

EducaCiência FastCode

Fala Galera,

- Artigo: 28/2020 Data: Outubro/2020
- Público Alvo: Desenvolvedores – Iniciantes
- Tecnologia: Java
- Tema: JSP – criando Método Insert
- 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 28/2020 traremos uma maneira simples de criar o Método Insert em um projeto JSP, e daremos continuidade no CRUD nos demais artigos (29,30).

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 | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: I A

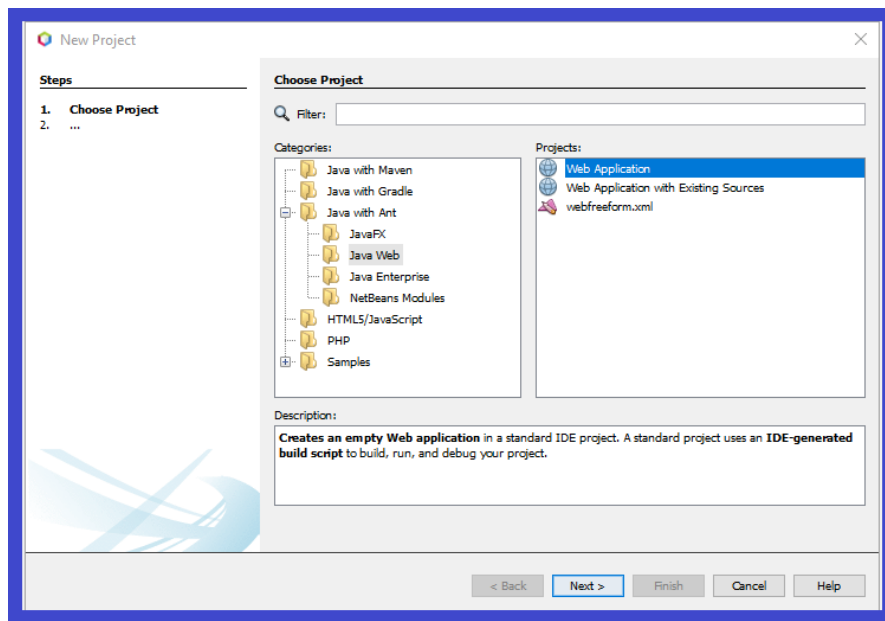
	id	nome	endereco	cidade	estado	email
▶	1	Fulano de Tal	Rua das Hortas	Sao Paulo	SP	fulano@detal.com.br
*	NULL	NULL	NULL	NULL	NULL	NULL

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

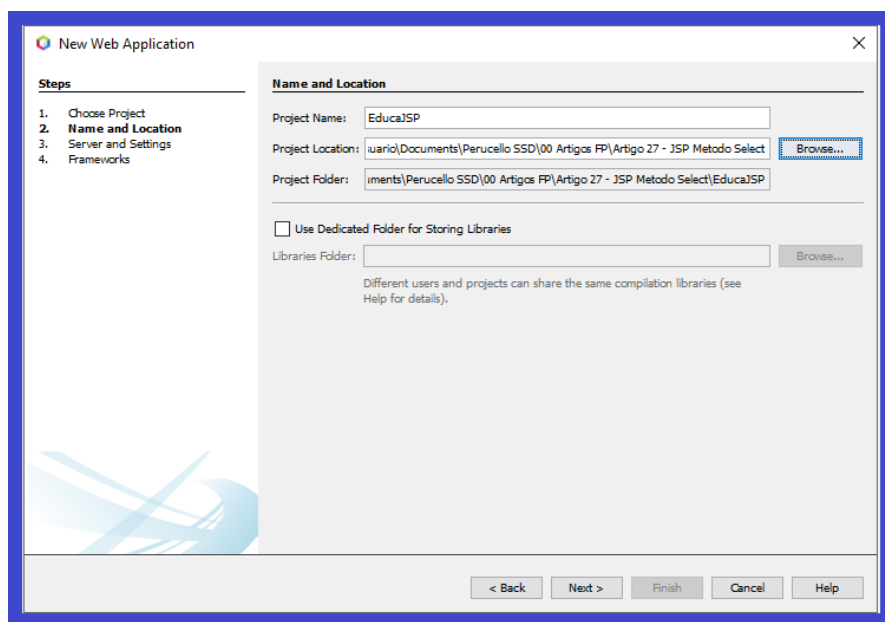
⇒ 27/2020 – abordamos o método Select

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.

