



Tutorial - Autoria em World Wide Web



Parte III - Hypertext Markup Language (HTML) - FORMULÁRIOS

(HTML 2.0 e algumas extensões do Netscape)

© Copyright 1995 Rede Nacional de Pesquisa - RNP

É PERMITIDA A REPRODUÇÃO DO PRESENTE MATERIAL, DESDE QUE TENHA FINS EDUCACIONAIS OU DE PESQUISA, DEVENDO, NO ENTANTO, SER MENCIONADOS EXPRESSAMENTE OS DIREITOS AUTORAIS RESERVADOS À RNP. É VEDADA A REPRODUÇÃO PARA QUAISQUER OUTROS FINS, A MENOS QUE CONCEDIDA EXPRESSA AUTORIZAÇÃO PELA RNP.

Formulários Eletrônicos

Este tutorial é uma tradução e adaptação do Carlos' FORM Tutorial
c-pero@uiuc.edu

ÍNDICE

I. Sobre este Tutorial	3
II . Codificação básica do formulário	4
III. Texto	5
a. entrada de texto comum	5
b. entrada de texto protegido, senha	6
c. entrada oculta	7
d. entrada de várias linhas de texto	8
IV. Menus	9
a. menus com opções	9
b. listas pagináveis	10
V. Botões	12
a. botões sim-ou-não	12
b. botões com opções	13
c. botões de submissão e limpeza	14

I. Sobre este Tutorial

Neste tutorial, você vai aprender a criar "Formulários Eletrônicos", que permitem um usuário do Web submeter informações para o provedor do serviço.

Existem vários tipos de campos de entrada, como:

- campos de entrada de texto
- menus de múltipla escolha ou escolha única
- botões sim-ou-não
- botões para submissão ou limpeza de formulário

Cada uma destes campos tem funcionalidade própria e você vai aprender em que situação utilizá-los em seus formulários.

Para melhor compreender a notação utilizada nos formulários, você primeiro precisa se familiarizar com HTML (Hypertext Markup Language). Consulte, se necessário, o nosso tutorial sobre HTML básico disponível em <http://www.cr-df.rnp.br/hipertextos/cr-df/cursos/html1/>

Se você já conhece HTML, as marcações utilizadas em formulários seguem a mesma convenção, e portanto, será fácil aprender.

Contudo, formulários HTML sempre seguem um padrão:

```
<TITLE>Título do Documento</TITLE>
```

no início, e o restante do documento a seguir.

Além disto, as marcações que definem o formulário precisam ser iniciadas e finalizadas, como estas marcações de listas.

```
<UL>
<LI>Primeiro item da lista
<LI>Segundo item da lista
</UL>
```

```
<FORM>
  conteúdo do formulário...
</FORM>
```

Uma última observação: você pode ter múltiplos formulários em um único hiperdocumento. Apenas certifique-se que os blocos `<FORM></FORM>` não se sobrepõem.

II. Codificação básica

Como vimos, um formulário começa com a marcação **<FORM>** e termina com **</FORM>**. Outros itens devem ser especificados:

- **Primeiro**, o formulário precisa saber como enviar a informação para o servidor. Existem dois métodos, GET e POST.

METHOD="GET"

A maioria dos documentos HTML são recuperados a partir da requisição de uma única URL ao servidor. Assim, um formulário que utilize este método, envia toda sua informação ao final da URL ativada.

METHOD="POST"

Este método transmite toda a informação fornecida via formulário, imediatamente após a URL ativada. Ou seja, quando o servidor recebe uma ativação de um formulário utilizando POST ele sabe que precisa continuar "ouvindo" para obter a informação. Este é o melhor método.

- **Segundo**, o formulário precisa saber para onde enviar a informação. Esta é a URL sendo ativada a partir do formulário, e ela é referenciada através da marcação **ACTION**. Esta URL em geral aponta para um script CGI que irá receber e decodificar os resultados. Lembre-se que se você está referenciando um script que reside no mesmo servidor do formulário, você não precisa incluir a URL completa.

