

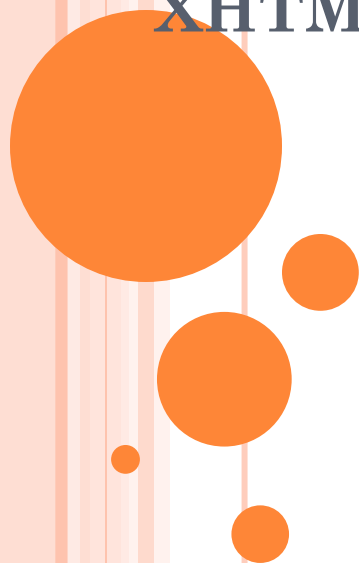
INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
CURSO TÉCNICO INTEGRADO EM INFORMÁTICA
CAMPUS CURRAIS NOVOS

Desenvolvimento Web

XHTML – Formulários

Professor: Bruno Gomes

Março de 2010



INTRODUÇÃO

- Um recurso bastante utilizado no desenvolvimento *Web* é o envio de dados de uma página para o servidor
 - quando fornecemos informações,
 - quando fazemos uma escolha em uma enquete, etc.
- Para tanto, utilizamos o recurso de formulários
 - Possuem vários elementos de interface que podem ser utilizados para o envio de dados para o servidor
 - Esses dados são ser processados para:
 - Gerar uma página com algum resultado
 - Atualizar um arquivo ou banco de dados, etc.



INTRODUÇÃO

- É possível também a interação do formulário com o cliente (navegador)
 - Por exemplo, ao utilizarmos uma linguagem de script do lado cliente (por exemplo, *JavaScript* ou *VBScript*) para validarmos um formulário
- *Validar* significa verificar se as entradas em um formulário estão corretas
 - Por exemplo, um campo de texto com um CEP deve ter um formato adequado (apenas números, cinco números seguidos de mais três números após um traço, etc.)



FORMULÁRIOS

- Um formulário deve ser definido dentro de uma *tag* **<form>**
- Elementos de entrada de dados com a *tag* **<input>**
 - Argumento **type** de **<input>** é utilizado para escolhermos o tipo do elemento



TEXTOS

- Para entrada de textos em uma única linha utilizamos o atributo **type** com o valor “*text*”
- O elemento **label** serve para acrescentarmos um texto explicando o propósito da entrada de texto

```
!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title> Tabelas </title>
  </head>
  <body>

    <form>
      <p>
        <label>Nome
          <input name="name" type="text" />
        </label>
      </p>
    </form>

  </body>
</html>
```

TEXTOS

- É possível limitar o número máximo de caracteres visíveis em um campo de texto
 - Atributo **size**
- Podemos também limitar o número máximo de caracteres permitidos
 - Atributo **maxlength**

```
<form>
<p>
<label>Nome
  <input name="name" type="text" maxlength="25" size="25"
/>
</label>
</p>
</form>
```

TEXTOS – INFORMAÇÕES ESCONDIDAS

- Informações passadas ao servidor que não são inseridas pelo usuário (e também não são visíveis por ele)
- Exemplo:

```
<form>
<p>
<input type="hidden"
      name="recebedor"
      value= bruno.gurgel@ifrn.edu.br />
<input type="hidden"
      name="assunto"
      value="Dados do formulário de notas dos alunos" />
</p>
</form>
```

TEXTOS – ENVIANDO SENHAS

- Um campo de textos para enviar dados sigilosos é inserido com o **type** *password*
- É exibido para o usuário apenas asteriscos enquanto os dados são digitados

```
<form>
<p>
<label>Nome
  <input name="name" type="password" size="25" />
</label>
</p>
</form>
```


TEXTOS – CAIXAS DE TEXTO

- Cria uma “caixa de texto” (textbox), ou seja, uma área de texto de mais de uma linha.
 - Tag `<textarea>`
 - Uma barra de rolagem aparece no canto direito da caixa quando há mais texto do que pode ser exibido
 - Você pode colocar algum texto predefinido no corpo da tag `<textarea>`

```
<p><label>Comentários<br />  
  <textarea name = "comments"  
    rows = "4" cols = "36">Digite algum comentário ou  
    sugestão aqui.  
  </textarea>  
</label>  
</p>
```



INTERAÇÃO COM O SERVIDOR

- O servidor Web serve (retorna ao cliente) apenas conteúdo estático
 - Exemplo, uma página da internet ou um outro arquivo
- Muitas vezes, é necessário interagir de forma dinâmica com o servidor
 - Obter informações de bancos de dados
 - Modificar páginas (ou partes dela) em tempo de execução
- Para tanto, é necessária a intervenção com outra linguagem ou programa, chamado de *script*



