

Formulários e Ajax

ESTUDAR O USO E MANIPULAÇÃO DE FORMULÁRIOS COM AJAX E JSON

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR

Formulários

Em ambientes web a elaboração e utilização de formulários é praticamente inevitável. Aos usuários, preencher dados ou recuperar informações para leitura ou alteração é uma ação trivial e bastante comum.

No entanto, existem diversos tipos de formulários e maneiras de se capturar dados dos usuários. Algumas podem ser divertidas, outras uma verdadeira obra de paciência.

Quando um programador for desenvolver um formulário, via navegador, leve em consideração além das questões técnicas, questões de usabilidade, navegabilidade, layout e praticidade.

Quando um usuário encontra um formulário sem praticidade, ou seja, de difícil preenchimento, normalmente ele desiste de concluir o cadastro, ou só finaliza se for realmente importante.

Vamos verificar alguns exemplos de formulários em navegadores web.

No exemplo a seguir temos um formulário básico, com framework css bootstrap.

[Início](#) / [Fale conosco](#)

Formulário de Contato

Assunto	<input type="text" value="Informe sua senha"/>
Nome	<input type="text" value="Informe seu nome"/>
E-mail	<input type="text" value="Informe seu e-mail"/>
Mensagem	<div><input type="text" value="Digite sua mensagem"/> <input type="button" value="Enviar mensagem"/></div>

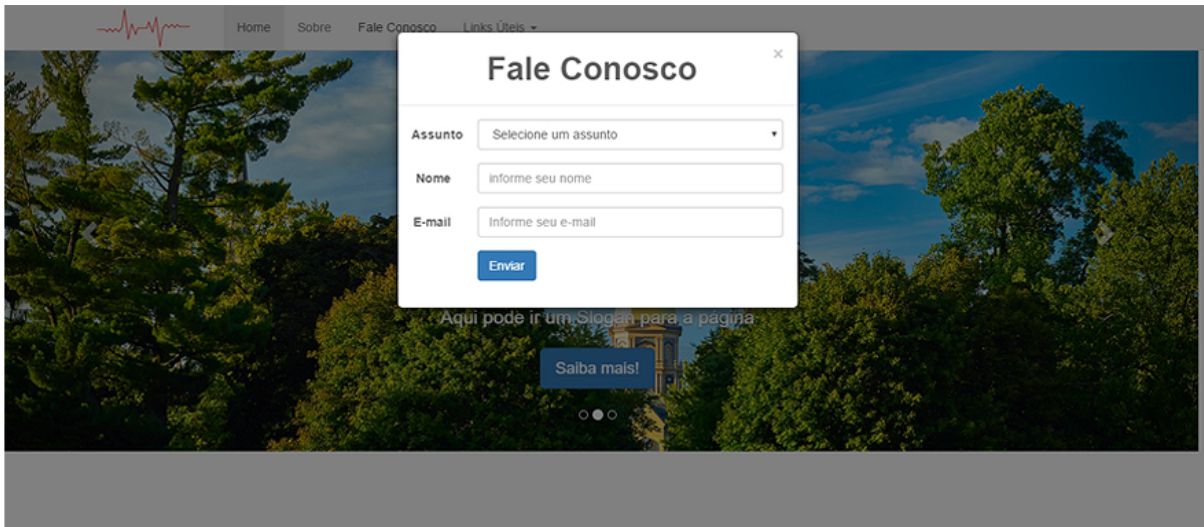
Legenda: EXEMPLO DE FORMULÁRIO WEB

Mais um exemplo de formulário com framework css bootstrap:



Legenda: EXEMPLO DE FORMULÁRIO WEB

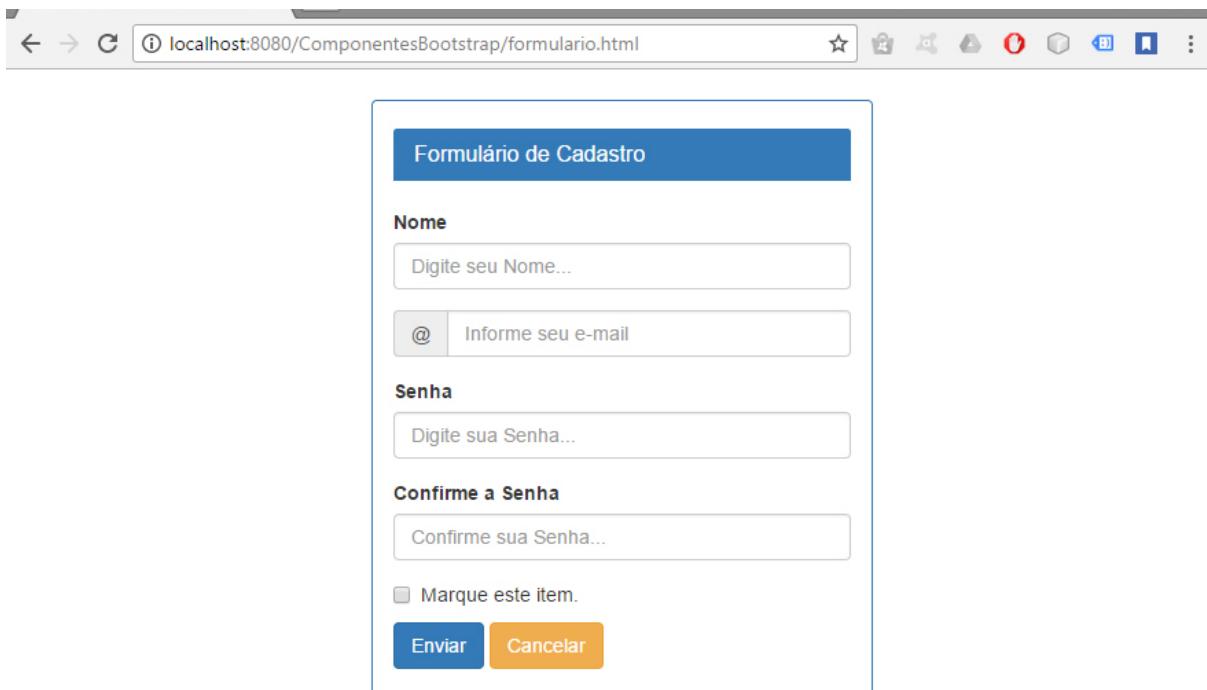
Um formulário no estilo modal, com framework css bootstrap:



The image shows a web page with a background image of trees. A modal form titled "Fale Conosco" is centered on the screen. The form has a close button (X) in the top right corner. It contains the following fields: "Assunto" (a dropdown menu with "Selecione um assunto"), "Nome" (a text input with placeholder "Informe seu nome"), and "E-mail" (a text input with placeholder "Informe seu e-mail"). Below these fields is a blue "Enviar" button. Below the modal, there is a blue button labeled "Saiba mais!" and a small text "Aqui pode ir um Slogan para a página".

Legenda: EXEMPLO DE FORMULÁRIO WEB

Outro exemplo de um formulário de cadastro, com framework css bootstrap:



The image shows a web browser window with the address bar displaying "localhost:8080/ComponentesBootstrap/formulario.html". The browser's toolbar includes back, forward, refresh, and search buttons, as well as a star icon for bookmarks. The main content area displays a registration form titled "Formulário de Cadastro". The form has a blue header bar with the title. It contains the following fields: "Nome" (a text input with placeholder "Digite seu Nome..."), "E-mail" (a text input with placeholder "Informe seu e-mail" and an "@" symbol on the left), "Senha" (a text input with placeholder "Digite sua Senha..."), and "Confirme a Senha" (a text input with placeholder "Confirme sua Senha..."). Below these fields is a checkbox labeled "Marque este item.". At the bottom of the form are two buttons: a blue "Enviar" button and an orange "Cancelar" button.

Legenda: EXEMPLO DE FORMULÁRIO WEB

Além da criação de formulários bem formatados, e com layout adequados, algumas tecnologias auxiliam na criação de uma boa interface. Neste sentido, algumas tecnologias podem auxiliar em muito o uso de formulários.