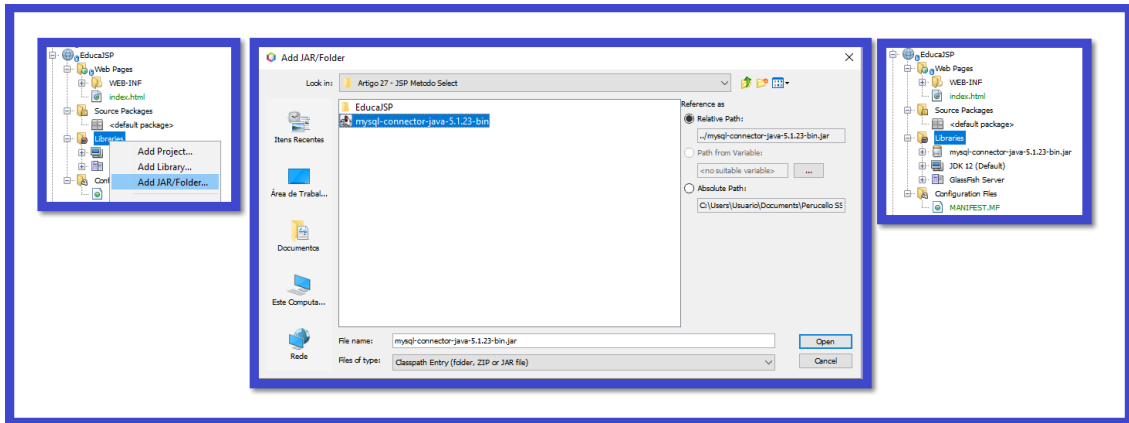
☐ Spring Web MVC

☐ JavaServer Faces

☐ Struts 1.3.10

< Back Next > **Finish** Cancel Help

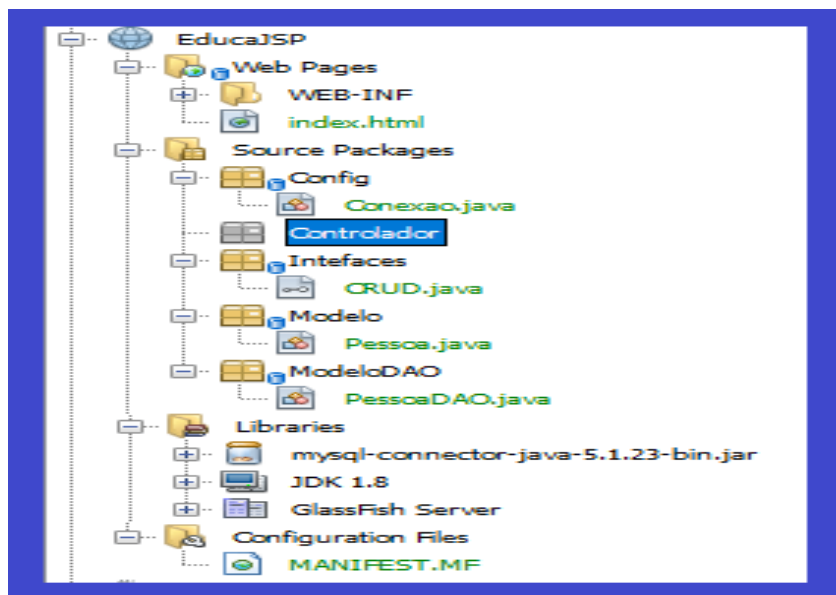
⇒ Para conexão com Banco de Dados , utilizamos do driver jdbc :



⇒ **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 = "EducaJP";
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 USER = "root";
16    private static final String PWD = "";
17
18    public void Conexao() {
19        try {
20            Class.forName(DRIVER);
21            con = DriverManager.getConnection(URL, USER, PWD);
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 => " + USER);
29            System.out.println("Senha => " + PWD);
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 ...9 lines
4
12
13 public class PessoaDAO implements CRUD{
14     Conexao cn=new Conexao();
15     Connection con;
16     PreparedStatement ps;
17     ResultSet rs;
18     Pessoa a = new Pessoa();
19
20 public PessoaDAO(){
21     super();
22     con = this.cn.getConnection();
23 }
24
25 @Override
26 public List listar() {
27     cn.Conexao();
28     ArrayList<Pessoa>list = new ArrayList<>();
29     String sql="select * from pessoa";
30     try {
31         con=cn.getConnection();
32         ps=con.prepareStatement(sql);
33         rs=ps.executeQuery();
34         while(rs.next()){
35             Pessoa a = new Pessoa();
36             a.setId(rs.getInt("Id"));
37             a.setNome(rs.getString("Nome"));
38             a.setEndereco(rs.getString("Endereco"));
39             a.setCidade(rs.getString("Cidade"));
40             a.setEmail(rs.getString("Email"));
41             list.add(a);
42         }
43     } catch (SQLException e) {
44     } finally{
45         cn.Desconectar(con);
46     }
47
48     return list;
49 }
50
51 @Override
52 public Pessoa list(int id) {
53     cn.Conexao();
54     String sql="select * from pessoa where Id=" + id;
55     try {
56         con=cn.getConnection();
57         ps=con.prepareStatement(sql);
58         rs=ps.executeQuery();
59         while(rs.next()){
60             a.setId(rs.getInt("Id"));
61             a.setNome(rs.getString("Nome"));
62             a.setEndereco(rs.getString("Endereco"));
63             a.setCidade(rs.getString("Cidade"));
64             a.setEmail(rs.getString("Email"));
65
66         }
67     } catch (SQLException e) {
68     } finally {
69         cn.Desconectar(con);
70     }
71
72     return a;
73 }
74
75
76

```

CRUD.java

⇒ Interface

