

Programação WEB

Guia para criação de um projeto Java web no Netbeans para fazer acesso de Login

Osvaldo Cesar Pinheiro de Almeida
(osvaldo.almeida@fatec.sp.gov.br)

Botucatu - SP
Agosto - 2022

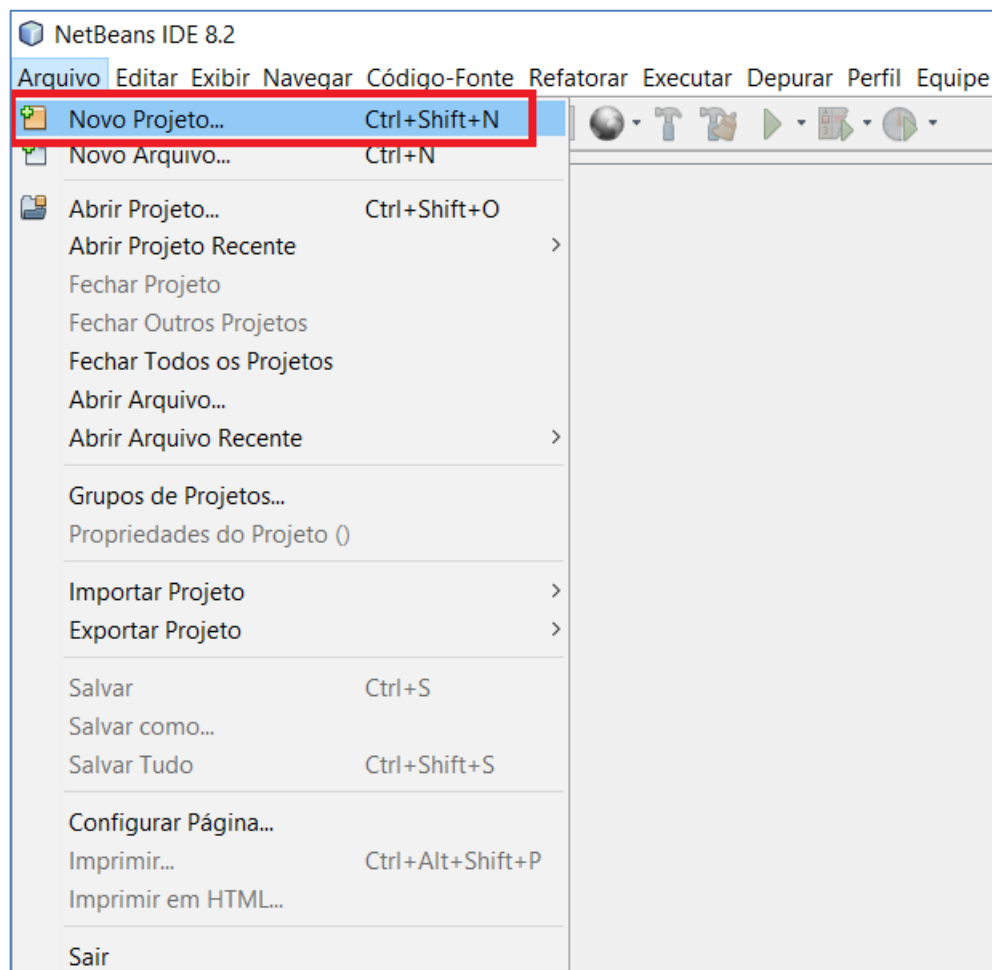
Inicialização

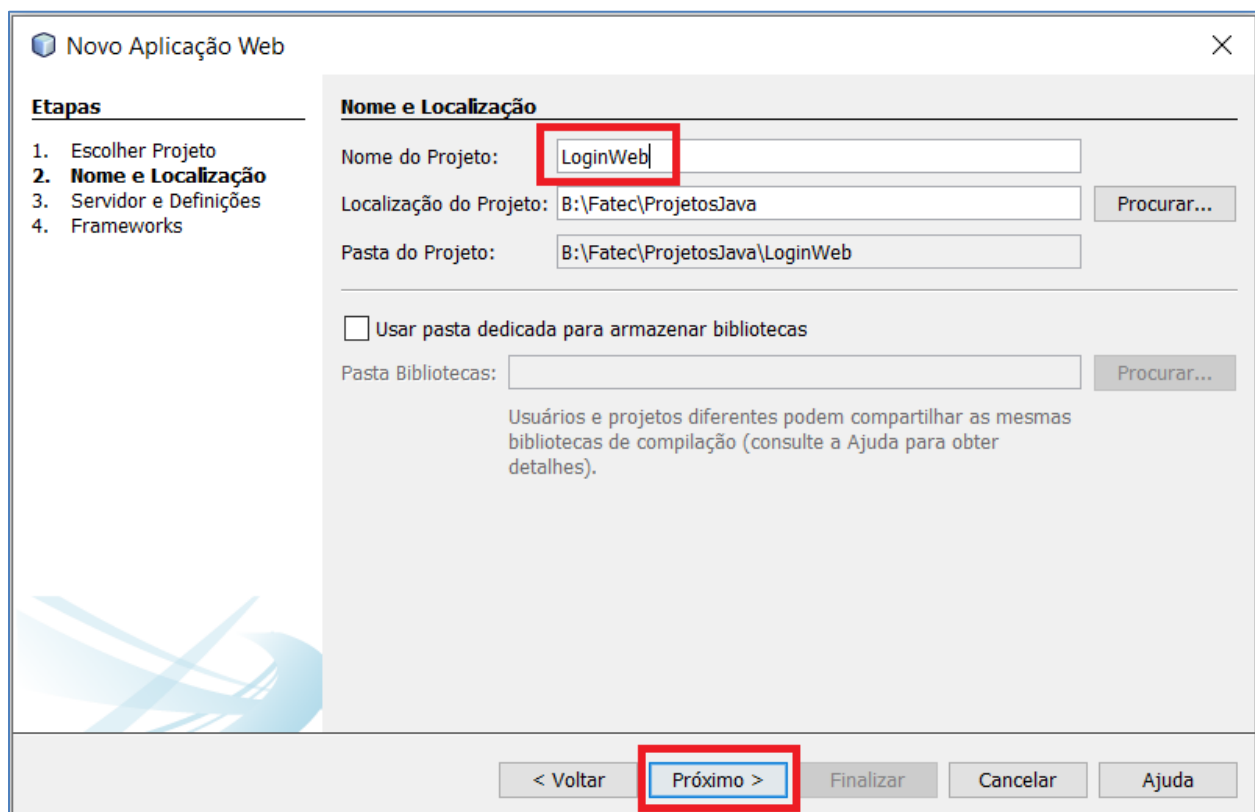
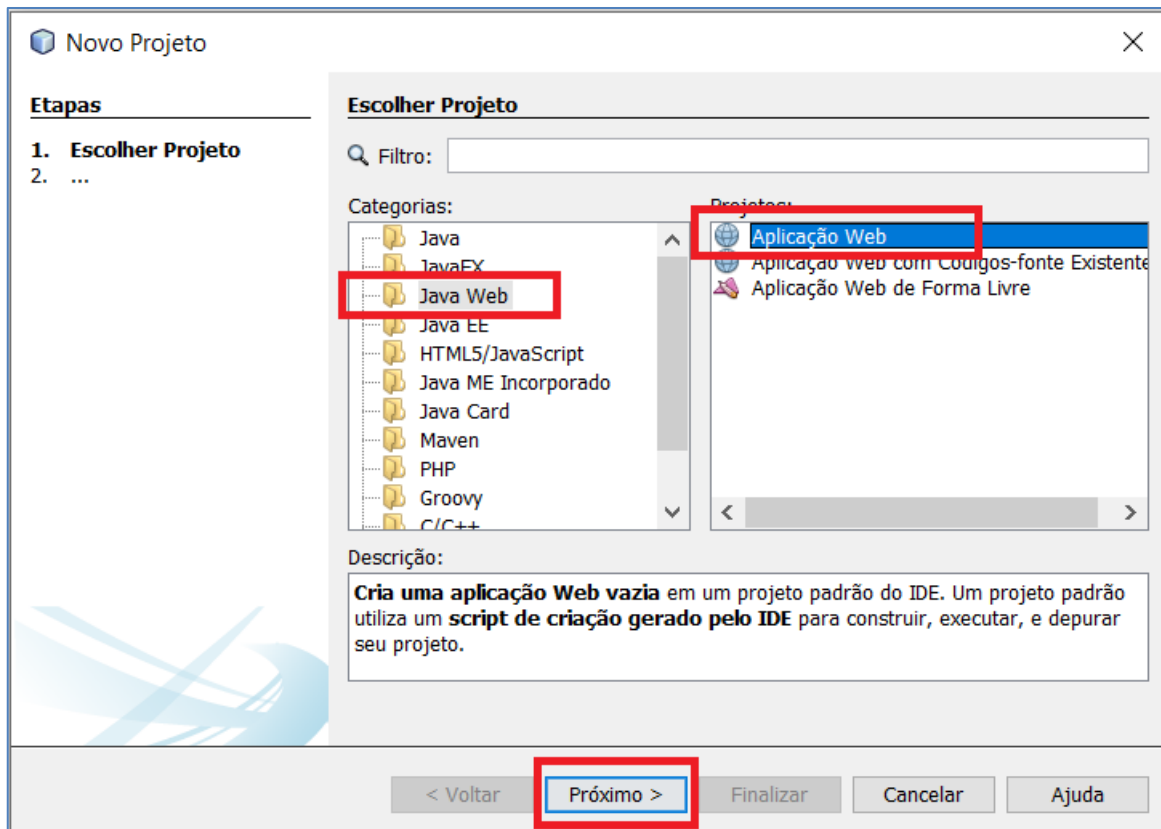
Requisitos iniciais:

