

INTEGRACION DE SOLUCIONES

- Tecnologías de Servicios Web

Referencias

Microsoft Corporation. (Marzo de 2006). Windows Communication Foundation Architecture Overview. Obtenido de Microsoft Developer Network: <https://msdn.microsoft.com/en-us/library/aa480210.aspx>

OASIS. (s.f.). XML.ORG. Obtenido de Sitio Oficial del lenguaje XML: <http://www.xml.org/>

Sharp, J. (2007). Windows Communication Foundation Step by Step. Redmond, Washington: Microsoft Press.

World Wide Web (W3C) España. (s.f.). Guía Breve de Servicios Web. Obtenido de Sitio Web del Consorcio World Wide Web (W3C) en España: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

World Wide Web Consortium (W3C). (s.f.). W3C. Obtenido de Sitio Web del World Wide Web Consortium (W3C): <https://www.w3.org/>

Web Services and Service-Oriented Architectures. The Savy Manager's Guide. Douglas K. Barry. USA 2003

Agenda

- Arquitecturas Orientadas a Servicios y WCF
- Introducción a Windows Communication Foundation
- Alternativas a WCF
- Construyendo un Servicio WCF

Tecnologías de Servicios Web

Servicio Web es una función bien definida, auto-contenida y no depende del contexto o estado de otros servicios.

Ejemplo: API de Google Maps

<https://developers.google.com/maps/web-services/overview?hl=es-419>

Tecnologías de Servicios Web

Servicios

- ✓ Componente reutilizable de software.

Contrato

Especificación de la finalidad, funcionalidad, forma de uso y restricciones del servicio.

Implementación

Contiene la lógica o el acceso a datos

Interfaz

Mecanismo de exposición del servicio a los usuarios

Tecnologías de Servicios Web

SOAP (Simple Object Access Protocol)

- **SOAP** (*Simple Object Access Protocol*) es un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja
- Los datos pueden ser transmitidos a través de HTTP, SMTP, etc. SOAP especifica el formato de los mensajes

Comunicación

Figura. Conceptos básicos de Servicios Web



Comunicación

1. Un proveedor de servicio describe su servicio usando Web Services Description Language (WSDL). Esta definición es publicada en el directorio de servicios. El directorio puede usar Description, Discovery and Integration (UDDI).
2. Un consumidor de servicios emite una o más consultas al directorio para localizar un servicio y determinar cómo comunicarse con él.
3. Parte del WSDL provisto por el proveedor de servicio es transmitido al Consumidor de servicio. Esto dice al consumidor de servicio cuales de las solicitudes y respuestas son para el proveedor de servicio.
4. El consumidor de servicio usa WSDL para enviar una solicitud al proveedor.
5. El proveedor de servicio otorga la respuesta esperada al consumidor.

```
<element name="ClienteInfoSolicitud">
  ....
  <element name="cuenta" tipo="string"/>
  ....
</element>
<element name="ClienteInfoRespuesta">
  <element name="nombre" tipo="string"/>
  <element name="telefono" tipo="string"/>
  ...
</element>
```



Directorio

Descripción del servicio usando WSDL

Respuestas a consultas usando WSDL

```
<m:ObtenerClienteInfo...>
  <cuenta>1823</cuenta>
</m:ObtenerClienteInfo>
```

Proveedor de servicio



Solicitud de servicio XML basado en WSDL

Respuesta de servicio XML basado en WSDL



Consumidor de servicio

```
<m:ObtenerClienteInfoResp...>
  <nombre>Universidad Veracruzana</nombre>
  <telefono>228-8-42-17-00</telefono>
  ...
</m:ObtenerClienteInfoResp >
```

Tecnologías de Servicios Web

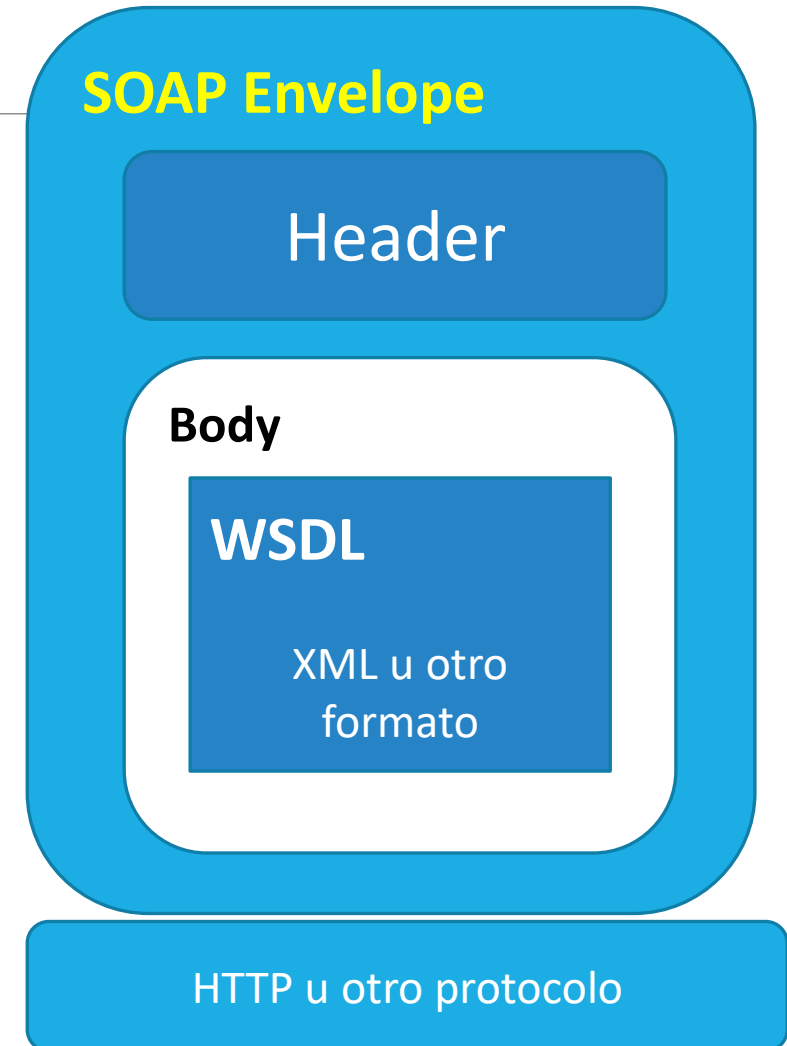
SOAP (Simple Object Access Protocol)

