

---

## Arquitecturas Orientadas aos Serviços (SOA – *Service-Oriented Architectures*)

### Web Services ASP.NET *Windows Communication Foundation WCF*

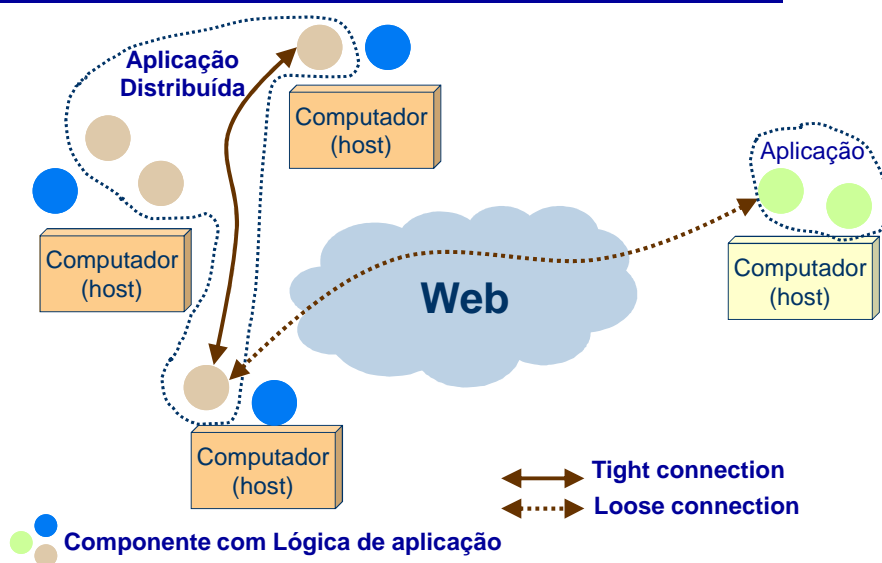
### *Interdependência entre partes*

---

Num sistema distribuído os aspectos relacionados com a interdependência entre as partes devem considerar:

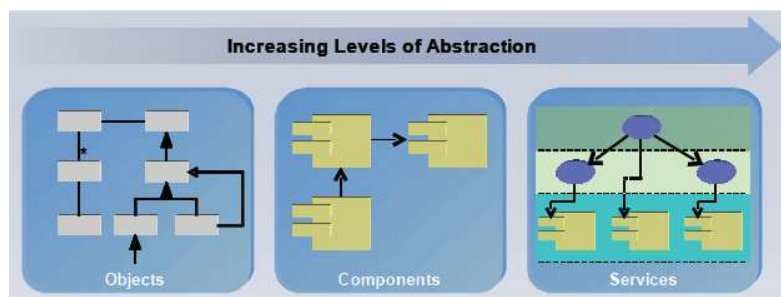
- Reduzir ao máximo o número de componentes conhecidas ou que comprometem as partes envolvidas;
- Se é feita uma mudança numa das partes então não deve ser necessário fazer consequentes mudanças noutras partes;
- Um aspecto a ter em conta é o tipo de ligação forte ou fraca:
  - *loose connection* (ligação fraca) - a ligação só existe no decurso da chamada de uma operação. Ex: HTTP; Web Services
  - *tight connection* (ligação forte) - existe o conceito de sessão. Ex: TCP; serviços *stateful*. Cada um dos interlocutores pressupõe comportamentos da outra parte.

## Ligação forte e ligação fraca (*loose-tight connections*)



## SOA – Service Oriented Architecture

- Colecção de serviços bem definidos onde cada serviço pode ser modificado de forma independente, aos outros serviços, para responder a mudanças de mercado ou de negócio.
- Proporciona um modelo de interacção através de mensagens baseado em standards abertos;



## Service Oriented Architecture - SOA

- Arquitecturas distribuídas baseadas no conceito de Serviço;
- Os serviços são os *building blocks* das arquitecturas SOA e disponibilizam funcionalidades usadas por aplicações de utilizador ou por outros serviços;
- Existe um mecanismo consistente para comunicar com os serviços, que deve ser *loosely coupled* e suportar o uso de interfaces explícitas;
- Aplica os conceitos comprovados de desenvolvimento orientado aos objectos e desenho baseado em componentes;
- A adopção de arquitecturas SOA, permite construir sistemas flexíveis:
- Permite a reutilização de componentes relacionadas com os processos de negócio;
- Facilita mudanças e a criação de novos processos de negócios numa organização;

## O que é um Serviço ?

### Princípio 1 - Fronteiras explícitas

- Interface pública e bem definida. Todas as interacções ocorrem através dessa interface;
- Interfaces simples e pequenas, indicando as mensagens que podem ser enviadas/recebidas ;
- Facilidade dos clientes poderem usar o serviço, conhecendo apenas a interface;
- Não são conhecidos detalhes de implementação do serviço

---

## Princípio 2 - Os Serviços são autónomos

- Um serviço deve estar isolado e desacoplado;
- A versão do serviço e o alojamento (*deployment*) deve ser independente do sistema onde está alojado;
- Os contratos (interfaces) dos serviços não devem mudar após a sua publicação;
- Adoptar aproximações pessimistas, isolando o serviço de falhas.

