



# Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

## PROJETO WEB:

Este projeto nos mostra uma **página JSP** que contém um formulário para digitarmos os valores dos 3 lados de um triângulo. Na mesma página, o sistema retorna o valor da área do triângulo, e uma mensagem se ele é escaleno, equilátero ou isósceles.

Estrutura do projeto depois de finalizado:





# Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

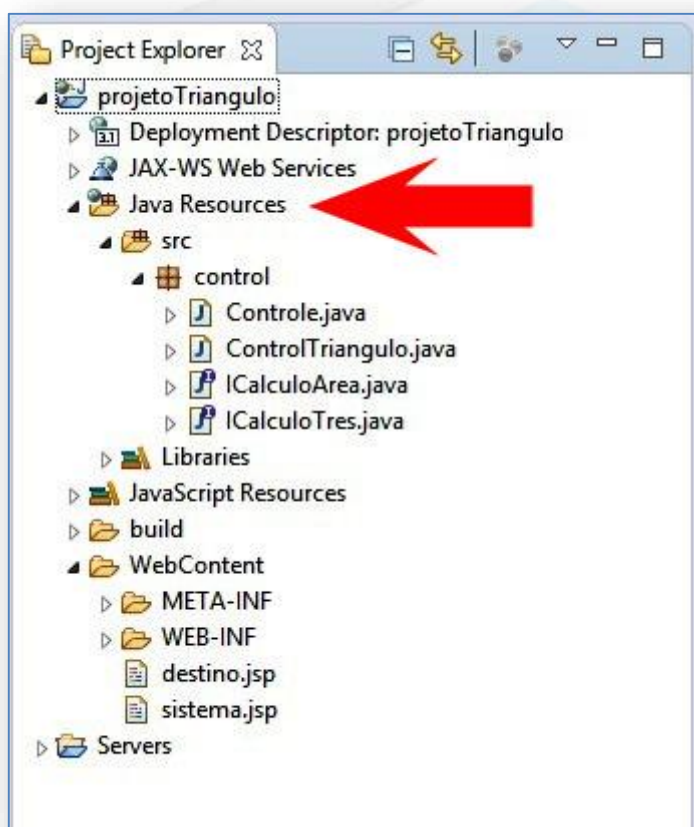
Para criar esse projeto:

Clicar no menu **File -> New -> Dynamic Web Project**

Vamos criar um pacote chamado **control**, onde vamos ter as classes do nosso sistema.

Para criar um novo pacote clique em **Java Resources** e depois, clique com o botão direito em **src** [abaixo de Java Resources].

Depois escolha **new -> package** e dê o nome **control**.





# Java

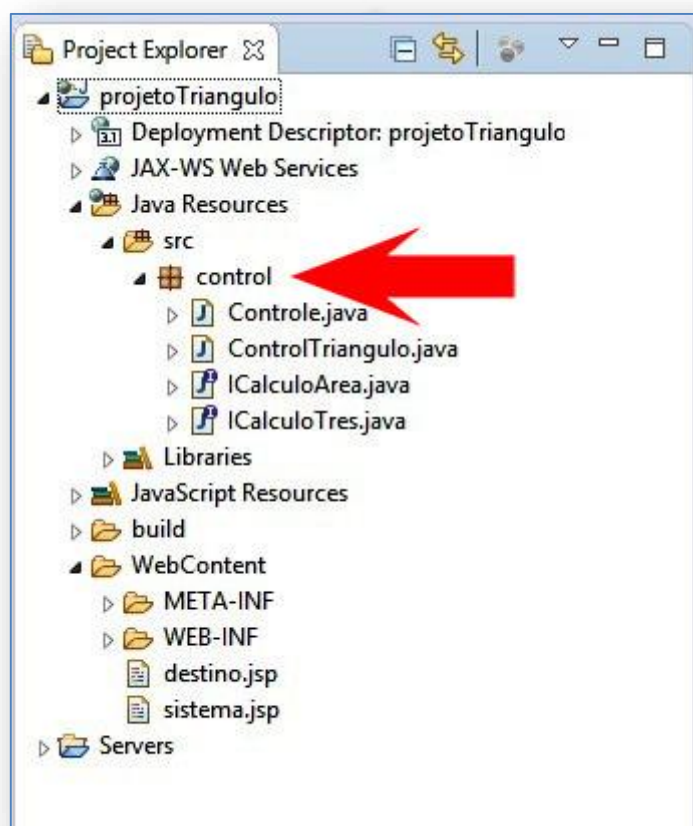
Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

Precisamos criar as **interfaces funcionais** de nosso sistema.

