

EducaCiência FastCode

Fala Galera,

- Artigo: 29/2020 Data: Outubro/2020
- Público Alvo: Desenvolvedores – Iniciantes
- Tecnologia: Java
- Tema: JSP – criando Método Delete
- 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 29/2020 traremos uma maneira simples de criar o Método Deletar em um projeto JSP, e daremos continuidade no CRUD no próximo artigo (30).

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

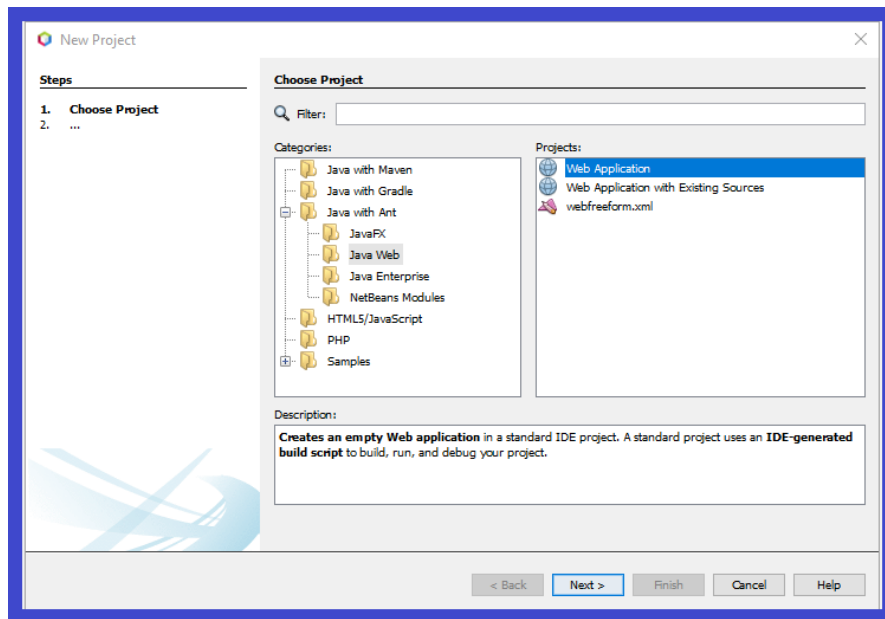
	id	nome	endereco	cidade	estado	email
▶	1	Fulano de Tal	Rua das Hortas	Sao Paulo	SP	fulano@detal.com.br
*	NULL	NULL	NULL	NULL	NULL	NULL

Utilizaremos da IDE Netbeans 11 para elaborarmos nosso propósito.
Para saber como foi criado o projeto, sugiro dar uma olhada no artigo:

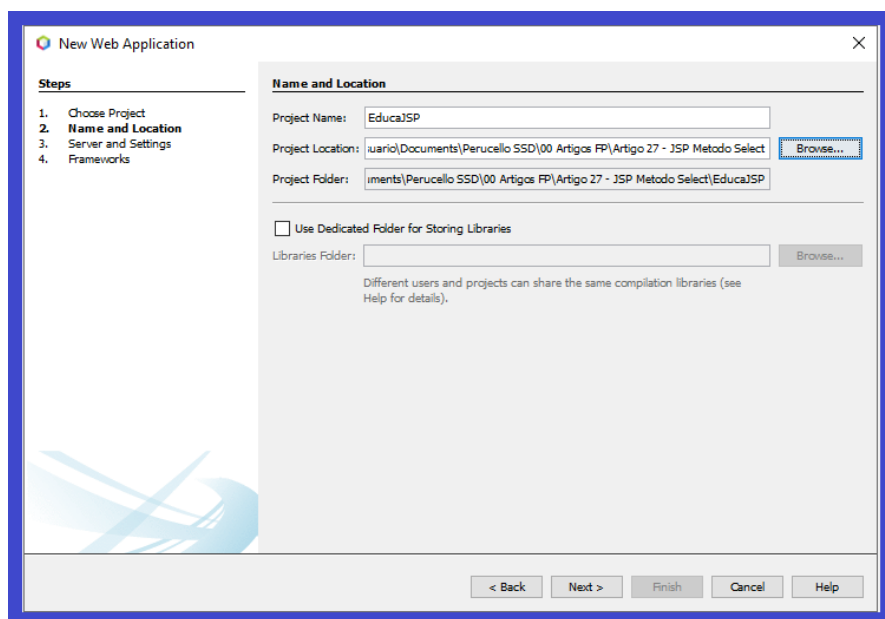
- ⇒ 27/2020 – abordamos o método Select
- ⇒ 28/2020 – abordamos o método Insert

