



{ **COTI INFORMÁTICA** }
ESCOLA DE NERDS }



Java WebDeveloper - Aula 05b - Noite Seg, Qua, Sex - Noite



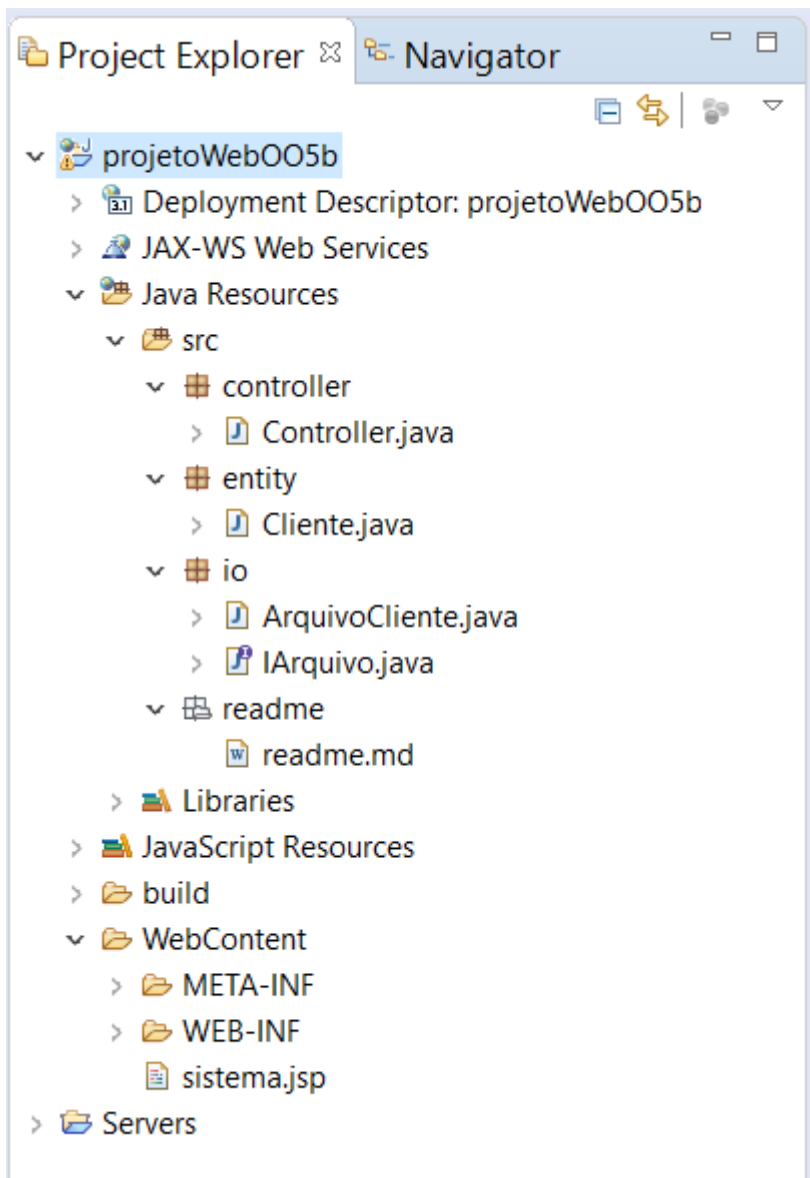
Tema da aula:

Projeto Servlet, Javascript, Validação, Interface, Gravação de arquivo.

—  Prof Edson Belém - profedsonbelem@gmail.com (<mailto:profedsonbelem@gmail.com>) 🕒 Segunda, Abr 06, 2020



Estrutura do projeto depois de finalizado:



Bibliotecas utilizadas:

- lib
 - gson-2.2.4.jar
 - jstl-1.2.jar
 - mysql-connector-java-5.1.23-bin.jar
 - servlet3-api.jar