---

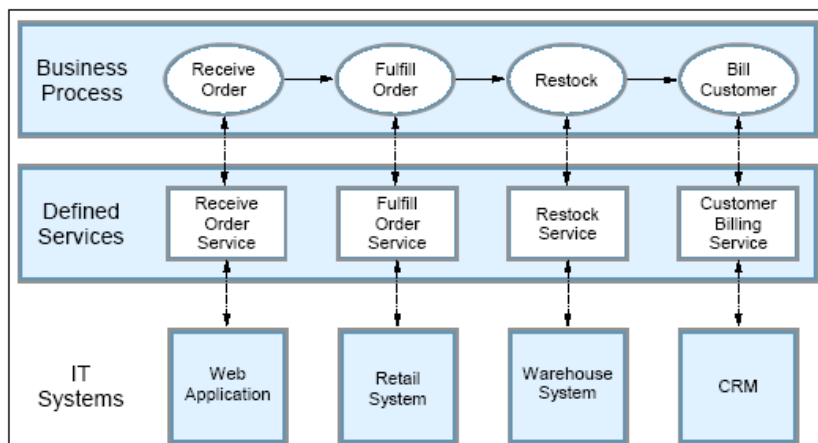
## Princípio 3 - Serviços publicam um Contrato e NÃO uma Classe

- Um serviço publica o *XML schema* das mensagens;
- Um serviço publica um contrato estável através de um *XML schema* bem definido, por exemplo, WSDL (*Web Service Definition Language*) ;
- Se o contrato tiver que mudar, então deve ser criada uma nova versão do serviço, por forma a não comprometer clientes do serviço;
- Modelos de dados internos ao serviço não devem ser públicos.

#### Princípio 4 – A compatibilidade entre serviços

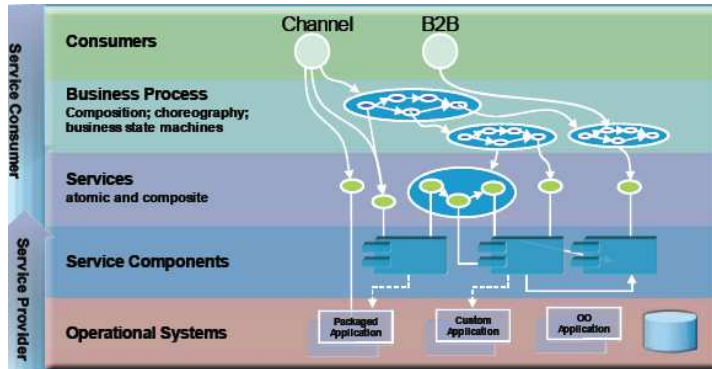
- Nem todos os requisitos de um serviço podem ser especificados no contrato;
- Os comportamentos podem ser conhecidos utilizando asserções (*Policy expressions; Policy assertions*) para representar eventuais requisitos sobre domínios específicos tais como: segurança; transacções; semântica sobre o conteúdo das mensagens; etc.
- Separar as compatibilidades estruturais das compatibilidades semânticas;

#### Novo processo de negócio de pedidos de encomenda



from: IBM RedBooks – "Patterns: Implementing Self-Service in an SOA environment"

## SOA não é tecnologia – É uma estratégia



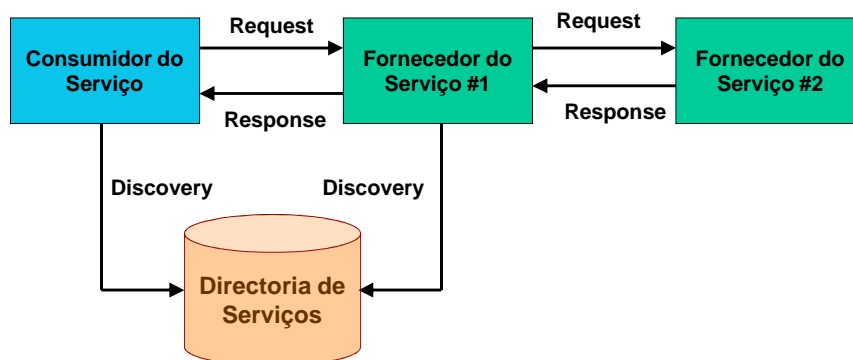
from IBM SOA Architect Summit

### IBM SOMA (Service Oriented Modeling and Architecture)

*Identification, Specification, Realization and Implementation of Services, Components, Flows*

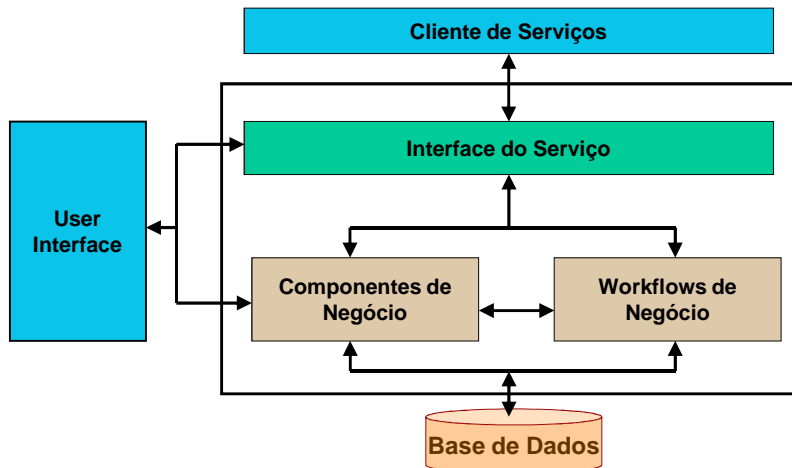
## Arquitetura orientada aos Serviços (SOA)

