

Aula 7: Formulários

Com certeza você já deve ter preenchido algum formulário em suas "viagens" pela Internet, pois esta é uma forma cada vez mais comum de interagir com os visitantes de um site, para obter dados do visitante ou registrar sua opinião. Os formulários são feitos como qualquer página. Eles são o assunto desta aula.

Objetivos:

Nesta aula você vai aprender tudo sobre construção de formulários e estará se habilitando a criá-los. Conhecerá:

- as formas de envio pelos métodos POST e GET,
- o envio por e-mail,
- os diversos elementos de interação: campos de entrada de texto, botões de seleção exclusiva e não-exclusiva e listas de seleção.

Pré-requisitos:

As aulas anteriores de links e tabelas serão importantes para entender bem os formulários.

1. Aspectos Gerais dos Formulários

A principal forma de trocar informações entre o usuário de uma página e o responsável por um site é através de formulários. A linguagem HTML oferece uma série de elementos de interação com o usuário que podem ser agrupados em um formulário.

O resultado desta interação é posteriormente enviado diretamente pela rede para um programa executado no servidor ou enviado por e-mail para o responsável pelo site.

A tag `<FORM> . . . </FORM>` delimita a área ocupada pelo formulário. Isto é, toda a seqüência de entrada de dados e de formatação do formulário deve ser delimitada por essa tag.

Esta tag possui dois atributos principais que indicam quem vai receber os dados do formulário (atributo `ACTION`) e a forma como eles serão enviados pela rede (atributo `METHOD`).

O atributo `ACTION` serve para informar a URL do programa CGI que irá processar o formulário. É possível utilizar este atributo para informar ao administrador do site que os dados do formulário serão enviados por e-mail. Neste caso, o método de envio deve, obrigatoriamente, ser definido como `POST`.

São duas as formas de envio pela rede:

- `Get` - os dados são anexados à URL.
- `Post` - os dados são enviados separados da URL.

Programas CGI
(Common Gateway Interface) – são programas executados pelo servidor Web, de onde recebem seus dados de entrada e para quem produzem sua saída (que geralmente é uma página Web). Frequentemente, são escritos em uma linguagem de *script*, como **Perl**

Quando o atributo METHOD recebe o valor GET, o conteúdo do formulário é enviado para o servidor anexado à URL definida no atributo ACTION. Este método é adequado a formulários pequenos, com pouca informação. Também não é conveniente enviar dados sigilosos por este método pois, além deles serem transmitidos sem nenhuma proteção, estarão visíveis na área de endereços do navegador.

O outro valor que pode ser atribuído a METHOD é POST. Neste caso, as informações do formulário são enviadas separadas da URL.

O exemplo a seguir mostra um formulário que será enviado para o endereço joao@nce.ufrj.br:

```
<FORM METHOD=POST ACTION="mailto:joao@nce.ufrj.br">
... </FORM>
```

O interior da tag <FORM>...</FORM> pode conter quatro novos tipos de tag que servem para definir os diversos elementos de interação. Estas tags são:

- <INPUT>
- <SELECT> ... </SELECT>
- <OPTION>
- <TEXTAREA> ... </TEXTAREA>

Antes de descrevermos os detalhes destas novas tags, vejamos um exemplo. Nas linhas abaixo, veremos o código que permitiu a criação do formulário mostrado na figura 7.1.

```
<HTML>
<HEAD>
  <TITLE> Curso CEDERJ: Construção de Páginas Web -
  Aula 7: Formulários
</TITLE>
</HEAD>
<BODY BGCOLOR="#DDFFFF">

  <FORM METHOD="POST" ACTION="mailto:aconci@ic.uff.br">

    <p>Se voce tem uma pagina relacionada a temas deste
    curso, ou visitou uma pagina que ache interessante,
    contribua apresentando este link para incluirmos aos
    nossos.</p>

    <p>Seu e-mail:
    <INPUT TYPE="TEXT" NAME ="Seu nome" SIZE="30"></p>

    <p>Qual o endereco do site?
    <INPUT TYPE="TEXT" NAME ="homepage" VALUE ="http://"
    SIZE =50 MAXLENGHT=100></p>

    <p>sexo:
    M <INPUT TYPE=radio NAME ="sex" VALUE =M>
    F <INPUT TYPE=radio NAME ="sex" VALUE =F></p><BR>

    <p>De onde voce vem?
    Brasil <INPUT TYPE=checkbox NAME ="country" VALUE
    ="Brasil" CHECKED>
    Portugal <INPUT TYPE=checkbox NAME ="country" VALUE
    ="Portugal">
```

```

Cabo Verde <INPUT TYPE=checkbox NAME ="country"
VALUE ="Cabo Verde">
Angola <INPUT TYPE=checkbox NAME ="country" VALUE
="Angola">
Mocambique <INPUT TYPE=checkbox NAME ="country"
VALUE ="Mocambique">
Timor Leste <INPUT TYPE=checkbox NAME ="country"
VALUE ="Timor Leste"></P>

<p>Seu interesse neste tema se relaciona a :
<SELECT NAME ="interesse">
<OPTION VALUE ="curiosidade"> curiosidade
<OPTION VALUE ="pesquisa">pesquisa
<OPTION VALUE ="programas">programas
<OPTION VALUE ="trabalhos">trabalhos
<OPTION VALUE ="imagens">imagens
<OPTION VALUE ="aplicacao">aplicacao
<OPTION VALUE ="outros">outros</SELECT ></p>

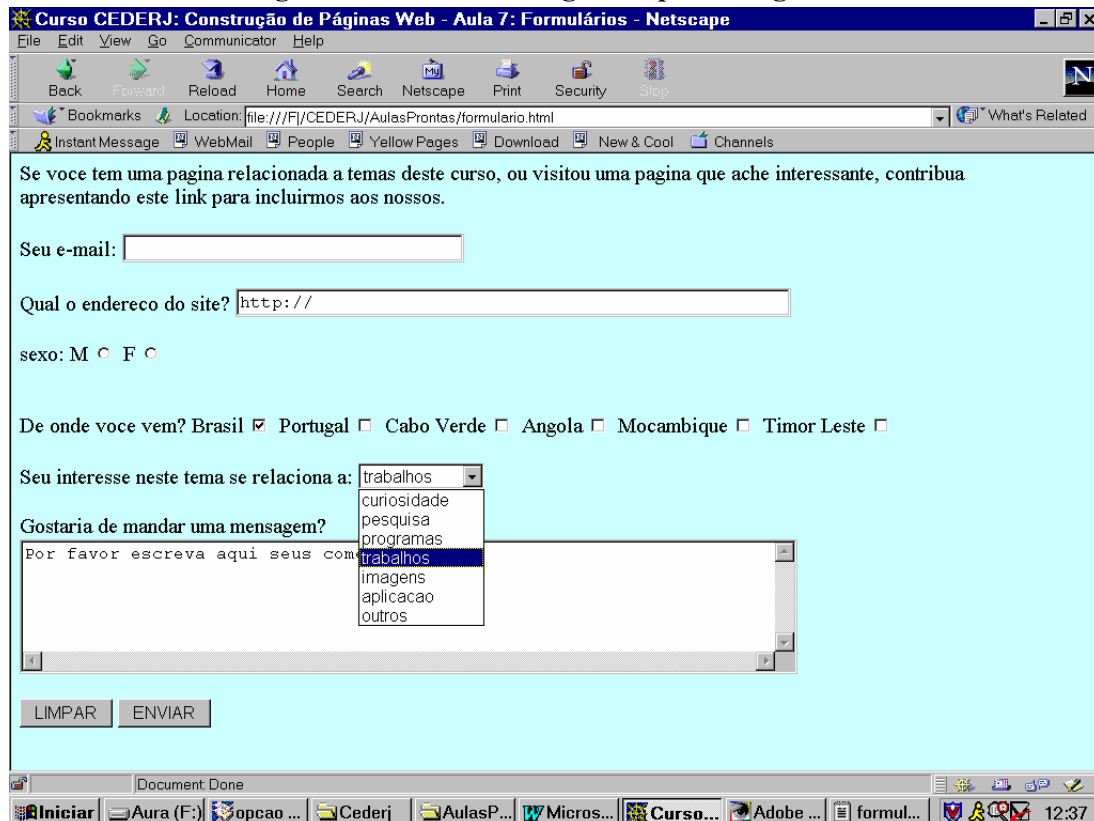
<p>Gostaria de mandar uma mensagem?
<TEXTAREA NAME ="message" ROWS=5 COLS=70>
Por favor escreva aqui seus comentários
</TEXTAREA></p>

<INPUT TYPE="RESET" VALUE="LIMPAR">
<INPUT TYPE="SUBMIT" VALUE="ENVIAR">
<INPUT TYPE ="HIDDEN" NAME="mail" VALUE
="aconci@ic.uff.br">

</FORM>
</BODY>
</HTML>

```

Figura 7.1 - Formulário gerado pelo código HTML anterior



2. A tag <INPUT>

A tag <INPUT> é bastante versátil e permite criar:

- campos para entrada de texto
- campos para entrada de senhas
- botões comuns
- botões estilo rádio
- botões de seleção múltipla

A tabela 7.1 mostra os atributos e valores possíveis de <INPUT>. O atributo `type` permite definir o tipo de entrada de dados e o aspecto do elemento de interação. Dependendo do valor atribuído à `type`, aparecerão janelas onde os dados serão digitados, campos para senhas ou botões dos diversos tipos. Para alguns atributos de `type`, os outros atributos de `input` podem nem fazer sentido.

