



# **Análise e Desenvolvimento de Sistemas**

## **Aula 5**

**Professor Ederson Cichaczewski**

## Conversa Inicial

Olá aluno, bem-vindo à quinta aula de Programação Visual. Nela, serão abordados os seguintes tópicos:

- Conceitos básicos da linguagem de marcação extensível XML.
- Linguagem ASP.NET, que permite rodar scripts em um servidor, assim como desenvolver páginas em estilo web forms.
- Desenvolvimento de uma aplicação ASP.NET no ambiente visual studio 2015.
- Aplicação em linguagem ASP.NET com visual BASIC.NET.
- Integração de um arquivo XML com a página em ASP.NET e code behind em visual basic.net.

**Confira no material online os comentários iniciais do professor Ederson.**

## Contextualizando

A linguagem de marcação extensível XML foi desenvolvida pelo World Wide Web Consortium W3C para descrever dados. Esse formato se tornou padrão para troca de dados entre aplicações da internet com importante integração com a linguagem ASP.NET.

A ASP.NET permite o desenvolvimento de aplicações WEB com base no .NET framework, possibilitando integrar códigos em linguagem C# ou VB.NET em páginas WEB por meio de um servidor, chamado IIS (*Internet Information Services*), que irá executar os scripts. A ferramenta de desenvolvimento que incorpora todos os recursos para o desenvolvimento de páginas com ASP.NET é o Visual Studio,

inclusive já possuindo um servidor IIS integrado para testar e debugar projetos, visto também que ASP.NET é compilado, e não apenas interpretado como o HTML.

Iremos ver nesta aula exemplos de interface gráfica para WEB com ASP.NET, inclusive integrando com XML.

**Confira no material online a contextualização feita pelo professor Ederson.**

## **Tema 1- Conceitos de XML**

### **XML – Linguagem de Marcação Extensível**

A XML (Linguagem de Marcação Extensível) define um conjunto de regras para formatação de um documento por meio de tags que facilitam o processamento de dados por parte dos humanos e dos computadores. Dentre as principais características desta linguagem, temos:

- É extensível, porque o desenvolvedor pode criar as suas próprias tags;
- A estrutura é definida por documentos de esquema ou DTDs (Document Type Definition);
- Apenas descreve dados, não se preocupa em como apresentá-los graficamente;
- Possui uma estrutura de código parecida com a linguagem de marcação de hipertexto HTML.

Veja algumas regras iniciais da XML:

- As tags de fechamento são obrigatórias;
- Aninhamento de elementos: a tag declarada por primeiro deve ser fechada depois da segunda;
- Ex: <rua> <nome> XV de Novembro </nome> </rua>;

- Todos os atributos devem ter um valor, e este ficar entre aspas;
- Há diferenciação entre letras minúsculas e maiúsculas.

### **Estrutura básica de um documento XML**

- **Cabeçalho:** <?xml version="1.0"?>

Ainda podemos incluir a codificação de caracteres e se depende de outro documento DTD:

```
<?xml version="1.0"encoding="UTF-8" standalone =
"no"?>
```

- **Elemento raiz (apenas um):**

```
<root>
<saud>Olá</saud>
</root>
```

- **Namespace:** define o vocabulário dos elementos em um URI (Identificador de Recurso Universal).

```
<root xmlns="http://www.w3.org/1999/xhtml"> ...
</root>
```

- **Elemento:** compreende os subelementos e o conteúdo que está entre as suas tags de abertura e fechamento:

```
<itemMenu itemNum="1">

    <hyperLink>default.html</hyperLink>

<titulo>Home</titulo>
</itemMenu>
```

- É preciso cuidar com a sintaxe, conforme definido pelo W3C.
- Pode conter um atributo ou mais, sem repetir:

```
<pais estado="Parana" cidade="Curitiba"> Brasil </pais>
```

## Entidades

Alguns caracteres especiais da sintaxe da linguagem, como “ < ”, “ & ” e “ > ”, ao serem utilizados no conteúdo do elemento devem ser tratados como entidades.

Ex: a operação 1 < 2 fica:

<oper> 1 &lt; 2 </oper>

<i>Caractere</i>	<i>Referência a entidade</i>
<	&lt;
>	&gt;
"	&quot;
'	&apos;
&	&amp;

## Espaços em branco

Durante o processo de análise do XML os espaços em branco, tabulações e quebras de linha são retirados.

