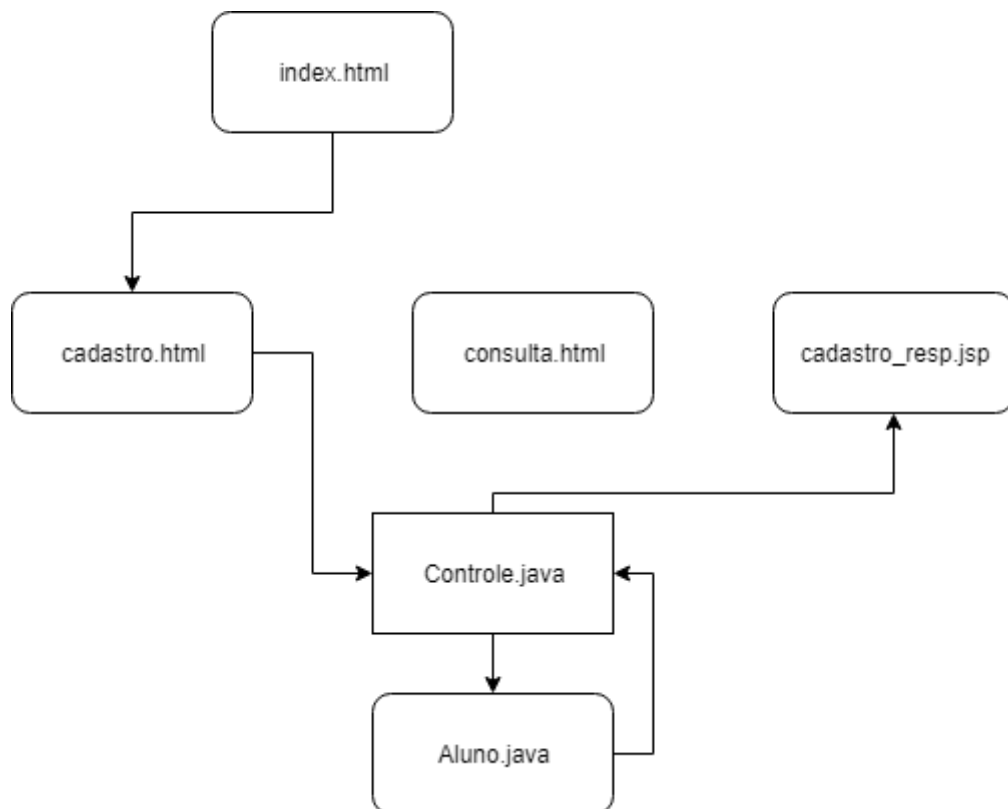


TUTORIAL PARA CRIAÇÃO DE UMA APLICAÇÃO COM HTML, JSP E SERVLET COM NETBEANS



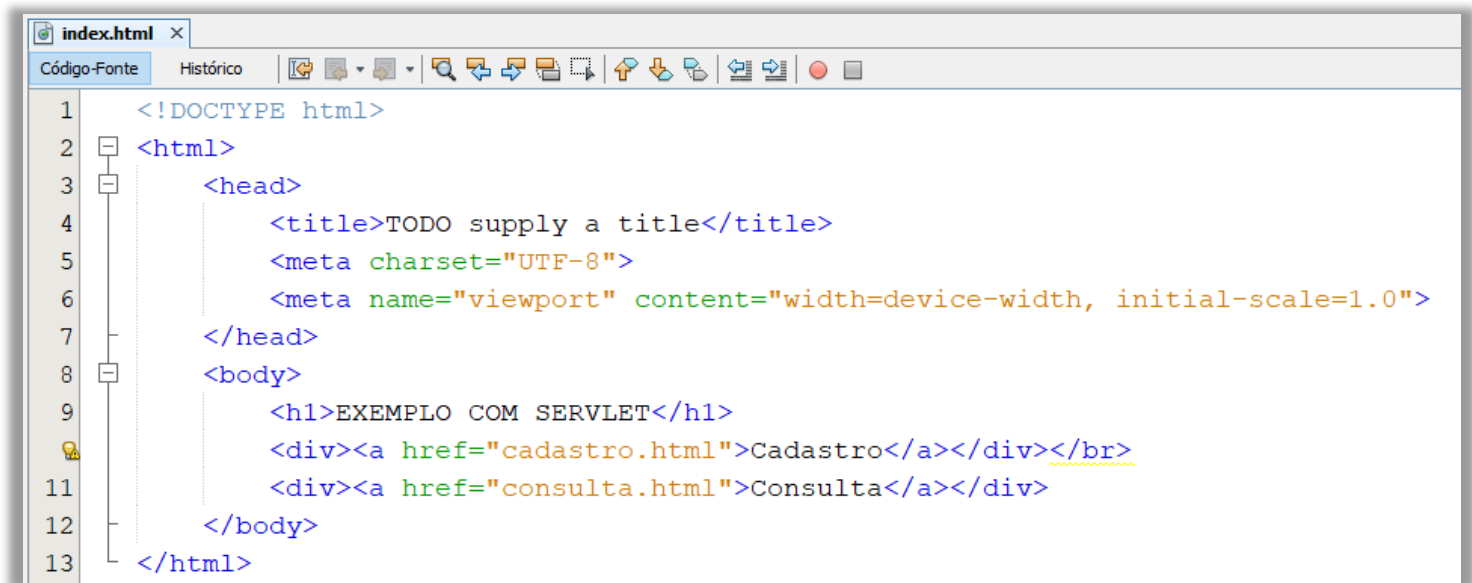
Sumário

CONSTRUINDO UMA APLICAÇÃO COM SERVLET	2
CRIANDO PROJETO	2
CRIANDO NOVO HTML (FRONT-END)	3
CRIANDO UMA CLASSE JAVA (BACK-END)	4
DEFININDO CÓDIGO DA CLASSE JAVA (GET/SET)	5
CRIANDO SERVLET (BACK-END)	6

CONSTRUINDO UMA APLICAÇÃO COM SERVLET

CRIANDO PROJETO

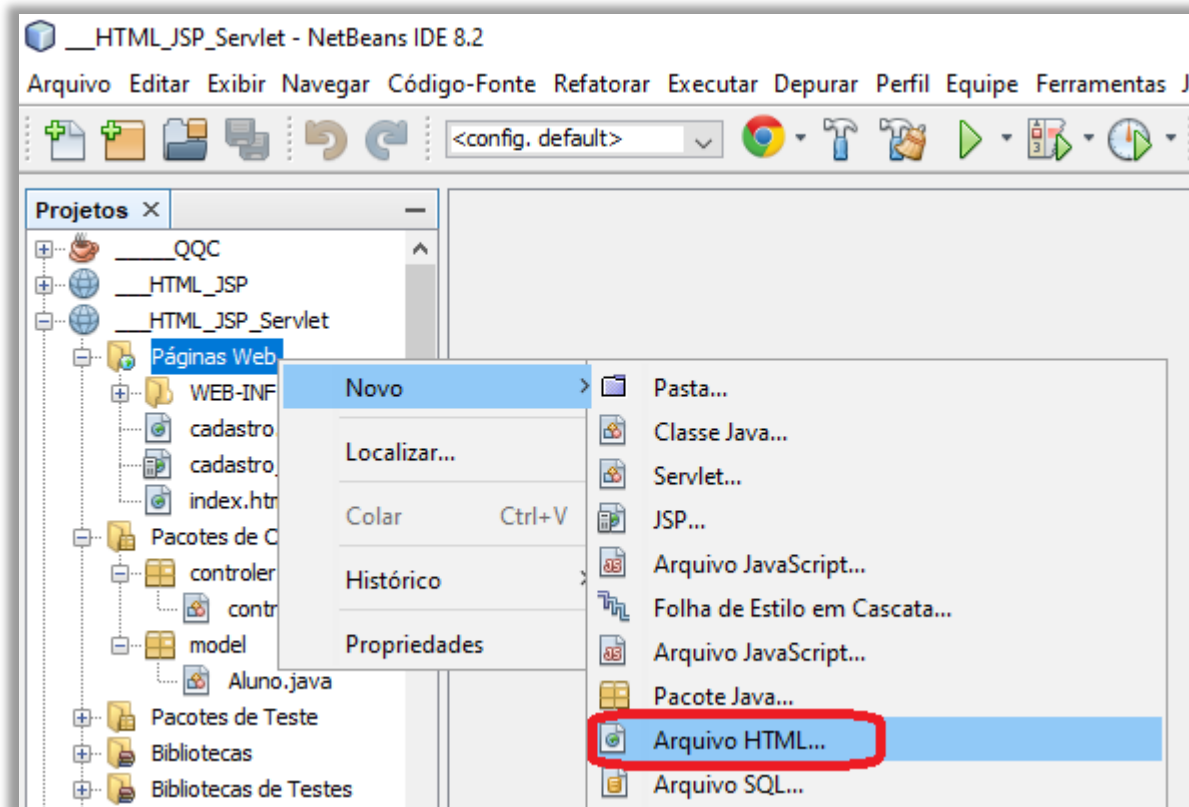
- 1) Entre no NetBeans e acione menu: Arquivo / Novo Projeto / Categoria: Java Web / Aplicação Web Próximo >
- 2) De um nome para o projeto (Ex: TesteServlet), escolha diretório de armazenamento e Próximo >
- 3) Escolha o servidor Web e clique em Finalizar
- 4) No arquivo index, adicione os links conforme apresentado abaixo:



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>TODO supply a title</title>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  </head>