```

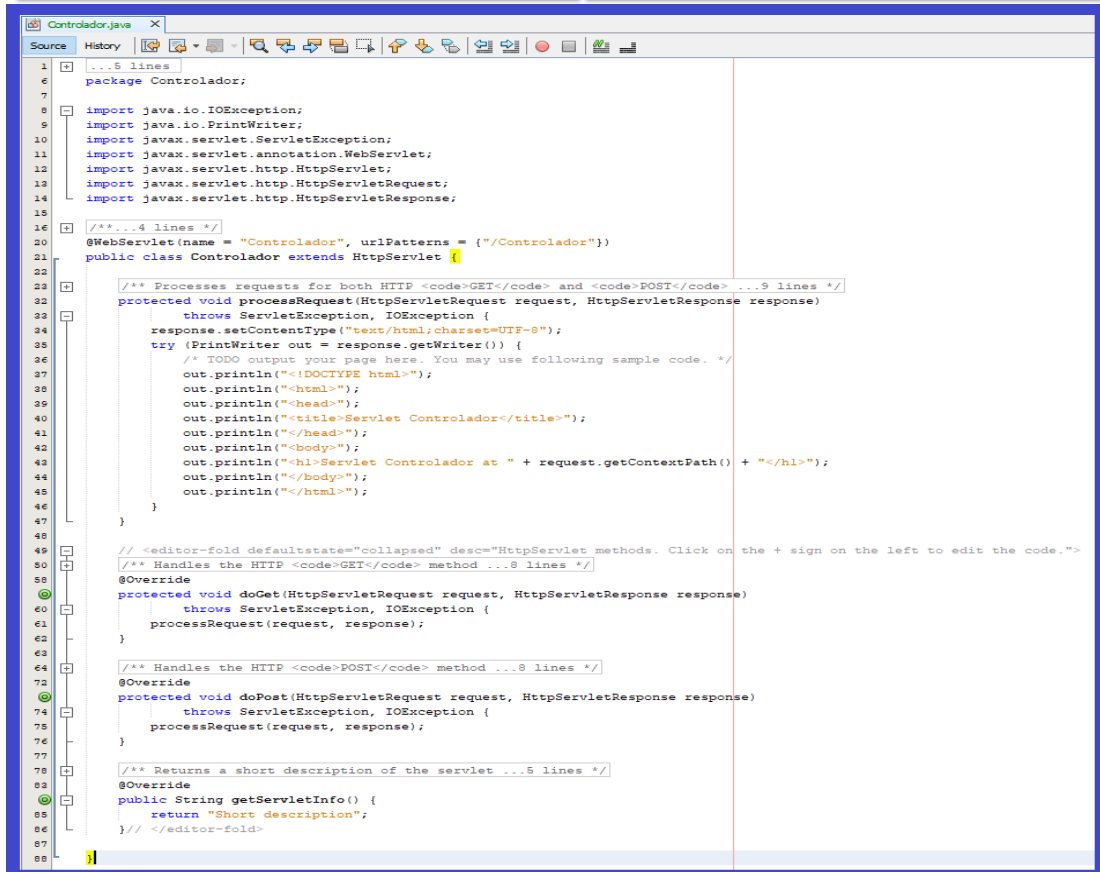
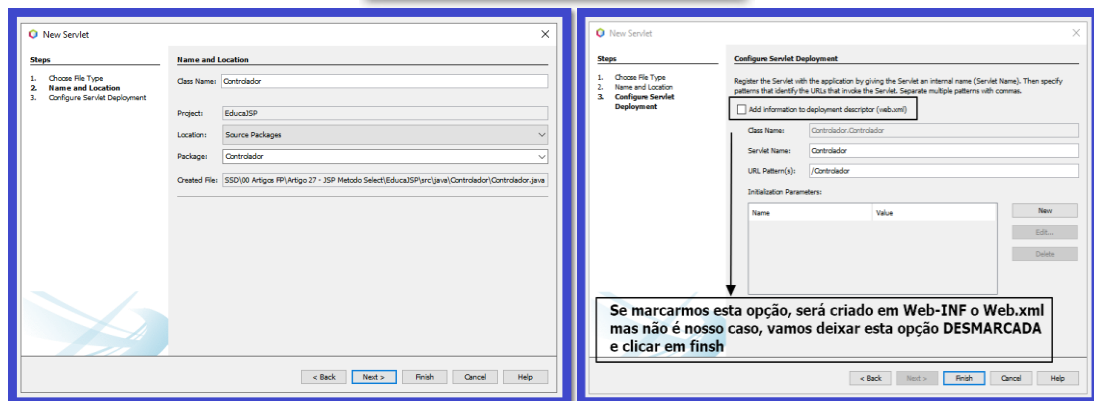
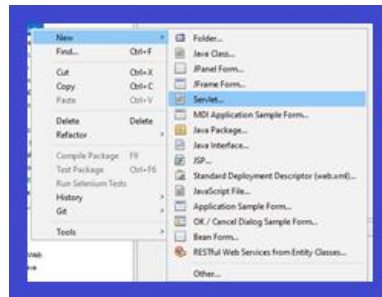
1 package Interfaces;
2
3
4 import Modelo.Pessoa;
5 import java.util.List;
6
7
8 public interface CRUD {
9     public List listar();
10    public Pessoa list(int id);
11 }
12

```

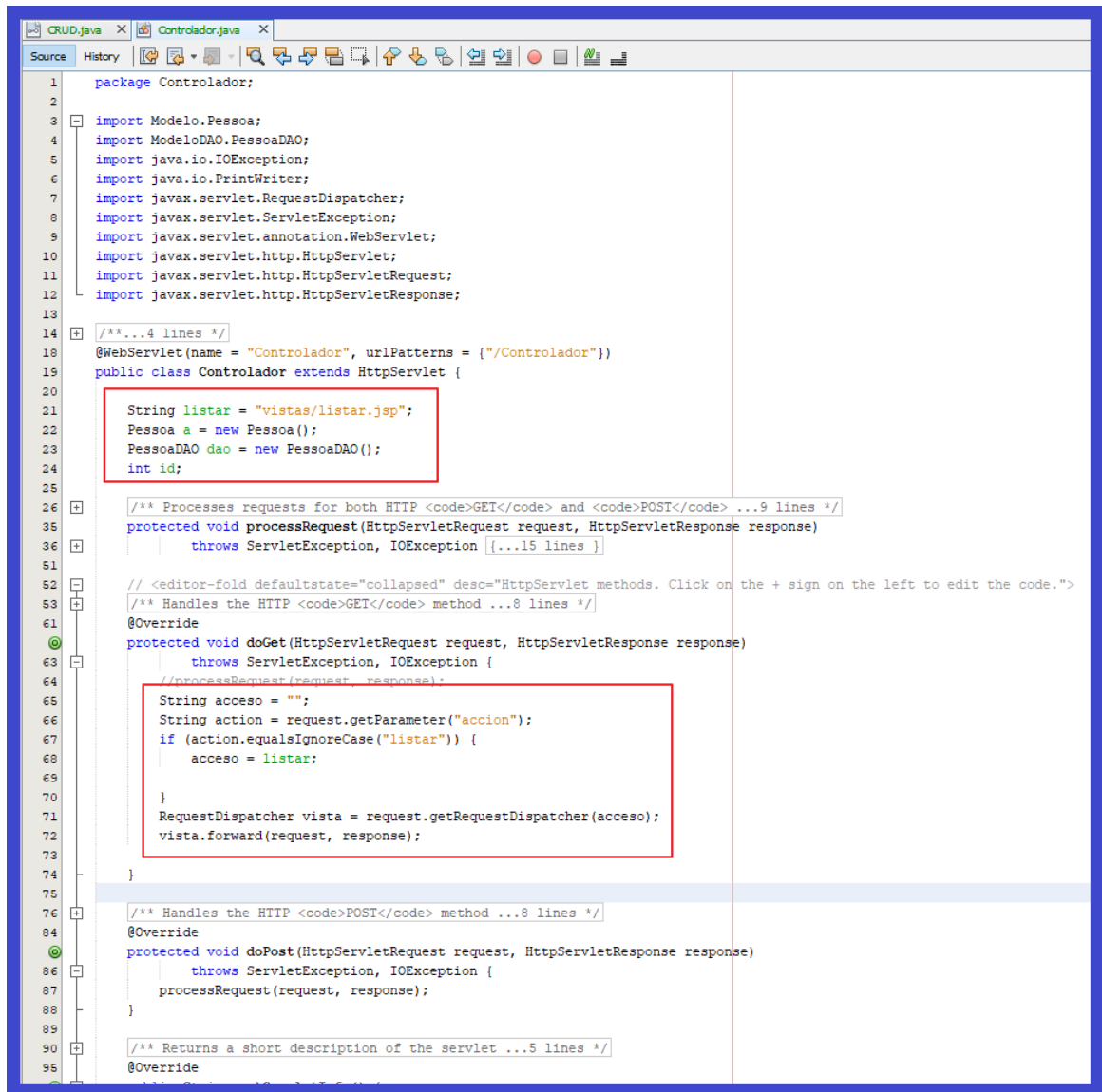
Neste cenário, ao criarmos nosso Servlet, optamos em duas maneiras sendo:

- 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



