

## Macoratti.net jQuery- Usando jQuery com ASP .NET

A **jQuery** é a biblioteca **JavaScript** mais usada em aplicações web atualmente. Ela foi desenvolvida em 2006 por [John Resig](#) e é suportada por uma comunidade de desenvolvedores além do time do projeto **jQuery**. Ela foi construída para ser leve, poder ser usadas nos principais navegadores e ser compatível com a especificação **CSS3**.

A **jQuery** simplifica a maneira de como acessar os elementos **DOM** tornando mais fácil a navegação pelas páginas. A biblioteca possui poderosos recursos Ajax, manipulação de eventos e efeitos de animação, permitindo um rápido desenvolvimento de aplicações web. Além disso a **jQuery** permite a criação de *plugins* que possibilita aos desenvolvedores criar recursos poderosos com base no núcleo **jQuery**.

**Obs: O DOM é uma especificação do consórcio W3C que não depende de plataforma ou de linguagem. Ele é usado para fazer alterações em documentos HTML e XML, sendo uma API muito usada para este fim.**

Por tudo isso você não pode ignorar a **jQuery** pois ela permite utilizar muitos recursos em suas aplicações web tornando-a mais interativa e amigável ao usuário.

Eu já apresentei a **jQuery** em meu artigo - [ASP .NET - Apresentando JQuery - Macoratti.net](#), e se você nunca ouviu falar ou nunca usou **jQuery** sugiro que você o leia pois não vou repetir toda a parte introdutória que mostra como usar **jQuery** em páginas ASP .NET.

Atualmente o **jQuery** esta na versão **1.6.1** e você pode baixá-la aqui: <http://jquery.com/> e esta disponível também tanto no Visual Studio como as versões Express quando você cria uma aplicação web usando o *template* padrão (*não o vazio*).

A **jQuery** esta incluída tanto nos projetos [ASP .NET Web Forms](#) como nos projetos [ASP .NET MVC](#). A *library* Microsoft Ajax foi projetada para atrair desenvolvedores JQuery e você pode mesclar plug-ins JQuery e controles de cliente Microsoft Ajax em sua aplicação.

Neste artigo vou mostrar como utilizar **jQuery** em páginas ASP .NET para realizar algumas tarefas do dia a dia envolvendo controles ASP .NET.

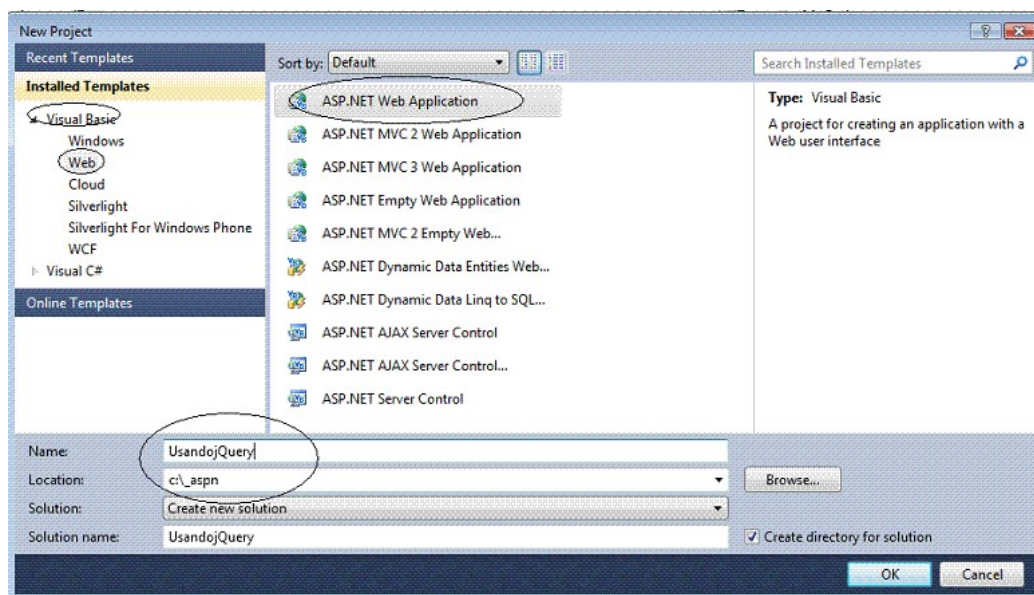
Recursos usados nos exemplos do artigo:

- [Visual Web Developer 2010 Express Edition](#);
- **jquery-1.4.1.min.js** que já vem embutida com o projeto ASP .NET Web Application;
- Navegadores: Internet [Explorer 8.0](#), [FireFox 3.6](#) e [Chrome 9.0](#);

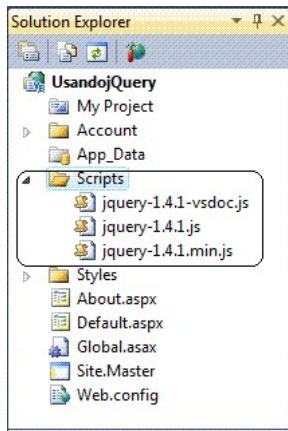
## Usando jQuery na prática

### 1- Criando o projeto

Abra o [Visual Web Developer 2010 Express Edition](#) e no menu **File-> New Project** crie um novo projeto usando a linguagem [Visual Basic](#) e o *template* **ASP .NET Web Application** com o nome **UsandojQuery** e clique no botão **OK**;

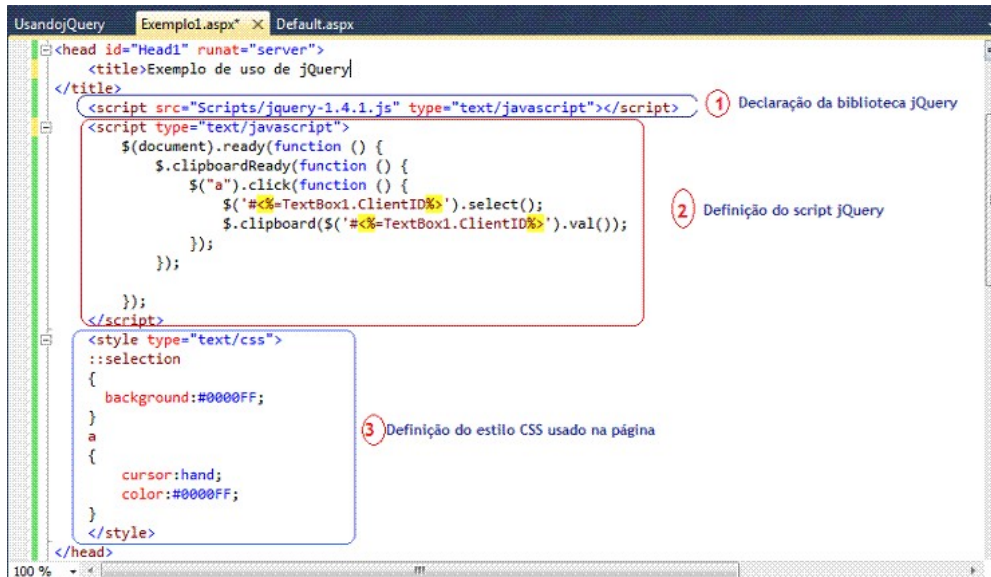


Após criar o projeto se você observar a janela **Solution Explorer** ir ver na estrutura do projeto a pasta **Script** a biblioteca **jQuery (jquery-1.4.1.min.js)**:



De forma geral nas páginas .aspx dos nossos exemplos poderemos distinguir 3 seções:

1. A declaração da biblioteca jQuery que iremos usar (tag `<script src="Scripts/jquery-1.4.1.js" ...>` (Estamos usando a biblioteca jQuery localmente na pasta Scripts);
2. A declaração do código jQuery entre as tags `<script type="text/javascript">`;
3. A declaração do estilo CSS usado na página na tag `<style type="text/css">`;



