



**Cetec**  
Capacitações

**CPS**  
Centro  
Paula Souza

 **GOVERNO DO ESTADO**  
**SÃO PAULO**

GOVERNO FEDERAL  
**BRASIL**  
PÁTRIA EDUCADORA

- Introdução ao Visual Studio 2012
- O que posso fazer com o .NET?
- Termos da plataforma
- Visão geral do .NET Framework
  - ✓ O que é o CLR (Common Language Runtime)?
- Download do Visual Studio
  - ✓ Instalação do Visual Studio 2012
- Conhecendo o ambiente
  - ✓ Tela Inicial
- Tipos de variáveis
  - ✓ bool
  - ✓ byte
  - ✓ sbyte
  - ✓ decimal
  - ✓ double
  - ✓ float

- ✓ short (Int16)
- ✓ ushort (Int16)
- ✓ int (Int32)
- ✓ uint (UInt32)
- ✓ long (Int64)
- ✓ ulong (UInt64)
- ✓ object (Qualquer tipo)
- ✓ char
- ✓ String

➤ Tipos de operadores

- ✓ Aritméticos
- ✓ Unários
- ✓ Lógicos
- ✓ Condicionais
- ✓ Relacionais
- ✓ Igualdade



## Introdução ao Visual Studio 2012

Visual Studio é um conjunto completo de ferramentas de desenvolvimento para construção de aplicações Web ASP.NET, serviços Web XML, aplicações desktop e aplicativos móveis.

Visual Basic, Visual C# e Visual C++ todos usam o mesmo ambiente de desenvolvimento integrado (IDE), que permite o compartilhamento de ferramentas e facilita a criação de soluções de linguagens.

Além disso, essas linguagens usam a funcionalidade do *.NET Framework*, que fornece acesso às tecnologias chaves que simplificam o desenvolvimento de aplicativos Web em ASP e serviços Web XML.

## O que posso fazer com .NET ?

O .NET permite desenvolver soluções como:

- Aplicativos Web
- Aplicativos para Servidores
- Aplicativos *Smart Client*
- Aplicativos de Console
- Aplicativos de Banco de Dados
- Serviços Windows
- Serviços Web

## Termos da plataforma

- **CLR** - Sigla de *Common Language Runtime*. Base comum a todas as linguagens .NET, o CLR é o ambiente que gerencia a execução de código escrito em qualquer linguagem. Faz parte do *Framework*.
- **FRAMEWORK** - É o modelo da plataforma .NET para construir, instalar e rodar qualquer aplicação, no desktop ou na Internet. Para executar um programa .NET, é preciso ter o *Framework* instalado.
- **IDE COMPARTILHADO** - Ambiente integrado de programação (Integrated Development Environment) do Visual Studio.NET. Diferentes linguagens usam o mesmo editor de código e depurador e compilam executáveis na linguagem MSIL. Além das linguagens da Microsoft, já há mais de 20 outras (Perl, Cobol, Pascal, etc) que podem usar esse ambiente.
- **MSIL - Microsoft Intermediate Language**. Quando se compila uma aplicação .NET, ela é convertida para uma linguagem intermediária, a MSIL, um conjunto de instruções independentes de CPU. Na hora de executar o programa, um novo compilador, chamado Just-in-time (JIT) Compiler, o converte para o código nativo, ou seja, específico para o processador da máquina.

## Termos da plataforma

- **MANAGED CODE** - Código administrado, ou seja, código escrito para rodar com o runtime do VS.NET. No VS.NET, somente o C++ produz programas que não dependem do runtime, o chamado Unmanaged code.

## Visão Geral do .NET Framework

É uma tecnologia que dá suporte à compilação e à execução da próxima geração de aplicativos e serviços Web XML. O *.NET Framework* foi criado para atender os seguintes objetivos:

- Para fornecer um ambiente de programação orientada a objetos consistente, quer o código objeto seja armazenado e executado localmente ou remotamente.
- Fornecer um ambiente de execução que minimize conflitos de versionamento de publicação.
- Fornecer um ambiente de execução que promova a execução segura de código criado por desconhecidos ou código de terceiros com baixo nível de confiança
- Para fornecer um ambiente de execução que elimina os problemas de desempenho dos ambientes interpretados ou com scripts.
- Para tornar a experiência do desenvolvedor consistente, através dos diversos tipos de aplicativos, como aplicativos baseados no Windows e Web.
- Para executar toda comunicação usando padrões da indústria, assim garantindo que códigos baseados no *.NET Framework* possam se integrar a qualquer outro código.