INTERAÇÃO COM O SERVIDOR

- Os dados do formulário geralmente são enviados para algum *script* no servidor para serem processados de alguma forma
- Algumas linguagens foram desenvolvidas com o propósito de interagir com os dados enviados da página e o servidor para gerar conteúdo dinâmico, como JSP (*Java Server Pages*) ou ASP (*Active Server Pages*)
 - Deve-se preferir o uso de uma dessas linguagens
- Você também pode utilizar o recurso de CGI
 - Método mais antigo que utilizar qualquer linguagem de programação para se programar a parte dinâmica da página
 - Servidor recebe o resultado de algum programa que esteja no próprio servidor ou em outra máquina acessível por ele na rede

EXEMPLO

Página estática

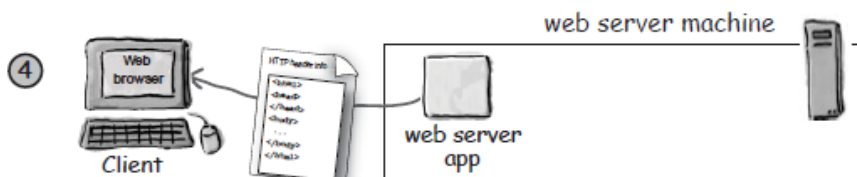
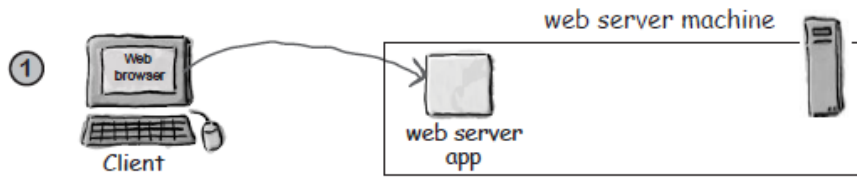
```
<html>
<body>
Data e hora atuais no
servidor: 02/04/2010
14:00 PM.
</body>
</html>
```

Página dinâmica com JSP

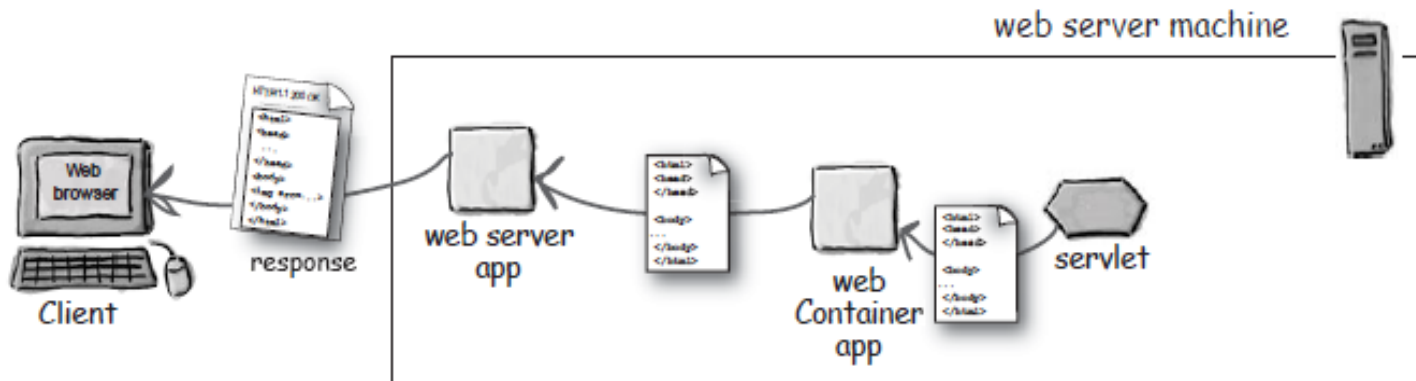
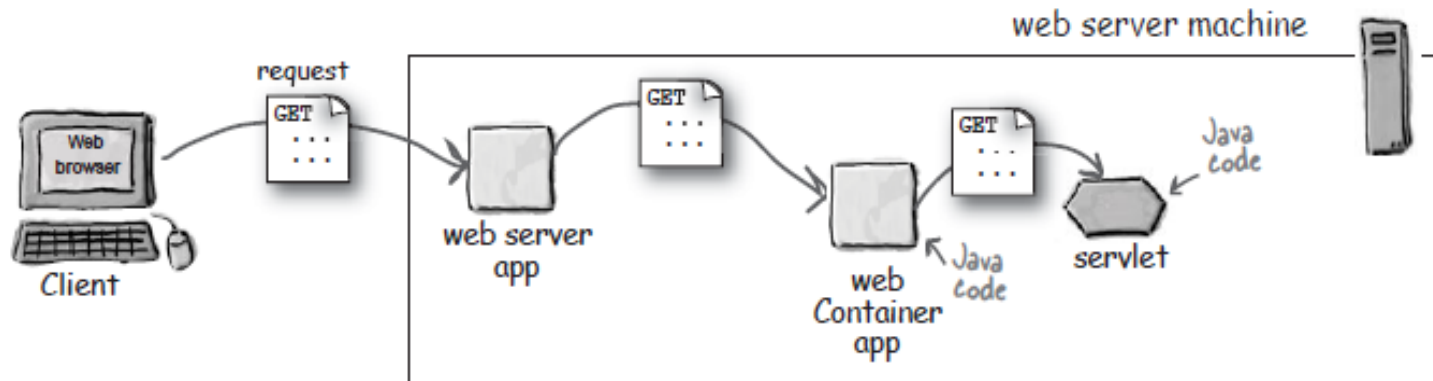
```
<html>
<body>
  Data e hora atuais no
  servidor:
  <%= new java.util.Date()
  %>
</body>
</html>
```



