



Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

Artigo

Invista em você! Saiba como a DevMedia pode ajudar sua carreira. ▶

# Validação com jQuery: Como usar o plugin Validation

Veja neste artigo como utilizar o Framework jQuery Validation para a criação de validações de formulários web.

Marcar como lido



Anotar



Uma etapa fundamental no desenvolvimento de uma aplicação web é a validação de formulários, pois é nesta fase onde definimos quais restrições ou validações o formulário deve atender, assim garantindo que os usuários forneçam apenas dados válidos. Diante disso, existem dois tipos populares de validação de formulários, o chamado client-site (validações que ocorrem no lado do cliente) e o server-side (validações que ocorrem no lado do servidor). Contudo, este artigo tem como foco apenas no client-side, onde utilizaremos o Framework jQuery Validation.



14



Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

Login



métodos úteis d

obrigatório, nú

campo

disso, fornece

uma API para criação dos nossos próprios métodos de validação personalizados. E para a apresentação das mensagens (de erro e sucesso) para o usuário durante a validação, o plugin Validation oferece mais de 37 traduções, incluído os padrões em inglês e Português do Brasil.

Para começar com o jQuery Validation é bem simples: basta incluir os scripts (vide **Listagem 1**) entre as tags de um HTML que se deseja validar de um formulário web.

```
<!-- Inclusão do jQuery-->
<script src="http://code.jquery.com/jquery-1.11.1.js"></script>
<!-- Inclusão do Plugin jQuery Validation-->
<script src="http://jqueryvalidation.org/files/dist/jquery.validate.js"></script>
```

### Listagem 1. Scripts para iniciar o jQuery Validation

Observe que no script não incluímos a sua versão mais recente, pois ainda não foi testada e validada totalmente com esta versão do jQuery a 1.11.1. Então, por questões de compatibilidade, optamos por utilizá-lo. Vejamos um exemplo básico de como utilizar o jQuery Validation na validação de um formulário de comentário bem simples, onde o usuário poderá submeter o mesmo quando preencher todos os campos, incluindo um e-mail válido. Para isso, primeiramente crie um arquivo chamado “index.html” e nele acrescente o código da **Listagem 2**.



14



Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

```
<meta charset="UTF-8">
<title>Validação com jQuery: Como usar o plugin Validation</title>
<!-- Inclusão do jQuery -->
<script src="http://code.jquery.com/jquery-1.11.1.js"></script>
<!-- Inclusão do Plugin jQuery Validation -->
<script src="http://jqueryvalidation.org/files/dist/jquery.validate.js"></script>
</head>
<body>

<form id="formComentario" method="post" action="">
  <fieldset>
    <legend>Por favor preencha o nome, e-mail e comentário</legend>
    <p>
      <label for="cnome">Nome (obrigatório, pelo menos 2 caracteres)</label>
      <input type="text" id="cnome" name="nome" required minlength="2">
    </p>
    <p>
      <label for="cemail">E-mail (obrigatório)</label>
      <input type="email" id="cemail" name="email" required >
    </p>
    <p>
      <label for="ccomentario">Seu comentário (obrigatório)</label>
      <textarea id="ccomentario" name="comentario" required ></textarea>
    </p>
    <p>
      <input type="submit" value="Enviar">
    </p>
  </fieldset>
</form>
<!-- Inclusão das validações com o Plugin jQuery Validation -->
<script src="validacao.js"></script>
</body>
</html>
```

## Listagem 2. Código da página index.html

Em seguida, crie um arquivo **JavaScript** chamado “validacao.js” com o código da



14



Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

```
//Validação si  
$("#formComentario").validate();
```

