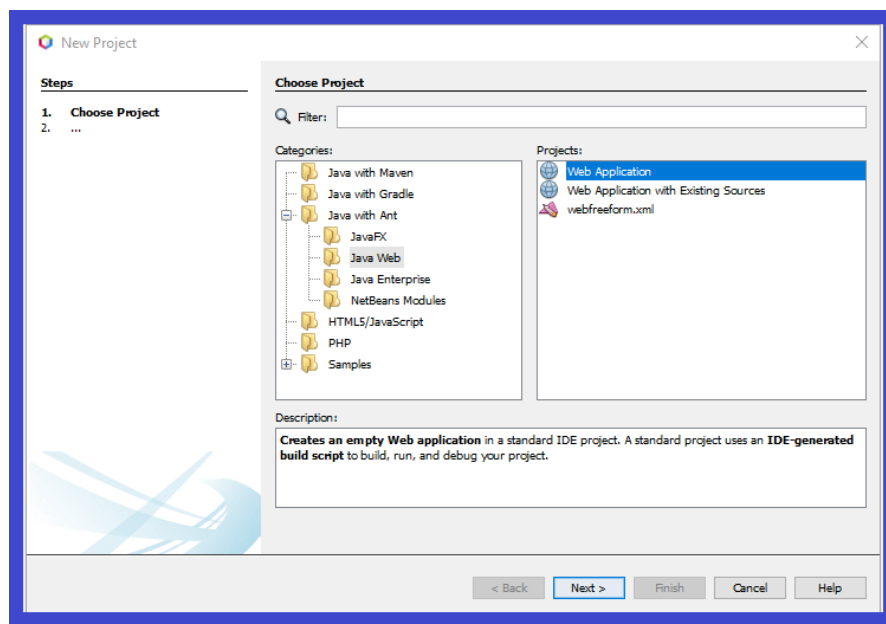
Utilizaremos da IDE Netbeans 11 para elaborarmos nosso propósito.

Para saber como foi criado o projeto, sugiro dar uma olhada no artigo:

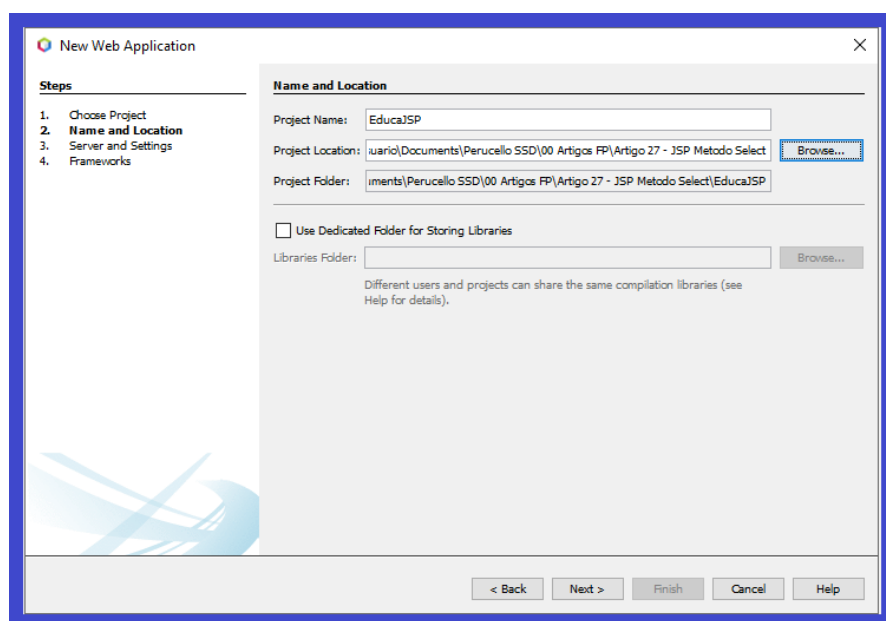
- ⇒ 27/2020 – abordamos o método Select
- ⇒ 28/2020 – abordamos o método Insert
- ⇒ 29/2020 – abordamos o método Deletar

Para isso, criamos nosso projeto JSP que se chama EducaJSP:

- ⇒ Novo Projeto



- ⇒ Usamos Servidor Glassfish 4.1.1



New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

Server: GlassFish Server Add...

