

Formulários em HTML

Disciplina: Desenvolvimento Web

Prof. Dr. Rafael Will M. de Araujo



Sumário

- Construir um formulário no HTML
- Como funciona a coleta de dados do usuário no HTML
- Tipos diferentes de campos
- Usar **GET** e **POST** no formulário
- Semântica ao construir um formulário

Conteúdo

1 Formulários HTML

2 Campos em formulários (inputs)

3 Envio de formulários

4 Exercício

Coleta de dados

- Por enquanto, a única informação que conseguimos coletar do nosso usuário é o recurso que ele deseja alcançar (via URL);
- Como podemos fazer para coletar outras informações, como por exemplo, um cadastro de cliente em uma loja?
- Para isso o HTML possui um elemento chamado de **formulário**;
- Representado pela *tag* `<form> </form>`. Esta *tag* se comporta como uma DIV visualmente.

Formulário

- Um formulário é composto basicamente de campos de coleta de dados (chamadas **inputs**) e de etiquetas desses campos (chamadas **labels**).

Exemplo de Formulário

```
1 <form>
2   Primeiro Nome:<br>
3   <input type="text" name="primeiroNome" id="primeiroNome"><br>
4   Último Nome:<br>
5   <input type="text" name="ultimoNome" id="ultimoNome"><br>
6   <input type="submit" value="Enviar">
7 </form>
```

- Note que as tags `<html>` `</html>`, `<head>` `</head>` e `<body>` `</body>` foram omitidas por simplicidade. Entretanto, elas devem estar presentes em qualquer página HTML.

Conteúdo

1 Formulários HTML

2 Campos em formulários (inputs)

3 Envio de formulários

4 Exercício

Formulários: campos (*inputs*)

- O HTML apresentou vários novos campos a serem usados. Vamos ver primeiro os principais a serem utilizados;
- Quase todos os campos de entrada de dados no HTML são identificados pela *tag* `<input>`. Esse marcador não possui *tag* de fechamento:
 - ▷ `<input name="nome">`
- Os campos do HTML podem ser separados em:
 - ▷ Coleta de texto (com exceção do tipo arquivo);
 - ▷ Coleta de opções;
 - ▷ Botões de controle.

Formulários: propriedades dos campos

- Os inputs possuem vários atributos, alguns que dependem do tipo inclusive. Vamos apresentar os mais importantes;
- São dois atributos obrigatórios:
 - ▷ **type**: indica o tipo do atributo (Ex: *text* ou *checkbox*)
 - ▷ **name**: indica o nome do atributo que o conteúdo do *input* irá enviar. Por exemplo, no campo com nome *primeiroNome*, se digitarmos **Rafael** e enviarmos o formulário, o corpo da requisição vai como **primeiroNome=Rafael**.

Outras propriedades

- Outras propriedades:
 - ▷ **value**: indica um valor inicial ao campo, ou a uma seleção;
 - ▷ **readonly**: indica que o campo está como leitura apenas;
 - ▷ **disabled**: indica que o campo não pode ser usado/enviado;
 - ▷ **size**: indica o tamanho do campo em caracteres (texto);
 - ▷ **maxlength**: indica o limite de caracteres que o campo permite (texto);
 - ▷ **min** e **max**: indica os valores máximo e mínimo do campo (número e data)
 - ▷ **multiple**: indica que pode escolher mais de um (seleção);
 - ▷ **placeholder**: coloca um texto explicativo dentro do campo;
 - ▷ **required**: indica que o campo é obrigatório (validação);

Campo de texto simples (1 linha)

- Tipo padrão de campo, feito para coletar textos de uma única linha. Quando o seu tipo (atributo **type**) não é declarado, ele terá o valor padrão **text**;
- Existem variações do campo de texto simples para diversos tipos de dados. A maneira de definir no HTML é a mesma (trocando apenas o atributo **type**);

Exemplo de campos de texto simples

```
1 <form>
2   Usuário: <br>
3   <input type="text" name="usuario" id="usuario"> <br>
4
5   Senha: <br>
6   <input type="password" name="senha" id="senha"> <br>
7 </form>
```