Como toda Interface, ela vai conter apenas a **assinatura dos métodos** - sem a sua implementação.

Para criar uma nova **interface funcional**, clique com o botão direito em control [veja imagem abaixo] e escolha **New -> Interface**. Depois, basta dar o nome para a **interface funcional**.





## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

O nome da nossa primeira Interface Funcional é **ICalculoTres**.

Essa Interface terá um método apenas. **Toda interface funcional deve ter apenas um método.**

Essa Interface irá receber os 3 lados de um triângulo e retornar se o triângulo é isósceles, escaleno ou equilátero. Recebe 3 valores double e retorna uma String.

---

### ICalculoTres.java

```
package control;  
  
@FunctionalInterface  
public interface ICalculoTres {  
    public String operacao(double a, double b, double c);  
}
```

---

O nome da nossa segunda Interface Funcional é **ICalculoArea**.

Essa Interface irá receber os 3 lados de um triângulo e retornar o valor do cálculo da área do triângulo. Recebe 3 valores double e retorna um valor Double.

### ICalculoArea.java

```
package control;  
  
@FunctionalInterface  
public interface ICalculoArea {  
    public Double operacao(double a, double b, double c);  
}
```



# Java

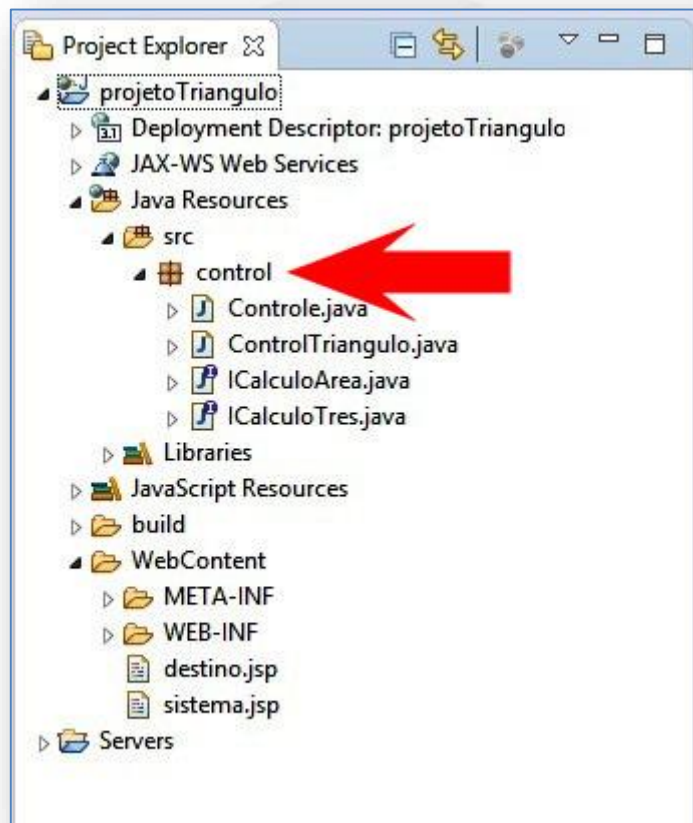
Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

}

Precisamos criar uma Classe chamada **Controle**. Nessa Classe, vamos implementar os métodos que foram declarados nas interfaces funcionais que criamos.

Para criar a Classe **Controle**, Clique com o botão direito no nome do pacote control e selecione **New -> Class**. Depois, é só dar o nome Controle.





## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

A Classe Controle irá implementar os métodos declarados nas Interfaces Funcionais. Essa implementação é como no exemplo abaixo:

```
public static ICalculoArea P = (a, b, c) -> (a + b + c) / 2;
```

A expressão mostrada acima é uma **Função Lambda**.