Tabela 7.1 Atributos de <INPUT>

Atributo	Valor	Descrição
TYPE	TEXT	Entrada para texto.
	PASSWORD	Campo para senha.
	RADIO	Botões de seleção exclusiva (<i>radio</i>).
	CHECKBOX	Botões de seleção múltipla.
	BUTTON	Botão comum.
	SUBMIT	Botão que envia o formulário.
	RESET	Botão que limpa os campos do formulário.
	HIDDEN	Campo que não será mostrado, útil para atribuir valores a NAME.
NAME	String	Nome da variável associada ao campo.
VALUE	String	Valor default do campo.
SIZE	Tamanho	Tamanho do campo TEXT ou PASSWORD.
MAXLENGTH	Tamanho	Número máximo de caracteres a ler.
CHECKED		Define um campo CHECKBOX como selecionado.

O valor **String** indica que o dado será um texto.

Tamanho corresponde a um valor numérico.

2.1. Usando a Tag <INPUT> para Criação de um Campo para Entrada de uma Linha de Texto

A definição do atributo `TYPE=TEXT` na tag <INPUT> permite a criação de um campo para entrada de uma linha de texto. A aparência que isso terá no formulário gerado é a de uma área retangular onde é possível escrever uma linha de texto. Repare, na figura 7.1, as áreas retangulares que aparecem após os textos: "Seu e-mail:" e "Qual o endereço do site?"

As informações digitadas por um usuário no formulário são enviadas ao servidor no formato "nome=valor". Por isso, cada elemento de interação contém um atributo `NAME` que deve obrigatoriamente ter um nome definido de modo que este elemento possa ser identificado pelo CGI. O exemplo abaixo faz com que o navegador exiba na página um campo de entrada de texto; caso o

usuário digite "joao" e envie o formulário, a informação será recebida pelo programa na forma de "conta=joao":
Conta:<INPUT TYPE=TEXT NAME=conta>

É possível preencher o campo com um valor padrão que é exibido após o carregamento da página. Isso é feito através do atributo VALUE, como mostrado na segunda janela de texto das linhas de HTML que geram a figura 7.1.

A omissão do atributo TYPE, na tag <INPUT>, faz com que o navegador assumo o tipo TEXT. Ou seja, esta é a opção **default**.

É possível definir o tamanho da área para entrada de texto através do atributo SIZE. Este atributo não limita o número de caracteres que o usuário pode digitar, apenas o tamanho do campo.

Para se indicar o número máximo de caracteres que pode ser digitado pelo usuário deve-se definir o atributo MAXLENGTH.

No exemplo abaixo, a área de entrada de texto tem o tamanho de 7 caracteres, mas o usuário pode digitar até 11 caracteres:
Telefone:
<INPUT TYPE=TEXT NAME=telef SIZE=7 MAXLENGTH=11>

2.2. Usando a Tag <INPUT> para Criação de um Campo para Entrada de Senha

A definição do atributo TYPE=PASSWORD na tag <INPUT> permite a criação de um campo para entrada de informações sigilosas (como por exemplo, senhas). A aparência deste campo no formulário é a mesma do campo de texto. No entanto, no campo PASSWORD, os caracteres digitados pelo usuário não são exibidos no monitor, sendo substituídos por asteriscos "*" .

Para tirar proveito do "sigilo", este tipo de campo deve ser enviado pelo método POST.

No exemplo abaixo, se o usuário digitar "pcpw02" e enviar o formulário, a informação recebida pelo programa será "senha=pcpw02":
Senha:<INPUT TYPE=PASSWORD NAME=senha>

É possível definir o tamanho da área para entrada de texto através do atributo SIZE e o número máximo de caracteres que pode ser digitado pelo usuário através do atributo MAXLENGTH, exatamente como no caso dos campos de entrada de texto comuns.