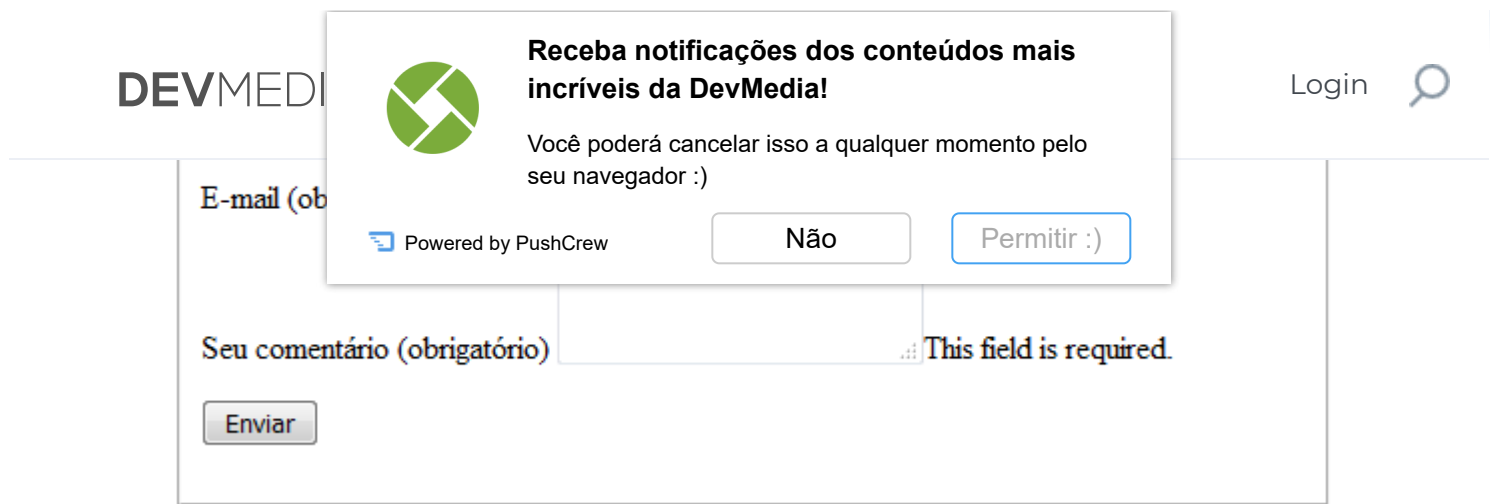
### Listagem 3. Código do arquivo validacao.js

Observe que com apenas um método “`validate()`” podemos validar o nosso formulário. O “`#formComentario`” faz referência ao nosso formulário da **Listagem 2**, onde declaramos a tag o “`id=formComentario`”, assim, o plugin `Validation` tem controle sobre o nosso formulário. Como definimos um único método de validação, o plugin entende que validará cada campo de acordo com sua especificidade. Por exemplo: como o nosso campo de e-mail está marcado no HTML como `type="email"` e `required`, o `Validation` só permitirá a entrada de e-mail válido e de preenchimento obrigatório. Se porventura não definíssemos o `required` no campo de e-mail, então o `Validation` só permitiria a entrada de e-mail válido, mas sem o preenchimento obrigatório.

Para visualizarmos a validação do nosso formulário basta acessar pelo navegador a página `index.html` e depois tentar enviar o formulário clicando no botão “`enviar`”, assim teremos o resultado semelhante à **Figura 1**.



14



The screenshot shows a web form with a notification overlay. The notification is a white box with a green circular logo containing a white 'X' and the text 'Receba notificações dos conteúdos mais incríveis da DevMedia!'. Below the logo, it says 'Você poderá cancelar isso a qualquer momento pelo seu navegador :)'. At the bottom of the notification are two buttons: 'Não' and 'Permitir :)'. The notification is 'Powered by PushCrew'. The form below has a label 'E-mail (ob' and a text input field. Below that is a label 'Seu comentário (obrigatório)' and a text input field. A red error message 'This field is required.' is displayed next to the comment input field. At the bottom left of the form is a button labeled 'Enviar'. In the top right corner of the page, there is a 'Login' link and a magnifying glass icon.

**Figura 1.** Resultado da validação do formulário utilizando o Plugin Validation

## Trabalhando com o método `validate()` do Plugin Validation

Nesta seção conheceremos as principais opções de validação com o método `validate()`, vejamos:

- **debug** : nessa opção podemos definir `true` para o plugin não permitir a submissão do formulário, mesmo que todos os dados sejam válidos; ou `false` para permitir o envio dos dados. Esta opção é interessante quando desejamos realizar algum tipo de teste.
- **messages**: com esta opção podemos definir mensagens personalizadas para cada regra, definida para cada campo, assim realizando um “override” das mensagens tidas como padrão do Plugin Validation. Contudo, vale ressaltar que o Validation fornece em seu repositório mensagens em português.
- **rules**: é nesta opção onde definimos as regras de validação para cada campo do nosso formulário. Vejamos na **Tabela 1** as principais regras de validação:



**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

required	
min	Define um número ou valor mínimo válido. Por exemplo, se definimos como 4, o usuário só poderá informar um número maior igual a 4. Abaixo disso a informação é inválida;
max	Define um número ou valor máximo válido. Por exemplo, se definimos como 10, o usuário só poderá informar um número que seja menor ou igual a 10. Acima disso a informação é inválida;
range	Determina um intervalo, contendo um valor mínimo e máximo válido. Por exemplo, se definirmos como [10, 20], o usuário só poderá informar um número que seja maior ou igual a 10 e menor ou igual a 20;
minlength	Determina uma quantidade mínima de caracteres. Por exemplo, se definirmos como 10, o usuário só poderá informar um valor com no mínimo 10 caracteres. Abaixo disso a informação é inválida;
maxlength	Define uma quantidade máxima de caracteres. Por exemplo, se definirmos como 5, o usuário só poderá informar um valor com no máximo 5 caracteres. Acima disso a informação é inválida;
rangelength	Define um intervalo, contendo uma quantidade mínima e máxima de caracteres válidos. Por exemplo, se definirmos como [2, 6], o usuário só poderá informar um valor com no mínimo 2 e no máximo 6 caracteres;
url	Se definido como true, o campo só aceitará URLs válidas;
date	Se definido como true, o campo só aceitará datas válidas;
	Caso definido como true, o input só aceitará datas no



14



**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

digits	
creditcard	cartão de crédito válido,
remote	Caso defina uma URL de verificação, por exemplo “verificar_se_existe_cpf.php”. Então o Validation realiza uma verificação remota com a url e com o valor informado no campo, assim, se a resposta para a requisição de validação for false, então o Validation entende que a informação no campo não é válida, agora caso a resposta for true então a informação é válida. Por exemplo: verificar se o CPF já existe na aplicação;
equalTo	Verifica se o valor dois campos são iguais. Por exemplo: Verificar se as senhas são iguais, no cenário de inserir uma senha e repetir a senha novamente.
accept	Esta regra é utilizada para o campo “input= file” para upload de arquivo, onde podemos definir quais mime-types são permitidos. Por exemplo, se definimos esta regra com o valor “audio/*”, então estamos dizendo ao Validation que só é permitido apenas arquivos de áudio;
extension	Nesta regra podemos definir uma certa extensão de arquivo que poderá ser aceito;
phoneUs	Se definido como true, o campo só aceitará valor de telefone válido dos EUA
require_from_group	Garante que pelo o menos um campo deve ser válido dentro de um grupo de campos.

**Tabela 1.** Regras do Plugin Validation



14



Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

Login



```
<script src="http://jqueryvalidation.org/files/dist/
/additional-methods.min.js"></script>
```

Agora faremos um exemplo de validação de formulário utilizando algumas regras destacadas nesta seção. Para isso, acrescente o código da **Listagem 4** no arquivo “index.html”. Logo certifique-se de incluir as **Listagens 1 e 3** também para funcionar a validação com o plugin Validation.

```
<form id="formularioValidation">
  <fieldset>
    <legend>Opções de validação do método validate() do plugin jQuery Validation</legend>
    <p>
      <label for="campo1">Regras: required, email e remote </label>
      <input type="email" id="campo1" name="campo1">
    </p>
    <p>
      <label for="campo2">Regras: minlength, maxlength e rangelength </label>
      <input type="text" id="campo2" name="campo2">
    </p>
    <p>
      <label for="campo3">Regras: min, max e range </label>
      <input type="text" id="campo3" name="campo3">
    </p>
    <p>
      <label for="campo4">Regras: accept </label>
      <input type="file" id="campo4" name="campo4">
    </p>
    <p><b>Grupo de telefones, onde pelo o menos um deve ser válido</b><br/>
      <label for="telefone_pessoal">1º require_from_group: Telefone pessoal: </label>
```



14





Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

```
<input type="text" value="Nome" />
</p>
<input type="text" value="Telefone" />
</fieldset>
</form>
```

telefone\_trabalho">

#### Listagem 4. Código da página index.html

Em seguida, acresceste o código da **Listagem 5** no arquivo `validacao.js`. É neste arquivo onde definimos as regras de validação para o formulário da **Listagem 4**. Certifique-se de incluir a chamada para o arquivo `validacao.js` com a tag na página `index.html` para funcionar a validação.

```
$( "#formularioValidation" ).validate({
  debug: true,
  rules: {
    campo1: {
      required: true,
      email: true,
      // remote: 'check-email.php' //Deve utilizar um arquivo
      // (por exemplo: check-email.php) para verificar algo, e
      // assim retornar um boolean true ou false para satisfazer esta opção;
    },
    campo2:{
      minlength: 3,
      maxlength: 4,
      // ou
      rangelength: [3, 4] //Realiza a mesma coisa dos anteriores
    },
    campo3:{
      min: 10,
      max: 15,
```