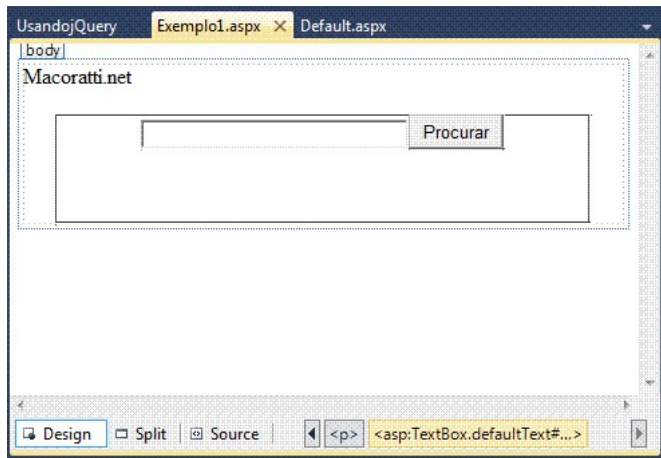
Para cada uma das tarefas que iremos realizar usando jQuery criaremos uma página .aspx no projeto da seguinte forma :

- No menu **Project -> Add New Item** e selecione o template **Web Form**;
- Informe o nome da página e clique no botão **Add**;

*Obs: Se você estiver usando o VS 2008 deverá ter aplicado o Service Pack 1 e a correção KB958502 <http://archive.msdn.microsoft.com/KB958502> para poder habilitar o intellisense do jQuery.*

## 2 - Incluindo um texto padrão em um controle TextBox

- Crie um novo web form chamado **Exemplo1.aspx**
- Inclua um controle **TextBox** e um controle **Button** no formulário conforme o leiaute abaixo:



```
<form id="form1" runat="server">
  <p>Macoratti.net</p>
  <div align="center">
    <fieldset style="width:400px;height:80px;">
      <p><asp:TextBox ID="TextBox1" width="200px" CssClass="defaultText"
        ToolTip="Informe seu critério de busca" runat="server"></asp:TextBox>
      <asp:Button ID="btnSubmeter" Text="Procurar" runat="server" /></p>
    </fieldset>
  </div>
</form>
```

O tag `<fieldset>` é usada para agrupar logicamente elementos em um formulário. Esta tag desenha uma caixa em torno dos elementos relacionados do formulário.

Observe que o controle `TextBox` possui a propriedade `CssClass` definida com o nome `defaultText`; vamos definir então o script de estilo conforme abaixo:

```
<style type="text/css" media="screen">
  .defaultText
  {
    font-style:italic;
    color:#CCCCC;
  }
</style>
```

Agora vamos definir o script `jQuery` para exibir o texto - **Informe seu critério de busca** - mesmo quando a caixa de texto não tem o foco. A seguir temos o código necessário para isso:

```
<script type="text/javascript">
  $(document).ready(function () {
    var caixaTextoProcurar = $('#<%=TextBox1.ClientID%>');
    caixaTextoProcurar.focus(
      function () {
        if (caixaTextoProcurar.val() == this.title) {
          caixaTextoProcurar.removeClass("defaultText");
          caixaTextoProcurar.val("");
        }
      }
    );

    caixaTextoProcurar.blur(
      function () {
        if (caixaTextoProcurar.val() == "") {
          caixaTextoProcurar.addClass("defaultText");
          caixaTextoProcurar.val(this.title);
        }
      }
    );
    caixaTextoProcurar.blur();
  });
</script>
```

- a função `document.ready()` retorna o controle `TextBox` usando o seu `ClientID` e o salva na variável local `caixaTextoProcurar`:  
`var caixaTextoProcurar = $('#<%=TextBox1.ClientID%>');`

- no evento `focus` verificamos se a variável contém o texto padrão : `caixaTextoProcurar.val() == this.title`

**Obs:** A propriedade `ToolTip` do controle `TextBox` é renderizado como `title` em tempo de execução, assim, o texto de `ToolTip` e retornado usando o `title`.