```
1 package Controlador;
2
3 import Modelo.Pessoa;
4 import ModeloDAO.PessoaDAO;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import javax.servlet.RequestDispatcher;
8 import javax.servlet.ServletException;
9 import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13
14 /** ...4 lines */
15 @WebServlet(name = "Controlador", urlPatterns = {"/Controlador"})
16 public class Controlador extends HttpServlet {
17
18     String listar = "vistas/listar.jsp";
19     Pessoa a = new Pessoa();
20     PessoaDAO dao = new PessoaDAO();
21     int id;
22
23     /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 lines */
24     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
25         throws ServletException, IOException {
26         // ...15 lines
27     }
28
29     /** Handles the HTTP <code>GET</code> method ...8 lines */
30     @Override
31     protected void doGet(HttpServletRequest request, HttpServletResponse response)
32         throws ServletException, IOException {
33         //processRequest(request, response);
34
35         String acesso = "";
36         String action = request.getParameter("action");
37         if (action.equalsIgnoreCase("listar")) {
38             acesso = listar;
39         }
40
41         RequestDispatcher vista = request.getRequestDispatcher(acesso);
42         vista.forward(request, response);
43     }
44
45     /** Handles the HTTP <code>POST</code> method ...8 lines */
46     @Override
47     protected void doPost(HttpServletRequest request, HttpServletResponse response)
48         throws ServletException, IOException {
49         processRequest(request, response);
50     }
51
52     /** Returns a short description of the servlet ...5 lines */
53     @Override
54     // ...
55 }
```

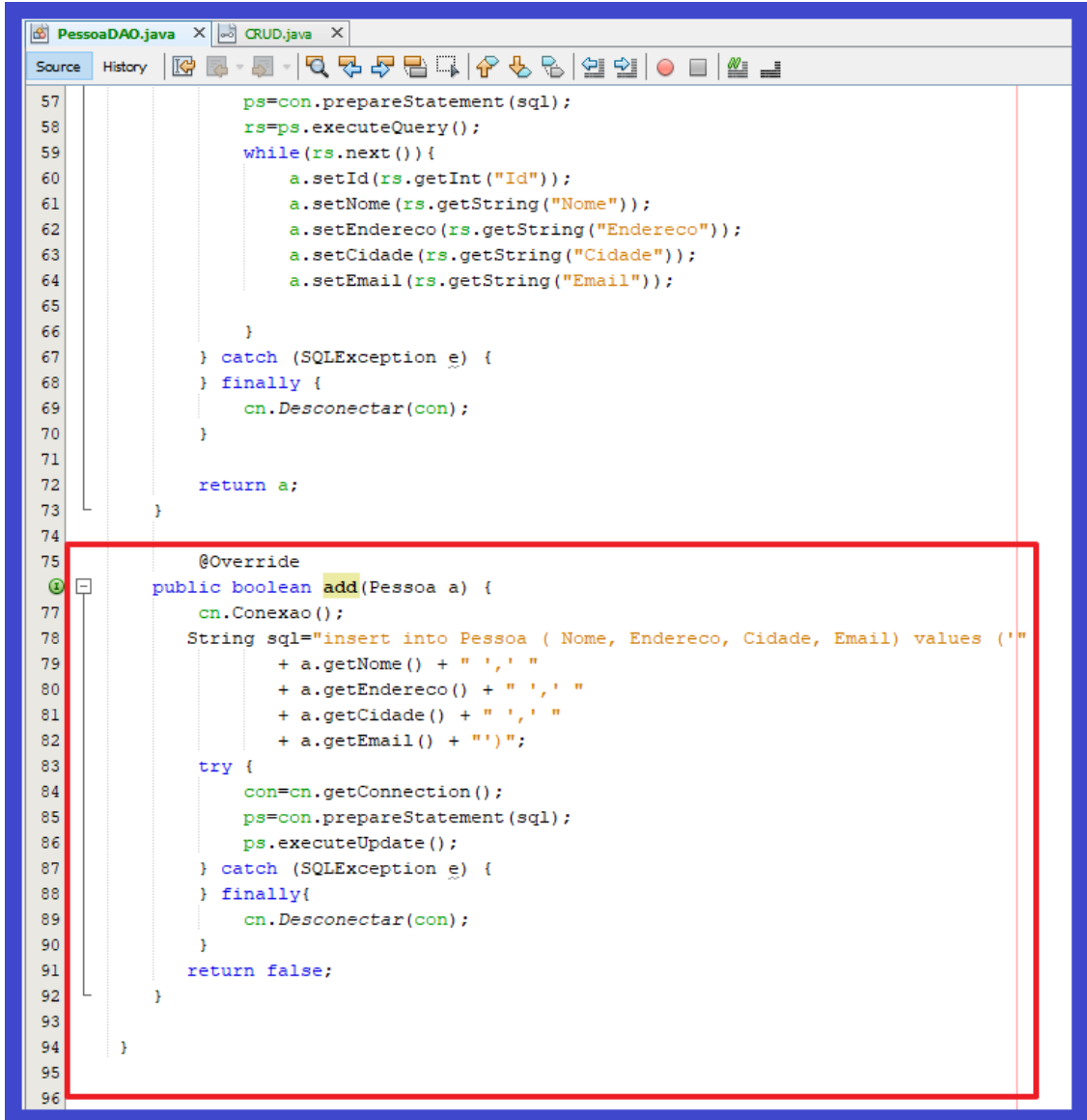
Pronto, agora que revisamos o artigo 27/2020 vamos pôr a mão na massa !

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

- Incluir método inserir no PessoaDAO
- Incluir método na nossa Interface CRUD
- Incluir a rotina do método Inserir no nosso Servlet, ou seja, no nosso doGet do Controlador
- Criar o jsp para inserir
- Adicionar botões inserir no Index.jsp e listar.jsp

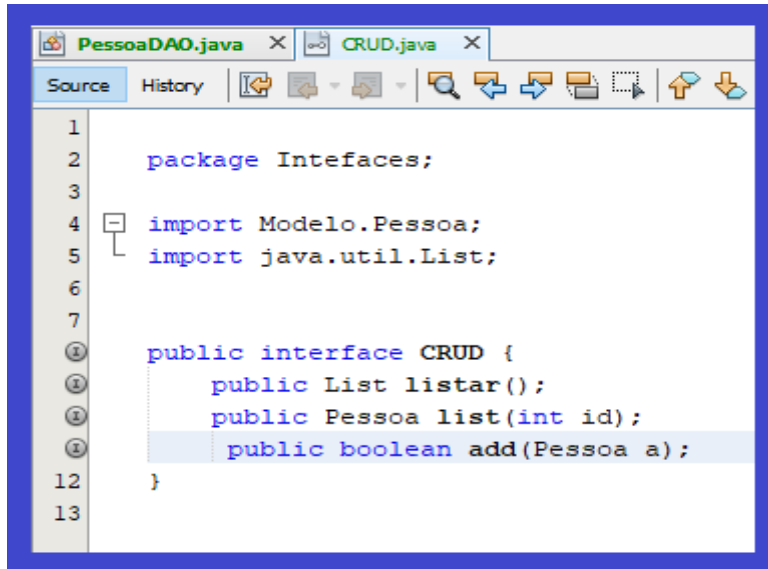
VAMOS NESSA !

- Incluir método inserir no PessoaDAO