ACTION="/cgi-bin/post-query" para um script no seu servidor

ACTION="http://www.cr-df.rnp.br/cgi-bin/post-query" para um script no servidor do CR/DF

Após você construir estas marcações, seu formulário geralmente terá a seguinte estrutura:

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Marcações de campos de entrada e HTML em geral

```
</FORM>
```

Observe que este formulário utiliza o método POST e envia as informações digitadas para um script local chamado post-query no diretório /cgi-bin do servidor.

Outra informação importante: cada marcação de entrada em um formulário tem uma opção **NAME** associada, de tal forma que o script saiba qual o nome, isto é, como chamar cada valor digitado. Certamente você pode definir mais de um campo de entrada textual ou menu dentro de um formulário, mas certifique-se de que cada um possui um nome diferente.

III. Texto

-
- a. Entrada de texto comum - **TEXT**
 - b. Entrada de texto protegido, senha - **PASSWORD**
 - c. Entrada oculta - **HIDDEN**
 - d. Entrada de várias linhas de texto - **TEXTAREA**
-

a. Entrada de texto comum - **TEXT**

A forma mais simples de campo de entrada é a marcação **text**. Este campo permite a digitação de uma única palavra ou linha de texto, e possui uma largura default de 20 caracteres.

Opções:

VALUE="" OPCIONAL

Utilizando a marcação **VALUE** você especifica que texto aparecerá no campo quando o formulário for exibido.

SIZE="" OPCIONAL

Esta marcação altera o tamanho deste campo exibido na tela.

Obs.: o usuário sempre poderá digitar mais caracteres do que o tamanho do campo na tela, pois o texto irá se deslocar a esquerda dentro do campo.

MAXLENGTH="" OPCIONAL

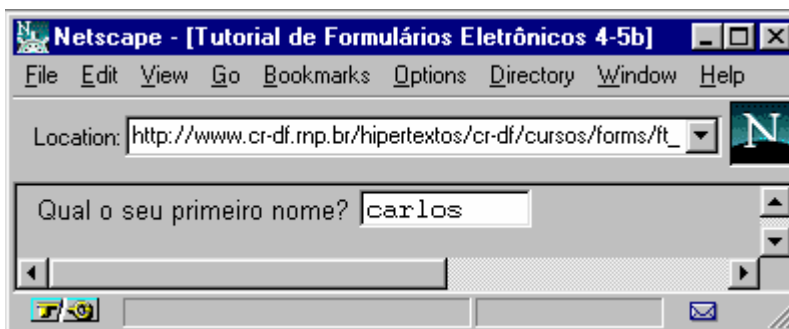
Se você deseja limitar o número de caracteres que o usuário pode digitar, basta usar esta marcação. O formulário irá emitir um bip de erro se o usuário tentar digitar além do permitido em **MAXLENGTH**.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Qual o seu primeiro nome?

```
<INPUT TYPE="text" NAME="primeiro_nome" VALUE="carlos" SIZE="10" MAXLENGTH="15">
```

```
</FORM>
```



Obs.:

- O valor informado em **NAME** deve utilizar sublinhado em vez de espaços em branco. Inserir espaços em branco nestes nomes pode causar problemas no servidor na decodificação dos valores informados.
- Se seu formulário possui apenas um campo de entrada textual, ao teclar **ENTER** neste campo o formulário será submetido, como se o usuário tivesse acionado o botão **SUBMIT**.

b. Entrada de texto protegido, senha - PASSWORD

Marcações de entrada do tipo **password** são idênticas aos campos do tipo text, exceto pelo fato de todos os caracteres serem exibidos como asteriscos (*).

-

Opções:

VALUE="" OPCIONAL

A marcação VALUE especifica um valor default para este campo.

SIZE="" OPCIONAL

Esta troca o tamanho do campo de password exibido na tela.

