Introdução ao ASP.NET

Prof. Me. Sérgio Carlos Portari Júnior
Sergio.junior@uemg.br
http://www.sergioportari.com.br

O que é o ASP.NET?

* Tecnologia da Microsoft para a criação de aplicações dinâmicas para a Web

* Criação de:
  * Websites dinâmicos;
  * Aplicações web;
  * Web services;

O que é o ASP.NET?

* É um ambiente completo de desenvolvimento e criação de aplicativos web.
* É uma nova versão da tecnologia ASP (Active Server Pages).
* Exibido em qualquer browser, sistema ou plataforma, pois sua execução é no servidor.
* Roda no IIS (Internet Information Server) versão 5 ou superior, com a SDK .Net Framework.

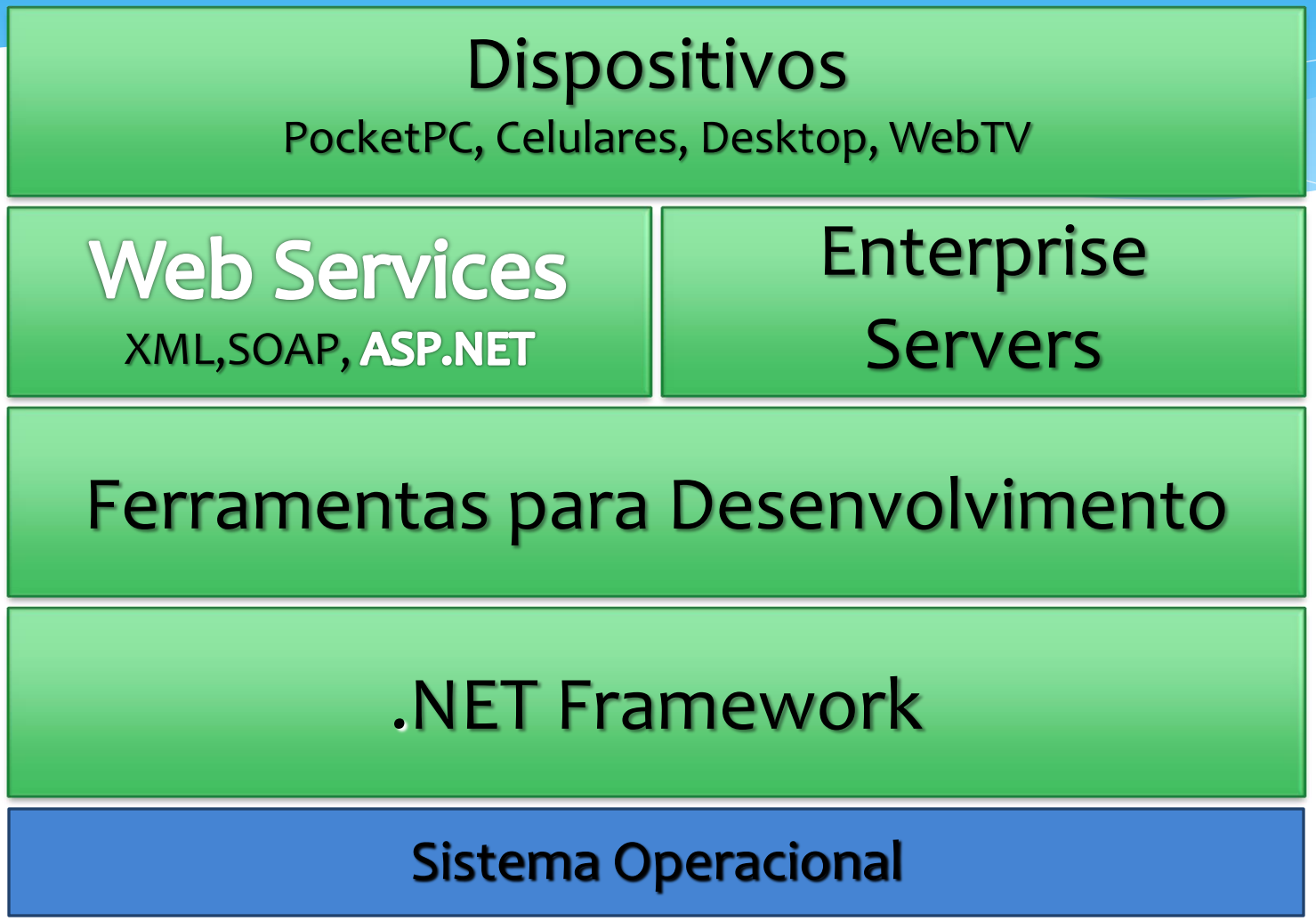
Mais sobre o ASP.NET

* Não é necessário conhecer HTML (mas é recomendado)
* Suporta mais de uma linguagem (C++, C#, VB, etc) simultaneamente.
* As páginas são compiladas, e não interpretadas como o ASP antigo, ou PHP. São geradas DLLs, que são executadas no servidor e produzem um HTML de resultado para o cliente.

Mais sobre o ASP.NET

* Possui uma característica chamada Code Behind (por trás do código), onde é possível separar os layouts e a programação propriamente dita (regras de negócio).
* Pode ser programado em linhas de códigos (qualquer editor de texto) ou utiliza-se uma IDE (recomendado o Microsoft Visual Studio, que desde a versão 2010 possui os recursos de edição do ASP.NET).
* Existe uma versão EXPRESS (gratuita) ou as versões Professinal ou Enterprise para desenvolvimento.

Os 5 pilares do framework .NET



Um pouco mais sobre o ASP.NET

- Suporte a várias linguagens
- Sucessor do ASP
- Controles ricos e programáveis
- Modularização de código
- Manutenção de estados
- Melhor performance (código compilado)
- Programação com eventos
- Componentes baseados em XML
- Mais fácil de configurar

Suporte a várias linguagens

* Utiliza o Common Language Runtime (CLR)
  * C#, VB.NET, J#, C++/CLI, IronPython, IronRuby
  * Características:
    * Garbage collection, administração de threads e memória
    * O debugger funciona com todas as linguagens