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**

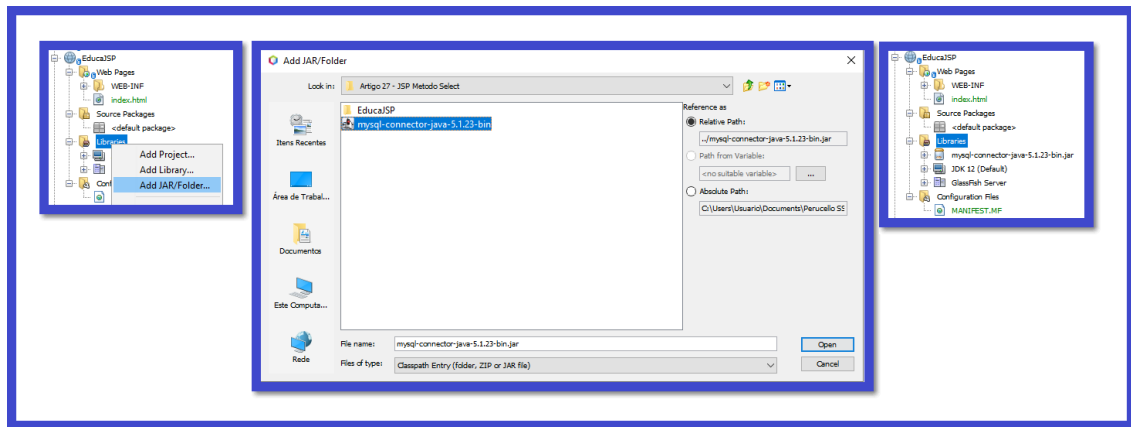
Frameworks

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

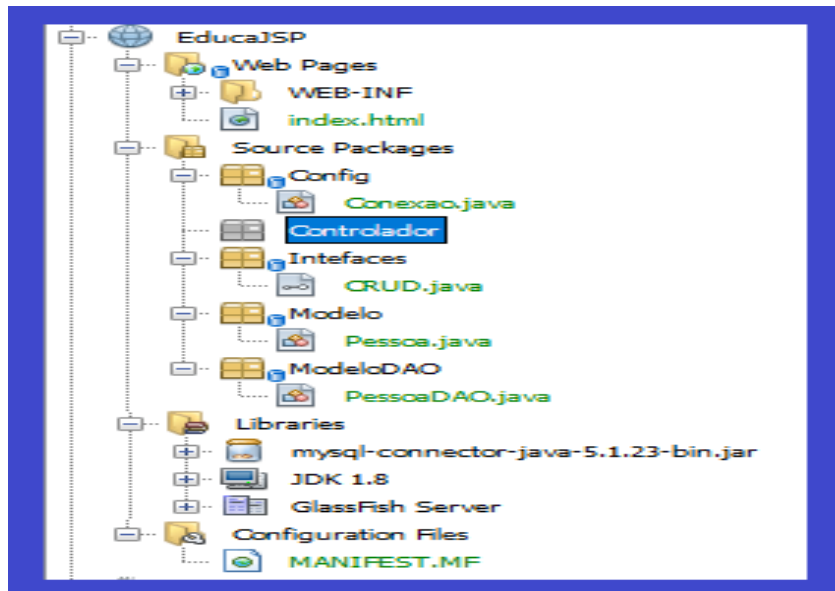
⇒ Para conexão com Banco de Dados , utilizamos do driver jdbc :