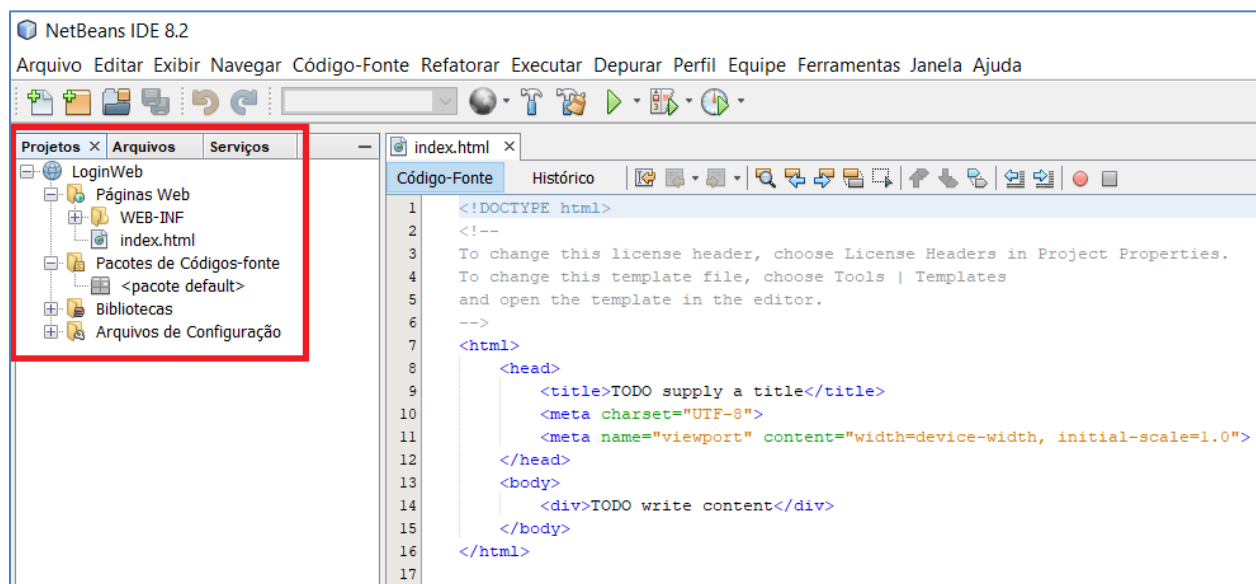
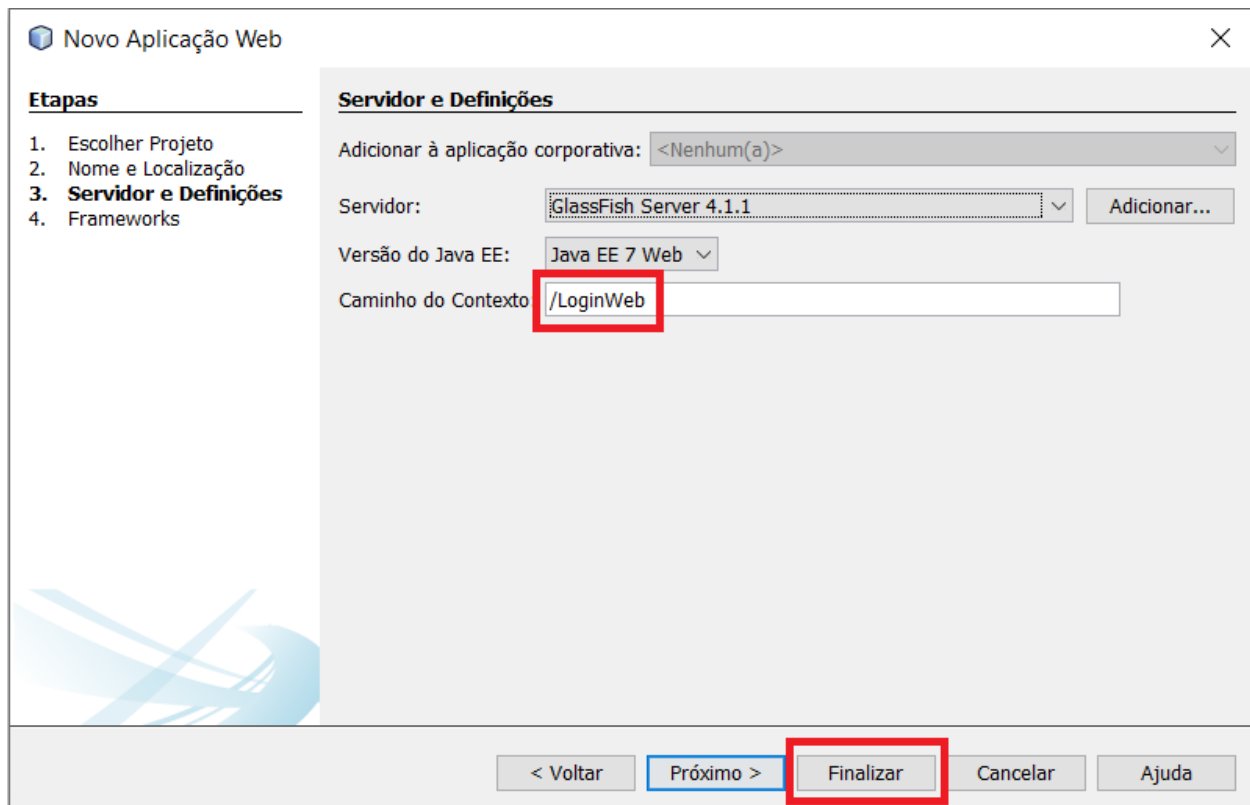
- Netbeans completo e Java (jdk) instalado - Versão recomendada do Netbeans 8.2 ou maior;
- Observe as imagens com destaque em vermelho.

Criando Projeto

Usando o Netbeans iremos criar um novo projeto:

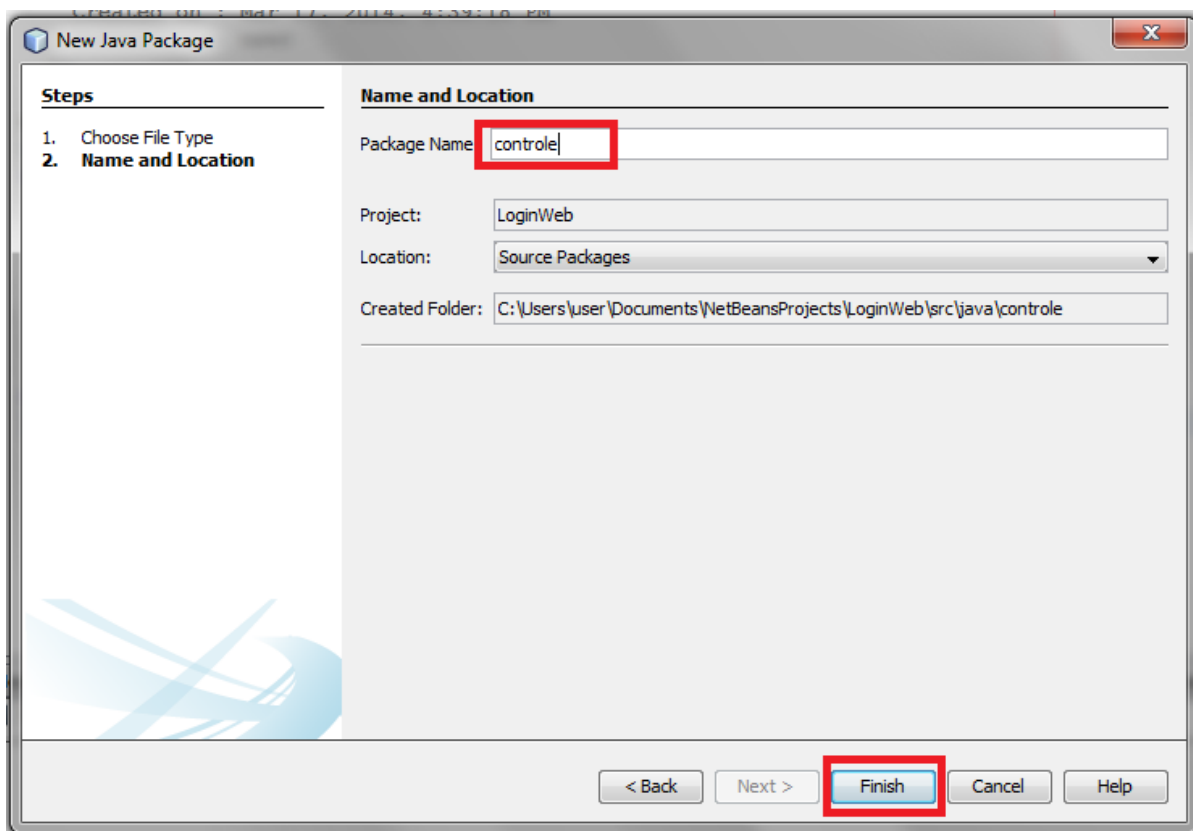
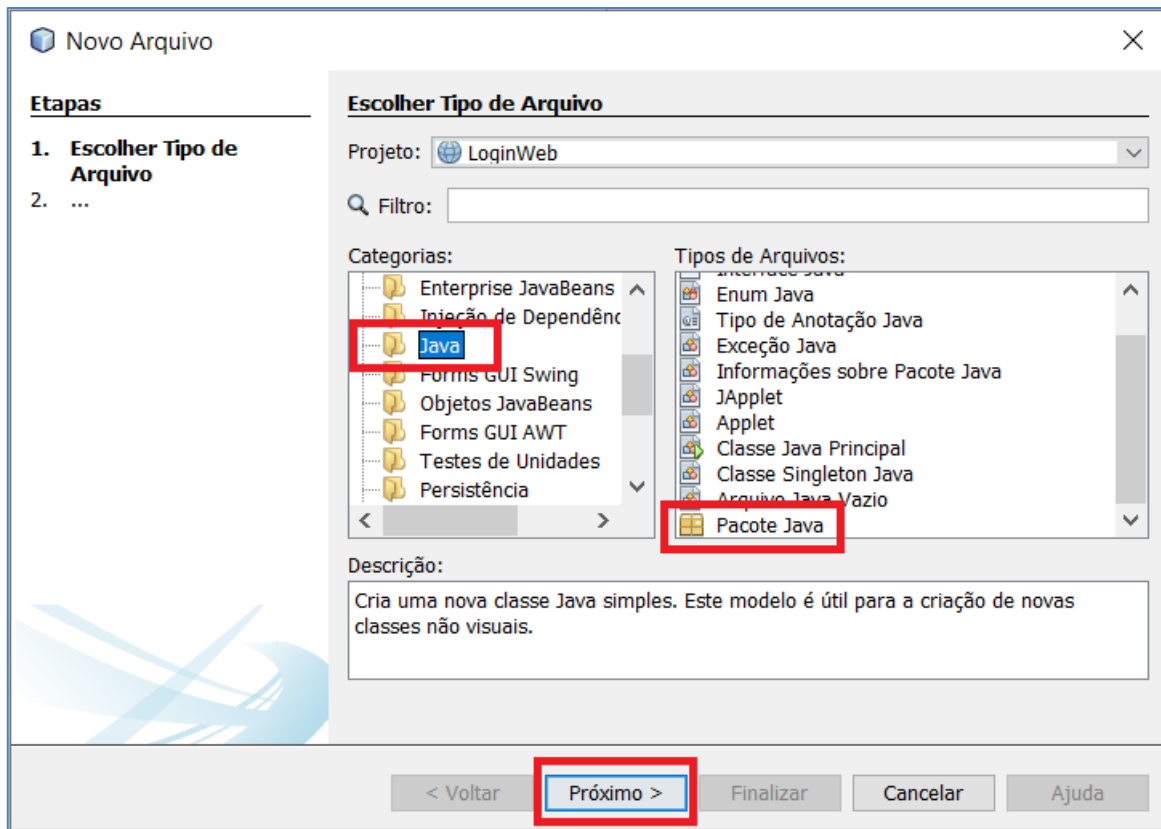


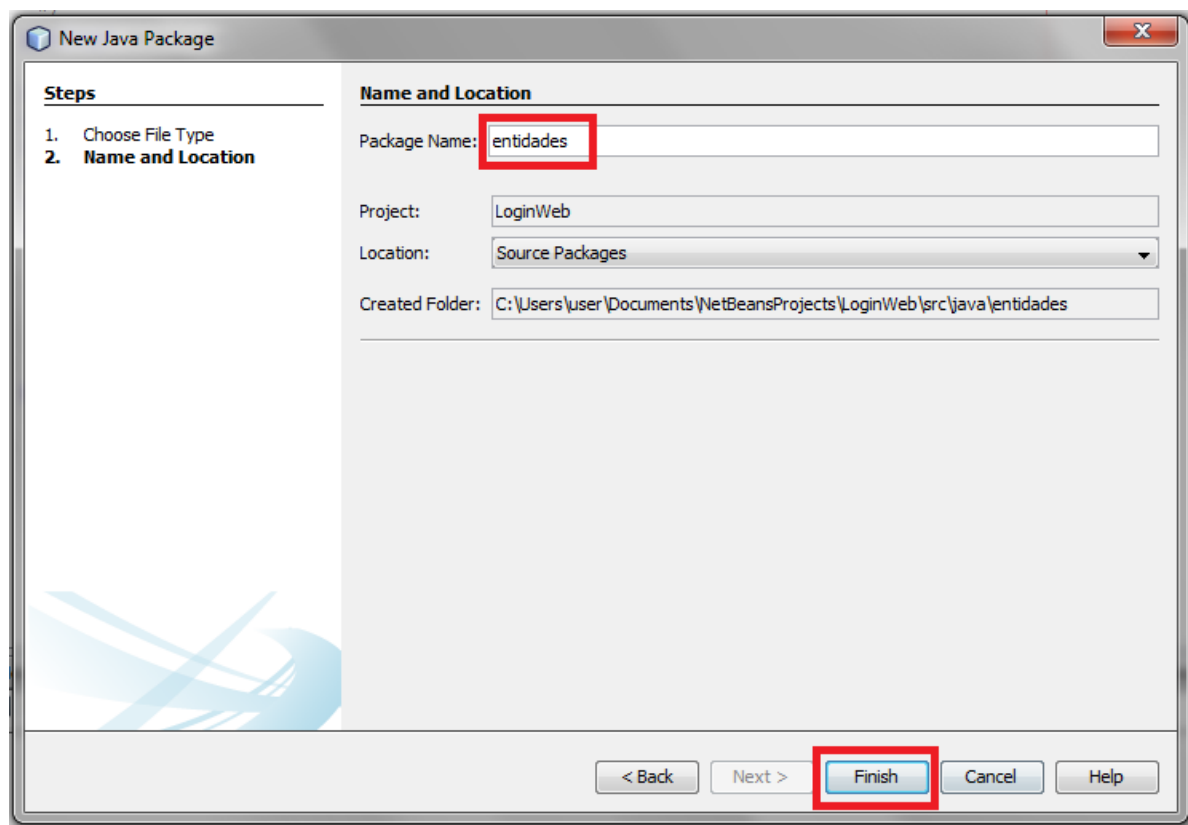
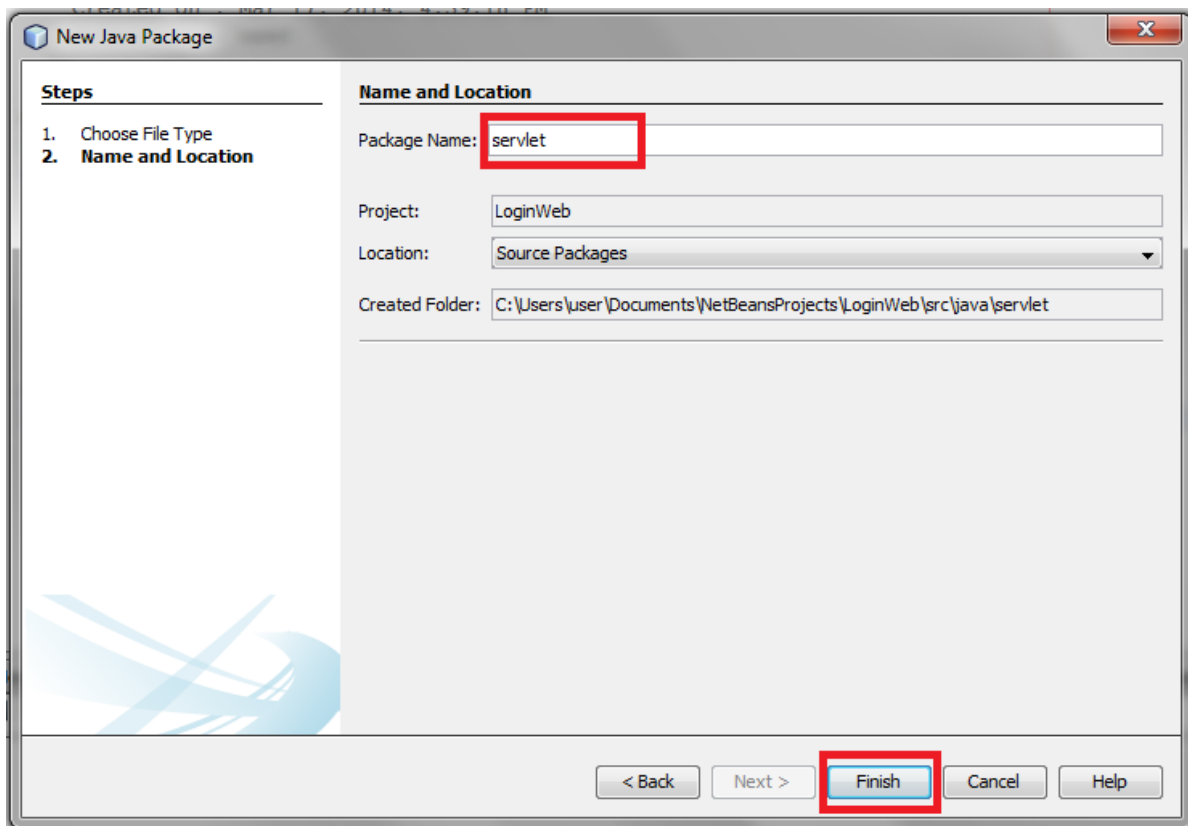




