

# *.Net Framework*



Aluno

Luiz GUSTavo Martins

Matéria

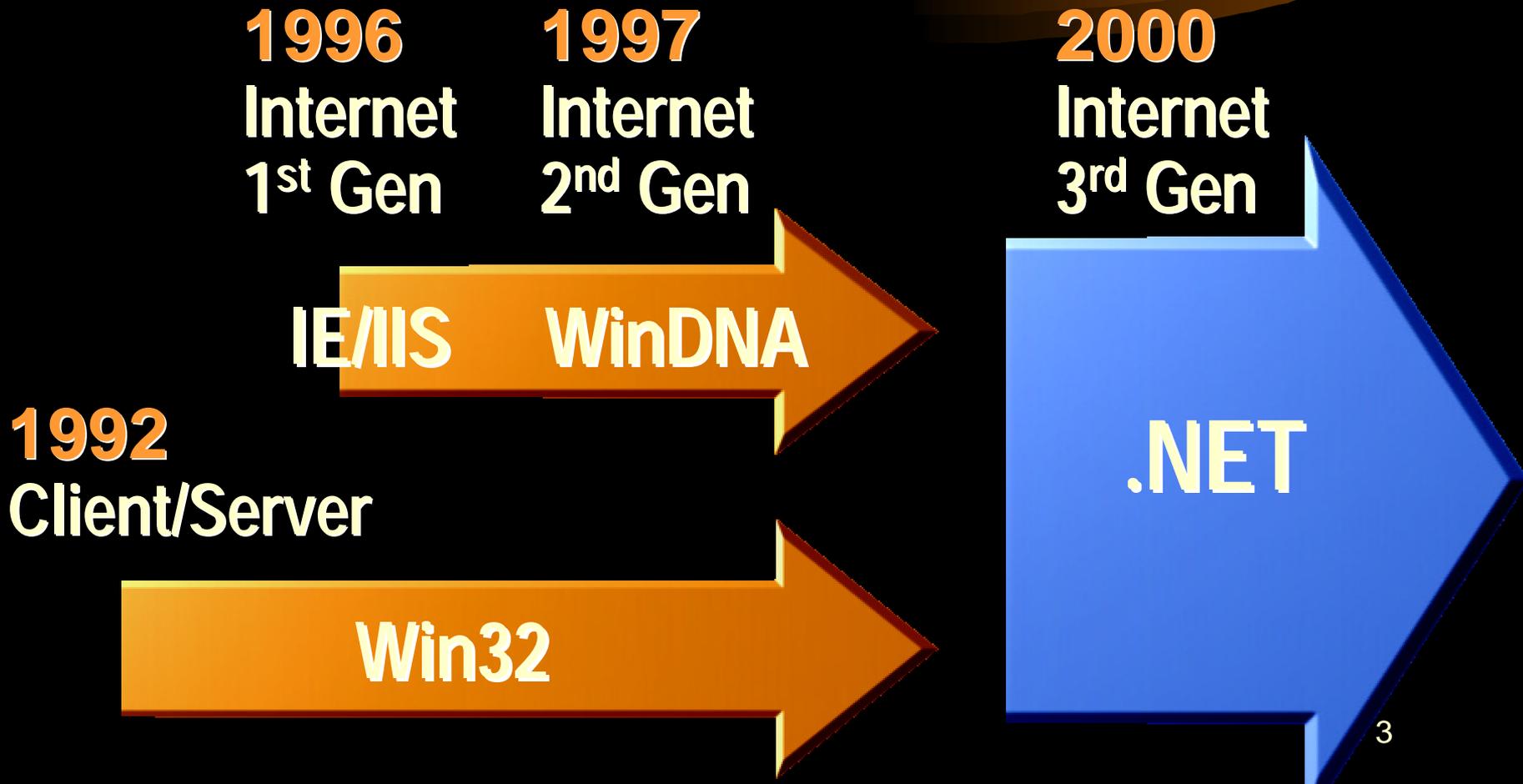
Sistemas de Objetos Distribuídos

# *Roteiro*



- Origem
- O que é .NET , Características e Como funciona
- CLR
- Web Forms
- Web Services
- Servidores .NET
- Pros e Contras
- Referências

# *Origem do .NET*



# *O que é .NET?*

- Fábula dos 3 cegos
- É só uma nova linguagem?
- É um Visual Studio novo?
- É só um ambiente para troca de mensagens e dados baseados em SOAP e XML?
- É tudo isso e muito mais!!!