### • Cenário 1



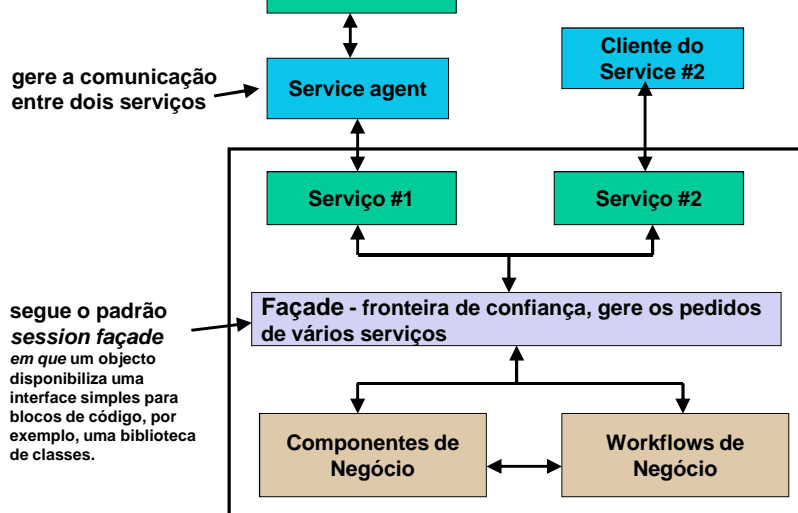
## Arquitectura orientada aos Serviços (SOA)

### • Cenário 2










## Arquitectura orientada aos Serviços (SOA)

### • Cenário 3



## Service Integration Maturity Model (SIMM)

							
<b>Business</b>	Silo Isolated Business Line Driven	Integrated Business Process Integration	Componentized Business	Services Componentized Business offers Services	Composite Services Processes Through Service Composition	Virtualized Services Geographically Independent Service Centers	Dynamically Re-Configurable Services Mix and Match Business and Location Capabilities
<b>Organization</b>	Application Specific Skills	IT Transformation	IT Governance	Emerging SOA Governance	SOA and IT Governance Alignment	SOA and IT Infrastructure Governance Alignment	Governance through Policy
<b>Methods</b>	Structured Analysis & Design	Object Oriented Modeling	Component Based Development	Service Oriented Modeling	Service Oriented Modeling	Service Oriented Modeling	Business Grammar Oriented Modeling
<b>Applications</b>	Modules	Objects	Components	Services	Process Integration via Services	Process Integration via Services	Dynamic Application Assembly
<b>Architecture</b>	Monolithic Architecture	Layered Architecture	Component Architecture	Emerging SOA	SOA	Grid Enabled SOA	Dynamically Reconfigurable Architecture
<b>Information</b>	Application Specific	LOB or Enterprise Specific	Canonical Models	Information As a Service	Enterprise Data Dictionary and Repository	Virtualized Data Services	Semantic Data Vocabularies
<b>Infrastructure</b>	LOB Platform Specific	Enterprise Standards	Common Reusable Infrastructure	Project-based SOA Environment	Common SOA Environment	Virtual SOA Environment	Dynamic Sense & Respond

## Mashups

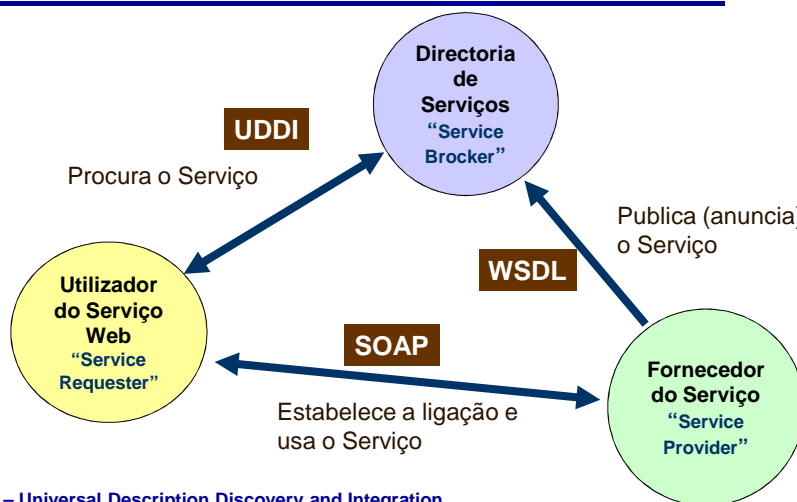
- Um  *mashup*  é uma combinação de informação oriunda de várias fontes por forma a criar algo de novo ou no mínimo útil.
- O ideal é que um utilizador possa construir um novo  *mashup*  sem programar.
- [http://www.youtube.com/watch?v=U9sENSA\\_sjl](http://www.youtube.com/watch?v=U9sENSA_sjl)
  - Vídeo sobre o que é um  *mashup*
- <http://mashable.com/2007/07/11/google-maps-mashups-2/>
  - Site sobre  *mashups*  baseados em Google Maps