MAXLENGTH="" OPCIONAL

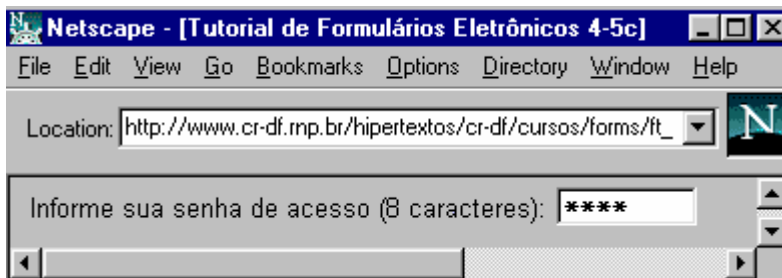
Limita o número de caracteres que o usuário pode informar como password.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Informe sua senha de acesso (8 caracteres):

```
<INPUT TYPE="password" VALUE="xpto" NAME="sua_senha" SIZE="8" MAXLENGTH="8">
```

```
</FORM>
```



Obs.:

- Se você pretende utilizar MAXLENGTH para limitar o número de caracteres informados, tenha certeza de especificar um SIZE com o mesmo tamanho, para fornecer ao usuário o sentimento de qual a largura do campo. Esta regra não foi utilizada no primeiro exemplo, que possui SIZE igual a 10, mas MAXLENGTH de 15, o que faz o campo se deslocar nos últimos 5 caracteres caso o valor informado possua mais do que 10 caracteres.
- Mesmo que o valor default VALUE esteja representado por asteriscos, o usuário pode visualizar o seu valor através da recuperação do fonte do hiperdocumento em HTML (uma opção comum na maioria dos clientes WWW).

c. Entrada oculta - HIDDEN

Além da marcação PASSWORD com um valor VALUE default, é possível "esconder" informação passada dentro de blocos FORM com uma marcação **HIDDEN**. Esta informação é recebida pelo script de decodificação no servidor, mas não é diretamente exibida ao usuário no formulário.

Opções:

VALUE="" OBRIGATÓRIO

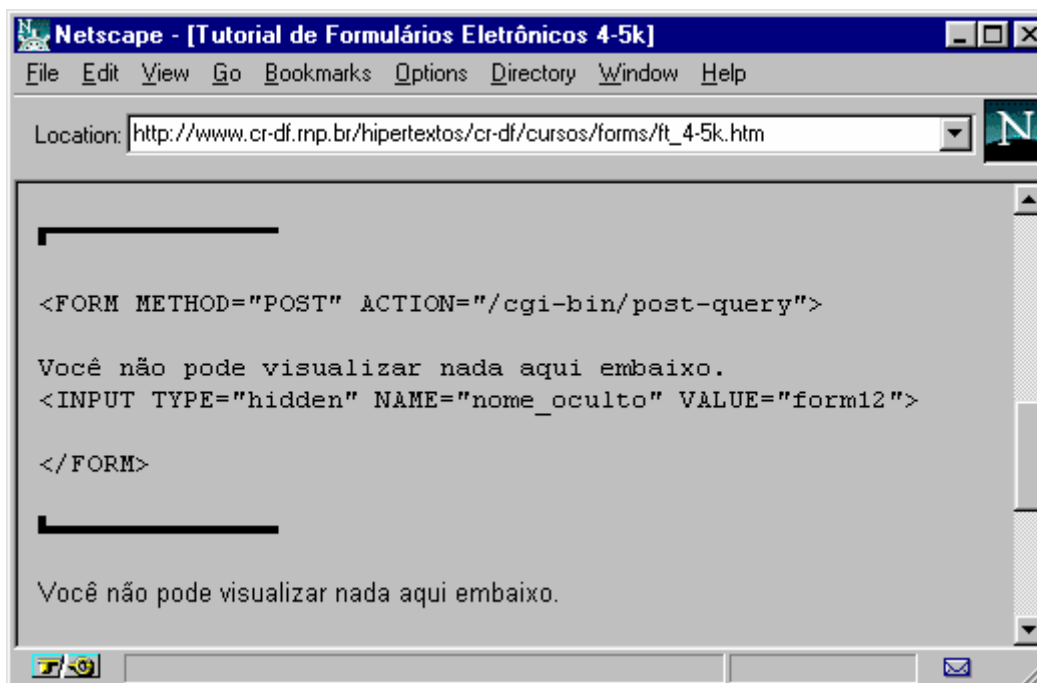
Através da marcação VALUE, você deve especificar o texto oculto a ser enviado ao script processador do formulário.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Você não pode visualizar nada aqui embaixo.

```
<INPUT TYPE="hidden" NAME="nome_oculto" VALUE="form12">
```

```
</FORM>
```