⇒ **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 getConexao() {
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() {
22         super();
23         con = this.cn.getConnection();
24     }
25
26     @Override
27     public List listar() { ...24 lines }
28
29     @Override
30     public Pessoa list(int id) { ...22 lines }
31
32     @Override
33     public boolean add(Pessoa a) { ...17 lines }
34
35 }
36
```

CRUD.java

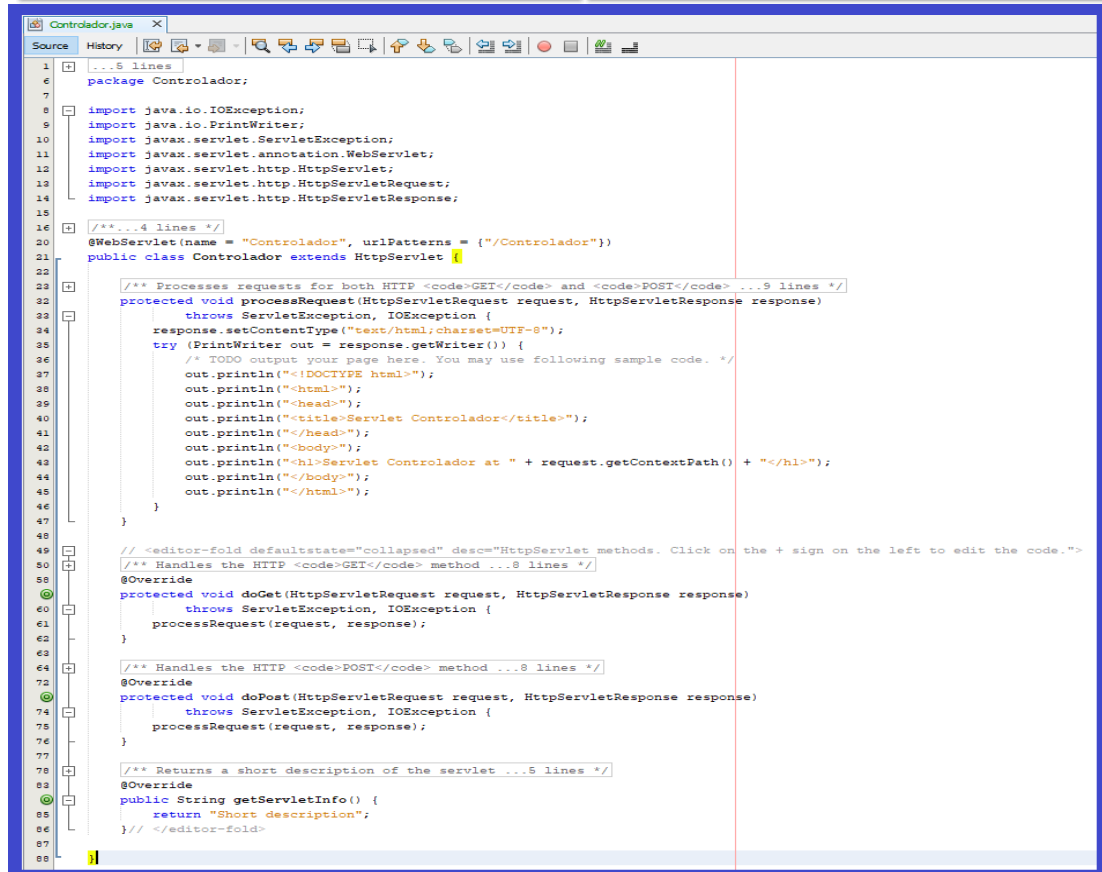
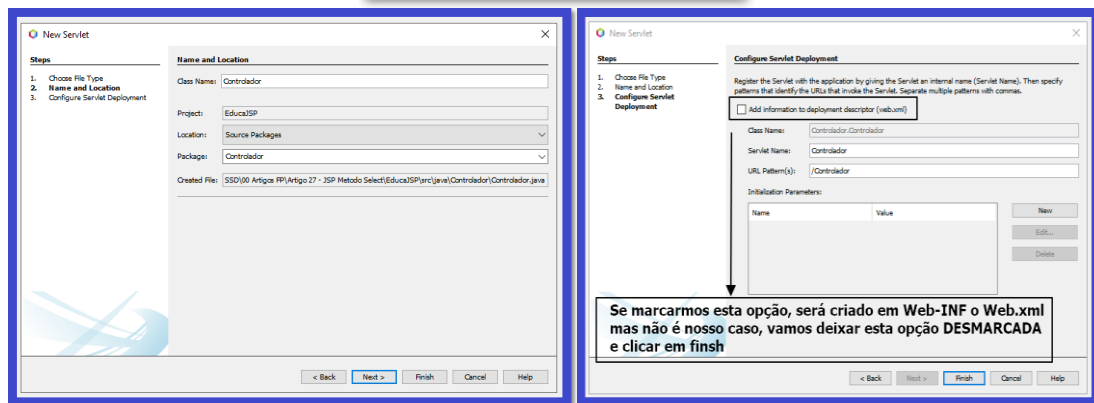
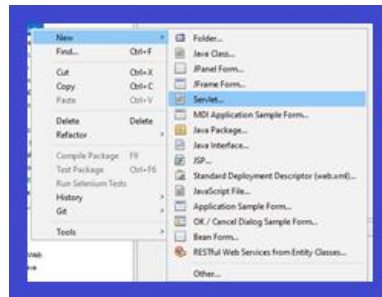
⇒ Interface

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