## O que são Web Services ?

- Serviços são componentes autónomas que processam mensagens XML bem definidas em formato *SOAP – Simple Object Access Protocol*;
- Os Serviços fornecem um interface bem conhecida, que pode ser descrita através de um documento *XML (Web Service Description Language –WSDL)* também conhecido por contrato WSDL, onde, entre outras, são descritas as mensagens que podem ser enviadas e recebidas do serviço;
- Um Serviço fornece *endpoints* (tipicamente um URL) a que os consumidores do serviço (por exemplo outros serviços) se podem ligar (*binding*)

## Web Services



UDDI – Universal Description Discovery and Integration  
WSDL – Web Services Definition Language  
SOAP – Simple Object Access Protocol

## Protocolos relacionados com os Web Services

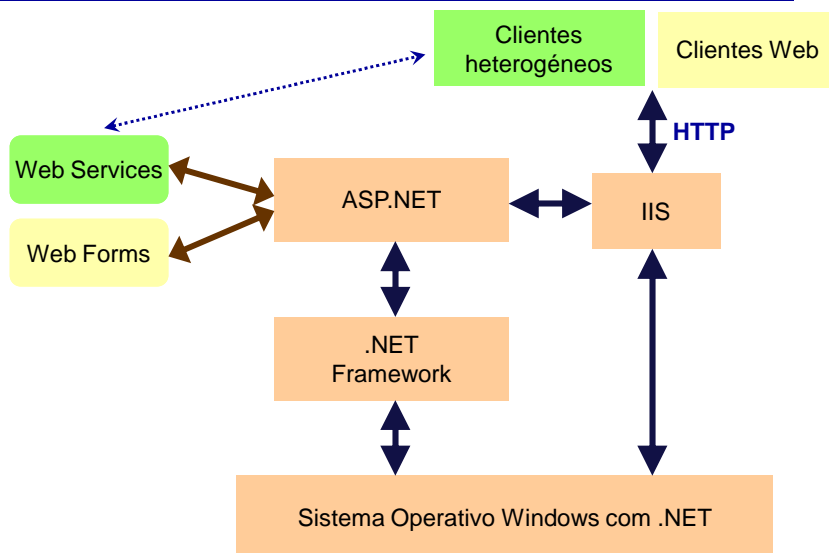
- **HTTP**
  - Protocolo de comunicação usado para o transporte de chamadas aos serviços
- **WSDL (*Web Service Description Language*)**
  - Documento de descrição do serviço– operações, tipos de dados, parâmetros, informação de *binding* para acesso ao serviço
- **SOAP (*Simple Object Access Protocol*)**
  - Representação das mensagens de chamada e resposta aos serviços
- **UDDI (*Universal Description, Discovery, and Integration*)**
  - Directoria de Web Services e outras informações de negócio
- **DISCO (WS-Inspecion)**
  - Especificação (Microsoft) para a descoberta (*Discovery*) de serviços

## Perspectivas para a utilização de *Web Services*

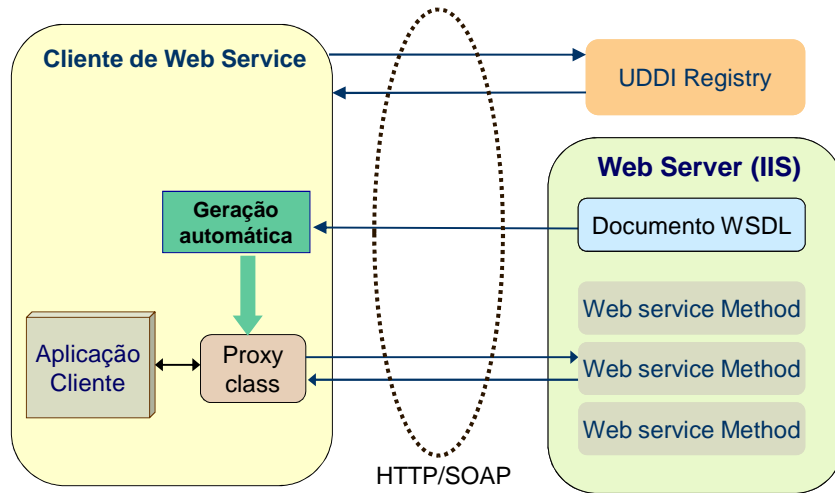
- **Standards abertos**
- **Facilitador da Interoperabilidade**
- **Independente de plataformas**
- **Mensagens e interfaces em XML**
- **Facilitador de implementação de arquitecturas SOA**
- **Largamente adoptado pela indústria**