Readme.md (<http://Readme.md>)

```
# estou na web o que preciso (agora faço site)
```

```
## Servlet3.java / mysql.connector.jar /  
   jstl (jsp em tag) / gson.json
```

```
###
```

 sistema.jsp

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <!DOCTYPE html>
4  <html>
5
6  <head>
7  </head>
8
9
10 <body>
11   <h2>Projeto  Cadastro Timoteo</h2>
12   <hr/>
13   <form id="form1" method="post" action="Controller">
14       Nome<br/>
15       <input type="text" value="" id="nome" name="nome" />
16       <br/>Email<br/>
17       <input type="text" value="" id="email" name="email" />
18       <br/>Sexo<br/>
19       <input type="text" value="" id="sexo" name="sexo"  size="2"/>
20       <br/>
21       <button type="button" onclick="recebe()">
22           Validar os Dados</button>
23   </form>
24
25
26   <div id="error" style="color:red;font-size:13pt;"></div>
27
28   <div id="respostaok" style="color:blue;font-size:13pt;">
29   ${cliente}
30   </div>
31
32   <div id="respostaNok" style="color:red;font-size:13pt;">
33   ${msgerror}
34   </div>
35
36
37   <script>
38   class Cliente{
39
40       constructor(nome,email,sexo){
41       this.nome = nome;
42       this.email = email;
43       this.sexo = sexo;
44       }
45
46       toString(){
47       return  this.nome + ", " + this.email  + ", " +      this.sexo;
48       }
49   }
50
51       function validaNome(nome){
52       var reg=/^[a-z A-Z]{2,50}$/;
53       return reg.test(nome);
54       }

```