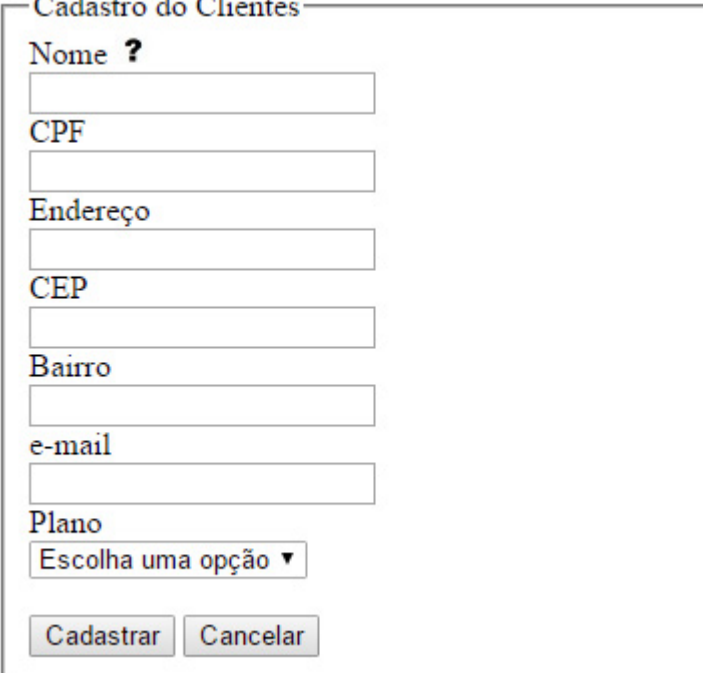
Vamos exemplificar o uso das tecnologias AJAX e JSON para este fim.

AJAX e JSON

O uso de AJAX é extremamente útil na recuperação de dados externos, e com ajuda do JavaScript e DOM (Document Object Model) a sua habilidade de atualizar os elementos HTML fica ampliada. Outra tecnologia que nos auxilia neste momento é a utilização do JSON, que permite consumir dados externos das mais diretas fontes.

Vamos criar um exemplo simples, sem estilos css definidos, para exemplificar o uso destas tecnologias.

Veja a imagem deste formulário:



Formulário de Cadastro do Cliente:

- Nome ?
- CPF
- Endereço
- CEP
- Bairro
- e-mail
- Plano
- Escolha uma opção ▼
- Cadastrar
- Cancelar

É um formulário simples, sem qualquer estilização css. Com a finalidade de capturar os dados preenchidos e enviar a um servidor.

Neste exemplo, não vamos abordar os códigos de captura e tratamento de dados no servidor.

Veja o código fonte do formulário acima:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 17</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div></div>
9.         <div>
10.             <form>
11.                 <fieldset>
12.                     <legend>Cadastro do Clientes</legend>
13.                     <div>
14.                         <label>Nome</label>
15.                         
16.                         <div>
17.                             <input id="idNome" name="idNome" type="text">
18.                         </div>
19.                     </div>
20.                     <div>
21.                         <label>CPF</label>
22.                         <div>
23.                             <input id="idCPF" name="idCPF" type="text">
24.                         </div>
25.                     </div>
26.                     <div>
27.                         <label>Endereço</label>
28.                         <div>
29.                             <input id="idEndereco" name="idEndereco" type="text">
30.                         </div>
31.                     </div>
32.                     <div>
33.                         <label>CEP</label>
34.                         <div>
35.                             <input id="idCEP" name="idCEP" type="text">
36.                         </div>
37.                     </div>
38.                     <div>
39.                         <label>Bairro</label>
40.                         <div>
```

```

41.         <input id="idBairro" name="idBairro" type="text" />
42.     </div>
43. </div>
44. <div>
45.     <label>e-mail</label>
46.     <div>
47.         <input id="idEmail" name="idEmail" type="email" />
48.     </div>
49. </div>
50. <div>
51.     <label>Plano</label>
52.     <div>
53.         <select id="idPlano" name="idPlano" class="form-control">
54.             <option value="">Escolha uma opção</option>
55.             <option value="1">Normal</option>
56.             <option value="2">Express</option>
57.             <option value="3">VIP</option>
58.         </select>
59.     </div>
60. </div>
61. <br>
62. <div>
63.     <label></label>
64.     <div>
65.         <button id="idConfirmar" name="idConfirmar" class="btn btn-primary">Confirmar</button>
66.         <button id="idCancelar" name="idCancelar" class="btn btn-danger">Cancelar</button>
67.     </div>
68. </div>
69. </fieldset>
70. </form>
71. </div>
72. <div></div>
73. <script>
74.     function procurarCliente() {
75.         var xmlhttp = new XMLHttpRequest();
76.         xmlhttp.onreadystatechange = function () {
77.             if (this.readyState == 4 && this.status == 200) {
78.                 var valor = "";
79.                 var cliente_encontrado = false;
80.                 myObj = JSON.parse(this.responseText);
81.                 nomeProcura = document.getElementById("idNome");

```