## Web Services na plataforma .NET

### Arquitetura ASP.NET



## Web Services .NET

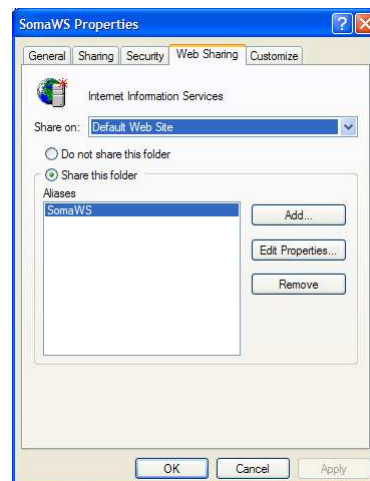


## Um primeiro Web Service em .NET – Alojamento do Serviço

### Windows XP – IIS 6.0

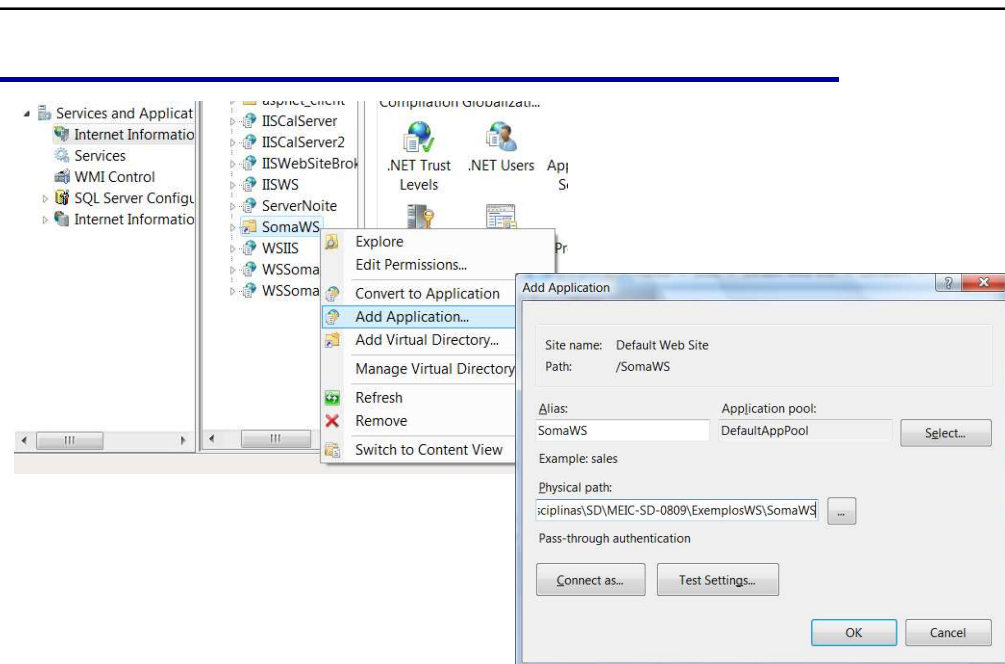
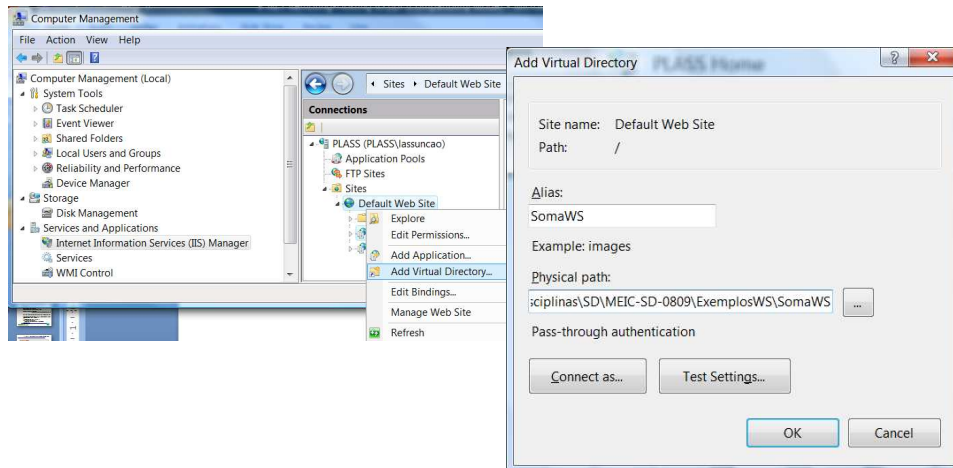
#### 1º Passo

- Crie uma nova directoria a partir da sua área de trabalho, por exemplo com o nome **SomaWS**;
- Nas propriedades da directoria **SomaWS**, transforme essa directoria numa directoria virtual do servidor Web (IIS), como se indica na figura;



## 1º Passo

## Windows Vista – IIS 7.0



## Um primeiro Web Service em .NET – O Serviço

### ■ 2º Passo

- Editar com *notepad*, ou outro editor de texto, um ficheiro com o nome *Soma.aspx*, (a extensão *asmx* é usada em .NET para os Web Services) com o conteúdo abaixo apresentado e coloque-o na directoria *SomaWS*, anteriormente criada.

