

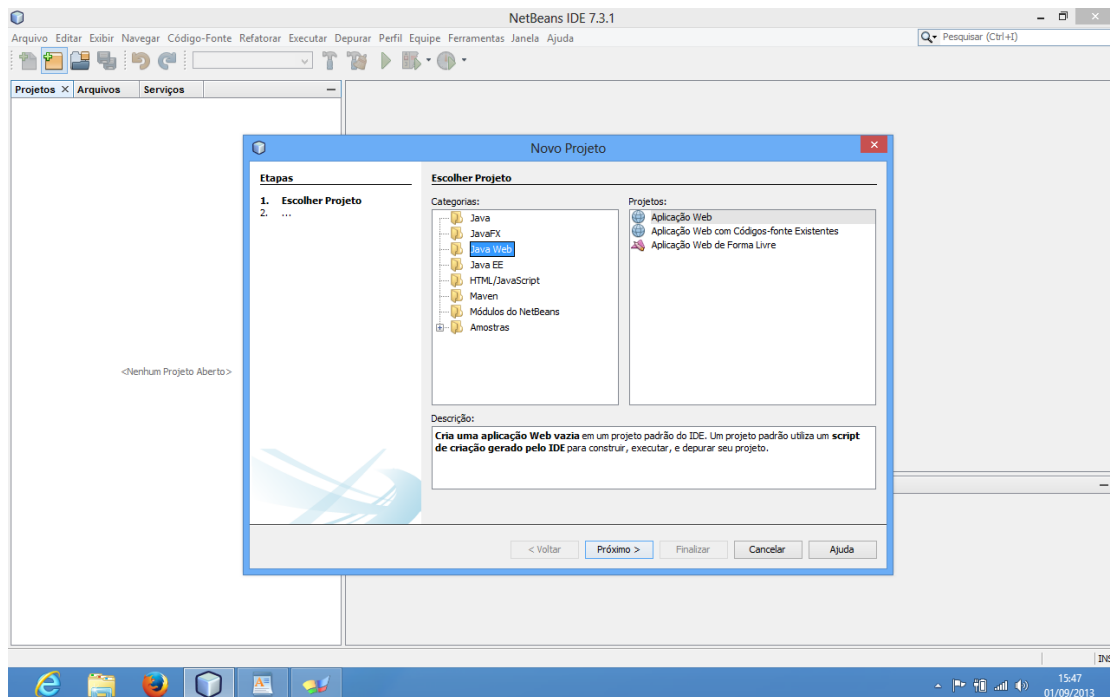
Criação de Novo Projeto Web.

Profa Mestre Sirley A Vitorio Oliveira

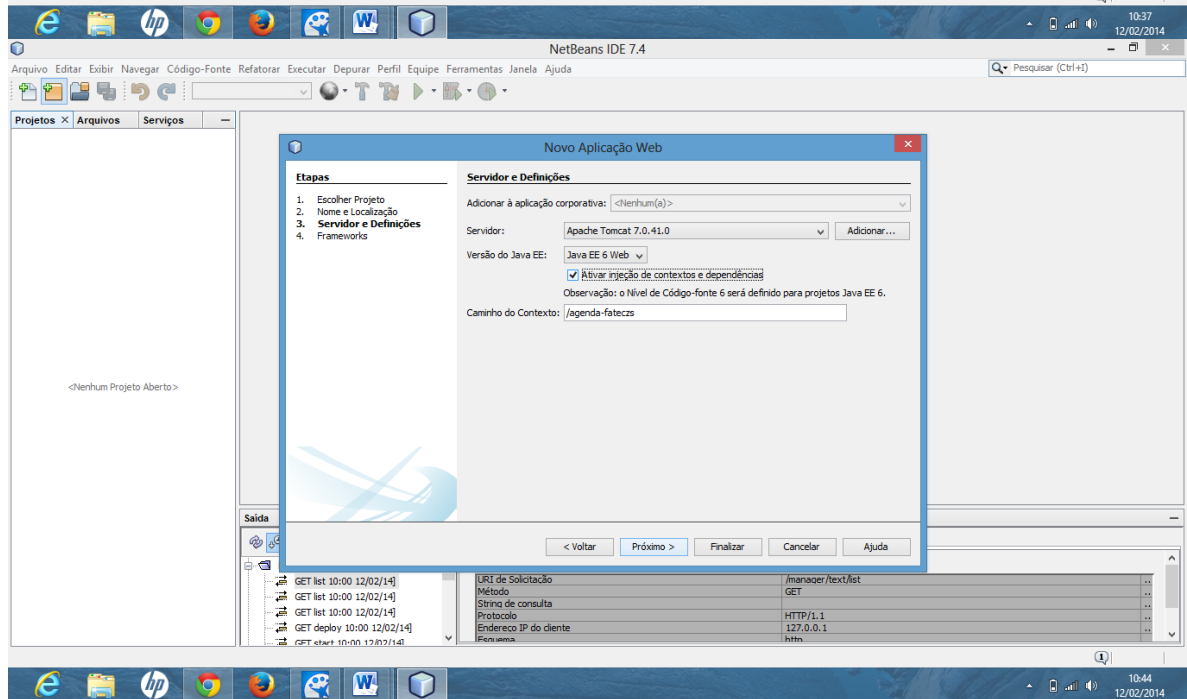
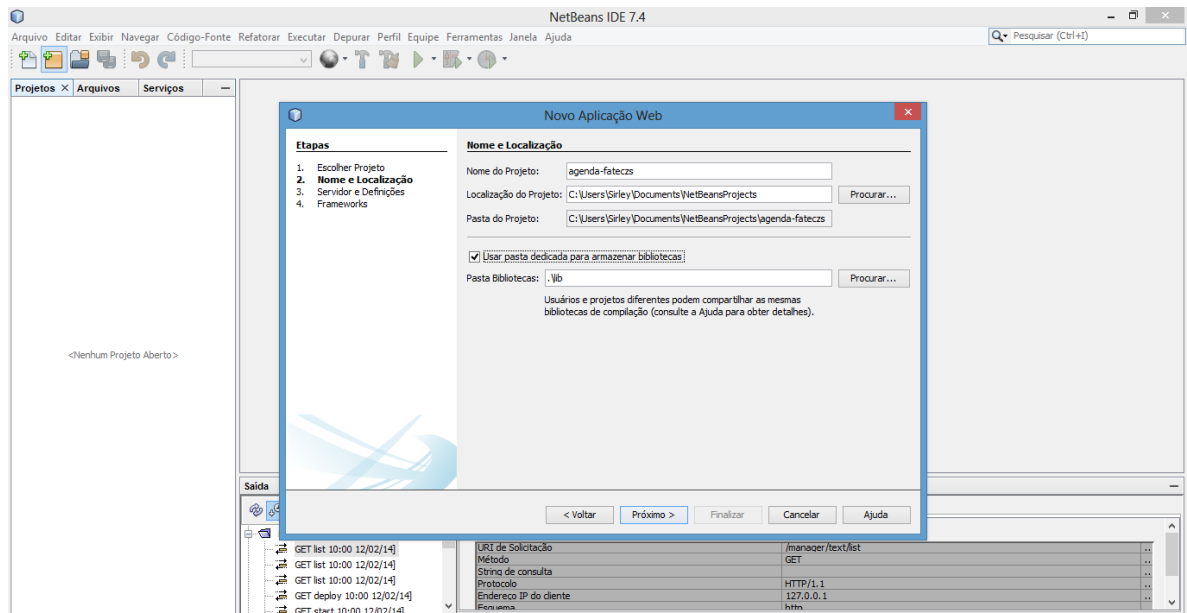
Objetivos.