8  <body>
9      <h1>EXEMPLO COM SERVLET</h1>
10     <div><a href="cadastro.html">Cadastro</a></div><br>
11     <div><a href="consulta.html">Consulta</a></div>
12 </body>
13 </html>
```

CRIANDO NOVO HTML (FRONT-END)

5) Com o botão direito sobre “Páginas Web” do seu projeto, Acione Novo / Arquivo HTML:



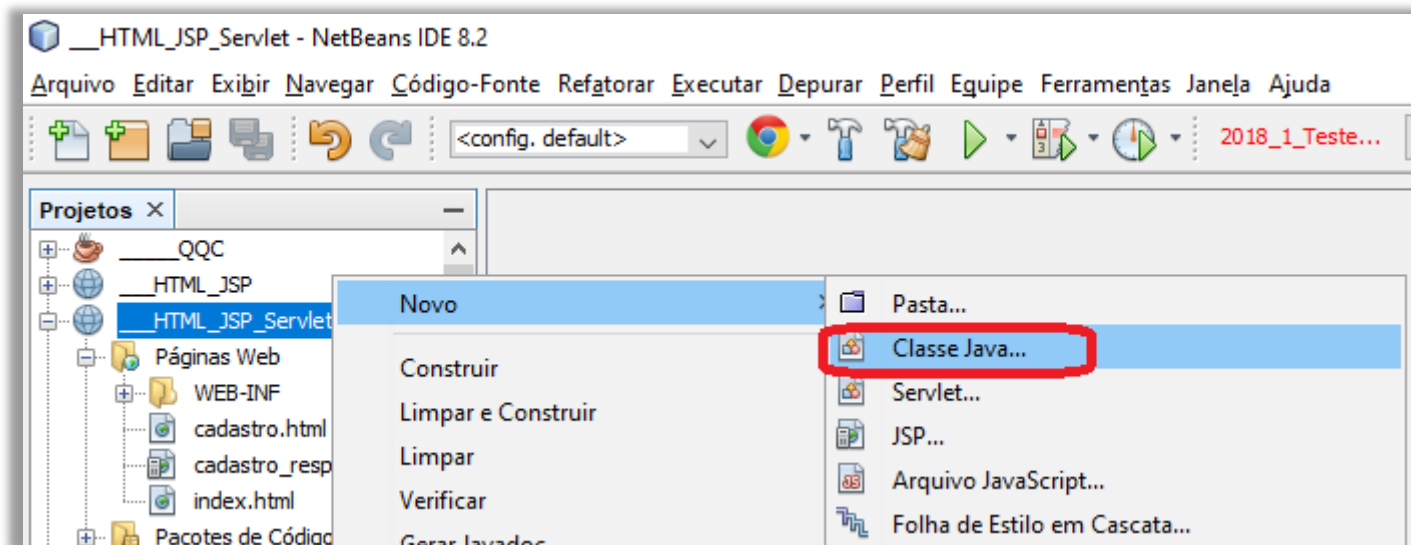
6) Dê o nome de cadastro e clique em [Finalizar](#)

7) Adicione o código (front-end) para entrada de dados:

```
cadastro.html x
Código-Fonte  Histórico
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>TODO supply a title</title>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  </head>
8  <body>
9      <h3>Cadastro - servlet</h3>
10     <form method="POST" action="controle" >
11     <pre>
12         Nome:  <input type="text" name="nome">
13         Nota 1: <input type="number" name="nota1">