Obs.:

- Mesmo que o valor especificado em HIDDEN VALUE não seja exibido no formulário, o usuário pode vê-lo através da operação de visualização do fonte do hiperdocumento, da mesma forma como no campo PASSWORD.

d. Entrada de várias linhas de texto - TEXTAREA

A marcação **TEXTAREA** não utiliza o formato convencional `INPUT TYPE="text"` dos exemplos anteriores. Ao contrário, uma marcação `<TEXTAREA>` delimita o seu início e a marcação `</TEXTAREA>` o seu fim.

Opções:

ROWS="" OBRIGATÓRIO

Especifica o número de linhas da entrada textual.

COLS="" OBRIGATÓRIO

Especifica o número de colunas da entrada textual.

Texto default OPCIONAL

Se você deseja que um texto seja exibido no campo textual ao abrir o formulário, simplesmente coloque este texto entre as marcações de início e fim da TEXTAREA.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

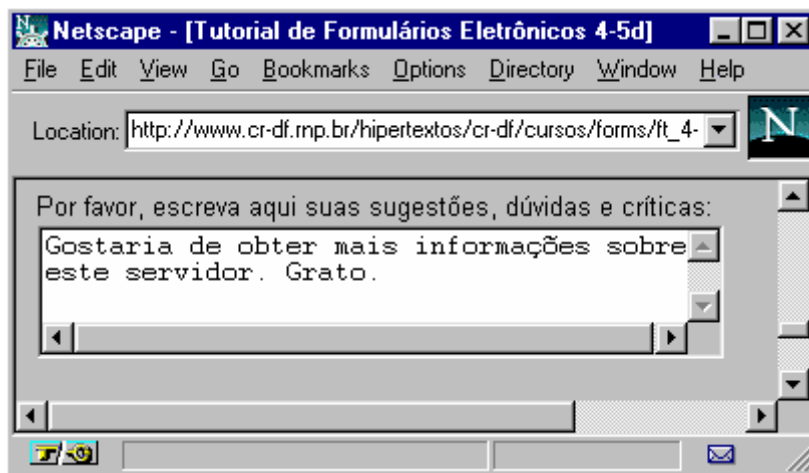
```
Por favor, escreva aqui suas sugestões, dúvidas e críticas:<BR>
```

```
<TEXTAREA NAME="critica" ROWS="3" COLS="40">
```

```
Gostaria de obter mais informações sobre  
este servidor. Grato.
```

```
</TEXTAREA>
```

```
</FORM>
```



Obs.:

- O usuário dispõe de Scrollbars para digitar além do número de linhas e colunas definidas em ROWS e COLS.

IV. Menus

a. Menus com opções - SELECT

b. Listas pagináveis - SELECT com SIZE

a. Menus com opções - SELECT

Menu de opção única

A marcação **SELECT** segue a mesma convenção de TEXTAREA. Ou seja, as opções de menu ficam entre a marcação de início **<SELECT>** e a de fim **</SELECT>**.

Opções:

OPTION OBRIGATÓRIO

Especifica uma opção presente no menu.

VALUE="" OPCIONAL

Especifica o valor da opção retornada ao servidor. Se não for definido, o nome da opção é enviado ao servidor.

SELECTED OPCIONAL

Por default, a primeira OPTION é exibida no menu. Esta marcação estabelece uma opção de menu a ser exibida inicialmente, quando não se deseja que seja a primeira OPTION.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Onde você pretende fazer turismo nas férias?

```
<SELECT NAME="lugares_para_ver">
```

```
<OPTION>Fortaleza
```

```
<OPTION VALUE="sul">Florianópolis ou Porto Alegre
```