Java EE Version: Java EE 7 W...
Note: Source Level 7 will be set for Java EE 7 project.

Context Path: /EducaJSP

< Back Next > Finish Cancel Help

New Web Application

Steps

1. Choose Project
2. Name and Location
3. Server and Settings
4. **Frameworks**

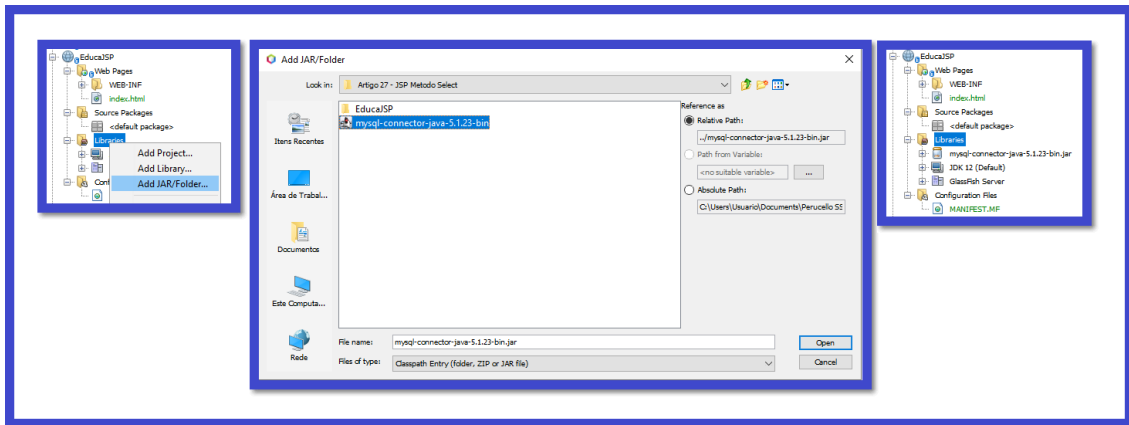
Frameworks

Select the frameworks you want to use in your web application.

☐ Spring Web MVC
☐ JavaServer Faces
☐ Struts 1.3.10

< Back Next > Finish Cancel Help

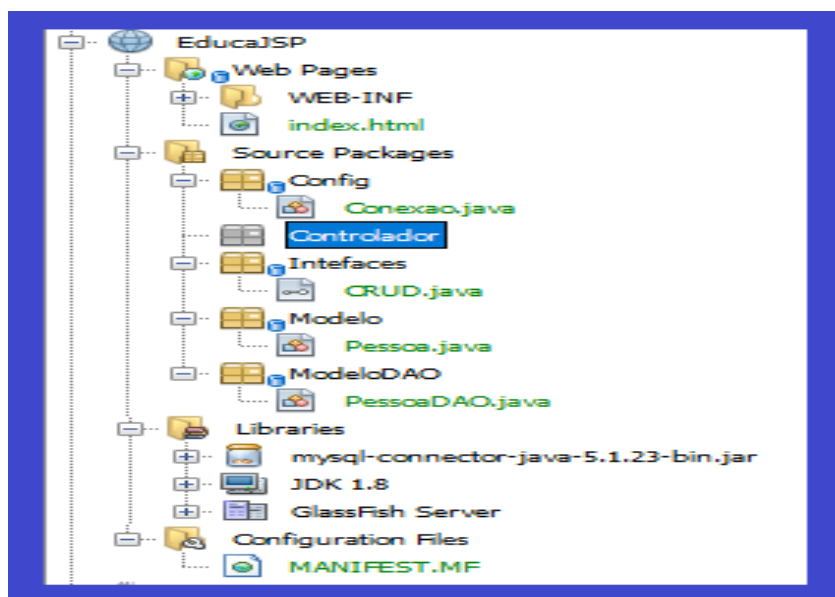
⇒ Para conexão com Banco de Dados , utilizamos do driver jdbc :