*Funções Lambda fazem parte do Java 8 no contexto de Programação Funcional. Uma Função Lambda pode ser criada como se fosse um método público e estático, sem haver necessidade de criar um objeto para utilizá-la.*

Função Lambda **P** implementa o método **operacao** da interface funcional **ICalculoArea**; recebe 3 parâmetros double (a, b, c) e retorna o valor que é calculado à direita da seta -> ou seja, a partir de a, b e c, retorna a soma dos 3 dividida por 2. Esse cálculo vai ser utilizado no cálculo final da área do triângulo.

```
package control;
```

```
public class Controle {  
  
    public static ICalculoTres TIPOTRI = (a, b, c) -> (a  
    == b & b == c & c == a) ? "O Triângulo é Equilátero<br>"  
    : (a == b | a == c | b == c) ? "O Triângulo  
    é Isósceles<br>" : "O Triângulo é Escaleno<br>";  
  
    public static ICalculoArea P = (a, b, c) -> (a + b +  
    c) / 2;  
  
    public static ICalculoArea AREA = (a, b, c) ->  
    Math.sqrt(  
        P.operacao(a, b, c) * (P.operacao(a, b, c) -  
    a) * (P.operacao(a, b, c) - b) * (P.operacao(a, b, c) -  
    c));  
  
}
```



## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

A área do triângulo é calculada por duas **Funções Lambda**. **P** calcula a soma dos 3 lados e divide por 2; **AREA** usa o cálculo de **P** para produzir o valor final da área do triângulo.

Note a maneira como utilizamos em **AREA** o cálculo que é feito em **P**: **P.operacao**. Dessa forma, "chamamos" a função Lambda **P** e recebemos o resultado.

O cálculo da área de um triângulo a partir das medidas de seus 3 lados é feito da seguinte forma:

$$A = \sqrt{p(p-a)(p-b)(p-c)}, \text{ sendo}$$
$$p = \frac{a+b+c}{2}$$

Na classe controle, ainda existe a **Função Lambda TIPOTRI**. Essa função recebe 3 valores double e executa um **ternário**. Se os valores dos 3 lados são iguais, retorna que o triângulo é equilátero. Se dois desses lados são iguais, retorna que o triângulo é isósceles, caso contrário, retorna que é escaleno.

---

Precisamos criar um **Servlet**, para receber via **AJAX**, os valores dos 3 lados do triângulo.

Para criar um novo **Servlet**, clicamos sobre o nome do pacote control com o botão direito, e selecionamos **New -> Servlet**. Depois, damos um nome ao **Servlet** e clicamos next, na tela seguinte, também clicamos em next e chegamos na tela abaixo:



# Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

**Create Servlet**  
Specify modifiers, interfaces to implement, and method stubs to generate.

Modifiers:  public  abstract  final

Interfaces:  Add... Remove

Which method stubs would you like to create?

<input checked="" type="checkbox"/> Constructors from superclass	<input type="checkbox"/> destroy	<input type="checkbox"/> getServletConfig
<input checked="" type="checkbox"/> Inherited abstract methods	<input type="checkbox"/> service	<input checked="" type="checkbox"/> doGet
<input type="checkbox"/> init	<input type="checkbox"/> doPut	<input type="checkbox"/> doDelete
<input type="checkbox"/> getServletInfo	<input type="checkbox"/> doOptions	<input type="checkbox"/> doTrace
<input checked="" type="checkbox"/> doPost		
<input type="checkbox"/> doHead		

? < Back Next > Finish Cancel

No nosso caso, só vamos utilizar o método **Post**, então desmarcamos a opção **doGet**.





## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

### ControlTriangulo.java

```
package control;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/ControlTriangulo")
public class ControlTriangulo extends
HttpServlet {
    private static final long serialVersionUID = 1L;

    public ControlTriangulo() {
        super();
    }

    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {

        Double l1 =
Double.parseDouble(request.getParameter("l1"));
        Double l2 =
Double.parseDouble(request.getParameter("l2"));
        Double l3 =
Double.parseDouble(request.getParameter("l3"));

        request.setAttribute("tipo",
Controle.TIPOTRI.operacao(l1, l2, l3));

        request.setAttribute("area", "A Área do Triângulo
é: " + Controle.AREA.operacao(l1, l2, l3));
```



## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

```
        request.getRequestDispatcher("destino.jsp").forward(request, response);
    }
}
```

O Servlet **ControlTriangulo** recebe via **POST**, os valores dos 3 lados do triângulo. **Recebe esses valores como Strings**. Converte esses valores para **Double**.

```
Double l1 = Double.parseDouble(request.getParameter("l1"));
```

O Servlet envia para a **expression language tipo**, o valor retornado pela **Função Lambda TIPOTRI**, da Classe Controle.

```
request.setAttribute("tipo", Controle.TIPOTRI.operacao(l1, l2, l3));
```

Da mesma forma, o Servlet envia para a **expression language area**, o valor calculado pela **Função Lambda AREA**, da Classe Controle.

```
request.setAttribute("area", "A área do Triângulo é: " + Controle.AREA.operacao(l1, l2, l3));
```

O recurso **destino.jsp** vai receber os atributos **tipo** e **area** mencionados acima e vai exibi-los no navegador.

```
request.getRequestDispatcher("destino.jsp").forward(request, response);
```

---

Precisamos criar a **página JSP** com o formulário em que vamos digitar os 3 lados do triângulo.