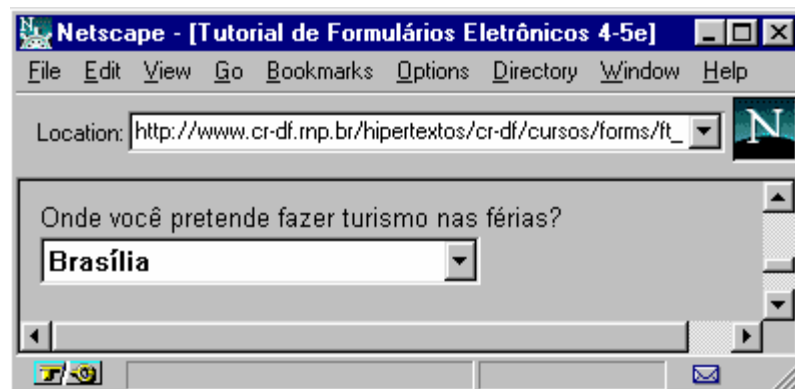
```
<OPTION>Rio de Janeiro
```

```
<OPTION SELECTED>Brasília
```

```
<OPTION VALUE="amazonia">Manaus
```

```
</SELECT>
```

```
</FORM>
```



Obs.:

- Se você possui mais do que 3 ou 4 opções de escolha, e o usuário só pode selecionar uma, então este elemento de entrada é o melhor. As pessoas frequentemente utilizam vários RADIOBUTTONS, que veremos a seguir.
- O script decodificador, no servidor, provavelmente apreciará uma única palavra para VALUE, em vez de múltiplas palavras. Você sempre pode utilizar sublinhado para separar expressões compostas.

b. Listas pagináveis - SELECT com SIZE

A única diferença entre este elemento de entrada e o anterior **SELECT** é a introdução da opção **SIZE**. Ela especifica quantas linhas com opções de menu serão exibidas na janela.

Opções:

MULTIPLE OPCIONAL

Especifica que múltiplas opções podem ser selecionadas, em oposição ao exemplo anterior do SELECT, onde apenas uma opção pode ser selecionada no menu.

SIZE="" OBRIGATÓRIO

Número de linhas (opções de menu) exibidas na janela.

OPTION OBRIGATÓRIO

Especifica uma opção da lista.

VALUE="" OPCIONAL

Especifica o valor da opção retornada ao servidor.

Se não for definido, o nome da opção é enviado ao servidor.

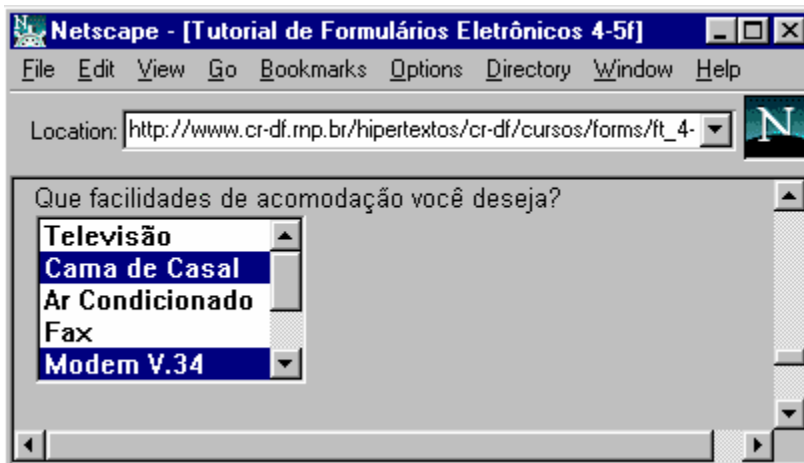
SELECTED OPCIONAL

Esta opção determina uma OPTION default para ser selecionada.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

```
Que facilidades de acomodação você deseja?<BR>
<SELECT NAME="opcional" MULTIPLE SIZE="5">
<OPTION>Televisão
<OPTION VALUE="casal" SELECTED>Cama de Casal
<OPTION>Ar Condicionado
<OPTION>Fax
<OPTION SELECTED>Modem V.34
<OPTION>Sauna
<OPTION>Lavanderia
<OPTION>Frigido-Bar
</SELECT>
```

```
</FORM>
```



Obs.:

- Este elemento de entrada é bom para listas com muitas opções, porque é possível controlar quantas serão exibidas por vez.
- Em alguns clientes WWW, é necessário teclar simultaneamente as teclas CONTROL ou SHIFT para selecionar múltiplos itens.