Criando as Classes Base do Projeto

Nessa etapa devem ser criados três pacotes (*packages*) para organizar os códigos de programação. O pacote serve como um diretório para organizar os arquivos de código fonte que são criados. Criar o pacote “servlet”, o pacote “controle” e o pacote “entidades”.

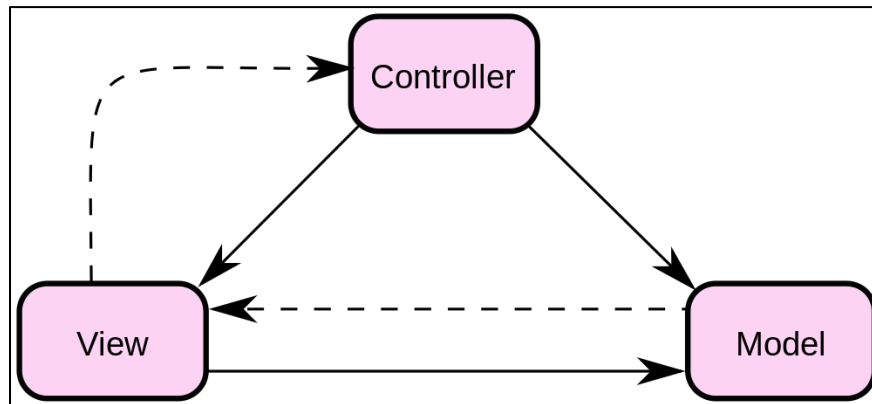




Usando o MVC

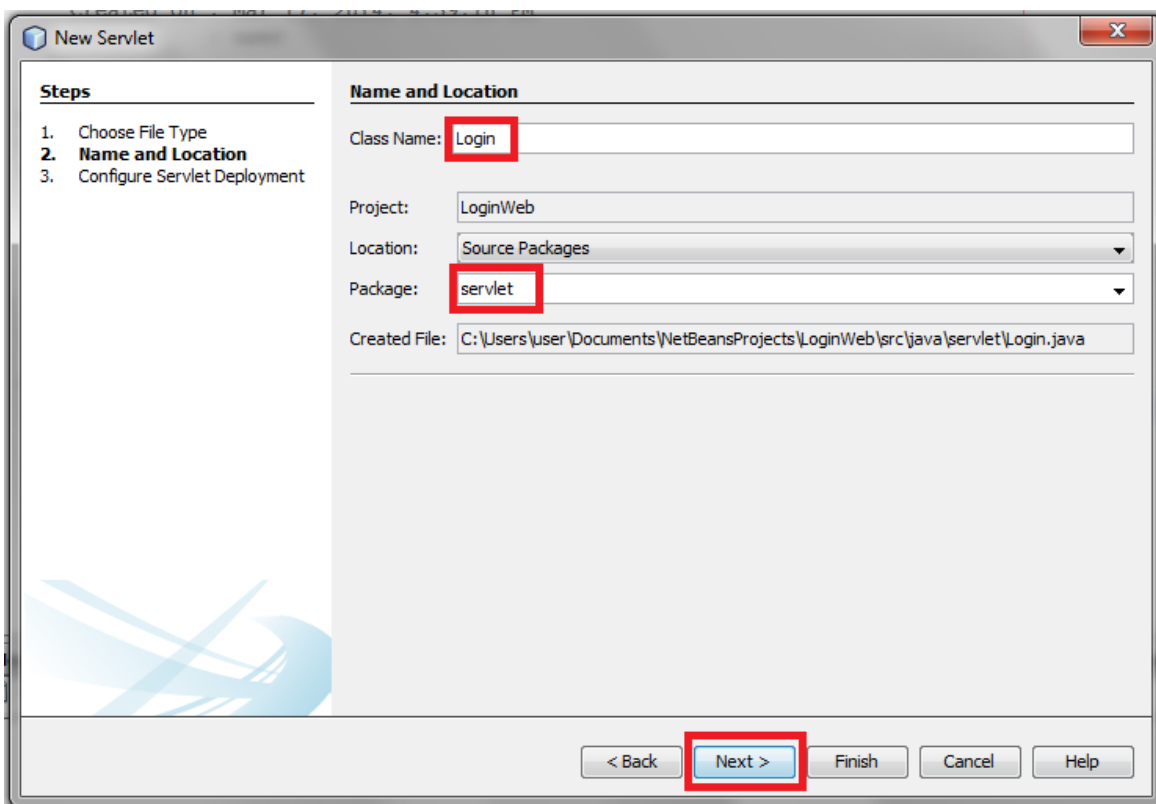
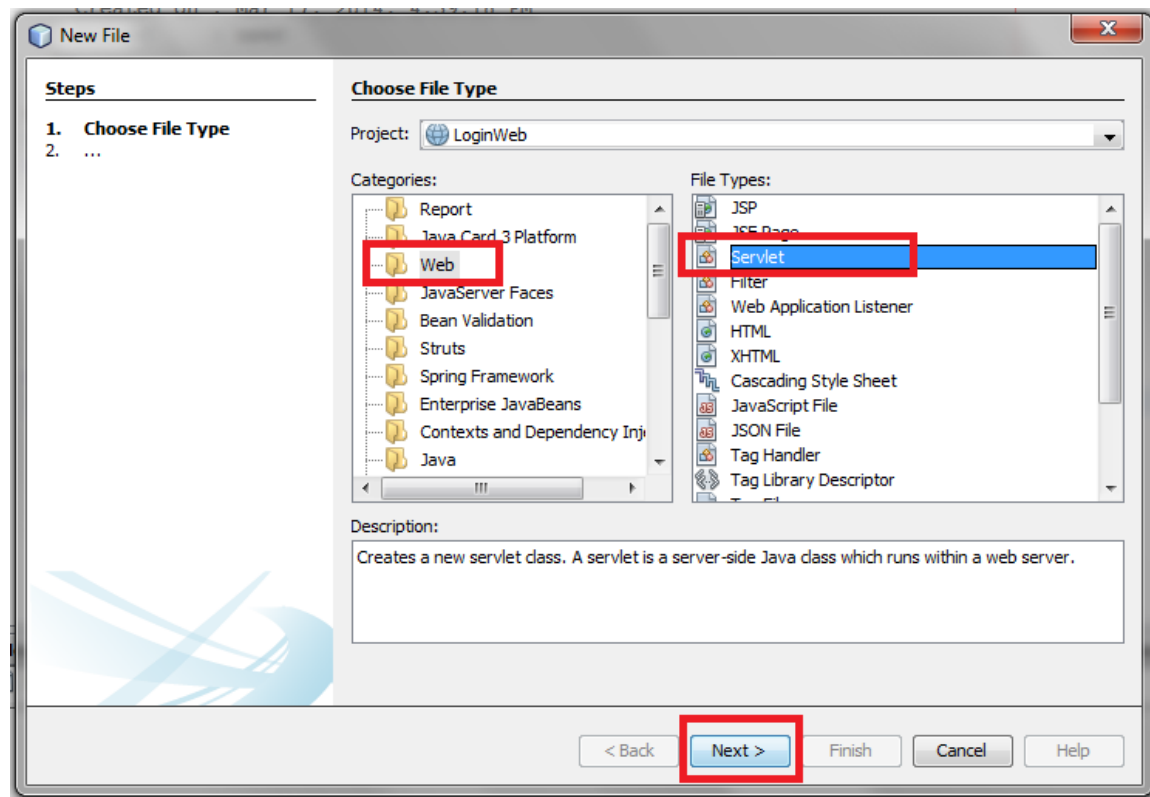
Para organizar a programação desse projeto utilizaremos o conceito de programação MVC (Model View Controller). Esse conceito envolve o uso de três estruturas:

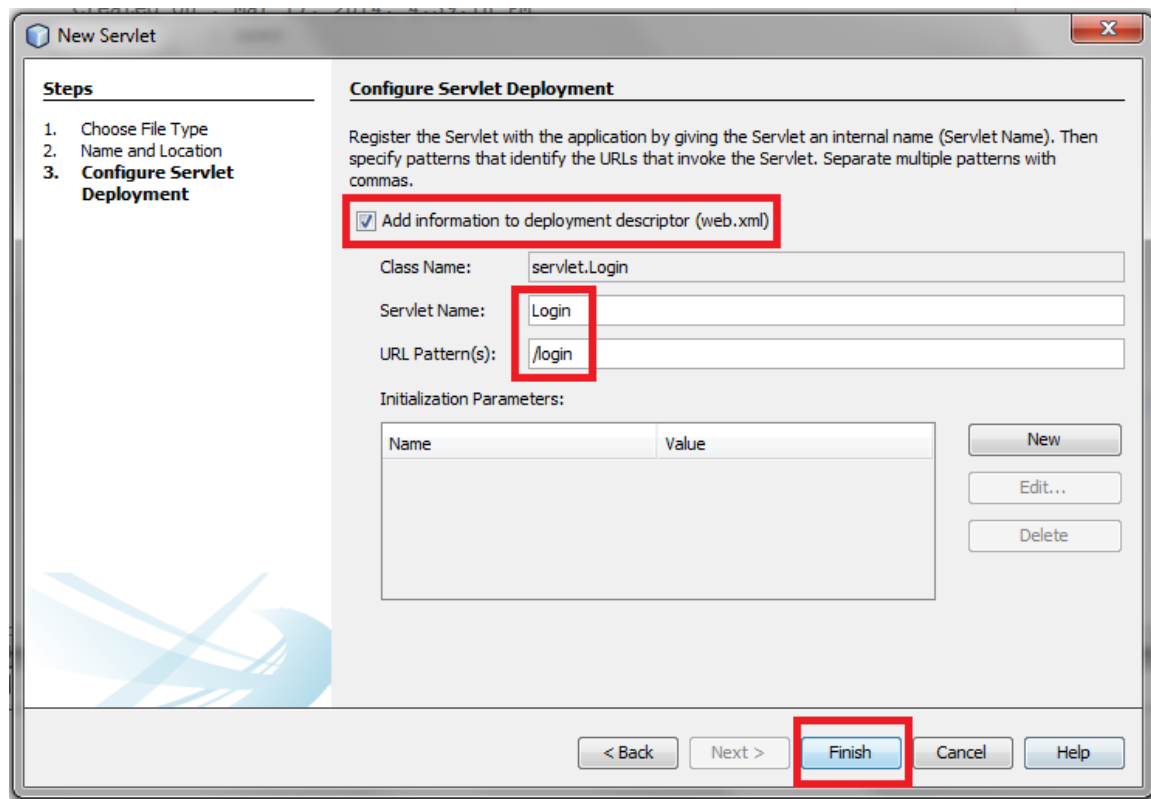
- **Model** – Representa o comportamento e modelo de dados do domínio da aplicação;
- **View** – Representa a interface de interação com o usuário;
- **Controller** – Representa o controle de fluxo e lógica da aplicação.



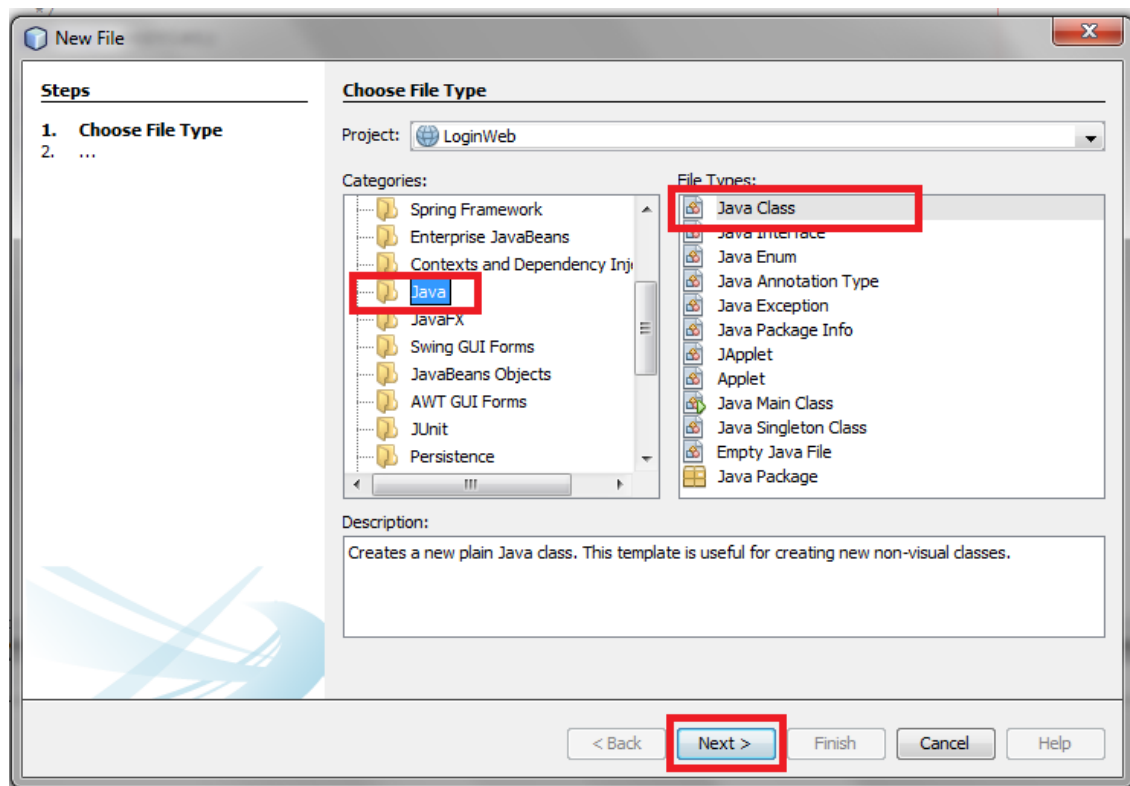
Criando as Classes Base do Projeto

Nessa etapa devem ser criadas três classes para serem utilizadas no projeto: classe Usuario, classe Controlador e classe Login. A classe Login será um *servlet*. Começaremos pela classe Login.