2.3. Usando a Tag <INPUT> para Criação de Botões de Seleção Exclusiva

Um tipo de elemento de interação muito comum são os botões de seleção exclusiva (que são os chamados **Radio Buttons**). Estes botões são utilizados quando o usuário deve escolher uma única resposta entre várias. Quando um dos botões é selecionado, automaticamente todos os outros são desmarcados.

Só uma opção entre as várias apresentadas pode estar marcada. A aparência destes é de um botãozinho redondo, como pode ser visto na figura 7.1.

Estes elementos são criados pela tag `<INPUT>` com o atributo `TYPE=RADIO`. Para indicar ao navegador quais botões são mutuamente exclusivos é preciso que eles tenham o atributo `NAME` definido com o mesmo valor.

O atributo `VALUE` deve ser definido de modo a diferenciar qual dos botões está selecionado (já que todos têm o mesmo `NAME`). Através do atributo `CHECKED` é possível indicar que uma das alternativas estará previamente selecionada. No exemplo abaixo, a opção "Superior" está previamente selecionada; se o usuário selecionar "Analfabeto" a informação enviada ao programa será "instruc=A".

```
Nível de Instrução:<br>
<INPUT TYPE=RADIO NAME=instruc VALUE=A>Analfabeto<br>
<INPUT TYPE=RADIO NAME=instruc VALUE=1>Primeiro grau
<br>
<INPUT TYPE=RADIO NAME=instruc VALUE=2>Segundo grau
<br>
<INPUT TYPE=RADIO NAME=instruc VALUE=S CHECKED>Nivel
superior<br>
```

2.4. Usando a Tag `<INPUT>` para Criação de Botões de Seleção Múltipla

Quando é desejável escolher mais de uma resposta, utilizam-se botões de seleção múltipla (*toggle buttons*). A criação destes botões é feita através da tag `<INPUT>` com o atributo `TYPE=CHECKBOX`.

No formulário, a aparência destes botões é de quadradinhos (como é mostrado na figura 7.1). Diferente dos botões exclusivos, neste caso, vários botões podem ser selecionados pelo usuário ou podem estar previamente selecionados pela definição do atributo `CHECKED`.

Através do atributo `CHECKED`, no exemplo a seguir, foi possível pré-selecionar as opções "Netscape" e "Explorer"; se o usuário enviar o formulário sem fazer nenhuma modificação, o programa vai receber "nav1=N" e "nav3=E":

```
Navegador utilizado:<BR>
<INPUT TYPE=CHECKBOX NAME=nav1 VALUE=N CHECKED>
Netscape<br>
<INPUT TYPE=CHECKBOX NAME=nav2 VALUE=M>Mosaic<br>
<INPUT TYPE=CHECKBOX NAME=nav3 VALUE=E CHECKED>
Explorer<br>
<INPUT TYPE=CHECKBOX NAME=nav4 VALUE=H>HotJava<br>
```

Os atributos `NAME` e `VALUE` devem ser definidos como nos botões de seleção exclusiva.

2.5. Usando a Tag <INPUT> para Criação de Botões de Ação

A tag <INPUT> também serve para a criação de botões de ação. Estes botões, diferente dos elementos de interação que vimos até agora, não estão associados a informações que serão enviadas ao servidor. A seleção de um destes botões pelo usuário faz com que uma ação seja realizada (pode ser uma ação padrão do navegador como: enviar o formulário ou apagar o conteúdo de seus campos ou ainda o resultado da execução de um programa numa linguagem de *script*).

Estes botões têm a aparência de teclas, como você pode observar na parte inferior da figura 7.1.

Estes botões são gerados pela atribuição dos valores `BUTTON`, `RESET` ou `SUBMIT` a `TYPE`. Os dois últimos valores têm ações especiais associadas. A definição do atributo `TYPE=SUBMIT` inclui um botão para fazer o envio do formulário. A definição do atributo `TYPE=RESET` inclui um botão que limpa os campos do formulário, permitindo que o usuário redefina suas escolhas.

Nestes botões, o atributo `VALUE` define um texto para ser escrito no interior do botão. A omissão deste atributo faz com que o navegador exiba um texto **default**, em língua inglesa.