* ADO.NET
  * Acesso a banco de dados
    * Microsoft SQL Server, XML, Oracle, OLE DB, ODBC

Um pouco mais sobre o ASP.NET

- Suporte a várias linguagens
- Sucessor do ASP
- Controles ricos e programáveis
- Modularização de código
- Manutenção de estados
- Melhor performance (código compilado)
- Programação com eventos
- Componentes baseados em XML
- Controles de autenticação de usuário
- Mais fácil de configurar

Sucessor do ASP

* O ASP.NET é o sucessor do ASP
  * O ASP foi completamente re-rescrito para tornar-se o ASP.NET
  * Herdou os melhores conceitos do ASP

Um pouco mais sobre o ASP.NET

- Suporte a várias linguagens
- Sucessor do ASP
- Controles ricos e programáveis
- Modularização de código
- Manutenção de estados
- Melhor performance (código compilado)
- Programação com eventos
- Componentes baseados em XML
- Mais fácil de configurar

Alguns tipos de controle..

- Controles que servem elementos HTML
- Controles do servidor Web
- Controles para validação de formulários
  - Checar um item de preenchimento obrigatório em um formulário;
- Controles do usuário
  - Controles criados pelo desenvolvedor;

Controles que servem HTML

- Esses controles são elementos HTML (ou outra linguagem de marcação suportada, como o XHTML) que contêm atributos que os deixam programáveis no servidor. (runat="server")

- Por padrão, os elementos HTML em uma página ASP.NET não são acessíveis no servidor.

- Funcionalidades: orientação a objetos, eventos no cliente e no servidor, manutenção de estado, interação com controles de validação

Alguns tipos de controle..

- Controles que servem elementos HTML
- Controles do servidor Web
- Controles para validação de formulários
  - Checar um item de preenchimento obrigatório em um formulário;
- Controles do usuário
  - Controles criados pelo desenvolvedor;

Controles do servidor Web

* Não é um-para-um em relação aos elementos HTML.

* Exemplos: RadioButtonList, buttons, text boxes, tables, datagrid, menus.

Alguns tipos de controle..

