

Instituto de

computação



# Desenvolvimento Web Servlets

*Prof. Leonardo Cruz.*

leonardocruz@id.uff.br

Departamento de Computação, UFF

# Recebendo dados de um Formulário

**Exemplo 2 - Fazendo uma requisição Get (pedi o FORM)  
Fazendo uma requisição Post (envia os dados)**

**veja Fonte Ex2**

# O Formulário

```
<html>
<body>
  <form method="post"
    action="http.....">
    Nome <br />
    <input type="text" name="nome" /><br />
    Endereco<br />
    <input type="text" name="endereco" /><br />
    <input type="submit" value="Enviar Para o Servlet" />
  </form>
</body>
</html>
```

Nome

Pedro

Endereco

Rua do lala

Enviar Para o Servlet

**Coloca o código do form no servlet ContatoServlet**

```
@WebServlet(urlPatterns = {"/ContatoServlet"})
```

```
public class ContatoServlet extends HttpServlet {
```

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException { response.setContentType("text/html");
```

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

```
    try {
```

```
        // gerando o Form
```

```
        out.println("<!DOCTYPE html>");
```

```
        out.println("<html>");
```

```
        out.println("<head>");
```

```
        out.println("<title>Servlet ContatoServlet</title>");
```

```
        out.println("</head>");
```

```
        out.println("<body>");
```

```
        out.println("<form method=\"post\" action=\"http://localhost:8084/");
```

```
        out.println("Nome<br>");
```

```
        out.println("<input type=\"text\" name=\"nome\" /><br />");
```

```
        out.println("Endereco<br>");
```

```
        out.println("<input type=\"text\" name=\"endereco\" /><br />");
```

```
        out.println("<input type=\"submit\" value=\"Enviar Para o Servlet\"");
```

```
        out.println("</form>");
```

# Chamada de ContatoServlet

localhost:8084/Ex2/ContatoServlet

Nome

Endereco

Enviar Para o Servlet

Requisição  
GET

```
view-source:http://localhost:8084/Ex2/ContatoServlet

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Servlet ContatoServlet</title>
5 </head>
6 <body>
7 <form method="post" action="http://localhost:8084/
8 Nome<br>
9 <input type="text" name="nome" /><br />
10 Endereco<br>
11 <input type="text" name="endereco" /><br />
12 <input type="submit" value="Enviar Para o Servlet"
13 </form>
14 </body>
15 </html>
16
```

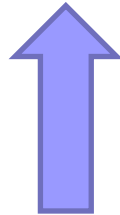
```
// gerando o Form
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet ContatoServlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<form method=\"post\" action=\"http://localhost:8084/E
out.println("Nome<br>");
out.println("<input type=\"text\" name=\"nome\" /><br />");
out.println("Endereco<br>");
out.println("<input type=\"text\" name=\"endereco\" /><br />");
out.println("<input type=\"submit\" value=\"Enviar Para o Servlet\"
out.println("</form>");
```

Código no navegador

Código Servlet

# Chamada de ContatoServlet

**http://localhost:8084/Ex2/ContatoServlet**



**Servidor: localhost**  
**Porta: 8084**



**Quem trata  
a solicitação**



**Requisição  
GET gera o form**

A screenshot of a web browser window. The address bar shows 'localhost:8084/Ex2/ContatoServlet'. The page content includes two text input fields labeled 'Nome' and 'Endereco'. Below these fields is a button labeled 'Enviar Para o Servlet'.

# O Formulário Gerado

```
6 <body>
7 <form method="post" action="http://localhost:8084/Ex2/ProcessarForm">
8 Nome<br>
9 <input type="text" name="nome" /><br />
10 Endereco<br>
11 <input type="text" name="endereco" /><br />
12 <input type="submit" value="Enviar Para o Servlet" />
13 </form>
...
```



**Servidor: localhost**  
**Porta: 8084**



**Quem trata  
a solicitação**

Outro servlet => ProcessarForm

**Tipo de requisição**