Rótulos (*labels*)

- É comum termos perto de cada campo um texto (rótulo) com seu significado (Ex: a palavra "E-mail" perto do campo de e-mail);
- Palavra soltas perto de campos não são automaticamente associados a eles, para que isso ocorra vamos usar a *tag label*;
- Essa *tag label* serve apenas para conectar um texto de significado para um *input* qualquer (e apenas *input*);
- Para fazer essa conexão, associamos o *input* ao seu *label* específico através do atributo **for** do *label*. Para isso, o valor do atributo **for** deve ser igual ao **id** do *input*;
 - ▷ O atributo **id** serve para criar um identificador único para um elemento HTML naquela página.

Exemplo de campos de texto simples com labels

```
1 <form>
2   <label for="usuario">Usuário</label>: <br>
3   <input type="text" name="usuario" id="usuario"> <br>
4
5   <label for="senha">Senha</label>: <br>
6   <input type="password" name="senha" id="senha"> <br>
7 </form>
```

Rótulos (*labels*)

- É comum termos perto de cada campo um texto (rótulo) com seu significado (Ex: a palavra "E-mail" perto do campo de e-mail);
- Palavra soltas perto de campos não são automaticamente associados a eles, para que isso ocorra vamos usar a *tag label*;
- Essa *tag label* serve apenas para conectar um texto de significado para um *input* qualquer (e apenas *input*);
- Para fazer essa conexão, associamos o *input* ao seu *label* específico através do atributo **for** do *label*. Para isso, o valor do atributo **for** deve ser igual ao **id** do *input*;
 - ▷ O atributo **id** serve para criar um identificador único para um elemento HTML naquela página.

Exemplo de campos de texto simples com labels

```
1 <form>
2   <label for="usuario">Usuário</label>: <br>
3   <input type="text" name="usuario" id="usuario"> <br>
4
5   <label for="senha">Senha</label>: <br>
6   <input type="password" name="senha" id="senha"> <br>
7 </form>
```

- Outro efeito do *label* associado a um campo é que ao clicar no *label*, ele **passa o foco para o campo associado**, facilitando a **acessibilidade** do seu site (principalmente para *checkboxes* e *radios*, que veremos nos próximos slides).

Campo de texto simples (1 linha)

- Com o HTML 5 foram introduzidos novos tipos de campos. Eles ajudam na validação automática do navegador, além de alguns terem *plugins* nativos implementados, como o campo *datetime*;
- Alguns desses novos campos são:
 - ▷ **email**: campo texto que valida um e-mail;
 - ▷ **number**: campo que aceita apenas números;
 - ▷ **url**: campo texto que valida uma URL;
 - ▷ **datetime**: campo que valida uma data com hora;
 - ▷ **time**: campo que valida horas e minutos;
 - ▷ **color**: campo texto que valida uma cor (hexadecimal);

Exemplos

```
1 <form>
2   <label for="email">E-Mail</label>: <br>
3   <input type="email" name="email" id="email"> <br>
4
5   <label for="numero">Número</label>: <br>
6   <input type="number" name="numero" id="numero"> <br>
7
8   <label for="url">URL</label>: <br>
9   <input type="url" name="url" id="url"> <br>
10
11  <label for="datahora">Data e Hora</label>: <br>
12  <input type="datetime-local" name="datahora" id="datahora"> <br>
13
14  <label for="hora">Hora</label>: <br>
15  <input type="time" name="hora" id="hora"> <br>
16
17  <label for="cor">Cor</label>: <br>
18  <input type="color" name="cor" id="cor"> <br>
19 </form>
```

Outros campos novos

- Note que alguns campos foram adicionais mais recentemente e não são compatíveis com todos os navegadores;

Mais exemplos

```
1 <form>
2   <label for="mes">Mês</label>: <br>
3   <input type="month" name="mes" id="mes"> <br>
4
5   <label for="semana">Semana</label>: <br>
6   <input type="week" name="semana" id="semana"> <br>
7
8   <label for="busca">Busca</label>: <br>
9   <input type="search" name="busca" id="busca"> <br>
10  <label for="intervalo">Intervalo</label>: <br>
11  <input type="range" name="intervalo" id="intervalo"> <br>
12 </form>
```

