**Desenvolvimento de Formulário para a Web**

**Otávio A. A. Silva¹**

**¹ Faculdade de Engenharia da Computação e Telecomunicações - Instituto de Tecnologia - Universidade Federal do Pará**

**Cidade Universitária “Prof. José da Silveira Netto” – Campus II (Profissional) Belém/PA, Brasil, 66075-110**

tavioalves@gmail.com

***Abstract.*** *This meta-article aims to show a step by step to create a form for the web, with all the necessary steps for its establishment and to demonstrate the use of mask and validation using jQuery plugins and use the WebStorage to save user data in your browser.*

***Resumo.*** *Este meta-artigo tem como objetivo mostrar um passo a passo para a criação de um formulário para a web, apresentando todos os passos necessários para a sua criação, além de mostrar a utilização da validação e máscara através de plugins do jQuery e o uso do WebStorage para salvar os dados do usuário em seu navegador.*

**1. Introdução**

A utilização de formulários na Internet atualmente é sem dúvida alguma muito importante, já que é a partir dela que pode-se coletar informações sobre os usuários que utilizam alguma aplicação, como o cadastro de usuários, pesquisas de campo ou a confirmação de pagamento de uma loja online. Por tanto, o processo de desenvolvimento de um formulário deve ser feito minuciosamente, para que usuário possa inserir suas informações sem erros e com facilidade, veremos isso nas próximas seções.

**2. Desenvolvimento do formulário**

Para realizar o desenvolvimento do nosso formulário, vamos utilizar básicamente a tag <form> em nosso HTML, onde a tag <fieldset> irá possuir o título do nosso formulário o código do mesmo pode ser visto abaixo.

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>Formulário de Cadastro</**title**>

<**meta** http-equiv="Content-Type" content="text/html; charset=UTF-8">

</**head**>

<**body**>

<**form** id="formulario">

<**fieldset**>

<**legend**>Preencha os dados do formulário abaixo:</**legend**>

</**fieldset**>

</**form**>

</**body**>

</**html**>

Após a criação da área do nosso formulário, é necessário criar os itens do mesmo. Para a divisão de cada item iremos usar as tags <div>. Para a criação dos itens, utilizaremos as tag <label> para identificação do item pelo usuário e a tag <input> para o usuário inserir o conteúdo do item. Em nosso caso teremos um formulário de cadastro com o seguintes itens: Nome, e-mail, cpf, endereço, número, complemento, bairro, CEP, telefone e celular. Inserindo em nossa tag <input> o *id* e *name* de cada item, onde serão utilizados posteriormente pelo jQuery. Desta forma o nosso form passa a ter a seguinte cara.

<**form** id="formulario">

<**fieldset**>

<**legend**>Preencha os dados do formulário abaixo:</**legend**>

<**div**>

<**label**>Nome:</**label**>

<**input** name="nome" id="nome"/>

</**div**>

<**div**>

<**label**>E-Mail:</**label**>

<**input** name="email" id="email"/>

</**div**>

<**div**>

<**label**>CPF:</**label**>

<**input** name="cpf" id="cpf"/>

</**div**>

<**div**>

<**label**>Endereço:</**label**>

<**input** name="endereco" id="endereco"/>

</**div**>

<**div**>

<**label**>Número:</**label**>

<**input** name="numero" id="numero"/>

</**div**>

<**div**>

<**label**>Complemento:</**label**>

<**input** name="complemento" id="complemento"/>

</**div**>

<**div**>

<**label**>Bairro:</**label**>

<**input** name="bairro" id="bairro"/>

</**div**>

<**div**>

<**label**>CEP:</**label**>

<**input** name="cep" id="cep"/>

</**div**>

<**div**>

<**label**>Telefone:</**label**>

<**input** name="telefone" id="telefone"/>

</**div**>

<**div**>

<**label**>Celular:</**label**>

<**input** name="celular" id="celular"/>

</**div**>

</**fieldset**>

</**form**>

Com a criação dos itens do nosso formulário é necessário que exista uma forma de envia-lo ou salvar no servidor. Para isso vamos a tag <input>, porém terá como *class,* e *type* o valor *submit*. Bastando por tanto colocar uma nova <div> dentro da tad <fieldset> para realizar a inserção do nosso botão.