14         Nota 2: <input type="number" name="nota2">
15                 <input type="submit" value="Enviar" >
16     </pre>
17     </form>
18 </body>
19 </html>
```

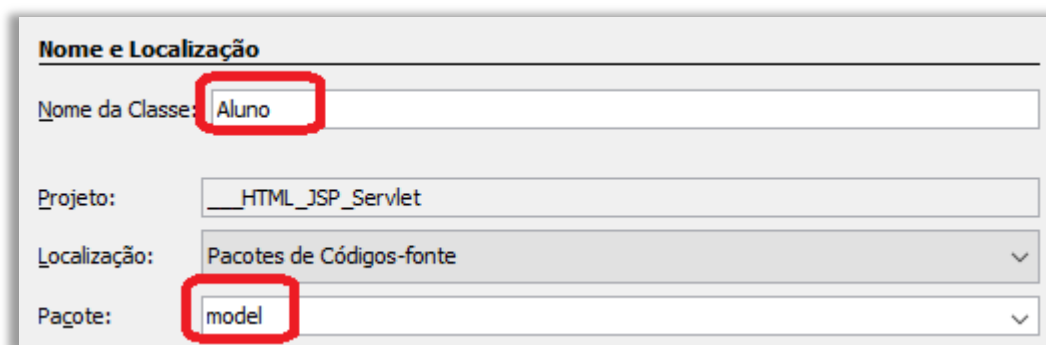
CRIANDO UMA CLASSE JAVA (BACK-END)

8) Com o botão direito sobre o nome do seu projeto, Acione Novo / Classe Java:



9) Defina o nome da classe e o pacote conforme tela abaixo e clique em

Finalizar

A screenshot of the 'Nome e Localização' (Name and Location) dialog box in NetBeans. The dialog has several fields: 'Nome da Classe:' (Class Name) with the value 'Aluno' highlighted by a red rectangle; 'Projeto:' (Project) with the value '___HTML_JSP_Servlet'; 'Localização:' (Location) with the value 'Pacotes de Códigos-fonte'; and 'Pacote:' (Package) with the value 'model' highlighted by a red rectangle. A 'Finalizar' (Finish) button is visible in the top right corner of the dialog.