Criação das classes Usuario e Controle:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

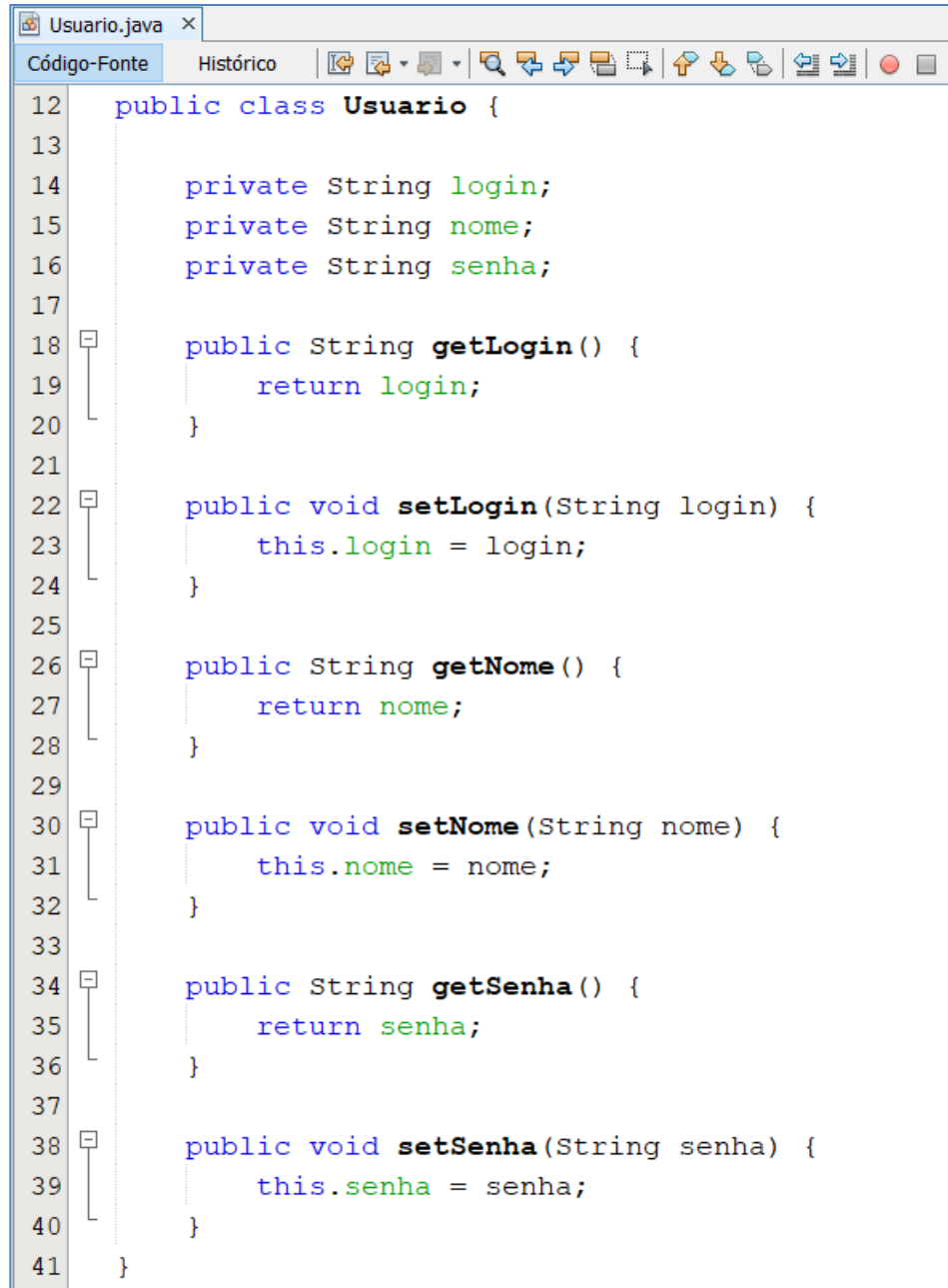
Location:

Package:

Created File:

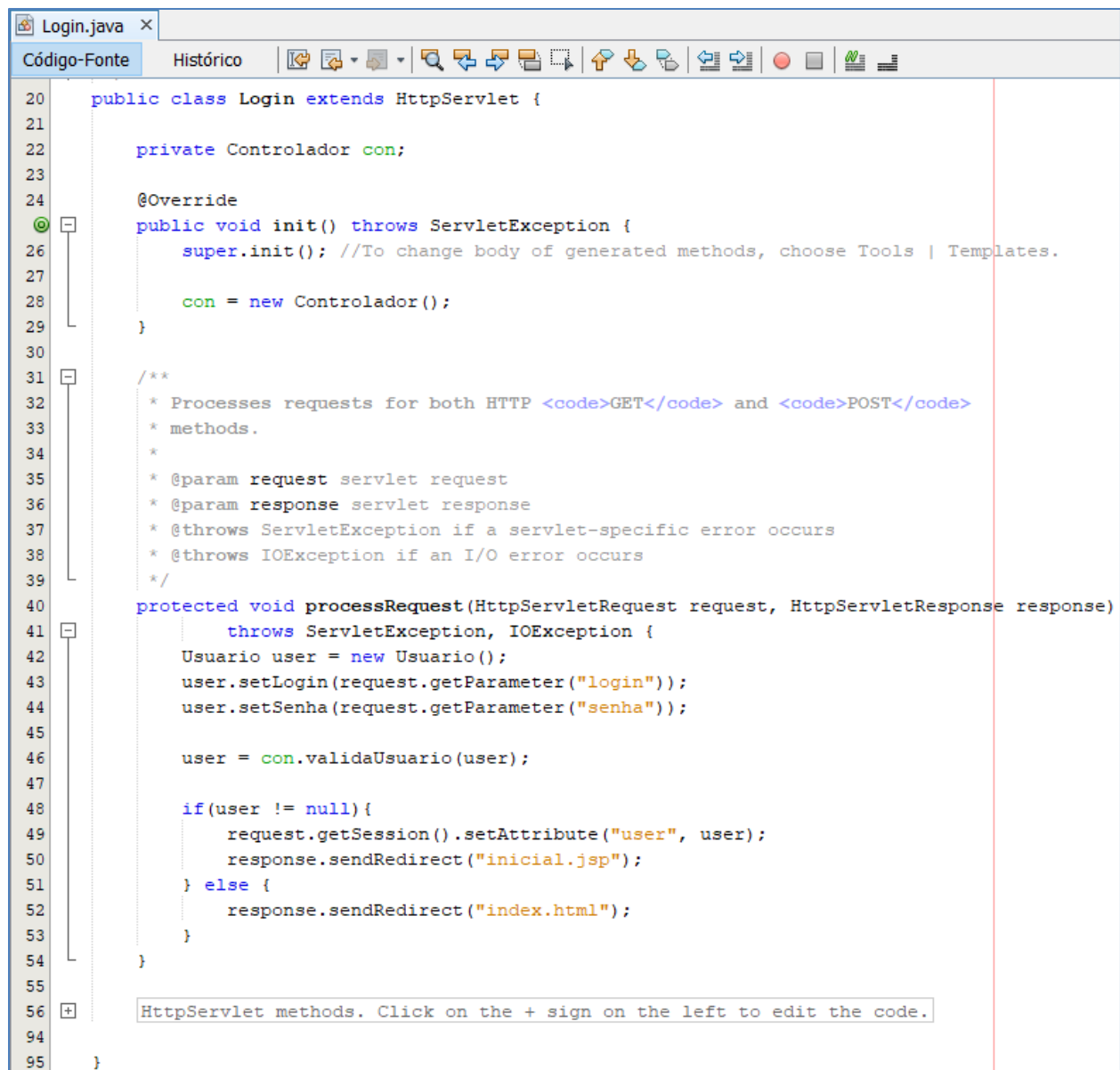
< Back Next > **Finish** Cancel Help

Codificação das classes:



```
12 public class Usuario {
13
14     private String login;
15     private String nome;
16     private String senha;
17
18     public String getLogin() {
19         return login;
20     }
21
22     public void setLogin(String login) {
23         this.login = login;
24     }
25
26     public String getNome() {
27         return nome;
28     }
29
30     public void setNome(String nome) {
31         this.nome = nome;
32     }
33
34     public String getSenha() {
35         return senha;
36     }
37
38     public void setSenha(String senha) {
39         this.senha = senha;
40     }
41 }
```

