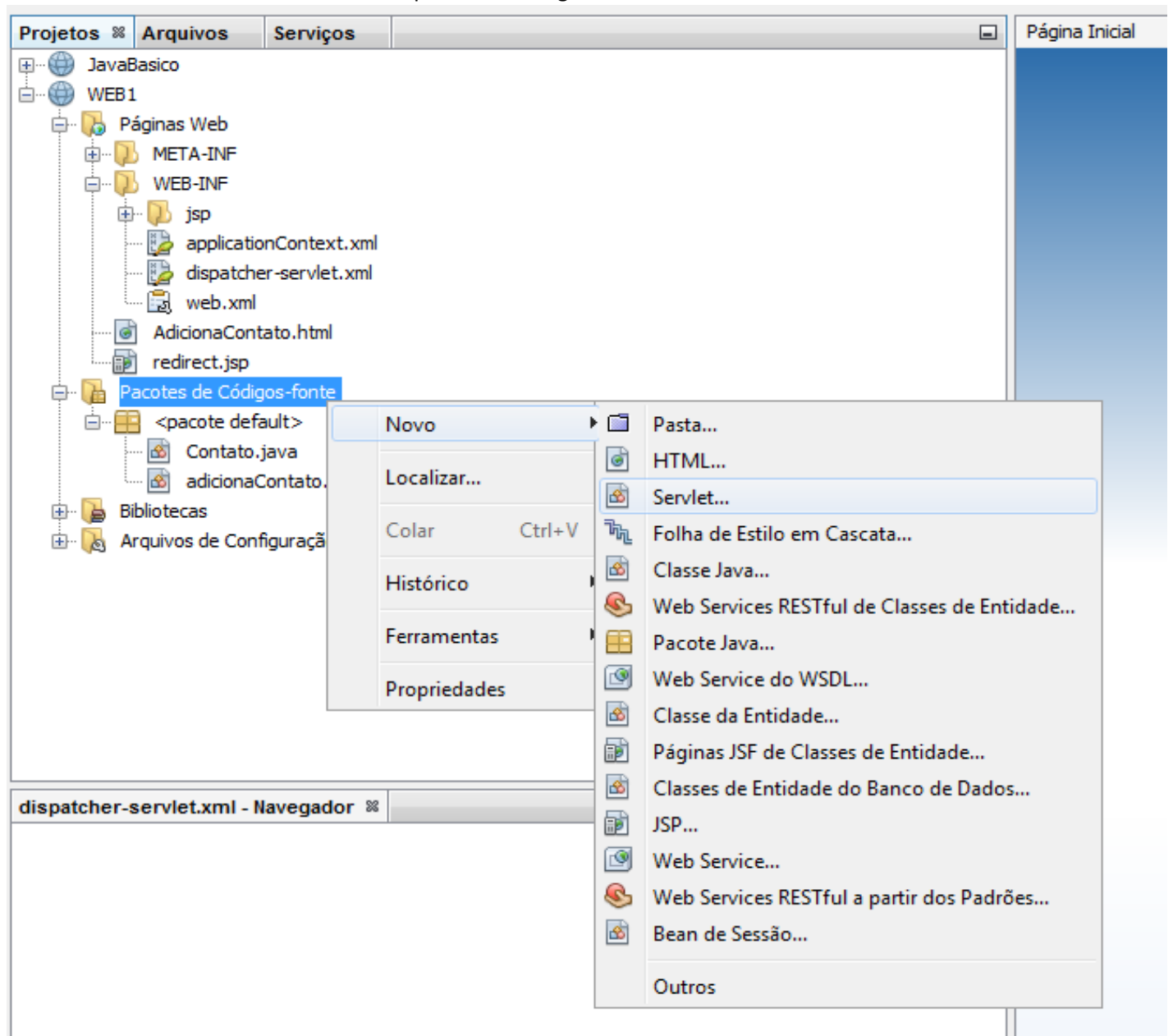