- Controles que servem elementos HTML
- Controles do servidor Web
- Controles para validação de formulários
  - Checar um item de preenchimento obrigatório em um formulário;
- Controles do usuário
  - Controles criados pelo desenvolvedor;

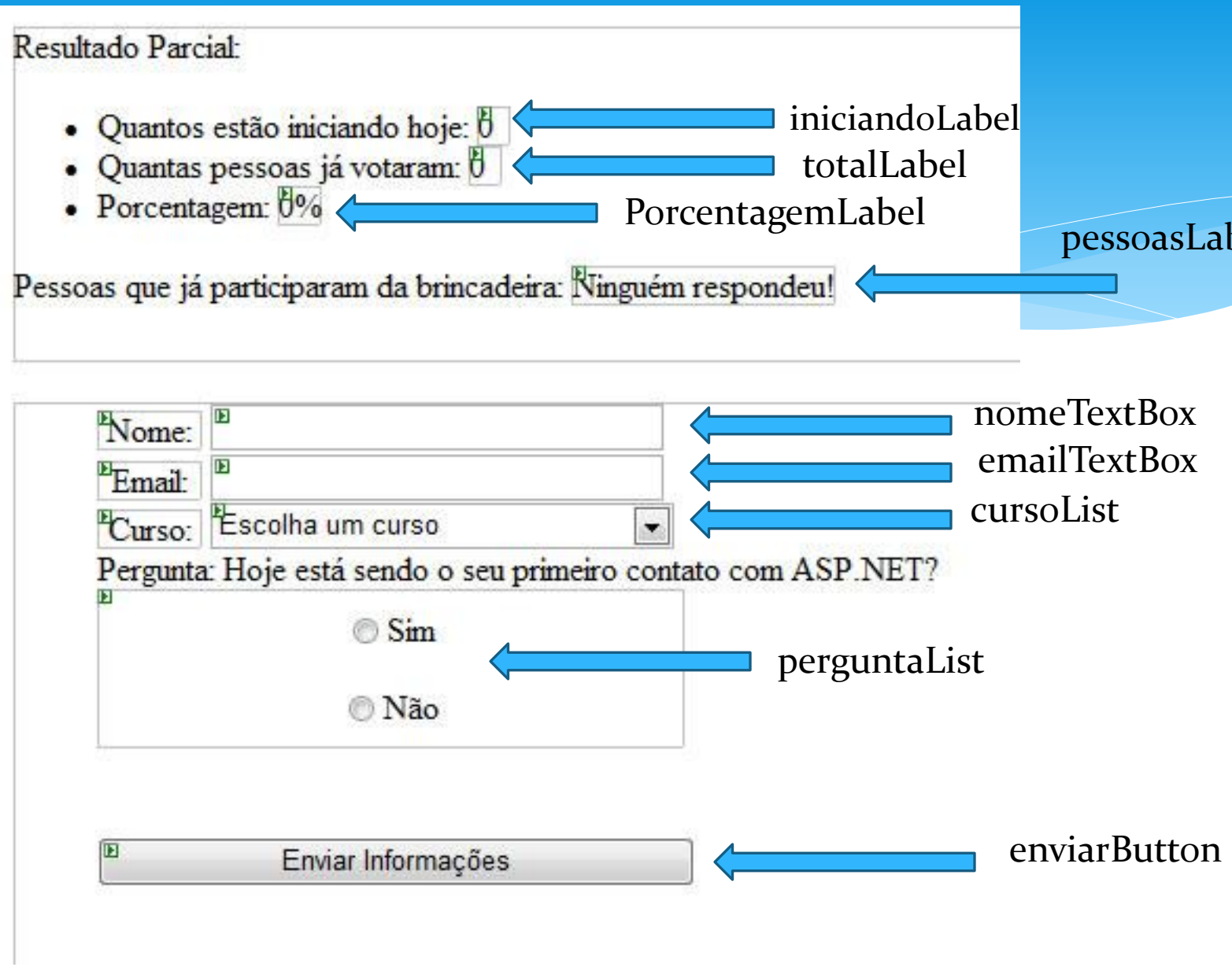
Exemplos de controles do ASP.NET

- Renderiza o HTML de acordo com o agente do cliente

Função	Nome
Mostrar texto	Label
Edição de texto	TextBox
Selecionar de uma lista	DropDownList
	ListBox
Exibir elemento gráfico	Image
	AdRotator
Seleção de valores em um formulário	CheckBox
	RadioButton
Seleção de data	Calendar
Botões	Button
	LinkButton
	ImageButton
Controle de navegação	HyperLink
Controles para Tabelas	Table
	TableCell
	TableRow
Conroles para fazer o agrupamento de outros Controles	CheckBoxList
	Panel
	RadioButtonList
Controles para fazer listagens	Repeater
	DataList
	DataGrid