- **SOAP** (*Simple Object Access Protocol*) es un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja
- Los datos pueden ser transmitidos a través de HTTP, SMTP, etc. SOAP especifica el formato de los mensajes

Tecnologías de Servicios Web

SOAP (Simple Object Access Protocol)

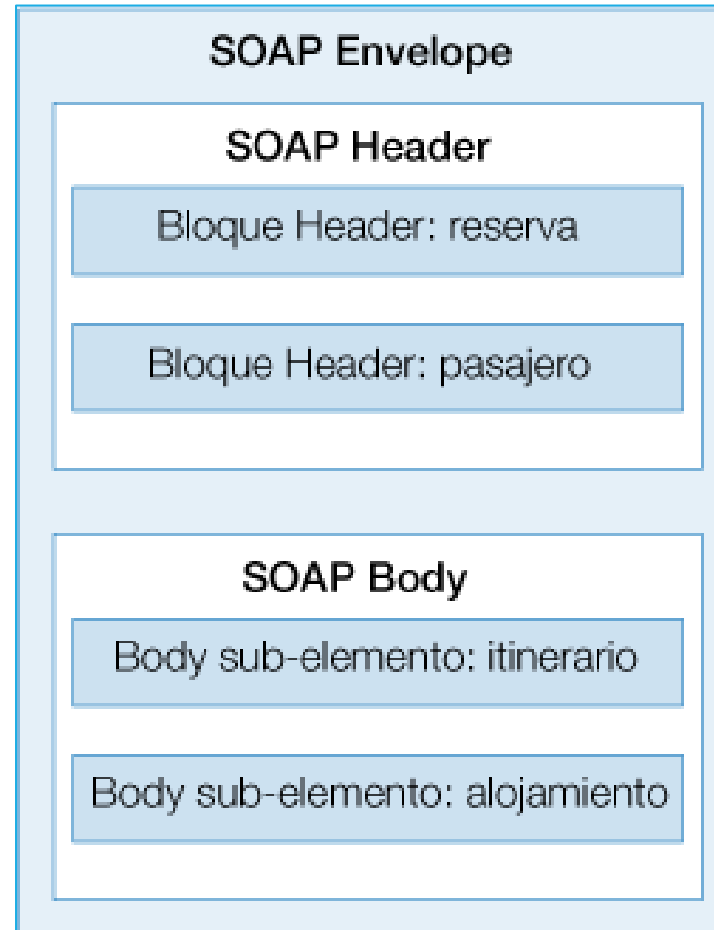
El mensaje SOAP está compuesto por un envelope (sobre), cuya estructura está formada por los siguientes elementos: header (cabecera) y body (cuerpo)



Clase 15. Tecnologías de Servicios Web

SOAP (Simple Object Access Protocol)

El mensaje SOAP está compuesto por un envelope (sobre), cuya estructura está formada por los siguientes elementos: header (cabecera) y body (cuerpo)



```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reserva xmlns:m="http://empresaviajes.ejemplo.org/reserva"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <m:referencia>
        uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d
      </m:referencia>
      <m:fechaYHora>2001-11-29T13:20:00.000-05:00</m:fechaYHora>
    </m:reserva>
    <n:pasajero xmlns:n="http://miempresa.ejemplo.com/empleados"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <n:nombre>Pepe Ejemplo</n:nombre>
    </n:pasajero>
  </env:Header>
  <env:Body>
    <p:itinerario
      xmlns:p="http://empresaviajes.ejemplo.org/reserva/viaje">
      <p:ida>
        <p:salida>Nueva York</p:salida>
        <p:llegada>Los Angeles</p:llegada>
        <p:fechaSalida>2001-12-14</p:fechaSalida>
        <p:horaSalida>última hora de la tarde</p:horaSalida>
        <p:preferenciaAsiento>pasillo</p:preferenciaAsiento>
      </p:ida>
      <p:vuelta>
        <p:salida>Los Angeles</p:salida>
        <p:llegada>Nueva York</p:llegada>
        <p:fechaSalida>2001-12-20</p:fechaSalida>
        <p:horaSalida>media-mañana</p:horaSalida>
        <p:preferenciaAsiento/>
      </p:vuelta>
    </p:itinerario>
    <q:alojamiento
      xmlns:q="http://empresaviajes.example.org/reserva/hoteles">
      <q:preferencia>ninguna</q:preferencia>
    </q:alojamiento>
  </env:Body>
</env:Envelope>
```



```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reserva xmlns:m="http://empresaviajes.ejemplo.org/reserva"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <m:referencia>
        uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d
      </m:referencia>
      <m:fechaYHora>2001-11-29T13:20:00.000-05:00</m:fechaYHora>
    </m:reserva>
    <n:pasajero xmlns:n="http://miempresa.ejemplo.com/empleados"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <n:nombre>Pepe Ejemplo</n:nombre>
    </n:pasajero>
  </env:Header>
  <env:Body>
    <p:itinerario
      xmlns:p="http://empresaviajes.ejemplo.org/reserva/viaje">
      <p:ida>
        <p:salida>Nueva York</p:salida>
        <p:llegada>Los Angeles</p:llegada>
        <p:fechaSalida>2001-12-14</p:fechasalida>
        <p:horaSalida>última hora de la tarde</p:horaSalida>
        <p:preferenciaAsiento>pasillo</p:preferenciaAsiento>
      </p:ida>
      <p:vuelta>
        <p:salida>Los Angeles</p:salida>
        <p:llegada>Nueva York</p:llegada>
        <p:fechaSalida>2001-12-20</p:fechaSalida>
        <p:horaSalida>media-mañana</p:horaSalida>
        <p:preferenciaAsiento/>
      </p:vuelta>
    </p:itinerario>
    <q:alojamiento
      xmlns:q="http://empresaviajes.example.org/reserva/hoteles">
      <q:preferencia>ninguna</q:preferencia>
    </q:alojamiento>
  </env:Body>
</env:Envelope>
```

Tecnologías de Servicios Web

WSDL (Web Services Description Language). Es un formato para describir una interfaz de Servicios Web.