Para adicionar espaços em branco use:

- A entidade &#160; <saud>Olá &#160; Mundo</saud>
- Uma seção CDATA, que possibilita que dados como os caracteres < ou &, assim como espaços em branco, sejam usados sem causar problemas, pois as seções CDATA não são analisadas pelo analisador XML, sendo tratadas como apenas texto:

<![CDATA[<saud>Olá Mundo </saud>]]>

- Atributo space:<saud xml:space="preserve">Olá Mundo </saud>

**Para mais informações sobre os conceitos de XML, confira o vídeo do professor Ederson no material online.**

## **Tema 2 – Conceitos de ASP.NET**

### **ASP.NET (Active Server Pages)**

ASP.NET (Active Server Pages) é uma linguagem que permite colocar em uma página WEB scripts que serão executados por um servidor. Dentre as principais características do ASP.NET, temos:

- Ele é compilado, não interpretado.
- Suas páginas possuem extensão .aspx
- Ele permite o desenvolvimento de páginas no estilo Web Forms.

### **Code Behind**

Code Behind é a linguagem que roda no servidor, pode ser VB.NET ou C#.

### **Servidor**

É necessário um servidor para rodar o código ASP.NET. Normalmente é um servidor IIS, mas se estiver desenvolvendo em Visual Studio, este ambiente possui um servidor local integrado (IIS Express) para rodar as páginas.

### **Estrutura de um arquivo ASP.NET**

Na estrutura de um arquivo ASP.NET, basicamente temos a mesma estrutura de uma página HTML5, adicionando um namespace XML e, adicionalmente:

- Um bloco de script executável no servidor, usando o atributo runat;
- Um cabeçalho com a diretiva Page que especifica a linguagem code behind;
- Os códigos ASP.NET ficam entre <% e %>.

Segue abaixo o código com a estrutura básica de uma página com extensão .aspx:

```
<%@ Page Language="vb" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title></title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>

      </div>
    </form>
  </body>
</html>
```

### **Web Forms + Code Behind**

Pode-se adicionar elementos visuais na página WEB .aspx assim como em Windows Forms.

Ex: caixa de texto

```
<asp:TextBox ID="textHello"
runat="server"></asp:TextBox>
```

Um arquivo separado é criado para o código em VB.NET (.aspx.vb) ou C# (.aspx.cs) que fará o tratamento de eventos.

### Ferramenta: Visual Studio 2015

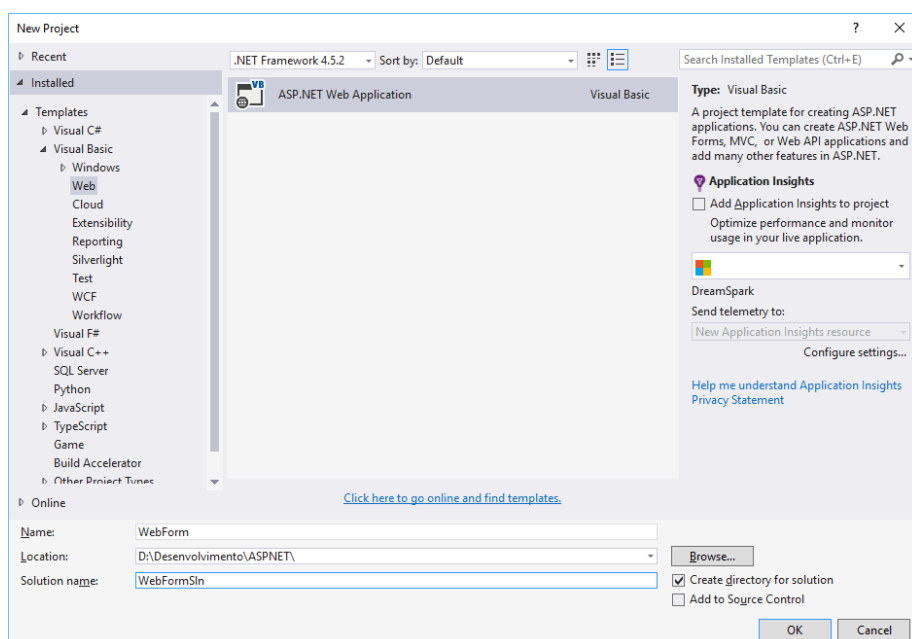
Esta ferramenta permite a criação de uma página ASP.NET, incluindo um toolbox com todos os elementos de form, visualização do design, compilação e rodar a página WEB com o servidor local IIS.

**Para mais informações sobre os conceitos de ASP.NET, confira o vídeo do professor Ederson no material online.**

## Tema 3 – ASP.NET no Visual Studio 2015

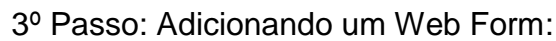
### Criando um projeto ASP.NET no Visual Studio Community 2015

1º Passo: Menu File → New → Project...