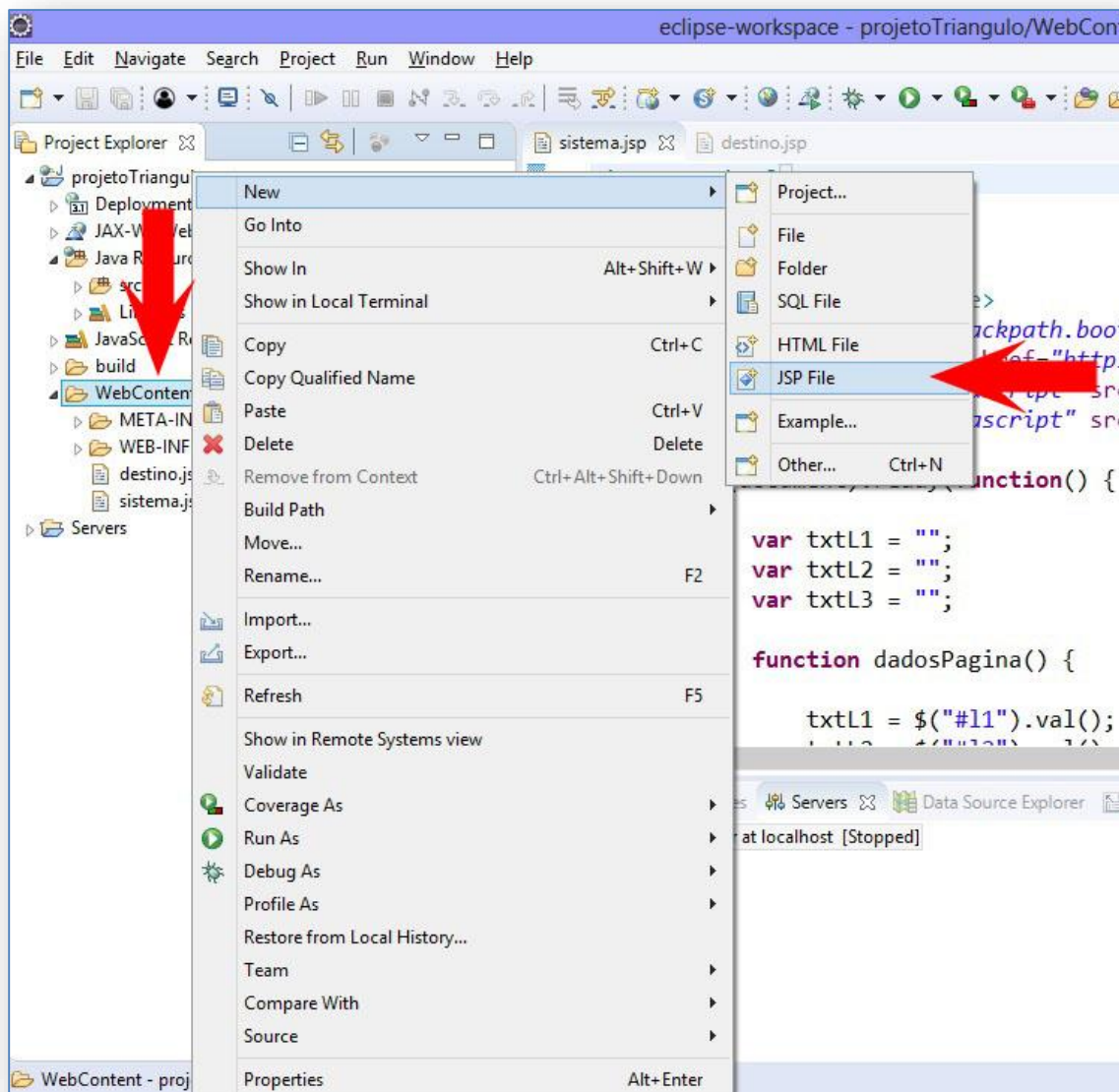


# Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

Para criar uma nova página JSP, primeiro clicamos com o botão direito em **WebContent** e selecionamos **New -> JSP File**