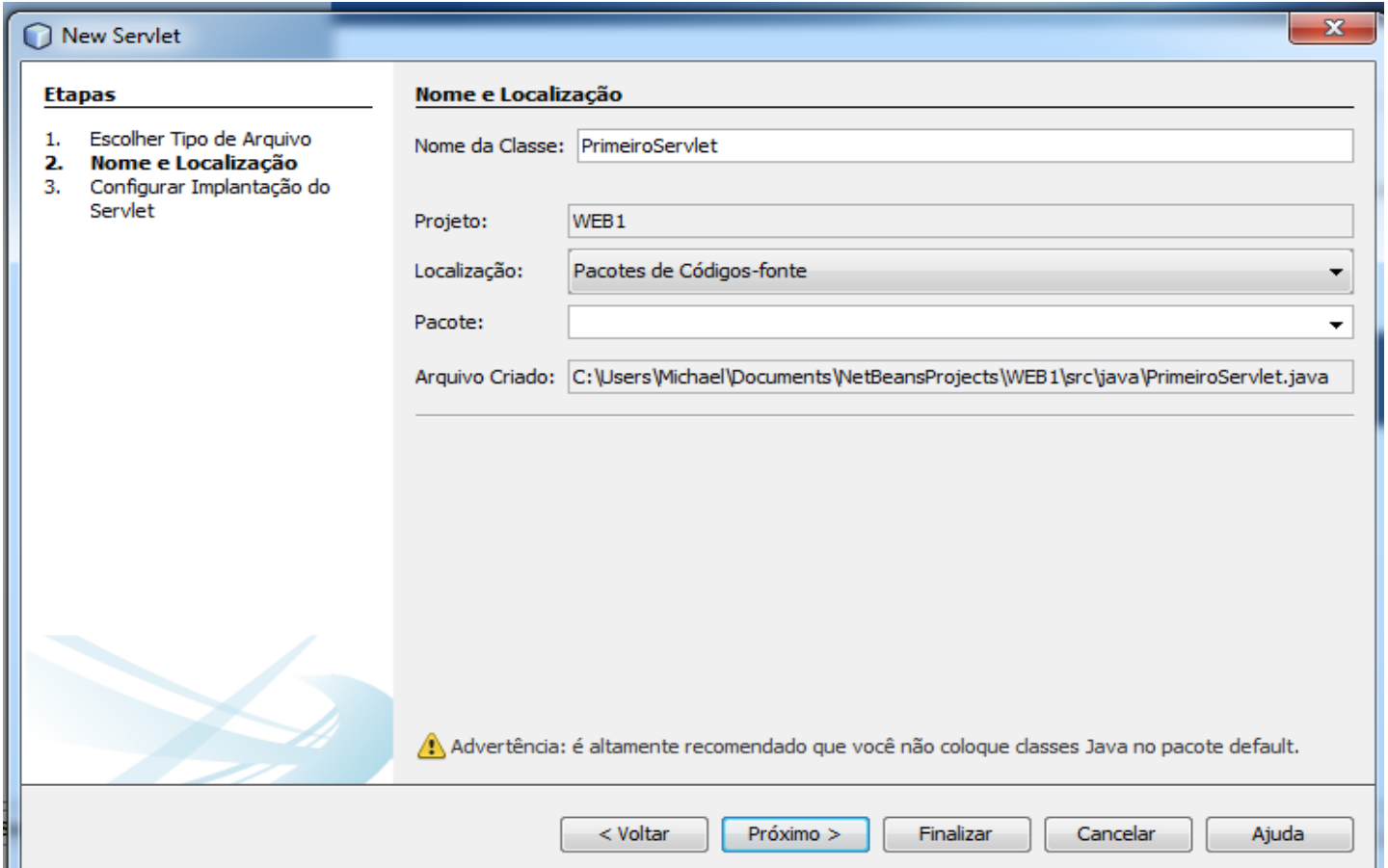