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

using System.Web.Services;

[WebService (Description="Exemplo Soma",
             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;
    }
}
```

## Um primeiro Web Service em .NET – Um possível Cliente

### ■ 3º Passo

- Editar com *notepad*, ou outro editor de texto, um ficheiro com o nome *clientSoma.html* com o conteúdo abaixo apresentado e coloque-o na directoria *SomaWS*, anteriormente criada.

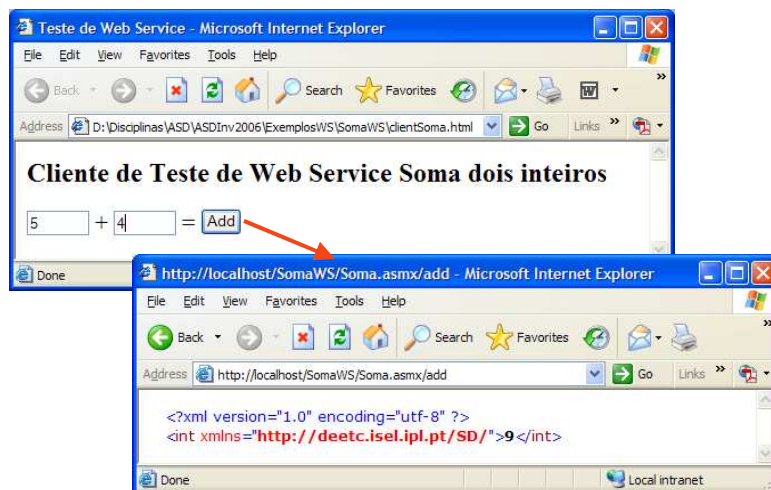
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title> Teste de Web Service</title>
  </head>
  <body>
    <h3>Cliente de Teste de Web Service Soma dois inteiros</h3>
    <form method=POST
      action='http://localhost/SomaWS/Soma.aspx/add'>
      <input type="text" size="5" name='op1'> +
      <input type="text" size="5" name='op2'> =
      <input type=submit value="Add">

    </form>
  </body>
</html>
```

## Um primeiro Web Service em .NET – *Invocar o serviço*

### ■ 4º Passo

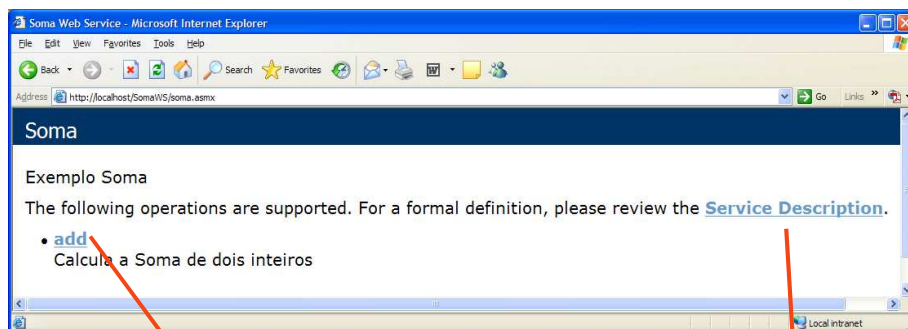
- executar o ficheiro *html*



## Usar como cliente do Web Service o browser IE

### ■ 5º Passo

- Chamada do Web service *Soma.asmx* a partir do browser IE, usando o URL *http://localhost/SomaWS/Soma.asmx*



permite chamar a operação "add"

permite visualizar o WSDL com a descrição do Web Service

## Chamada da operação "add"

The screenshot shows two overlapping browser windows. The top window displays the 'add' operation interface with the title 'add' and the description 'Calcula a Soma de dois inteiros'. It includes a 'Test' section with instructions to use the HTTP POST protocol. Below this is a table for parameters:

Parameter	Value
op1:	5
op2:	4

An 'Invoke' button is located below the table. The bottom window shows the XML response from the service:

```
<?xml version="1.0" encoding="utf-8" ?>
<int xmlns="http://deetc.isel.ipl.pt/SD/">9</int>
```

## Mensagens SOAP da operação "add"

The screenshot shows a browser window displaying the SOAP message for the 'add' operation. The message is a sample request and response. The request is a POST to /SomaWS/Soma.aspx with the following headers:

```
POST /SomaWS/Soma.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://deetc.isel.ipl.pt/SD/add"
```

The request body is an XML document:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <soap:Body>
    <add xmlns="http://deetc.isel.ipl.pt/SD/">
      <op1>int</op1>
      <op2>int</op2>
    </add>
  </soap:Body>
</soap:Envelope>
```

The response is an HTTP 200 OK with the following headers:

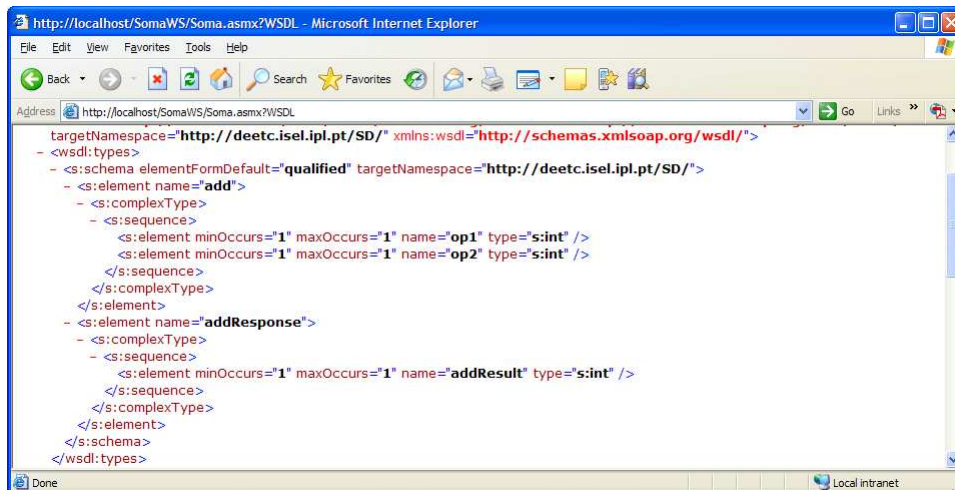
```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

