

Curso: “*Tecnología Web*”

Profesores:

Jose Emilio Labra Gayo (Universidad de Oviedo, España)
Raúl Monge (UTFSM, Chile)

***** Curso sobre Tecnología Web ***** Versión 2005

Contenido

1.- Tecnologías XML

Definición y validación: DTDs, Espacios de nombres, XML Schema
Transformación y consulta de XML: XSLT, Xpath, Xquery
Programación XML: SAX, DOM

2.-Servicios Web

SOAP, WSDL, UDDI

3.-Web Semántica

Fundamentos, RDF, OWL

4.-Nuevas aplicaciones

***** Curso sobre Tecnología Web ***** Versión 2005

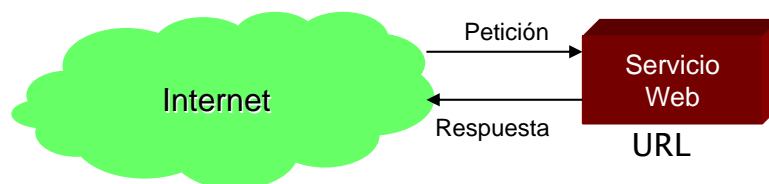
Sábado		
Clase 5	9h – 10:30h	Servicios Web SOAP, WSDL, UDDI
Desayuno		
Clase 6	10:45h – 12:45h	Práctica servicios Web Creación de clientes y servicios web
Almuerzo		
Clase 7	14h – 16h	Web Semántica Fundamentos, RDF, OWL
break		
Clase 8	16:15h – 17:45h	Práctica Web semántica Creación de ontologías Uso de razonadores
break		
Clase 9	18h – 19:30h	Nuevas aplicaciones Arquitecturas orientadas a servicios Coreografía y orquestación Servicios web semánticos

Servicios Web

Servicios Web Posible definición

Aplicaciones auto-contenidas, auto-descritas que pueden ser publicadas, localizadas e invocadas a través de la Web

Una vez desarrolladas, otras aplicaciones (y otros servicios Web) pueden descubrirlas e invocar el servicio dado



Factores que influyeron en su aparición

Computación Distribuida: RPC, CORBA, RMI, DCOM

Sistemas fuertemente acoplados

Integración de aplicaciones: EAI (*Enterprise Application Integration*)

Reacción frente a sistemas ERP monolíticos

Aparición de XML

Adopción por principales industrias

XML-RPC

Necesidad de intercambios B2B

Sistemas de integración EDI, RosettaNet, ebXML

Comercio electrónico y burbuja de Internet

Necesidad de nuevas fórmulas

Microsoft vs. Java

Compatibilidad

Servicios Web Objetivos

Independencia del lenguaje y de la plataforma

Separación de especificación de la implementación

Interoperabilidad

Utilización de estándares: XML, SOAP, WSDL, UDDI...

Acoplamiento débil: Sistemas basados en mensajes

Interacciones síncronas y asíncronas

A través de **Internet**

Sin control centralizado

Utilización de Protocolos establecidos

Consideraciones de seguridad

Modularidad y Reusabilidad de servicios

Escalabilidad: Uno-a-uno frente a uno-a-muchos

***** Curso sobre Tecnología Web ***** Versión 2005

Servicios Web Principales Vocabularios

Protocolo de transporte

HTTP/HTTPs (principalmente)

Codificación de datos y mensajes

SOAP (Simple Object Access Protocol)

Descripción del servicio

WSDL (Web Service Description Language)

Búsqueda y localización de servicios

UDDI (Universal Discovery, Description and Integration)

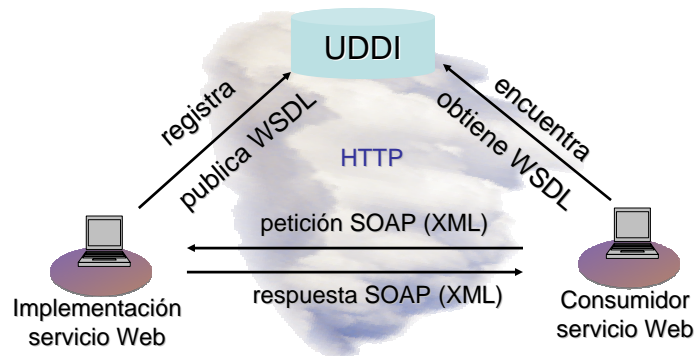
Otra definición

Programas accesibles en Internet que exponen su funcionalidad recibiendo/enviando mensajes SOAP a través de HTTP(s) y describen su interfaz en WSDL