```
@WebServlet(urlPatterns = {"/ProcessarForm"})  
public class ProcessorForm extends HttpServlet {  
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
        PrintWriter out = response.getWriter();  
        try {  
            // pegando os parâmetros do request  
            String nome = request.getParameter("nome");  
            String endereco = request.getParameter("endereco");  
            // gerando a resposta  
            out.println("<!DOCTYPE html>");  
            out.println("<html>");  
            out.println("<head>"); out.println("<title>Servlet ContatoServlet</title>");  
            out.println("</head>"); out.println("<body>");  
            out.println("<h1>Contato </h1>");  
            out.println("<p>Nome: "+nome+"</p>");  
            out.println("<p>Endereço: "+endereco+"</p>");  
            out.println("</body>");  
            out.println("</html>");  
        } catch (Exception e) {  
            out.println("Erro: " + e.getMessage());  
        }  
    }  
}
```



```
@WebServlet(urlPatterns = {"/ProcessarForm"})

public class ProcessorForm extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            // pegando os parâmetros do request
            String nome = request.getParameter("nome");
            String endereco = request.getParameter("endereco");
            // gerando a resposta
```

```
<html>
<body>
    <form action="http://localhost:8084/Ex2/ContatoServlet">
        Nome <br />
        <input type="text" name="nome" /><br />
        Endereco<br />
        <input type="text" name="endereco" /><br />
        <input type="submit" value="Enviar Para o Servlet" />
```

```
out.println("</html>");
```

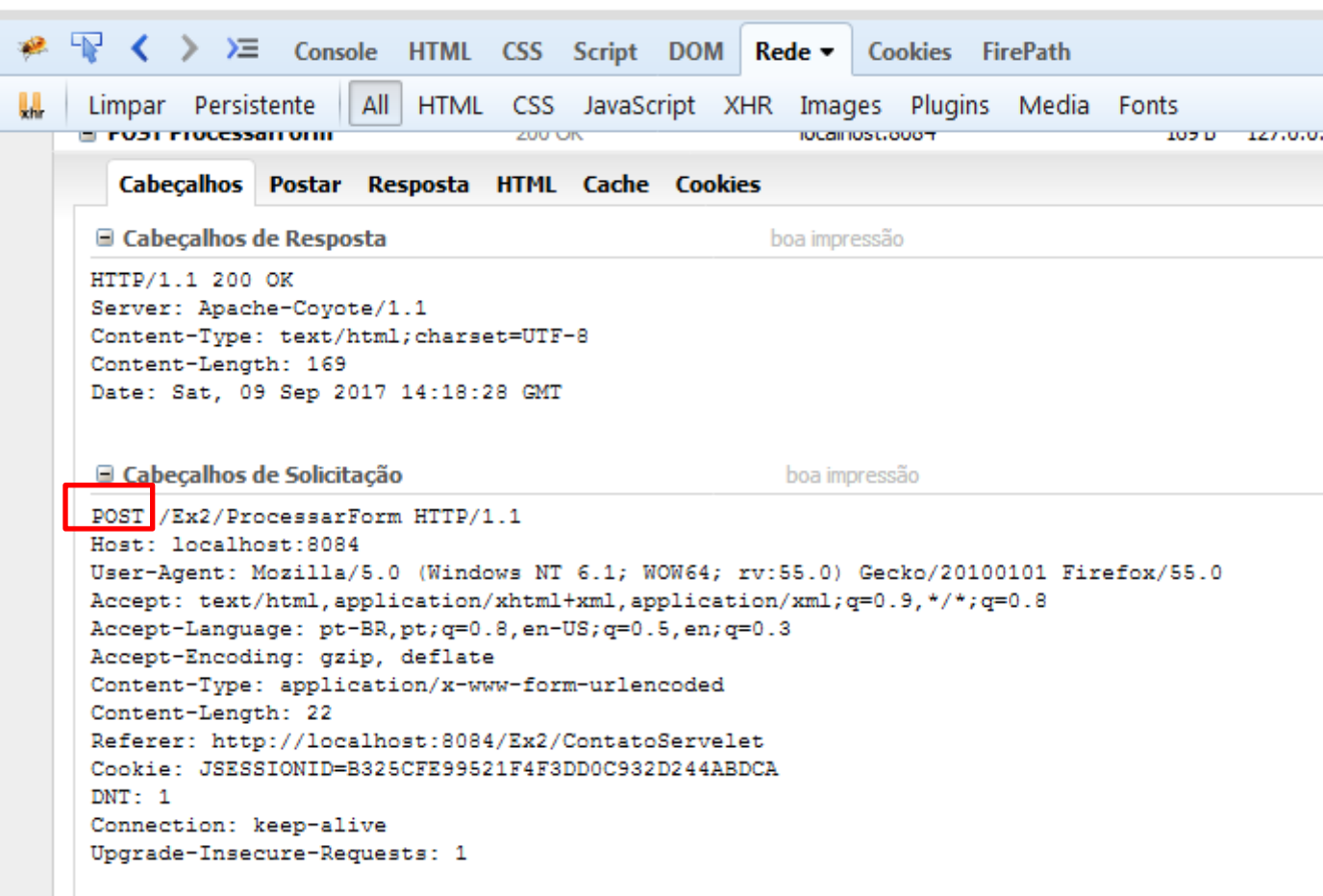


## Contato

Nome: leo

Endereço: lala

# Resposta do servlet



The screenshot shows a web browser's developer console with the 'Rede' (Network) tab selected. A request to 'POST /Ex2/ProcessarForm' is visible. The 'Cabecalhos' (Headers) sub-tab is active, showing the 'Cabecalhos de Resposta' (Response Headers) and 'Cabecalhos de Solicitação' (Request Headers). The 'POST' method in the request header is highlighted with a red box.

**Cabecalhos de Resposta** [boa impressão](#)

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=UTF-8
Content-Length: 169
Date: Sat, 09 Sep 2017 14:18:28 GMT
```