DEFININDO CÓDIGO DA CLASSE JAVA (GET/SET)

10) Abra o código gerado e coloque o código na classe, seguindo as orientações abaixo:

a) Digite os atributos nome, nota1, nota2, media e situação conforme apresentado no código;

b) Clique com o botão direito dentro do escopo da classe e acione a opção “inserir código” / Getter e Setter. Desta forma, crie cada método get e set conforme apresentado abaixo.

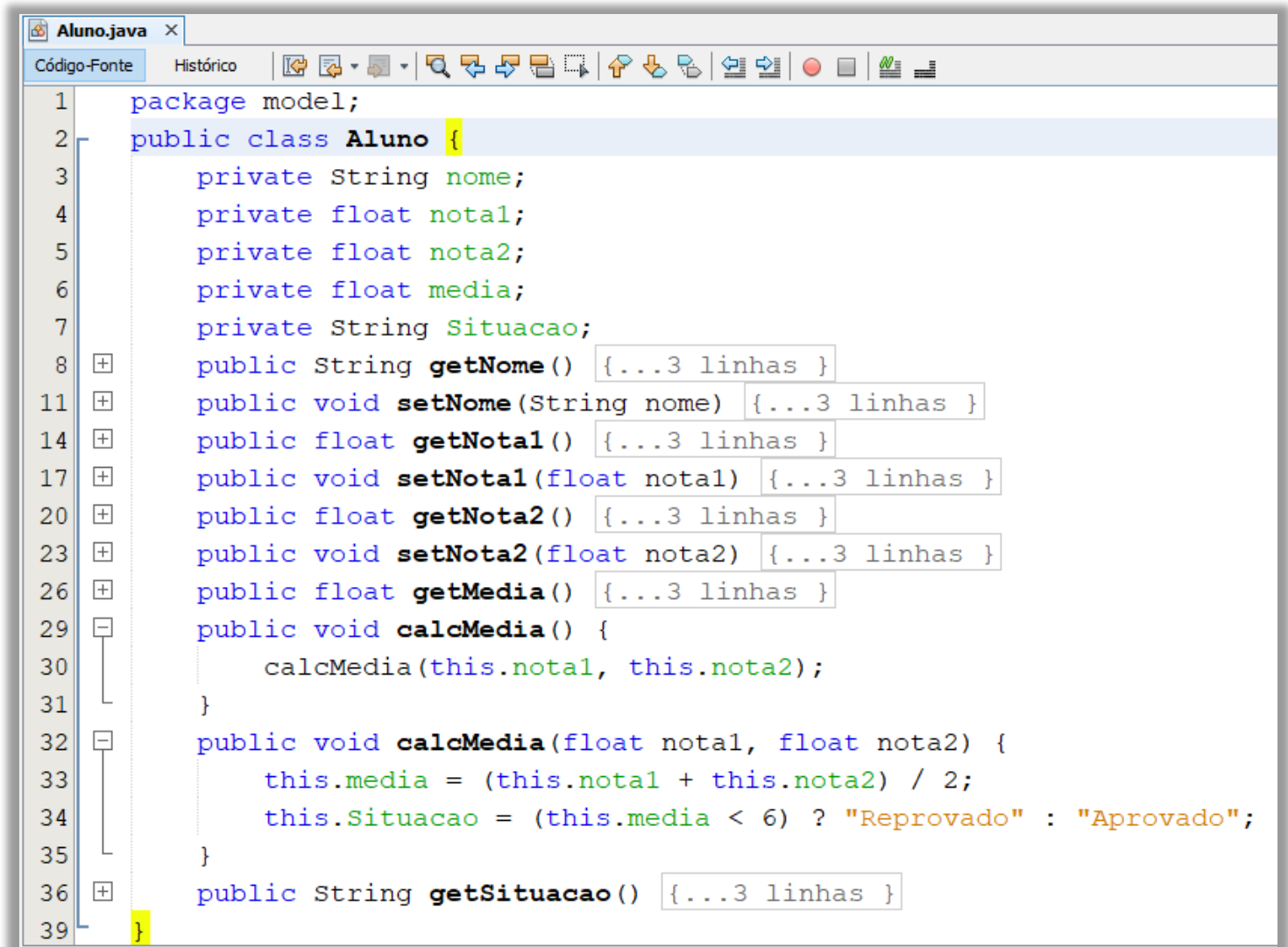
The image shows a screenshot of an IDE window titled "Aluno.java". The code is as follows:

```
1 package model;
2
3 public class Aluno {
4
5     private String nome;
6     private float nota1;
7     private float nota2;
8     private float media;
9     private String Situacao;
10
11
12
13
14 }
15
```

Annotations and diagrams:

- A bracket on the right side of the class body (lines 4 to 14) is labeled "ESCOPO DA CLASSE".
- A bracket on the right side of the attributes (lines 5 to 9) is labeled "ATRIBUTOS".
- A bracket on the right side of the methods (lines 11 to 13) is labeled "MÉTODOS".
- A bracket on the left side of the closing brace (line 14) is labeled "FORA DO ESCOPO DA CLASSE".
- A red-bordered box at the bottom right contains the text: "NÃO PROGRAMA NADA AQUI!".