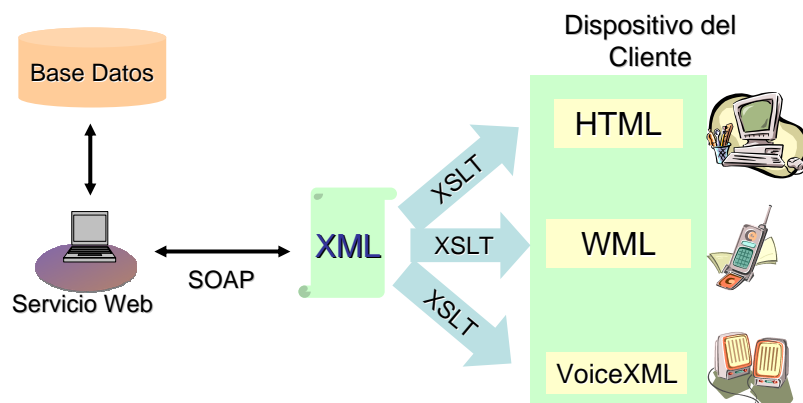
***** Curso sobre Tecnología Web ***** Versión 2005

Servicios Web

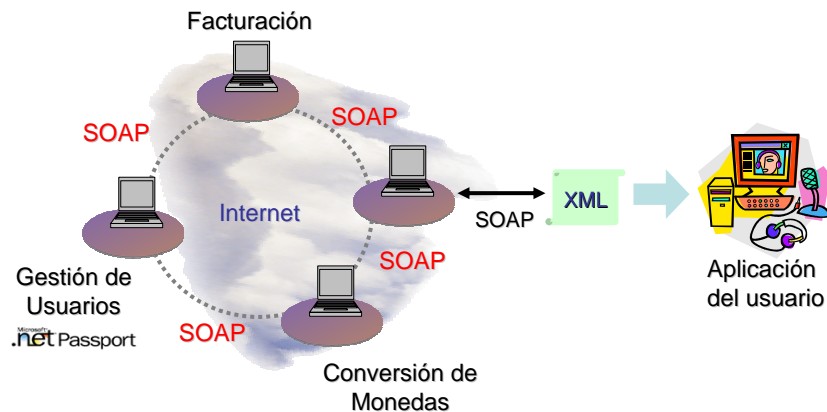
Principales Vocabularios



Arquitectura de Aplicaciones



Arquitectura de Aplicaciones



SOAP Evolución

SOAP: Define el formato de los mensajes
SOAP = Simple Object Access Protocol
Aunque tiene poco de objetos...

Evolución

Desarrollado a partir de XML-RPC
SOAP 1.0 (1999), 1.1 (2000), 1.2 (2002)
Participación inicial de Microsoft
Adopción posterior de IBM, Sun, etc.
Aceptación industrial



Contenido

SOAP Formato general

SOAP especifica el formato de mensajes

Es independiente del protocolo de transporte

Aunque se define un enlace (*binding*) con HTTP

envelope: Pueden especificarse datos globales
(codificación, espacios de nombres, etc.)

Contiene: **header** (opcional) + **body** (obligatorio)

body contiene datos en formato XML

header contiene meta-información

Extensiones obligatorias/opcionales

SOAP Header

header incluye información sobre el mensaje

Facilita futuras extensiones

Seguridad, transacciones, etc.

Información procesable por intermediarios

Atributos pre-definidos

mustUnderstand (true/false)

Si el elemento no puede procesar dicha información
devuelve un error

actor

Indica qué nodo debe procesar la información

Si no aparece, debe procesarla el nodo receptor final

fault: Formato predefinido de mensajes de error

Se incluye el elemento **fault** en el cuerpo

Subelementos predefinidos

faultcode: Código del error

Predefinidos: VersionMismatch, MustUnderstand,
DTDNotSupported, DataEncodingUnknown,
Sender, Receiver

faultstring: Explicación legible por personas

detail: Información específica de la aplicación

Puede contener elementos XML

faultactor: URI del nodo que causó el error

```
<?xml version='1.0' ?>
<soap:Envelope xmlns:soap='http://www.w3.org/2001/12/soap-envelope'>
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Receiver</faultcode>
      <faultstring>Error al procesar</faultstring>
      <detail>
        <p:detalles xmlns:p='http://www.mafia.it/pizzas'>
          <mensaje>La pizza Barbacoa no puede llevar
            tanto queso</mensaje>
        </p:detalles>
      </detail>
    </p:pizza>
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```

Atributo **encodingStyle** define reglas de codificación

Algunos tipos básicos predefinidos

- Enteros, cadenas, flotantes

Contiene reglas específicas para:

- Estructuras

- Arrays

- Referencias

Se complementa con XML Schemas

Pueden definirse otros sistemas de codificación

Tipos básicos

```
<?xml version='1.0' ?>
<soap:Envelope xmlns:soap='http://www.w3.org/2001/12/soap-envelope'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema'
  encodingStyle='http://www.w3.org/2001/12/soap-encoding'>
  <soap:Body>
    <p:pizza>
      <p:código xsi:type='soap:int'>234</p:comida>
      <p:tamaño xsi:type='soap:string'>familiar</p:tamaño>
    </p:pizza>
  </soap:Body>
</soap:Envelope>
```

Estructuras