Campo de texto de múltiplas linhas / caixa de texto (*textarea*)

- Essa *tag* define uma área de texto para serem escritas várias linhas de texto. Nesse caso, a *tag* possui abertura e fechamento:

Exemplo de caixa de texto

```
1 <forms>
2   <label for="mensagem">Mensagem</label>:<br>
3   <textarea id="mensagem" name="mensagem" rows="10" cols="30"></textarea>
4 </forms>
```

Caixa de seleção (*select*)

- Essa *tag* define uma caixa de seleção para o usuário. As opções são escritas diretamente dentro do elemento;
- A *tag* possui abertura e fechamento: `<select> </select>`;
- Dentro da *tag* acrescenta-se uma lista de opções que o usuário terá disponível. Cada opção tem que estar dentro da *tag* `<option></option>`;
- A *tag* **option** possui um atributo com o valor dela (*value*), caso seja diferente do texto da opção;
 - ▷ **Curiosidade:** em outras linguagens, as caixas de seleção são conhecidas também por: *combobox* ou *dropdownlist*

Exemplos de caixas de seleção

```
1 <form>
2 <label for="disciplina">Disciplina</label>:<br>
3 <select name="disciplina" id="disciplina">
4 <option>Desenvolvimento Web</option>
5 <option>Programação Orientada a Objetos</option>
6 <option>Banco de Dados</option>
7 </select>
8 <br>
9 <label for="estado">Estado</label>:<br>
10 <select name="estado" id="estado">
11 <option value=""> - Selecione - </option>
12 <option value="RJ">Rio de Janeiro</option>
13 <option value="SE">Sergipe</option>
14 <option value="SP">São Paulo</option>
15 </select> <br>
16 </form>
```


Caixa de marcação (*checkbox*)

- Ao invés de usar o *select*, o *checkbox* deixa todas as opções listadas na tela;
- O agrupamento de opções é feito pelo *name* dos *inputs*;
- O que é enviado para cada *checkbox* é o valor associado ao atributo *value*;
- Caso mais de um *checkbox* de um mesmo *name* seja selecionado, é enviado um vetor de valores para o servidor;
- O *checkbox* pode ser marcado e desmarcado pelo usuário;

Exemplos de caixas de marcação (*checkbox*)

```
1 <form>
2 <p>Disciplinas:</p>
3 <input type="checkbox" id="disciplina-1" name="disciplinas" value="POO">
4 <label for="disciplina-1">Programação Orientada a Objetos</label><br>
5 <input type="checkbox" id="disciplina-2" name="disciplinas" value="DevWeb">
6 <label for="disciplina-2">Desenvolvimento Web</label><br>
7 <input type="checkbox" id="disciplina-3" name="disciplinas" value="BD">
8 <label for="disciplina-3">Banco de Dados</label><br>
9 <br>
10 <p>Linguagens:</p>
11 <input type="checkbox" id="linguagem-1" name="linguagens" value="python">
12 <label for="linguagem-1">Python</label><br>
13 <input type="checkbox" id="linguagem-2" name="linguagens" value="java">
14 <label for="linguagem-2">Java</label><br>
15 <input type="checkbox" id="linguagem-3" name="linguagens" value="ruby">
16 <label for="linguagem-3">Ruby</label><br>
17 <br>
18 <label for="deseja">Deseja receber novidades das linguagens selecionadas?</label>
19 <input type="checkbox" id="deseja" name="deseja" value="S">
20 </form>
```

Caixa de marcação (*checkbox*)

- Podemos utilizar o atributo **checked** (sem valor associado a ele), para que um *checkbox* já venha marcado por padrão.

