
Web Services

Plataforma .Net

Visual Studio 2005/2008/2010

■ **Web Services na plataforma .NET**

- Directiva `<%@ WebService . . . %>` e código do *Web Service*
 - Código *Inline*
 - Código *Pre Compiled*
 - Código *Behind*
- Utilização do *Visual Studio 2005/2008/2010*
 - Criar um *Web Service*
 - Criar um cliente para o *Web Service*

Código Inline

- O ficheiro com extensão `.asmx` é colocado numa directoria virtual associada ao IIS e inclui o código da classe que implementa as operações do serviço;
- O ficheiro começa com a directiva `WebService` e com os atributos `Language` e `Class`:

`<%@ WebService Class="nome qualificado da classe" Language="C#" %>`

```
<%@ WebService Class="SomaNS.Soma" Language="C#" %>

using System.Web.Services;

namespace SomaNS {

    [WebService (Description="SomaWS-Inline", Namespace="http://deetc.isel.ipl.pt/SD/")]
    public class Soma {

        [WebMethod (Description="Calcula a Soma de dois inteiros")]
        public int add(int op1, int op2) {
            return op1+op2;
        }
    }
}
```

SomaWS-Inline.asmx

Código Pre Compiled

- O ficheiro com extensão `.asmx` só contém a directiva `<%@WebService;`
- O código da classe que implementa as operações do *Web Service* terá de existir num *assembly* colocado numa subdirectoria de nome *bin* a partir da directoria virtual associada ao IIS;

opcional

`<%@ WebService Language="C#" Class="<nome da classe>, <Assembly>" %>`

```
<%@ WebService Class="SomaNS.Soma, soma" Language="C#" %>
```

SomaWS-PreCompiled.asmx

```
using System;
using System.Web.Services;
namespace SomaNS {
```

```
    [WebService (Description="SomaWS-PreCompiled",
        Namespace="http://deetc.isel.ipl.pt/SD/")]
    public class Soma {
```

```
        [WebMethod (Description="Calcula a Soma de dois inteiros")]
        public int add(int op1, int op2) {
            return op1+op2;
        }
    }
}
```

soma.dll – *assembly* que contém a classe do serviço. Deve ser colocado na directoria *bin* a partir da directoria virtual do IIS

Nota: Embora a indicação do *assembly* seja opcional, a sua indicação, melhora o desempenho no caso de existirem vários *assemblies* na directoria *bin*

Código CodeBehind

- Usado pelo Visual Studio .NET;
- O ficheiro com extensão .asmx apenas contém a directiva `<%@WebService` tal como para o caso *Pre Compiled*;
- Existe um ficheiro (code-behind) que contém o código da classe que implementa o *Web Service*. O nome do ficheiro consiste no mesmo nome do ficheiro .asmx mais a extensão .cs;
- Na directoria `\bin` é colocado o *assembly* resultante da compilação da classe que implementa o *Web Service*;
- A directiva `WebService` inclui o atributo `Codebehind`=“<nome do ficheiro com a classe>”, que é usado pelo *Visual Studio* para efeitos de gestão do projecto, mas não tem qualquer significado em tempo de execução;

```
<%@ WebService Language="C#" Codebehind="SomaWS-CodeBehind.aspx.cs"
    Class="SomaNS.SomaCB" %>
```

A precedência é do código *Inline*, isto é, se existir o código no ficheiro .asmx e também um *Assembly* com a classe na directoria `\bin`, é considerado o código *Inline*

Criar um Web Service com o Visual Studio .NET 2005/2008/2010

- Os *Web Services* são alojados e acedidos via *Servidor Web*. O *Visual Studio 2005/2008/2010* permite 3 modos distintos de disponibilizar esse *Servidor Web*:
 1. Alojamento do *Web Service* directamente na raiz de um servidor IIS:
 - Esta opção só se deve utilizar quando o serviço estiver devidamente testado (*Web Service* em produção).
 2. Alojamento do *Web Service* numa directoria virtual criada num servidor IIS:
 - Esta era a única solução de desenvolvimento de *web services* até ao surgimento do *Visual Studio 2005*.
 3. Alojamento do *Web Service* num servidor dedicado suportado pelo próprio *Visual Studio 2005/2008/2010*:
 - Com esta opção o VS2005 /VS2008/VS2010 cria um *Servidor Web* e associa a raiz desse *Servidor Web* a uma directoria do sistema de ficheiros (que está associada ao projecto)

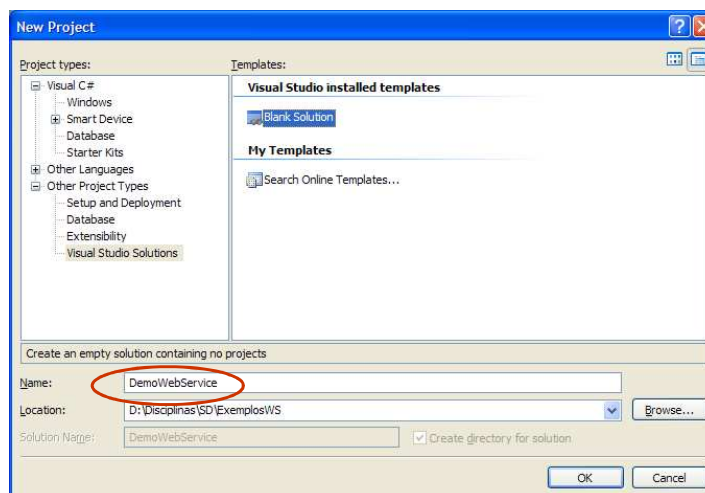
Alojar o *Web Service* num servidor dedicado suportado pelo próprio *Visual Studio 2005/2008/2010*

ISEL/DEETC - Sistemas Distribuídos

7

Criar um *Web Service* com o *Visual Studio .NET 2005/2008/2010*

- Criar uma nova solução, por exemplo *DemoWebService*:

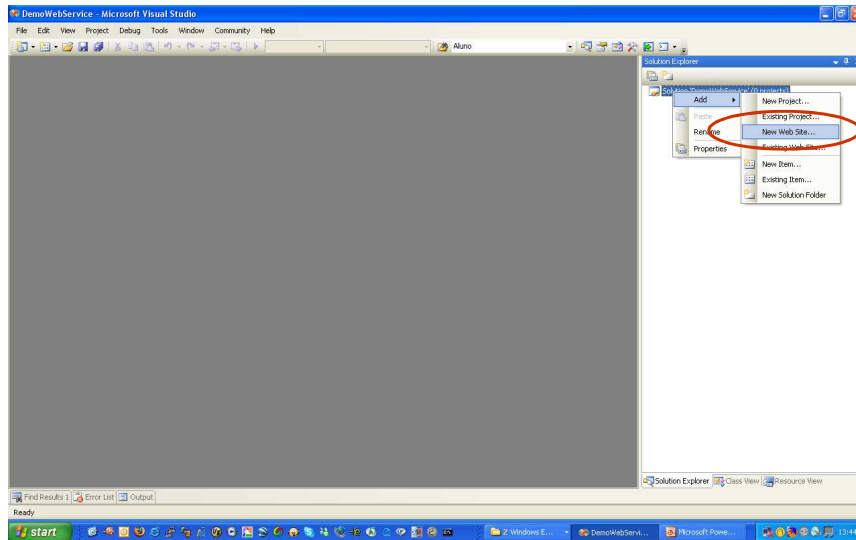


ISEL/DEETC/SES - Sistemas Distribuídos

8

Criar um Web Service com o Visual Studio .NET 2005/2008/2010

■ Adicionar um novo Web Site:

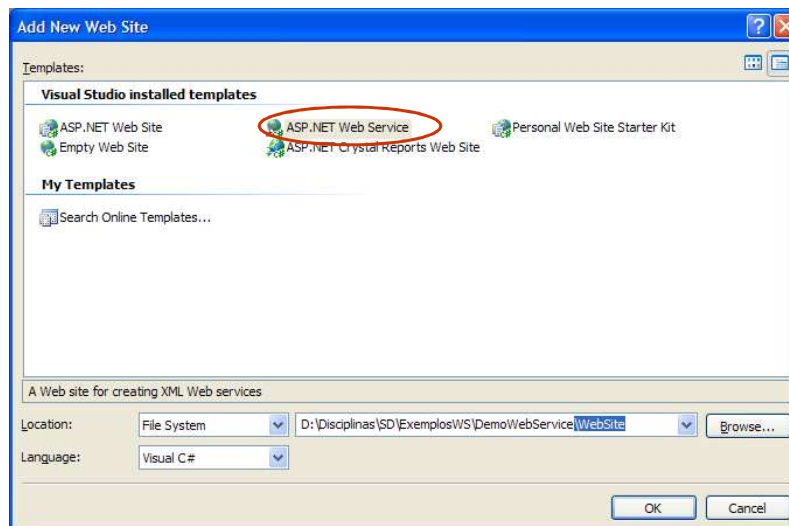


ISEL/DEETC/SES - Sistemas Distribuídos

9

Criar um Web Service com o Visual Studio .NET 2005/2008

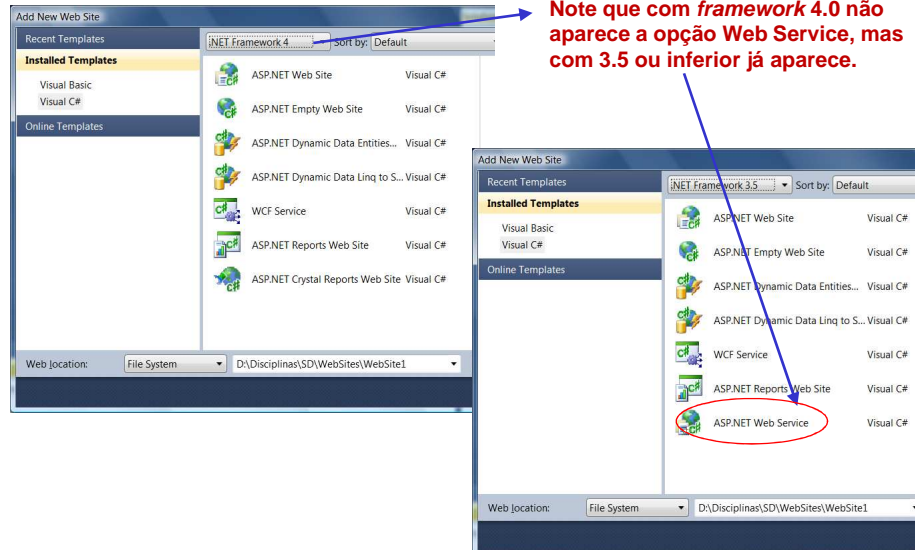
■ Indicar a localização da directoria que suporta o Web Site:



ISEL/DEETC/SES - Sistemas Distribuídos

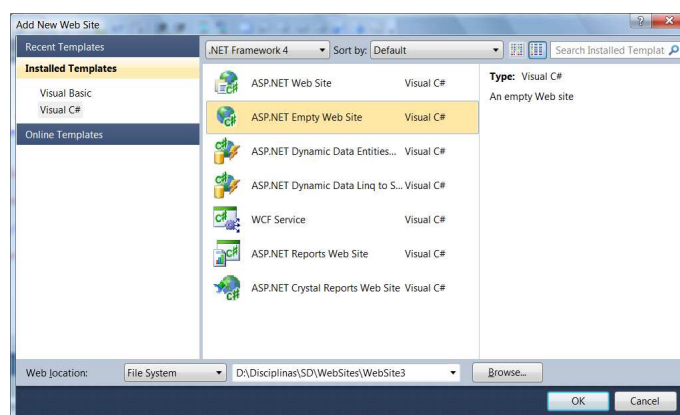
10

Criar um Web Service com o Visual Studio .NET 2010



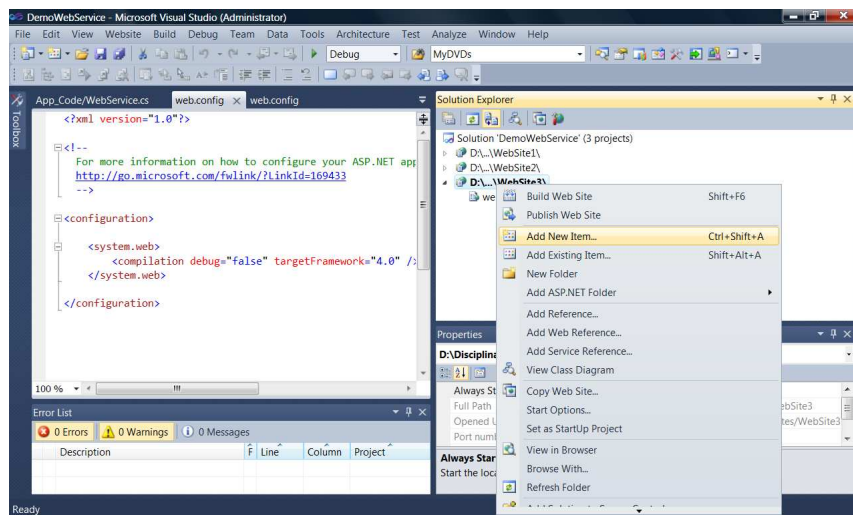
Criar um Web Service com o Visual Studio .NET 2010

Criar um Empty Web Site



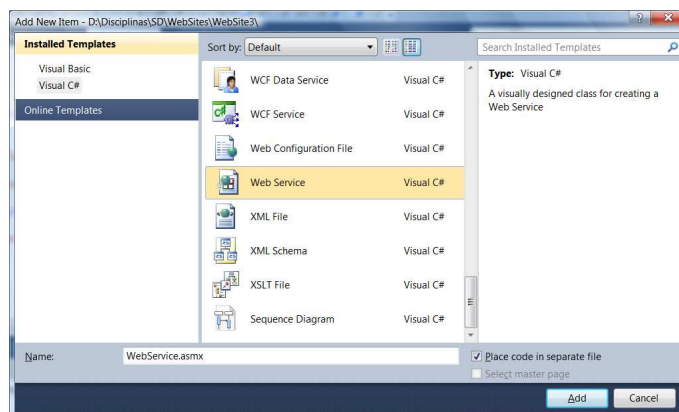
Criar um Web Service com o Visual Studio .NET 2010

Adicionar New Item



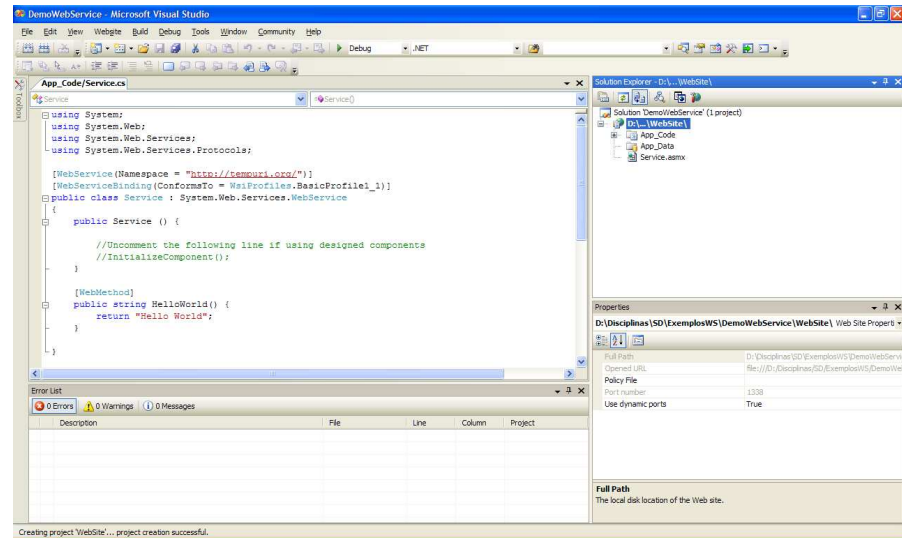
Criar um Web Service com o Visual Studio .NET 2010

Selecionar Web Service



Criar um Web Service com o Visual Studio .NET 2005/2008/2010

Web Service criado:

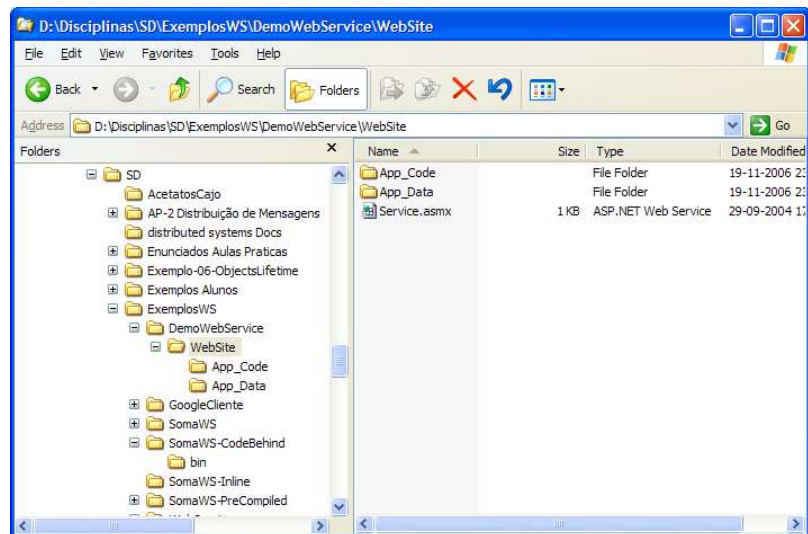


ISEL/DEETC/SES - Sistemas Distribuídos

15

Criar um Web Service com o Visual Studio .NET 2005/2008/2010

Estrutura do File System correspondente ao Web Service criado:

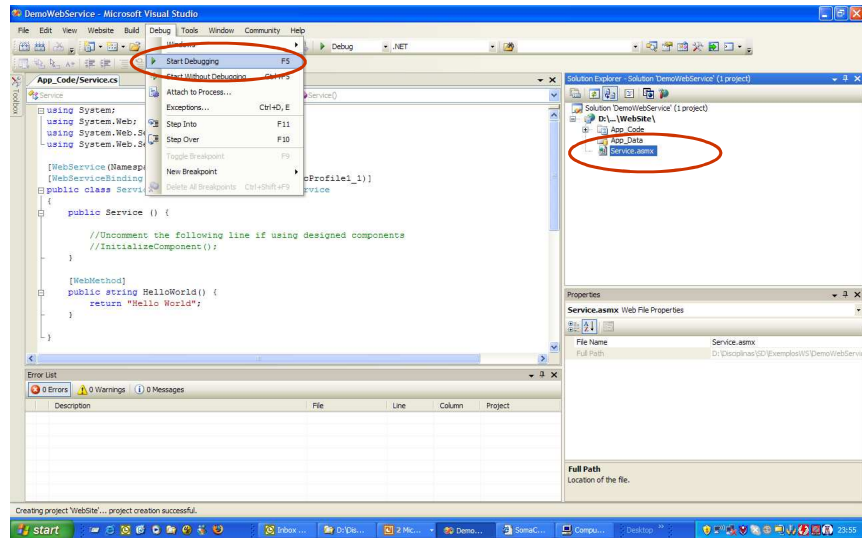


ISEL/DEETC/SES - Sistemas Distribuídos

16

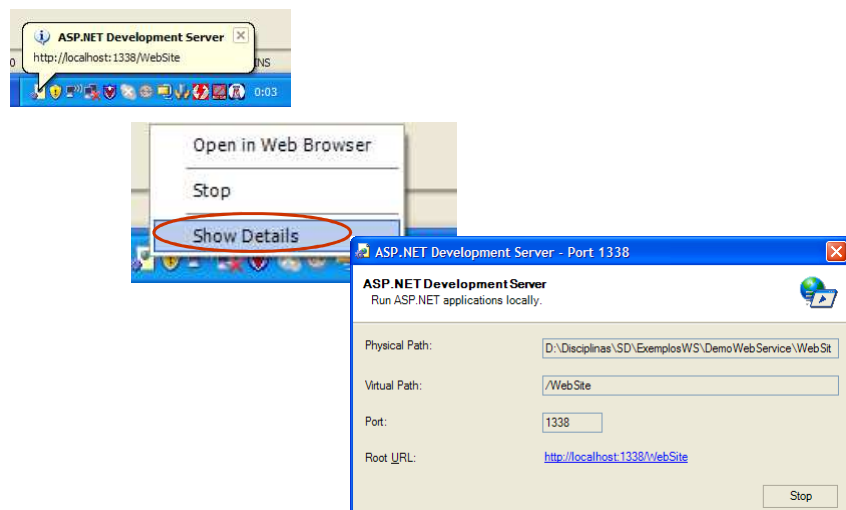
Criar um Web Service com o Visual Studio .NET 2005/2008/2010

■ Execução do Web Service – Fase 1:



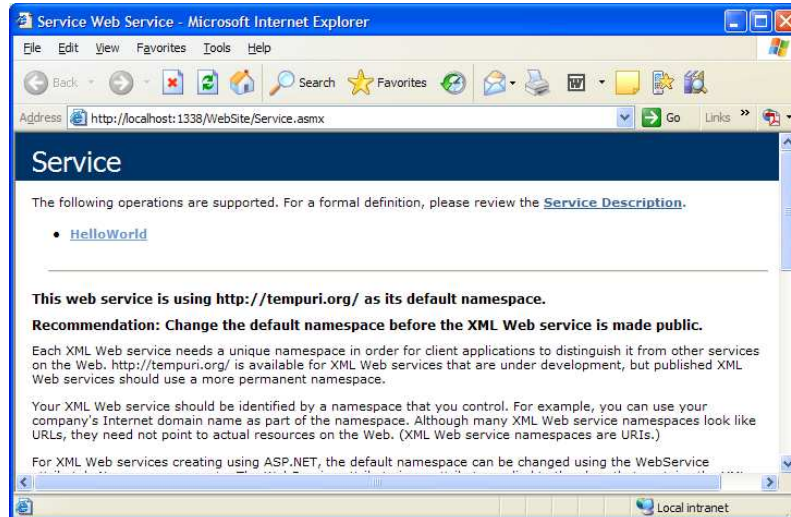
Criar um Web Service com o Visual Studio .NET 2005/2008/2010

■ Execução do Web Service – Pormenores do Site Web dedicado:



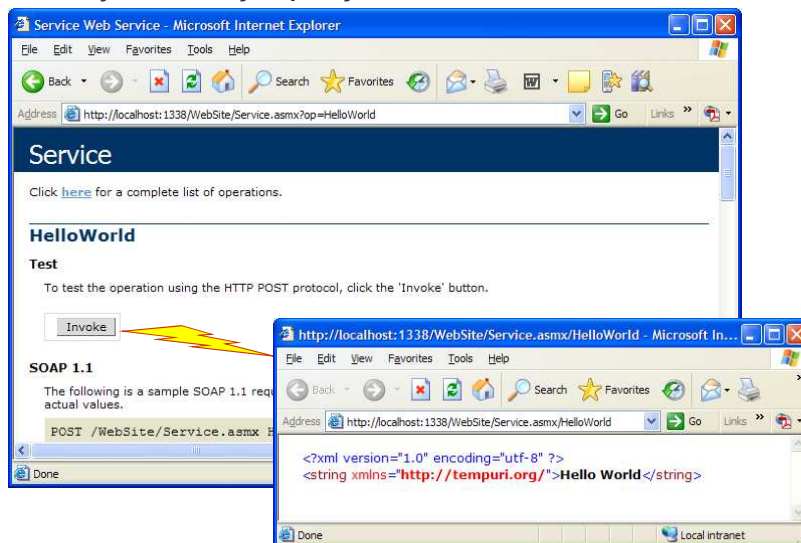
Criar um Web Service com o Visual Studio .NET 2005/2008/2010