⇒ **Foi alterado para Java 8 no artigo 27/2020**

⇒ Criamos nossos pacotes sendo:

- Config** => nesse pacote nossa classe chamada Conexao.java que será responsável pela conexão com Banco de Dados
- Controlador** => nosso Controlador.java que será nosso Servlet
- Interface** => nossa interface CRUD dos Métodos
- Modelo** => na classe Pessoa.java nossos Construtores e nossos Métodos Getters/Setters
- ModeloDAO** => nossa classe PessoaDAO.java que receberá nossos métodos que atuarão junto ao Banco de Dados



Vamos lá então:

Conexao.java

```
1 package Config;
2
3 import java.sql.*;
4
5 public class Conexao {
6
7     Connection con;
8
9     private static final String DATABASE = "EducaJSP";
10    private static final String PORT = "3306";
11    private static final String HOST = "jdbc:mysql://localhost:" + "/" + DATABASE;
12    private static final String DRIVER = "com.mysql.jdbc.Driver";
13    private static final String SSL = "?useTimezone=true&serverTimezone=UTC&useSSL=false";
14    private static final String URL = "jdbc:mysql://localhost:3306" + "/" + DATABASE + SSL;
15    private static final String USR = "root";
16    private static final String FWD = "";
17
18    public void Conexao() {
19        try {
20            Class.forName(DRIVER);
21            con = DriverManager.getConnection(URL, USR, FWD);
22            System.out.println("Banco de Dados => " + DATABASE);
23            System.out.println("Host => " + HOST);
24            System.out.println("Porta => " + PORT);
25            System.out.println("Driver => " + DRIVER);
26            System.out.println("URL => " + URL);
27            System.out.println("SSL => " + SSL);
28            System.out.println("Usuario => " + USR);
29            System.out.println("Senha => " + FWD);
30        } catch (ClassNotFoundException | SQLException e) {
31            System.err.println("Error = " + e);
32            System.out.println("Erro de conexao com Banco de Dados - verificar se está ativo - Mysql ! " + e);
33        }
34    }
35
36    public static void Desconectar(Connection con) {
37        try {
38            if (con != null) {
39                con.close();
40            }
41        } catch (SQLException e) {
42            System.out.println("ERRO : " + e.getMessage());
43        }
44    }
45
46    public Connection getConnection() {
47        return con;
48    }
49 }
```

Pessoa.java

```

1 package Modelo;
2
3 public class Pessoa {
4     int id;
5     String email;
6     String nome;
7     String endereco;
8     String cidade;
9
10    public Pessoa() {
11    }
12
13    public Pessoa(int id, String email, String nome, String endereco, String cidade) {
14        this.id = id;
15        this.email = email;
16        this.nome = nome;
17        this.endereco = endereco;
18        this.cidade = cidade;
19    }
20
21    public int getId() {
22        return id;
23    }
24
25    public void setId(int id) {
26        this.id = id;
27    }
28
29    public String getEmail() {
30        return email;
31    }
32
33    public void setEmail(String email) {
34        this.email = email;
35    }
36
37    public String getNome() {
38        return nome;
39    }
40
41    public void setNome(String nome) {
42        this.nome = nome;
43    }
44
45    public String getEndereco() {
46        return endereco;
47    }
48
49    public void setEndereco(String endereco) {
50        this.endereco = endereco;
51    }
52
53    public String getCidade() {
54        return cidade;
55    }
56
57    public void setCidade(String cidade) {
58        this.cidade = cidade;
59    }
60
61 }
62
63