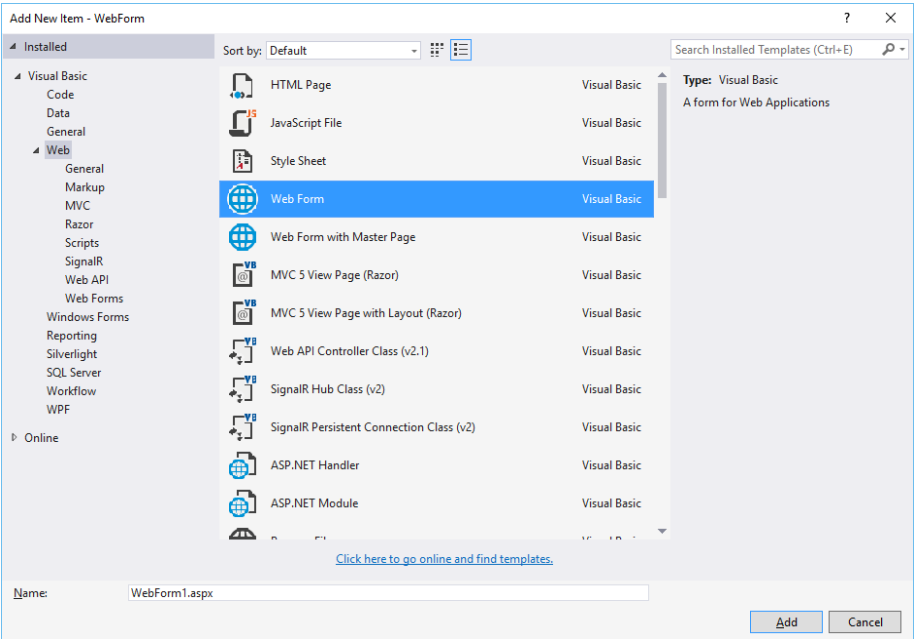


Escolher a opção Empty.

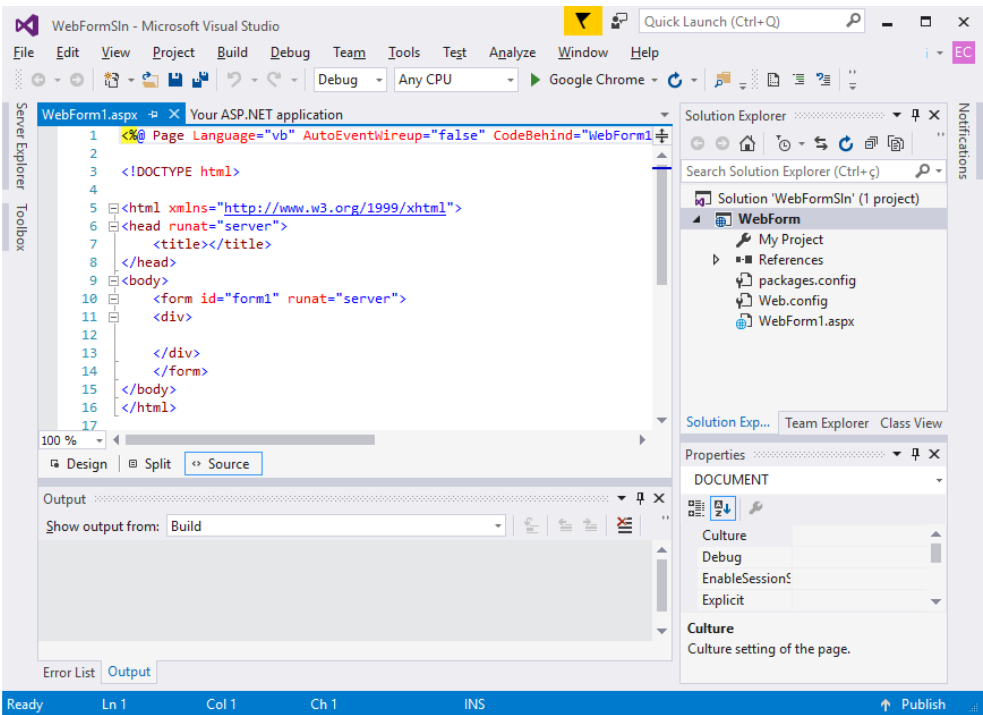


The screenshot shows the Visual Studio interface. In the top right, the 'Solution Explorer' pane is visible, showing a project named 'Solution 'WebFormSln' (1 project)'. Below it, a search bar is present. In the bottom left, the 'GET HELP' section is visible. The main area shows a context menu for the 'Add' option, which is highlighted. The menu items are: Build, Rebuild, Clean, View, Analyze, Publish..., Add Application Insights Telemetry..., Application Insights, Scope to This, New Solution Explorer View, Add (highlighted), Manage NuGet Packages..., Set as StartUp Project, Debug, Source Control, and Cut. The 'Add' option is the one being focused on.