```
struct Pizza {
  int código;
  string nombre;
};
```

```
<Pizza xmlns='cualquier_URI'>
  <código>234</código>
  <nombre>Barbacoa</nombre>
</Pizza>
```

Arrays

```
<pizzas xsi:type='soap:Array' soap:arrayType='p:Pizzas[2]'>
  <pizza> <código>234</código>
    <nombre>Barbacoa</nombre>
  </pizza>
  <pizza><código>237</código>
    <nombre>Barbacoa</nombre>
  </pizza>
</pizzas>
```

Arrays parciales

```
<pizzas xsi:type='soap:Array' soap:arrayType='p:Pizzas[10]'
  soap:offset='[4]'>
  <pizza> <código>234</código>
    <nombre>Barbacoa</nombre>
  </pizza>
  <pizza><código>237</código>
    <nombre>Barbacoa</nombre>
  </pizza>
</pizzas>
```

5º y 6º
elemento

```
<pizzas xsi:type='soap:Array' soap:arrayType='p:Pizzas[10]'>
  <pizza soap:position='2'> <código>234</código>
    <nombre>Barbacoa</nombre>
  </pizza>
  <pizza soap:position='5'><código>237</código>
    <nombre>Barbacoa</nombre>
  </pizza>
</pizzas>
```

2º y 5º
elemento

SOAP Ejemplo con HTTP

```
POST /Suma/Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: longitud del mensaje
SOAPAction: "http://tempuri.org/suma"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <suma xmlns="http://tempuri.org/">
      <a>3</a>
      <b>2</b>
    </suma>
  </soap:Body>
</soap:Envelope>
```

SOAP Ejemplo de respuesta

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: longitud del mensaje
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <sumaResponse xmlns="http://tempuri.org/">
      <sumaResult>5</sumaResult>
    </sumaResponse>
  </soap:Body>
</soap:Envelope>
```

WSDL (Web Services Description Language)

Describe:

Qué puede hacer el servicio

Dónde reside

Cómo invocarlo

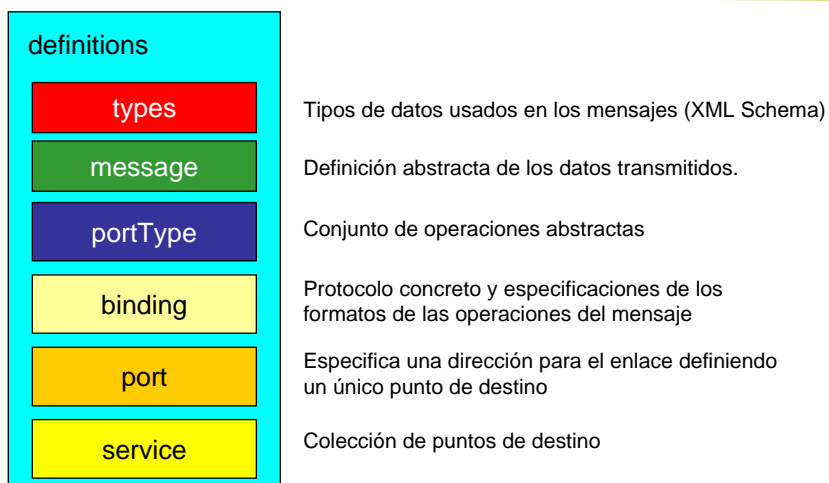
Vocabulario basado en capas

Es posible concentrarse en una capa cada vez

Evolución: Iniciativa conjunta de Ariba, IBM y Microsoft

(2001) Propuesto a W3C como recomendación (WSDL 1.1)

(2003) En desarrollo WSDL 2.0



WSDL Ejemplo

```

<?xml version="1.0" encoding="utf-8" ?>
<definitions xmlns:s=. . .
<types>
<s:schema
  <s:element name="suma">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="a" type="s:int" />
        <s:element minOccurs="1" maxOccurs="1" name="b" type="s:int" />
      </s:sequence>
    </s:complexType>
  </s:element>
  . . .
<message name="sumaSoapIn">
  <part name="parameters" element="s0:suma" />
</message>
. . .

```

***** Curso sobre Tecnología Web ***** Versión 2005

WSDL Ejemplo

```

. . .
<portType name="ServicioSumaSoap">
  <operation name="suma">
    <input message="s0:sumaSoapIn" />
    <output message="s0:sumaSoapOut" />
  </operation>
</portType>
. . .
<binding name="ServicioSumaSoap" type="s0:ServicioSumaSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="suma">
    <soap:operation soapAction="http://tempuri.org/suma" style="document" />
    <input> <soap:body use="literal" /> </input>
    <output> <soap:body use="literal" /> </output>
  </operation>
</binding>
<service name="ServicioSuma">
  <port name="ServicioSumaSoap" binding="s0:ServicioSumaSoap">
    <soap:address location="http://localhost/Suma/Service1.asmx" />
  </port>
</service>
</definitions>

```

5

UDDI (Universal Discovery, Description and Integration)

Consortio formado por IBM, Hp, Sun, Microsoft, Oracle, etc.