```
57         ps=con.prepareStatement(sql);
58         rs=ps.executeQuery();
59         while(rs.next()){
60             a.setId(rs.getInt("Id"));
61             a.setNome(rs.getString("Nome"));
62             a.setEndereco(rs.getString("Endereco"));
63             a.setCidade(rs.getString("Cidade"));
64             a.setEmail(rs.getString("Email"));
65
66         }
67     } catch (SQLException e) {
68     } finally {
69         cn.Desconectar(con);
70     }
71
72     return a;
73 }
74
75 @Override
76 public boolean add(Pessoa a) {
77     cn.Conexao();
78     String sql="insert into Pessoa ( Nome, Endereco, Cidade, Email) values ('"
79         + a.getNome() + " ',' "
80         + a.getEndereco() + " ',' "
81         + a.getCidade() + " ',' "
82         + a.getEmail() + "')";
83
84     try {
85         con=cn.getConnection();
86         ps=con.prepareStatement(sql);
87         ps.executeUpdate();
88     } catch (SQLException e) {
89     } finally{
90         cn.Desconectar(con);
91     }
92     return false;
93 }
94
95 }
96 }
```

- Incluir método na nossa Interface CRUD

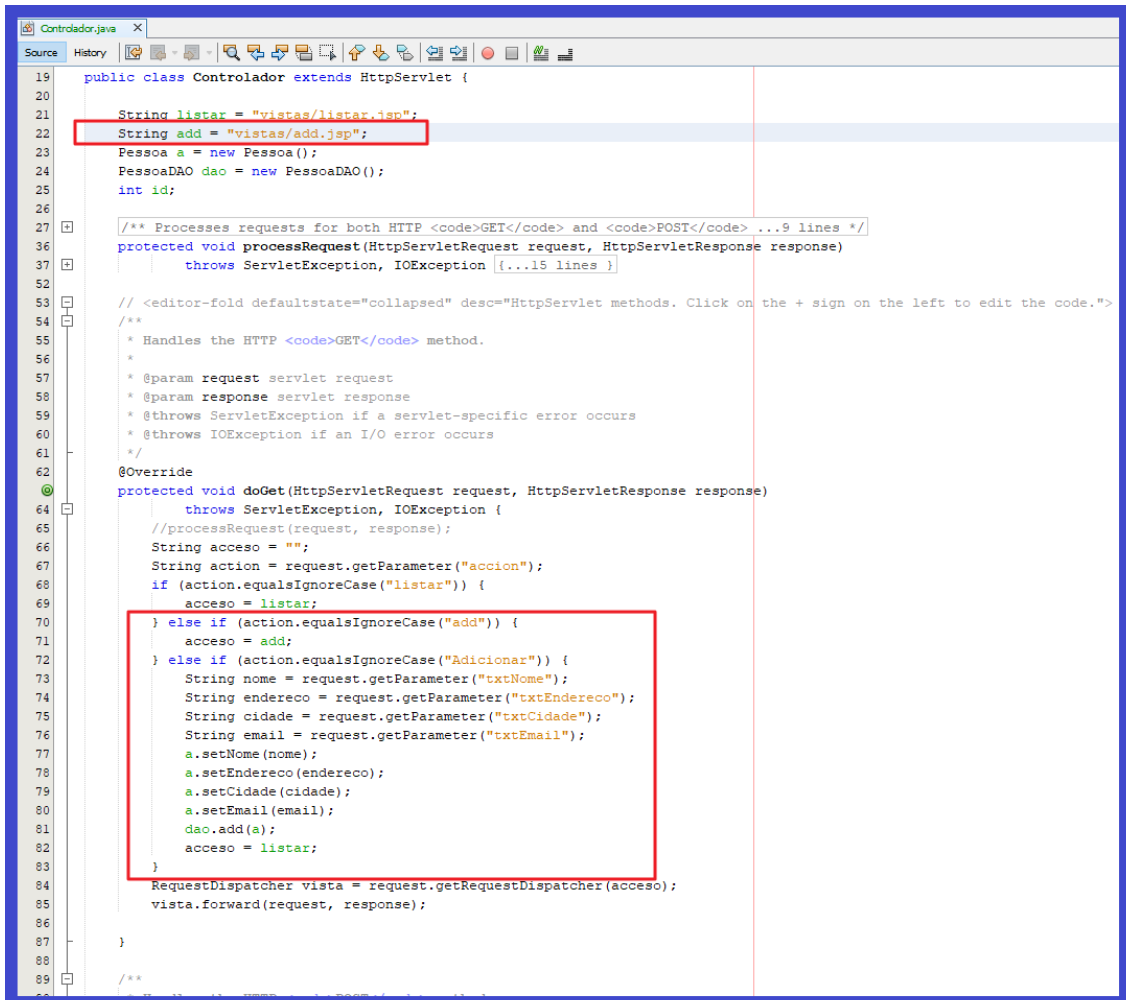