Escolher a opção Web Form.

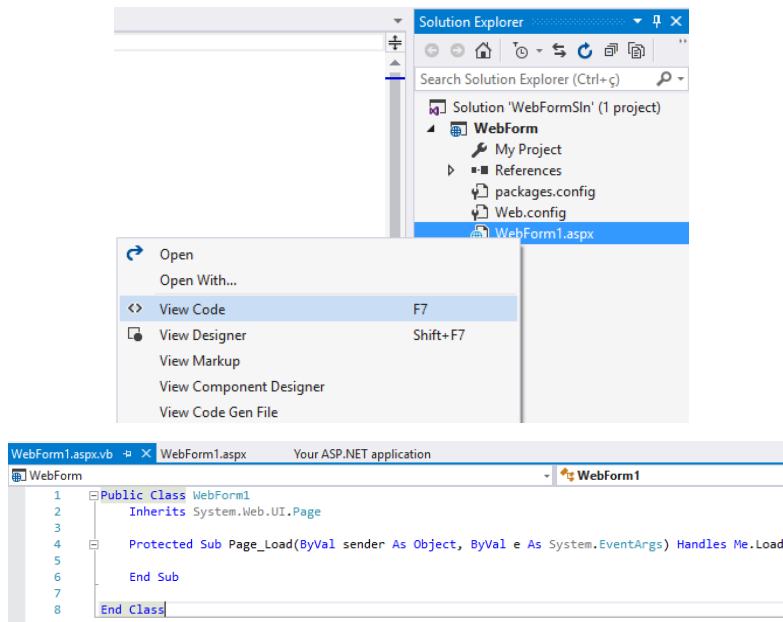


Pronto, o projeto está criado. O ambiente de desenvolvimento fica assim:



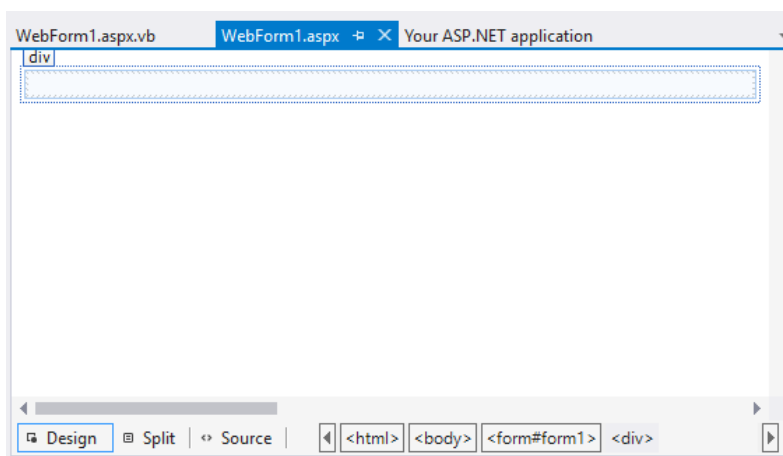
## Visualizando o Code Behind.aspx.vb

Neste exemplo, ir em Solution Explorer à direita da tela e, clicar com o botão direito do mouse no arquivo WebForm1.aspx, depois clicar em View Code. Irá abrir uma aba de código com o arquivo WebForm1.aspx.vb.



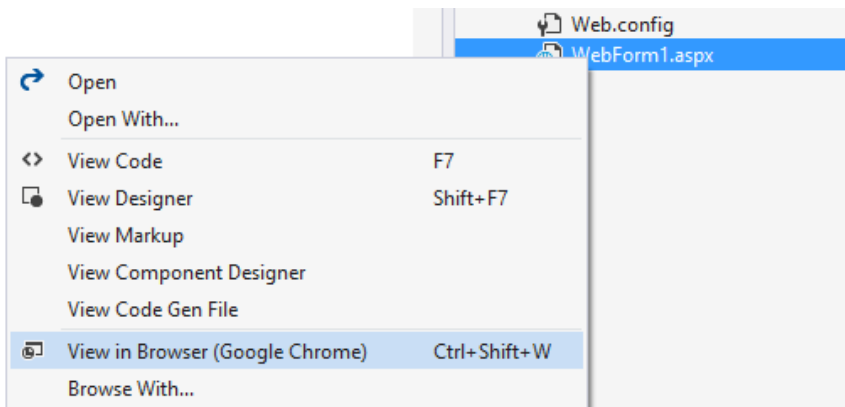
## Visualizando o modo Design

Podemos visualizar o modo Design para ver como está ficando a página em uma estrutura muito próxima à visualização no navegador, e inclusive arrastar componentes do toolbox para adicionar componentes à página. Ir no rodapé da área de visualização do código .aspx e clicar em Design.