## CLR (Common Language Runtime)

O .NET Framework tem um ambiente de tempo de execução chamado de *Common Language Runtime*, que executa o código e provê serviços que tornam o processo de desenvolvimento mais fácil.

Compiladores e ferramentas expõem as funcionalidades do CLR e habilitam você escrever código que se beneficia desse ambiente de execução gerenciado.

Código que você desenvolve com um compilador de linguagem que tem como alvo o *runtime* é chamado de código gerenciado; ele se beneficia de recursos como integração entre linguagens, tratamento de exceção entre linguagens, segurança aprimorada, suporte a versionamento e implantação, um modelo simplificado para interação entre componentes, e serviços de depuração e de perfil.

O *runtime* automaticamente trata o leiaute de objetos e gerencia referências a objetos, liberando-os quando eles não estão sendo usados. Objetos cujos tempos de vida são gerenciados dessa forma são chamados de dados gerenciados.

A coleta de lixo elimina perdas de memória, bem como alguns outros erros de programação comuns. Se o código for gerenciado, você poderá usar dados gerenciados, dados não gerenciados ou ambos no seu aplicativo do *.NET Framework*.

## CLR (Common Language Runtime)

Devido ao fato de compiladores de linguagens fornecerem seus próprios tipos, como tipos primitivos, você nem sempre pode saber (ou precisa saber) se seus dados estão sendo gerenciados.

O tempo de execução oferece os seguintes benefícios:

- Melhorias de desempenho.
- A capacidade de utilizar facilmente componentes desenvolvidos em outras línguas.
- Tipos extensíveis fornecidos por uma biblioteca de classes.
- Recursos da linguagem como a herança, interfaces, e sobrecarga para programação orientada a objeto.
- Suporte a definição explícita de threads que permite a criação de aplicações *multi-thread* escalonáveis.
- Suporte a manipulação estruturada de exceções .
- Suporte a atributos personalizados.
- Coleta de lixo.
- Uso de *delegates* em vez de ponteiros de função para aumentar a segurança de tipos e segurança.

## Download do Microsoft Visual Studio 2012

Para realizar o download do Microsoft Visual Studio 2012, basta acessar o portal do Dreamspark pelo link <http://etec.onthehub.com>, clicar em *Ferramentas de Desenvolvimento* e clicar sob o software.

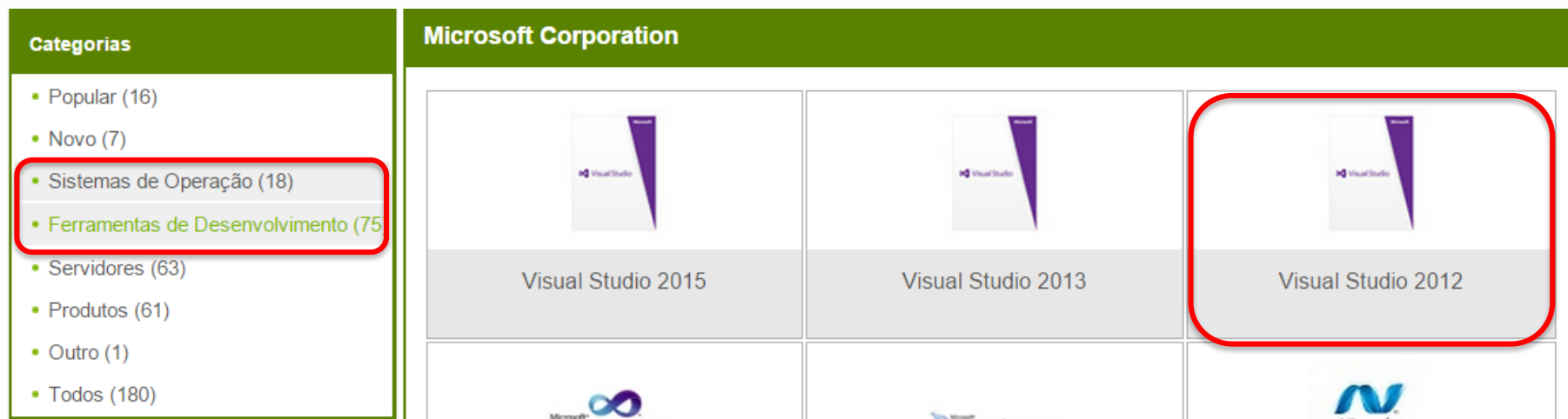
Basta fazer o login com seu e-mail institucional (@etec.sp.gov.br) e seguir as orientações para download.