Normalmente, estes botões são colocados no fim do formulário. O exemplo, a seguir, mostra a utilização destes botões para criação de botões de ação com os títulos definidos pelas *strings* atribuídas à `VALUE`:

```
<FORM>
    ...
    <INPUT TYPE=SUBMIT NAME="Envia" VALUE="Envia dados
    agora" >
    <INPUT TYPE=RESET NAME="Limpa" VALUE="Apagar
    tudo" >
    ...
</FORM>
```

3. A Tag <textarea> : Área de Texto

No exemplo inicial desta aula, a tag <TEXTAREA> foi usada para desenhar a caixa onde o usuário escrevia. Este é seu uso mais comum, isto é, quando se deseja enviar mais de uma linha de texto, deve-se utilizar a tag <TEXTAREA> ... </TEXTAREA> .

O atributo `NAME` deve ser definido para indicar a variável que será associada ao texto inserido. O tamanho da área de texto é definido pelo número de linhas (atributo `ROWS`) e pelo número de colunas (atributo `COLS`).

No interior da tag é possível colocar um **texto** a ser exibido após o carregamento da página. O exemplo a seguir mostra essa utilização, neste caso o texto: "Faça aqui seus comentários:" aparecerá inicialmente no interior da caixa e será substituído pelo texto que o usuário digitar.

```
Comentários:<BR>
<TEXTAREA NAME=coment ROWS=3 COLS=40>
Faça aqui seus comentários:
</TEXTAREA>
```

4. As Tags <SELECT> e <OPTION>: Listas de Seleção

Quando há muitas opções a escolher, pode não ser muito prático utilizar botões de seleção exclusiva ou botões de seleção múltipla para não ocupar muito espaço na página. Uma alternativa a eles são as listas de seleção criadas através da tag <SELECT> ... </SELECT>.

No exemplo do início da aula, esta tag foi usada para apresentar ao usuário as opções que aparecem após a linha: "Seu interesse neste tema se relaciona a:". Como você pode ver na figura 7.1, no formulário, ela criará um "menu suspenso" que aparece na forma de uma janela com lista de opções. Este tipo de menu é chamado de *pull down*.

No interior da tag <SELECT> deve ser colocada cada uma das opções da lista (no interior da tag <OPTION>). O atributo SELECTED pode ser utilizado para definir qual opção está previamente selecionada. O exemplo a seguir mostra a utilização deste atributo:

Sexo:

```
<SELECT NAME=sexo>
<OPTION SELECTED>Masculino
<OPTION>Feminino
</SELECT>
```

O atributo MULTIPLE indica que mais de uma opção pode ser selecionada, conseqüentemente, se ele estiver definido, podemos utilizar o SELECTED em mais de uma opção.

O atributo NAME deve ser utilizado para definir o nome da variável que será enviada ao programa. O atributo SIZE indica o número de opções mostradas de uma só vez pelo navegador, sendo que o valor **default** é SIZE=1 (como pode ser visto nas linhas HTML do exemplo inicial). As demais opções podem ser acessadas através de uma barra de rolagem, que se abre ao ser clicado o ícone à direita da lista (com aparência de um triângulo apontando para baixo). A figura 7.1 mostra o exato momento em que a lista de opções está sendo exibida. Numa lista de SIZE=1, após o carregamento da página, apenas o OPTION que estiver com o atributo *selected* definido é mostrado.

Exercício:

Utilize as novas tags para criar uma página de pesquisa de opinião sobre "**preferências esportivas**". Desenhe no formulário campos com botões exclusivos para identificar o sexo, o estado civil e a faixa etária da pessoa que responde à pesquisa. Atribua a botões de seleção múltipla diversas modalidades esportivas, e permita que o usuário que selecionar "outras" inclua uma nova. Use campo de texto para identificar a cidade onde mora a pessoa. Através de <select> permita que ela aponte o estado onde mora. Permita que ela inclua comentários através de um campo de *textarea*. Inclua os botões para envio ou redefinição das respostas do formulário. Faça, finalmente, a página ter um alinhamento de parágrafo agradável e inclua um título.

Resumo:

Nesta aula, você aprendeu novas tags para definição dos elementos de interação dos formulários na linguagem HTML. Viu como definir a forma como os elementos de interação apareceram em um documento e o conteúdo que o formulário, quando preenchido, retornará ao programa que manipulará os dados enviados pelo usuário. Fez ainda uso destes recursos, testando seus novos conhecimentos no projeto uma página de pesquisa de opinião (no exercício sugerido).

Auto-avaliação:

Você construiu a página de pesquisa com facilidade? Então está pronto para criar páginas com esse recurso poderoso de interação com os visitantes. Se não está bem seguro, já sabe o que deve fazer. Ou precisamos lembrar? Por certo que não, afinal estamos no final de uma etapa, a próxima é nossa última aula do primeiro módulo do nosso curso.