■ Execução do Web Service `Service.asmx`:



Criar um Web Service com o Visual Studio .NET 2005/2008/2010

■ Invocação do serviço Operação - `HelloWorld`:



Os detalhes do código gerado automaticamente

- O ficheiro *code-behind* já tem a definição de uma classe para o *Web Service*.
- Esta classe deriva de `System.Web.Services.WebService`, e permite que o novo serviço herde métodos que possibilitam o acesso a objectos de contexto do protocolo HTTP e manutenção de estado existentes em ASP.NET, por exemplo os objectos `Application` e `Session`. Esta utilização é opcional;
- O ficheiro `Service.asmx` apenas contém uma linha de texto:

```
<%@ WebService Language="C#" CodeBehind="~/App_Code/Service.cs" Class="Service" %>
```

Os detalhes do código gerado automaticamente

- A implementação do *Web Service* fica contida no ficheiro `Service.cs` e pode ser completada com outros métodos, por exemplo um novo método que sabe adicionar dois inteiros:

```
using ...;
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Service : System.Web.Services.WebService {
    public Service () { ... }

    [WebMethod]
    public string HelloWorld() { return "Hello World"; }

    [WebMethod]
    public int add(int op1, int op2) { return op1+op2; }
}
```

Utilização de ComplexTypes

[WebMethod]

```
public ComplexType[] GetArrayOfComplexTypes() {
    ComplexType ct1 = new ComplexType("aaa", 1, new string[] { "aaa", "aaa" });
    ComplexType ct2 = new ComplexType("bbb", 2, new string[] { "bbb", "bbb" });

    return new ComplexType[] { ct1, ct2 };
}

public class ComplexType {
    public string str;
    public int num;
    public string[] tabStr;

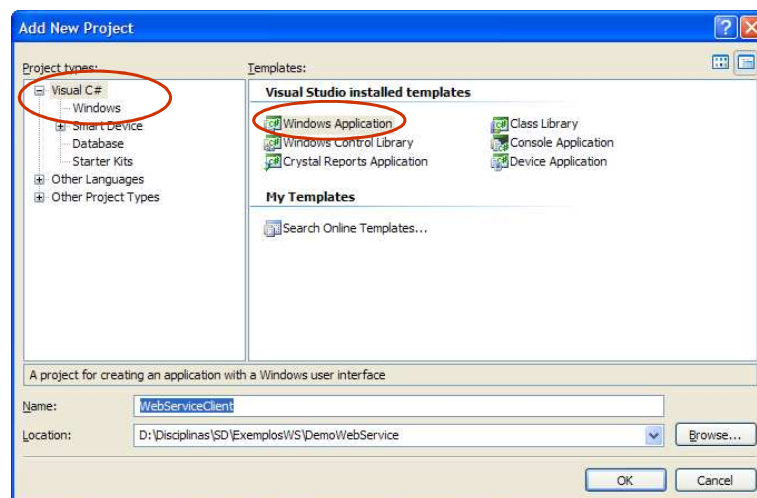
    public ComplexType() { }
    public ComplexType(string _str, int _num, string[] ts)
    {
        str = _str; num = _num; tabStr = ts;
    }
}
```

Resultado

```
- <ArrayOfComplexType xmlns:xsi="
  xmlns:xsd="http://www.w3.o
- <ComplexType>
  <str>aaa</str>
  <num>1</num>
  - <tabStr>
    <string>aaa</string>
    <string>aaa</string>
  </tabStr>
</ComplexType>
- <ComplexType>
  <str>bbb</str>
  <num>2</num>
  - <tabStr>
    <string>bbb</string>
    <string>bbb</string>
  </tabStr>
</ComplexType>
</ArrayOfComplexType>
```

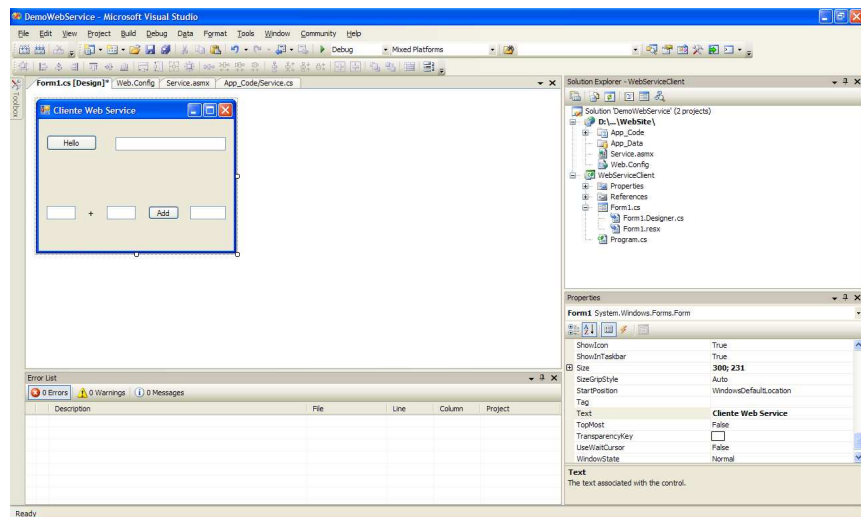
Criar um cliente para o Web Service

- Adicionar à solução uma aplicação cliente, por exemplo uma aplicação Win Forms:



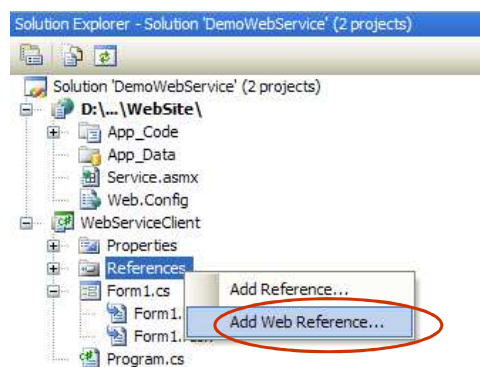
Criar um cliente para o Web Service

■ Desenho da interface com o utilizador:

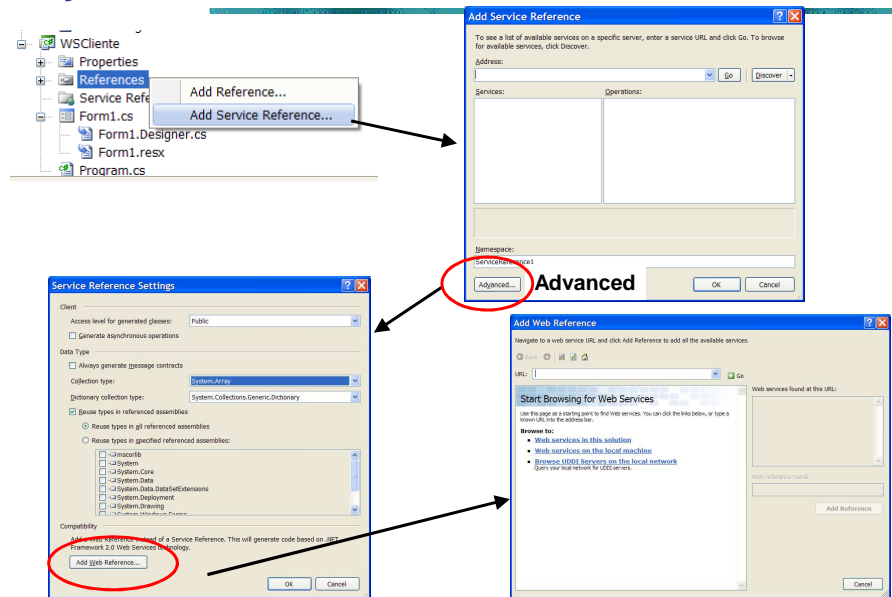


Criar um cliente para o Web Service

■ Adição de uma referência para o Web Service criado previamente – Fase 1:



Adição de Referência no Visual Studio 2008/2010

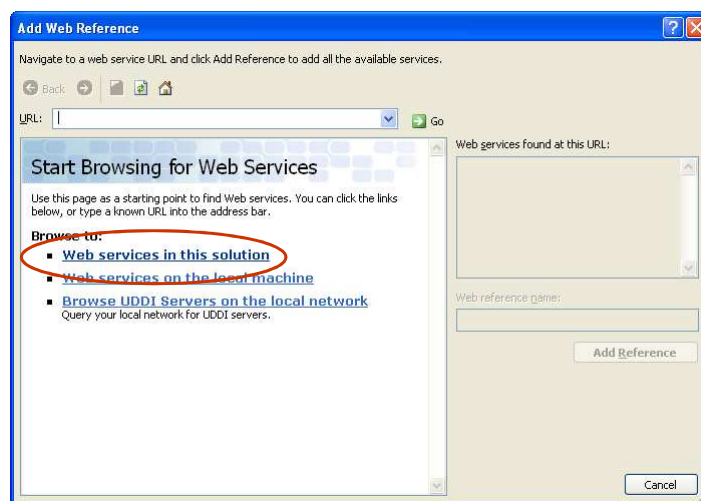


ISEL/DETC/SES - Sistemas Distribuídos

27

Criar um cliente para o Web Service

■ Adição de uma referência para o Web Service criado previamente – Fase 2:

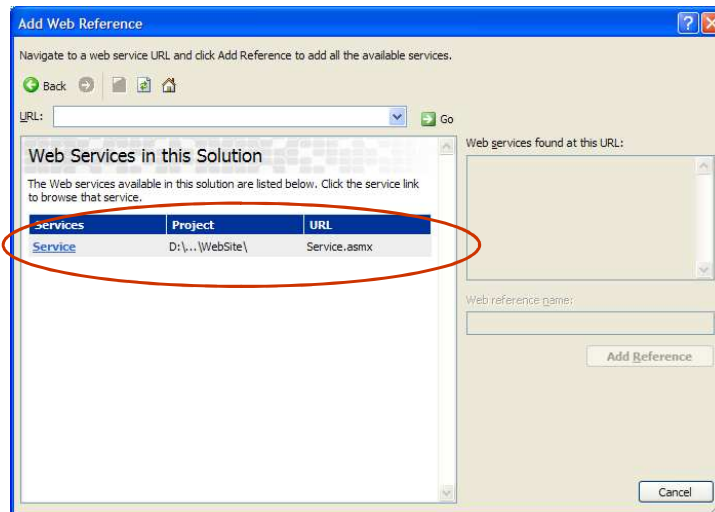


ISEL/DETC/SES - Sistemas Distribuídos

28

Criar um cliente para o Web Service

■ Adição de uma referência para o Web Service criado previamente – Fase 3:

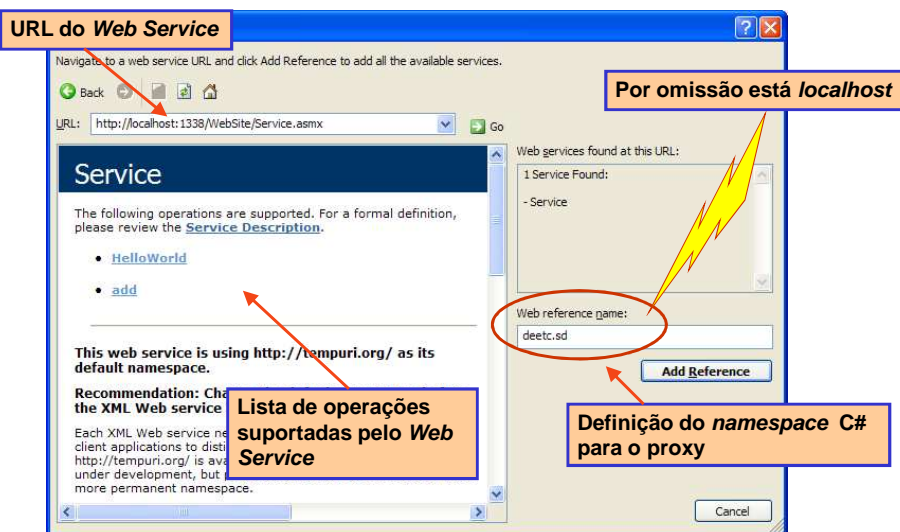


ISEL/DETC/SES - Sistemas Distribuídos

29

Criar um cliente para o Web Service

■ Adição de uma referência para o Web Service criado previamente – Fase 4:



ISEL/DETC/SES - Sistemas Distribuídos

30

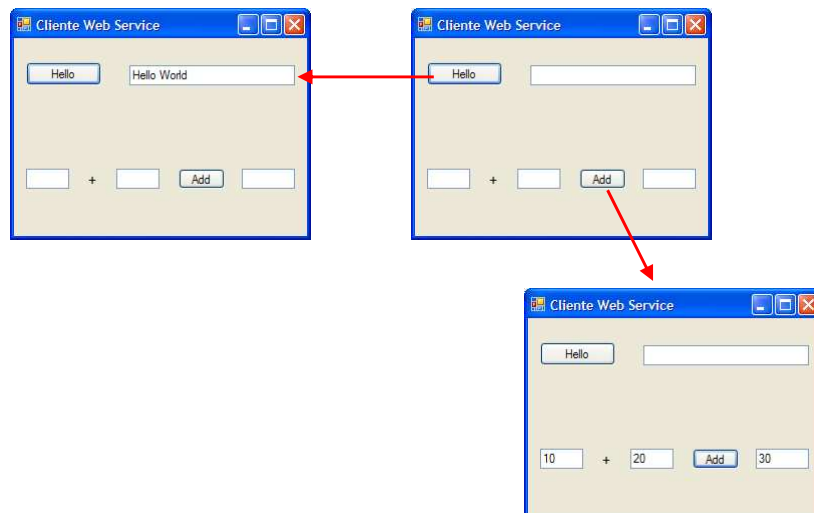
Criar um cliente para o Web Service

■ Adição de uma referência para o Web Service criado previamente – Fase 5:



Utilização do Web Service com o cliente

■ Resultado da execução:



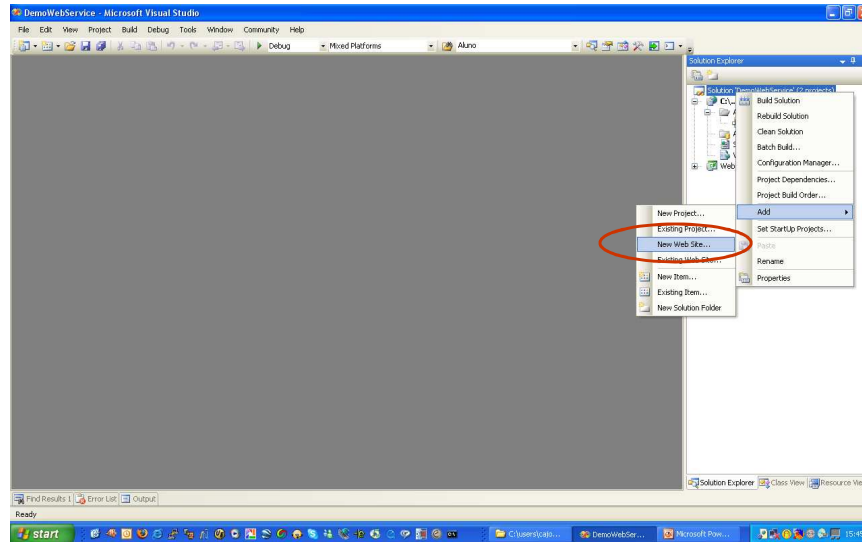
Estabelecimento de *Web References* – Resumo

1. Sobre o projecto da aplicação cliente do serviço seleccionar com botão direito do rato a opção *Add Web Reference*. No caso do VS2008/VS2010 navegar em *Add Service Reference* até *Add Web Reference* (ver slide 22).
2. Indicar o endereço do *Web Service*:
 - <http://localhost:1338/WebSite/Service.asmx>
3. Em *Web Reference Name* indicar o *namespace* com que será conhecido o proxy. Este *namespace* será prefixado com o nome do projecto:
 - `WebServiceClient.deetc.sd`
4. Na directoria do projecto é criada a directoria *Web References*:
 - `<Project root>\Web References\deetc.sd`
5. Onde é criado o ficheiro `Reference.cs` com a classe *proxy* sob o *namespace* `WebServiceClient.deetc.sd`
6. Nesta directoria é ainda criado o ficheiro com a definição da interface WSDL do *Web Service*:
 - `Service.wsdl`
7. O acesso ao serviço, poderá ter que ser autenticado dependendo das políticas de segurança adoptadas.

Alojar o *Web Service* numa directoria virtual criada num servidor IIS

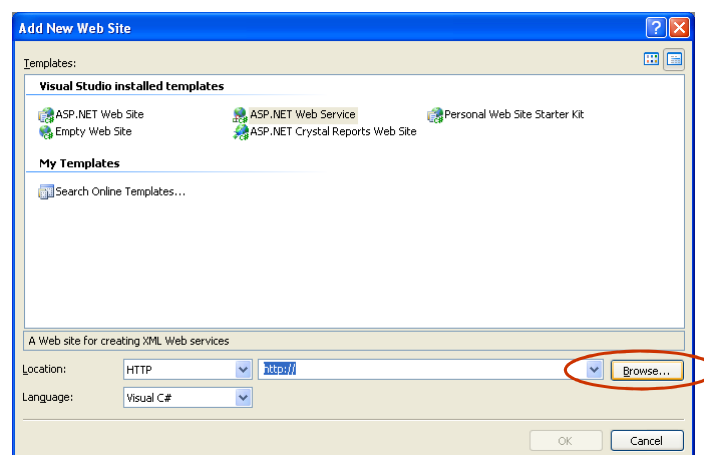
Criar um Web Service com o Visual Studio .NET 2005/2008/2010