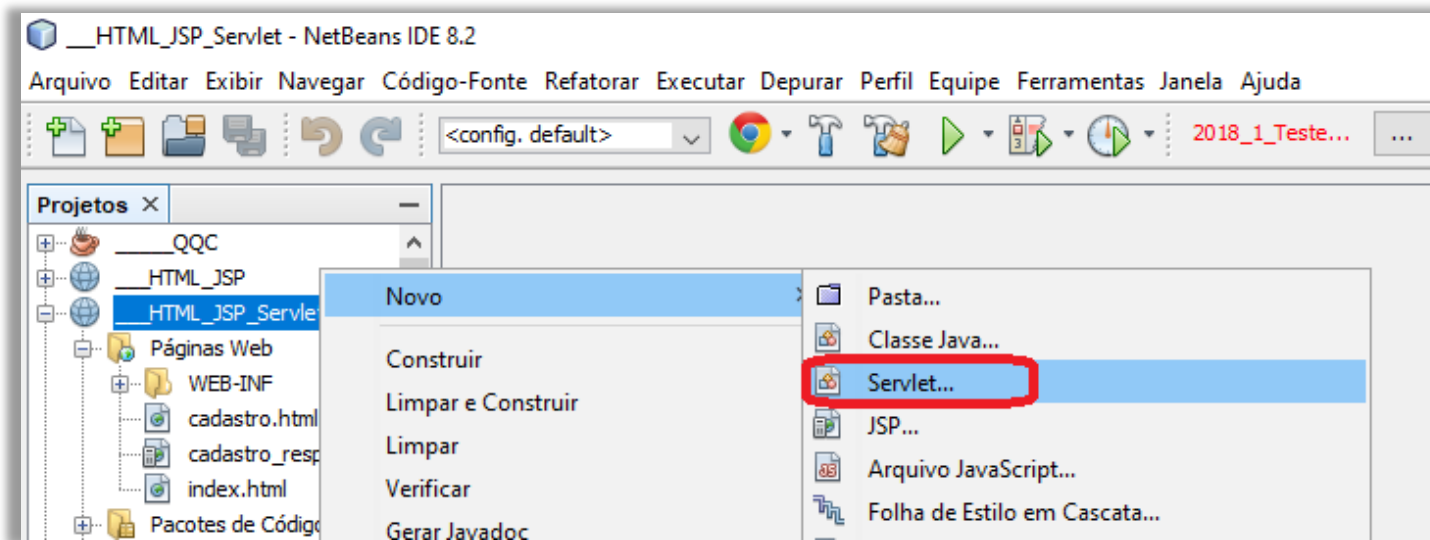
c) Digite manualmente os dois métodos calcMedia apresentados abaixo.



```
1 package model;
2 public class Aluno {
3     private String nome;
4     private float nota1;
5     private float nota2;
6     private float media;
7     private String Situacao;
8     public String getNome() {...3 linhas }
11    public void setNome(String nome) {...3 linhas }
14    public float getNota1() {...3 linhas }
17    public void setNota1(float nota1) {...3 linhas }
20    public float getNota2() {...3 linhas }
23    public void setNota2(float nota2) {...3 linhas }
26    public float getMedia() {...3 linhas }
29    public void calcMedia() {
30        calcMedia(this.nota1, this.nota2);
31    }