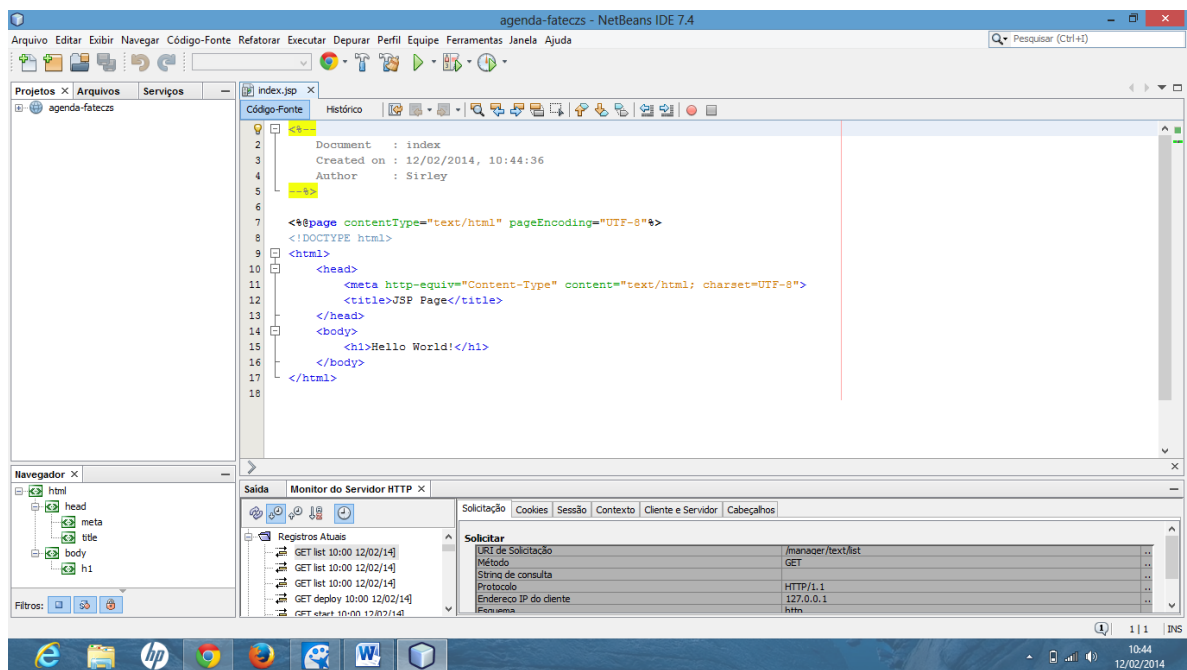
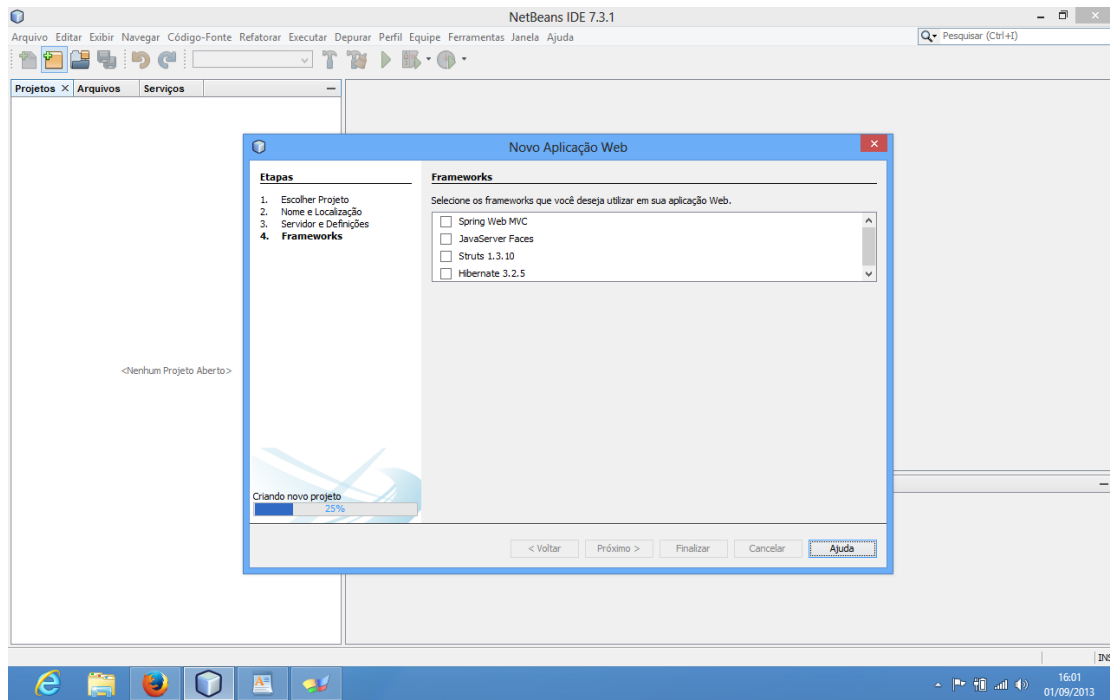
1. Criar um novo projeto Web usando Java.
2. Compreender quais são os diretórios importantes de uma aplicação Web.
3. Compreender quais são os arquivos importantes de uma aplicação Web.
4. Aprender onde colocar suas páginas e arquivos estáticos.

Criando um novo projeto.

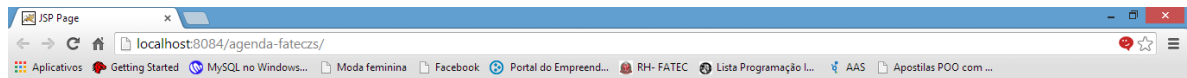


Nome do Projeto: agenda-fateczs





Executar Arquivo.



Hello World!



Voltar para o Netbeans.

Como estamos usando o servidor de aplicação TomCat, visualizamos o padrão JAKARTA no netbeans.

A Apache Jakarta fornece diretrizes de como estruturar suas aplicações Web para assegurar que elas trabalhem corretamente com o servidor Tomcat. Quando você cria um projeto na IDE e seleciona o Tomcat, esta estrutura é respeitada.