Após o download, basta executar o arquivo de instalação e seguir as orientações.

## Download do Microsoft Visual Studio 2012

Após o login, clique ao lado esquerdo na opção, **Ferramentas de Desenvolvimento**, e a seguinte tela irá aparecer, como mostra a figura abaixo.

Em seguida clique no ícone **Visual Studio 2012**.



## Download do Microsoft Visual Studio 2012

Na próxima tela clique em

**Visual Studio Premium 2012**

Adicione ao carrinho de Compras

### Visual Studio Premium 2012



Visual Studio Premium 2012 provides business analysts, project managers and testers, alongside developers and users, with the tools to integrate their processes and collaborate seamlessly.

#### Download

Visual Studio Premium 2012 32-bit (English) - DreamSpark

Disponível para: Alunos/Docentes/Funcionários

**Grátis**



**Adicionar ao Carrinho de Compras**



## Download do Microsoft Visual Studio 2012

Na próxima tela clique em

Iniciar Download

Itens Todos os preços são mostrados em Dólares norte-americanos

 Iniciar Download



Visual Studio 2012 Ultimate 32-bit (English) - DreamSpark - Download

Quantidade

\$0.00

Chave do Produto:

Instruções:

Burning the .ISO/.IMG file onto a disc

Mensagens:

- DreamSpark EULA