V. Botões

- a. Botões sim ou não - CHECKBOX
- b. Botões com opções - RADIO
- c. Botões de submissão e limpeza

a. Botões sim ou não - CHECKBOX

Retornando ao formato de **INPUT TYPE=""**, a marcação **CHECKBOXES** é perfeita para escolher entre duas opções.

Opções:

VALUE="" OPCIONAL

Especifica o valor da opção enviado ao servidor. Se não for definido, o valor "on" é enviado ao script decodificador.

CHECKED OPCIONAL

Esta marcação define a opção selecionada por default.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

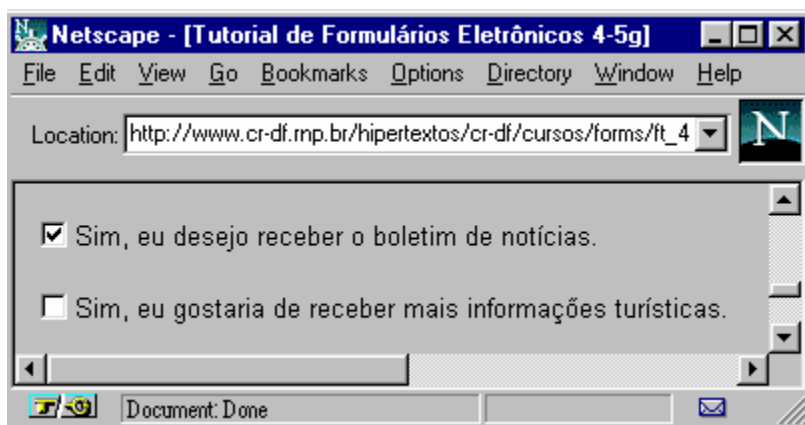
```
<INPUT TYPE="checkbox" NAME="boletim" CHECKED>Sim, eu desejo receber o boletim de notícias.
```

```
<P>
```

```
<INPUT TYPE="checkbox" NAME="info">Sim, eu gostaria de receber mais informações turísticas.
```

```
</DL>
```

```
</FORM>
```



Obs.:

- Este elemento funciona melhor para respostas do tipo "sim/não" ou "on/off".
- Você pode simular uma lista SELECT MULTIPLE criando CHECKBOXES com o mesmo NAME, mas diferentes VALUES

b. Botões com opções - RADIO

RADIOBUTTONS utilizam uma notação similar a de CHECKBOXES, contudo, apenas uma opção pode ser escolhida.

Opções:

VALUE="" OBRIGATÓRIO

Especifica o valor da opção a ser enviado para o servidor.

Se não for definido, um valor "on" é enviado ao script decodificador.

CHECKED RECOMENDADO

Esta marcação especifica qual botão estará selecionado por default.

Uma vez que uma seleção precisa ser feita, é melhor prover uma opção pre-selecionada.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Suas preferências na viagem:

```
<DL>
```

```
<DD>Classe do Bilhete:
```

```
<INPUT TYPE="radio" NAME="classe" VALUE="eco">econômica
```

```
<INPUT TYPE="radio" NAME="classe" VALUE="exe">executiva
```

```
<INPUT TYPE="radio" NAME="classe" VALUE="pri" CHECKED>primeira
```

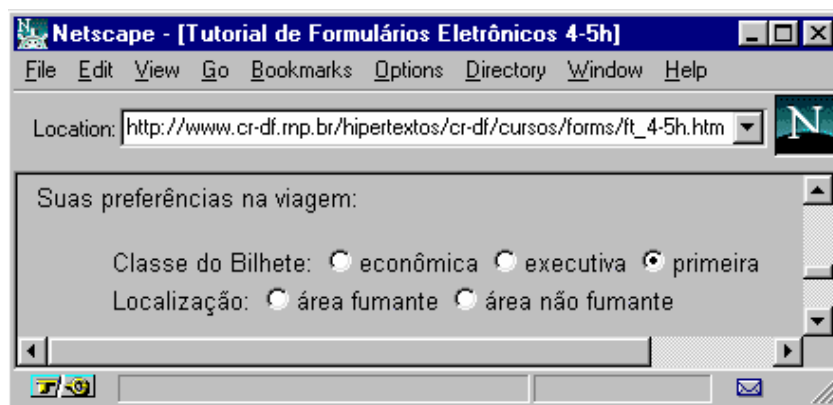
```
<DD>Localização:
```

```
<INPUT TYPE="radio" NAME="local" VALUE="frente">área fumante
```

```
<INPUT TYPE="radio" NAME="local" VALUE="tras">área não fumante
```

```
</DL>
```

```
</FORM>
```