# *O que É .NET Framework?*



- Plataforma para desenvolvimento e execução de software
- Criada visando o desenvolvimento de software para a internet e aplicações distribuídas

# *Características da Plataforma*

*.Net*

- **Independência de Sistema Operacional**
  - A plataforma compila o programa para código nativo antes da primeira execução
- **Integração entre Linguagens**
  - Todas as linguagens .NET utilizam a mesma API de desenvolvimento

# *Características da Plataforma*

*.NET*

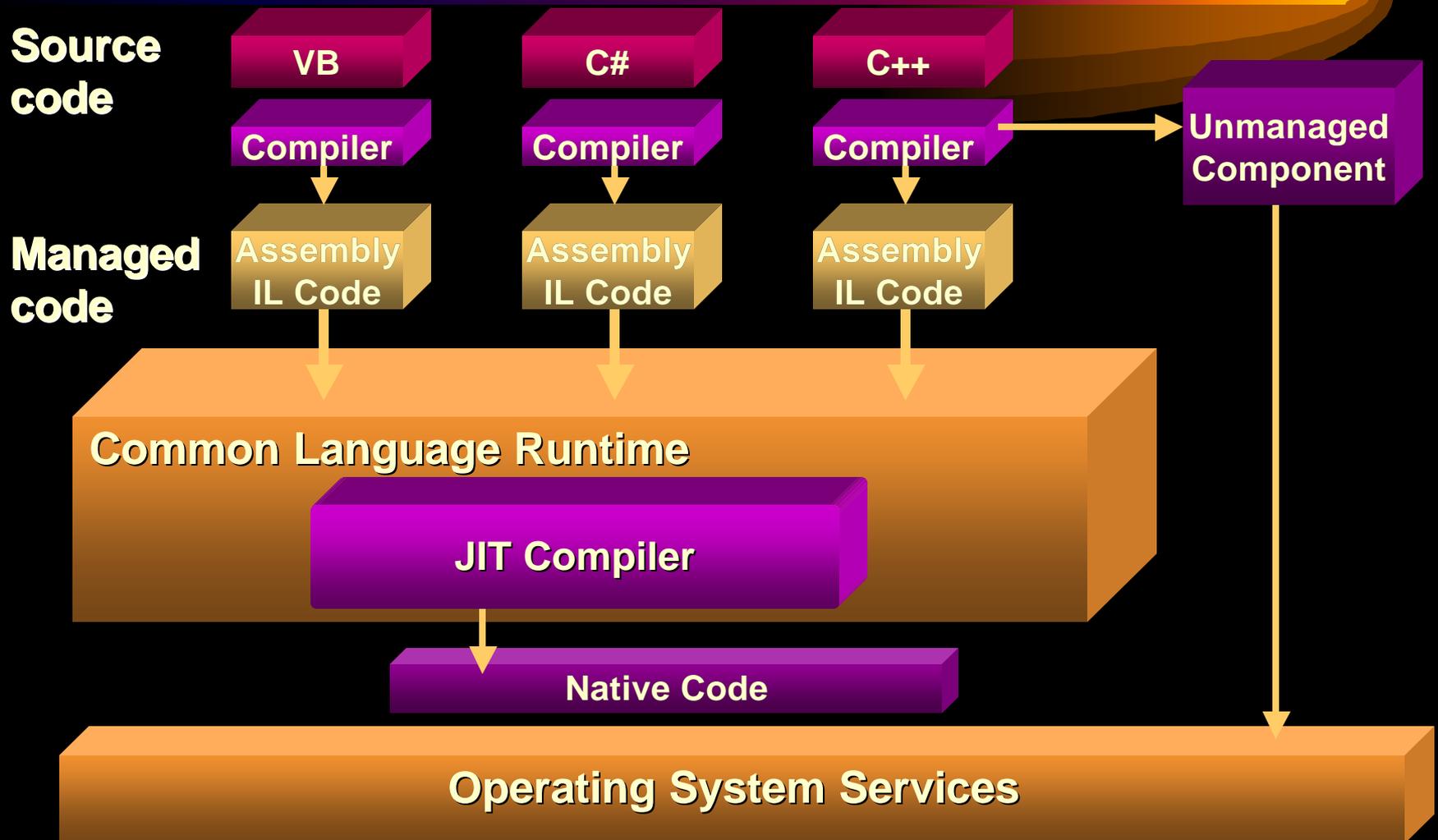
- Facilidade de Distribuição
  - Ao ser compilado os executáveis anotam as versões das DLL's que foram utilizadas
  - Os executáveis rodam utilizando sempre as DLL's com que foram compilados
  - Não é necessário mais registrar o programa. O “assembly” contém toda informação necessária para o funcionamento do programa