<**div**>

<**input** class="submit" onClick="enviar()" type="submit" value="Enviar"/>

</**div**>

Com a conclusão da criação do nosso formulário podemos perceber na figura 1, que o mesmo esta sem nenhum tipo de formatação, por tanto com um aspecto bastante desagradável para o usuário.

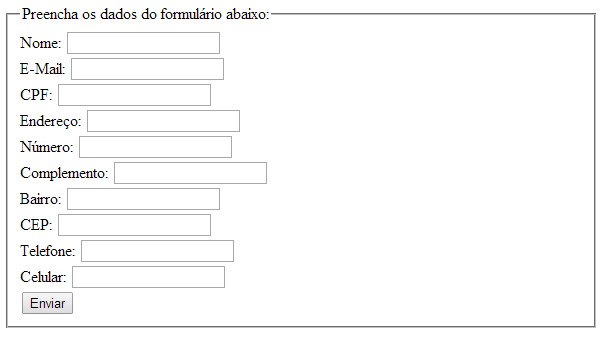


Fig. 1 – Formulário sem formatação

Para resolver esse problemas iremos utilizar os conhecimentos adquiridos nos trabalhos anteriores para a criação de um style que possa solucionar esse incomodo. Vamos mudar a fonte do nosso formulário, assim como fazer a justificação das *labels*, *inputs* e botão, criar a formatação das mensagens de erro caso exista algum item com informação incorreta ou sem a mesma. Criamos uma pasta css e criamos o arquivo estilo.css dentro da pasta. Onde o mesmo possuirá as seguintes informações.

\* { **font-family**: Verdana; **font-size**: 96%; }

**label** { **width**: 10em; **float**: left; }

**label**.error{**color**:red; **float**:none; **padding-left**: 10px;}

**div** { **clear**: both; }

**input**, **textarea** { **width**: 15em; }

.submit { **margin-left**: 10em; }

Necessitamos agora referenciar nosso estilo ao nosso HTML, fazemos isso com a inserção da linha de código abaixo em nosso <head>.

<**link** rel="stylesheet" href="css/estilo.css" type="text/css">

Com isso nosso formulário tem um aspecto bem melhor ao usuário, o mesmo pode ser visto na figura 2.

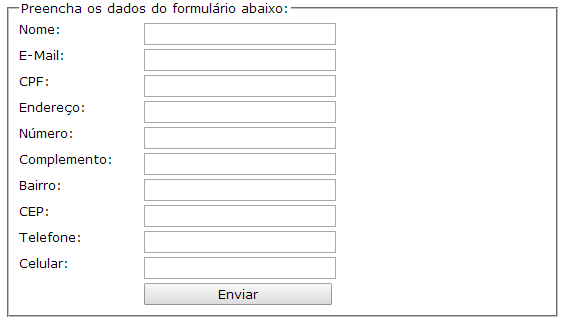


Fig. 2 – Formulário com formatação

**3. Inserção da validação e máscara**

Antes que o formulário possa ser enviado pelo usuário, é necessário que os dados inseridos ao menos tenham coerência. Ou seja, não existe número de telefone com letras ou um CPF que contenha letras. Por tanto é necessário que exista um script que trate as informações antes de serem enviadas.

Através do jQuery podemos fazer isso, onde o mesmo  é uma biblioteca JavaScript cross-browser desenvolvida para simplificar os scripts client side que interagem com o HTML. Usada por cerca de 77% dos 10 mil sites mais visitados do mundo, jQuery é a mais popular das bibliotecas JavaScript.