UDDI 1.0 (2000) Fundación del registro

UDDI 2.0 (2001) Alineación con estándares y taxonomía de servicios más flexible

UDDI 3.0 (2002) Interacción de implementaciones públicas y privadas

2 partes

Descripción de negocios

Páginas blancas (información de contacto)

“ amarillas (información de la industria)

“ verdes (información técnica y especificaciones)

Registro de servicios

Provider: Información sobre la entidad que ofrece el servicio

0...n

Service: Información descriptiva sobre una familia particular de ofertas

0...n

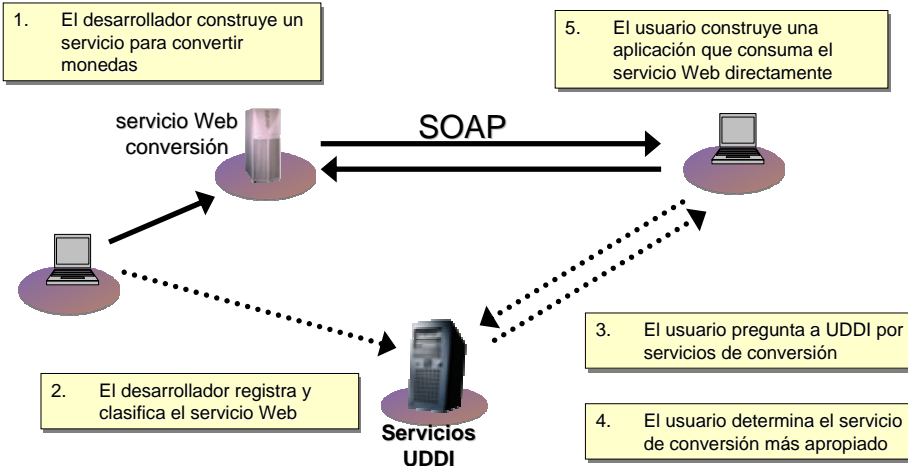
Binding: Información técnica sobre un punto de entrada a un servicio

tModel: Descripciones de especificaciones de servicios

Binding contiene referencias a *tModels*. Estas referencias declaran las especificaciones del interfaz

0...n

UDDI Funcionamiento



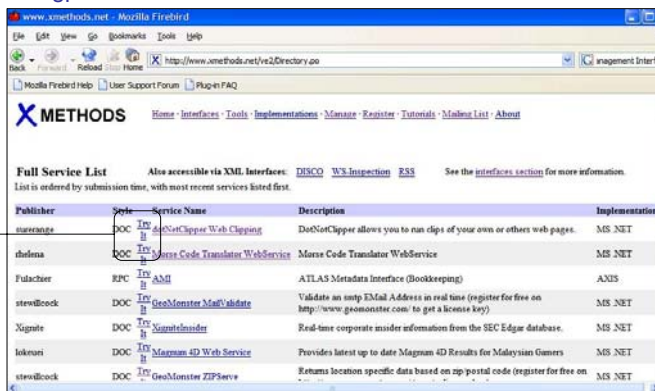
Uso de Servicio Web Ejemplos

Consultar listados de servicios Web

www.xmethods.net

www.bindingpoint.com

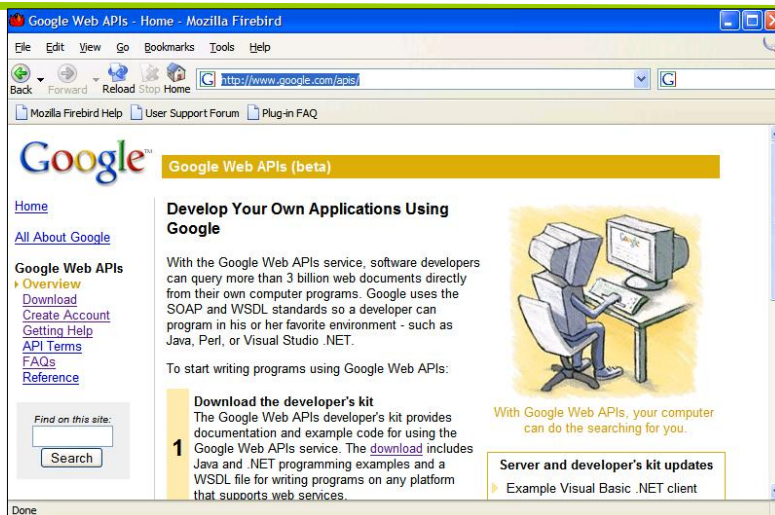
Pueden ejecutarse



Publisher	Service Name	Description	Implementation
merange	DotNetClipper Web Clipping	DotNetClipper allows you to run clips of your own or others web pages.	MS .NET
theloma	Morse Code Translator WebService	Morse Code Translator WebService	MS .NET
Fulcher	ATLAS Metadata Interface (Bookkeeping)	ATLAS Metadata Interface (Bookkeeping)	AXIS
stevillcock	GeoMonster MailValidate	Validate an smtp EMail Address in real time (register for free on http://www.geomonster.com/ to get a license key)	MS .NET
Xignite	Real-time corporate insider information from the SEC Edgar database.	Real-time corporate insider information from the SEC Edgar database.	MS .NET
tokruai	Magnum 4D Web Service	Provides latest up to date Magnum 4D Results for Malaysian Gamers	MS .NET
stevillcock	Examms location specific data based on zip/postal code (register for free on	Examms location specific data based on zip/postal code (register for free on	MS .NET