14



Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

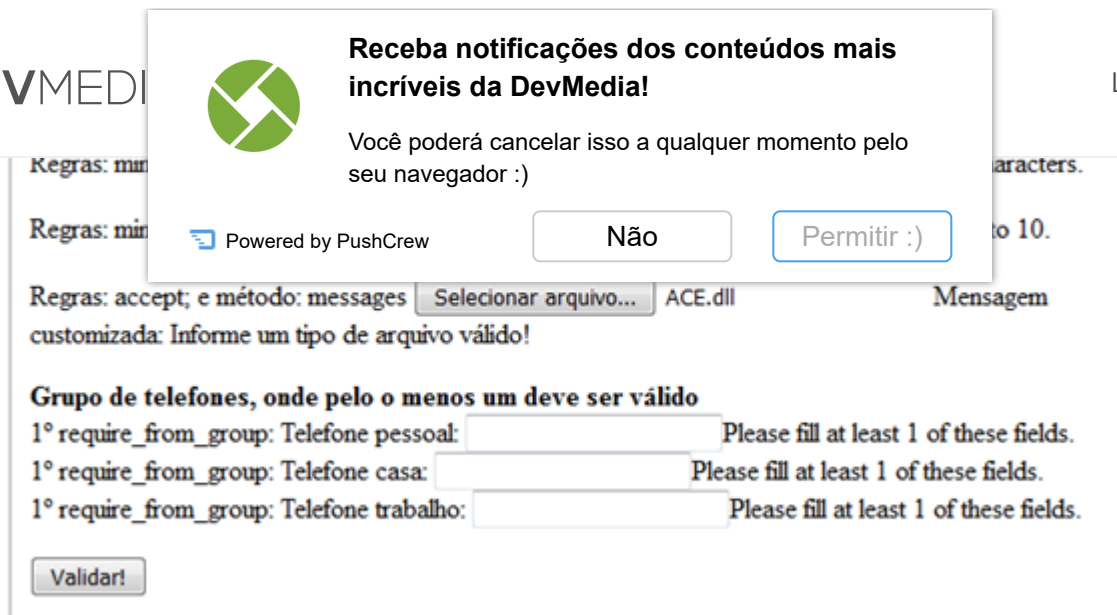
```
    },
    telefone: {
      require_from_group: [1, ".grupo_telefone"]
    },
    telefone_casa: {
      require_from_group: [1, ".grupo_telefone"]
    },
    telefone_trabalho: {
      require_from_group: [1, ".grupo_telefone"]
    }
  },
  messages: {
    //exemplo
    campo4: {
      accept: "Mensagem customizada: Informe um tipo de arquivo válido!"
    }
  }
});
```

### Listagem 5. Código do arquivo validacao.js

Por fim, para executarmos este exemplo acessando a página `index.html` pelo navegador. Em seguida, preencha o formulário com dados inválidos e depois clique em “Validar!”. Assim, teremos o resultado semelhante à **Figura 2**.



14



**Figura 2.** Resultado da validação utilizando o método `validate()`

## Criando um método personalizado de validação utilizando o plugin Validation

Além do [jQuery Validation](#) fornecer diversas opções de validação com o método `validator()`, ainda sim pode surgir uma nova necessidade de validação que ainda não foi implementada nativamente, por isso o `Validation` fornece em sua API um método estático chamado “`addMethod(...)`”, que permite ao desenvolvedor criar seu próprio método de validação.

Para demonstrar essa API criaremos uma nova validação que fará a verificação se o valor informado é um número de telefone celular válido, logo esta validação vai além de verificar a quantidade de dígitos no formato de 11 e 10 dígitos, incluindo o DDD. Então para começar, precisamos entender a sintaxe simples do método estático “`addMethod(...)`”, como mostra o código a seguir:





Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

Diante do cenário e da sintaxe apresentada, partiremos para a implementação do nosso exemplo. Para isso, adicione o código da **Listagem 6** no arquivo `index.html`, onde ficará o nosso formulário. Em seguida, certifique-se de incluir a **Listagem 1** também para funcionar a validação com o plugin `Validation`.

```
<form id="formularioValidaTelefoneCelular">
    <fieldset>
        <legend>Criando uma validação customizada com o método estático addMe
        <b>Validação do número de celular</b>
        <p>
            <label for="telefone_celular">Telefone celular:</label>
            <input type="text" id="telefone_celular" name="telefone_celular">
        </p>
        <input type="submit" value="Validar!">
    </fieldset>
</form>
```