```

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 }
13

```

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

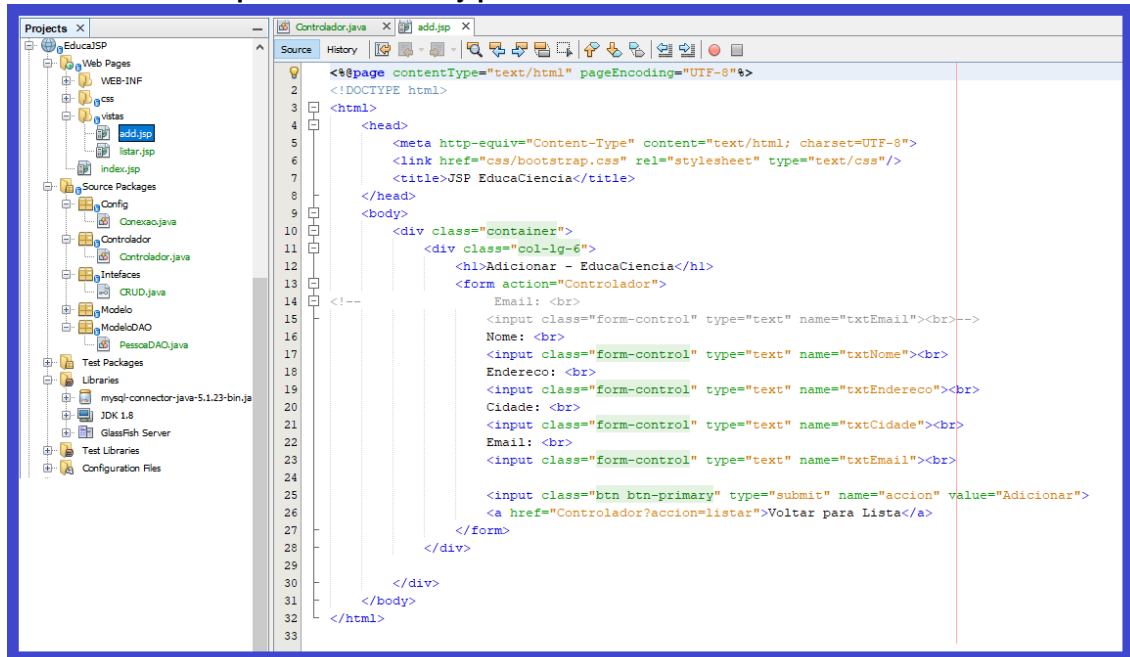