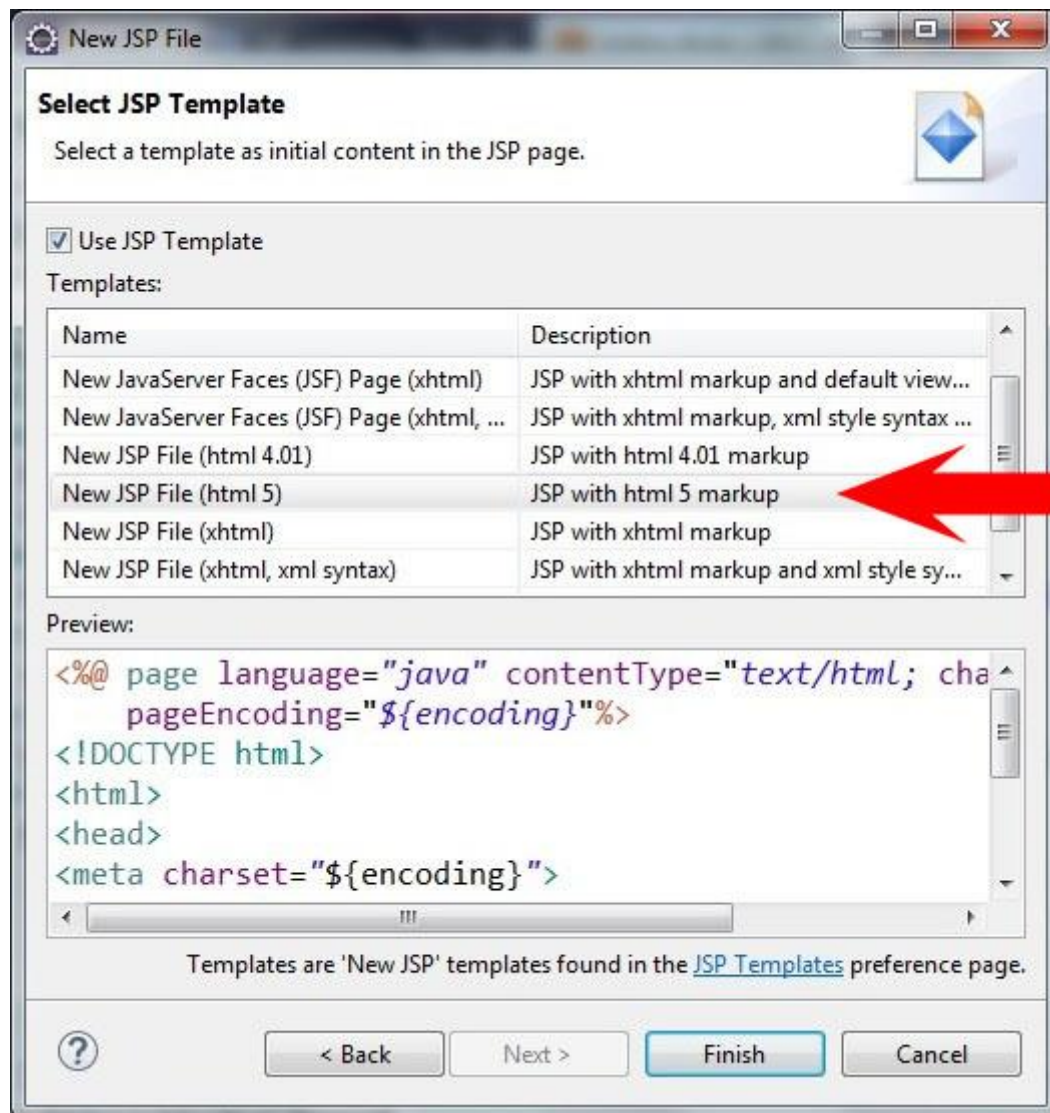
# Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

Em seguida, damos um nome ao arquivo JSP e clicamos Next.

Para o nosso projeto, na tela seguinte, selecionamos o template pronto **New JSP File (html 5)**, como na imagem abaixo. Depois clicamos em Finish.





# Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

## sistema.jsp

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <title>Triângulo</title>
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.css" />
    <script type="text/javascript"
src="https://code.jquery.com/jquery-3.2.1.js"></script>
    <script type="text/javascript"
src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
    <script>
      $(document).ready(function() {

        var txtL1 = "";
        var txtL2 = "";
        var txtL3 = "";

        function dadosPagina() {

          txtL1 = $("#l1").val();
          txtL2 = $("#l2").val();
          txtL3 = $("#l3").val();

        }

        $("#btn1").on('click', function() {

          dadosPagina();
          $.ajax({
            url : "ControlTriangulo",
            method : "POST",
            data : {
```



## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

```
        l1 : txtL1,
        l2 : txtL2,
        l3 : txtL3
    }
    }).done(function(resultado) {
        $("#resposta").html(resultado);
    });
});
});
</script>
</head>
<body>
    <div class="container">
        <form id="form1" style="margin-top: 20px;">
            <h5>Entre com os 3 lados do triângulo:</h5>
            Lado1<br><input type="number" step=0.1
name="L1" size="10" id="L1"><br>
            Lado2<br><input type="number" step=0.1
name="L2" size="10" id="L2"><br>
            Lado3<br><input type="number" step=0.1
name="L3" size="10" id="L3"><br>
            <br>
            <button type="button" id="btn1" class="btn btn-
primary">Enviar AJAX</button>
            <br>
        </form>
        <br><br>
        <div id="resposta" style="color: #4B0082;">Aqui
vai aparecer a resposta...</div>
    </div>
</body>
</html>
```



## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

Na tag <head>, vemos os links CDN de todos os recursos de que vamos necessitar: JQuery, JQuery-UI, Bootstrap.

Logo abaixo desses links temos a tag <script> que marca o início das funções JQuery e AJAX que vamos utilizar.

```
$(document).ready(function() {
```

No momento em que a página é carregada, inicia o código [veja abaixo] que "zera" os valores das variáveis `txtL1`, `txtL2` e `txtL3`, que vamos utilizar para envio dos dados do formulário html.

```
var txtL1 = "";  
var txtL2 = "";  
var txtL3 = "";
```

Abaixo desse código, existe o código da função que vai instanciar as variáveis `txtL1`, `txtL2` e `txtL3` com os valores inseridos no formulário html.

```
function dadosPagina() {  
  
    txtL1 = $("#l1").val();  
    txtL2 = $("#l2").val();  
    txtL3 = $("#l3").val();  
  
}
```

`.val()` indica que os valores vêm do formulário HTML, no caso, dos inputs com os IDs `l1`, `l2` e `l3`.

E, terminando esse trecho de código, temos a função que é disparada no evento do clique no botão `btn1` e, via AJAX, chama o servlet `ControlTriangulo` enviando via POST os valores de `txtL1`, `txtL2` e `txtL3`.



## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

```
$("#btn1").on('click', function() {  
  
    dadosPagina();  
    $.ajax({  
        url : "ControlTriangulo",  
        method : "POST",  
        data : {  
            l1 : txtL1,  
            l2 : txtL2,  
            l3 : txtL3  
        }  
    }).done(function(resultado) {  
        $("#resposta").html(resultado);  
    });  
});
```

Abaixo da tag `<script>` temos o formulário html onde vamos digitar os valores dos 3 lados do triângulo. As classes CSS utilizadas nesse layout são do Bootstrap, que incluímos na tag `<head>`.

Ex:

```
class="container"  
class="btn btn-primary"
```

---

## destino.jsp

```
<%@ page language="java" contentType="text/html;  
charset=UTF-8"  
pageEncoding="UTF-8"%>
```

```
<strong> ${tipo} ${area} </strong>
```





## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

Essa é a última página de nosso projeto. Nela serão inseridos os dados enviados pelo **Servlet**, via **expression language**.

```
.done(function(resultado) {  
    $("#resposta").html(resultado);  
});
```

O trecho de código acima está na página  **sistema.jsp**, e “pega” o que foi enviado pelo **Servlet** em resposta ao processamento da requisição enviada via  **formulário HTML**, e essa resposta, que é o conteúdo da página  **destino.jsp**, é inserida no html da  **div** com o  **id resposta**, sobrescrevendo o texto “Aqui vai aparecer a resposta...”

### Tela do sistema ao carregar

Entre com os 3 lados do triângulo:

Lado1

Lado2

Lado3

Aqui vai aparecer a resposta...



## Java

Prof. Fernando Gomes

Projeto com Programação Funcional, Funções Lambda, JSP, AJAX, JQUERY, Bootstrap, Expression Language, Interface Funcional e Servlet.

### Tela do sistema com o resultado do cálculo

Entre com os 3 lados do triângulo:

Lado1  
10

Lado2  
11

Lado3  
11

**Enviar AJAX**

**O Triângulo é Isósceles**  
**A área do Triângulo é: 48.98979485566356**

Escola de Nerds  
www.cotiinformatica.com.br