Uso de servicios Web

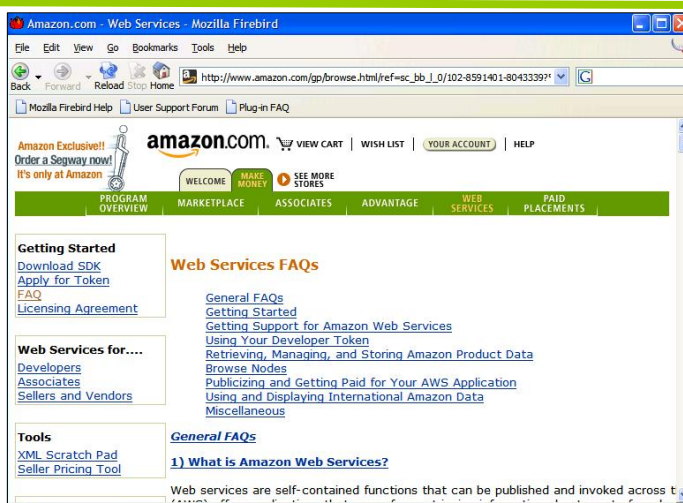
Ejemplos: Google



Curso sobre Tecnología Web Versión 2005

Uso de servicios Web

Ejemplos: Amazon



Curso sobre Tecnología Web Versión 2005

Implementación Servicios Web

Java

APIs de Sun: JAXRPC, JAXM, SAAJ,

Librerías de Apache: Axis

Microsoft .NET

ASP.NET para C#, VBasic, etc.

MS SOAP Toolkit

Otros:

SOAP::Lite (Perl), NuSOAP (PHP), Axis (C++)

Implementación: APIs Java

SAAJ (SOAP with Attachments API for Java)

Tratar mensajes SOAP como objetos Java

JAX-RPC (Java API for XML based RPC)

Modelo de programación

Conversión WSDL/XML ↔ Java

Manejo de SOAP y SOAP con Attachments

API para cliente: WSDL, Invocación y proxy dinámico

JWSDL

Acceso a descripciones WSDL

JAXR (Java API for XML Registries)

Acceso a registros de servicios Web (UDDI)

Implementación: Apache Axis

Sucesor de Apache SOAP (software abierto)

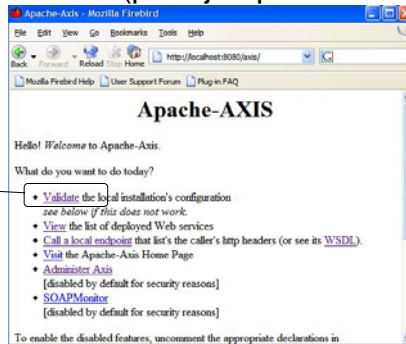
Soporta JAX-RPC y SAAJ

Arquitectura flexible y extensible

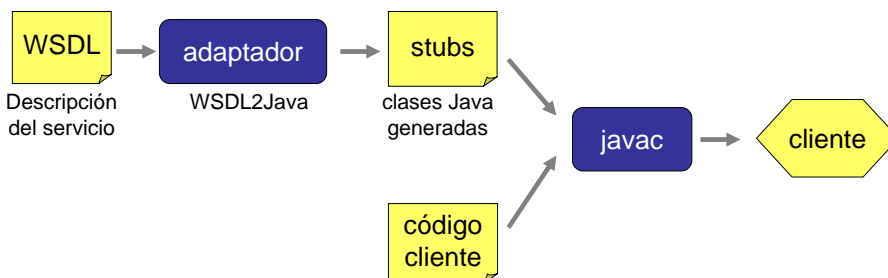
Necesita servidor de aplicaciones (por ejemplo Tomcat)

Validar la instalación:

<http://localhost:8080/axis>



Implementación: Creación de un Cliente



Implementación: Creación de un cliente

1.- Acceder a WSDL

```
http://petra.eutitio.uniovi.es/~labra/ws/suma.php?wsdl
```

Almacenar como suma.wsdl

2.-Generar stubs

```
> java org.apache.axis.wsdl.WSDL2Java -p suma suma.wsdl
```

3.- Comprobar clases generadas

```
> ls suma/*.java
ServicioSuma.java          ServicioSumaLocator.java
ServicioSumaBindingStub.java ServicioSumaPortType.java
```