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:



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

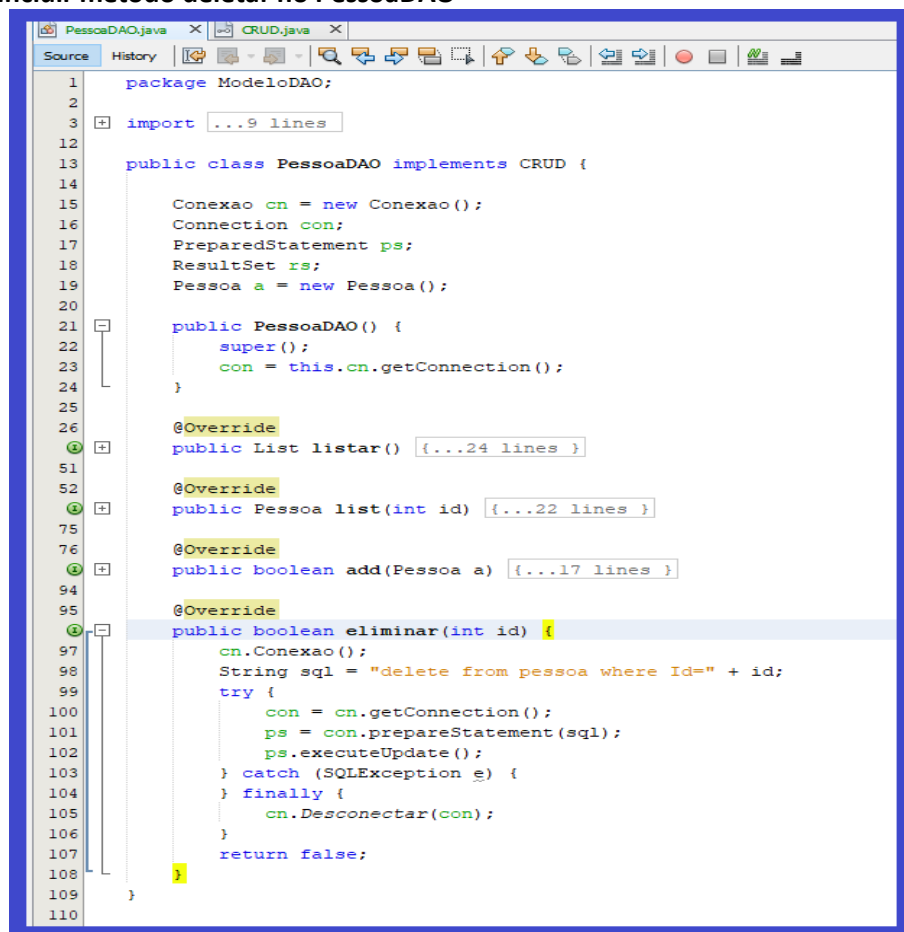
Pronto, agora que revisamos o projeto vamos pôr a mão na massa !

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

- a) Incluir método Deletar no PessoaDAO
- b) Incluir método na nossa Interface CRUD
- c) Incluir a rotina do método Deletar no nosso Servlet, ou seja, no nosso doGet do Controlador
- d) Criar o jsp para Deletar
- e) Adicionar a opção deletar no listar.jsp

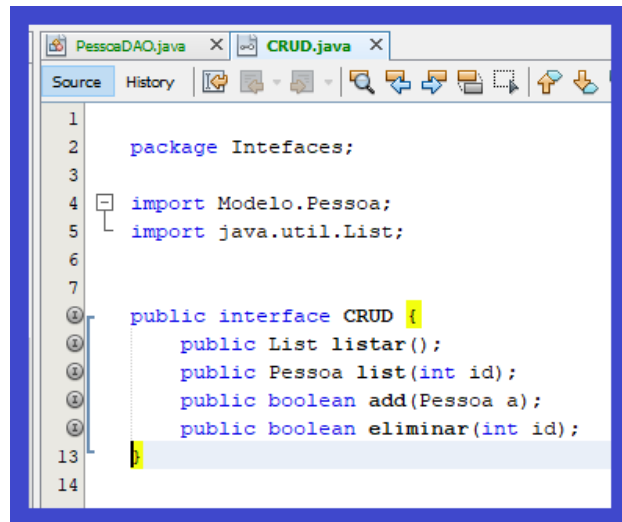
VAMOS NESSA !

- Incluir método deletar no PessoaDAO