INTERAÇÃO COM O SERVIDOR - CGI



INTERAÇÃO COM O SERVIDOR – SERVLET/JSP



ENVIANDO DADOS PARA O SERVIDOR

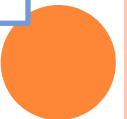
- Atributo **action** de `<form>` especifica o recurso (*script*, *p. ex.*) que irá receber e tratar os dados do formulário
- Atributo **type** de `<input>`, com o valor “submit” faz com que apareça um botão, que quando pressionado irá enviar os dados para o recurso

```
<form id="formnome" action="FormSubmit.jsp" method="get">
<p>
<label>Nome
  <input name="nome" type="text" maxlength="15" size="15" />
  <input name="sobrenome" type="text" maxlength="30" size="30" />
  <input type="submit" value="Enviar" />
</label>
</p>
</form>
```

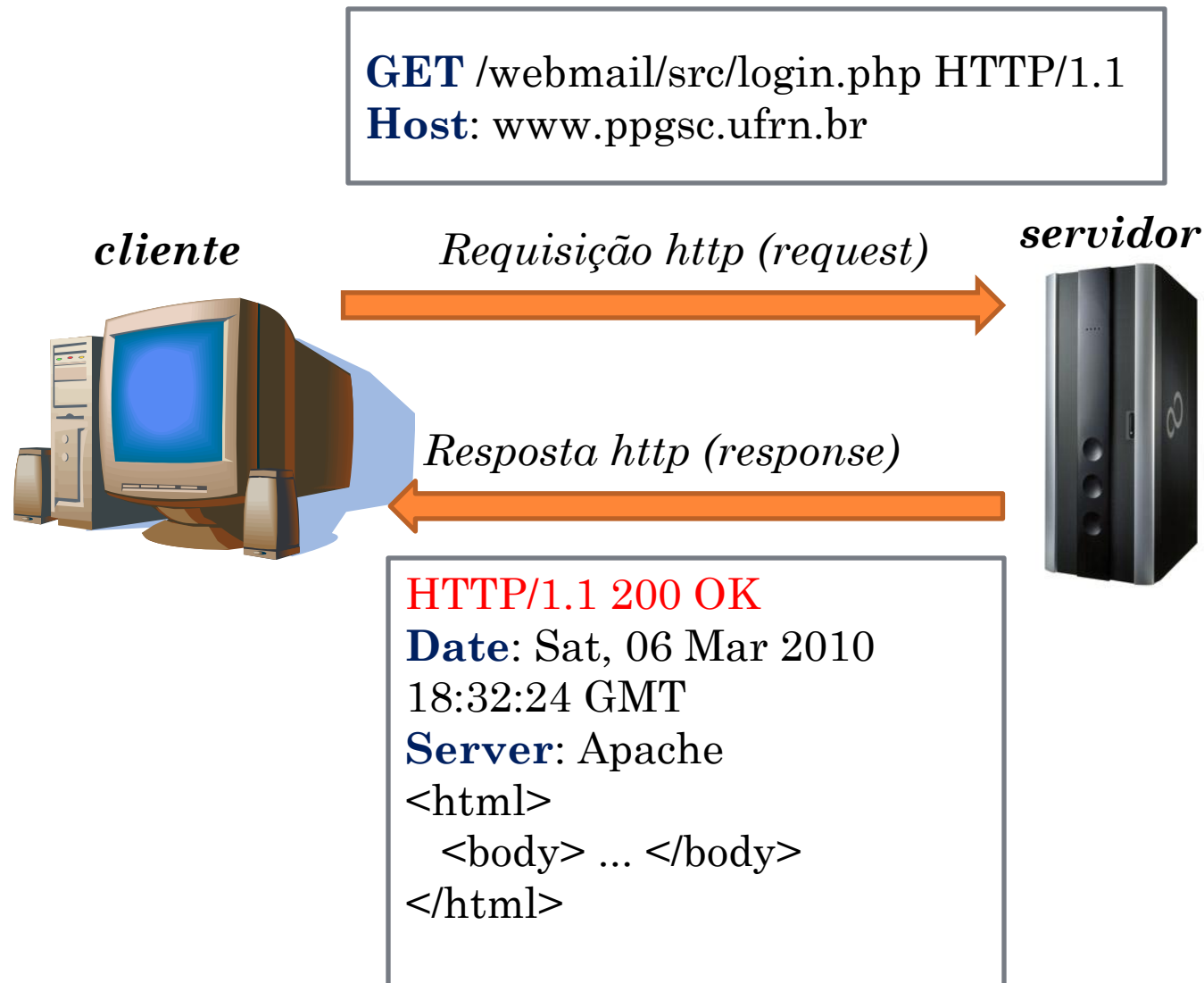
MÉTODOS GET E POST

- Atributo `method` do form é utilizado para informarmos de que forma iremos enviar os dados para o servidor
- Dados codificados na URL:
 - método **get**
- Dados dentro da requisição HTTP:
 - método **post**

```
<form id="formnome" action="FormSubmit.jsp" method="get">  
  ...  
</form>
```



REQUISIÇÃO HTTP



GET E POST

○ GET

- [http://www.ifrn.edu.br/~bruno/index.jsp?nome="bruno"&senha="123456"](http://www.ifrn.edu.br/~bruno/index.jsp?nome='bruno'&senha='123456')
- Dados codificados na URL
- Tamanho da URL com os dados de formulário ou outras informações é limitado – podendo gerar perda de informações

○ POST

- Dados enviados no corpo da requisição
- Não há limitação de tamanho
- Dados não ficam expostos ao usuário do navegador
- URL anterior ficaria:
 - <http://www.ifrn.edu.br/~bruno/index.jsp>



CHECK BOXES

- *Check Boxes* são utilizados para selecionar mais de uma opção dentre um conjunto de escolha

```
<p>  
  <strong>Áreas de interesse:</strong><br />  
  
  <label>Educação  
    <input name = "areas" type = "checkbox"  
      value = "educacao" /></label>  
  <label>Tecnologia  
    <input name = "areas" type = "checkbox"  
      value = "tecnologia" /></label>  
  <label>Esporte  
    <input name = "areas" type = "checkbox"  
      value = "esporte" /></label>  
  <label>Cultura  
    <input name = "areas" type = "checkbox"  