```

PessoaDAO.java

```

1 package ModeloDAO;
2
3 import Config.Conexao;
4 import Modelo.Pessoa;
5 import java.sql.Connection;
6 import java.sql.PreparedStatement;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9 import java.util.ArrayList;
10 import java.util.List;
11 import Interfaces.CRUD;
12
13 public class PessoaDAO implements CRUD {
14
15     Conexao cn = new Conexao();
16     Connection con;
17     PreparedStatement ps;
18     ResultSet rs;
19     Pessoa a = new Pessoa();
20
21     public PessoaDAO() { ...4 lines }
22
23     @Override
24     public List listar() { ...24 lines }
25
26     @Override
27     public Pessoa list(int id) { ...22 lines }
28
29     @Override
30     public boolean add(Pessoa a) { ...17 lines }
31
32     @Override
33     public boolean eliminar(int id) { ...13 lines }
34
35 }
36
37

```

CRUD.java

⇒ Interface

```

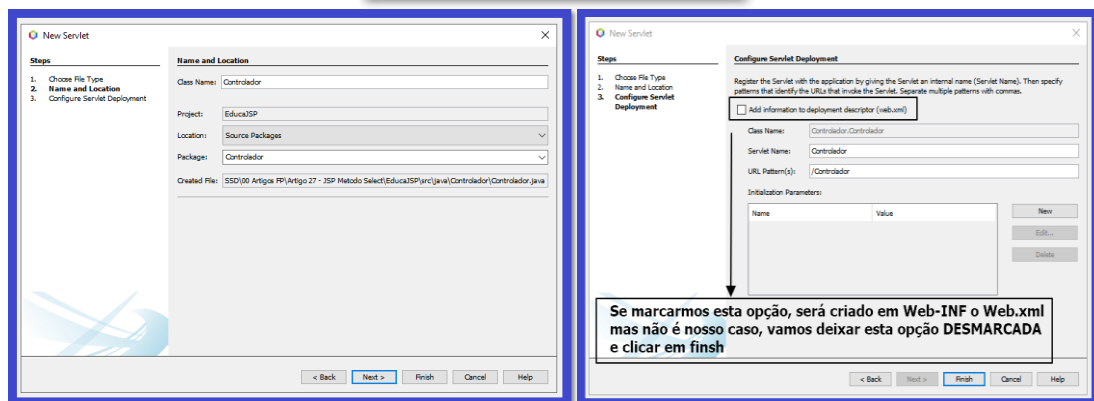
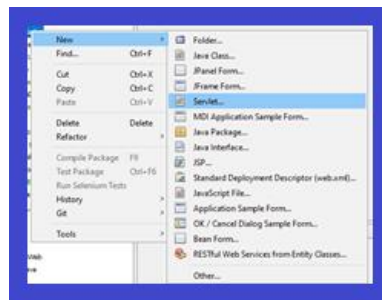
1
2 package Interfaces;
3
4 import Modelo.Pessoa;
5 import java.util.List;
6
7
8 public interface CRUD {
9     public List listar();
10    public Pessoa list(int id);
11    public boolean add(Pessoa a);
12    public boolean eliminar(int id);
13 }
14

```

Neste cenário, ao criarmos nosso Servlet, optamos em duas maneiras sendo:

- Criando com web.xml no pacote Web-INF ou
- Se anotamos no Controlador.