The response body is an XML document:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <soap:Body>
    <addResponse xmlns="http://deetc.isel.ipl.pt/SD/">
      <addResult>int</addResult>
    </addResponse>
  </soap:Body>
</soap:Envelope>
```

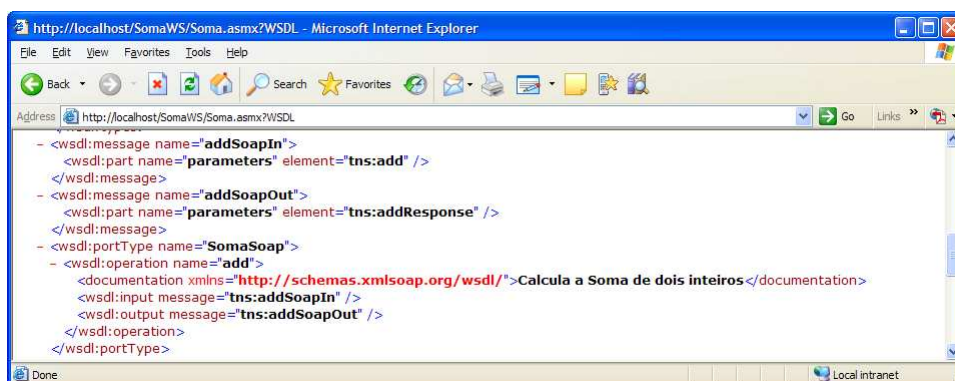


## Descrição do serviço em WSDL – definição de tipos



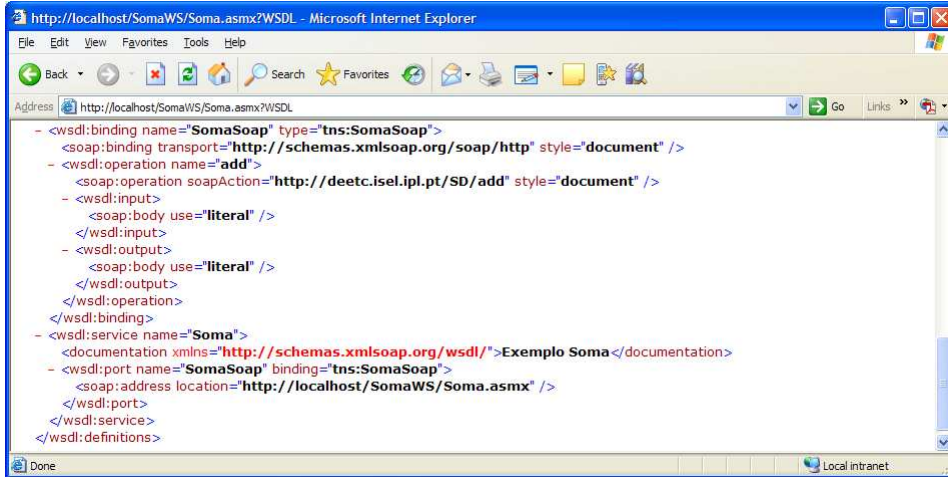
```
targetNamespace="http://deetc.isel.ipl.pt/SD/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://deetc.isel.ipl.pt/SD/">
      <s:element name="add">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="op1" type="s:int" />
            <s:element minOccurs="1" maxOccurs="1" name="op2" type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="addResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="addResult" type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
```

## Descrição do serviço em WSDL – definição de mensagens e operações



```
<wsdl:message name="addSoapIn">
  <wsdl:part name="parameters" element="tns:add" />
</wsdl:message>
<wsdl:message name="addSoapOut">
  <wsdl:part name="parameters" element="tns:addResponse" />
</wsdl:message>
<wsdl:portType name="SomaSoap">
  <wsdl:operation name="add">
    <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Calcula a Soma de dois inteiros</documentation>
    <wsdl:input message="tns:addSoapIn" />
    <wsdl:output message="tns:addSoapOut" />
  </wsdl:operation>
</wsdl:portType>
```

## Descrição do serviço em WSDL – definição de *binding* e *location* do serviço



```
<?xml version='1.0' encoding='utf-8'?>
<wsdl:definitions xmlns:tns="http://schemas.xmlsoap.org/soap/http" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" targetNamespace="http://schemas.xmlsoap.org/soap/http">
  <wsdl:binding name="SomaSoap" type="tns:SomaSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  </wsdl:binding>
  <wsdl:operation name="add">
    <soap:operation soapAction="http://deetc.isel.ipl.pt/SD/add" style="document" />
  </wsdl:operation>
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:definitions>
<wsdl:service name="Soma">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Exemplo Soma</documentation>
  <wsdl:port name="SomaSoap" binding="tns:SomaSoap">
    <soap:address location="http://localhost/SomaWS/Soma.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

## Proxy para acesso ao Web Service

- Gerar o proxy com ajuda do utilitário **WSDL.EXE**, disponível em:  
C:\Program Files\Microsoft SDKs\Windows\v7.0\bin\NETFX 4.0 Tools  
**> wsdl /out:soma.cs /n:SD http://localhost/SomaWS/Soma.asmx**
- Gera um ficheiro em **C#** com o código do proxy.

```
namespace SD {
  public class Soma : System.Web.Services.Protocols.SoapHttpClientProtocol {
    public Soma() {
      this.Url = "http://localhost/SomaWS/Soma.asmx";
    }
    ...
    public int add(int op1, int op2) {
      object[] results = this.Invoke("add", new object[] {
        op1,
        op2});
      return ((int)(results[0]));
    }
  }
}
```

## Cliente Win Forms que acede ao Serviço

using deetc.sd;  
using System.Net;

...