**Listagem 6.** Código do arquivo `index.html`

Em seguida, adicione o código da **Listagem 7** no arquivo `validacao.js`, que conterà a nossa nova validação e a chamada para validar o formulário. Certifique-se de incluir a chamada do arquivo `validacao.js` com a tag na página `index.html` para funcionar a validação.



14



Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

```
return  
} else if  
return  
}  
if (["00", "01", "02", "03", , "04", , "05", , "06", , "07", , "08", "09", "10"]  
.indexOf(value.substring(0, 2)) != -1) {  
    return (this.optional(element) || false);  
}  
if (value.length < 10 || value.length > 11) {  
    return (this.optional(element) || false);  
}  
if (["6", "7", "8", "9"].indexOf(value.substring(2, 3)) == -1) {  
    return (this.optional(element) || false);  
}  
return (this.optional(element) || true);  
}, 'Informe um número de telefone celular válido!');  
//Nosso formulário, pronto para ser validado  
$("#formularioValidaTelefoneCelular").validate({  
    debug: true,  
    rules: {  
        telefone_celular: {  
            required: true,  
            celular: true,  
        }  
    }  
});
```

### Listagem 7. Código do arquivo validacao.js

Por fim, para executarmos o nosso exemplo basta acessar a página `index.html` em nosso navegador. Com isso, preencha o formulário com dados inválidos e depois clique em “Validar!”. Assim, teremos o resultado semelhante à **Figura 3**.



14



**Figura 3.** Resultado da nossa nova validação

Por fim, criamos e testamos a nossa nova validação de formulário. Diante disso, este exemplo pode ser tomado como base para a criação de novas validações.

## Refatorando regras e mensagens de validação

Durante a criação de diferentes validações de formulários em nossa aplicação utilizando o Plugin Validation, as vezes nos deparamos com vários campos com as mesmas regras e mensagens. Para simplificar, a refatoração pode reduzir essas duplicações através dos seguintes métodos estáticos do Plugin Validation:

- `addMethod` : adiciona uma nova regra de validação;
- `addClassRules` : adiciona uma regra de validação a um elemento.

Para exemplificar a refatoração de um código com duplicações de regras e mensagens, utilizaremos o seguinte exemplo: imagine que temos dez campos no formulário para cadastro de clientes, onde cada um precisa ser preenchido obrigatoriamente e com, no mínimo, dois caracteres; e para cada regra precisa-se criar mensagens personalizadas.

Podemos apelidar métodos de validação existentes com diferentes mensagens e agrupá-los em uma única classe. Para a implementação do nosso exemplo crie um

exemplo chamado “Validação de Formulário” e deixe-o disponível no código de exemplo 3.





**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)



Powered by PushCrew

Não

Permitir :)

```
<form id="form" method="post" action="">
  <fieldset>
    <legend>Refatorando regras e mensagens de validação - Cadastro de cliente </legend>
    <p>
      <label for="campo1">campo 1</label>
      <input class="cliente" type="text" id="campo1" name="campo1" >
    </p>
    <p>
      <label for="campo2">campo 2</label>
      <input class="cliente" type="text" id="campo2" name="campo2" >
    </p>
    <p>
      <label for="campo3">campo 3</label>
      <input class="cliente" type="text" id="campo3" name="campo3" >
    </p>
    <p>
      <label for="campo4">campo 4</label>
      <input class="cliente" type="text" id="campo4" name="campo4" >
    </p>
    <p>
      <label for="campo5">campo 5</label>
      <input class="cliente" type="text" id="campo5" name="campo5" >
    </p>
    <p>
      <label for="campo6">campo 6</label>
      <input class="cliente" type="text" id="campo6" name="campo6" >
    </p>
    <p>
      <label for="campo7">campo 7</label>
      <input class="cliente" type="text" id="campo7" name="campo7" >
    </p>
    <p>
      <label for="campo8">campo 8</label>
      <input class="cliente" type="text" id="campo8" name="campo8" >
    </p>
    <p>
      <label for="campo9">campo 9</label>
```