```

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      * Handles the HTTP <code>GET</code> method.
35      *
36      * @param request servlet request
37      * @param response servlet response
38      * @throws ServletException if a servlet-specific error occurs
39      * @throws IOException if an I/O error occurs
40      */
41     @Override
42     protected void doGet(HttpServletRequest request, HttpServletResponse response)
43         throws ServletException, IOException {
44         //processRequest(request, response);
45         String acesso = "";
46         String action = request.getParameter("accion");
47         if (action.equalsIgnoreCase("listar")) {
48             acesso = listar;
49         } else if (action.equalsIgnoreCase("add")) {
50             acesso = add;
51         } else if (action.equalsIgnoreCase("Adicionar")) {
52             String nome = request.getParameter("txtNome");
53             String endereco = request.getParameter("txtEndereco");
54             String cidade = request.getParameter("txtCidade");
55             String email = request.getParameter("txtEmail");
56             a.setNome(nome);
57             a.setEndereco(endereco);
58             a.setCidade(cidade);
59             a.setEmail(email);
60             dao.add(a);
61             acesso = listar;
62         }
63         RequestDispatcher vista = request.getRequestDispatcher(acesso);
64         vista.forward(request, response);
65     }
66
67     /**
68      *
69      */
70
71 }
72

```

○ Criar o JSP para inserir – add.jsp

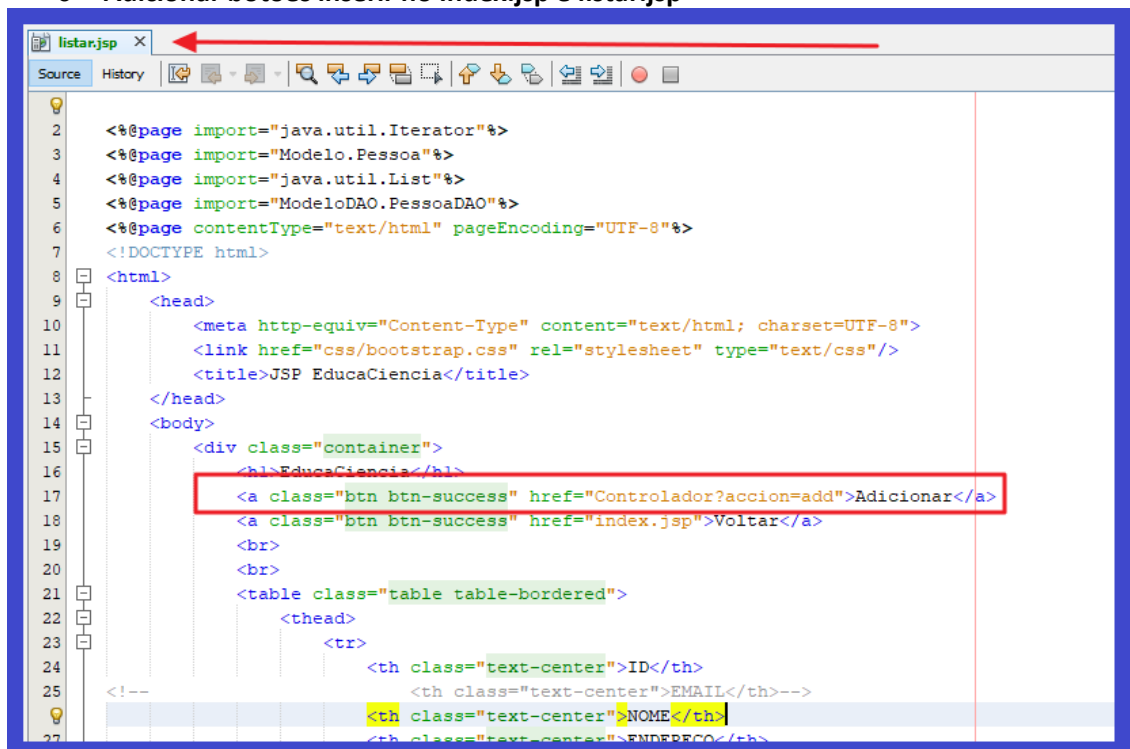


```

1 <%%page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
7 <title>JSP EducaCiencia</title>
8 </head>
9 <body>
10 <div class="container">
11 <div class="col-lg-6">
12 <h1>Adicionar - EducaCiencia</h1>
13 <form action="Controlador">
14 <!--
15 Email: <br>
16 <input class="form-control" type="text" name="txtEmail"><br>-->
17 Nome: <br>
18 <input class="form-control" type="text" name="txtNome"><br>
19 Endereco: <br>
20 <input class="form-control" type="text" name="txtEndereco"><br>
21 Cidade: <br>
22 <input class="form-control" type="text" name="txtCidade"><br>
23 Email: <br>
24 <input class="form-control" type="text" name="txtEmail"><br>
25 <input class="btn btn-primary" type="submit" name="acao" value="Adicionar">
26 <a href="Controlador?acao=listar">Voltar para Lista</a>
27 </form>
28 </div>
29 </div>
30 </body>
31 </html>

```

○ Adicionar botões inserir no Index.jsp e listar.jsp



```

1 <%%page import="java.util.Iterator"%>
2 <%%page import="Modelo.Pessoa"%>
3 <%%page import="java.util.List"%>
4 <%%page import="ModeloDAO.PessoaDAO"%>
5 <%%page contentType="text/html" pageEncoding="UTF-8"%>
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10 <link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
11 <title>JSP EducaCiencia</title>
12 </head>
13 <body>
14 <div class="container">
15 <h1>EducaCiencia</h1>
16 <a class="btn btn-success" href="Controlador?acao=add">Adicionar</a>
17 <a class="btn btn-success" href="index.jsp">Voltar</a>
18 <br>
19 <br>
20 <table class="table table-bordered">
21 <thead>
22 <tr>
23 <th class="text-center">ID</th>
24 <th class="text-center">EMAIL</th>
25 <th class="text-center">NOME</th>
26 <th class="text-center">ENDERECO</th>
27 </tr>