Número do Contrato: 1203901222

Subtotal:	\$0.00
Tarifas:	\$0.00
Total:	\$0.00

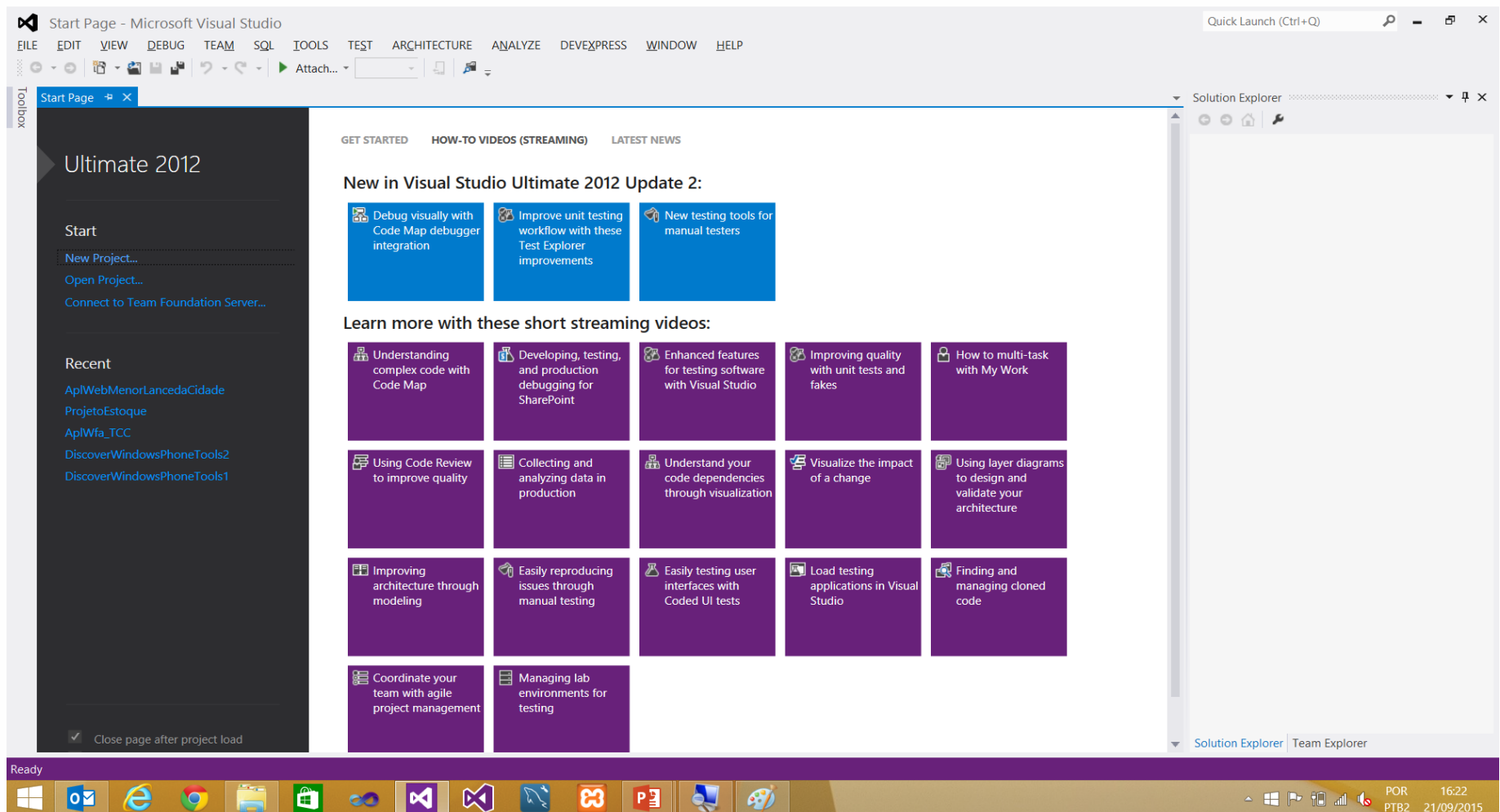


Continuar Compras



Sua Conta/Pedidos

# Conhecendo o ambiente

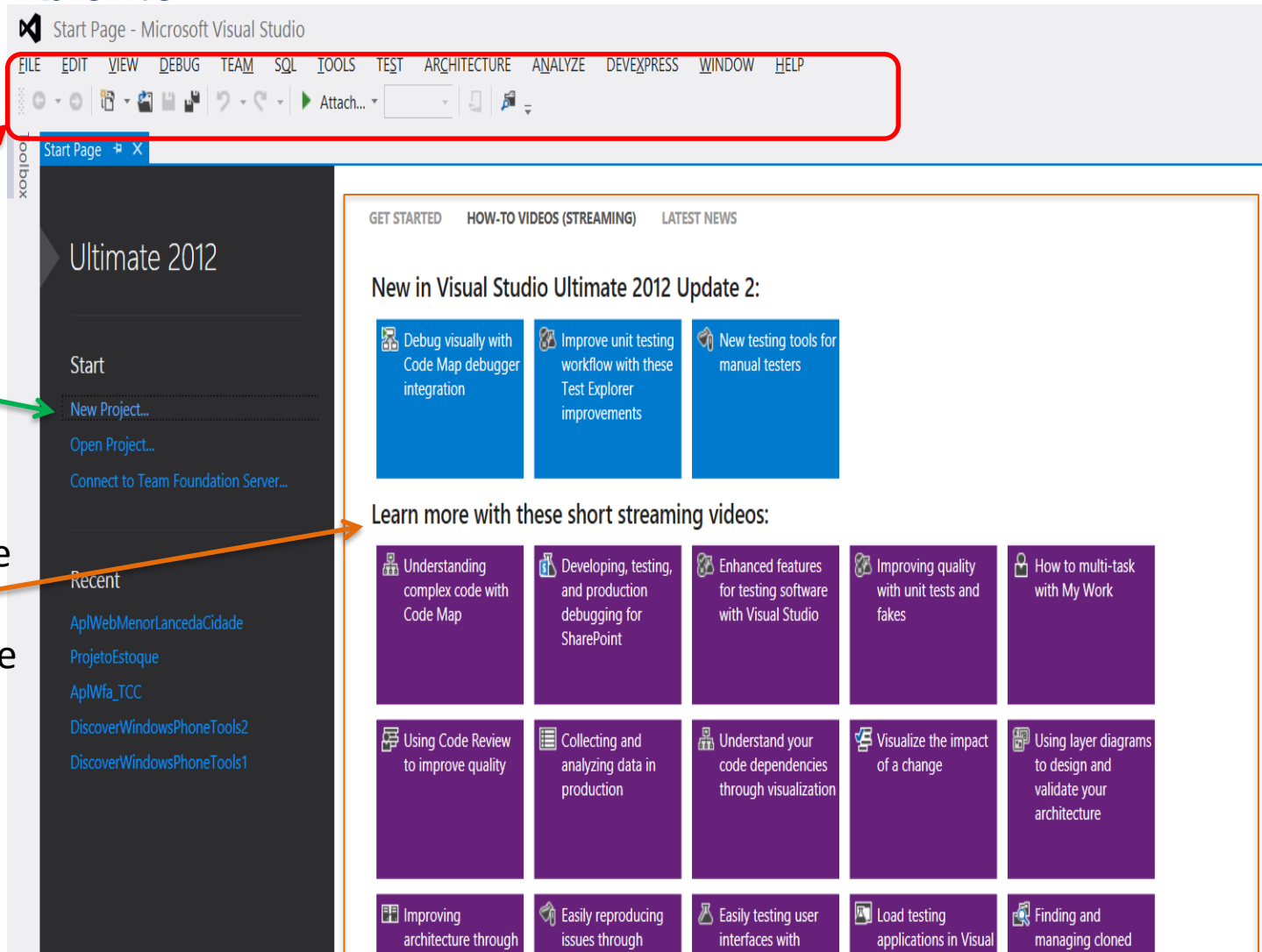


## Conhecendo o ambiente

Barra de Menu e  
Barra de  
Ferramentas

Página inicial para  
criar ou abrir  
projetos

Informações, dicas e  
referências de  
documentação sobre  
as tecnologias

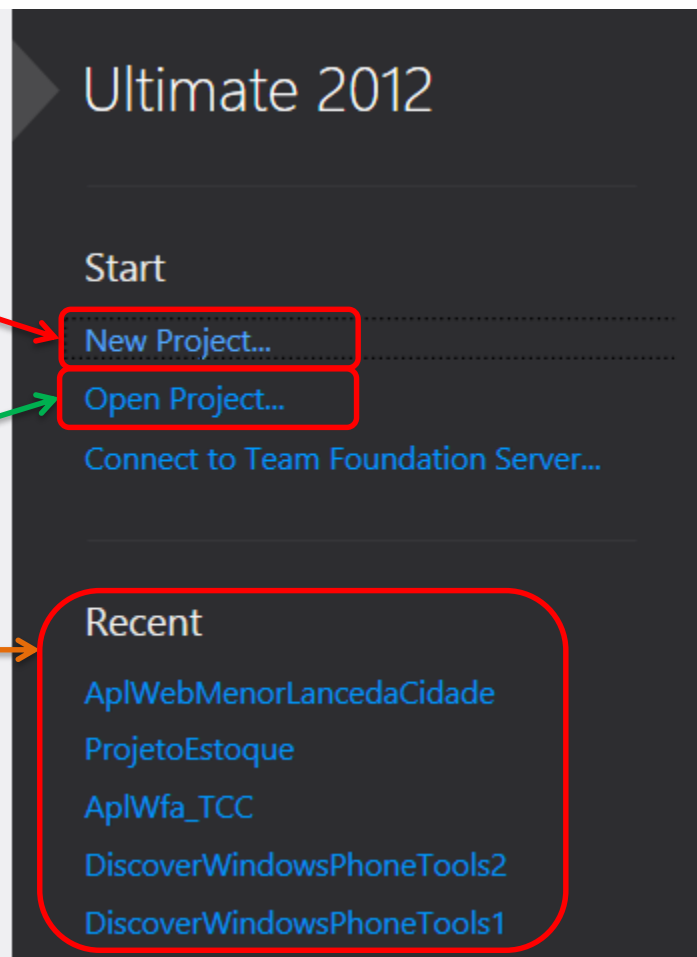


## Conhecendo o ambiente

Opção para criar um projeto novo

Opção para abrir um projeto existente

Lista de todos os projetos abertos recentemente.