Demonstração de controles

* Exemplos de uso de controles do ASP.NET com o Microsoft Visual Web Developer ou o Microsoft Visual Studio


Resultado Parcial:
Quantos estão iniciando hoje: 0
Quantas pessoas já votaram: 0
Porcentagem: 0%
Pessoas que já participaram da brincadeira: Ninguém respondeu!
iniciandoLabel
totalLabel
PorcentagemLabel
pessoasLabel
Nome:
Email:
Curso:
Escolha um curso
nomeTextBox
emailTextBox
cursoList
Pergunta: Hoje está sendo o seu primeiro contato com ASP.NET?
Sim
Não
perguntaList
Enviar Informações
enviarButton

Um pouco mais sobre o ASP.NET

- Suporte a várias linguagens
- Sucessor do ASP
- Controles ricos e programáveis
- Modularização de código
- Manutenção de estados
- Melhor performance (código compilado)
- Programação com eventos
- Componentes baseados em XML
- Mais fácil de configurar

Modularização de código

Camada Apresentação

.aspx

Camada lógica da aplicação

C#

Camada acesso aos dados

C# +

ADO.NET

ou outros

Um pouco mais sobre o ASP.NET

- Suporte a várias linguagens
- Sucessor do ASP
- Controles ricos e programáveis
- Modularização de código
- Manutenção de estados
- Melhor performance (código compilado)
- Programação com eventos
- Componentes baseados em XML
- Mais fácil de configurar

Manutenção de estados

- Armazenar informações por um certo intervalo de tempo
- Podemos criar um objeto e ele ficar na memória (não será destruído após a página ter sido enviada para o cliente)
- O objeto pode ser criado para um usuário ou para toda a aplicação
- Application State: visível por toda a aplicação
- Session State: alocado para cada usuário