Tecnologías de Servicios Web

WSDL (Lenguaje de Descripción de Servicios Web)

- **WSDL** (*Lenguaje de Descripción de Servicios Web*), permite que un **servicio** y un **cliente** establezcan un **acuerdo** en lo que se refiere a los detalles de **transporte de mensajes y su contenido**, a través de un documento procesable por dispositivos
- WSDL representa una especie de contrato entre el proveedor y el que solicita. WSDL especifica la sintaxis y los mecanismos de intercambio de mensajes



Tecnologías de Servicios Web

WSDL Tiene tres partes:

Definiciones

Operaciones

Enlace de servicios

Tecnologías de Servicios Web. WSDL.

DEFINICIONES

Las definiciones son generalmente expresadas en XML e incluyen definiciones de tipos de datos y definiciones de mensajes que usan las definiciones de tipos de datos.

OPERACIONES

Las operaciones describen acciones para los mensajes soportados por un Servicio Web.

Cuatro tipos de operaciones: One-way, Request/response, Solicit response, Notification.

ENLACES

Las operaciones están agrupadas en tipos de puerto.

Los enlaces de servicio conectan tipos de puerto a un puerto.

Clase 15. Tecnologías de Servicios Web

Definiciones

Definiciones de tipos de datos

Definiciones de mensajes

Definiciones de tipos de datos

Definiciones de mensajes

Definiciones de tipos de datos

Definiciones de mensajes

Operaciones

Operación

Operación

Operación

Tipo de
puerto

Enlaces de servicios

Enlaces

Puerto y
Direcciones de red

Tecnologías de Servicios Web

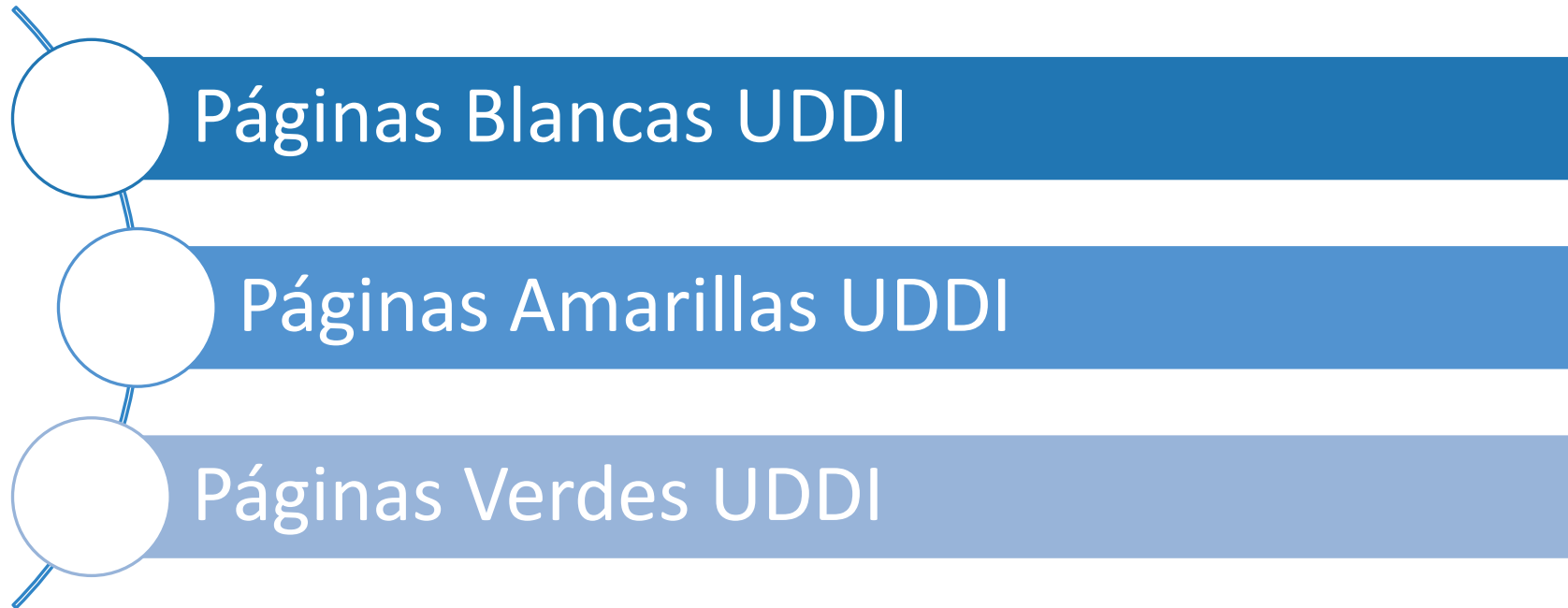
UDDI (Universal Description, Discovery and Integration). Provee la definición de un conjunto de servicios, soportando la descripción y descubrimiento de:

Tecnologías de Servicios Web

- 1) Negocios, organizaciones y otros proveedores de Servicios Web.
- 2) Los servicios Web disponibles.
- 3) Las interfaces técnicas que pueden ser usadas para acceder a los servicios.