## ASP.NET - Conceito

ASP.NET é um modelo unificado de desenvolvimento Web que inclui os serviços necessários para a construção de aplicações web de nível empresarial com um mínimo de codificação.

ASP.NET faz parte do framework .NET , e ao codificar aplicativos ASP.NET você tem acesso as classes no framework .NET. Você pode codificar os aplicativos em qualquer linguagem compatível com o Common Language Runtime (CLR), incluindo Microsoft Visual Basic, C#, JScript .NET e J#. Estas linguagens permitem você desenvolver aplicações que se beneficiam do ambiente integrado de execução, proteção de tipos, herança, e assim por diante.

O ASP.NET inclui:

- Uma estrutura de página e controles
- O compilador do ASP.NET
- Infraestrutura de Segurança
- Facilidades no gerenciamento de estado
- Arquivo de configuração da aplicação
- Recursos para monitoramento da integridade e desempenho
- Suporte a depuração
- Uma estrutura de serviços da Web XML
- Ambiente de hospedagem extensível e gerenciamento de ciclo de vida da aplicação
- Um ambiente de designer extensível



## ASP.NET - Características

A páginas ASP.NET são semelhantes as páginas HTML, mas elas incluem elementos adicionais que o ASP.NET reconhece e processa quando a página é executada.

As características que distinguem uma página ASP.NET de uma HTML, são:

- Extensão da página como **.aspx** para que possa ser processada pelo ASP.NET;
- Um recurso opcional a diretiva **@Page** ou outras diretiva, conforme apropriado para o tipo de página que você está criando.
- O elemento do form está configurado corretamente para ASP.NET. O elemento form só é necessário se a página contém controles cujos valores você deseja usar durante processamento da página.
- Código do servidor , se você adicionar seu próprio código para a página.

## Diretivas @

A páginas ASP.NET possuem diretivas para configurar as propriedades

## ASP.NET – Diretivas @

Diretiva	Descrição
<a href="#">@ Assembly</a>	Vincula um assembly para a página ou controle de usuário atual declarativamente.
<a href="#">@ Controle</a>	Define atributos específicos de controle usados pelo analisador de página ASP.NET e compilador e pode ser incluído somente em arquivos .ascx (user controls).
<a href="#">@ Implements</a>	Indica que uma página ou controle de usuário implementa uma interface especificada do .NET Framework declarativamente.
<a href="#">@ Import</a>	Importa um namespace em uma página ou controle de usuário explicitamente.
<a href="#">@ Mestre</a>	Identifica uma página como uma página mestra e define atributos usados pelo analisador de página ASP.NET e compilador e pode ser incluído somente em arquivos .master .
<a href="#">@ MasterType</a>	Define a classe ou o caminho virtual usado para digitar a propriedade de uma página <a href="#">Master</a>
<a href="#">@ OutputCache</a>	Controla a saída de cache de diretivas de uma página ou controle de usuário declarativamente .

## ASP.NET – Diretivas @

Diretiva	Descrição
<a href="#"><u>@ Page</u></a>	Define atributos específicas de página usados pelo analisador de página ASP.NET e compilador e pode ser incluído somente em arquivos .aspx.
<a href="#"><u>@ PreviousPageType</u></a>	Cria uma referência fortemente tipada para a página de origem do alvo de uma postagem entre páginas .
<a href="#"><u>@ Referência</u></a>	Links à uma página, <i>user control</i> ou controle COM para a página atual ou declarativamente <i>user control</i> .
<a href="#"><u>@ Register</u></a>	Associa alias a namespaces e classes, o que permite que controles de usuário e controles do servidor personalizados sejam processados quando incluídos em uma página ou controle de usuário solicitados.

## ASP.NET – Estrutura de página e controle

O framework de página e controles ASP.NET é um framework de programação que roda em um servidor web para produzir e renderizar dinamicamente as páginas web em ASP.NET. Páginas web em ASP.NET podem ser requisitadas a partir de qualquer browser ou dispositivo cliente, e o ASP.NET renderiza a marcação (como o HTML) para o browser que fez a requisição. Como regra, você pode usar a mesma página para múltiplos navegadores, porque o ASP.NET renderiza o código de marcação apropriado para o navegador que está fazendo a solicitação. No entanto, você pode criar o página da Web do ASP.NET para direcionar um navegador específico, tal como Microsoft Internet Explorer 6, e tirar proveito dos recursos deste navegador. O ASP.NET suporta controles móveis para dispositivos habilitados para Web, como telefones celulares, computadores portáteis e assistentes digitais pessoais (PDAs).