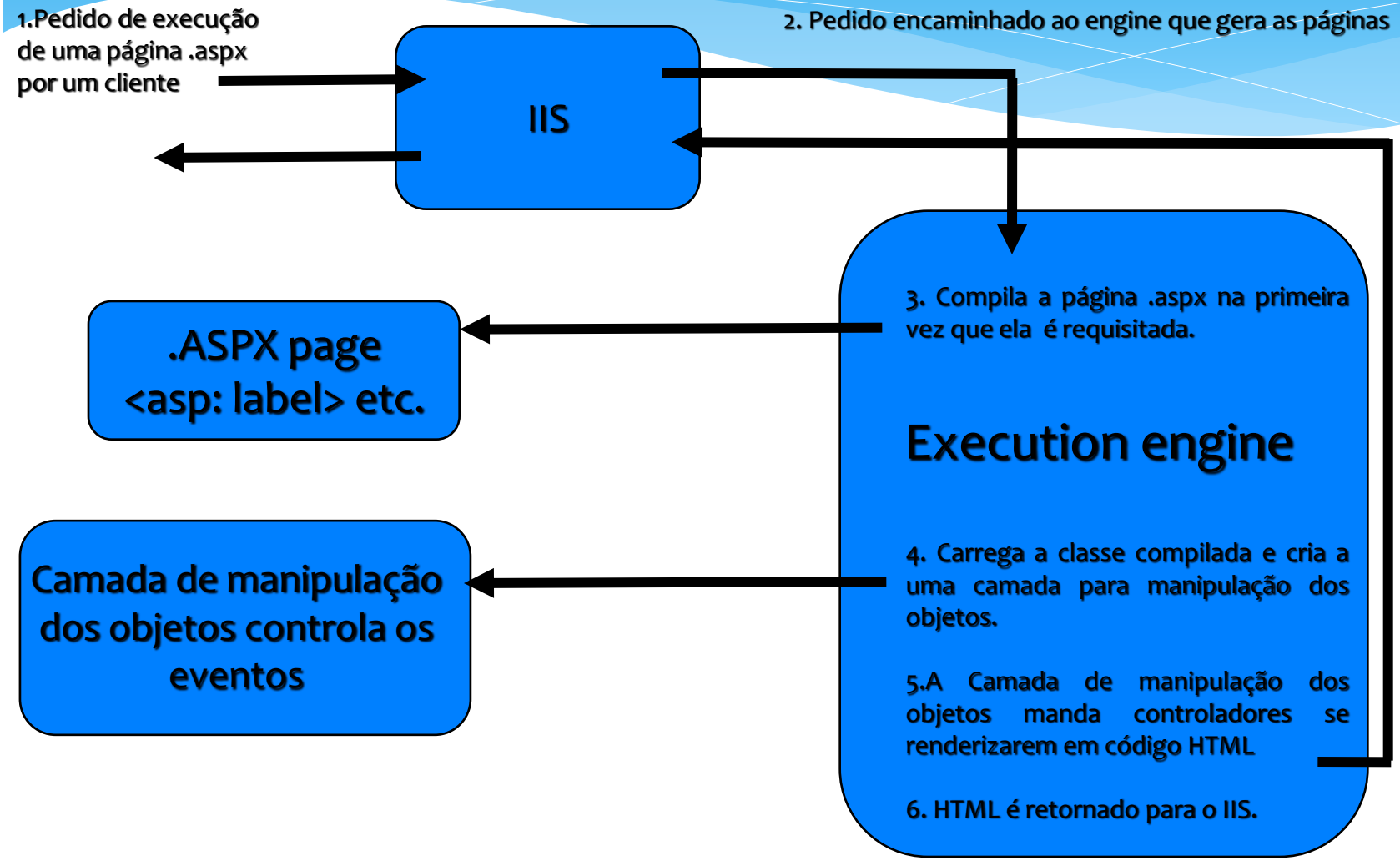
Estocagem do Session State

* Escolhe-se no web.config como será o modo de armazenamento. Pode ser:
  * Em Processo (InProc)
    * Padrão, será armazenado na memória.
  * For a do processo (StateServer)
    * Usa-se um servidor separado só para a estocagem.
  * Banco de Dados (SQLServer)
    * Um banco de dados SQL irá armazenar.

Um pouco mais sobre o ASP.NET

- Suporte a várias linguagens
- Sucessor do ASP
- Controles ricos e programáveis
- Modularização de código
- Manutenção de estados
- Melhor performance (código compilado)
- Programação com eventos
- Componentes baseados em XML
- Mais fácil de configurar