```

82.
83.         for (i in myObj.cliente) {
84.             if (myObj.cliente[i].nome == nomeProcura.value) {
85.                 document.getElementById("idCPF").value = r
86.                 document.getElementById("idEndereco").value = r
87.                 document.getElementById("idCEP").value = r
88.                 document.getElementById("idBairro").value = r
89.                 document.getElementById("idEmail").value = r
90.                 document.getElementById("idPlano").focus()
91.                 cliente_encontrado = true;
92.             }
93.         }
94.         if (cliente_encontrado == false){
95.             alert("Cliente não encontrado!");
96.         }
97.     }
98. };
99.     xmlhttp.open("GET", "dados_cliente.json", true);
100.    xmlhttp.send();
101.    }
102.    </script>
103.    </body>
104. </html>

```

Para o consumo de dados, vamos utilizar JSON, conforme já apresentado nos tópicos anteriores.

Veja o arquivo de exemplo em formato JSON:

```

1. {
2.   "cliente":[
3.     {
4.       "nome":"João",
5.       "cpf":"0123654",
6.       "endereco":"Av. Paulista, 1000",
7.       "cep":"01234-001",
8.       "bairro":"Vila Norte",
9.       "email":"joao@gmail.com"
10.    },
11.    {
12.      "nome":"Maria",
13.      "cpf":"01231111",
14.      "endereco":"Av. Nove de Julho, 1000",
15.      "cep":"01789-001",
16.      "bairro":"Vila Sul",
17.      "email":"maria@gmail.com"
18.    },
19.    {
20.      "nome":"Pedro",
21.      "cpf":"45612343",
22.      "endereco":"Av. Santo Amaro, 1000",
23.      "cep":"01321-001",
24.      "bairro":"Vila Leste",
25.      "email":"pedro@gmail.com"
26.    }
27.  ]
28. }