32    public void calcMedia(float nota1, float nota2) {
33        this.media = (this.nota1 + this.nota2) / 2;
34        this.Situacao = (this.media < 6) ? "Reprovado" : "Aprovado";
35    }
36    public String getSituacao() {...3 linhas }
39 }
```

CRIANDO SERVLET (BACK-END)

11) Com o botão direito sobre o nome do seu projeto, Acione Novo / Servlet:



12) Dê o nome da classe e do pacote conforme demonstrado abaixo e clique em Finalizar

Nome e Localização

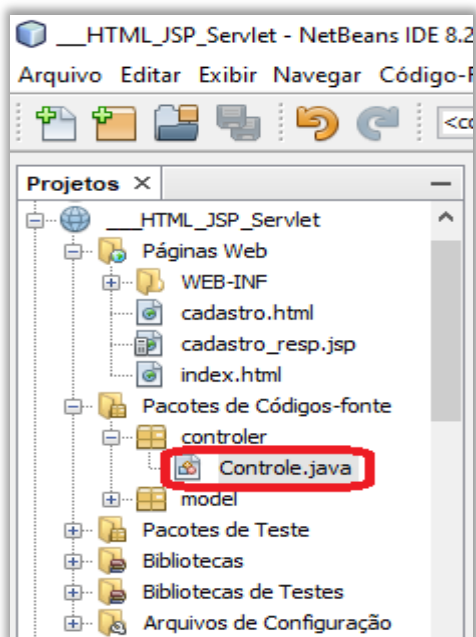
Nome da Classe: Controle

Projeto: __HTML_JSP_Servlet

Localização: Pacotes de Códigos-fonte

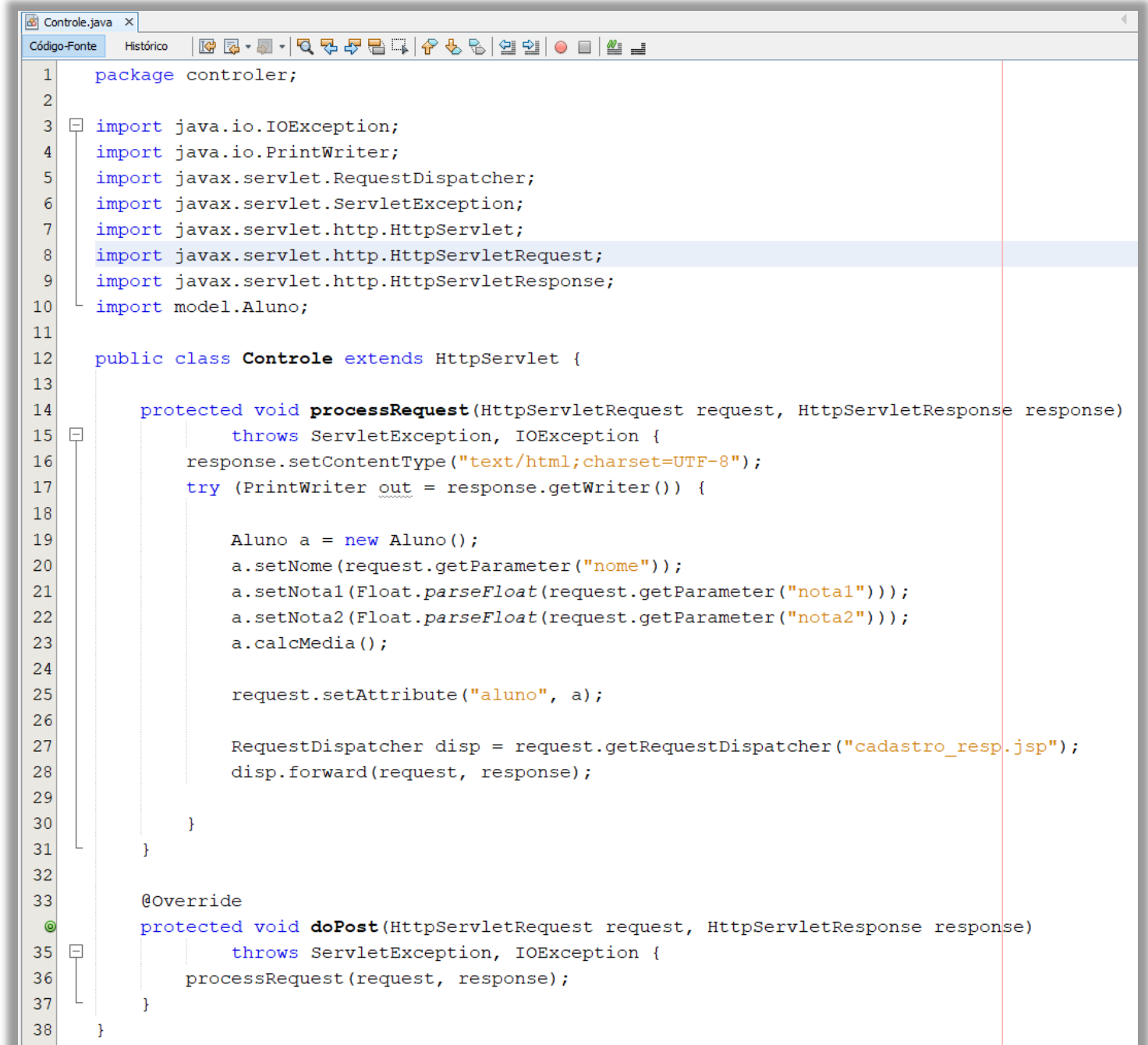
Pacote: controler

- 13) Seu arquivo Servlet (Controle.java) deve ter ficado na pasta controler conforme demonstrado abaixo.