# *Características da Plataforma*

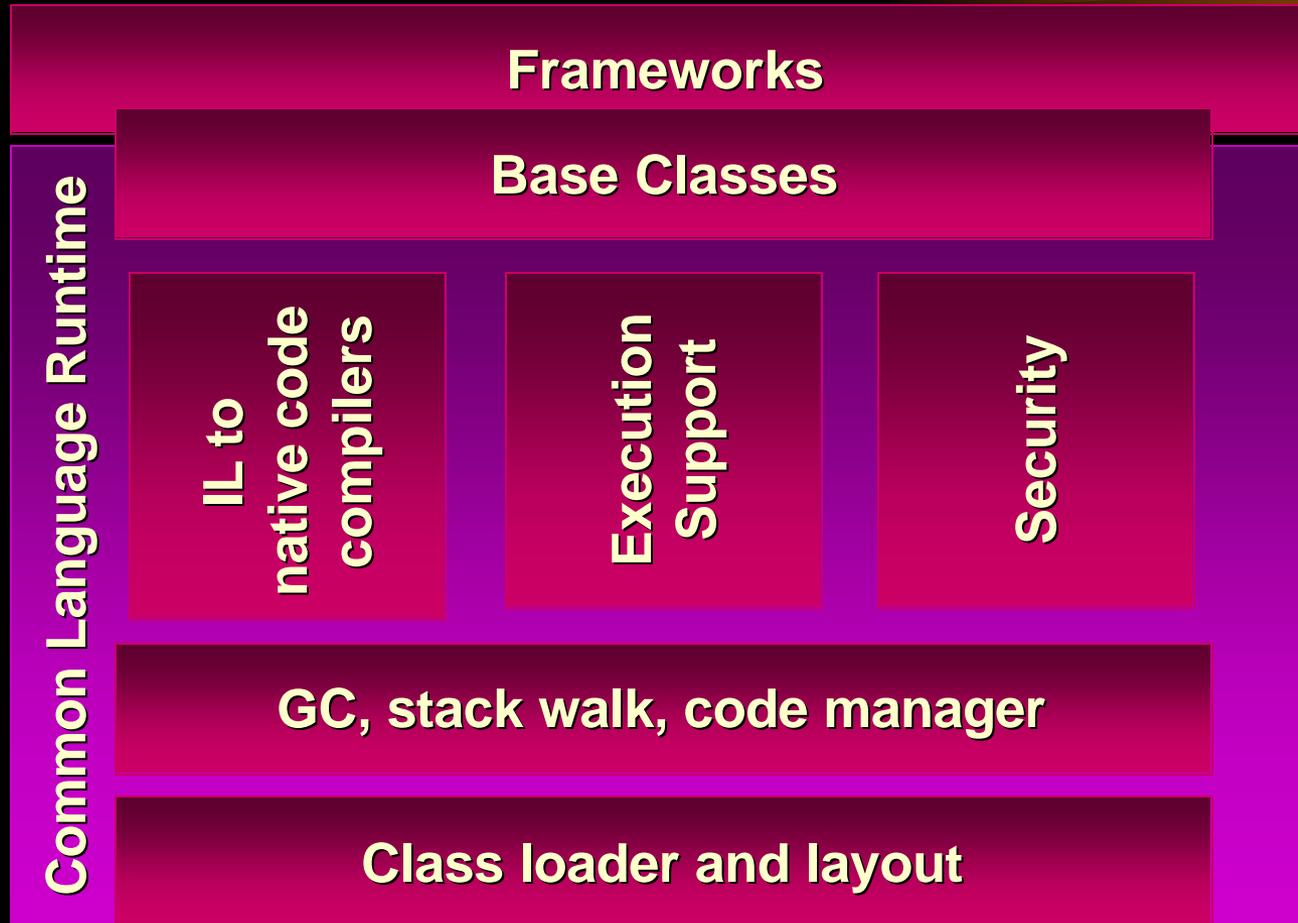
*.NET*

- **Confiabilidade**
  - Tipagem forte (“type safety”) , todas as classes .NET derivam de uma mesma classe mãe, Object
  - Coleta de lixo em todas linguagens .NET

# Como Funciona?



# CLR



- CLR – Common Language Runtime
  - Foi escrita do 0
  - Controla a execução de código no .NET Framework
  - Compila código IL para código nativo da máquina antes da primeira execução do programa
  - Carrega classes necessárias para o programa sendo executado
  - Reforça as checagens de segurança
  - Não é uma Máquina Virtual como em Java