**Cabecalhos de Solicitação** [boa impressão](#)

```
POST /Ex2/ProcessarForm HTTP/1.1
Host: localhost:8084
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 22
Referer: http://localhost:8084/Ex2/ContatoServlet
Cookie: JSESSIONID=B325CFE99521F4F3DD0C932D244ABDCA
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

# Recebendo dados de um Formulário

- Através do método `getParameter("campo")` da classe `HttpServletRequest` podemos obter o valor de um determinado campo de formulário;

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

- No parâmetro desta função, inserimos **o mesmo nome de campo utilizado no formulário HTML** que submeteu os dados;
- O valor retornado por esta função será um dado do tipo **String**.
- Para atribuir tal valor a uma variável de outro tipo, devemos efetuar uma conversão;

# Tipo de Requisição

- Os requests podem ser de dois tipos:
- **POST** → Função é enviar dados para o servidor
- **GET** → Função é requisitar dados do servidor
  - POST: usualmente por forms
  - GET: usualmente por links/barra de url

# Requisição com GET

- Podemos passar alguns **parâmetros pela** requisição do tipo GET por meio do seguinte esquema:
- `http://servidor/servlet?param1=valor1`
- O “macete” é a interrogação: **?**
- Esse caractere indica que:
  - ☐ o endereço **já acabou**
  - ☐ tudo que vem em seguida é parâmetro

# Requisição com GET

- `http://servidor/servlet?param1=valor1`

`localhost:8084/Ex2/ProcessarForm?nome="Pedro"&endereco="Rua do Lala"`

- Depois da ?
  - Nome do parâmetro (no exemplo, **param1**)
  - Sinal de igualdade
  - Valor do parâmetro (no exemplo, **valor1**)

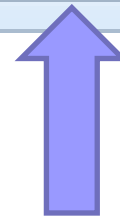
# Requisição com GET

- E se quiser passar mais de um parâmetro?

basta separá-los com o uso de um &

- `http://servidor/servlet?param1=valor1&param2=valor2`