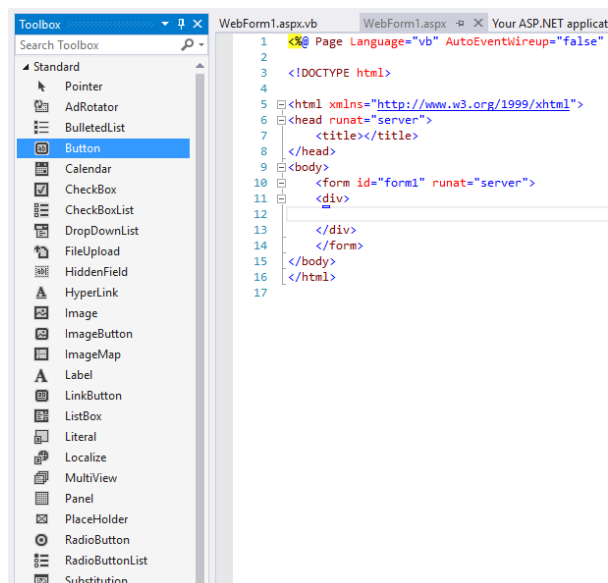
## Rodando no Browser

Para visualizar a página em seu formato final no navegador, clicar com o botão direito do mouse no arquivo WebForm1.aspx, depois clicar em View in Browser. Normalmente, na primeira vez que fizer isso o Visual Studio vai perguntar qual é o navegador que deseja utilizar para rodar a página.



## Toolbox

Podemos puxar componentes do toolbox tanto no modo Design como no modo Source.



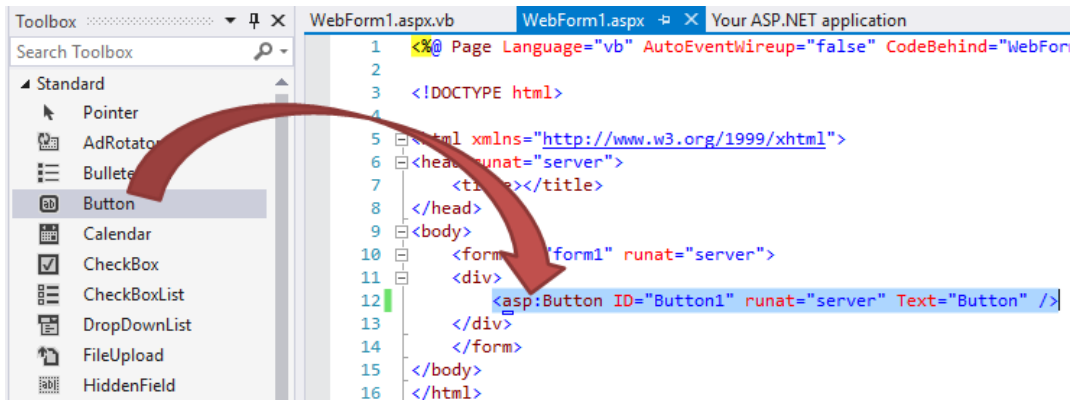
**No material online, o professor Ederson fala mais sobre a ASP.NET no Visual Studio 2015, confira!**

## Tema 4 – Criando uma página ASP.NET

### Inserindo componentes (controles) Web Forms

Para adicionar componentes gráficos na página ASP.NET basta arrastar da Toolbox para o código, então será inserida a codificação correspondente.