Se utilizarmos o GlassFish, será adotado a estrutura que utiliza o padrão **Java BluePrints**.

Visão Geral.

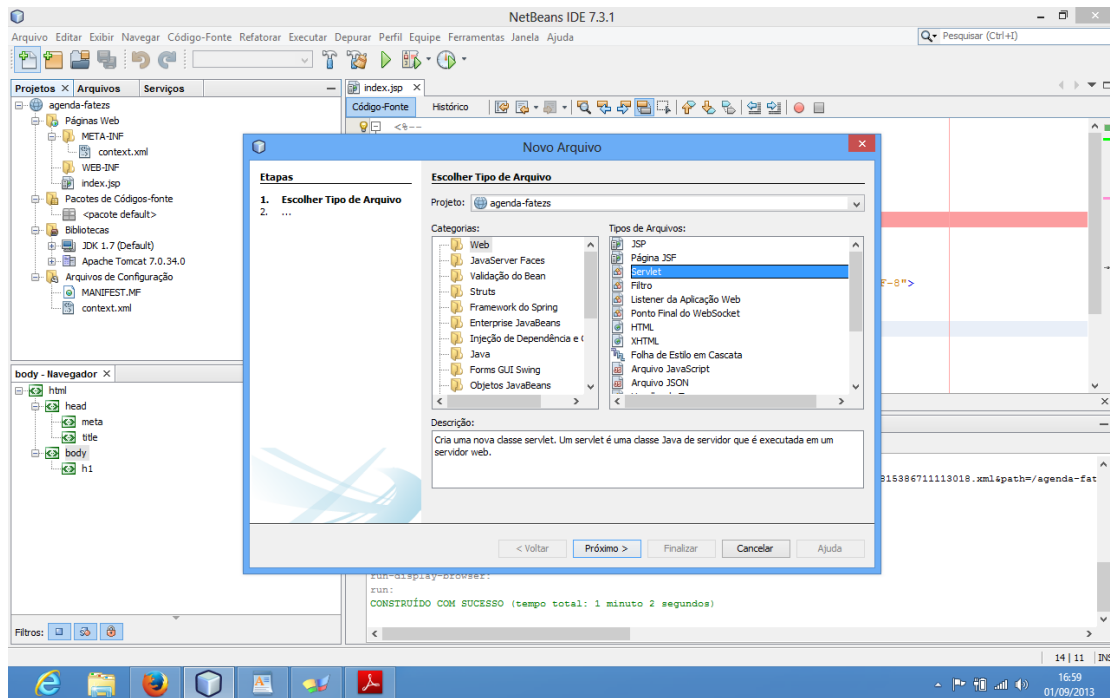
- Diretório WEB-INF : páginas web. (Janela arquivos: diretórioweb).
- Diretório Código Fonte (ou Source Packages): arquivos java não compilados, servlets, outros. (Janela arquivos: src).
- Diretório Arquivos de Configuração: arquivos para armazenamento das configurações da nossa aplicação. (Janela Arquivos: web- WEB-INF).

Desenvolvendo Servlets.

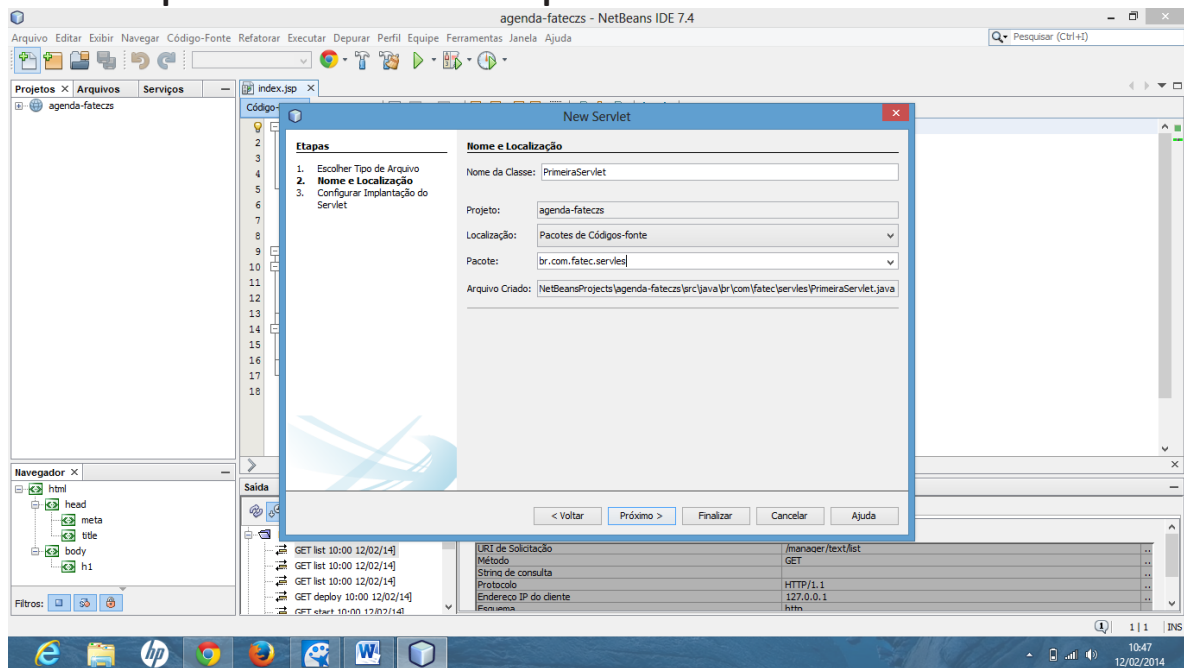
o NetBeans criou na conclusão do projeto, um arquivo JSP (JavaServer Pages). A Servlet recebe informações desse arquivo. Servlet é uma tecnologia capaz de criar páginas dinâmicas.