CODIFICANDO SERVELT

14) Verifique o código e deixe sua classe Servlet conforme demonstrado abaixo:



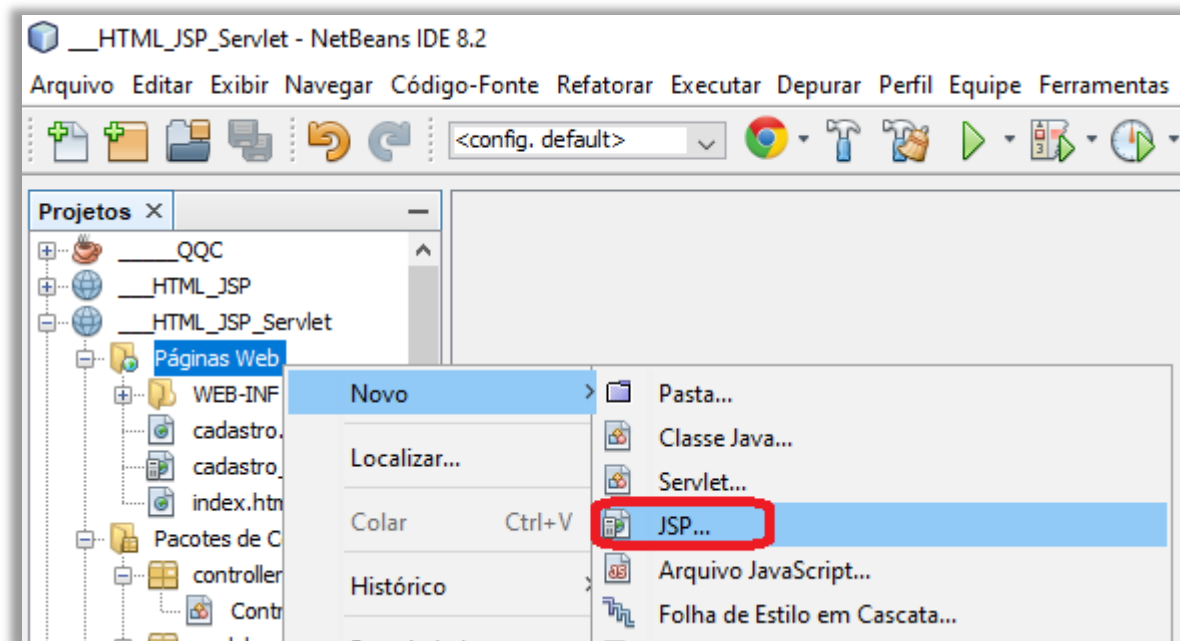
```
1 package controller;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import javax.servlet.RequestDispatcher;
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10 import model.Aluno;
11
12 public class Controle extends HttpServlet {
13
14     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
15         throws ServletException, IOException {
16         response.setContentType("text/html;charset=UTF-8");
17         try (PrintWriter out = response.getWriter()) {
18
19             Aluno a = new Aluno();
20             a.setNome(request.getParameter("nome"));
21             a.setNota1(Float.parseFloat(request.getParameter("nota1")));
22             a.setNota2(Float.parseFloat(request.getParameter("nota2")));
23             a.calcMedia();
24
25             request.setAttribute("aluno", a);
26
27             RequestDispatcher disp = request.getRequestDispatcher("cadastro_resp.jsp");
28             disp.forward(request, response);
29         }
30     }
31
32     @Override
33     protected void doPost(HttpServletRequest request, HttpServletResponse response)
34         throws ServletException, IOException {
35         processRequest(request, response);
36     }
37 }
38 }
```

Obs:

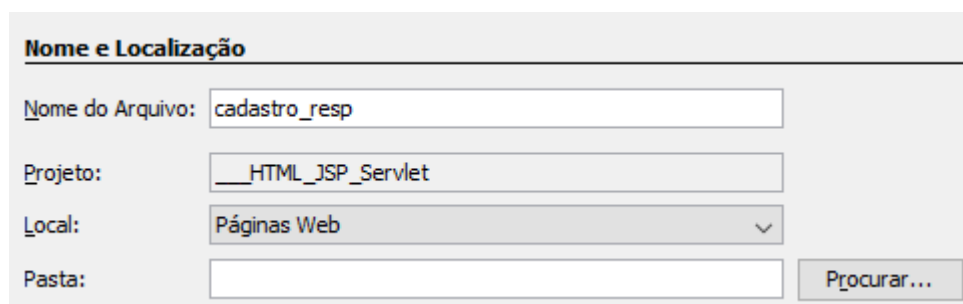
- a) Na linha 19, uma classe java (Aluno) é instanciada para armazenar os dados do aluno;
- b) Da 20 a 23, são passadas as informações para o objeto;
- c) Na linha 25 o objeto a está sendo armazenado em um atributo para leitura no front-end;
- d) Nas linhas 27 e 28 determina-se o JSP do front-end que receberá o retorno do Servlet.