# *Como Funciona?*

- EXEs e DLLs tem um formato específico de arquivo (PE – Portable Executable)
- Windows consegue executar e carregar EXEs e DLLs porque entende esse formato
- Os programas .NET tem alguns dados a mais que avisam o sistema operacional que a CLR deve ser ativada e a partir disso cuida do carregamento e execução desse arquivo e de todos componentes requisitados por ele

# Como Funciona?

- Gerando o Assemblie



# *Executavel Portátil*

- Exemplo

## **MyExample.DLL**

**Metadata**

**IL  
Managed  
code**

**Resources**

# Meta-Dados

- “Dados sobre os dados”
- Inclui informações sobre tipos que define ou que utiliza
- Informação sobre versão
- Referência a *assemblies* externos com dll que são usadas e suas respectivas versões
- Descrições sobre módulos, classes, campos, propriedades, eventos, etc

# Meta-Dados

- A CLR usa para verificação, reforço de segurança, *layout* da memória e para a execução
- O Carregador de Classes utiliza para achar e carregar classes .NET ja que os meta-dados contém informação de onde essas classes se encontram, mesmo que em outro *assembly* na rede
- JIT usa meta-dados para compilar o programa antes da primeira execução

# *Meta-Dados*

- Classes reflexivas podem ler os meta-dados em tempo de execução
- Sem meta-dados a plataforma não poderia prover verificação de tipos, segurança, administração de memória, depuração, etc
- Resumindo não existiria .NET sem meta-dados

# *I.L. – Intermediate Language*

- Linguagem totalmente orientada a objetos
- Projetada para ser compilada e não interpretada
- A IL é para a plataforma .NET o que o bytecode é para Java

# *Operações entre as linguagens*

- Um componente é criado usando a linguagem A e é chamado MyNamespace.dll
- Em um componente separado escrito na linguagem B nós podemos importar o assembly...
  - #using MyNamespace.dll
- E então podemos acessar a classe normalmente
  - using namespace MyNamespace;
  - MyClass \*cls = new MyClass();
  - result = cls->MyMethod();

# *Características Entre As Linguagens*

- Suporte entre linguagens e componentes
  - Herança
  - Tratamento de Exceções
  - Depuração
- Bom para estender componentes de terceiros usando a linguagem de sua escolha

# *Exemplo em Managed C++*

```
#using <mscorlib.dll>
using namespace System

void main()
{
    Console::WriteLine(L"visite
www.gustheman.com !!");
}
```

# *Exemplo em VB.NET*

```
Import System
```

```
Public Module modmain
```

```
    Sub Main()
```

```
        Console.WriteLine("visite www.gustheman.com !!")
```

```
    End Sub
```

```
End Module
```

# *Exemplos em C#*



```
using System;
```

```
class MainApp
```

```
{
```

```
    public static void Main()
```

```
    {
```

```
        Console.WriteLine("visite www.gustheman.com !!");
```

```
    }
```

```
}
```

# *Exemplo em IL*

```
.class private auto ansi MainApp
Extends [mscorlib]System.Object
{
    .method public hidebg static
        void Main() cil managed
    {
        .entrypoint
        .maxstack 8
    }
}
```

# Exemplo de IL

```
ldstr "visite www.gustheman.com!!"  
call void [mscorlib]System.Console::WriteLine(  
  classe System.String )  
ret  
} // fim do metodo MainApp::Main  
.method public hidebysig specialname  
  rtspecialname instance void .ctor() cil managed  
{  
  .maxstack 8
```

# *Exemplo de IL*

```
ldarg.0  
call instace void  
[mscorlib]System.Object::.ctor()  
ret  
} // fim do metodo MainApp::.ctor  
} // fim da classe MainApp
```

# Web Forms

- Nova maneira de se desenvolver páginas HTML
- A página é compilada na primeira execução e a versão compilada é usada nas demais chamadas
- Desenvolvimento fácil, parecido com o desenvolvimento antigo do VB ou Visual C++
- A Página gerada só tem HTML padrão, podendo ser vista em qualquer navegador (browser) sem problemas