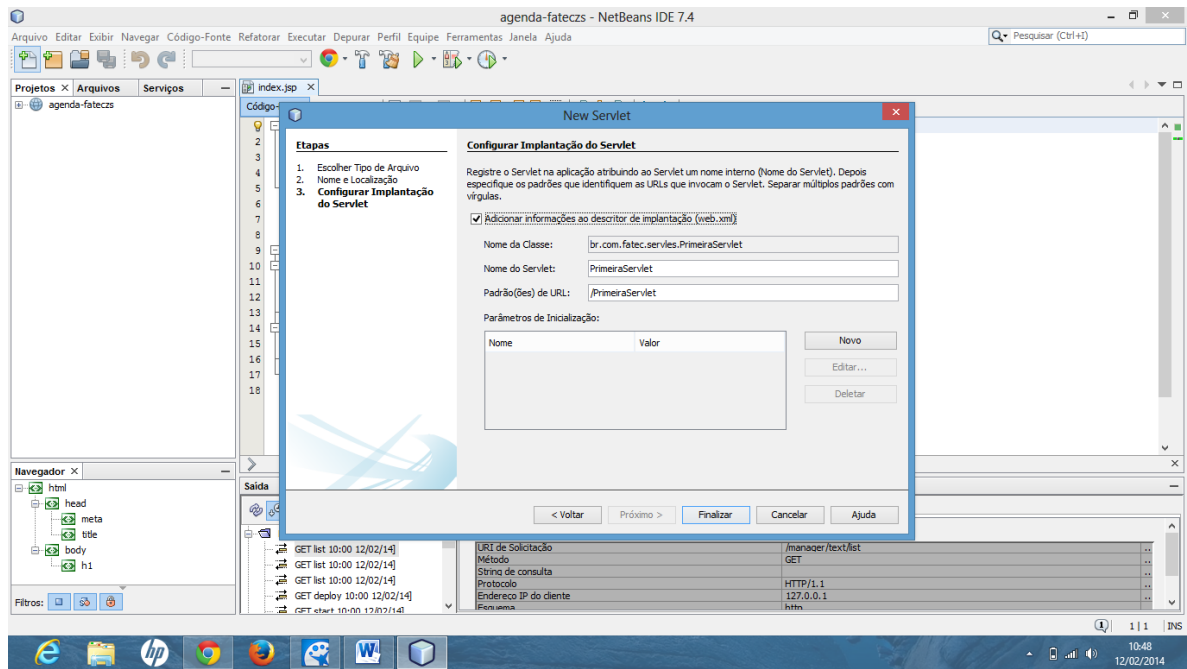
Vamos criar nossa primeira Servlet.

Arquivo- Novo arquivo.

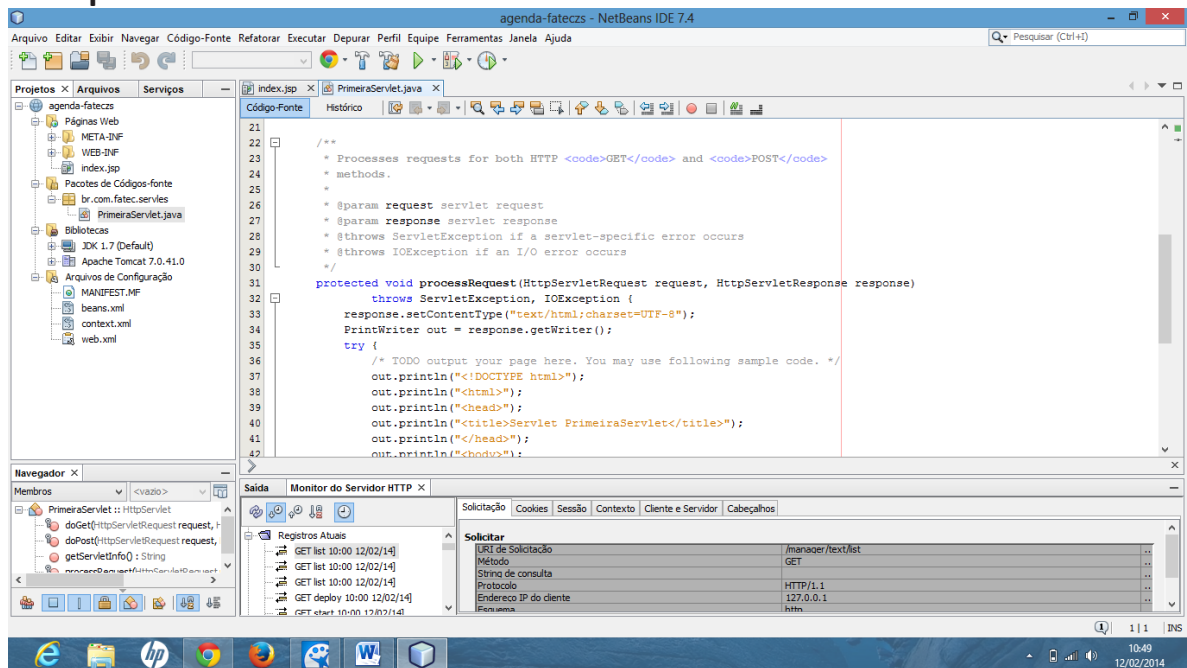


Criar o arquivo PrimeiraServlet no pacote br.com.fatec.servlets

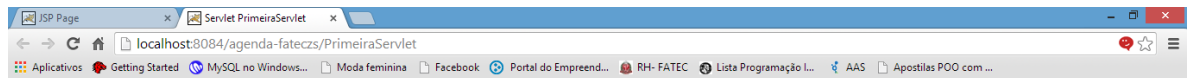




O Arquivo web.xml foi criado.



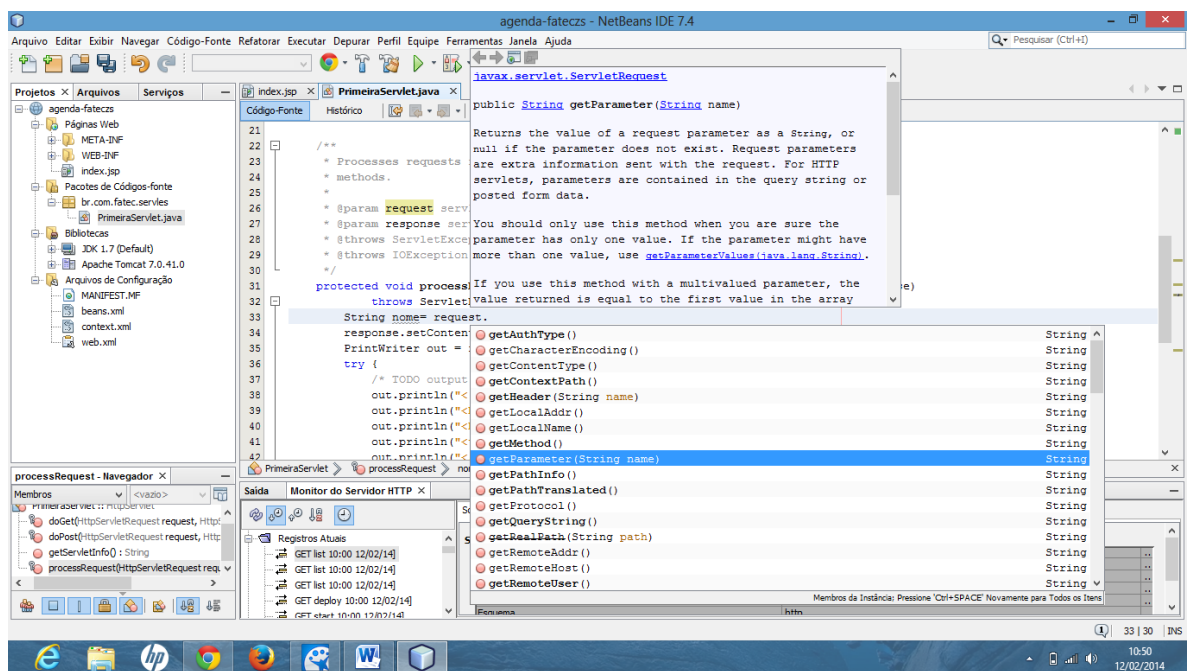
Executar Arquivo.