```

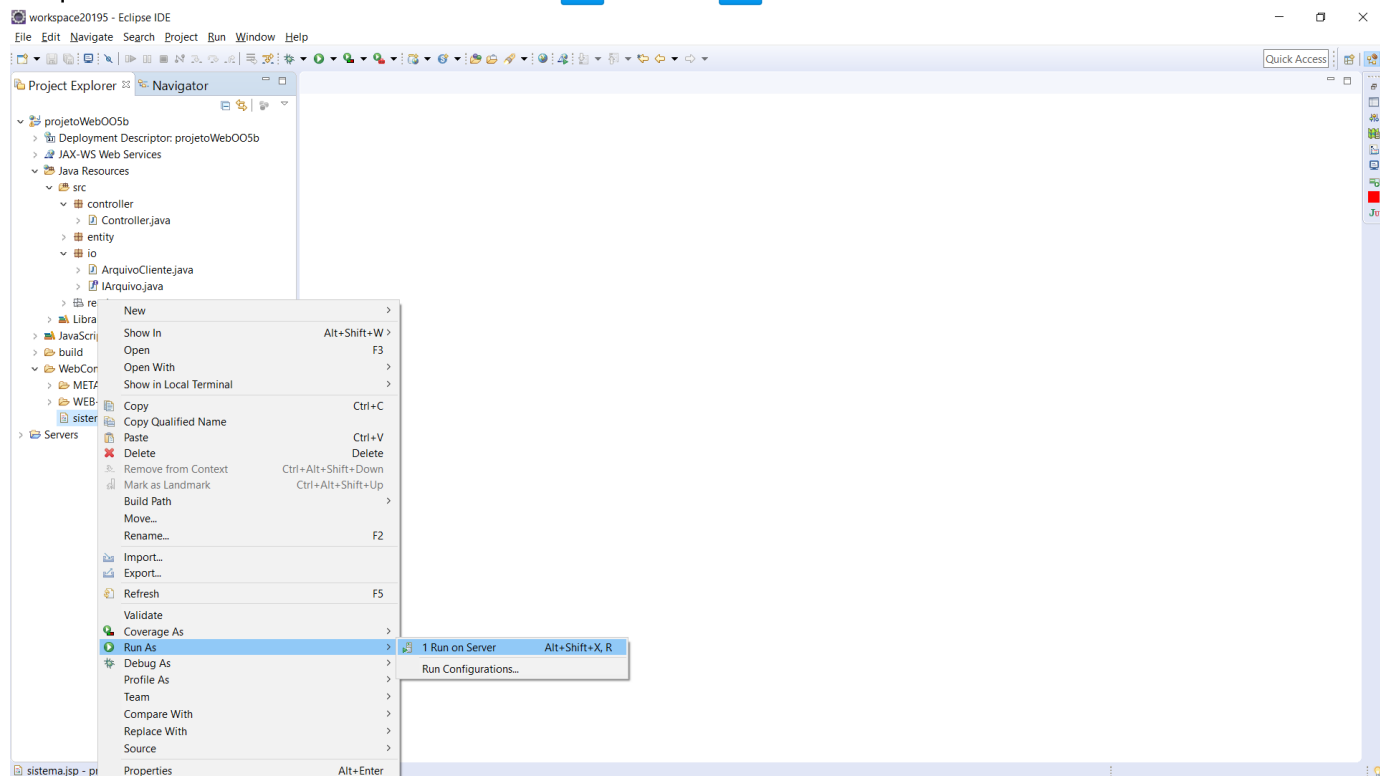
55
56     function validaSexo(sexo){
57         var reg=/^m|M|f|F$/;
58         return reg.test(sexo);
59     }
60
61     function validaEmail(email){
62         var reg=/^([a-zA-Z0-9])+@([a-zA-Z0-9])+\.[a-z]+$/;
63         return reg.test(email);
64     }
65
66
67     var obj = null;
68
69
70     function recebe(){
71         obj= new Cliente(
72             document.getElementById("nome").value,
73             document.getElementById("email").value,
74             document.getElementById("sexo").value
75             );
76
77         console.log('Dados :', JSON.stringify(obj));
78         console.log('Dados :', obj.toString());
79         console.log('teste1', validaNome(obj.nome));
80         console.log('teste2', validaSexo(obj.sexo));
81         console.log('teste3', validaEmail(obj.email));
82
83         if (validaNome(obj.nome) && validaSexo(obj.sexo)
84             && validaEmail(obj.email)){
85             document.getElementById("form1").submit();
86         }else{
87             document.getElementById("error").innerHTML=
88                 "<li>Dados Invalidos</li>";
89         }
90     }
91 </script>
92
93 </body>
94 </html>

```



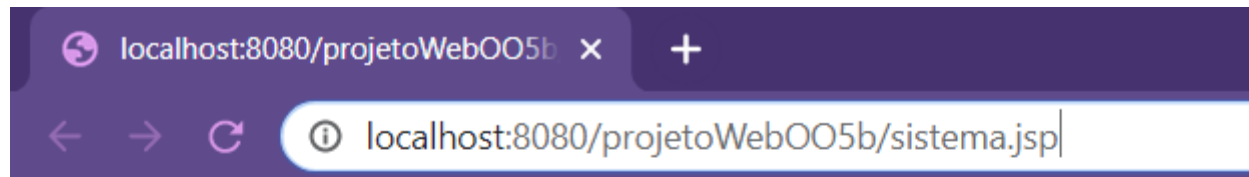
Para rodar a página:

Clique na classe com o botão direito ➡ run as ➡ run on server



👁 Resultado no console.

🔗 <http://localhost:8080/projetoWeb005b/sistema.jsp>



Projeto Cadastro Timoteo

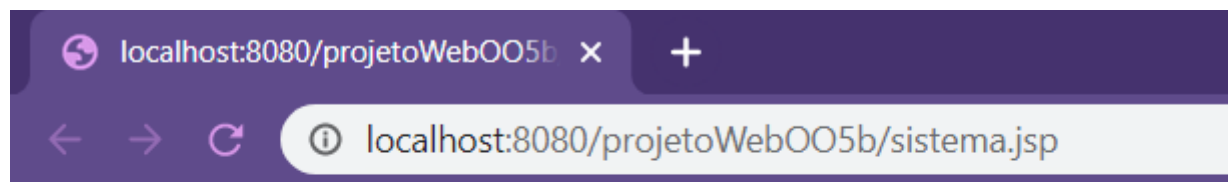
Nome

Email

Sexo

Validar os Dados

✎ Digitar para cadastrar um cliente.




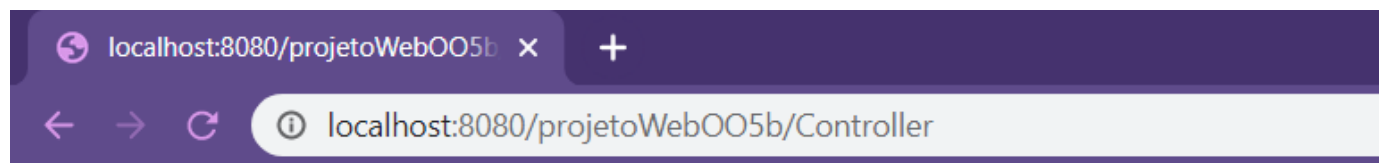
Projeto Cadastro Timoteo

Nome

Email

Sexo

 Clicando em validar




Projeto Cadastro Timoteo

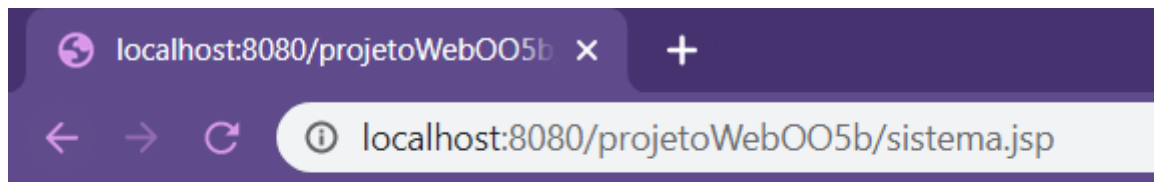
Nome

Email

Sexo

Cliente [id=null, nome=belem, email=profedsonbelem@gmail.com, sexo=m]

 Digitando o nome com uma letra apenas



Projeto Cadastro Timoteo

Nome

Email

Sexo

Validar os Dados

- **Dados Invalidos**



Cliente.java