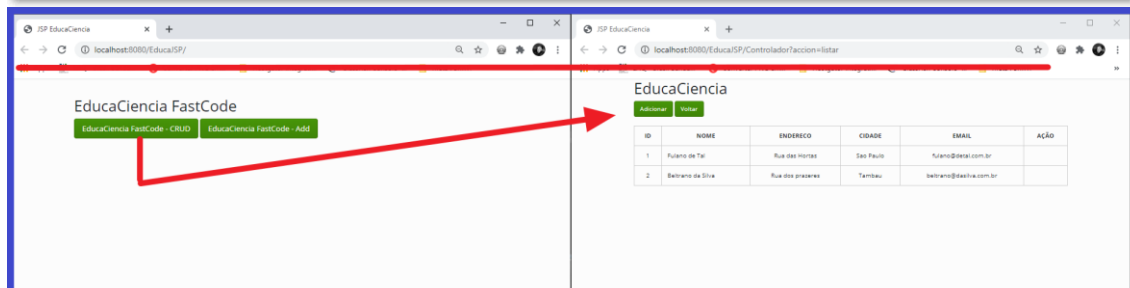
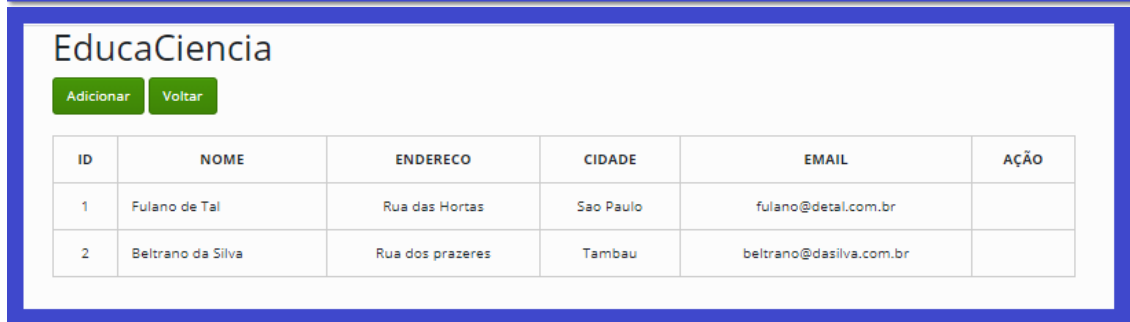
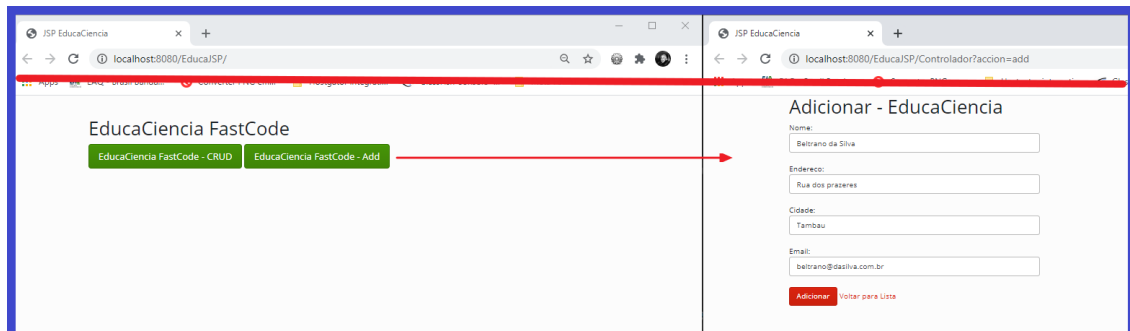
```

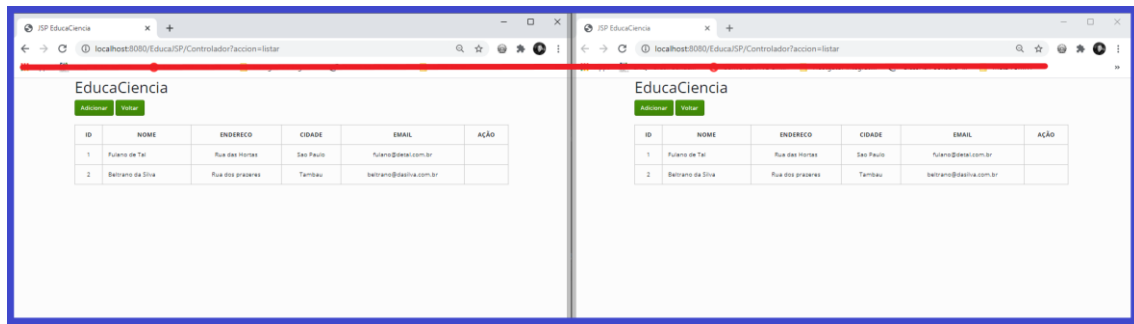
```

1  index.jsp
2  <%@page contentType="text/html" pageEncoding="UTF-8"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7  <link href="./css/bootstrap.css" rel="stylesheet" type="text/css"/>
8  <title>JSP EducaCiencia</title>
9  </head>
10 <body>
11 <br>
12 <br>
13 <div class="container">
14 <h1>EducaCiencia FastCode </h1>
15 <h1> </h1>
16 <a class="btn btn-success btn-lg" href="Controlador?acao=listar">EducaCiencia FastCode - CRUD</a>
17 <a class="btn btn-success btn-lg" href="Controlador?acao=add">EducaCiencia FastCode - Add</a>
18 </div>
19 </body>
20 </html>
21

```

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





Nota-se que os códigos funcionaram corretamente, no entanto, saliento que os artigos 27/2020 e 28/2020 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 !