```
localhost:8084/Ex2/ProcessarForm?nome="Pedro"&endereco="Rua do Lala"
```



# Requisição com GET

- Podemos passar quantos parâmetros quisermos?
- **NÃO com o GET tem limitação com relação ao número de caracteres**



# No nosso exemplo....

- Temos um efeito estranho...



# Contato

Nome: "Pedro"

Endereço: "Rua do Lala"



ConsoleHTMLCSSScriptDOMRedeCookiesFirePath

LimparPersistenteAllHTMLCSSJavaScriptXHRImagesPluginsMediaFonts

URL	Status	Domínio	Tamanho
GET ProcessarForm?nome=%22Pe	200 OK	localhost:8084	182 B

ParâmetrosCabecalhosRespostaHTMLCacheCookies

Cabecalhos de Resposta

boa impresso

HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Content-Type: text/html; charset=UTF-8  
Content-Length: 182  
Date: Sat, 09 Sep 2017 14:28:19 GMT

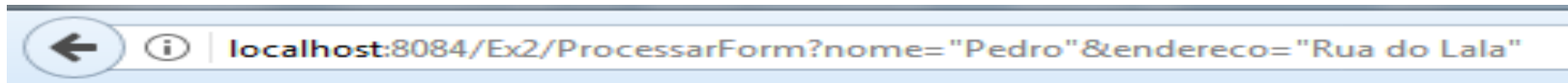
Cabecalhos de Solicitao

boa impresso

GET /Ex2/ProcessarForm?nome=%22Pedro%22&endereco=%22Rua%20do%20Lala%22 HTTP/1.1  
Host: localhost:8084  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3  
Accept-Encoding: gzip, deflate  
Cookie: JSESSIONID=B325CFE99521F4F3DD0C932D244ABDCA  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1

# ProcessarForm processa método POST

```
6 <body>  
7 <form method="post" action="http://localhost:8084/Ex2/ProcessarForm">  
8 Nome<br>
```



## Contato

Nome: "Pedro"

Endereço: "Rua do Lala"



Temos um efeito estranho...



## Requisição com GET para ProcessarForm ?

# Observação

- Usamos o mesmo servlet para processar uma requisição **Get** e uma requisição **Post**

```
localhost:8084/Ex2/ProcessarForm?nome="Pedro"&endereco="Rua do Lala"
```

```
<form action=http://localhost:8084/Ex2/ContatoServlet>
```

Nome

Endereco

Do ponto de vista do  
**servlet, o que muda?**

# Observação

- Usando NetBeans, **NADA**, os dados chegam, com o **GET**, da mesma forma que com o **POST**
- Quer dizer que não temos como diferenciar um do outro no **servlet**?

# Observação

- Quando criamos o servlet `ProcessarForm` (na realidade o **NetBeans**) o método que tratou as requisições foi **`processRequest`**

---

```
@WebServlet(urlPatterns = {"/ProcessarForm"})
public class ProcessarForm extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            // pegando os parâmetros do request
            String nome = request.getParameter("nome");
            String endereco = request.getParameter("endereco");
            // gerando a resposta
            out.println("<!DOCTYPE html>");
            out.println("<html>");
```

# Observação

```
// <editor-fold defaultstate="collapsed" desc="HttpServletRequest methods. Click on the
/**...*/
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
```



**Redireciona para processRequest**

```
/**...*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
```



**Redireciona para processRequest**

```
/**...*/
@Override
public String getServletInfo() {
    return "Short description";
}
```

# Observação

- **processRequest** é criado pelo **NetBeans** e não faz parte da api servlet



# Rejeitando GET

- E se não quisermos que nosso **servlet** responda com requisições GET?

Retirando a chamada **processRequest** do **doGet**