Tecnologías de Servicios Web

El sistema de registro de Negocio UDDI contiene tres directorios:



Arquitecturas Orientadas a Servicios y Windows Communication Foundation (WCF)

Tecnologías de Servicios Web

Arquitecturas Orientadas a Servicios (SOA)

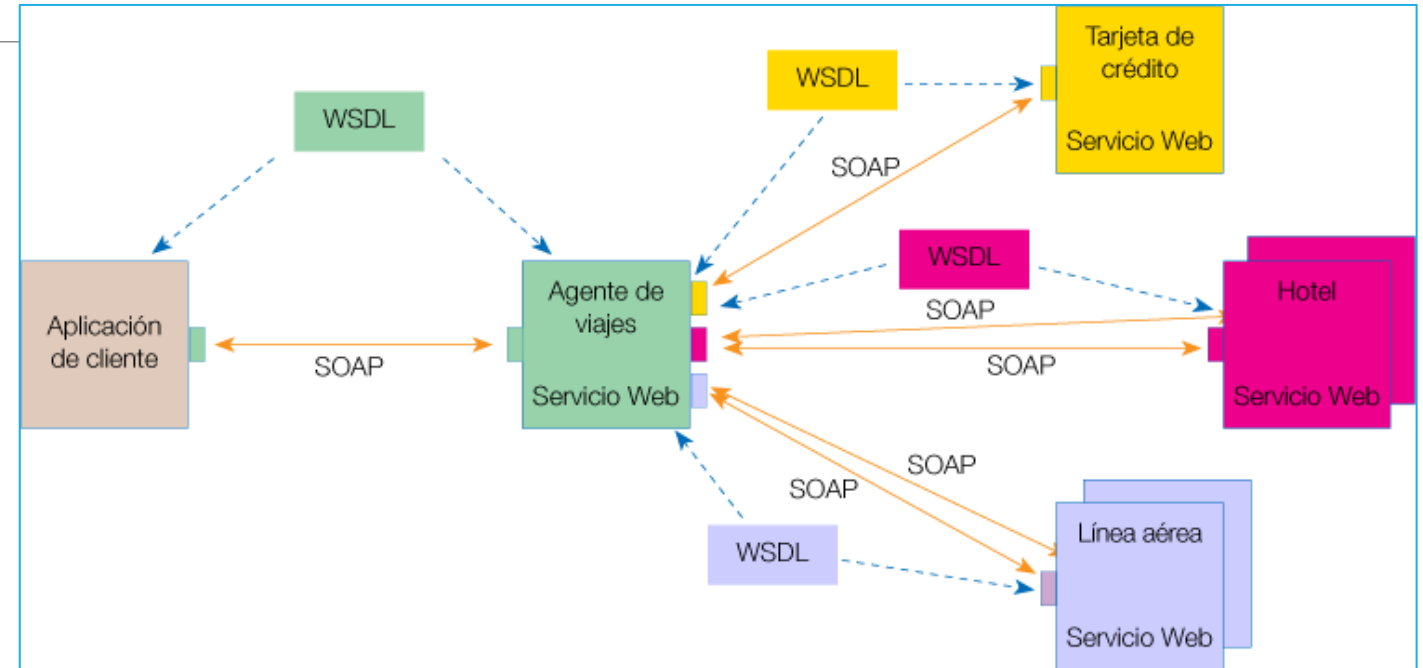
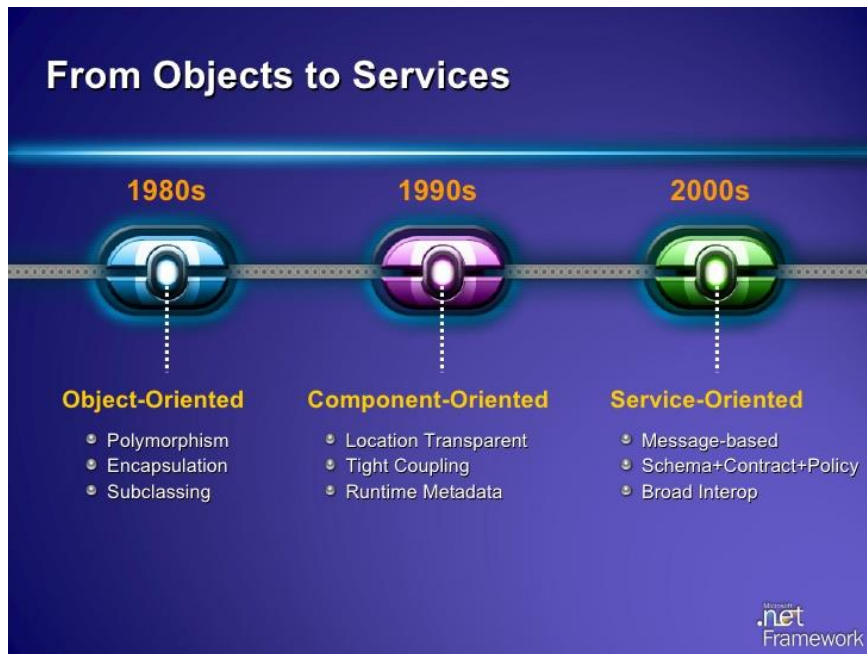
SOA consiste en un grupo de recursos en una red que están disponibles como servicios independientes y pueden ser accedidos sin conocimiento de como están implementados

Windows Communication Foundation (WCF) es una plataforma ideal para implementar una Arquitectura Orientada a Servicios o SOA



Tecnologías de Servicios Web

Servicios Web



Los **Servicios Web** son aplicaciones o tecnologías que intercambian datos entre sí, con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web

Tecnologías de Servicios Web

4 principios de la orientación a servicios

Para diseñar e implementar SOA, se deben conocer los **4 principios de la orientación a servicios** (The Four Tenets of Service Orientation)



Tecnologías de Servicios Web

4 principios de la orientación a servicios

Para diseñar e implementar SOA, se deben conocer los **4 principios de la orientación a servicios** (The Four Tenets of Service Orientation):

- 1. Los límites son explícitos.** No se deben hacer suposiciones de como un servicio o cliente procesarán las peticiones. Todo debe ser explícito.
- 2. Los servicios son autónomos.** Pueden ser desconectados o modificados.