Iremos optar por anotar , e vamos seguir os seguintes passos:



```

1  package Controlador;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5  import javax.servlet.ServletException;
6  import javax.servlet.annotation.WebServlet;
7  import javax.servlet.http.HttpServlet;
8  import javax.servlet.http.HttpServletRequest;
9  import javax.servlet.http.HttpServletResponse;
10
11  /** ...4 lines */
12  @WebServlet(name = "Controlador", urlPatterns = {"/Controlador"})
13  public class Controlador extends HttpServlet {
14
15      /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 lines */
16      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
17          throws ServletException, IOException {
18          response.setContentType("text/html;charset=UTF-8");
19          try (PrintWriter out = response.getWriter()) {
20              /* TODO output your page here. You may use following sample code. */
21              out.println("<!DOCTYPE html>");
22              out.println("<html>");
23              out.println("<head>");
24              out.println("<title>Servlet Controlador</title>");
25              out.println("</head>");
26              out.println("<body>");
27              out.println("<h1>Servlet Controlador at " + request.getContextPath() + "</h1>");
28              out.println("</body>");
29              out.println("</html>");
30          }
31      }
32
33      // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
34      /** Handles the HTTP <code>GET</code> method ...8 lines */
35      @Override
36      protected void doGet(HttpServletRequest request, HttpServletResponse response)
37          throws ServletException, IOException {
38          processRequest(request, response);
39      }
40
41      /** Handles the HTTP <code>POST</code> method ...8 lines */
42      @Override
43      protected void doPost(HttpServletRequest request, HttpServletResponse response)
44          throws ServletException, IOException {
45          processRequest(request, response);
46      }
47
48      /** Returns a short description of the servlet ...5 lines */
49      @Override
50      public String getServletInfo() {
51          return "Short description";
52      }
53  }
54  </editor-fold>
55
56
57
58

```

O único método que manipularemos será o doGet onde nos artigos 27/2020, 28/2020 e 29/2020 onde criamos as condições para listar, inserir e deletar.

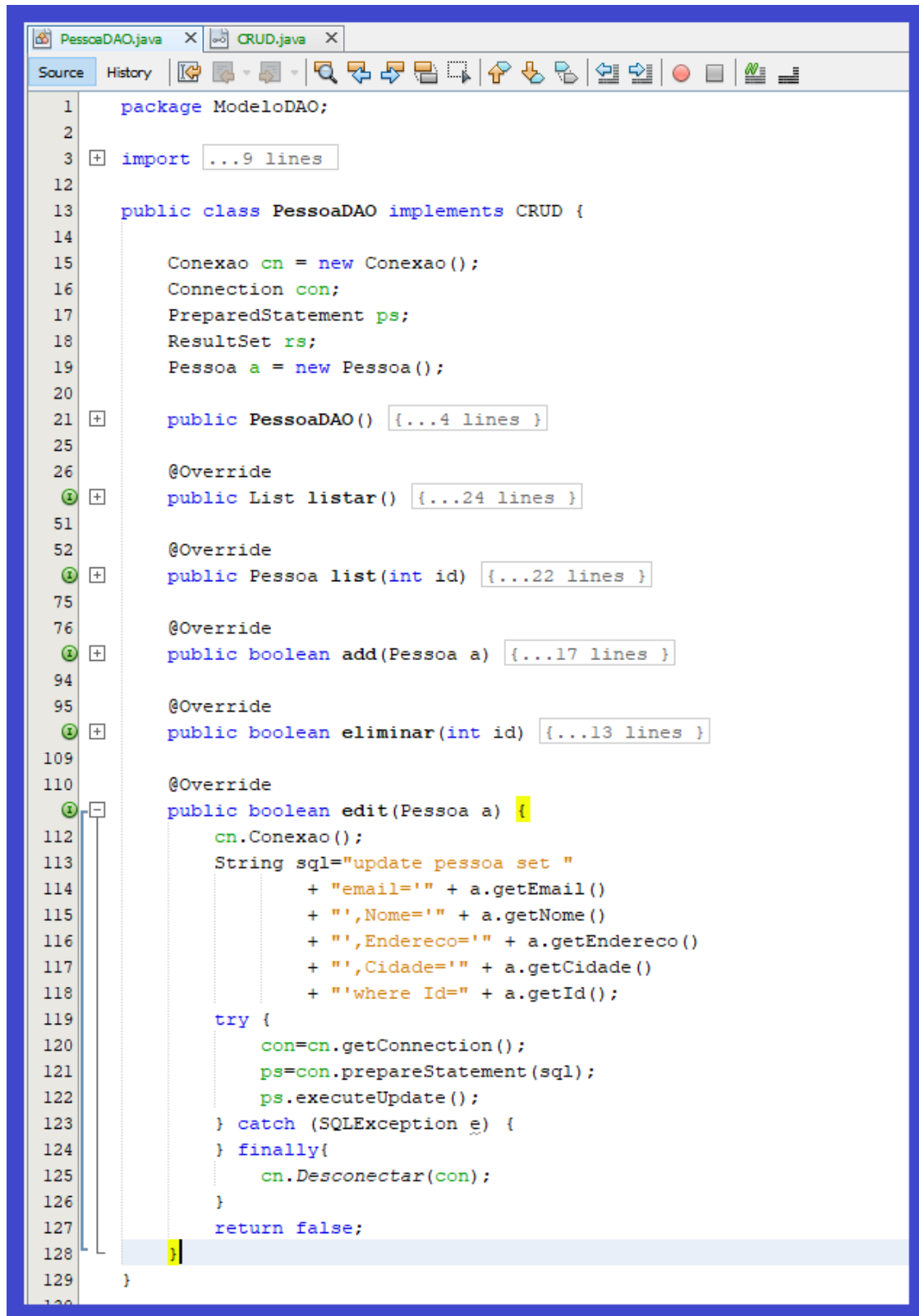
Pronto, agora que revisamos o projeto vamos pôr a mão na massa !