      value = "cultura" /></label>  
</p>
```



RADIO BUTTONS

- Radio *buttons* (“botões de rádio”) permitem a escolha de somente uma dentre várias opções

```
<p>
  <strong>Como você encontrou nosso Web site?</strong><br />

  <label> Página de busca
    <input name = "howtosite" type = "radio"
      value = "busca" checked = "checked" />
  </label>
  <label>Link a partir de outro site
    <input name = "howtosite" type = "radio"
      value = "link" />
  </label>
  <label>Indicação de um amigo
    <input name = "howtosite" type = "radio"
      value = "amigo" />
  </label>
  <label>Outro meio
    <input name = "howtosite" type = "radio"
      value = "outro" />
  </label>
</p>
```

CAIXA DE SELEÇÃO

- Apresenta uma lista na qual o usuário deve escolher uma dentre as opções exibidas

```
<p>
  <label>Estado
  <select name = "estado">
    <option selected = "selected">AL</option>
    <option>BA</option>
    <option>CE</option>
    <option>MA</option>
    <option>PB</option>
    <option>PE</option>
    <option>PI</option>
    <option>RN</option>
    <option>SE</option>
    <option>Outro</option>
  </select>
</label>
</p>
```

ENVIANDO UM EMAIL

```
<form action="mailto:bruno_gurgel@yahoo.com.br" method="post"
enctype="text/plain">
```

```
<h3>Enviando um email para o professor</h3>
```

```
<p>
```

```
    Name: <input type="text" name="nomec" value="nome" />
```

```
</p>
```

```
<p>
```

```
    Mail: <input type="text" name="mail" value="email" />
```

```
</p>
```

```
<p>
```

```
    Comment:
```

```
    <input type="text" name="comment" value="yourcomment" size="40" />
```

```
</p>
```

```
<p>
```

```
    <input type="submit" value="Send" />
```

```
    <input type="reset" value="Reset" />
```

```
</p>
```

```
</form>
```

EXERCÍCIO

- Crie um formulário para cadastro de um novo funcionário. O formulário deve conter os seguintes campos
 - Nome,
 - matrícula,
 - CPF,
 - identidade,
 - Cargo, que pode ser uma das opções abaixo:
 - Professor, técnico administrativo, técnico de laboratório, diretor de ensino, diretor geral, terceirizado
 - endereço, telefone
 - sexo (masculino ou feminino)