Para o desenvolvimento do nosso formulário iremos utilizar dois plugins do jQuery, o de validação e de máscara. Antes de tudo, criaremos uma pasta chamada js (Javascript) na pasta raiz do projeto, e iremos baixar as versões apropriadas do jquery e seus respectivos plugins, após o download mover os scripts para a pasta js do projeto. Feito isso, é necessário referência os scripts em nosso arquivo HTML dentro da tag <head>.

<**script src="js/jquery-1.7.2.min.js"** ></**script**>

<**script src="js/jquery.validate.min.js"**></**script**>

<**script src="js/jquery.maskedinput.min.js"**></**script**>

Feito isso necessitamos criar o script contendo as informações para validação, ou seja, as regras que serão tomadas para serem validadas, os itens que necessitam de validação e que mensagem será apresentada caso ocorra erro. Usaremos a tag <script> contendo o script abaixo.

<**script**>

$(document).ready(**function**()

{

$('#formulario').validate(

{

rules:{

nome:{

required: true

},

email:{

required: true,

email:true

},

cpf:{

required: true,

},

endereco:{

required: true

},

numero:{

required: true

},

complemento:{

required: true

},

bairro:{

required: true

},

cep:{

required: true

},

telefone:{

required: true

},

celular:{

required: true

}

},

messages:{

nome:{

required: "Campo Obrigatório",

},

email:{

required: "Campo Obrigatório",

email: "Digite um email válido"

},

cpf:{

required: "Campo Obrigatório",

},

endereco:{

required: "Campo Obrigatório",

},

numero:{

required: "Campo Obrigatório",

},

complemento:{

required: "Campo Obrigatório",

},

bairro:{

required: "Campo Obrigatório",

},

cep:{

required: "Campo Obrigatório",

},

telefone:{

required: "Campo Obrigatório",

},

celular:{

required: "Campo Obrigatório",

}

}

}

);

})

</**script**>

Básicamente, o script faz o seguinte, verifica se o documento esta pronto para ser lido, após isso faz a validação do mesmo contendo as regras que foram estabelecidas abaixo, ou seja, os inputs que possuem “nome”, “email” e etc em seus name são *required,*por tanto são necessários para a validação do formulário, e caso contenham algum erro ira ser apresentado a mensagem que foi estabelecida em *message*.

Agora necessitamos criar as máscaras do nosso formulário, onde as mesmas tem como função estabelecer o formato de algum campo. Por exemplo, para o campo CPF precisaremos de uma mascara no formato 999.999.999-99, onde o “9” representa os números, ou seja, esse campo só poderá conter números e contendo os 11 dígitos estabelecidos na máscara. Em nosso formulário, utilizaremos máscaras nos campos CPF, CEP, Telefone e Celular. Sendo que as máscaras devem ficar dentro do *$(document).ready(****function****()*. Por tanto o nosso script ficará da seguinte maneira.

$('#cep').mask("99999-999")***;***

$('#cpf').mask("999.999.999-99")***;***

$('#telefone').mask("(99)9999-9999")***;***

$('#celular').mask("(99)9999-9999")***;***

Após a inserção da validação e da máscara podemos ver na figura 3, que o objetivo foi alcançado e os plug-ins exercem suas funções de maneira muito bem e sem necessidade de um código tão grande.