Para incluirmos nosso método Delete, visto o projeto que já temos , teremos que realizar as seguintes atividades:

- Incluir método Atualizar no PessoaDAO
- Incluir método na nossa Interface CRUD
- Incluir a rotina do método Atualizar no nosso Servlet, ou seja, no nosso doGet do Controlador
- Criar o jsp para Atualizar
- Adicionar a opção Atualizar no listar.jsp

VAMOS NESSA !

- Incluir método Atualizar no PessoaDAO

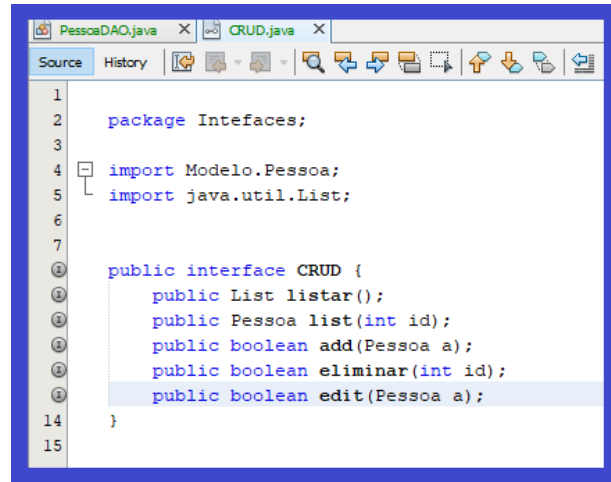


```

1  package ModeloDAO;
2
3  import ...9 lines
12
13  public class PessoaDAO implements CRUD {
14
15      Conexao cn = new Conexao();
16      Connection con;
17      PreparedStatement ps;
18      ResultSet rs;
19      Pessoa a = new Pessoa();
20
21  public PessoaDAO() {...4 lines }
25
26  @Override
27  public List listar() {...24 lines }
51
52  @Override
53  public Pessoa list(int id) {...22 lines }
75
76  @Override
77  public boolean add(Pessoa a) {...17 lines }
94
95  @Override
96  public boolean eliminar(int id) {...13 lines }
109
110  @Override
111  public boolean edit(Pessoa a) {
112      cn.Conexao();
113      String sql="update pessoa set "
114          + "email='" + a.getEmail()
115          + "',Nome='" + a.getNome()
116          + "',Endereco='" + a.getEndereco()
117          + "',Cidade='" + a.getCidade()
118          + "'where Id=" + a.getId();
119
120      try {
121          con=cn.getConnection();
122          ps=con.prepareStatement(sql);
123          ps.executeUpdate();
124      } catch (SQLException e) {
125      } finally{
126          cn.Desconectar(con);
127      }
128      return false;
129  }
130