4.- Compilar clases generadas

```
> javac suma/*.java
```

Implementación: Creación de un cliente

ClienteSuma.java

```
import suma.*;
public class ClienteSuma {
    public static void main(String[] args) throws Exception {
        try {
            ServicioSumaLocator loc = new ServicioSumaLocator();
            ServicioSumaPortType p = loc.getServicioSumaPort();
            System.out.println("2 + 3 = " + p.suma(2,3));
        } catch (Exception e) {
            System.err.println("Excepción: " + e);
        }
    }
}
```

4.- Compilar cliente

```
> javac ClienteSuma.java
```

5.-Ejecutar cliente

```
> java ClienteSuma
2 + 3 = 5
```

Implementación: Creación de un cliente

Ejercicio: Consultar temperatura del aeropuerto de Avilés...

<http://live.capescience.com/wsdl/GlobalWeather.wsdl>

ClienteTemp.java

```
public class ClienteTemp {  
    public static void main(String args[]) throws Exception {  
        try {  
            GlobalWeather_ServiceLocator loc = new GlobalWeather_ServiceLocator();  
            GlobalWeather_Port s = loc.getGlobalWeather();  
            System.out.println("Temperatura en Aeropuerto de Asturias: " +  
                s.getWeatherReport("LEAS").getTemperature().getString());  
        } catch (Exception e) {  
            System.err.println("Excepción: " + e);  
        }  
    }  
}
```

***** Curso sobre Tecnología Web ***** Versión 2005

Implementación: Creación Servicio Web

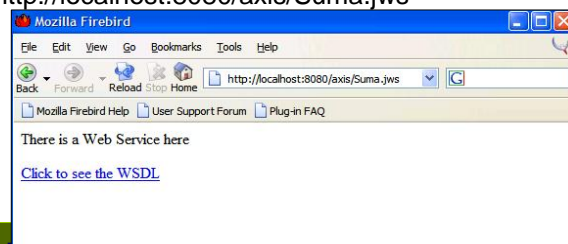
Método simple: JWS

Suma.jws

```
public class Suma {  
    public int suma(int a, int b) {  
        return a + b;  
    }  
}
```

Almacenar en:
<TOMCAT>\webapps\axis\Suma.jws

<http://localhost:8080/axis/Suma.jws>



***** Curso sobre Tecnología Web ***** Versión 2005

Implementación: Creación Servicio Web

Utilizar JWS tiene sus limitaciones

- Debe disponerse del código fuente
- Los errores aparecen en tiempo de ejecución
- La clase no puede tener *package*
- Sólo se pueden transferir datos simples
- No se puede configurar el servicio

Método riguroso: WSDD (Web Service Deployment Descriptor)

- Permite desplegar (deploy) y quitar (undeploy) servicios
- Pueden utilizarse servicios compilados
- Control de las Conversiones de tipos

Implementación: Creación Servicio Web

ServSuma.java

```
package ServSuma;  
  
public class ServSuma {  
    public int suma(int a, int b){  
        return (a + b);  
    }  
}
```

1.- Compilar servicio

```
> javac ServSuma.java
```

2.-Copiar ServSuma.class a

<TOMCAT>/webapps/WEB-INF/classes/ServSuma/ServSuma.class

También puede dejarse un .jar en WEB-INF/lib

Implementación: Creación Servicio Web

deploy.wsdd

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
  xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
  <service name="ServSuma" provider="java:RPC">
    <parameter name="className" value="ServSuma.ServSuma"/>
    <parameter name="allowedMethods" value="*/"/>
  </service>
</deployment>
```

3.- Desplegar servicio

```
> java org.apache.axis.client.AdminClient deploy.wsdd
Processing file deploy.wsdd
<Admin>Done processing</Admin>
```

Puede ser necesario reiniciar servidor

4.- Acceder a

<http://localhost:8080/axis/services/ServSuma>

***** Curso sobre Tecnología Web ***** Versión 2005

Otras características de Axis

Invocación dinámica

Dynamic Invocation Interface

Invocación mediante Proxy

Conversión Java2WSDL

Permite generar WSDL a partir de clases/interfaces
Java

Generación de ficheros WSDD para
deploy/undeploy

Seguridad

Otros protocolos de transporte

***** Curso sobre Tecnología Web ***** Versión 2005

Interoperabilidad: Acceso desde .NET a servicio Java

1.- Acceso a WSDL y creación de Stubs (o proxys)

```
> wsdl http://localhost:8080/axis/services/ServSuma?wsdl
...
Writing file 'C:\usr\labra\cursos\XMLInnova\WebServ\ClienteNet\ServSumaService.cs'.
```

En algunas versiones es necesario editar ServSumaService.cs y modificar this.URL para que incluya el puerto 8080

2.- Compilación de proxys

```
> csc /t: library ServSumaService.cs
```

3.- Creación de cliente

cliente.cs