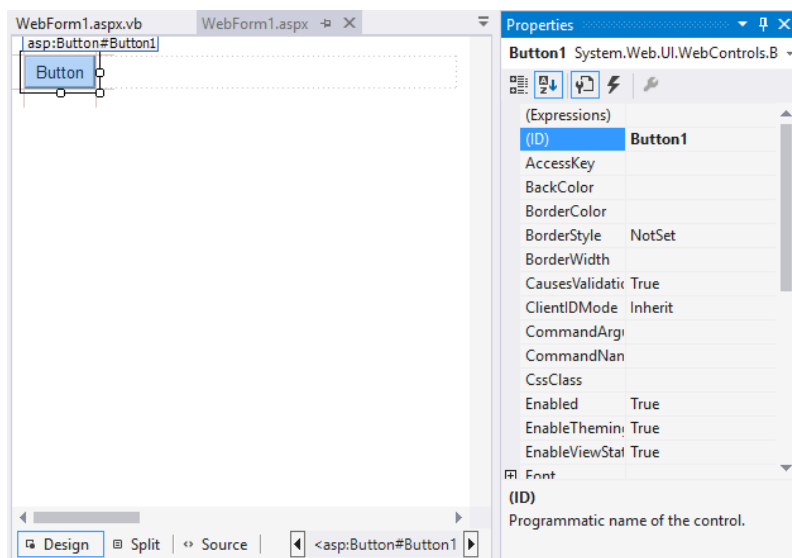
- Vamos puxar um botão (Button).



- Adicione também um TextBox.
- Coloque um cabeçalho h1 no código HTML com o texto “Minha Página ASP.NET”.

### Editando as propriedades do componente

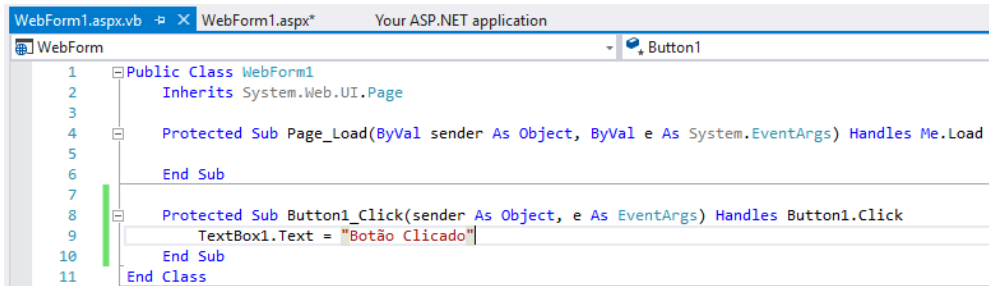
Ir no modo Design e clicar sobre o componente. Ver a janela Properties. É possível mudar o nome do componente, o texto que aparece sobre ele, entre outros parâmetros.



## Inserindo ações aos eventos

- Dar um duplo click no botão, irá abrir o código.aspx.vb
- Inserir o código abaixo no evento de click do botão:

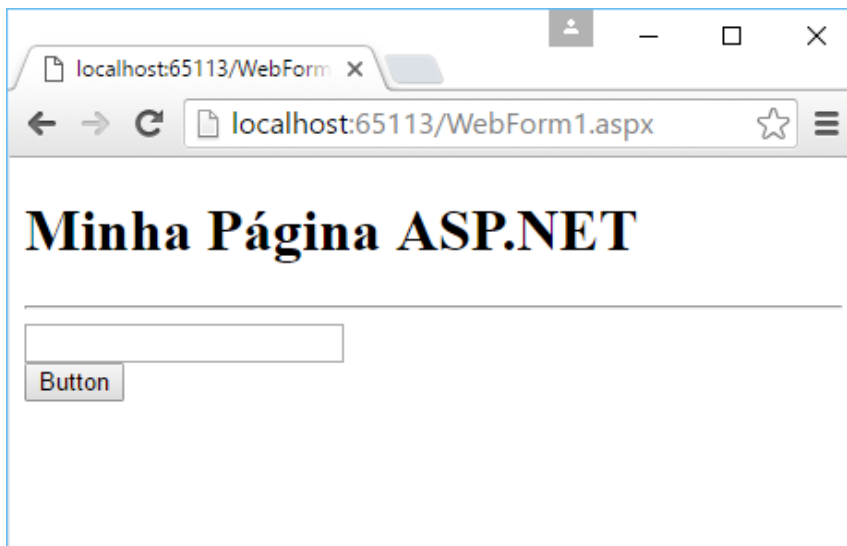
TextBox1.Text = “Botão Clicado”