- em caso positivo removemos o estilo usado: `caixaTextoProcurar.removeClass("defaultText");` e limpamos o seu conteúdo: `caixaTextoProcurar.val("");`

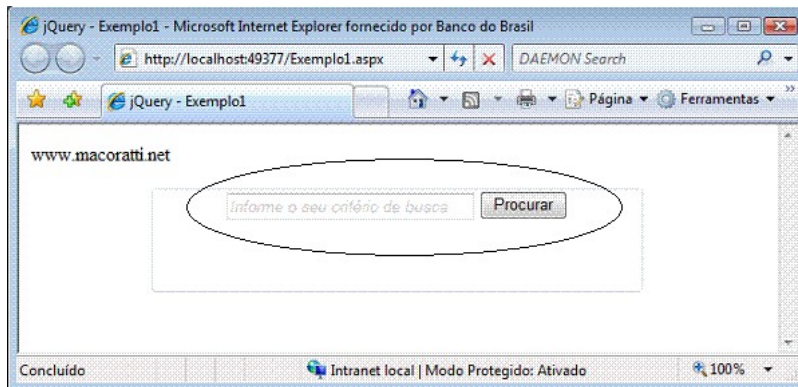
- No evento `blur` verificamos se o `TextBox` está vazio : `if (caixaTextoProcurar.val() == "")`

- se estiver vazio anexamos o estilo : `caixaTextoProcurar.addClass("defaultText");`

- e incluímos o texto padrão : `caixaTextoProcurar.val(this.title);`

- chama o evento `blur` na carga da página de forma que o `TextBox` é inicializado quando perder o foco: `caixaTextoProcurar.blur();`

Executando o projeto teremos o resultado exibido pela figura abaixo:



O código completo do formulário é exibido a seguir:

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="Exemplo1.aspx.vb" Inherits="UsandojQuery.Exemplo1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
  <title>jQuery - Exemplo1
</title>
  <script src="Scripts/jquery-1.4.1.js" type="text/javascript"></script>
  <script type="text/javascript">
    $(document).ready(function () {
      var caixaTextoProcurar = $('#<%=TextBox1.ClientID%>');
      caixaTextoProcurar.focus(
        function () {
          if (caixaTextoProcurar.val() == this.title) {
            caixaTextoProcurar.removeClass("defaultText");
            caixaTextoProcurar.val("");
          }
        }
      );
      caixaTextoProcurar.blur(
        function () {
          if (caixaTextoProcurar.val() == "") {
            caixaTextoProcurar.addClass("defaultText");
            caixaTextoProcurar.val(this.title);
          }
        }
      );
      caixaTextoProcurar.blur();
    });
  </script>
  <style type="text/css" media="screen">
    .defaultText
    {
      font-style:italic;
      color:#CCCCC;
    }
  </style>
</head>
<body>
  <form id="form1" runat="server">
    <p>www.macoratti.net</p>
    <div align="center">
      <fieldset style="width:400px;height:80px;">
        <p><asp:TextBox ID="TextBox1" width="200px" CssClass="defaultText" ToolTip="Informe o seu critério de busca" runat="server"></asp:TextBox>
        <asp:Button ID="btnSubmeter" Text="Procurar" runat="server" /></p>
      </fieldset>
    </div>
  </form>
</body>
</html>
```

### 3 - Realizando a validação em um formulário de login (usando um plugin jQuery)

Vamos realizar a validação de um formulário de login bem simples usando jQuery. Para este nosso exemplo vamos validar os campos login e senha da seguinte forma:

- Os campos login e senha são obrigatórios;

- O campo senha deverá ter no mínimo 8 dígitos;

A jQuery vem com vários plugins para usar uma funcionalidade adicional para a biblioteca de base. Para manusear a questão da validação, o plugin de validação jQuery é muito útil e proporciona uma rica coleção de propriedades e métodos para o uso.

Vamos usar o plugin para validação que pode ser obtido em : <http://plugins.jquery.com/project/validate>

Após fazer o download do arquivo descompacte-o para uma pasta e inclua o arquivo **jquery.validate.js** na pasta Scripts do projeto.

Para habilitar o plugin de validação no nosso formulário podemos simplesmente arrastar e soltar o plugin no formulário. Após isso será criado de forma automática o seguinte código na página ASPX:

```
<scriptsrc="Scripts/jquery.validate.js" type="text/javascript" ></script>
```

Para validar um formulário web utilizando o plugin, aplicamos o método a seguir sobre o elemento do formulário:

**validate([options])**

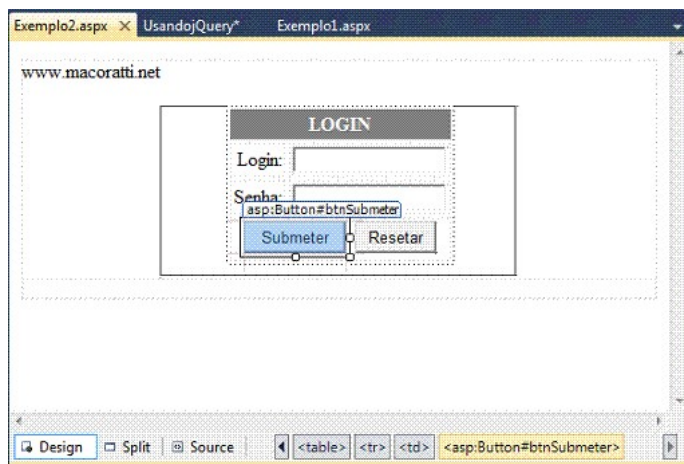
Abaixo temos algumas das opções que podemos usar com **validate**:

- **rules**: Constituído por pares chave/valor. A chave é o nome do elemento, enquanto o valor é a regra de validação para ser aplicado ao elemento;
- **messages**: Consiste em pares chave/valor. A chave é o nome do elemento, enquanto o valor é a mensagem personalizada para ser exibida se a validação falhar por esse elemento;
- **errorLabelContainer**: Esta é a área atual do recipiente dentro do qual as mensagens de erro são exibidas;
- **errorContainer**: Esta é a área de conteúdo principal que contém o *errorLabelContainer*;
- **wrapper**: Envolve as mensagens de erro no elemento necessário e é usado em conjunto com *errorLabelContainer*;
- **onsubmit**: Valida o formulário ao enviar. Pode ser definido como false para validar outros eventos;
- **highlight** : Esta opção especifica como realçar os campos inválidos;
- **unhighlight**: Esta opção reverte as alterações feitas pela opção *highlight*;

- Crie um novo web form chamado **Exemplo2.aspx**

- Inclua dois controles **TextBox** e dois controles **Button** no formulário conforme o leiaute abaixo:

- Incluir uma área <div> abaixo dos controles de **login** onde serão exibidas as mensagens de erro: **<div align="center" class="alertmsg"></div>**



```
<form id="form1" runat="server">
<p>www.macoratti.net</p>
<div align="center">
<fieldset style="width:300px;height:140px;">
<table border="0" cellpadding="3" cellspacing="3">
<tr><td colspan="2" class="header">LOGIN</td></tr>
<tr>
<td align="right">
<asp:Label ID="lblLogin" runat="server" Text="Login: "></asp:Label>
</td>
<td align="left">
<asp:TextBox ID="txtLogin" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td align="right">
<asp:Label ID="lblSenha" runat="server" Text="Senha: "></asp:Label>
</td>
<td align="left">
<asp:TextBox ID="txtSenha" runat="server" TextMode="Password"></asp:TextBox>
</td>
</tr>
</table>
</div>
```



```

<tr>
  <td align="center" colspan="2">
    <asp:Button ID="btnSubmeter" runat="server" Text="Submeter" />&nbsp;  
    <asp:Button ID="btnResetar" runat="server" Text="Resetar" />
  </td>
</tr>
</table>
</fieldset>
<br />
<div align="center" class="alertmsg">
</div>
</div>
</form>

```

Vamos definir também o seguinte script para o estilo usado no formulário:

```

<style media="screen" type="text/css">
.header
{
  background-color:Gray;
  font-weight:bold;
  color:White;
  text-align:center;
}
.alertmsg
{
  color:#FF0000;
}
</style>

```

A seguir vamos definir o script **jQuery** , incluindo as declarações para usar a biblioteca **jQuery** e o plugin de validação, conforme abaixo:

```

<script src="Scripts/jquery-1.4.1.js" type="text/javascript"></script>
<script src="Scripts/jquery.validate.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
  $(document).ready(function () {
    var validator = $("#form1").validate({
      rules: {
        txtLogin: "required",
        txtSenha: { required: true, minlength: 8 }
      },
      messages: {
        txtLogin: "Informe o seu login",
        txtSenha: { required: "Informe a sua senha",
          minlength: "A senha deverá possuir no mínimo 8 dígitos"
        }
      },
      wrapper: 'li',
      errorLabelContainer: $("#form1 div.alertmsg")
    });
    $("#btnReset").click(function (e) {
      e.preventDefault();
      $("#txtLogin").val("");
      $("#txtSenha").val("");
    });
  });
</script>

```

-Na função **document.ready** chamamos a função **validate** : **var validator = \$("#form1").validate**

-Incluimos a opção **rules** : definimos o login e senha como obrigatório e a senha com o mínimo de 8 caracteres: **txtLogin: "required",txtSenha: { required: true, minlength: 8 }**

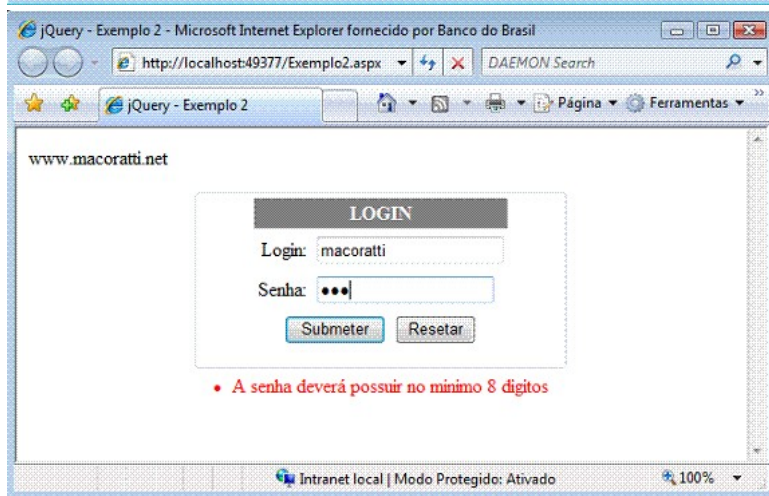
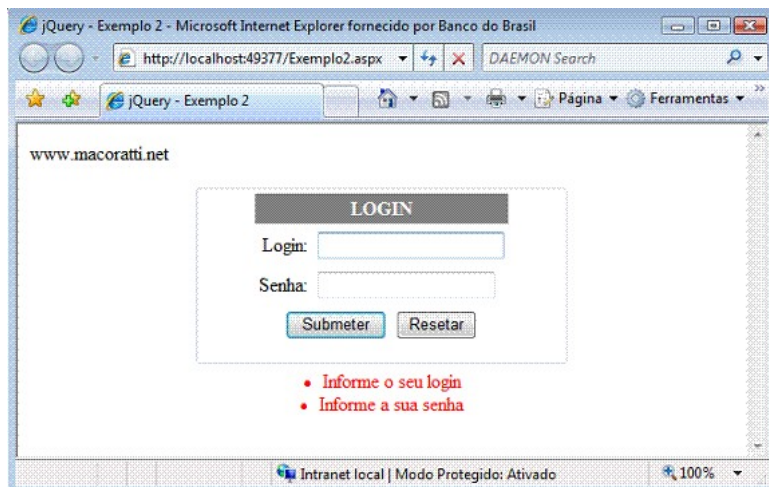
- incluímos a opção **messages** : onde definimos as mensagens a serem exibidas ao usuário:  
**txtLogin: "Informe o seu login", txtSenha: { required: "Informe a sua senha", minlength: "A senha deverá possuir no mínimo 8 dígitos" }**

- Definimos a opção **wrapper** : onde definimos que as labels de erros devem ser exibidas em um elemento  
**ListItem : wrapper: 'li',**

-Definimos o **errorLabelContainer** onde as etiquetas de erro serão exibidas quando a validação falhar:  
**errorLabelContainer: \$("#form1 div.alertmsg")**

- No evento **Click** do botão **Resetar** limpamos os campos do formulário:  
**\$("#btnReset").click(function (e) { e.preventDefault(); \$("#txtLogin").val(""); \$("#txtSenha").val("");});**

Executando este formulário iremos obter:



O código completo do formulário pode ser visto a seguir:

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="Exemplo2.aspx.vb" Inherits="UsandojQuery.Exemplo2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>jQuery - Exemplo 2</title>
    <script src="Scripts/jquery-1.4.1.js" type="text/javascript"></script>
    <script src="Scripts/jquery.validate.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function () {