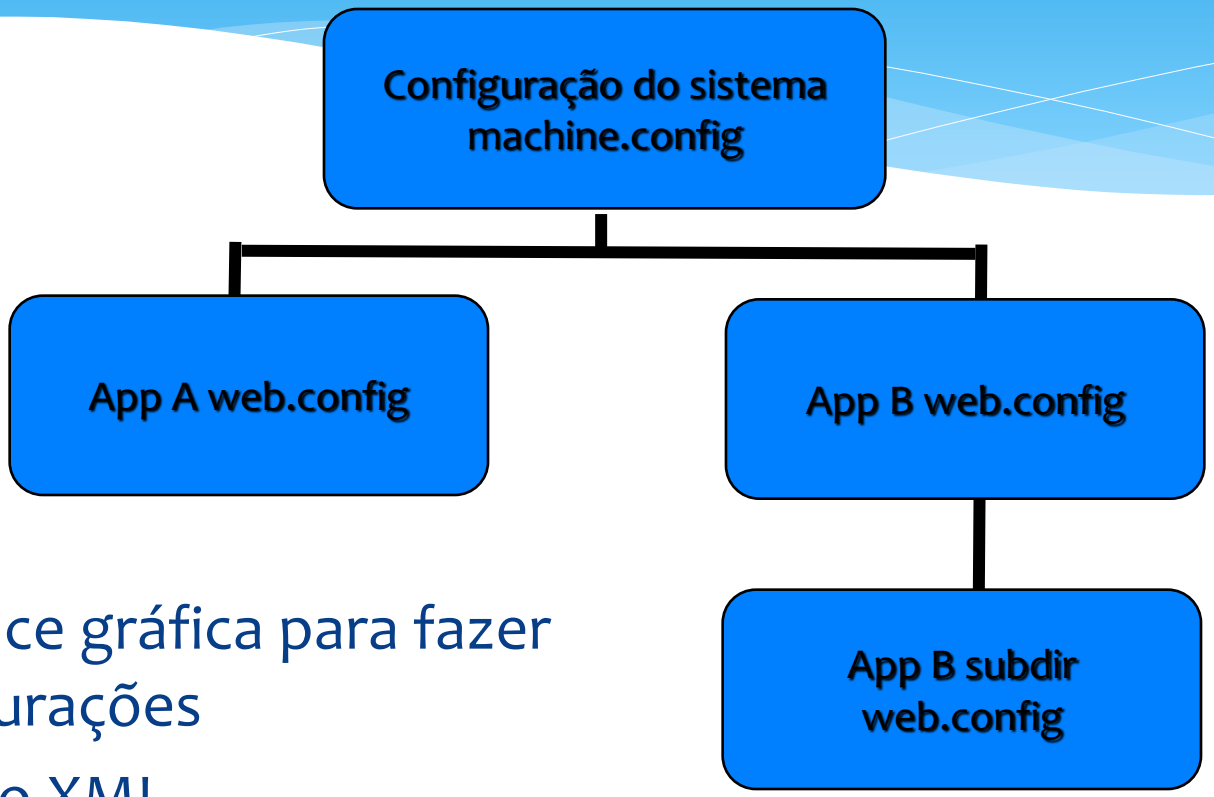
Engine de execução de uma página



Um pouco mais sobre o ASP.NET

- Suporte a várias linguagens
- Sucessor do ASP
- Controles ricos e programáveis
- Modularização de código
- Melhor performance (código compilado)
- Programação com eventos
- Componentes baseados em XML
- Mais fácil de configurar

Mais fácil de configurar



* Interface gráfica para fazer configurações
* Arquivo XML
* Estrutura hierárquica

Alguns atributos do web.config

- <authentication>
  - Como o ASP.NET deve autenticar ou identificar usuários
  - Windows, Forms, Passport, None
- <authorization>
  - Permitir ou negar o acesso de alguns usuários
- <compilation>
  - Debug, defaultLanguage, batch, tempDirectory
  - <compilers>, <assemblies>
- <customErros>
  - Como tratar erros
  - Mode, defaultRedirect

Alguns atributos do web.config

* <connectionString>
  * String de conexão para um banco de dados
* <mailSettings>
  * Host, password, user
* <configSections>
  * Especificar definições customizadas, criadas pelo desenvolvedor

ASP.NET, HTML e C#

* As páginas ASP.NET produzem arquivos de extensão ".aspx". Normalmente, elas são divididas em duas seções principais:

```
1   <%@ Page Language="C#" %>
2   <script runat="server">
3      void limpar(object sender, EventArgs e) {
4         Label2.Text = " ";
5      }
6   </script>
7
8   <html>
9   <body>
10     <form runat="server">
11        <div align="center">
12           <asp:Label id="Label1" runat="server" font-size="Large" font-bold="True"> Página de
13  Exemplo</asp:Label>
14           <asp:Label id="Label2" runat="server">Esta é uma página de exemplo, onde é exemplificada
15  a divisão entre o código HTML e ASP.NET</asp:Label>
16           <asp:Button id="Button1" onclick="limpar" runat="server" text="Limpar"/>
17        </div>
18     </form>
19  </body>
20  </html>
```

Código ASP.NET

Código HTML

ASP.NET, HTML e C#

* Nota-se, que o elemento script possui um atributo **runat**, que possui o valor **server**.
* Isto significa que todo o conteúdo deste elemento estará sendo executado no servidor.
* Isto é necessário pelo fato de ser o servidor, o lugar onde está instalado o .NET framework (o cliente não precisará tê-lo instalado).

ASP.NET, HTML e C#

* A linguagem C# possui muitas semelhanças com as linguagens C, C++ e Java.
* Assim, o código é *case-sensitive* (existe diferença entre letras maiúsculas e minúsculas), e declarações de variáveis, operadores e estruturas de controle são utilizados praticamente da mesma maneira que nestas linguagens.
* Por exemplo, para se declarar uma variável do tipo inteiro, basta escrever o seguinte:

```
int variavel;
```

ASP.NET, HTML e C#

* A forma do código ASP.NET se comunicar com o código HTML é através de funções que são invocadas por controles de servidor (que serão visto adiante, em detalhes), por meio dos eventos (onclick por exemplo).
* Outra forma desta comunicação acontecer ocorre no momento em que a página é carregada, por meio da função Page_Load.