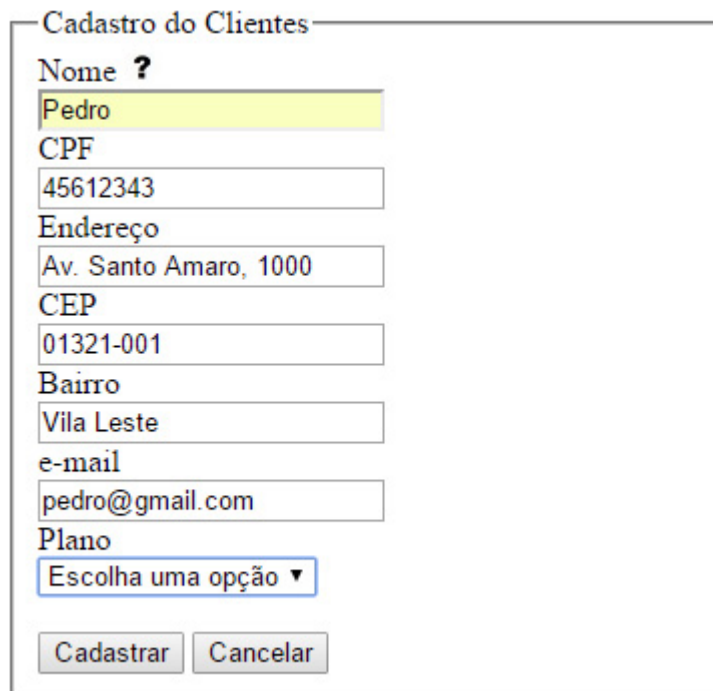
```

Observe a imagem de ponto de interrogação (?), ao lado do campo nome. Esta imagem possui um link, que acessa uma função JavaScript.

Esta função utiliza as tecnologias AJAX e JSON para as seguintes ações:

- Capturar o valor do campo nome.
- Estabelecer a conexão do arquivo ou fonte de dados indicado.
- Ler o arquivo ou fonte de dados.
- Procurar o valor no campo nome, informado no formulário, no arquivo ou fonte de dados especificado pelo JSON.
- Caso encontre um valor correspondente, obtém os demais dados do cliente e insere nos campos do formulário, sem a necessidade de efetuar um novo refresh, ou atualização do navegador.
- Caso não encontre, informa uma mensagem de erro.

Esta funcionalidade é simulada e apresentada na imagem a seguir:



O formulário, intitulado "Cadastro do Clientes", contém os seguintes campos e valores:

- Nome ?**: Campo com o valor "Pedro" destacado em amarelo.
- CPF**: Campo com o valor "45612343".
- Endereço**: Campo com o valor "Av. Santo Amaro, 1000".
- CEP**: Campo com o valor "01321-001".
- Bairro**: Campo com o valor "Vila Leste".
- e-mail**: Campo com o valor "pedro@gmail.com".
- Plano**: Campo de seleção com a opção "Escolha uma opção" selecionada.

Na base do formulário, há dois botões: "Cadastrar" e "Cancelar".

Verifique que apenas com o preenchimento de um nome, que exista na fonte de dados, e após o clique no link indicado por "?", os demais campos são preenchidos automaticamente.

Este recurso é muito útil, pois agiliza o processo de cadastro, e evita erros de digitação.

Este script ainda pode ser alterado para efetuar uma atualização ou exclusão de dados, em vez de realizar apenas um novo cadastro.

Vamos comentar um trecho deste código:

```
1. for (i in myObj.cliente) {
2.     if (myObj.cliente[i].nome == nomeProcura.value){
3.         document.getElementById("idCPF").value = myObj.cliente[i].
4.         document.getElementById("idEndereco").value = myObj.cliente[i].
5.         document.getElementById("idCEP").value = myObj.cliente[i].
6.         document.getElementById("idBairro").value = myObj.cliente[i].
7.         document.getElementById("idEmail").value = myObj.cliente[i].
8.         document.getElementById("idPlano").focus();
9.         cliente_encontrado = true;
10.    }
11. }
```


Na linha 2, o script compara o valor retornado, de cada ocorrência, da fonte JSON, com o valor informado no formulário, e se caso forem iguais, atualiza os demais campos do formulário.

Vamos agora atualizar este formulário com estilos css, com ajuda do framework bootstrap.

Observe a imagem a seguir:

Cadastro do Clientes

Nome	<input type="text" value="Nome do cliente"/>	?
CPF	<input type="text" value="Número do CPF"/>	
Endereço	<input type="text" value="Endereço residencial"/>	
CEP	<input type="text" value="CEP"/>	
Bairro	<input type="text" value="Bairro"/>	
e-mail	<input type="text" value="e-mail de contato"/>	
Plano	<input type="text" value="Escolha uma opção"/>	▼

Este é o mesmo formulário, agora estilizado.

Vamos avaliar como ficou o seu código fonte:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 17</title>
5.         <meta charset="UTF-8">
6.         <meta name="viewport" content="width=device-width, initial-scale=1">
7.         <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">
8.         <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
9.         <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>
10.     </head>
11.     <body>
12.         <br>
13.         <div class="col-md-3"></div>
14.         <div class="col-md-6">
15.             <form class="form-horizontal">
16.                 <fieldset>
17.                     <legend>Cadastro do Clientes</legend>
18.                     <div class="form-group">
19.                         <label class="col-md-2 control-label" for="idNome">Nome</label>
20.                         
21.                         <div class="col-md-9">
22.                             <input id="idNome" name="idNome" type="text" placeholder="Nome" />
23.                         </div>
24.                     </div>
25.                     <div class="form-group">
26.                         <label class="col-md-2 control-label" for="idCPF">CPF</label>
27.                         <div class="col-md-9">
28.                             <input id="idCPF" name="idCPF" type="text" placeholder="CPF" />
29.                         </div>
30.                     </div>
31.                     <div class="form-group">
32.                         <label class="col-md-2 control-label" for="idEndereco">Endereço</label>
33.                         <div class="col-md-9">
34.                             <input id="idEndereco" name="idEndereco" type="text" placeholder="Endereço" />
35.                         </div>
36.                     </div>
37.                     <div class="form-group">
38.                         <label class="col-md-2 control-label" for="idCEP">CEP</label>
39.                         <div class="col-md-9">
40.                             <input id="idCEP" name="idCEP" type="text" placeholder="CEP" />
```

```

41.         </div>
42.     </div>
43.     <div class="form-group">
44.         <label class="col-md-2 control-label" for="idBairro">
45.             <div class="col-md-9">
46.                 <input id="idBairro" name="idBairro" type="text">
47.             </div>
48.         </div>
49.     <div class="form-group">
50.         <label class="col-md-2 control-label" for="idEmail">
51.             <div class="col-md-9">
52.                 <input id="idEmail" name="idEmail" type="email">
53.             </div>
54.         </div>
55.     <div class="form-group">
56.         <label class="col-md-2 control-label" for="idPlano">
57.             <div class="col-md-9">
58.                 <select id="idPlano" name="idPlano" class="form-control">
59.                     <option value="">Escolha uma opção</option>
60.                     <option value="1">Normal</option>
61.                     <option value="2">Express</option>
62.                     <option value="3">VIP</option>
63.                 </select>
64.             </div>
65.         </div>
66.     <div class="form-group">
67.         <label class="col-md-2 control-label" for="idConfirmar">
68.             <div class="col-md-9">
69.                 <button type="submit" id="idConfirmar" name="idConfirmar">
70.                 <button type="reset" id="idCancelar" name="idCancelar">
71.             </div>
72.         </div>
73.     </fieldset>
74. </form>
75. </div>
76. <div class="col-md-3"></div>
77. <script>
78.     function procurarCliente() {
79.         var xmlhttp = new XMLHttpRequest();
80.         xmlhttp.onreadystatechange = function () {
81.             if (this.readyState == 4 && this.status == 200) {