Tecnologías de Servicios Web

4 principios de la orientación a servicios

Para diseñar e implementar SOA, se deben conocer los **4 principios de la orientación a servicios** (The Four Tenets of Service Orientation):

- 3. Los servicios comparten esquema y contrato.** Los Servicios publican sus operaciones y la estructura de sus datos mediante interfaces.

Si se desea modificar el servicio, estas interfaces deben actualizarse añadiendo nuevos contratos y esquemas pero continúan funcionando los existentes para los clientes actuales.



Tecnologías de Servicios Web

4 principios de la orientación a servicios

Para diseñar e implementar SOA, se deben conocer los **4 principios de la orientación a servicios** (The Four Tenets of Service Orientation):

- 4. La compatibilidad del servicio se basa en políticas.** El servicio y el cliente deben conocer las políticas de requerimientos del servicio de antemano.



Introducción a Windows Communication Foundation

Introducción a Windows Communication Foundation

ANTECEDENTES

Tecnologías de Servicios Web

Introducción a Windows Communication Foundation

Inicios de las aplicaciones para Computadoras Personales

- En los inicios, las PCs no requerían mucha interconectividad.

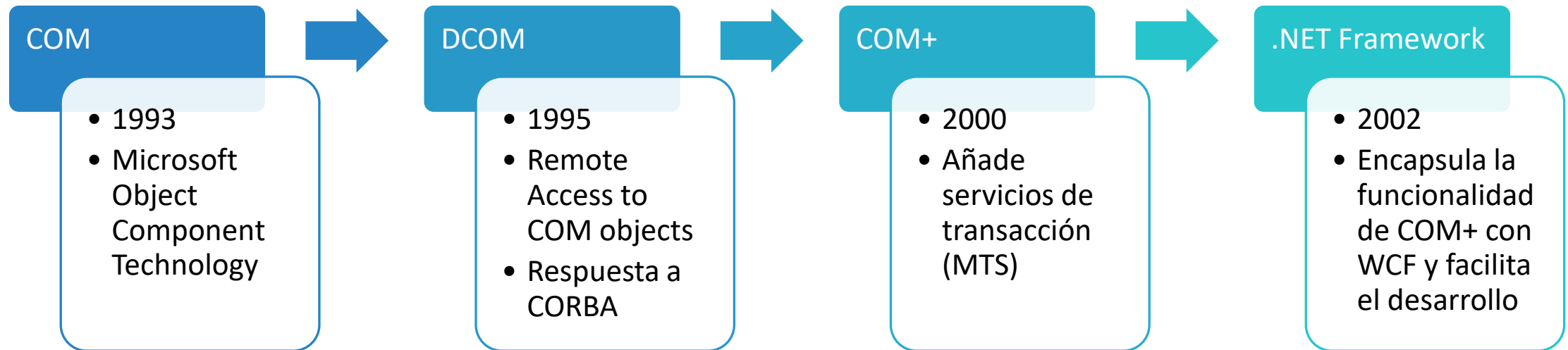


Tecnologías de Comunicación Inter-Proceso

- Con los avances de soluciones de red, las soluciones de negocios podían compartir datos más fácilmente.
- Ahora las computadoras podían comunicarse entre sí
- MS creó los Objetos COM, DCOM y finalmente COM+ para acceder a componentes ejecutándose en otros equipos.
- Esto evolucionó con el .NET Framework cambiando a Enterprise Services.

Tecnologías de Servicios Web

Evolución de las comunicaciones entre dispositivos



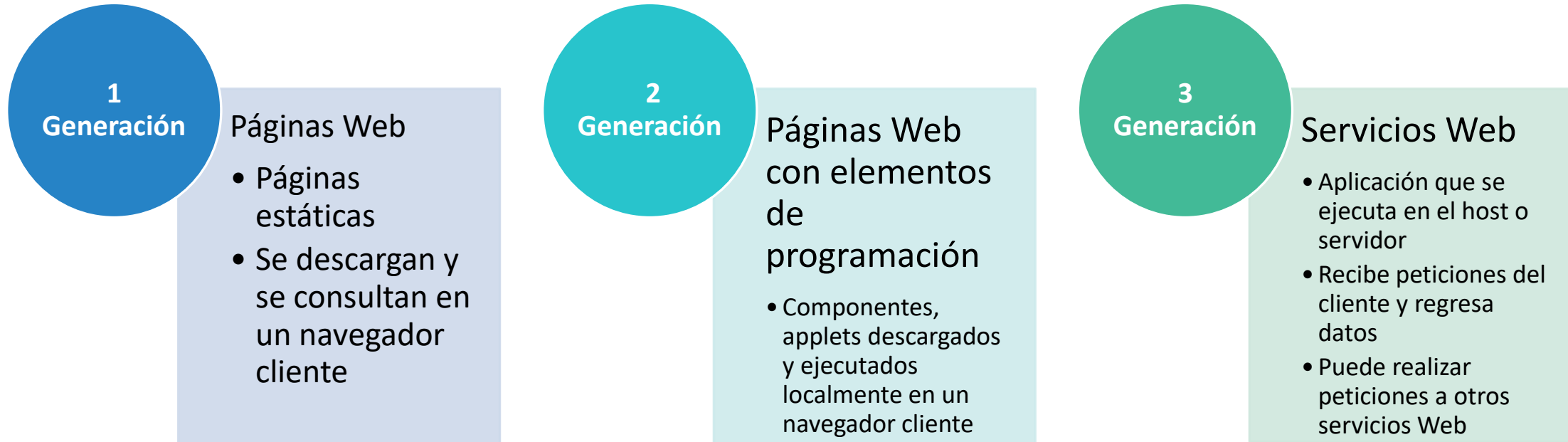
Tecnologías de Servicios Web

La Web y los Servicios Web



Mientras MS desarrollaba COM, DCOM, COM+ y el .NET, la Web apareció.