```
using System;

public class ClienteSumaNet {
    public static void Main() {
        ServSumaService srv = new ServSumaService();
        Console.WriteLine("2 + 3 = {0}", srv.suma(2,3));
    }
}
```

4.- Compilación de cliente

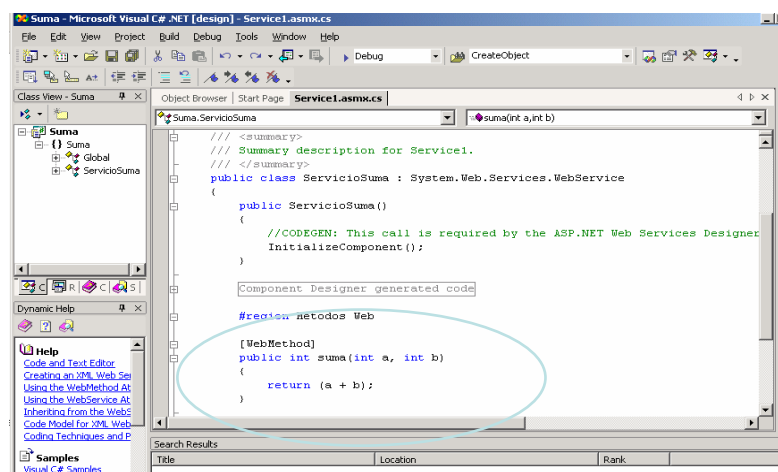
```
> csc cliente.cs /reference:ServSumaService.cs
```

5.- Ejecución

```
> cliente
2 + 3 = 5
```

Versión 2005

Interoperabilidad Servicios Web en .NET



Curso sobre Tecnología Web

Versión 2005



MTI
INTERNACIONAL
MAESTRER EN TECNOLOGÍAS
DE LA INFORMACIÓN

Interoperabilidad Servicios Web en .NET

Archivo Edición Ver Favoritos Herramientas Ayuda
http://localhost/Suma/Service1.aspx?op=suma

ServicioSuma

Click [here](#) for a complete list of operations.

suma

Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Value
a:	<input type="text"/>
b:	<input type="text"/>

SOAP

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /Suma/Service1.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/suma"

<?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>
```

***** Curso sobre Tecnología Web *****
Versión 2005



MTI
INTERNACIONAL
MAESTRER EN TECNOLOGÍAS
DE LA INFORMACIÓN

Interoperabilidad Servicios Web en .NET

http://localhost/Suma/Service1.aspx?wsdl - Microsoft Internet Explorer proporcionado por Navegación
http://localhost/Suma/Service1.aspx?wsdl

```
<?xml version="1.0" encoding="utf-8" ?>
- <definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:s0="http://tempuri.org/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="http://tempuri.org/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  >
  <types>
  <message name="sumaSoapIn">
  <message name="sumaSoapOut">
  <message name="autorSoapIn">
  <message name="autorSoapOut">
  <message name="sumaHttpGetIn">
  <message name="sumaHttpGetOut">
  <message name="autorHttpGetIn">
  <message name="autorHttpGetOut">
  <message name="sumaHttpPostIn">
  <message name="sumaHttpPostOut">
  <message name="autorHttpPostIn">
  <message name="autorHttpPostOut">
  <portType name="ServicioSumaSoap">
  <portType name="ServicioSumaHttpGet">
  <portType name="ServicioSumaHttpPost">
  <binding name="ServicioSumaSoap" type="s0:ServicioSumaSoap">
  <binding name="ServicioSumaHttpGet" type="s0:ServicioSumaHttpGet">
  <binding name="ServicioSumaHttpPost" type="s0:ServicioSumaHttpPost">
  <service name="ServicioSuma">
```