Páginas web em ASP.NET são completamente orientadas a objetos. Dentro de páginas web em ASP.NET você pode trabalhar com elementos HTML usando propriedades, métodos e eventos. A estrutura da página ASP.NET remove os detalhes da implementação da separação de cliente e servidor inerente a aplicativos baseados na Web, apresentando um modelo unificado para responder a eventos do cliente no código que é executado no servidor. O framework também mantém automaticamente o estado de uma página e os controles daquela página durante o ciclo de vida de processamento da mesma.

## ASP.NET – Estrutura de página e controle

O framework de páginas e controles ASP.NET também permite você encapsular funcionalidades de interface comuns em controle fáceis de usar e reutilizáveis. Controles são escritos uma vez, podem ser usados em muitas páginas e são integrados à página da Web do ASP.NET, onde são colocados durante o processamento.

A estrutura de página ASP.NET e controles também fornece recursos para controlar a aparência geral do seu site da Web por meio de temas e skins. Você pode definir temas e skins e depois aplicá-los em um nível de página ou em um nível de controle.

Além dos temas, você pode definir páginas mestre que você pode utilizar para criar um layout consistente para as páginas da sua aplicação. Uma única página mestre define o layout e o comportamento padrão que você deseja para todas as suas páginas (ou grupo de páginas) da sua aplicação. Você pode então criar páginas individuais de conteúdo que contenham o conteúdo específico que você deseja exibir. Quando os usuários fazem uma requisição do conteúdo das páginas, ela é mesclada com a página mestra para produzir uma saída que combina o layout da página mestra com o conteúdo da página.

As páginas Web do ASP.NET podem ser abertas em qualquer navegador ou dispositivo do cliente, executando no lado servidor. As páginas Web são:

- Baseado na tecnologia Microsoft ASP.NET, no qual o código que é executado dinamicamente no servidor gera saída de página Web para o navegador ou dispositivo de cliente.



## ASP.NET – Estrutura de página e controle

- Compatível com qualquer navegador ou dispositivo móvel. Um página da Web ASP.NET automaticamente processa o compilador de navegador HTML correto para recursos como estilos, layout e assim por diante. Como alternativa, você pode projetar suas páginas da web ASP.NET para serem executadas em um navegador específico, como Microsoft Internet Explorer 6 e tirar proveito dos recursos de navegador específico.
- Compatível com todas as linguagens com suporte ao .NET Common Language Runtime, incluindo Microsoft Visual Basic, Microsoft Visual C#, Microsoft J# e Microsoft JScript.NET.
- Construído sobre o Microsoft .NET Framework. Isso proporciona todos os benefícios do framework, incluindo um ambiente gerenciado, tipo de segurança e herança.
- Flexível, pois você pode adicionar controles criados pelo usuário ou de terceiros.

## Componentes de uma página

A página web ASP.NET é dividida em duas partes, componente visual e lógica da programação. O elemento visual consiste em um arquivo contendo marcação estática, como HTML ou controles de servidor ASP.NET ou ambos. A lógica para a página da web ASP.NET consiste em um código que você cria para interagir com a página. O código pode residir em um bloco de script na página ou em uma classe separada.

## Tag ASP

Para podermos executar instruções no servidor devemos utilizar a tag `<% %>`, como mostra o exemplo abaixo:

```
<html>
  <body bgcolor="yellow">
    <center>
      <h2>Curso APS.NET B&aacute;sico</h2>
      <p><% Response.Write(DateTime.Now) %></p>
    </center>
  </body>
</html>
```

## Tipos de variáveis

Abaixo mostra a lista de variáveis disponíveis na linguagem de programação C#.