Servlet PrimeiraServlet at /agenda-fateczs



Execute as modificações abaixo para imprimir uma frase:



```
protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
```

```
    throws ServletException, IOException {
```

```
        String nome= request.getParameter("nome");
```

```
        response.setContentType("text/html;charset=UTF-8");
```

```
        PrintWriter out = response.getWriter();
```

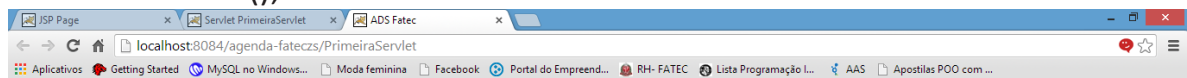
```
        try {
```

```
            /* TODO output your page here. You may use following sample
```

```

code. */
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<title>ADS Fatec </title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Servlet PrimeiraServlet at " +
request.getContextPath() + "</h1>");
    out.println("<h1>Autor: " + nome + "</h1>");
    out.println("</body>");
    out.println("</html>");
} finally {
    out.close();
}

```



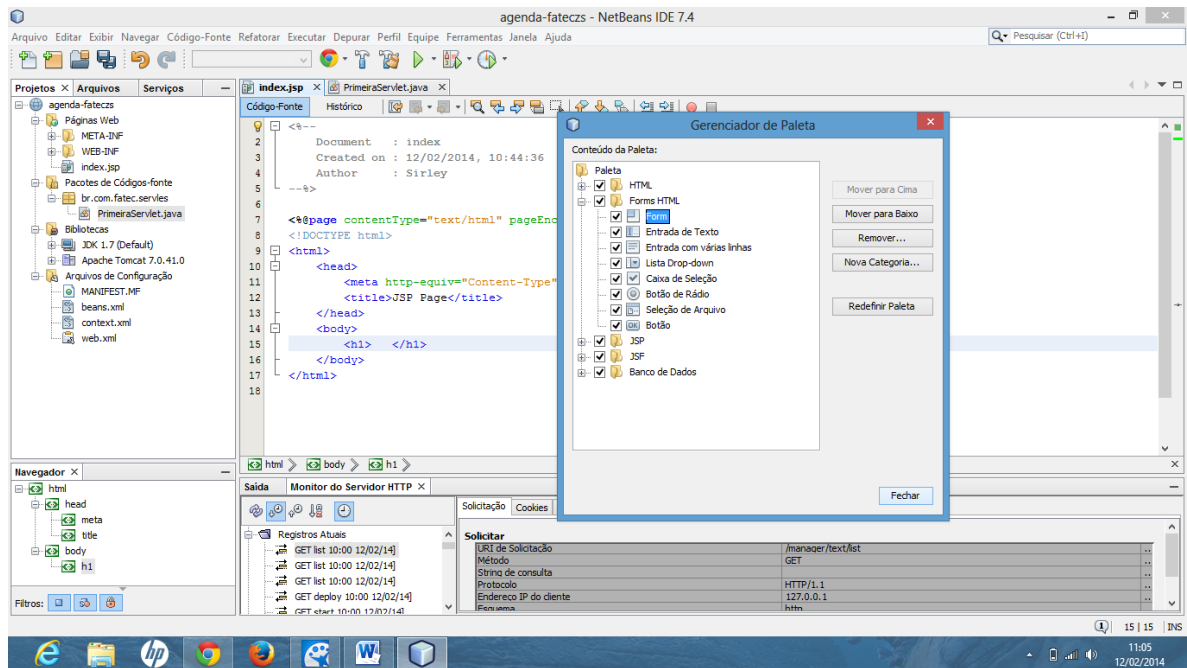
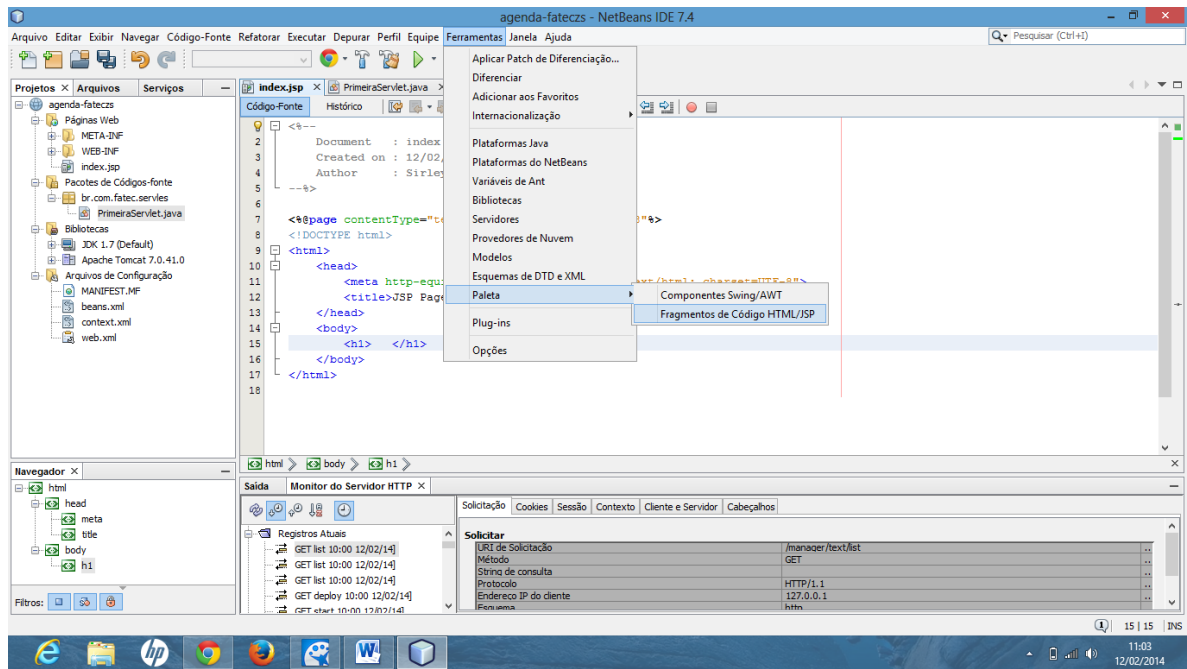
Servlet PrimeiraServlet at /agenda-fateczs