***** Curso sobre Tecnología Web *****
Versión 2005

Interoperabilidad Servicios Web en PHP

suma.php

```

<?php
include "nusoap.php";

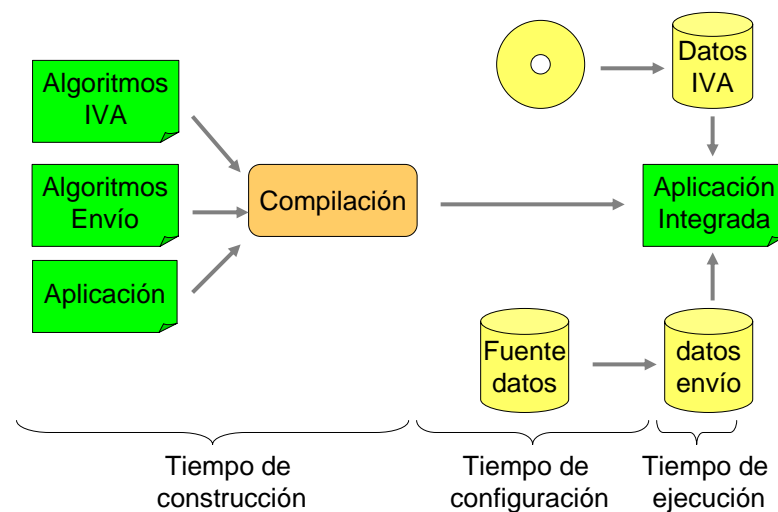
$namespace = "http://petra.euitio.uniovi.es/~labra/ws/suma.php?wsdl";
$servidor = new soap_server;
$servidor->configureWSDL ("ServicioSuma", $namespace,
    "http://petra.euitio.uniovi.es/~labra/ws/suma.php");
$servidor->wsdl->schemaTargetNamespace = $namespace;
$servidor->register ('suma', array ('a' => 'xsd:float', 'b' => 'xsd:float'),
    array ('return' => 'xsd:float'),
    'http://petra.euitio.uniovi.es/~labra/ws/suma.php', "", "", "");
$servidor->service ($HTTP_RAW_POST_DATA);

function suma ($a, $b) {
    if (!$a || !$b) {
        return new soap_fault ("Client", "", "Se necesitan dos argumentos");
    }
    if ((gettype ($a) != "integer" && gettype ($a) != "double") ||
        (gettype ($b) != "integer" && gettype ($b) != "double")) {
        return new soap_fault ("Client", "", "El tipo debe ser entero o real");
    }
    return $a + $b;
}
?>
    
```

Curso sobre Tecnología Web

Versión 2005

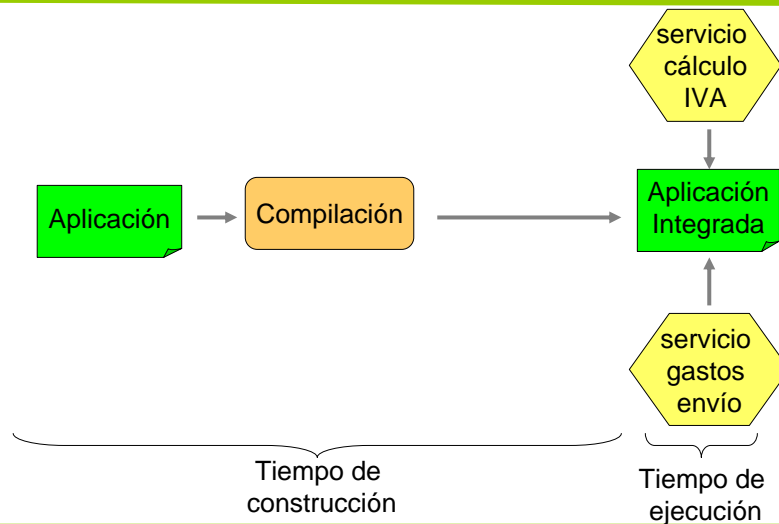
Arquitecturas Orientadas a Servicios Modelo tradicional



Curso sobre Tecnología Web

Versión 2005

Arquitecturas Orientadas a Servicios Modelo Orientado a Servicios



Curso sobre Tecnología Web

 Versión 2005

Arquitecturas Orientadas a Servicios Principales características

Importancia de las *interfaces*

- Descripción rigurosa de interfaces (legibles por máquinas)
- Recomendación: Partir de WSDL + XML Schema

Modelos *débilmente acoplados*

- Sistemas de comunicación asíncrona
- Estilo documento vs. estilo RPC
- Colas de mensajes
- Ej. Solicitar un libro

Interoperabilidad

- Independencia de lenguajes y plataformas
- Adaptación de arquitecturas ya existentes
- Utilización de estándares
- Modelo REST vs SOAP

Curso sobre Tecnología Web

 Versión 2005

Servicios Web Retos

Gestión de servicios Web

WSDM - Web Services Distribution Management

Agregación de servicios

Ejemplo. Reserva de avión + hotel

Evolución de los servicios

Cambio de la Interfaz

Modelización de **procesos de negocios**

BPEL - Business Process Execution Language

Contratos, facturación

¿Quién gana dinero? ¿Qué pasa cuando algo falla?

Seguridad y fiabilidad

XML Security

Calidad de servicios

Tiempos de respuesta, soporte, monitorización, etc.

***** Curso sobre Tecnología Web ***** Versión 2005

Más información



www.wsindex.org

Información de servicios Web y Web semántica

www.searchwebservice.com

Portal de servicios Web orientado a empresas

www.webservices.org

Sobre servicios Web

www.xmethods.net

Lista de servicios Web

www.soapware.org

Portal sobre SOAP

www.w3c.org/2002/ws

Especificaciones relacionadas con servicios Web

***** Curso sobre Tecnología Web ***** Versión 2005

Repaso... Tecnologías XML

Definición de XML

- DTD
- Namespaces
- XML Schema

Transformación de XML

- XSL = XSLT + XPath + XSL-FO

Vocabularios XML

- XHTML, MathML, SVG, SMIL, WML, VoiceXML

Tratamiento de XML

- SAX, DOM, JAXP, JAXB

Transporte de XML: Servicios Web

- SOAP, WSDL, UDDI