Passo a passo para trabalharmos com SERVLETS

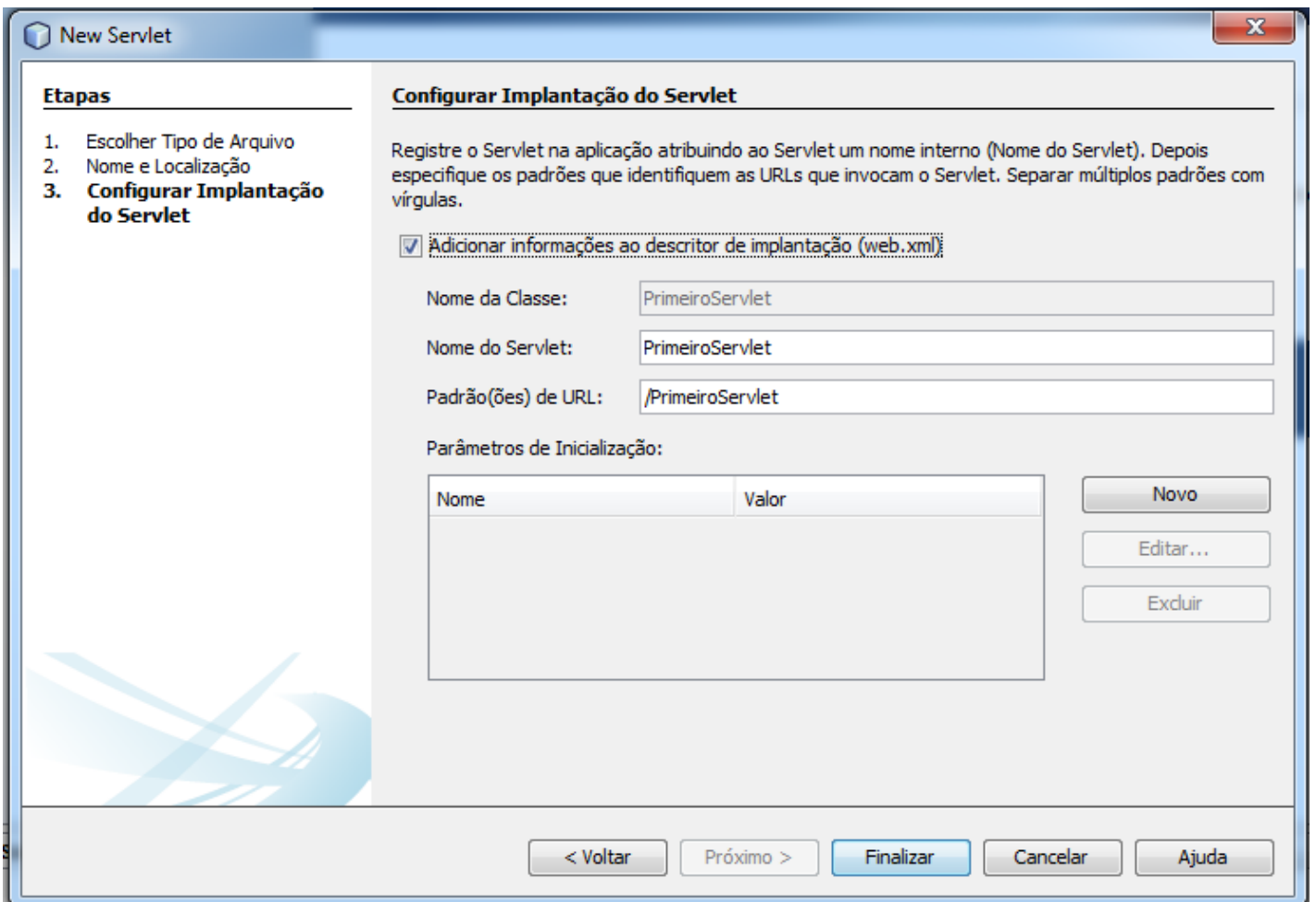
1 - Vamos criar um Servlet dentro de um pacote de código fonte.