Autor: null

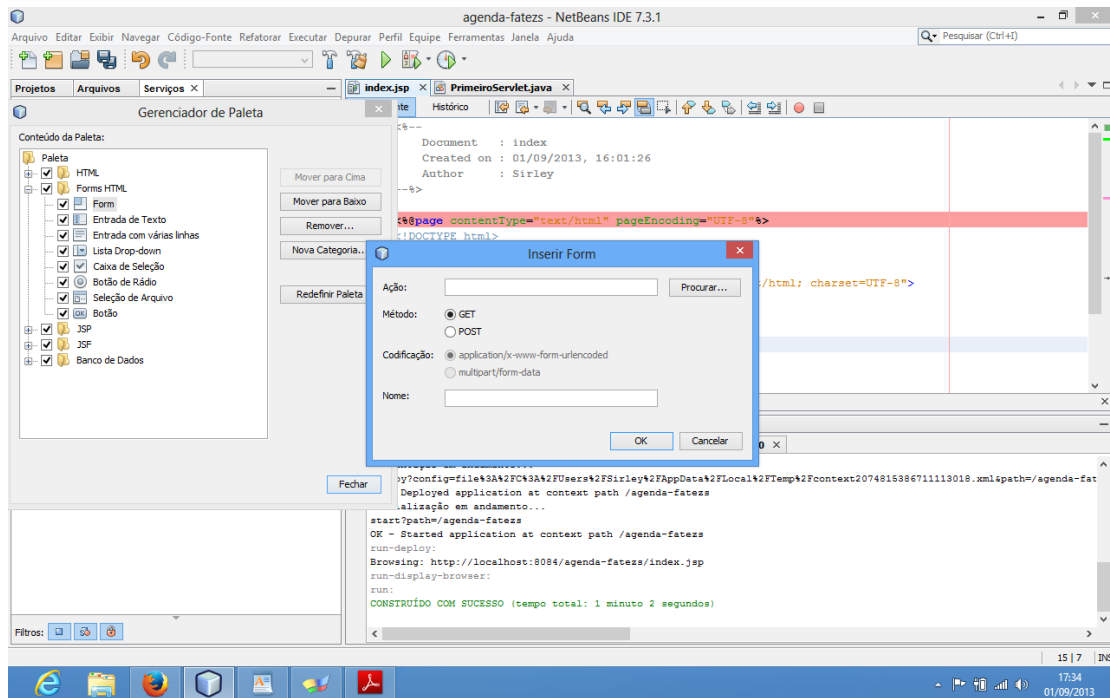


Vamos adicionar um formulário no arquivo index.jsp para receber o nome do usuário.

1. No arquivo index, apagar o conteúdo de <body> e deixar o cursor posicionado entre <body> <body>
2. No menu: ferramentas, paleta.



Arrastar o ícone Form até o arquivo index.

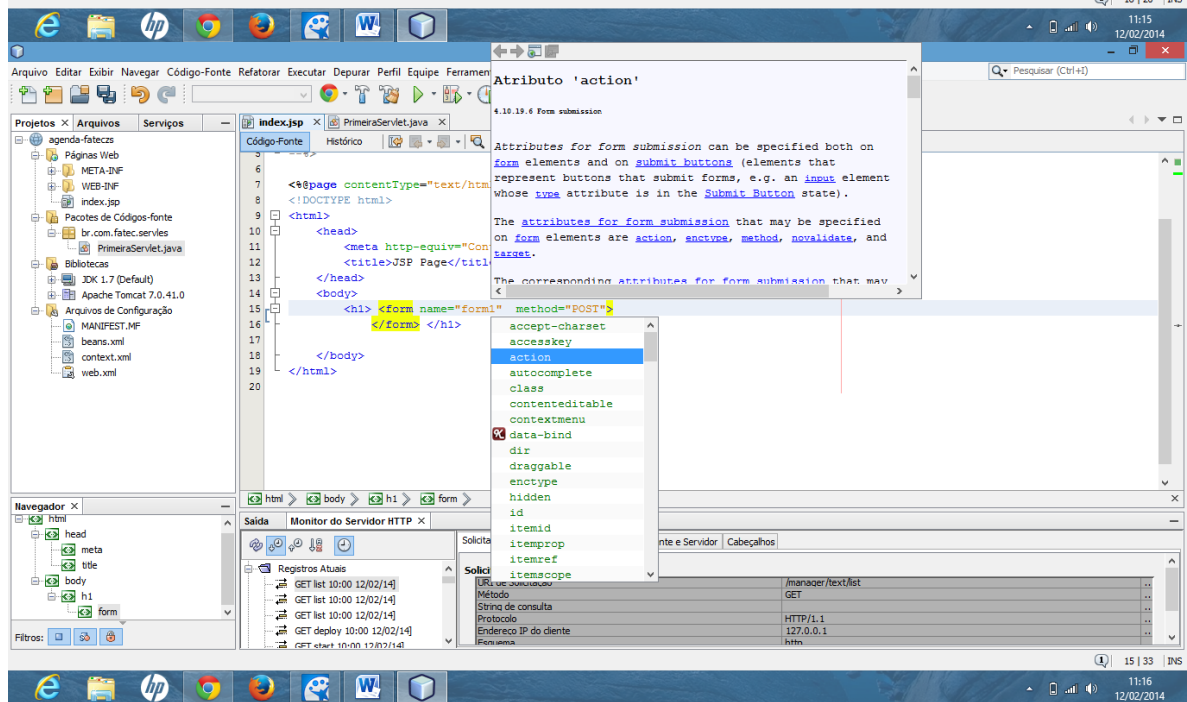
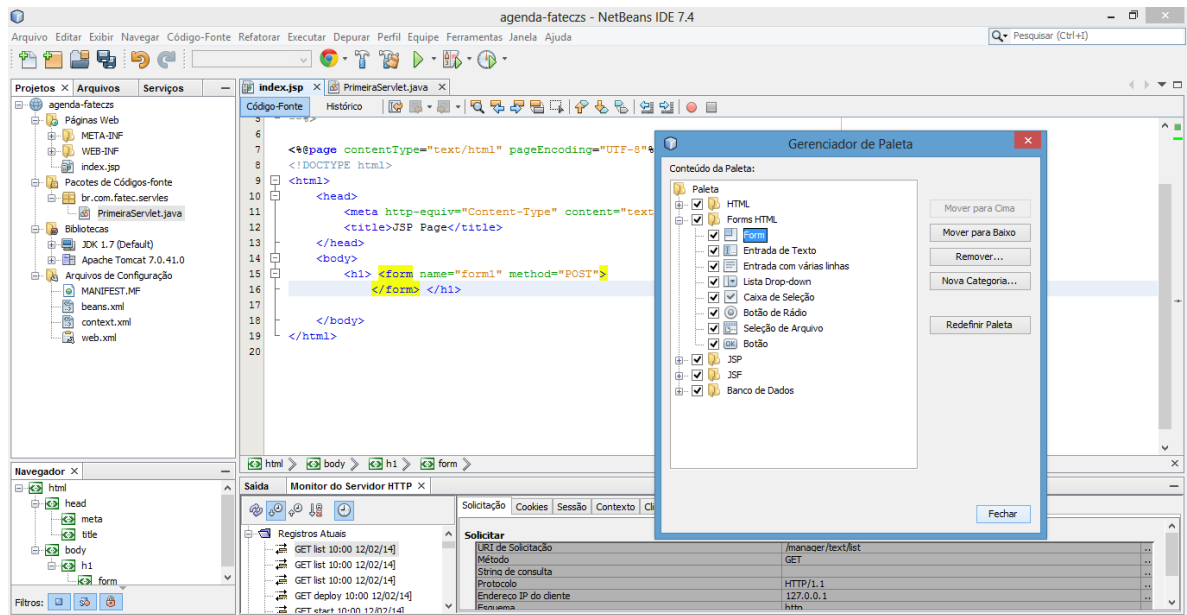