```

- Incluir método na nossa Interface CRUD



```

1
2 package Interfaces;
3
4 import Modelo.Pessoa;
5 import java.util.List;
6
7 public interface CRUD {
8     public List listar();
9     public Pessoa list(int id);
10    public boolean add(Pessoa a);
11    public boolean eliminar(int id);
12    public boolean edit(Pessoa a);
13 }
14
15

```

- Incluir a rotina do método Atualizar no nosso Servlet, ou seja, no nosso doGet do Controlador

```

1 package Controlador;
2
3 import ...10 lines
4
5 /**...4 lines */
6 @WebServlet(name = "Controlador", urlPatterns = {"/Controlador"})
7 public class Controlador extends HttpServlet {
8
9     String listar = "vistas/listar.jsp";
10    String add = "vistas/add.jsp";
11    String edit = "vistas/edit.jsp";
12    Pessoa a = new Pessoa();
13    PessoaDAO dao = new PessoaDAO();
14    int id;
15
16    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 lines */
17    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
18        throws ServletException, IOException { ...15 lines }
19
20    /** Handles the HTTP <code>GET</code> method ...8 lines */
21    @Override
22    protected void doGet(HttpServletRequest request, HttpServletResponse response)
23        throws ServletException, IOException {
24        //processRequest(request, response);
25        String acesso = "";
26        String action = request.getParameter("action");
27        if (action.equalsIgnoreCase("listar")) {
28            acesso = listar;
29        } else if (action.equalsIgnoreCase("add")) {
30            acesso = add;
31        } else if (action.equalsIgnoreCase("Adicionar")) {
32            String nome = request.getParameter("txtNome");
33            String endereco = request.getParameter("txtEndereco");
34            String cidade = request.getParameter("txtCidade");
35            String email = request.getParameter("txtEmail");
36            a.setNome(nome);
37            a.setEndereco(endereco);
38            a.setCidade(cidade);
39            a.setEmail(email);
40            dao.add(a);
41            acesso = listar;
42        } else if (action.equalsIgnoreCase("eliminar")) {
43            id = Integer.parseInt(request.getParameter("id"));
44            a.setId(id);
45            dao.eliminar(id);
46            acesso = listar;
47        } else if (action.equalsIgnoreCase("editar")) {
48            request.setAttribute("id-a", request.getParameter("id"));
49            acesso = edit;
50        } else if (action.equalsIgnoreCase("Atualizar")) {
51            id = Integer.parseInt(request.getParameter("txtid"));
52            String email = request.getParameter("txtEmail");
53            String nome = request.getParameter("txtNome");
54            String endereco = request.getParameter("txtEndereco");
55            String cidade = request.getParameter("txtCidade");
56            a.setId(id);
57            a.setNome(nome);
58            a.setEndereco(endereco);
59            a.setCidade(cidade);
60            a.setEmail(email);
61            dao.edit(a);
62            acesso = listar;
63        }
64        RequestDispatcher vista = request.getRequestDispatcher(acesso);
65        vista.forward(request, response);
66    }
67
68    /** Handles the HTTP <code>POST</code> method ...8 lines */
69    @Override
70    protected void doPost(HttpServletRequest request, HttpServletResponse response)
71        throws ServletException, IOException {
72        processRequest(request, response);
73    }
74
75    /** Returns a short description of the servlet ...5 lines */
76    @Override
77    public String getServletInfo() { ...3 lines }
78 }