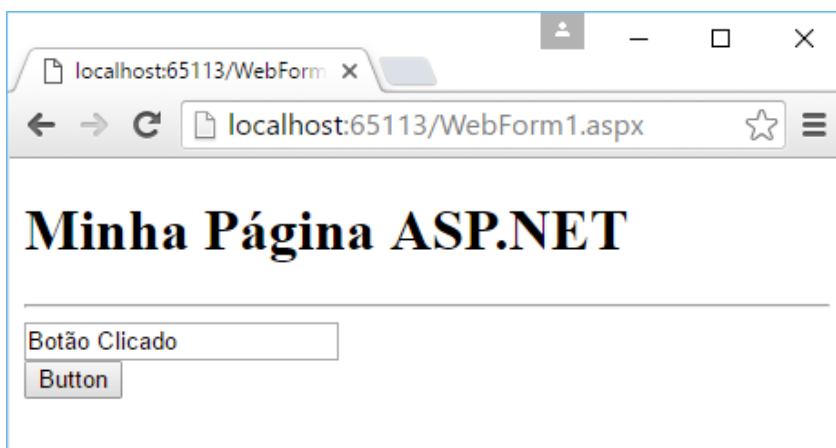
```
1 Public Class WebForm1
2     Inherits System.Web.UI.Page
3
4     Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
5
6     End Sub
7
8     Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
9         TextBox1.Text = "Botão Clicado"
10    End Sub
11 End Class
```

## Visualizando a página Web

Clique em View in Browser para visualizar a página rodando no navegador.



Clique no botão (necessário servidor IIS Express integrado rodando). Será executado aquele evento implementado no clique do botão, preenchendo o TextBox com a string especificada.



**No material online, o professor Ederson explica passo a passo a criação de uma página ASP.NET.**

## **Tema 5 – ASP.NET com XML**

### **Criando uma Página ASP.NET com XML**

Existe uma infinidade de aplicações do uso do XML em uma página ASP.NET. Vamos ver um exemplo usando o controle **AdRotator**.

- O controle AdRotator tem a função de alternar anúncios em uma mesma posição da página.
- Os dados dos anúncios ficam em um arquivo XML.
- O AdRotator seleciona randomicamente uma imagem com um hyperlink.

### **Adicionando o controle AdRotator**

Este código. A seguir, adiciona o controle AdRotator e especifica o arquivo .xml que será consultado.

```
<asp:AdRotator id="paisRotator" runat="server"
AdvertisementFile="AdRotatorInfo.xml"></asp:AdRotator>
```

Nada é necessário fazer no code behind, visto que este controle faz tudo sozinho.

Vamos continuar do exemplo anterior e adicionar o controle AdRotator. Segue o código fonte completo:

### **arquivo WebForm.aspx**

```
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind = "WebForm1.aspx.vb"
Inherits="WebForm.WebForm1" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title></title>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
<h1>Minha Página ASP.NET</h1><hr />
```

```
<asp:label ID="displayLabel" runat="server" Font-
size=
"Large" Font-Bold="True">Controle AdRotator</asp:label>
```

```
<br />
```

```
<asp:AdRotator id="paisRotator" runat="server"
```

```
AdvertisementFile="AdRotatorInfo.xml">
```

```
</asp:AdRotator> <br />
```

```
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
```

```
<br />
```

```
<asp:Button ID="Button1" runat="server"
Text="Botão" />
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```



## Arquivo AdRotatorInfo.xml

As imagens devem estar dentro da pasta do projeto, em uma subpasta chamada “images”, e possuir os mesmos nomes conforme está no código a seguir.

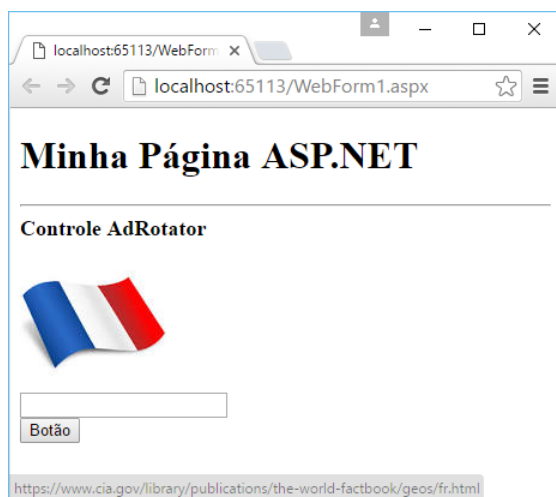
```
<?xml version="1.0" encoding="utf-8"?>
<Advertisements>
  <Ad>
    <ImageUrl>images/us.jpg</ImageUrl>

    <NavigateUrl>https://www.cia.gov/library/publications/the-world-
factbook/geos/us.html</NavigateUrl>
    <AlternateText>Informação dos Estados
Unidos</AlternateText>
    <Impressions>1</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>images/france.jpg</ImageUrl>

    <NavigateUrl>https://www.cia.gov/library/publications/the-world-
factbook/geos/fr.html</NavigateUrl>
    <AlternateText>Informação da França</AlternateText>
    <Impressions>1</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>images/italy.jpg</ImageUrl>