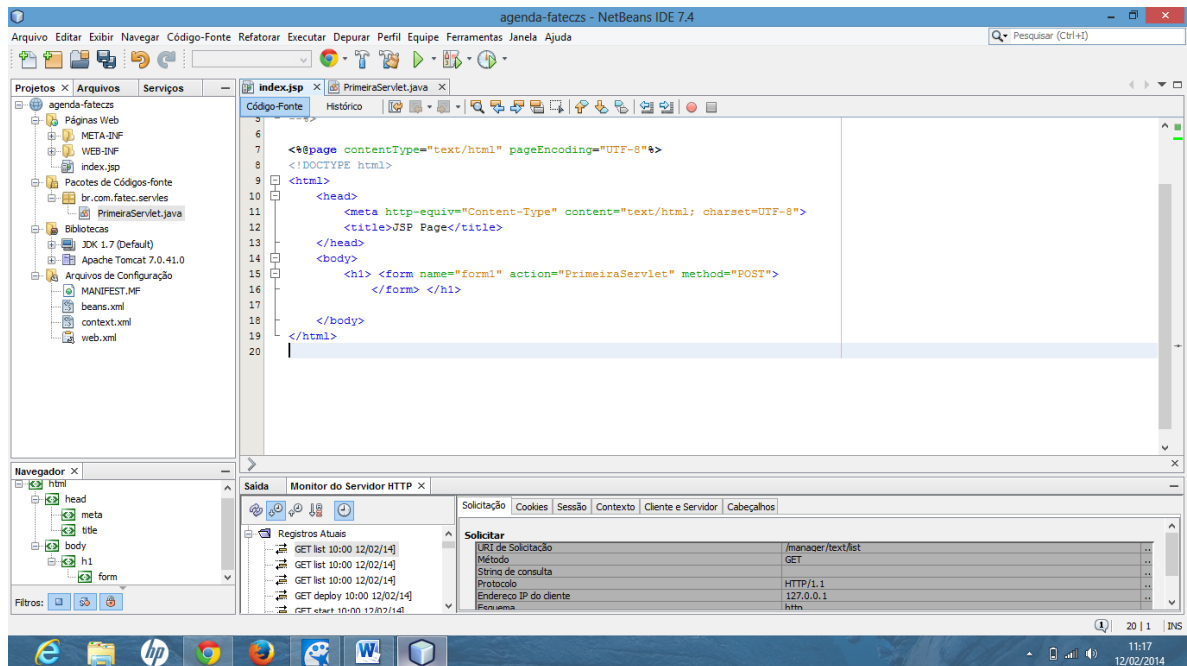
Resultado final no arquivo index:

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
  <title>JSP Page</title>
</head>
<body>
  <form name="Formulario1" action="PrimeiroServlet"
method="POST">
    </form>
  </body>
</html>
```

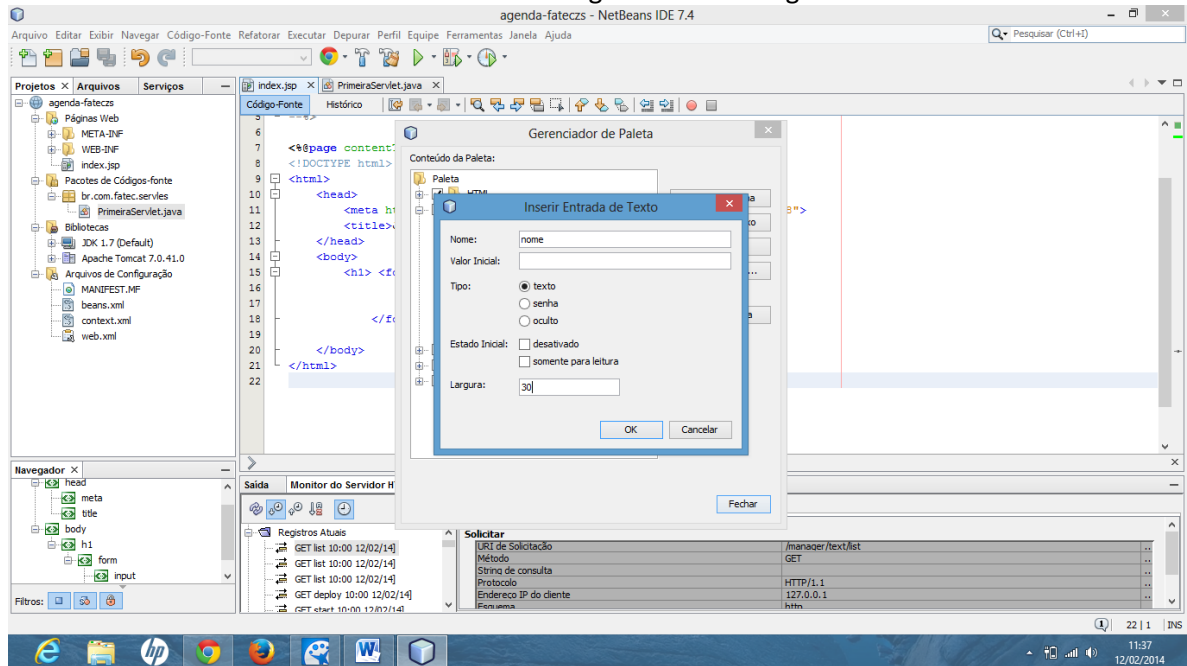
Agora vamos inserir os campos do formulário.

No arquivo index, deixar o cursor entre as tags **<form> ... </form>**.

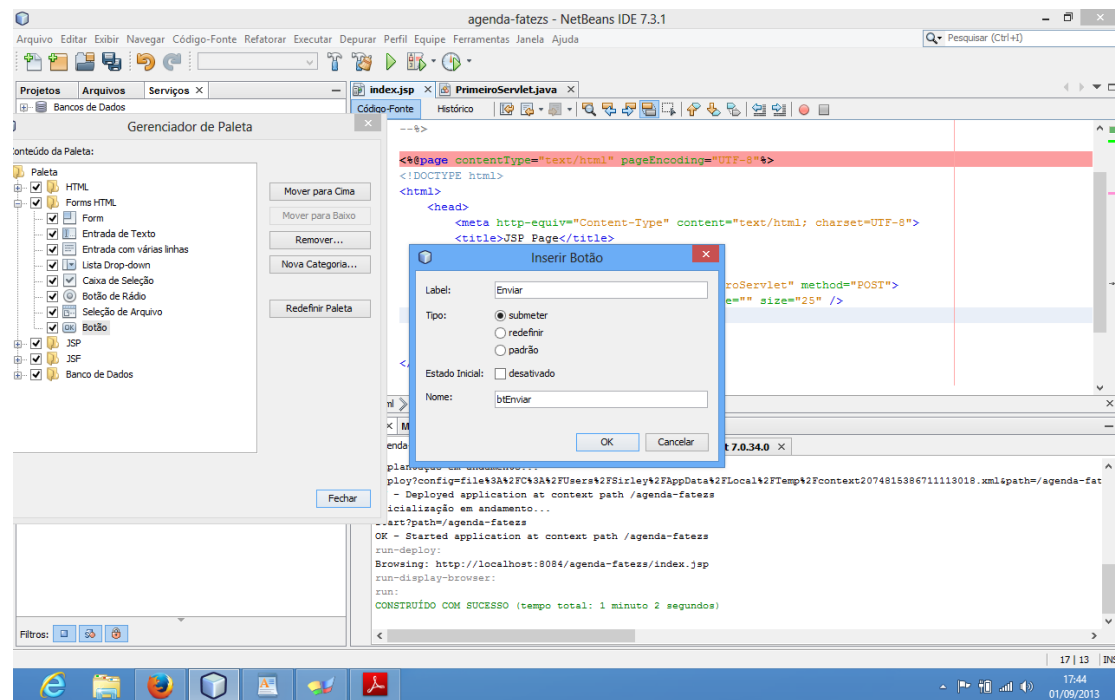




Inserir Entrada de texto: ferramentas- Paleta- Fragmentos de código HTML



Acrescentar Botão de envio no formulário.



Código:

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
```

```
</head>
```

```
<body>
```

```
<h1> <form name="form1" action="PrimeiraServlet" method="POST">
```

```
<input type="text" name="nome" value="" size="30" />
```

```
<input type="submit" value="Enviar" name="btEnviar" />
```