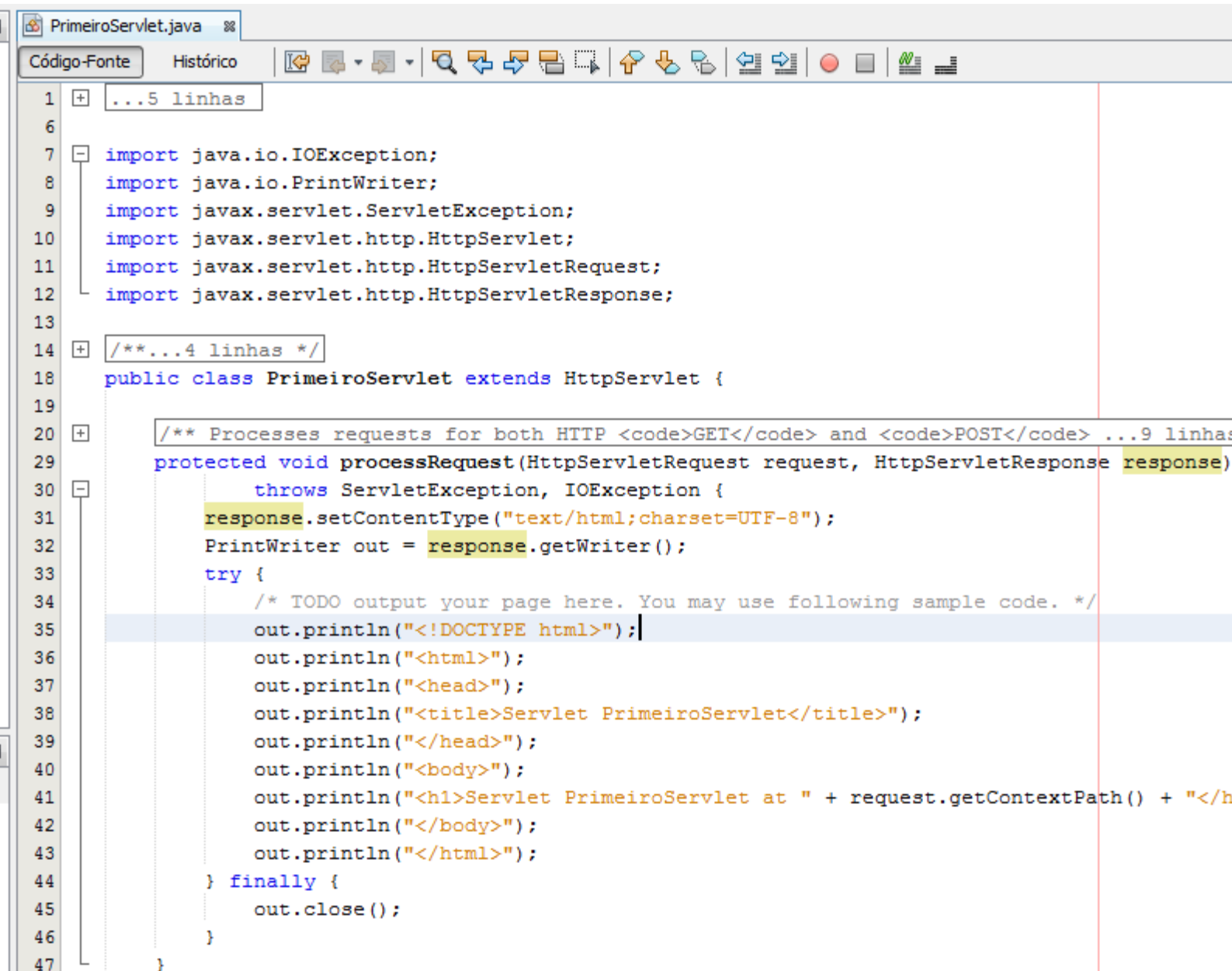
2 – Escolhemos um nome para nosso Servlet



3 – Selecionamos a opção para adicionar informações ao descritor WEB.XML, caso não seja selecionado, será necessário a inclusão manual no arquivo WEB.XML conforme visto em sala.



4 – Nosso servlet foi criado, observamos a estrutura dele abaixo:



```
1  ...5 linhas
2
3
4
5
6
7  import java.io.IOException;
8  import java.io.PrintWriter;
9  import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13
14 /**...4 linhas */
15
16
17
18 public class PrimeiroServlet extends HttpServlet {
19
20     /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 linhas
21
22     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
23         throws ServletException, IOException {
24         response.setContentType("text/html;charset=UTF-8");
25         PrintWriter out = response.getWriter();
26         try {
27             /* TODO output your page here. You may use following sample code. */
28             out.println("<!DOCTYPE html>");
29             out.println("<html>");
30             out.println("<head>");
31             out.println("<title>Servlet PrimeiroServlet</title>");
32             out.println("</head>");
33             out.println("<body>");
34             out.println("<h1>Servlet PrimeiroServlet at " + request.getContextPath() + "</h1>");
35             out.println("</body>");
36             out.println("</html>");
37         } finally {
38             out.close();
39         }
40     }
41 }
42
43
44
45
46
47
```

Podemos observar que ele importa do pacote javax.servlet a classe HttpServlet, ele utiliza basicamente o HTTP para tratamento de requisições e respostas, que é o protocolo de entrada utilizado na grande maioria das aplicações WEB.

Para escrevermos uma servlet, é criada uma classe Java que estenda HttpServlet e sobrescreva um método chamado processRequest. Esse método será o responsável por atender requisições e gerar as respostas adequadas.

Repare que o método recebe dois objetos que representam, respectivamente, a requisição feita pelo usuário e a resposta que será exibida no final (HttpServletRequest request e HttpServletResponse response)

É possível obter um objeto que represente a saída a ser enviada ao usuário através do método getWriter da variável response. E, a partir disso, utilizar um PrintWriter para imprimir algo na resposta do cliente:

Nosso primeiro exemplo de implementação não executa nada de lógica e apenas mostra uma página em HTML falando para o usuário que é nossa primeira servlet e usa um método request.getContextPath para falar qual é o local que a página está guardada.