            var validator = $("#form1").validate({
                rules: {
                    txtLogin: "required",
                    txtSenha: { required: true, minlength: 8 }
                },
                messages: {
                    txtLogin: "Informe o seu login",
                    txtSenha: { required: "Informe a sua senha",
                        minlength: "A senha deverá possuir no mínimo 8 dígitos"
                    }
                },
                wrapper: 'li',
                errorLabelContainer: $("#form1 div.alertmsg")
            });
            $("#btnReset").click(function (e) {
                e.preventDefault();
                $("#txtLogin").val("");
                $("#txtSenha").val("");
            });
        });
    </script>
    <style media="screen" type="text/css">
        .header
```

```
{
    background-color:Gray;
    font-weight:bold;
    color:White;
    text-align:center;
}
.alertmsg
{
    color:#FF0000;
}
</style>
</head>
<body>
<form id="form1" runat="server">
<p>www.macoratti.net</p>
<div align="center">
<fieldset style="width:300px;height:140px;">
<table border="0" cellpadding="3" cellspacing="3">
<tr><td colspan="2" class="header">LOGIN</td></tr>
<tr>
<td align="right">
<asp:Label ID="lblLogin" runat="server" Text="Login: "></asp:Label>
</td>
<td align="left">
<asp:TextBox ID="txtLogin" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td align="right">
<asp:Label ID="lblSenha" runat="server" Text="Senha: "></asp:Label>
</td>
<td align="left">
<asp:TextBox ID="txtSenha" runat="server" TextMode="Password"></asp:TextBox>
</td>
</tr>
<tr>
<td align="center" colspan="2">
<asp:Button ID="btnSubmeter" runat="server" Text="Submeter"/>&nbsp;  
<asp:Button ID="btnResetar" runat="server" Text="Resetar" />
</td>
</tr>
</table>
</fieldset>
<br />
<div align="center" class="alertmsg">
</div>
</div>
</form>
</body>
</html>
```

Vimos como é simples usar os recursos do **jQuery** para realizar tarefas simples em páginas ASP .NET e também como usar um *plugin* de validação.

Na segunda parte do artigo vou mostrar como usar os recursos do [jQuery com o GridView e Ajax](#).

Aguarde...

Pegue o projeto completo aqui:  [UsandojQuery.zip](#)

**"Passará o céu e a terra, mas as minhas palavras jamais passarão." (Mateus 24:35)**

Referências:

- [ASP .NET - Usando os novos modelos de Projeto - Macoratti.net](#)
- [ASP .NET - Apresentando JQuery - Macoratti.net](#)
- [ASP .NET - Evitando o Copiar/Colar em controles ... - Macoratti.net](#)
- [ASP .NET - Exibindo textos e imagens com o recurso ... - Macoratti.net](#)

---

José Carlos Macoratti