C# Type	.NET Framework type	Valores possíveis de se armazenar
<b>bool</b>	System.Boolean	Tipo de dados booleano, ou seja, pode ser apenas true ou false
<b>byte</b>	System.Byte	0 a 255 (8 bits)
<b>sbyte</b>	System.SByte	-128 a 127 (8 bits)
<b>char</b>	System.Char	Um caractere (16 bits)
<b>decimal</b>	System.Decimal	Ponto flutuante decimal de 128 bytes (? 1.5 x 10 <sup>-45</sup> a ? 1.7 x 10 <sup>308</sup> ). Este tipo tem uma precisão de 28 casas decimais
<b>double</b>	System.Double	Ponto flutuante binário de 8 bytes (? 1.5 x 10 <sup>-45</sup> a ? 1.7 x 10 <sup>308</sup> ). Este tipo tem uma precisão de 15 casas decimais
<b>float</b>	System.Single	Ponto flutuante binário de 4 bytes (? 1.5 x 10 <sup>-45</sup> a ? 3.4 x 10 <sup>38</sup> ). Este tipo tem uma precisão de 7 casas decimais
<b>int</b>	System.Int32	Inteiro de 32 bits com sinal (-2.147.483.648 a 2.147.483.147)
<b>uint</b>	System.UInt32	Inteiro de 32 bits sem sinal (0 a 4.294.967.295)
<b>long</b>	System.Int64	Inteiro com sinal de 64 bits (-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807)
<b>ulong</b>	System.UInt64	Inteiro sem sinal de 64 bits (0 a 18.446.744.073.709.551.615)
<b>object</b>	System.Object	Qualquer tipo.

## Tipos de variáveis

C# Type	.NET Framework type	Valores possíveis de se armazenar
<b>short</b>	System.Int16	-32,768 a 32,767 (16 bits)
<b>ushort</b>	System.UInt16	0 a 65,535 (16 bits)
<b>string</b>	System.String	Unicode com até 1 gigabyte de caracteres. Dentre os tipos de dados nativos do C#, este é o único cuja passagem (a funções ou métodos) é feita por referência, ou seja, o que é passado não é seu valor, mas um ponteiro indicando o local da memória em que se encontra aquela variável.

## Tipos de Operadores

Podem ser definidos na linguagem C# da seguinte forma:

- **Aritméticos**
- **Unários**
- **Lógicos**
- **Condicionais**
- **Relacionais**
- **Igualdade**



## Operadores Aritméticos

Operador	Uso
+	Efetuar a somar de tipos numéricos. Também usado para concatenar strings.
-	Efetuar a subtração de tipos numéricos
/	Efetuar a divisão de tipos numéricos
*	Efetuar a multiplicação de tipos numéricos
%	Efetuar a divisão de dois números e retornar o resto da divisão.

## Operadores Unários

Operador	Uso
+	Especifica números positivos.
-	Especifica números negativos.
!	Negação booleana. O contrário de.
++ (pré)	Incremento pré-fixado. Primeiro é efetuado o incremento e depois a variável é avaliada. Exemplo: ++contador
++ (pós)	Incremento pós-fixado. Primeiro a variável é avaliada e depois é efetuado o incremento. Exemplo: contador++
-- (pré)	Decremento pré-fixado. Primeiro é efetuado o decremento e depois a variável é avaliada. Exemplo: --contador
-- (pós)	Decremento pós-fixado. Primeiro a variável é avaliada e depois é efetuado o decremento. Exemplo: contador--

## Operadores Lógicos

Operador	Uso
&&	Operador lógico AND (E LÓGICO). Gera resultado verdadeiro se TODOS os operando forem verdadeiros.
	Operador lógico OR (OU LÓGICO). Gera resultado verdadeiro se PELO MENOS um operando for verdadeiro.
!	O contrário de.

## Operadores Relacionais

Operador	Uso
<	Menor que.
<=	Menor ou Igual a.
>	Maior que.
>=	Menor ou Igual a.
is	Comparar em tempo de execução se um objeto é igual a um tipo de componente.

## Operadores Igualdade

Operador	Uso
==	Igualdade
!=	Diferente

## Referências

[https://msdn.microsoft.com/pt-br/library/ms178194\(v=vs.100\).aspx](https://msdn.microsoft.com/pt-br/library/ms178194(v=vs.100).aspx)