La Web permite comunicación entre dispositivos de distintas arquitecturas de todo el mundo (no sólo Windows).



Introducción a Windows Communication Foundation

ELEMENTOS



Tecnologías de Servicios Web

Usando XML como un Formato Común de Datos

Para compartir datos entre aplicaciones ejecutándose en diferentes computadoras, los desarrolladores deben tener un formato común independiente de la arquitectura.

eXtensible Markup Language o XML es un Formato de datos aceptado universalmente.

- Basado en texto
- Humanamente leíble
- Permite definir una gramática para describir cualquier tipo de datos
- Aplicaciones deben estar de acuerdo en diseño o esquema de los datos

```
<Persona> ·  
· · · · <Nombre>Juan</Nombre> ·  
· · · · <Apellido>Ochoa</Apellido> ·  
· · · · <Edad>22</Edad> ·  
</Persona>
```

```
<Persona Nombre="Juan" Apellido="Ochoa" Edad="22" />
```

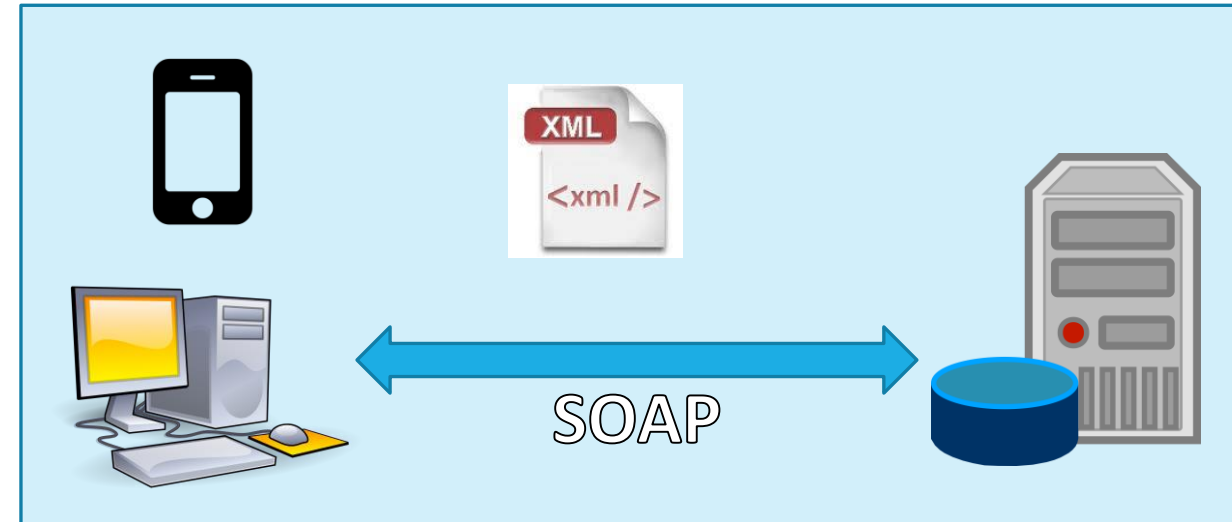


Tecnologías de Servicios Web

Además de XML se requiere un protocolo para enviar y recibir peticiones

Simple Object Access Protocol. Esta especificación **SOAP** define:

- El formato del mensaje SOAP
- Como deben ser codificados los datos
- Como enviar los mensajes
- Como manejar las respuesta a estos mensajes



Tecnologías de Servicios Web

Usando el protocolo SOAP y XML como el formato común de los mensajes queda una pregunta:

¿Cómo saber los mensajes que un cliente envía hacia un Servicio Web y la respuesta que recibirá?

- Mediante un documento **Web Services Description Language (WSDL)** un servicio Web puede conocer los mensajes que un cliente puede enviar y la respuesta que recibirá.



Tecnologías de Servicios Web

Seguridad en WCF

Seguridad en el envío de mensajes

- La organización encargada de proponer mecanismos estándar de seguridad se llama **Organization for the Advancement of Structured Information Standards (OASIS)**
 - Pares de username/passwords
 - Certificados X.509 (ssl)
 - Tokens
 - Entre otros.

Privacidad en el envío de mensajes

- Cliente y servicio Web deben estar de acuerdo en los mecanismos para asegurar la privacidad de su conversación.
 - Encriptación de mensajes
 - Llaves públicas y privadas



Tecnologías de Servicios Web

Service Endpoints

Un host pone disponible un servicio web por medio de un endpoint a donde los clientes pueden realizar peticiones y consiste en 3 piezas:

1. La dirección del servicio. Depende del protocolo. Por ejemplo HTTP: `http://localhost:50551/Service1.svc`
2. El **binding** soportado por el servicio. Describe como el cliente se puede conectar al servicio y el formato de los datos esperados:



Tecnologías de Servicios Web

Service Endpoints

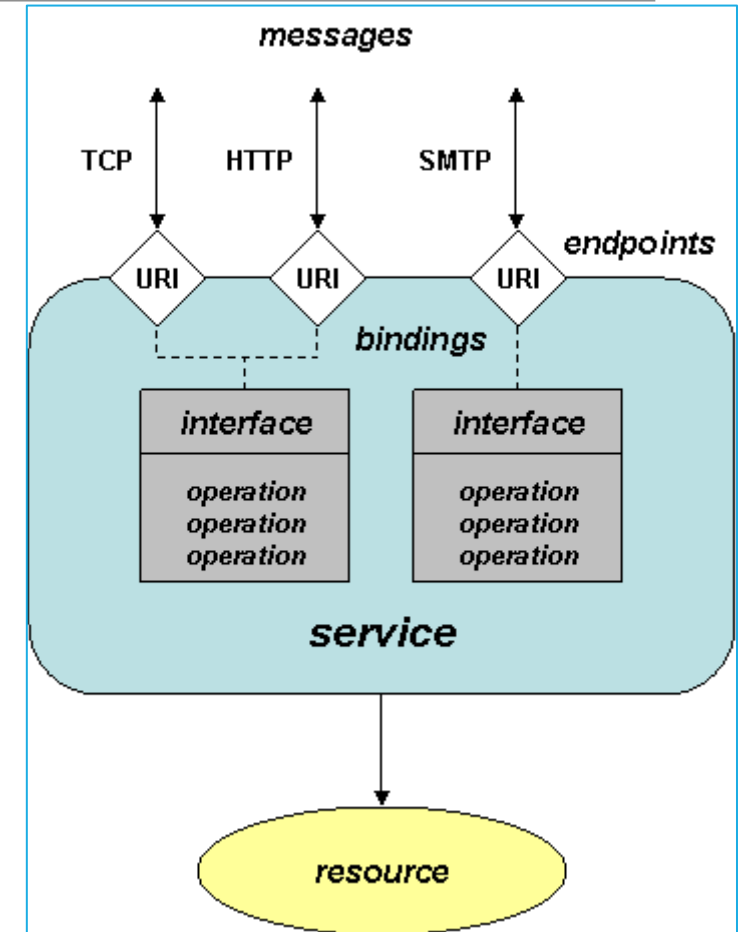
- a. Protocolo de transporte. HTTP, HTTPS, TCP, Named-Pipes (shared folders) y Message Queues (e-mail).
- b. Formato de codificación de mensajes. XML, TEXT, BINARIO, JSON.
- c. Requerimientos de seguridad. SSL, username/password, etc.
- d. La confiabilidad de los comunicaciones con el servicio. Las redes pueden fallar, por lo que el servicio debe asegurar la integridad de las conversaciones.



Tecnologías de Servicios Web

Service Endpoints

3. El contrato implementado por el servicio. Es una interface anotada con el atributo [*ServiceContract*]. Describe las operaciones implementadas por el servicio marcadas con el atributo [*OperationContract*]. El servicio debe describir la estructura de los datos compuestos y como deben ser serializados.



Introducción a Windows Communication Foundation

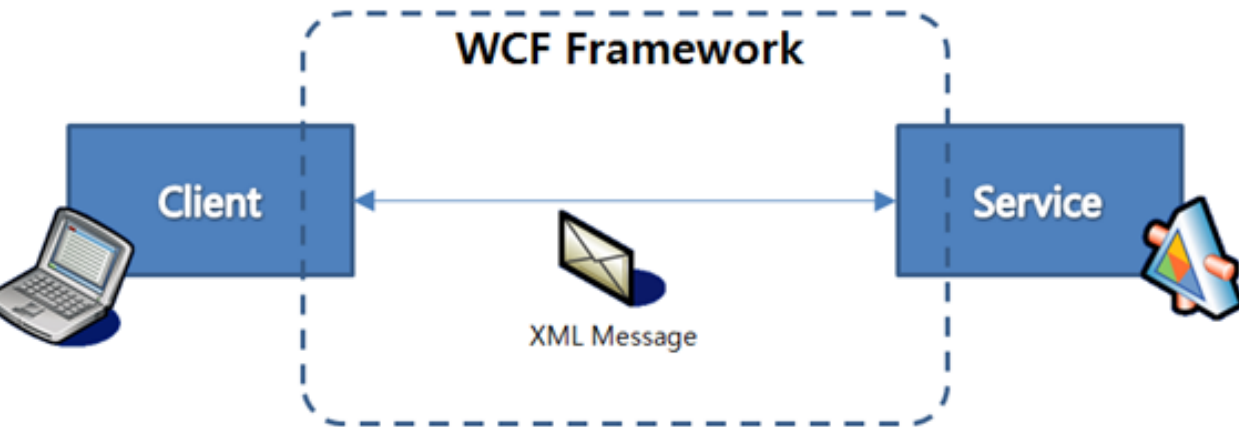
QUÉ ES

Tecnologías de Servicios Web

¿Qué es Windows Communication Foundation?

Es una plataforma de mensajería parte del .NET Framework que permite el desarrollo de aplicaciones basadas en **Arquitecturas Orientadas a Servicios (SOA)**.

- Provee un modelo unificado de programación para utilizar las tecnologías de comunicación como SOAP, Servicios Web, Mensajes XML, además de administrar los mecanismos de seguridad y privacidad en sus capas de bajo nivel.