```
@Override  
protected void doGet(HttpServletRequest request, HttpServletResponse  
    throws ServletException, IOException {  
    processRequest(request, response);  
}
```

# Rejeitando GET

- E se não quisermos que nosso **servlet** responda com requisições GET?
  - Excluir o método **processRequest** e escrever o método específico para **doGet** e **doPost** especificamente

# Rejeitando GET

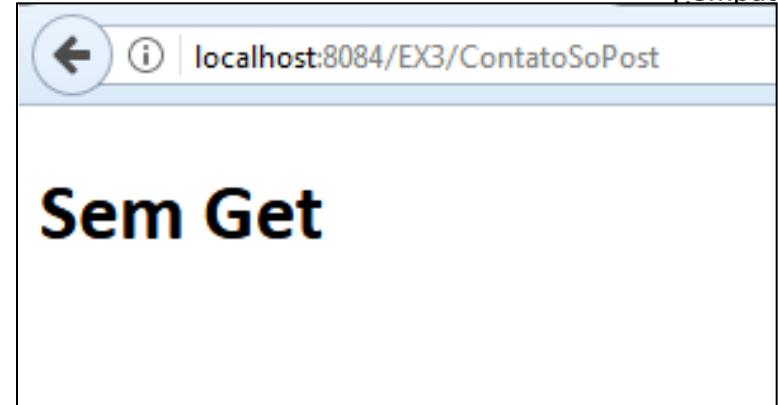
```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException { response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    try {
        // pegando os parâmetros do request
        String nome = request.getParameter("nome");
        String endereco = request.getParameter("endereco");

        // gerando a resposta
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet ContatoServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Contato </h1>");
        out.println("<p>Nome: "+nome+"</p>");
        out.println("<p>Endereço: "+endereco+"</p>");
        out.println("</body>");
        out.println("</html>");
    } catch (Exception e) {
        // tratamento de exceção
    }
}
```

**Retirar processRequest**

# Rejeitando GET

## doGet específico



```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    try {
        // gerando a resposta
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet ContatoServelet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Sem Get </h1>");
        out.println("</body>");
        out.println("</html>");
    } finally {
        out.close();
    }
}
```

Vide fonte ex3

# Rejeitando GET

## doPost específico

```
protected void doPost(HttpServletRequest request, HttpServletResponse resp
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    try {
        // pegando os parâmetros do request
        String nome = request.getParameter("nome");
        String endereco = request.getParameter("endereco");
        // gerando a resposta
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet ContatoServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Contato </h1>");
        out.println("<p>Nome: "+nome+"</p>");
        out.println("<p>Endereço: "+endereco+"</p>");
        out.println("</body>");
        out.println("</html>");
    } finally {
```

# Servlets - Mais um exemplo

- Criação do pacote - diretório  
(**EX4\src\java\br\servlet**)

- Cálculo do IMC (Imc.java)

EX4\src\java\br\servlet\Imc.java

- **http://localhost:8084/EX4/Imc?peso=96,8&altura=1,85**

A estrutura do diretório é transparente

- **Vide fonte ex4**

# Servlets – Alterando o Imc

## ■ Criando 3 servlet:

- Imc: recebe a solicitação e trata os valores
- ImcView: apresenta o valo do Imc
- ImcError: mostrar a mensagem de erro caso os dados estejam errados

- Vide fonte ex5



```
@WebServlet(name = "Imc", urlPatterns = {"/Imc"})
public class Imc extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse
        throws ServletException, IOException {
    try {
        String pesoT = request.getParameter("peso");
        pesoT = pesoT.replaceAll(",", ".");
        String alturaT = request.getParameter("altura");
        alturaT = alturaT.replaceAll(",", ".");
        double peso = Double.valueOf(pesoT);
        double altura = Double.valueOf(alturaT);
        Double imc = peso / (altura * altura);
        request.setAttribute("imc", imc);

        RequestDispatcher rd = request.getRequestDispatcher("/ImcView");
        rd.forward(request, response);

    } catch (Exception e) {
        RequestDispatcher rd = request.getRequestDispatcher("/ImcError");
        rd.forward(request, response);
    }
}
```



# Redirecionamento de Resposta

- Existem casos onde se deseja que um servlet tenha somente a responsabilidade de fazer um algum processamento...
- E deixar a geração de interface com o usuário para outro servlet
- **Redirecionamento de resposta**

# RequestDispatcher


- Classe usada para o redirecionamento de resposta
- Deve ser passada a página ou servlet para qual será redirecionado

```
RequestDispatcher rd = request.getRequestDispatcher("/ImcView");
```

# Forward

- Desconsidera qualquer saída do servlet via “out.print” ou outras