```
1 package entity;
2
3 public class Cliente {
4
5     private Integer id;
6     private String nome;
7     private String email;
8     private String sexo;
9
10    public Cliente() {
11
12    }
13
14    public Cliente(Integer id, String nome, String email, String sexo) {
15        super();
16        this.id = id;
17        this.nome = nome;
18        this.email = email;
19        this.sexo = sexo;
20    }
21
22    @Override
23    public String toString() {
24        return "Cliente [id=" + id + ", nome=" + nome + ", email="
25            + email + ", sexo=" + sexo + "]";
26    }
27
28    public Integer getId() {
29        return id;
30    }
31
32    public void setId(Integer id) {
33        this.id = id;
34    }
35
36    public String getNome() {
37        return nome;
38    }
39
40    public void setNome(String nome) {
41        this.nome = nome;
42    }
43
44    public String getEmail() {
45        return email;
46    }
47
48    public void setEmail(String email) {
49        this.email = email;
50    }
51
52    public String getSexo() {
53        return sexo;
54    }
```

```
55  
56     public void setSexo(String sexo) {  
57         this.sexo = sexo;  
58     }  
59  
60 }
```



Controller.java

```

1  package controller;
2
3  import java.io.IOException;
4
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 import entity.Cliente;
12
13 @WebServlet("/Controller")
14 public class Controller extends HttpServlet {
15
16     private static final long serialVersionUID = 1L;
17
18     public Controller() {
19         super();
20     }
21
22     protected void doGet(HttpServletRequest request,
23         HttpServletResponse response) throws ServletException, IOException {
24     }
25
26     protected void doPost(HttpServletRequest request,
27         HttpServletResponse response) throws ServletException, IOException {
28         // envio na web (string)
29
30         Cliente c = new Cliente(null,
31             request.getParameter("nome"),
32             request.getParameter("email"),
33             request.getParameter("sexo"));
34         try {
35             if (c.getNome().equalsIgnoreCase("rafael")) {
36                 throw new Exception("Nome Invalido ...."
37                     + c.getNome());
38             }
39             request.setAttribute("cliente", c);
40         } catch (Exception ex) {
41             request.setAttribute("msgerror", ex.getMessage());
42         } request.getRequestDispatcher("sistema.jsp")
43             .forward(request, response);
44     }
45 }

```



lArquivo.java

```

1  package io;
2
3  import entity.Cliente;
4
5  public interface IArquivo {
6
7      public void abrir() throws Exception;
8
9      public void gravar(Cliente c) throws Exception;
10
11     public void fechar() throws Exception;
12
13 }

```



Arquivo.java

```

1  package io;
2
3  import java.io.FileWriter;
4  import com.google.gson.Gson;
5  import entity.Cliente;
6
7  public class ArquivoCliente implements IArquivo {
8
9      FileWriter fw;
10
11      @Override
12      public void abrir() throws Exception {
13          fw = new FileWriter("/tmp/cliente.json", false);
14      }
15
16      @Override
17      public void gravar(Cliente c) throws Exception {
18          fw.write(new Gson().toJson(c));
19          fw.flush();
20      }
21
22      @Override
23      public void fechar() throws Exception {
24          fw.close();
25      }
26 }

```