```

```

82.         var valor = "";
83.         var cliente_encontrado = false;
84.         myObj = JSON.parse(this.responseText);
85.         nomeProcura = document.getElementById("idNome");
86.
87.         for (i in myObj.cliente) {
88.             if (myObj.cliente[i].nome == nomeProcura.value)
89.                 document.getElementById("idCPF").value = r
90.                 document.getElementById("idEndereco").value = r
91.                 document.getElementById("idCEP").value = r
92.                 document.getElementById("idBairro").value = r
93.                 document.getElementById("idEmail").value = r
94.                 document.getElementById("idPlano").focus();
95.                 cliente_encontrado = true;
96.             }
97.         }
98.         if (cliente_encontrado == false){
99.             alert("Cliente não encontrado!");
100.        }
101.    }
102.    };
103.    xmlhttp.open("GET", "dados_cliente.json", true);
104.    xmlhttp.send();
105.    }
106.    </script>
107.    </body>
108. </html>

```

Após o preenchimento do formulário, e posterior procura na fonte de dados, teremos o seguinte resultado:

Cadastro do Clientes

Nome	<input type="text" value="Pedro"/>	?
CPF	<input type="text" value="45612343"/>	
Endereço	<input type="text" value="Av. Santo Amaro, 1000"/>	
CEP	<input type="text" value="01321-001"/>	
Bairro	<input type="text" value="Vila Leste"/>	
e-mail	<input type="text" value="pedro@gmail.com"/>	
Plano	<input type="text" value="Escolha uma opção"/>	▼
<div><input type="button" value="Cadastrar"/> <input type="button" value="Cancelar"/></div>		

Neste caso, também conseguimos preencher todos os campos do formulário, sem a necessidade de realizar um novo refresh, ou atualização do navegador.

