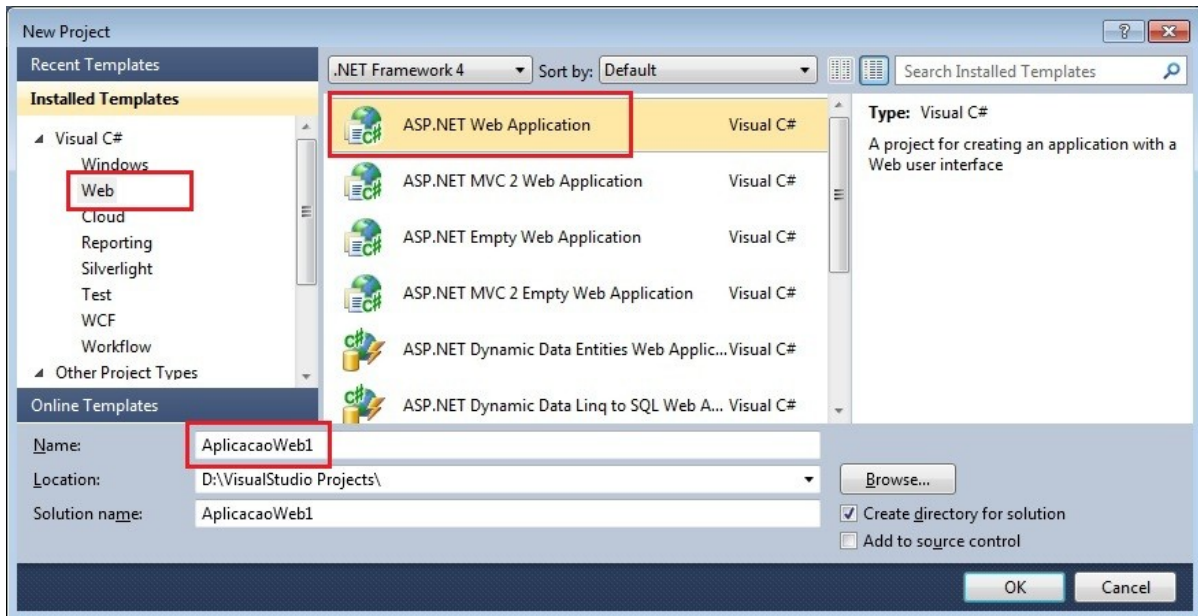


Apostila: [Desenvolvimento Web com ASP.NET](#)

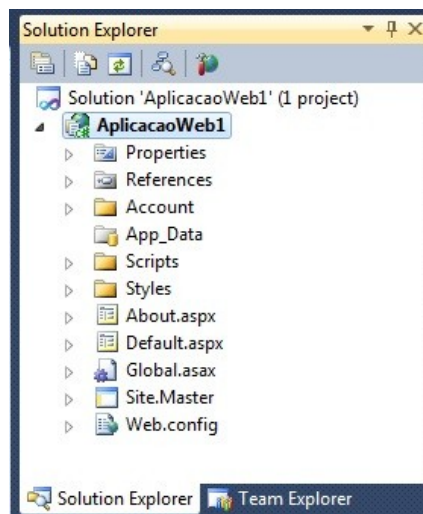
Professor: [Fabiano Papaiz](#)

ASP.NET – Criando a Primeira Aplicação Web

Abra o Visual Studio e crie um novo projeto do tipo **ASP.NET Web Application** e chame-o de **AplicacaoWeb1**, conforme ilustrado na figura abaixo:



Após a criação, podemos verificar na janela **Solution Explorer** que o VS criou a estrutura do nosso web site com pastas e arquivos, conforme figura a seguir:



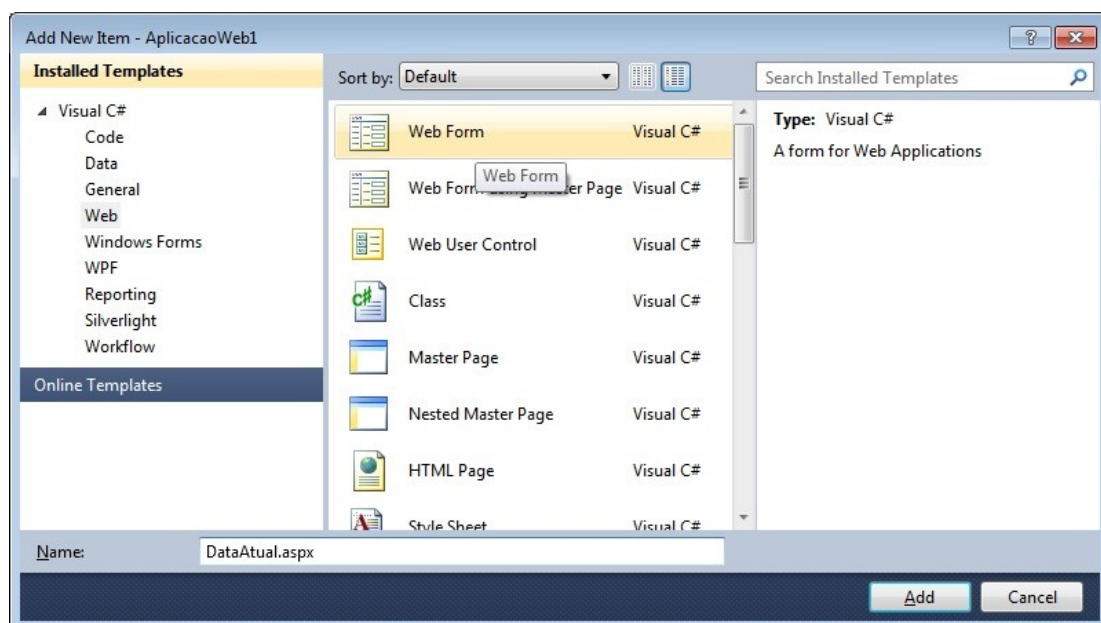
Uma aplicação web ASP.NET pode conter diversos tipos de arquivos, a tabela a seguir explica de forma resumida alguns desses tipos:

Tipo de arquivo	Extensão	Descrição
Web Form	.aspx	Representa uma página que os usuários poderão ver em seus browsers.
Master Page	.master	Estes arquivos nos permitem definir estruturas globais de layout para as páginas
Web User Control	.ascx	Contém "pedaços de páginas" que podem ser reutilizados em diversas páginas do site.
Web Configuration File	.config	Contém configurações globais que são utilizadas pelo site criado
Skin File	.skin	Contém as definições de como os controles serão exibidos (aparência)
Class	.cs	Código fonte C# do web site. Cada Web Form (.aspx) possui um arquivo .cs vinculado – conhecidos como arquivos <i>Code Behind</i> .
Global Application Class	.asax	Neste arquivo podemos programar alguns eventos de aplicação acontecem, como quando a aplicação é iniciada no servidor web, quando uma sessão de usuário for iniciada, quando um erro for disparado etc

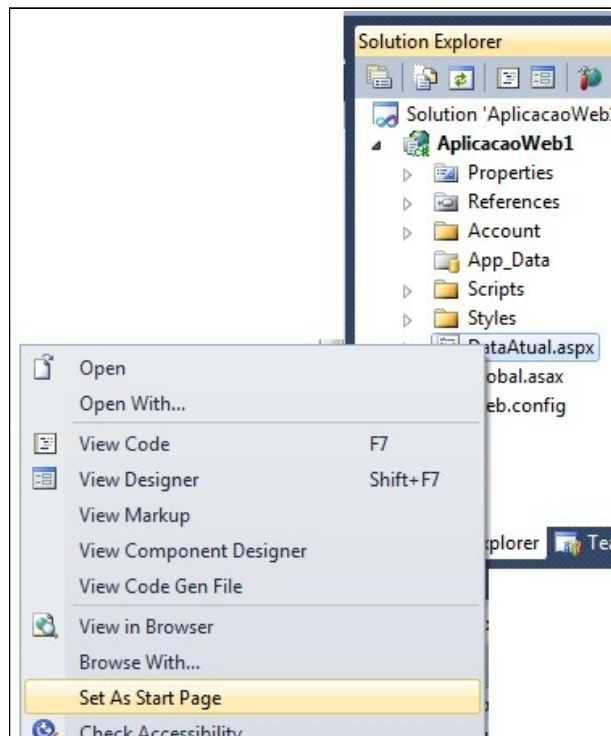
Vamos continuar a construir nossa aplicação web.

Na janela [Solution Explorer](#), selecione os arquivos [About.aspx](#), [Default.aspx](#) e [Site.master](#) e exclua-os do projeto.

Depois insira um novo [Web Form](#) através do menu [Project / Add New Item](#). Chame-o de [DataAtual.aspx](#), conforme figura:



Agora vamos definir este Web Form como a página inicial da nossa aplicação. Fazemos isso selecionando-a na janela **Solution Explorer**, clicando com o botão direito e marcando a opção **Set As Start Page**, conforme ilustração:



Na paleta de componentes, arrastem um **Label** para dentro da região do **body** da nossa página. Definam sua propriedade **ID** para **lblData Atual**.

Agora vamos inserir o código para exibir a data atual neste label assim que a página for carregada. Dê um duplo-clique em qualquer área da página que esteja fora do body – isso fará com que o arquivo **.cs** (code behind) vinculado à nossa página seja exibido. Percebam que já estará definido um evento chamado **Page_Load** – este evento é disparado assim que a página for carregada pelo servidor web.

Insira o código a seguir dentro deste evento:

```
namespace AplicacaoWeb1
{
    public partial class DataAtual : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            lblDataAtual.Text = "Olá, a data atual é " + DateTime.Now.ToString();
        }
    }
}
```

Agora execute sua aplicação e veja o resultado. Após a página ser exibida, pressionem **Ctrl+F5** para recarregá-la e vejam que a data e hora serão atualizadas. Esta página

foi gerada dinamicamente pelo servidor web e enviada para o browser do usuário. Examinem o código fonte da página gerada e vejam que não há código ASP.NET dentro dela, apenas o HTML.

Perceba que um ícone aparecerá na barra de tarefas, ele se refere ao [ASP.NET Development Server](#), o qual é um servidor web que o VS inicializa para visualizarmos nossa aplicação durante o seu desenvolvimento. Quando uma aplicação estiver finalizada, ela irá ser executada por um servidor web mais robusto e confiável, o [IIS](#) (*Internet Information Services*).