Exemplos de caixas de marcação (*checkbox*)

```
1 <form>
2 <p>Disciplinas:</p>
3 <input type="checkbox" id="disciplina-1" name="disciplinas" value="POO">
4 <label for="disciplina-1">Programação Orientada a Objetos</label><br>
5 <input type="checkbox" id="disciplina-2" name="disciplinas" value="DevWeb" checked>
6 <label for="disciplina-2">Desenvolvimento Web</label><br>
7 <input type="checkbox" id="disciplina-3" name="disciplinas" value="BD">
8 <label for="disciplina-3">Banco de Dados</label><br>
9 <br>
10 <p>Linguagens:</p>
11 <input type="checkbox" id="linguagem-1" name="linguagens" value="python" checked>
12 <label for="linguagem-1">Python</label><br>
13 <input type="checkbox" id="linguagem-2" name="linguagens" value="java" checked>
14 <label for="linguagem-2">Java</label><br>
15 <input type="checkbox" id="linguagem-3" name="linguagens" value="ruby">
16 <label for="linguagem-3">Ruby</label><br>
17 <br>
18 </form>
```

Caixa de opções (*radio*) - "botões de rádio"

- Funciona de maneira similar ao *checkbox*, mostrando todas as opções, com a diferença de que **apenas uma delas pode ser selecionada** (no mesmo grupo);
- O agrupamento de opções é feito pelo *name* dos *inputs*;
- O que é enviado para cada *radio* é o valor associado ao atributo *value* (o padrão é *on*);
- Um *radio* não permite ser desmarcado. A única maneira de desmarcá-lo é selecionando outro *radio* dentro do mesmo grupo.
 - ▷ Atributo **checked**: é opcional. Usado para marcar qual *radio* deve ser marcado por padrão.

Exemplo de caixas de opções (*radio*)

```
1 <form>
2   <p>Tipos de Chat:</p>
3   <input type="radio" id="chat-1" name="chat" value="ZAP">
4   <label for="chat-1">Whatsapp</label><br>
5   <input type="radio" id="chat-2" name="chat" value="ICQ">
6   <label for="chat-2">I Seek You</label><br>
7   <input type="radio" id="chat-3" name="chat" value="IRC">
8   <label for="chat-3">Internet Relay Chat</label><br>
9   <br>
10  <p>Linguagens:</p>
11  <input type="radio" id="linguagem-1" name="linguagem" value="python">
12  <label for="linguagem-1">Python</label><br>
13  <input type="radio" id="linguagem-2" name="linguagem" value="java" checked>
14  <label for="linguagem-2">Java</label><br>
15  <input type="radio" id="linguagem-3" name="linguagem" value="ruby">
16  <label for="linguagem-3">Ruby</label><br>
17 </form>
```

Botões

- Para executar ações dentro do formulários, podemos criar botões com funcionalidades específicas;
- Botões podem ser escritos como uma variação da *tag* **input**, mas mais comumente são escritos com suas próprias *tags* **button**;
- Temos três tipos (*types*) de botões nativos no HTML:
 - ▷ **button**: botão simples;
 - ▷ **submit**: botão de submissão;
 - ▷ **reset**: botão de limpeza;

Botões

- Botões do tipo **button** não possuem função nativa no HTML. São usados mais para executar funções JavaScript;
- Botões do tipo **submit** são utilizados para enviar os dados dentro do formulário para o servidor. Para funcionar, esse botão **deve estar dentro de um formulário**;
- Botões do tipo **reset** são utilizados para **colocar o valor padrão** nos campos (ou no caso dos *radios* e *checkboxes* o estado padrão). Assim como o *submit*, ele só funciona se estiver dentro de um formulário.

Exemplo de botões

```
1 <form>
2   <label for="nome">Nome</label>:<br>
3   <input type="text" id="nome" name="nome" placeholder="Digite seu nome..." required><br>
4   <label for="email">E-mail</label>:<br>
5   <input type="email" id="email" name="email" value="email@teste.com" required><br>
6   <br>
7   <p>Botões com Input</p>
8   <input type="button" value="Botão tipo button">
9   <input type="submit" value="Botão tipo submit">
10  <input type="reset" value="Botão tipo reset">
11  <br>
12  <p>Botões com Button</p>
13  <button type="button">Botão do tipo button</button>
14  <button type="submit">Botão do tipo submit</button>
15  <button type="reset">Botão do tipo reset</button>
16 </form>
```

Conteúdo

- 1 Formulários HTML
- 2 Campos em formulários (inputs)
- 3 Envio de formulários**
- 4 Exercício

Formulário

- Além dos campos de texto e marcação, um formulário deve ter um **botão de submissão** para que possa ser enviado ao servidor;
- Para usarmos o **form** precisamos definir as seguintes propriedades:
 - ▷ **action**: para onde o formulário deve enviar os dados;
 - ▷ **method**: como o formulário deve enviar os dados;
 - ▷ **name**: um nome para identificar o formulário no Javascript.