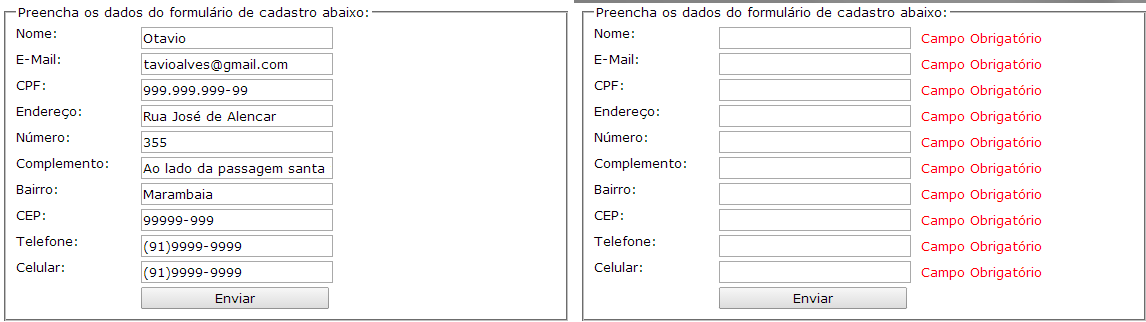


Fig. 3 – Exemplo dos plugins de máscara e validação em ação

**4. Salvando dados do usuário com o WebStorage**

Com HTML5, páginas da web podem armazenar dados localmente no navegador do usuário.

Anteriormente, isso era feito com cookies. No entanto, a Web de armazenamento é mais seguro e mais rápido. Os dados não são incluídas com cada solicitação do servidor, mas usados ​​apenas quando solicitado. Também é possível armazenar grandes quantidades de dados, sem afetar o desempenho do site.

Por tanto após o usuário inserir todos os dados, os mesmo devem ser guardados pelo seu navegador ou enviado para o servidor. Para isso utilizaremos a função abaixo que estará em nossa tag <script>. A mesma é acionada quando o botão enviar é clicado.

function enviar()

{ localStorage.setItem("Nome",nome.value); localStorage.setItem("Email",email.value);

localStorage.setItem("CPF",cpf.value); localStorage.setItem("Endereço",endereco.value); localStorage.setItem("Número",numero.value localStorage.setItem("Complemento",complemento.value);

localStorage.setItem("Bairro",bairro.value);

localStorage.setItem("CEP",cep.value); localStorage.setItem("Telefone",telefone.value); localStorage.setItem("Celular",celular.value);

}

Podemos ver que todos os dados foram gravados com sucesso pelo navegador através do console de desenvolvedor do navegador. Apresentado na figura 4.

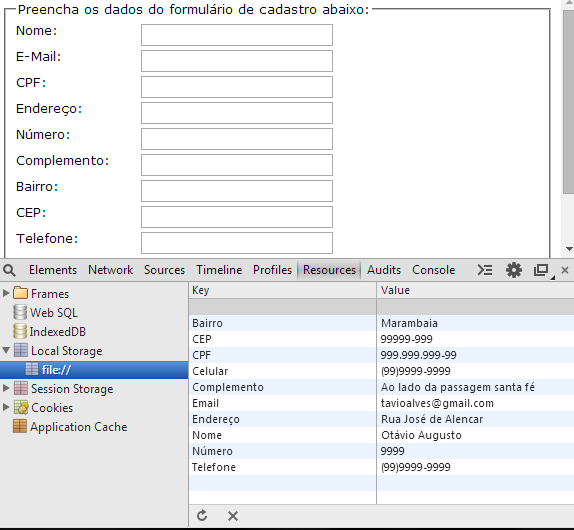


Fig. 4 – Apresentação do Web Storage

**5. Conclusão**

A utilização de formulários por parte de aplicações Web nunca deixarão de existir. Já que é uma maneira bastante eficaz para se obter informações corretas de um usuário, porém ficou claro que é necessário que exista um controle para que não existam brechas para que alguma informação possa ser salva de maneira errônea. Por tanto é necessário que exista ao menos conhecimento por parte do desenvolvedor de ferramentas como jQuery, que vão auxilia-lo no desenvolvimento de formulários de maneira bastante prática, já que com uma pequena quantidade de linhas podemos ver o que o nosso objetivo foi atingindo com êxito, além da utilização do WebStorage que mostra ser uma eficiente forma de armazenar dados localmente sem a necessidade de cookies.

**6. Referência**

jQuery – Validação http://blog.thiagobelem.net/validacao-de-formularios-com-jquery/

jQuery – Máscaras http://vinteum.com/jquery-mask-mascaras-para-campos-html-utilizando-jquery/

Web Storage - http://www.w3schools.com/html/html5\_webstorage.asp