Mais sobre C#: Tipos de dados

* Os tipos de dados utilizados em ASP.NET, no nosso caso, utilizando C# como base da programação, serão os mesmos de C#.
* Se optar por utilizar o VB.Net, ou outra linguagem, os tipos de dados serão os tipos da linguagem escolhida.
* Veja a tabela com os tipos de dados em C#:

Tipo	Descrição
byte	Inteiro de 1 bit sem sinal (0 a 255).
sbyte	Inteiro com sinal de 8 bits (-127 a 128).
int	Inteiro de 32 bits com sinal (-2.147.483.648 a 2.147.483.147).
uint	Inteiro de 32 bits sem sinal (0 a 4.294.967.295).
long	Inteiro com sinal de 64 bits (-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807).
ulong	Inteiro sem sinal de 64 bits (0 a 18.446.744.073.709.551.615).
short	Inteiro com sinal de 16 bits (-32.768 a 32.767).
ushort	Inteiro sem sinal de 16 bits (0 a 65.535).
decimal	Ponto flutuante decimal de 128 bytes (? 1.5×10^{-45} a ? 1.7×10^{308}). Este tipo tem uma precisão de 28 casas decimais.
double	Ponto flutuante binário de 8 bytes (? 1.5×10^{-45} a ? 1.7×10^{308}). Este tipo tem uma precisão de 15 casas decimais.
float	Ponto flutuante binário de 4 bytes (? 1.5×10^{-45} a ? 3.4×10^{38}). Este tipo tem uma precisão de 7 casas decimais.
bool	Tipo de dados booleano, ou seja, pode ser apenas *true* ou *false*.
char	Um único caractere unicode de 16 bits.
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.

Mais sobre C#: Decisões

* **if ... else if ... else** – estruturas de decisão simples, em que a cláusula if ocorre apenas uma vez, a cláusula else if pode ocorrer nenhuma ou várias vezes e a cláusula else é opcional.

```
if (condição1) {
    instruções1;
} else if (condição2) {
    instruções2;
} else {
    instruções3;
}
```

Mais sobre C#: Decisões

* **switch … case** - estruturas de decisão caracterizadas pela possibilidade de uma variável possuir vários valores. A cláusula switch ocorre uma vez, a cláusula case pode ocorrer de uma a várias vezes, e default é opcional.

```
switch (variável) {
    case "1º valor que a variável pode assumir": instruções1; break;
    case "2º valor que a variável pode assumir": instruções2; break;
    case "3º valor que a variável pode assumir": instruções3; break;
    default: instruções para condições não previstas explicitamente;
}
```

Mais sobre C#: Repetições

* **for** - estrutura de repetição caracterizada pela existência de três parâmetros: um valor inicial, uma condição para parada das iterações, e a quantidade de incrementos ou decrementos a cada iteração.

```
for (i=0; i>valor; i++)  {
    instruções;
}
```

Mais sobre C#: Repetições

* **foreach** – esta estrutura de repetição é uma variação do for. Sua diferença está no fato de ser necessário apenas especificar uma variável inteira e a coleção (array) cujos registros serão percorridos.

```
foreach (int i in vetor)  {
    instruções;
}
```

Mais sobre C#: Repetições

* **while** - estrutura de repetição que realiza as operações desejadas enquanto a condição especificada for verdadeira.

```
while (condição) {
   instruções;
}
```

Mais sobre C#: Repetições

* **do ... while** - estrutura de repetição semelhante à anterior, com o diferencial de que as condições são verificadas no final da execução, permitindo as operações especificadas sejam executadas pelo menos uma vez.

```
do {
    instruções;
} while (condição);
```

Mais sobre C#: Operadores

Categoria	Operadores
Aritméticos	+, -, *, /, %
Atribuição	=, +=, -=, *=, /=
Concatenação	+
Criação de Objetos	New
Igualdade/Diferença	==, !=
Incremento/Decremento	++, --
Lógicos	&, ^, |, &&, ||, ~, !
Primários	typeof, sizeof, checked, unchecked
Relacionais	<, >, <=, >=, is

Referências

* http://www.w3schools.com/aspnet/aspnet_vsasp.asp
* http://www.asp.net/learn/videos/
* http://www.devhood.com/tools/tool_details.aspx?tool_id=930
* http://www.devhood.com/tools/tool_sub.aspx?sort=date_submitted&order=desc&page_number=1&category_id=5
* http://msdn.microsoft.com