Formulários: atributo *action*

- O atributo **action** deve conter a URL do servidor que irá receber esses dados para processamento. Pode ser URL absoluta (http://SERVIDOR/novoCliente) ou apenas relativa se for o mesmo servidor (/novoCliente):
 - ▷ Dê preferência a usar a URL relativa. (Por que?)

Exemplo de Formulário

```
1 <form action="novoCliente">
2   <label for="primeiroNome">Primeiro Nome</label>:<br>
3   <input type="text" name="primeiroNome" id="primeiroNome"><br>
4
5   <label for="ultimoNome">Último Nome</label>:<br>
6   <input type="text" name="ultimoNome" id="ultimoNome"><br>
7   <input type="submit" value="Enviar">
  </form>
```


Formulários: atributo *method*

- Define o método HTTP utilizado para enviar os dados. Os valores possíveis são GET e POST, sendo o POST o mais adequado;

Exemplo de Formulário

```
1 <form action="novoCliente" method="POST">
2   <label for="primeiroNome">Primeiro Nome</label>:<br>
3   <input type="text" name="primeiroNome" id="primeiroNome"><br>
4
5   <label for="ultimoNome">Último Nome</label>:<br>
6   <input type="text" name="ultimoNome" id="ultimoNome"><br>
7
8   <input type="submit" value="Enviar">
9 </form>
```

Diferença entre os métodos GET e POST

- No formulário podemos usar tanto o método GET quanto o POST;
- Ambos vão enviar as mesmas informações ao servidor, que as entenderá de uma mesma maneira;
- Como o GET colocar todos os dados coletados diretamente na URL, não é recomendado utilizar o GET quando essas informações são muito numerosas ou **sensíveis** (como senhas, número do cartão de crédito, etc);
- O POST por sua vez, codifica a informação e coloca ela diretamente no corpo da requisição. Isso deixa um pouco mais segura a transmissão, além de não haver limite do tamanho da informação a ser transmitida.

Conteúdo

- 1 Formulários HTML
- 2 Campos em formulários (inputs)
- 3 Envio de formulários
- 4 Exercício

Exercício

- Edite o exercício com a página da empresa fictícia *Cursinho Web*, para que ela possua uma página para contato, com o nome *contato.html*.
- Utilize os seguintes *names* para os campos:
 - ▷ Nome completo (type text): **nome**
 - ▷ E-mail (type email): **email**
 - ▷ Disciplina que mais interessa (select): **disciplina**
 - ▷ Linguagens que a pessoa conhece (checkboxes): **linguagens**
 - ◇ C#
 - ◇ Java
 - ◇ JavaScript
 - ◇ Python
 - ◇ PHP
 - ◇ Ruby
 - ▷ Onde conheceu (radio): **conheceu**

Exercício: *contato.html*

Cursinho Web - Contato x +

← → ↻ ⓘ File | /home/rwill/tmpfsDisk/curso/contato.html

[Página inicial](#) | [Disciplinas oferecidas](#) | [Entre em contato](#)

Dados para contato:

Nome completo:

E-mail:

Qual disciplina te interessa mais?

Desenvolvimento Web ▼

Quais linguagens você conhece?

☐ C#

☐ Java

☐ JavaScript

☐ Python

☐ PHP

☐ Ruby

Onde você conheceu o Cursinho Web?

☐ Amigos

☐ Google

☐ Youtube

☐ Instagram

Limpar

Enviar

Cursinho Web - Todos os direitos reservados
Contato: email@provedor.com