```
RequestDispatcher rd = request.getRequestDispatcher("/ImcView");  
rd.forward(request, response);  
out.println("qualquer coisa....");
```




isso é ignorado...

# Include

- Considera qualquer saída do servlet via “out.print” ou outras

```
RequestDispatcher rd = request.getRequestDispatcher("/ImcView");  
rd.include(request, response);  
out.println("qualquer coisa....");
```



isso não é ignorado...

# sendRedirect

- Outra forma de fazer redirecionamento
- Similar ao **Forward**
- O método `sendRedirect(String)` da resposta redireciona para uma URL e deixa o navegador tratar a URL

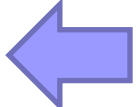
```
request.sendRedirect("http://www.uff.br");
```

# sendRedirect x Forward

- **sendRedirect várias diferenças se comparada ao Forward**
  - A url na barra de endereço do navegador reflete o alvo especificado
  - Dados armazenados no objeto de requisição anterior são descartados

# Servlets - Mais um exemplo

- Servlets + POO...
- Vide fonte ex6

public class **ContatoPessoal** {  **A Classe ContatoPessoal**

```
    private String Nome;  
    private String Endereco;
```

```
    public ContatoPessoal(String Nome, String Endereco) {  
        this.Nome = Nome;  
        this.Endereco = Endereco;    }
```

```
    public String getNome() { return Nome; }  
    public void setNome(String Nome) { this.Nome = Nome; }  
    public String getEndereco() { return Endereco; }  
    public void setEndereco(String Endereco) { this.Endereco =  
        Endereco;  
    }
```



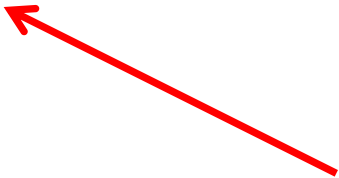
```
public class Contato extends HttpServlet {  
    protected void processRequest(HttpServletRequest request, HttpServletResponse  
        throws ServletException, IOException { response.setContentType("text/html");  
    try {  
        // pegando os parâmetros do request  
        String nome = request.getParameter("nome");  
        String endereco = request.getParameter("endereco");  
        // criação do objeto ContatoPessoal  
        ContatoPessoal meuContato = new ContatoPessoal(nome, endereco);  
  
        out.println("<!DOCTYPE html>");  
        out.println("<html>");  
        out.println("<head>");  
        out.println("<title>Servlet Contato</title>");  
        out.println("</head>");  
        out.println("<body>");  
        out.println("<h1>Contato</h1>");  
        out.println("<p>Nome:" + meuContato.getNome() + "</p>");  
        out.println("<p>Nome:" + meuContato.getEndereco() + "</p>");  
        out.println("</body>");  
    }  
}
```



# ServletConfig Interface

- Um objeto do ServletConfig é criado pelo container para cada servlet. Este objeto pode ser usado para obter informações de configuração do mapeamento (anotações @WebServlet).

```
@WebServlet(name = "Teste", urlPatterns = {"/Teste"},  
initParams = {  
    @WebInitParam(name = "dir-imagens", value = "c:/temp")  
})
```



**Definição do parâmetro de  
inicialização**

# Exemplo: ServletConfig

```
@WebServlet(name = "Teste", urlPatterns = {"/Teste"},
initParams = {
    @WebInitParam(name = "dir-imagens", value = "c:/temp")
})
public class Teste extends HttpServlet {
    private String dirImagens = "";

    public void init(ServletConfig config) throws ServletException {
        System.out.println("Método init()");
        dirImagens = config.getInitParameter("dir-imagens");
    }
}
```

Uso do método init

Obtendo parâmetro de inicialização da anotação e imprimindo esta informação no servlet.

Veja fonte ServletConfig

# Exemplo: ServletConfig

```
protected void processRequest(HttpServletRequest request,
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        /* TODO output your page here. You may use followi
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet Teste</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Diretorio usado " + dirImagens +
        out.println("</body>");
        out.println("</html>");
    } finally {
        out.close();
```