```

- Criar edit.jsp para que tenhamos um html para realizarmos o processo de Atualizar

```

1
2 <%@page import="Modelo.Pessoa"%>
3 <%@page import="ModeloDAO.PessoaDAO"%>
4 <%@page contentType="text/html" pageEncoding="UTF-8"%>
5 <!DOCTYPE html>
6 <html>
7 <head>
8 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
9 <link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
10 <title>JSP EducaCiencia</title>
11 </head>
12 <body>
13 <div class="container">
14 <div class="col-lg-6">
15 <%
16 PessoaDAO dao = new PessoaDAO();
17 int id=Integer.parseInt((String)request.getAttribute("id-a"));
18 Pessoa a = (Pessoa)dao.list(id);
19 %>
20 <h1>Modificar - EducaCiencia</h1>
21 <form action="Controlador">
22 <!--
23 Email:<br>
24 <input class="form-control" type="text" name="txtEmail" value="<%= a.getEmail() %>"><br>-->
25 Nomes: <br>
26 <input class="form-control" type="text" name="txtNome" value="<%= a.getNome() %>"><br>
27 Endereço <br>
28 <input class="form-control" type="text" name="txtEndereco" value="<%= a.getEndereco() %>"><br>
29 Cidade <br>
30 <input class="form-control" type="text" name="txtCidade" value="<%= a.getCidade() %>"><br>
31 Email:<br>
32 <input class="form-control" type="text" name="txtEmail" value="<%= a.getEmail() %>"><br>
33 <input type="hidden" name="txtid" value="<%= a.getId() %>">
34 <input class="btn btn-primary" type="submit" name="accion" value="Atualizar">
35 <a href="Controlador?accion=listar">Voltar</a>
36 </form>
37 </div>
38 </div>
39 </div>
40 </body>
41 </html>
42

```

○ Adicionar opção deletar em listar.jsp

```

13 </head>
14 <body>
15 <div class="container">
16 <h1>EducaCiência</h1>
17 <a class="btn btn-success" href="Controlador?acao=add">Adicionar</a>
18 <a class="btn btn-success" href="index.jsp">Voltar</a>
19 <br>
20 <br>
21 <table class="table table-bordered">
22 <thead>
23 <tr>
24 <th class="text-center">ID</th>
25 <!-- <th class="text-center">EMAIL</th-->
26 <th class="text-center">NOME</th>
27 <th class="text-center">ENDERECO</th>
28 <th class="text-center">CIDADE</th>
29 <th class="text-center">EMAIL</th>
30 <th class="text-center">AÇÃO</th>
31 </tr>
32 </thead>
33 <%
34 PessoaDAO dao = new PessoaDAO();
35 List<Pessoa> list = dao.listar();
36 Iterator<Pessoa> iter = list.iterator();
37 Pessoa a = null;
38 while (iter.hasNext()) {
39     a = iter.next();
40 }
41 %>
42 <tbody>
43 <tr>
44 <td class="text-center"><%= a.getId() %></td>
45 <td class="text-center"><%= a.getNome() %></td>
46 <td class="text-center"><%= a.getEndereco() %></td>
47 <td class="text-center"><%= a.getCidade() %></td>
48 <td class="text-center"><%= a.getEmail() %></td>
49 <td class="text-center"><%= a.getNome() %></td-->
50 <td class="text-center">
51 <a class="btn btn-danger" href="Controlador?acao=eliminar&id=<%= a.getId() %>">Remover</a>
52 <a class="btn btn-warning" href="Controlador?acao=editar&id=<%= a.getId() %>">Editar</a>
53 </td>
54 </tr>
55 <tr>
56 <td colspan="6"><%= "" %></td>
57 </tr>
58 </tbody>
59 </table>
60 </div>
61 </body>
62 </html>

```



Feito isso, se tudo estiver OK, basta iniciarmos a aplicação em RUN.

1

2

3

4

Nota-se que os códigos funcionaram corretamente, no entanto, saliento que os artigos 27/2020, 28/2020 e 29/2020 são baseados no curso de Java que ministro na Escola Evolua – Ensino Profissionalizante e como propósito de ajuda à comunidade, estamos trazendo parte da didática em forma de artigo comunitário e assim podemos contribuir com a comunidade Tecnológica como um todo.

Agradeço imensamente a Diretoria da Escola Evolua de Sumaré.

Os códigos estarão disponíveis no Git.

Até mais !
Espero ter ajudado !