■ Adicionar um novo Web Site – Fase 1:



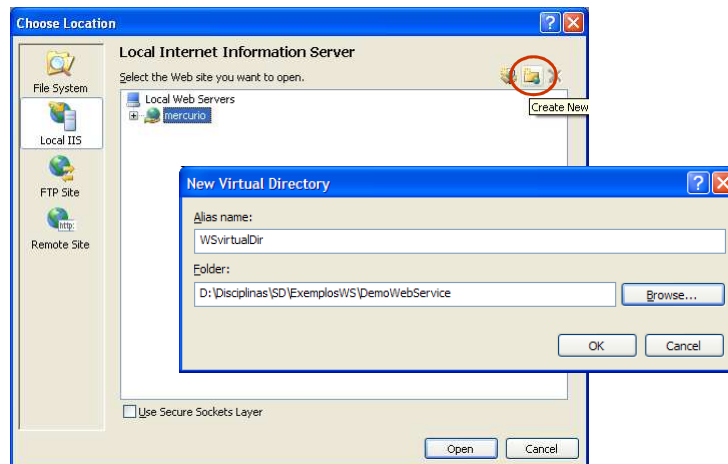
Criar um Web Service com o Visual Studio .NET 2005/2008/2010

■ Adicionar um novo Web Site – Fase 2:



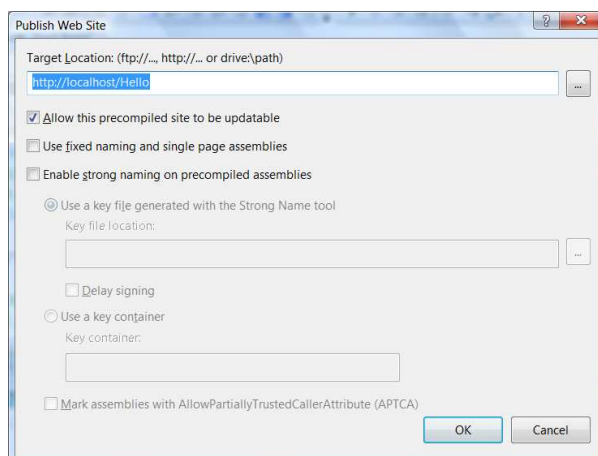
Criar um Web Service com o Visual Studio .NET 2005

■ Adicionar um novo Web Site – Fase 3:



Publicação do Web Service no IIS

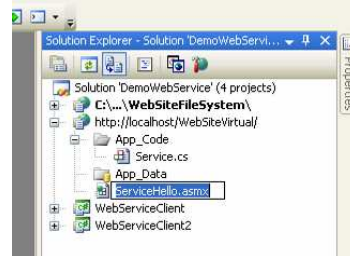
Uma boa prática é usar o *web site* no *file system* durante toda a fase de desenvolvimento. Após a fase de desenvolvimento pode fazer-se a publicação no IIS da máquina onde se pretende alojar o web service.



Hello – pode ser uma directoria virtual no IIS.

Alteração do nome e classe do Web service

- Por omissão o *Visual Studio* cria o *Web Service* com o nome `Service.aspx`;
- Caso se pretenda usar um nome diferente, basta fazer *click* com o botão direito do rato e fazer *rename* ou premir F2;
- O nome da classe que implementa o *Web Service* por omissão é também *Service*, mas pode também ser mudado no editor no ficheiro que contém o *Code behind*.



```
...  
public class ServiceHello : System.Web.Services.WebService  
{  
    public ServiceHello() { ... }  
    ...  
}
```

No Cliente não esquecer de actualizar o proxy fazendo *Add Web Reference* de novo.

Espaços de nomes associados aos Web Services

■ System.Web

- Classes e interfaces na gestão da comunicação *Browser-Web Server*
 - `HttpRequest`, `HttpResponse`, `HttpCookie`, `HttpClientCertificate`, etc.

■ System.Web.Services

- Conjunto de classes que permitem a criação de serviços *Web* no contexto da componente *ASP.NET*
- Classes: `WebMethodAttribute`, `WebService`, `WebServiceAttribute`, `WebServiceBindingAttribute`
- A classe `WebService` quando usada para derivar um novo *Web Service* permite o acesso a objectos associados à aplicação, como: *Application*, *Session*, *User*, e *Context* [Herança]

■ System.Net

- Fornece uma interface para os vários protocolos de *networking*;

Utilização de comandos em “*command prompt*”

- ☐ **csc.exe** (ajuda - ...>csc /?)
 - Compilador de C# acessível na consola do VS .NET
 - `csc /t:library /out:bin\<assembly>.DLL <proxy>.cs <modelo-dados>.cs`
- ☐ **wsdl.exe** (ajuda - ...>wsdl /?)
 - Gera um ficheiro (C# por omissão) com a classe proxy de um *Web Service* (classe que implementa o serviço)
 - `wsdl /out:<proxy>.cs /n:deetc.sd <interface>.wsdl`
- ☐ **xsd.exe** (ajuda - ...>xsd /?)
 - Gera um ficheiro (C# por omissão) com o modelo de dados associado aos serviços, a partir de um ficheiro XSD (*XML Schema Definition*)
 - `xsd /c /out:<modelo-dados>.cs /n:deetc.sd <xml schema>.xsd`
- ☐ **Adicionar à variável PATH** (adaptar de acordo com a versão e configuração de cada máquina)
 - “C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727” [csc, ...]
 - “C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\Bin” [wsdl, xsd, ...]
 - [Windows Vista] C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin [wsdl, xsd, ...]
- ☐ **Ou executar o *batch file* do Visual Studio:**
 - “C:\Program Files\Microsoft Visual Studio 10.0\Common7\Tools\vsvars32.bat”