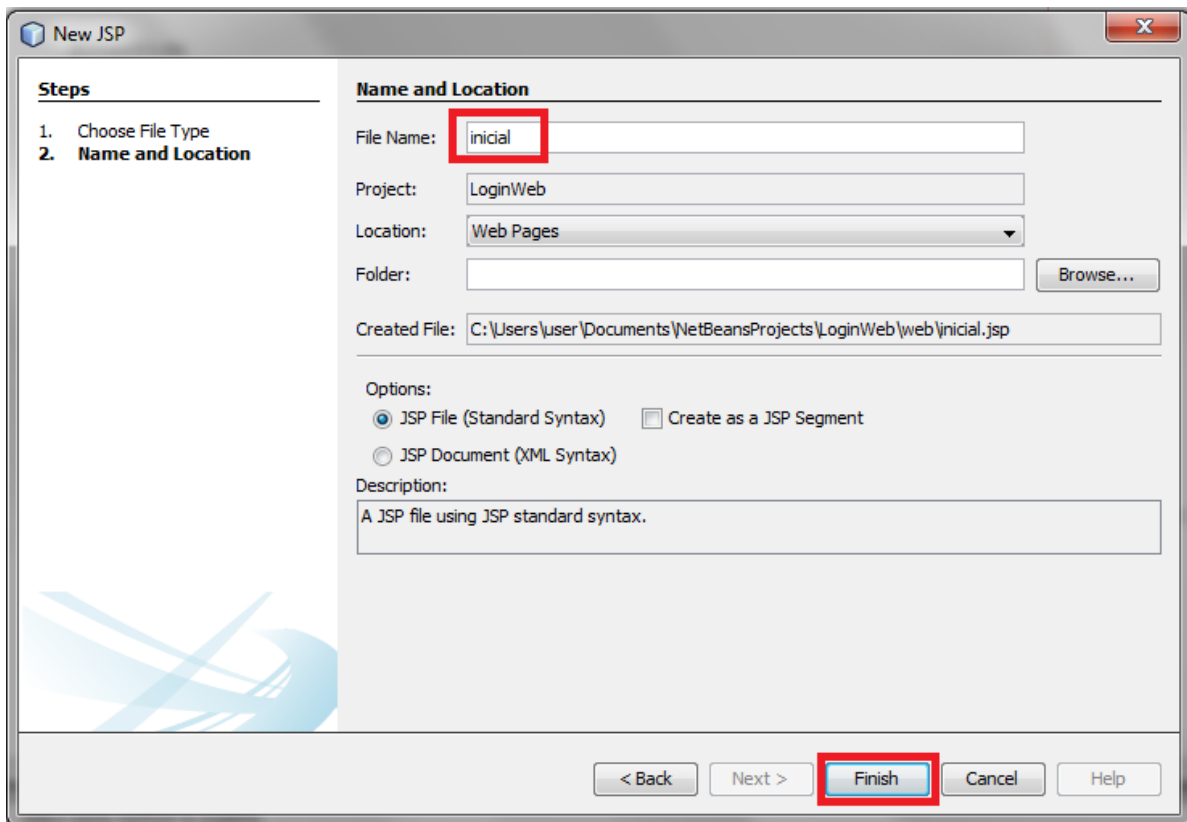
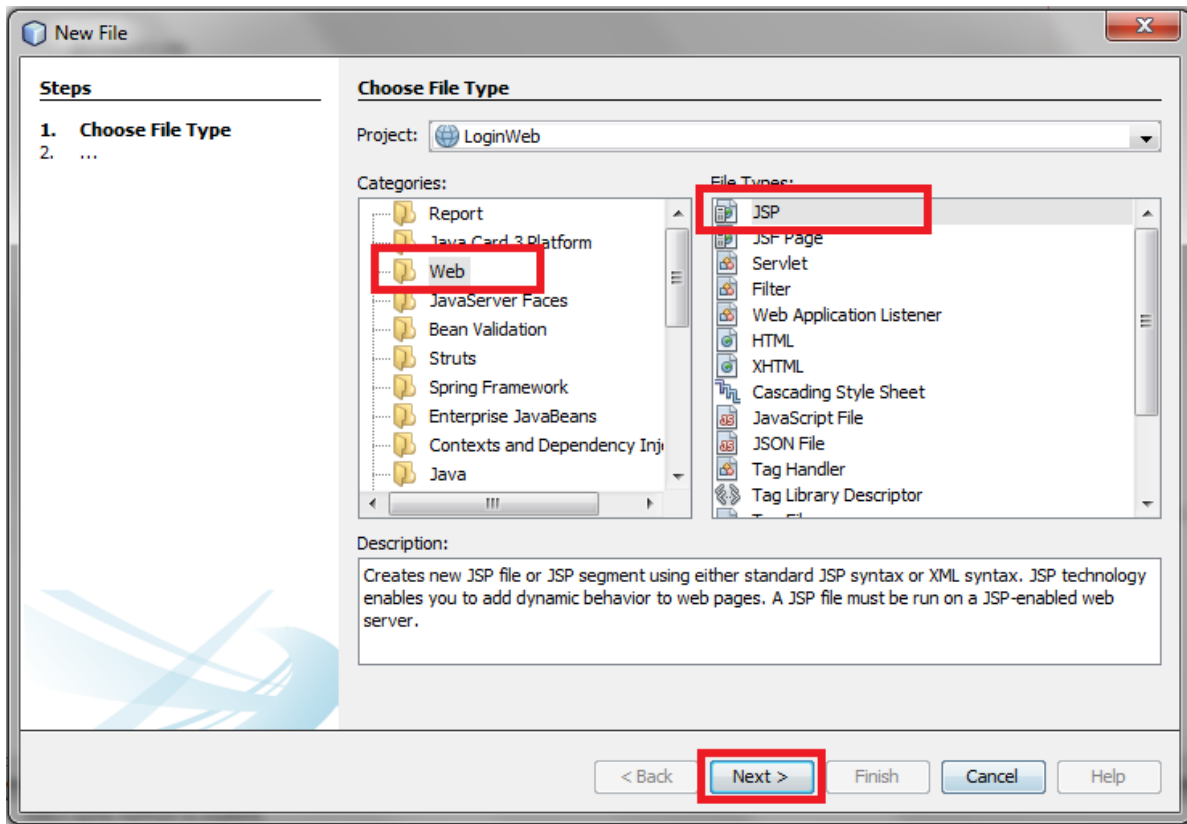
```
Controlador.java x
Código-Fonte  Histórico
6  package controle;
7
8  import entidades.Usuario;
9
10 /**
11  *
12  * @author Cesar
13  */
14 public class Controlador {
15
16
17     public Usuario validaUsuario(Usuario user) {
18
19         //Consulta no Banco de Dados
20         Usuario novo = new Usuario();
21         novo.setLogin("teste");
22         novo.setSenha("teste");
23         novo.setNome("Nome do Usuário ");
24
25         if(user.getLogin().equals(novo.getLogin()) &&
26             user.getSenha().equals(novo.getSenha())) {
27             return novo;
28         }
29         return null;
30     }
31 }
```