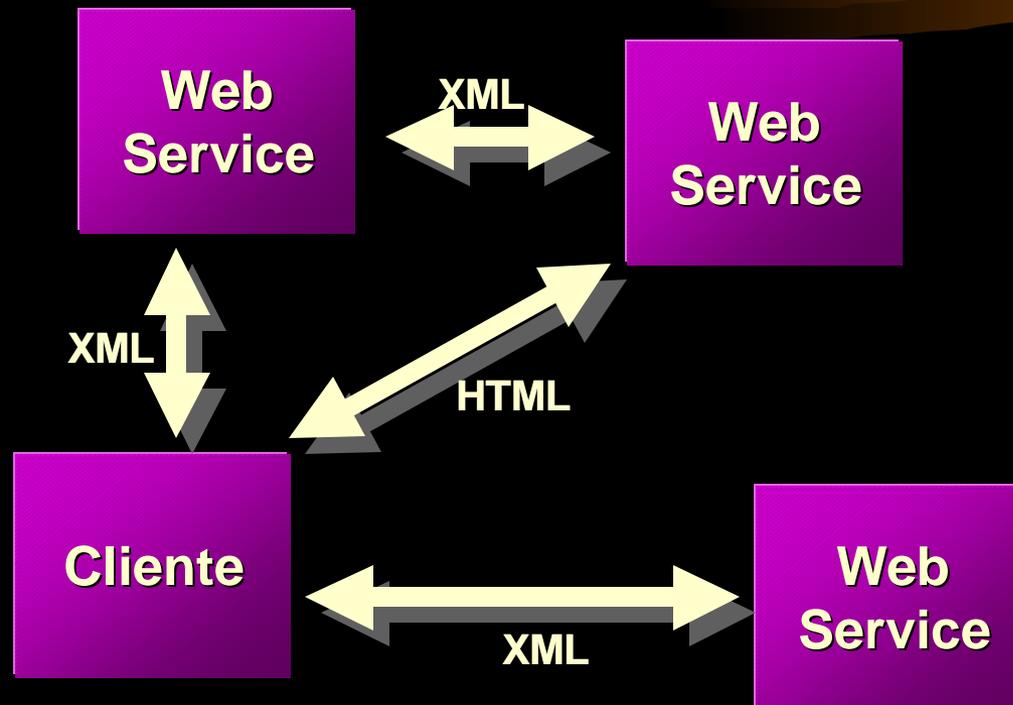
# *Web Forms*

- Separação clara entre a lógica da página (código) e o design (HTML)
- A parte lógica pode ser escrita em qualquer linguagem .NET
- Código ASP antigo ainda é suportado
- Melhor controle de sessão, permitindo maior escalabilidade

# Web Services

- Software como um Serviço
- Possibilita o uso de componentes através do protocolo padrão de HTTP
- Software podem se comunicar usando XML e Internet
- Micro\$oft ja disponibiliza o serviço de autenticação Passport
- O projeto HailStorm é uma evolução do Passport, com muito mais serviços

# Web Services



# Web Services

- UDDI – Universal Description, Discovery and Integration – Páginas Amarelas de Web Services
- SOAP + XML
- WSDL – Web Services Description Language –  
Descrição da interface oferecida pelo web service
  - Feita em XML
  - Parecido com IDL (COM)

# *Smart Devices (aparelhos espertos)*

- O que torna um aparelho esperto é o fato de ele poder acessar Web Services
- Possibilita o usuário interagir com seus dados não importando o lugar, o tipo nem o número de aparelhos que se usa
- Dentre os aparelhos espertos temos os PCs, laptops, telefones, palms e até mesmo o Xbox

# *Servidores .NET*

- Servidores que ajudam na distribuição e hospedagem de software e serviços .NET
- Na família de servidores oferecidos temos:
  - SQL server 2000, alta disponibilidade e suporte a XML nativo
  - Application center, servidor que controla grupos de servidores, tornando mais fácil a administração

# *Servidores .NET*

- Mais Servidores
  - Exchange – servidor de mails
  - Biz talk – fornece ferramentas necessárias para integração de aplicativos heterogêneos.
  - Host Integration Server – Integra novos sistemas com os dados e recursos de mainframe

# *.Net Pros*

- Tem suporte a múltiplas linguagens
  - C#, Managed C++, VB.NET, ASP.NET
  - Perl, Cobol, Pascal estão sendo portados por outras empresas
- É amigável a padrões abertos (SOAP, HTML, XML)
- O código é compilado, não importando a linguagem ou o método de distribuição utilizados

# *.Net contras*

- É mais uma plataforma de desenvolvimento, o que significa reescrever código e aprender novos truques
- Atualmente só disponível para Windows

*Perguntas?*



# Referências

- Site oficial : [www.microsoft.com/net](http://www.microsoft.com/net)
- [www.linhadecodigo.com.br](http://www.linhadecodigo.com.br)
- [www.gotdotnet.com](http://www.gotdotnet.com)
- [www.oreilly.com](http://www.oreilly.com)
- Livro .Net Framework Essentials
  - Thuan Thai & Hoang Q. Lam – O'REILLY
- MSDN
- [www.google.com](http://www.google.com)