DICA:

Para saber mais sobre estes assuntos, acesse:

bootstrap

<http://getbootstrap.com.br/> (<http://getbootstrap.com.br/>)

<https://www.w3schools.com/bootstrap/> (<https://www.w3schools.com/bootstrap/>)

AJAX

https://www.w3schools.com/js/js_ajax_examples.asp

(https://www.w3schools.com/js/js_ajax_examples.asp)

https://www.w3schools.com/js/js_json_intro.asp (https://www.w3schools.com/js/js_json_intro.asp)

Usabilidade e acessibilidade

<https://tableless.com.br/o-que-e-usabilidade/> (<https://tableless.com.br/o-que-e-usabilidade/>)

<http://acessibilidadelegal.com/> (<http://acessibilidadelegal.com/>)

Recomendamos a criação de novos formulários, acessando novas fontes de dados. Um exemplo interessante a ser desenvolvido é a captura e atualização de dados de CEP, atualizando os dados existentes para um novo cadastro de cliente.

Validação de campo

É possível e útil validar campos antes de enviar os dados a algum serviço. Em alguns campos a validação é obrigatória.

Em nosso exemplo, não faz sentido iniciar a função de procurar cliente sem que o nome esteja digitado.

Vamos verificar esta funcionalidade.

Veja a seguir, os fragmentos de códigos que devem ser alterados:

```
1. 
```

Agora a imagem "pergunta.jpg" chamará a função "validarCampo".

?Nesta função faremos uma verificação se o campo nome possui algum caracter digitado, em nosso exemplo, se há mais de um caracter.

Veja o exemplo a seguir:

```
1. function validarCampo(){
2.     if (document.getElementById("idNome").value.length > 1){
3.         procurarCliente();
4.     }else{
5.         alert("O campo nome não pode ficar vazio.");
6.         document.getElementById("idNome").focus();
7.     }
8. }
```

Nesta função, a propriedade length do campo "idNome" informa o tamanho atual da string contida no campo. Caso não seja maior do que um, ou seja, esteja vazia, uma mensagem de erro é apresentada e o foco retorna ao campo. Caso exista algum valor neste campo, a script direciona a ação para a função "procurarCliente()".

Neste processo, é possível verificar uma etapa, uma forma de validação de campo.

Crie suas validações, aos demais campos. Experimente.

Conclusão

Os formulários podem obter muitos recursos interessantes e ágeis com ajuda do JavaScript. Uma destas vantagens é a atualização e consumo de dados sem a necessidade de realizar atualizações da janela do navegador, o que torna o processo mais rápido.

ATIVIDADE

Para preenchimento de um campo idCPF, com dados proveniente de uma fonte externa (JSON), indique a alternativa que realiza esta operação corretamente:

- A. `document.getElementById("idCPF").value = myObj;`
- B. `document.getElementById("idCPF").value = cpf;`
- C. `document.getElementById("idCPF").value = myObj.cliente[i].cpf;`
- D. `document.getElementById("idCPF") = myObj.cpf;`

ATIVIDADE

De acordo com o código a seguir, indique a alternativa que explique sua finalidade:

```
if (document.getElementById("idNome").value.length > 1){  
  // comandos  
}
```

- A. avalia se idNome é um numero
- B. avalia se idNome ?é impar
- C. avalia se idNome ?possui mais de um caracter
- D. avalia se idNome existe no formulário

ATIVIDADE

De acordo com o código a seguir, indique a alternativa que explique sua finalidade:

```
if (myObj.cliente[i].nome == nomeProcura.value){  
    // comandos  
}
```

- A. verifica se o campo cliente[i].nome e maior do que a variável nomeProcura.value
- B. verifica se o campo cliente[i].nome e igual a variável nomeProcura.value
- C. verifica se o campo cliente[i] e maior do que a variável nomeProcura
- D. verifica se o campo cliente[i].nome e menor do que a variável nomeProcura

REFERÊNCIA

- MORRISON, M. Use a cabeça JavaScript. 5º Ed. Rio de Janeiro: Alta Books, 2012. 606 p.
- OLIVIERO. C. A. J. Faça um site JavaScript orientado por projeto. 6º ed. São Paulo: Érica, 2010. 266 p.
- ZAKAS, Nicholas C. JavaScript de alto desempenho. 8º Ed. São Paulo: Novatec, 2010. 245 p.