Obs.:

- Como já foi mencionado, RADIOBUTTONS não são boa escolha para listas com muitos itens, porque o cliente tem problemas para exibir muitos botões. É melhor utilizar o menu SELECT.
- Uma vez que um botão é selecionado, ele não pode ser desmarcado sem selecionar outro botão que possua o mesmo NAME. A seleção default pode ser restaurada com a utilização do botão RESET, introduzido a seguir.
- Se você utiliza vários RADIOBUTTONS com o mesmo NAME e nenhum VALUES, o servidor não será capaz de dizer que botão foi selecionado, uma vez que o valor "on" será retornada para qualquer um deles.

c. Botões de submissão e limpeza - RESET E SUBMIT

Em vez do usuário corrigir cada INPUT, um botão **RESET** pode ser utilizado para restaurar todos os campos a seus valores default, como se nenhuma informação houvesse sido digitada.

E finalmente, o FORM precisa de uma opção para enviar toda a informação digitada para o servidor, uma vez que o usuário terminou de preencher todos os campos de entrada. O botão **SUBMIT** transfere toda a informação para a URL especificada no elemento ACTION.

Opções:

VALUE="" OPCIONAL

Especifica o texto a ser exibido no botão.

Se não for especificado, os textos default "Reset" e "Submit Query" serão colocados nos botões RESET e SUBMIT, respectivamente.

NAME="" OPCIONAL

Se NAME for definido em um botão SUBMIT, o formulário irá transmitir o valor do conteúdo do elemento VALUE, permitindo que você tenha múltiplos botões SUBMIT numa espécie de versão simplificada de um RADIOBUTTONS.

```
<FORM METHOD="POST" ACTION="/cgi-bin/donothing">
```

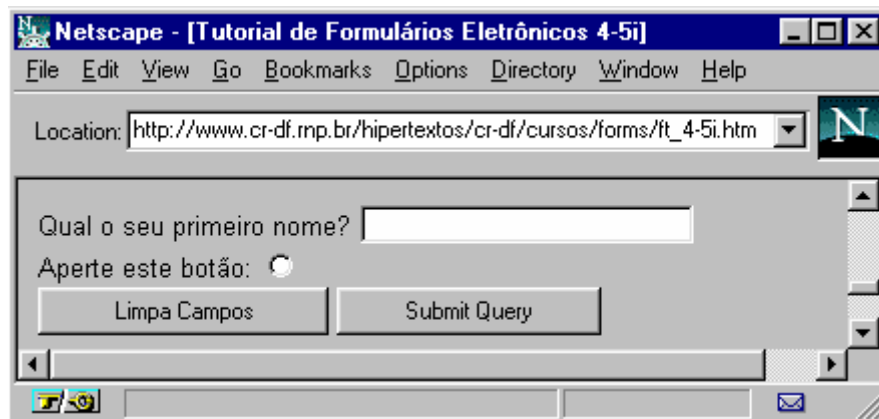
```
Qual o seu primeiro nome? <INPUT TYPE="text"><BR>
```

```
Aperte este botão: <INPUT TYPE="radio" NAME="nada"><BR>
```

```
<INPUT TYPE="reset" VALUE="Limpa campos">
```

```
<INPUT TYPE="submit">
```

```
</FORM>
```



Obs.:

- Normalmente, o script decodificador retorna um hiperdocumento após ele processar a informação do formulário.
- Nem todos os clientes suportam múltiplos botões SUBMIT. É melhor utilizar apenas um único botão SUBMIT para transmitir a informação digitada no formulário.