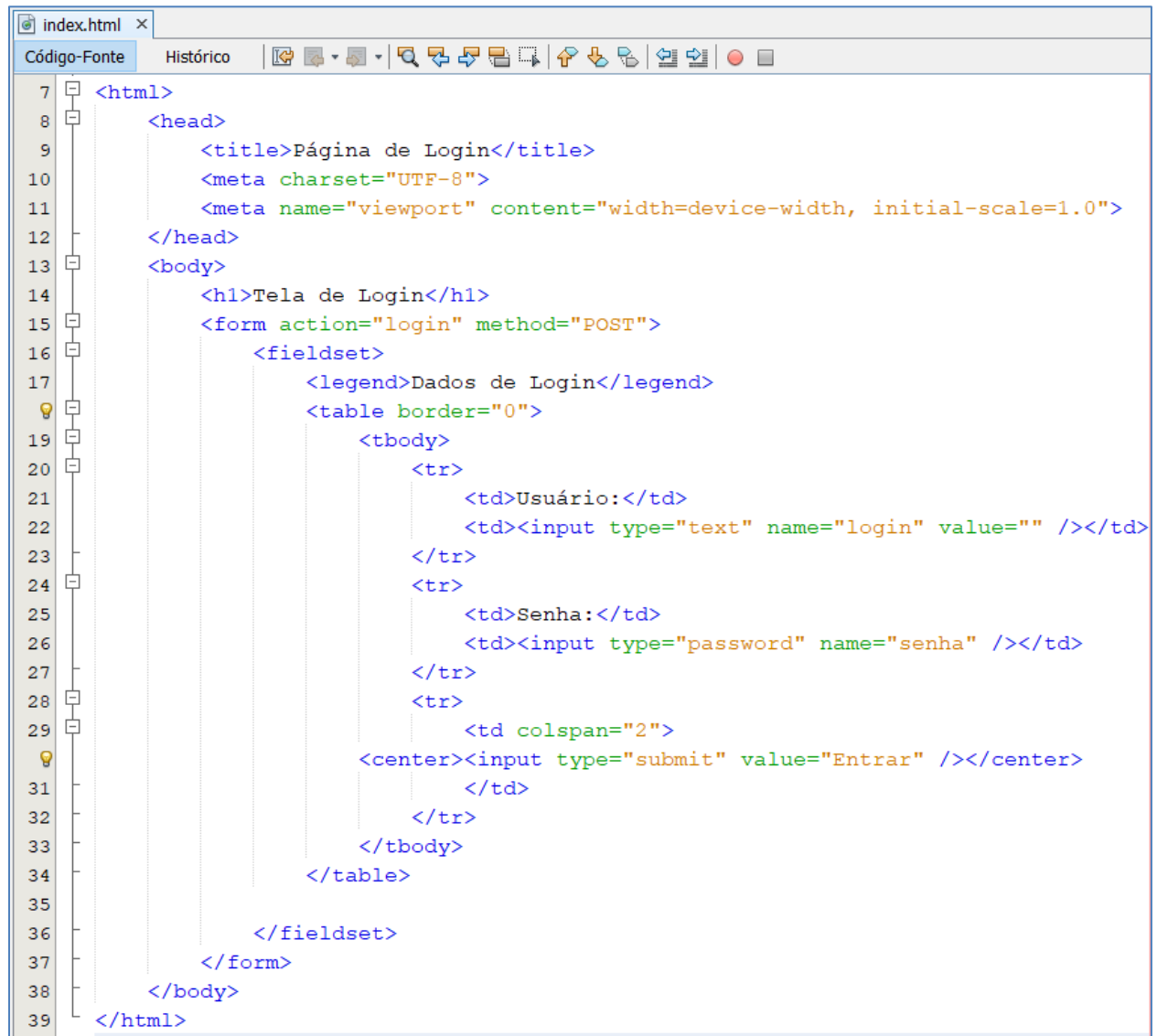
```
20 public class Login extends HttpServlet {
21
22     private Controlador con;
23
24     @Override
25     public void init() throws ServletException {
26         super.init(); //To change body of generated methods, choose Tools | Templates.
27
28         con = new Controlador();
29     }
30
31     /**
32      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
33      * methods.
34      *
35      * @param request servlet request
36      * @param response servlet response
37      * @throws ServletException if a servlet-specific error occurs
38      * @throws IOException if an I/O error occurs
39      */
40     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
41         throws ServletException, IOException {
42         Usuario user = new Usuario();
43         user.setLogin(request.getParameter("login"));
44         user.setSenha(request.getParameter("senha"));
45
46         user = con.validaUsuario(user);
47
48         if(user != null){
49             request.getSession().setAttribute("user", user);
50             response.sendRedirect("inicial.jsp");
51         } else {
52             response.sendRedirect("index.html");
53         }
54     }
55
56     HttpServlet methods. Click on the + sign on the left to edit the code.
94
95 }
```

Criando as Páginas de Visualização

Nessa etapa devem ser criadas duas páginas JSP de visualização. A página “index.html” criada ao gerar o projeto e a página “inicial.jsp”. A página “index.html” terá o formulário de login e senha para acessar a página “inicial.jsp”, simulando um processo de autenticação de usuário em um sistema.



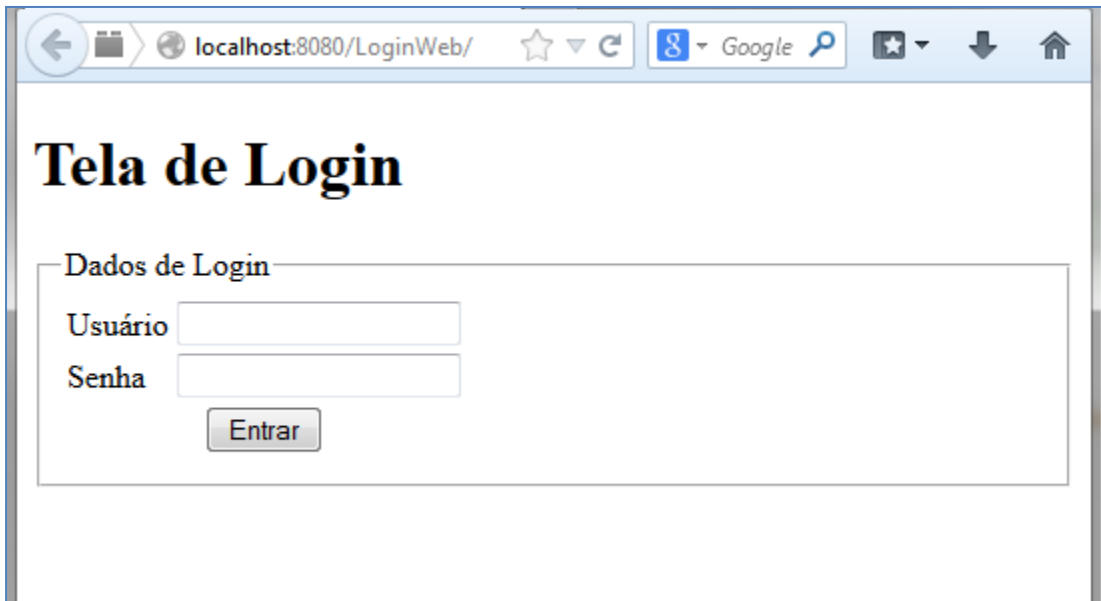
Codificação das páginas:



The image shows a code editor window with a tab labeled 'index.html'. The editor has a menu bar with 'Código-Fonte' and 'Histórico', and a toolbar with various icons. The code is written in HTML and is color-coded. The code defines a login page with a title, meta tags, and a form containing a legend, a table with input fields for 'login' and 'senha', and a submit button labeled 'Entrar'.

```
7 <html>
8   <head>
9     <title>Página de Login</title>
10    <meta charset="UTF-8">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12  </head>
13  <body>
14    <h1>Tela de Login</h1>
15    <form action="login" method="POST">
16      <fieldset>
17        <legend>Dados de Login</legend>
18        <table border="0">
19          <tbody>
20            <tr>
21              <td>Usuário:</td>
22              <td><input type="text" name="login" value="" /></td>
23            </tr>
24            <tr>
25              <td>Senha:</td>
26              <td><input type="password" name="senha" /></td>
27            </tr>
28            <tr>
29              <td colspan="2">
30                <center><input type="submit" value="Entrar" /></center>
31              </td>
32            </tr>
33          </tbody>
34        </table>
35      </fieldset>
36    </form>
37  </body>
38 </html>
```


Se tudo estiver correto, basta executar o programa.



A screenshot of a web browser window. The address bar shows 'localhost:8080/LoginWeb/'. The page title is 'Tela de Login'. Below the title, there is a form titled 'Dados de Login'. Inside the form, there are two input fields: 'Usuário' and 'Senha'. Below these fields is a button labeled 'Entrar'.

Tela de Login

Dados de Login

Usuário

Senha

Testando com o usuário e senha “teste” e “teste”.



A screenshot of a web browser window. The address bar shows 'localhost:8080/LoginWeb/inicial.jsp'. The page title is 'Usuario Logado!!!'. Below the title, there is a table with two columns: 'Nome' and 'Login'. The table contains one row with the values 'Nome do usuário' and 'teste'.

Usuario Logado!!!

Nome	Login
Nome do usuário	teste

Bom trabalho!!!