```
1 package ModeloDAO;
2
3 import ...9 lines
4
12 public class PessoaDAO implements CRUD {
13
14     Conexao cn = new Conexao();
15     Connection con;
16     PreparedStatement ps;
17     ResultSet rs;
18     Pessoa a = new Pessoa();
19
20
21     public PessoaDAO() {
22         super();
23         con = this.cn.getConnection();
24     }
25
26     @Override
27     public List listar() {...24 lines}
28
29     @Override
30     public Pessoa list(int id) {...22 lines}
31
32     @Override
33     public boolean add(Pessoa a) {...17 lines}
34
35     @Override
36     public boolean eliminar(int id) {
37         cn.Conexao();
38         String sql = "delete from pessoa where Id=" + id;
39         try {
40             con = cn.getConnection();
41             ps = con.prepareStatement(sql);
42             ps.executeUpdate();
43         } catch (SQLException e) {
44         } finally {
45             cn.Desconectar(con);
46         }
47         return false;
48     }
49 }
```


- Incluir método na nossa Interface CRUD



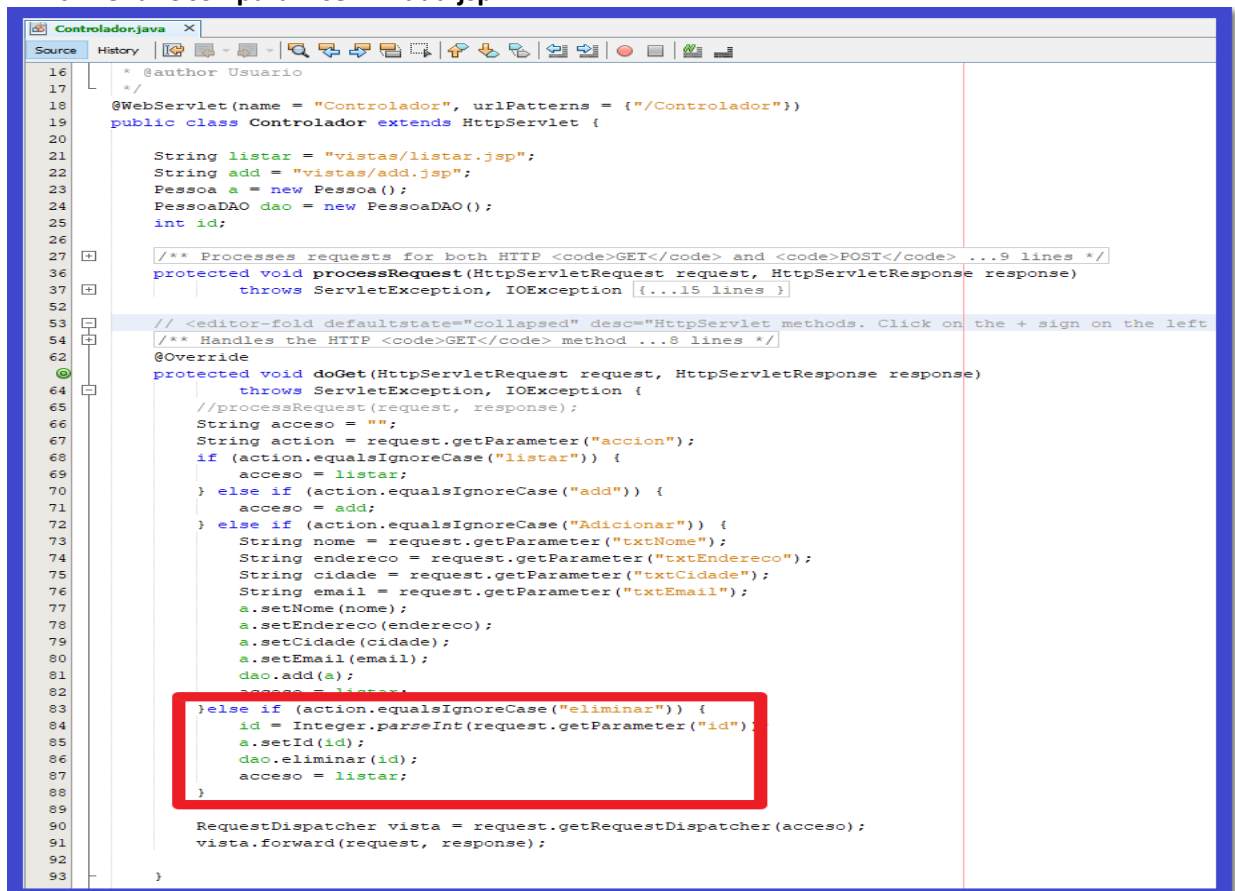
```