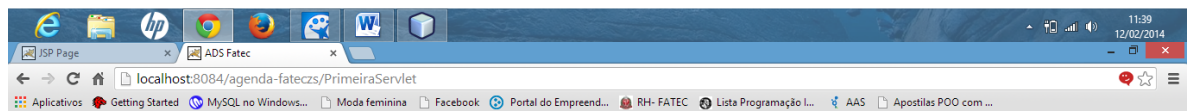
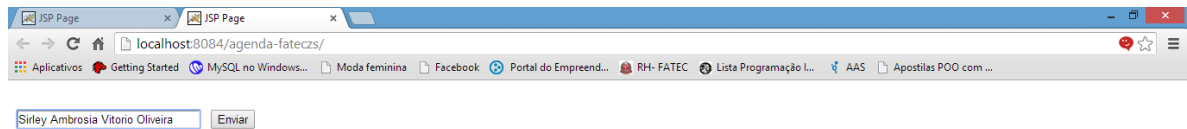
```
</form> </h1>
```

```
</body>
```

```
</html>
```

Executar projeto.

Resultado final.



Servlet PrimeiraServlet at /agenda-fateczs

Autor: Sirley Ambrosia Vitorio Oliveira



Agora você já sabe como criar um projeto Web e uma Servlet.

Observe que o Servlet possui a extensão **.java**. Portanto, Servlets são classes Java, desenvolvidas de acordo com uma estrutura bem definida e que, quando instaladas junto a um Servidor que implemente um Servidor de Aplicações ou Servlet Container (um servidor que permita a execução de Servlets), podem tratar requisições recebidas de clientes. Observe dois métodos importantes: **doGet()** e **doPost()**. Ambos chamam **processRequest()**, onde você alterou o código para receber os dados vindos do formulário. Estes métodos possuem a habilidade de resgatar informações enviadas pelo usuário tanto pelo método **GET**, como pelo método **POST**. Na geração do formulário **HTML**, você criou a tag **<form/>** selecionando em **METHOD** o item **POST**, o que, portanto, é resgatado por este Servlet. Embora existam outros protocolos de envio, **POST** e **GET** são os mais usados.

a classe Servlet criada, notará que ela estende a classe **javax.servlet.http.HttpServlet**, uma classe abstrata que estende a **javax.servlet.GenericServlet**. A criação de Servlets exige as classes do pacote **javax.servlet** e **javax.servlet.http**, que pertencem a API Servlet do Java, que faz parte do Java EE. Perceba então que há dois **imports** de pacotes nesta classe.

A saída de informações, para serem impressas no **HTML** de retorno ao usuário,

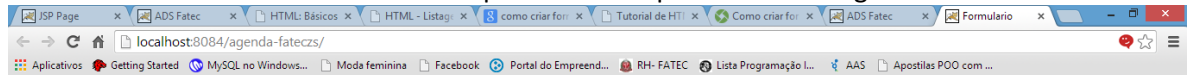
é feito pelo método **println**, de **java.io. PrintWriter**. O arquivo descritor de contexto (Deployment Descriptor) é o padrão de uma aplicação Web, segundo a especificação Java Servlet/Java EE, em um arquivo chamado de **web.xml**. Abra esse arquivo e verifique suas informações.

Vamos melhorar nosso formulário.

Links úteis HTML:

1. <http://www.clem.ufba.br/tuts/html/c11.htm>
2. <http://www.devmedia.com.br/como-criar-formularios-html-sem-usar-tabelas-tableless/28278>
3. <http://chasqueweb.ufrgs.br/~paul.fisher/apostilas/html/basicos/>
4. <http://www.lsi.usp.br/~help/html/comandos.html>

Vamos incluir um título e nomear o campo de entrada para termos a seguinte saída:



Formulário de Cadastro.

Nome:



Código:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Formulário</title>
  </head>
  <body>
    <h1> Formulário de Cadastro.</h1>
    <p></p>

    <h1> <form name="form1" action="PrimeiraServlet" method="POST">
      Nome: <input type="text" name="nome" value="" size="30" />
      <input type="submit" value="Enviar" name="btEnviar" />
    </form> </h1>

  </body>
```

Alterar a linha:

```
out.println("<h1> <FONT COLOR=ff0000> Autor: </FONT>" + nome + "</h1>");
```

gera a saída:



Servlet PrimeiraServlet at /agenda-fateczs

Autor: Sirley Ambrosia Vitorio Oliveira



Exercícios.

1. Crie um novo projeto Web deois uma servlet com formulário para fixar os comandos estudados.
2. Melhore seu formulário acrescentando:
 - a) Formatações de texto: negrito, itálico, cores(pesquisar tabela de cores em HTML)
 - b) Caixas de texto: endereço, cep, rg,cpf
 - c) Botões Radiais : sexo masculino/ feminino, estado civil.
 - d) Caixa de seleção.