.NET Framework 4.5

Windows Presentation Foundation Built-in Ribbon controls Databinding improvements Ability to add breakpoints to databindings Data source change aware views (Live Shaping) Validation improvements Improved legacy UI integration Dispatcher improvements Speed-up of large data sets	Windows 8 support Support for Windows Runtime (WinRT) .NET Profile for Metro-style apps Improved support for sharing DLLs between .NET profiles	ASP.NET Built-in providers now natively support SQL Azure Support for implementing WebSocket receivers Built-in JavaScript + CSS combining and minification Asynchronous pipeline support (Response, Request, HttpHandlers)	ASP.NET MVC 4 Async controllers Built-in mobile templates + JQuery Mobile support Alternate views (e.g. print version, mobile site) Support for Recipes: intelligent codegen	ASP.NET Web Pages 2 New site templates Versatile validation support Support for OAuth and OpenID Built-in map embedding tools; supports Google, Bing and others Did you know? ASP.NET Web Pages is a great mobile way to work! Try with Windows 8, Forms and MVC. WP sites use Razor and are specially developed with WebForms.	Web Forms Strongly typed data binding MVC-like support for Models HTML5 encoded binding expressions HTML5 support Control support for new semantic elements Validator and UpdatePanel now support new HTML5 elements	Tooling IntelliSense improvements More JavaScript support JS Expressions used by default New templates and snippets
Windows Communication Foundation Support for UDP multicast channel TCP channels now work with partial trust Asynchronous operations Streamin improvements Simplified configuration Can now generate service stubs from WSDL WebSocket support		Windows Workflow Foundation C# Expressions State machine workflows are back! Workflow versioning Code-first activity design Faster execution Designer usability improvements	Managed Extensibility Framework 2.0 Debugging improvements Support for explicit bindings between objects Support for binding POCO's: no more attribute requirements	ADO.NET Sparse columns support improved Passwords are now stored encrypted Asynchronous operations	Entity Framework 4.5 Enumeration support Migrations for schema changes Designer improvements Spatial data type support Table-valued function support Multi-result sprec support	Multiple diagrams per model Code-first support Auto-compiled LINQ queries
Base Class Library Usability improved for WeakReferences and Streams Key interfaces now have async versions New ArraySegment and ReadOnlyDictionary classes Support for CLR objects over 2 GB in size Resource file management performance improved Unicode support for console applications	Task Parallel Library Task thread controls improved: Task.WaitAll, WaitAny, various timeout primitives available TPL Dataflow: Tools for parallel data flow processing	C# 5.0 Support for async programming: async and await keywords Methods can access call site info as parameters (CallerInfo)	Visual Basic 11 Iterator implementations (Yield) Async ja Await equal to C# Global keyword for namespace handling Call Hierarchy view available	Visual C++ 11 C++11 standard support improved Auto-vectorization and parallelization of loops Support for controlling CPU-driven processing (C++ AMP)	Wait, there's more! Visual Studio "11" is also coming, including support for .NET 4.5, Windows 8 and more. There are plenty of other new goodies, too. Also, .NET team is working on Project Roslyn, which enables interesting scenarios for integrating the C#-VB compilers to your own app. Roslyn is in CTP, but will ship post-4.5.	

Alternativas a WCF

Tecnologías de Servicios Web

JAX-WS (Java API for XML Web Services)



GlassFish » Metro » JAX-WS



Es una interfaz de programación de aplicaciones (API) de Java en Extensible Markup Language (XML) para la creación de servicios web (WS). Es parte de la plataforma Java EE de Oracle

JAX-WS se desarrolla como un proyecto de código abierto y forma parte del proyecto GlassFish

```
package appclient;

import helloservice.endpoint.HelloService;
import javax.xml.ws.WebServiceRef;

public class HelloAppClient {
    @WebServiceRef(wsdlLocation =
        "META-INF/wsdl/localhost_8080/helloservice/HelloService.wsdl")
    private static HelloService service;

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println(sayHello("world"));
    }

    private static String sayHello(java.lang.String arg0) {
        helloservice.endpoint.Hello port = service.getHelloPort();
        return port.sayHello(arg0);
    }
}
```



```
package helloservice.endpoint;

import javax.jws.WebService;
import javax.jws.WebMethod;

@WebService
public class Hello {
    private String message = new String("Hello, ");

    public void Hello() {
    }

    @WebMethod
    public String sayHello(String name) {
        return message + name + ".";
    }
}
```





PHP Web Services NuSOAP

Es un kit de herramientas (ToolKit) para desarrollar Web Services bajo el lenguaje PHP.

Está compuesto por una serie de clases que pueden hacer mucho más fácil el desarrollo de Web Services.

Provee soporte para el desarrollo de clientes y de servidores. NuSOAP está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1

```

1 <?php
2
3 // incluimos la libreria ó toolkit nusoap que descargamos p
4 require_once('lib/nusoap.php');
5
6 ?>
    
```

2.- Creamos la instancia al servidor

```

1 <?php
2 // Vamos a crear la instancia del servidor.
3 $server = new nusoap_server();
4
5 ?>
    
```

3.- Inicializamos el soporte WSDL

```

1 <?php
2 $server->configureWSDL('Mi Web Service #1', 'urn:mi_ws1');
3
4 ?>
    
```

4.- Registramos la estructura de datos usado por el servicio

```

1 <?php
2 // Parametros de entrada
3 $server->wsdl->addComplexType( 'datos_persona_entrada',
4                               'complexType',
5                               'struct',
6                               'all',
7                               'all',
8                               array('nombre' => array('name' => 'nombre',
9                               'email' => array('name' => 'email',
10                              'telefono' => array('name' => 'telefono',
11                              'ano_nac' => array('name' => 'ano_nac'
    
```

5.- Registramos el método/función para que el Web Service lo reconozca y se pueda utilizar:

```

1 <?php
2 $server->register( 'calcula_edad', // nombre del metodo o funcion
3                  array('datos_persona_entrada' => 'tns:datos_persona_entrada'), //
4                  array('return' => 'tns:datos_persona_salidad'), // parametros de s
5                  'urn:mi_ws1', // namespace
6                  'urn:hellowsdl2#calcula_edad', // soapaction debe ir asociado al n
7                  'rpc', // style
8                  'encoded', // use
9                  'La siguiente funcion recibe los parametros de la persona y calcul
10
11 };>
    
```

```

<?php

require_once('lib/nusoap.php');

$cliente = new nusoap_client('http://www.mauricioalpizar.com/ejemplos/nusoap/mi_ws1.php');
//print_r($cliente);

$datos_persona_entrada = array( "datos_persona_entrada" => array(
    'nombre' => "Mauricio A.",
    'email' => "ealpizar@ticosoftweb.com",
    'telefono' => "8700-5455",
    'ano_nac' => 1980
));

$resultado = $cliente->call('calcula_edad',$datos_persona_entrada);

print_r($resultado);

?>
    
```