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

```

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

- Criar o JSP para inserir – add.jsp



```

16  * @author Usuario
17  */
18  @WebServlet(name = "Controlador", urlPatterns = {"/Controlador"})
19  public class Controlador extends HttpServlet {
20
21      String listar = "vistas/listar.jsp";
22      String add = "vistas/add.jsp";
23      Pessoa a = new Pessoa();
24      PessoaDAO dao = new PessoaDAO();
25      int id;
26
27      /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 lines */
28      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
29          throws ServletException, IOException {
30
31
32
33
34
35      /** Handles the HTTP <code>GET</code> method ...8 lines */
36      @Override
37      protected void doGet(HttpServletRequest request, HttpServletResponse response)
38          throws ServletException, IOException {
39
40          //processRequest(request, response);
41          String acesso = "";
42          String action = request.getParameter("accion");
43          if (action.equalsIgnoreCase("listar")) {
44              acesso = listar;
45          } else if (action.equalsIgnoreCase("add")) {
46              acesso = add;
47          } else if (action.equalsIgnoreCase("Adicionar")) {
48              String nome = request.getParameter("txtNome");
49              String endereco = request.getParameter("txtEndereco");
50              String cidade = request.getParameter("txtCidade");
51              String email = request.getParameter("txtEmail");
52              a.setName(nome);
53              a.setEndereco(endereco);
54              a.setCidade(cidade);
55              a.setEmail(email);
56              dao.add(a);
57              acesso = listar;
58          } else if (action.equalsIgnoreCase("eliminar")) {
59              id = Integer.parseInt(request.getParameter("id"));
60              a.setId(id);
61              dao.eliminar(id);
62              acesso = listar;
63          }
64
65          RequestDispatcher vista = request.getRequestDispatcher(acesso);
66          vista.forward(request, response);
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93

```

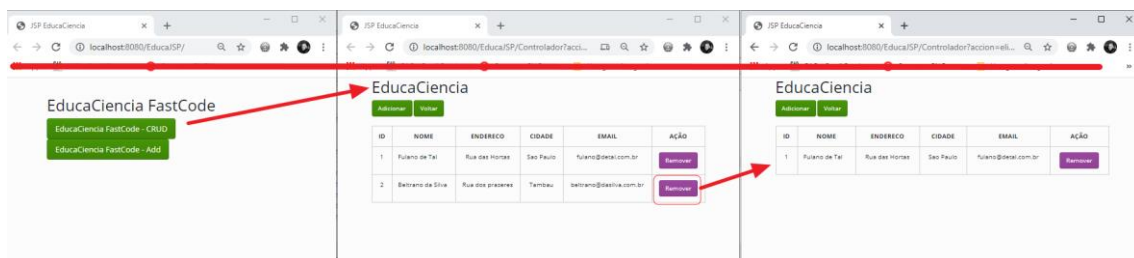
○ Adicionar opção deletar em listar.jsp

```

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      <tbody>
34      <!--
35      PessoaDAO dao = new PessoaDAO();
36      List<Pessoa> list = dao.listar();
37      Iterator<Pessoa> iter = list.iterator();
38      Pessoa a = null;
39      while (iter.hasNext()) {
40          a = iter.next();
41      }
42      -->
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?accion=eliminar&id=<%= a.getId() %>">Remover</a>
52      </td>
53      </tr>
54      </tbody>
55      </table>
56      </div>
57      </body>
58      </html>

```

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





Two screenshots of a web application named 'EducaCiencia' running on a browser at localhost:8080.

The first screenshot shows the 'listar' action, displaying a table with two records:

ID	NOME	ENDERECO	CIDADE	EMAIL	AÇÃO
1	Fulano de Tal	Rua das Hortas	Sao Paulo	fulano@desai.com.br	Remover
2	Beltrano da Silva	Rua dos prazeres	Tambau	beltrano@dasilva.com.br	Remover

The second screenshot shows the 'eliminar' action for ID=2, displaying a table with one record:

ID	NOME	ENDERECO	CIDADE	EMAIL	AÇÃO
1	Fulano de Tal	Rua das Hortas	Sao Paulo	fulano@desai.com.br	Remover

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 !