CRIANDO JSP (FRONT-END)

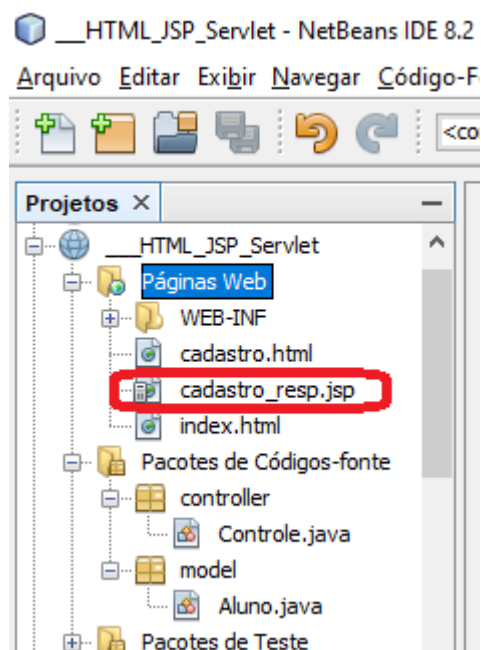
15) Com o botão direito sobre a pasta “Páginas Web” do seu projeto, Acione Novo / JSP:



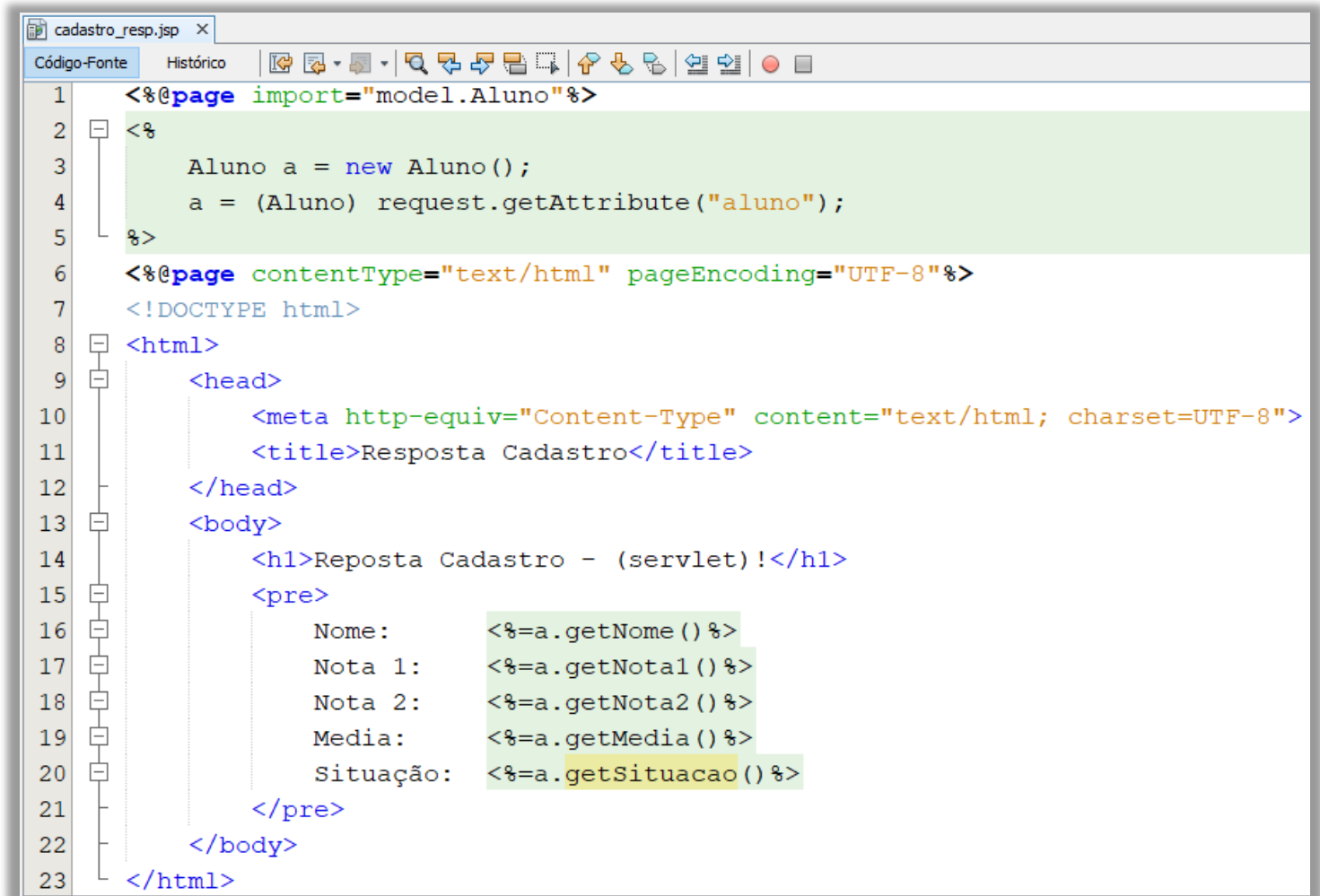
16) De o nome do JSP e clique em [Finalizar](#)



17) O arquivo cadastro_resp.jsp será criado conforme demonstrado abaixo:



18) Faça as alterações no arquivo conforme definido abaixo:



```
1 <%@page import="model.Aluno"%>
2 <%
3     Aluno a = new Aluno();
4     a = (Aluno) request.getAttribute("aluno");
5 %>
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>
7 <!DOCTYPE html>
8 <html>
9     <head>
10         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11         <title>Resposta Cadastro</title>
12     </head>
13     <body>
14         <h1>Resposta Cadastro - (servlet)!</h1>
15         <pre>
16             Nome:      <%=a.getNome() %>
17             Nota 1:    <%=a.getNota1() %>
18             Nota 2:    <%=a.getNota2() %>
19             Media:     <%=a.getMedia() %>
20             Situação:  <%=a.getSituacao() %>
21         </pre>
22     </body>
23 </html>
```

Obs:

- a) Na linha 1, uma TAG JSP que indica a referência à classe Aluno;
- b) Na linha 3 a classe aluno é instanciada;
- c) Na linha 4, obtêm-se o conteúdo do atributo “aluno”, fazendo um *casting* com a classe Aluno;
- d) Das linhas 16 a 20, a exibição dos dados do objeto aluno, recebidos do Servlet.