**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

```
<p>
<input
</p>
</fieldset>
</form>
```

### Listagem 8. Código do arquivo index\_parte\_2.html

Em seguida, crie um arquivo chamado “ validacao\_parte\_2.js ” e depois acrescente neste o código da **Listagem 9**, que conterà o nosso código de validação alinhado com os requisitos do nosso exemplo. Em seguida, certifique-se de incluir o arquivo validacao\_parte\_2.js com a tag na página index\_parte\_2.html para funcionar a validação.

```
//Valida o formulário
$( "#form" ).validate({
  debug: true
});
// apelido necessário para o cRequired com nova mensagem
$.validator.addMethod("cRequired", $.validator.methods.required,
  "Informação do cliente necessária");
// apelido cMinlength
$.validator.addMethod("cMinlength", $.validator.methods.minlength,
  $.validator.format("Informação do cliente deve ter pelo menos {0} caracteres"));
// combina os dois, aplicando as regras nos campos que contenham a classe chamada "cl"
$.validator.addClassRules("cliente", { cRequired: true, cMinlength: 2 });
```

### Listagem 9. Código do arquivo validacao\_parte\_2.js



14





Receba notificações dos conteúdos mais incríveis da DevMedia!

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

Login



apelidos ou nome

segundo passan

Minlength”; no

ras de

validação; e no último parâmetro passamos as mensagens customizadas. Por fim, utilizamos o método estático `addClassRules` para aplicar as duas regras nos campos que contenham a classe chamada "cliente".

Para executarmos o nosso exemplo basta acessar a página `index.html` pelo navegador. Em seguida, preencha o formulário com dados inválidos e depois clique no botão “enviar”, assim, teremos o resultado semelhante à **Figura 4**.

Refatorando regras e mensagens de validação - Cadastro de cliente

campo 1	<input type="text"/>	Informação do cliente necessária
campo 2	<input type="text" value="1"/>	Informação do cliente deve ter pelo menos 2 caracteres
campo 3	<input type="text"/>	Informação do cliente necessária
campo 4	<input type="text"/>	Informação do cliente necessária
campo 5	<input type="text" value="1"/>	Informação do cliente deve ter pelo menos 2 caracteres
campo 6	<input type="text" value="1"/>	Informação do cliente deve ter pelo menos 2 caracteres
campo 7	<input type="text"/>	Informação do cliente necessária
campo 8	<input type="text" value="1"/>	Informação do cliente deve ter pelo menos 2 caracteres
campo 9	<input type="text" value="1"/>	Informação do cliente deve ter pelo menos 2 caracteres
campo 10	<input type="text" value="1"/>	Informação do cliente deve ter pelo menos 2 caracteres



14

DEV MEDIA

Login



que este Plugin

nte este artigo.

**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

Powered by PushCrew

Não

Permitir :)

### Bibliografia:

- ZAEFFERER, Jörn. [jQuery Validation Plugin](#).
- EQUIPE JQUERY [jQuery API](#).

### Tecnologias:

JavaScript

jQuery

Marcar como lido



Anotar

Por **Brendo**

Em 2015

RECEBA NOSSAS NOVIDADES

Informe o seu e-mail




14



**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

 Powered by PushCrew

Não

Permitir :)

## Suporte ao aluno - Deixe a sua dúvida.

### ASSINATURA DEVMEDIA

Faça parte dessa comunidade 100% focada em programação e tenha acesso ilimitado. Nosso compromisso é tornar a sua experiência de estudo cada vez mais dinâmica e eficiente. Portanto, se você quer programar de verdade seu lugar é aqui. Junte-se a mais de...

**+ 800 MIL**  
PROGRAMADORES

**69,90\***  
/ MÊS

Séries

Projetos completos

Cursos




14

DEV MEDIA

**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

 Powered by PushCrew

Não

Permitir :)

Login



## Assine

A assinatura é cobrada através do seu cartão de crédito. \*Tempo mínimo de assinatura: 12 meses.



### Plataforma para Programadores

Compre por telefone:

(21) 3593-6903



Revistas

Baixe o App

Fale conosco

Trabalhe conosco


Assinatura Empresarial

Av. Ayrton Senna 3000, Shopping Via Parque,  
grupo 3087 - Barra da Tijuca - Rio de Janeiro - RJ



**Receba notificações dos conteúdos mais incríveis da DevMedia!**

Você poderá cancelar isso a qualquer momento pelo seu navegador :)

 Powered by PushCrew

Não

Permitir :)



14