    <NavigateUrl>https://www.cia.gov/library/publications/the-world-
factbook/geos/it.html</NavigateUrl>
    <AlternateText>Informação da Itália</AlternateText>
    <Impressions>1</Impressions>
  </Ad>
</Advertisements>
```

## Rodando a página no browser



Cada vez que atualiza a página, muda a imagem da bandeira e o respectivo link. É possível ver o link passando o mouse por cima da imagem.

**Para mais informações sobre a ASP.NET, confira o material online.**

### Trocando ideias

Nesta aula iniciamos com uma introdução à linguagem XML e seguindo com ASP.NET, linguagem utilizada para desenvolver páginas WEB com componentes no estilo Windows Forms e que roda scripts em um servidor, utilizando uma linguagem C# ou VB.NET por trás (code behind). A integração do ASP.NET com XML abre ainda mais o leque de recursos para desenvolvimento de páginas WEB dinâmicas.

Entre exemplos de sites que são desenvolvidos utilizando a tecnologia ASP.NET, podemos citar o site da própria Microsoft, obviamente, e também o site MySpace, entre outros. Acesse-os clicando nos links a seguir.

<http://www.microsoft.com/pt-br/>

<http://www.myspace.com/>

O conhecimento adquirido nesta aula é fundamental para o seu aprendizado, de forma a ter em suas mãos as

ferramentas disponíveis conforme as necessidades de projeto que surgirem;

Não fique com dúvidas sobre o assunto, estude o tema consultando as referências bibliográficas desta aula e também outras fontes de pesquisa.

## Na Prática

Para consolidar os conhecimentos vistos nesta aula, vamos desenvolver uma aplicação ASP.NET que consiste em uma página WEB que faz o sorteio de um número de 0 a 100.

- 1º Passo: crie um novo projeto de uma aplicação WEB com code behind C#;
- 2º Passo: insira um cabeçalho h1 com o texto “Sorteador”;
- 3º Passo: insira dois componentes: 1 botão e 1 caixa de texto;
- 4º Passo: renomeie o texto do botão para “Sortear”. Então edite o evento clique do botão, inserindo o seguinte código:

```
Random rnd = new Random();
```

```
TextBox1.Text = rnd.Next(100).ToString();
```

- 5º Passo: Salve tudo e rode a página no navegador. Aperte o botão “Sortear” várias vezes e veja o número mudando na caixa de texto.

Segue o código HTML5 completo:

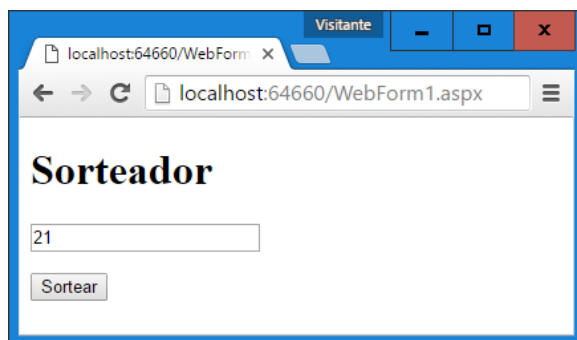
```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeBehind="WebForm1.aspx.cs"  
Inherits="WebSortedor.WebForm1" %>
```

```

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1>Sorteador</h1>
        </div>
        <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
        <p>
            <asp:Button ID="Button1" runat="server"
Text="Sortear" OnClick="Button1_Click" />
        </p>
    </form>
</body>
</html>

```



## Síntese

Nesta aula vimos os conceitos da linguagem de marcação extensível XML. Vimos também uma introdução sobre a linguagem ASP.NET, seus recursos básicos e desenvolvimento com Visual Studio. Por fim, criamos umas páginas ASP.NET com code behind em Visual Basic .NET e também uma página ASP.NET que possui integração com arquivo XML.

O bom entendimento desta aula é fundamental, visto que ela trata dos recursos de interface visual baseada em forms para WEB, que é utilizada na criação de sites da internet que rodam scripts no servidor e trabalham com arquivos de dados em XML.

**Confira a síntese do professor Ederson no material online.**

## Referências

DEITEL, P. J. Ajax, **Rich Internet Applications e desenvolvimento WEB para programadores**. São Paulo: Pearson, 2008.

WAHLIN, D. **XML e ASP.NET para desenvolvedores**. São Paulo: Pearson Education. 2003.

SHEPHERD, George. **Microsoft ASP.NET 3.5 Passo a Passo**. São Paulo: Bookman, 2009.