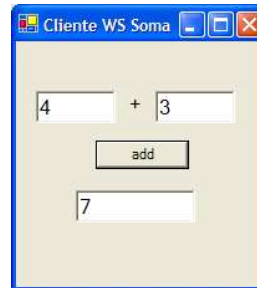
Classe Proxy do Web Service

```
private void button1_Click(object sender, System.EventArgs e) {
```

```
    Soma sproxy = new Soma();
```

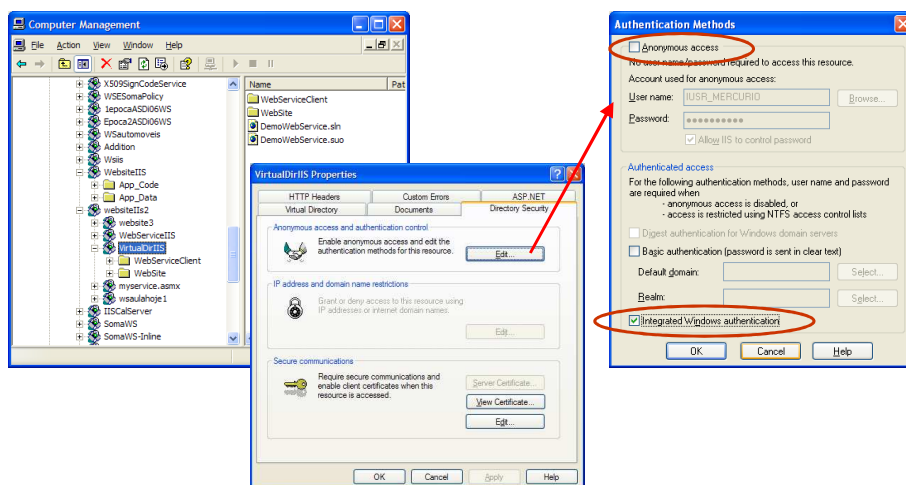
```
    int res=sproxy.add(int.Parse(textBox1.Text), int.Parse(textBox2.Text));  
    textBox3.Text=res.ToString();
```

```
}
```

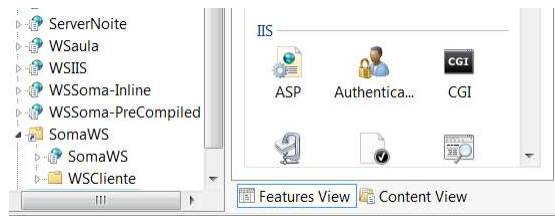


## Clientes do Web Service e a segurança do IIS !!!

Windows XP – IIS 6.0



## Clientes do Web Service e a segurança do IIS !!!

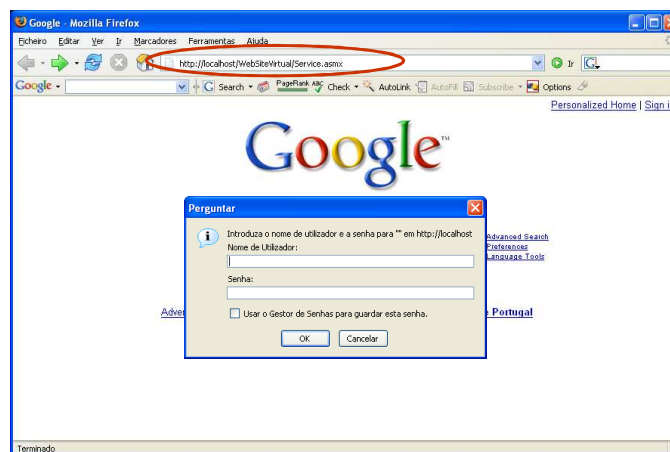


Windows Vista – IIS 7.0

Authentication	
Group by:	No Grouping
Name	Status
Anonymous Authentication	Enabled
ASP.NET Impersonation	Disabled
Forms Authentication	Disabled
Windows Authentication	Disabled

## Clientes do Web Service e a segurança do IIS !!!

- E se a directoria virtual do IIS não aceitar pedidos anónimos?



## Clientes do Web Service e a segurança do IIS !!!

- Tratamento do evento do botão <buttonInvoke> na aplicação Win Forms:

```
...
private void buttonInvoke_Click(object sender, EventArgs e) {
    Service svc = new Service();
    // antes da chamada podemos iniciar o proxy com credenciais
    svc.Credentials = CredentialCache.DefaultCredentials;

    this.textBoxHelloWorldResponse.Text = svc.HelloWorld();
}
...
```



Contém as credenciais do contexto de segurança corrente (*user name, password, e domain*) do utilizador que está *logged in*.  
Implica incluir o namespace: *System.Net*;

## Arquitectura dos Web Services em .NET / Relação com IIS

