



## Tema 7

# Servicios Web

Carlos Montellano

A background map of the Americas, showing North and South America, is visible at the top of the slide. The title 'Introducción I' is centered over this map in a white serif font.

# Introducción I

- ❖ Los Servicios Web son la piedra angular de cualquier desarrollo de sistemas distribuidos actual
- ❖ Las computadoras hablan unos a otros a través de la web usando HTTP y otros protocolos.
- ❖ Un servicio web no tiene interfaz gráfica
- ❖ Provee una API de métodos que pueden ser invocados en la web
- ❖ Diseñados para proveer “servicios”

A faint, stylized map of Europe serves as the background for the top half of the slide. The map is in shades of blue and white, showing major landmasses and some geographical features like rivers and coastlines. It is partially obscured by a dark blue horizontal band that contains the title.

# Evolucion

- ❖ Los servicios web evolucionaron a partir de tecnologías anteriores que cumplían el mismo objetivo, como RPC, ORPC (DCOM, CORBA y JAVA RMI).
- ❖ Los servicios web están destinados a resolver tres problemas principales:
  1. Interoperabilidad
  2. Cortafuegos
  3. Complejidad

A faint, stylized map of South America is visible in the background of the slide, showing major countries and geographical features.

# Concepto de Servicio Web


- ❖ Un **Servicio Web** es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
- ❖ Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de computadoras como Internet.



# ¿Por qué utilizar los Servicios Web XML?

**Sitio de viajes  
Northwind Traders**

Seleccionar destino: **Redmond**

La previsión meteorológica es:  **Lluvia**

El tipo de cambio es: **\$1.56**

El billete de avión sólo cuesta: **\$1,999.98**



**Servicio Web XML  
meteorológico**

**Servicio Web XML  
tipo de cambio**

**Servicio Web XML  
precio del billete**

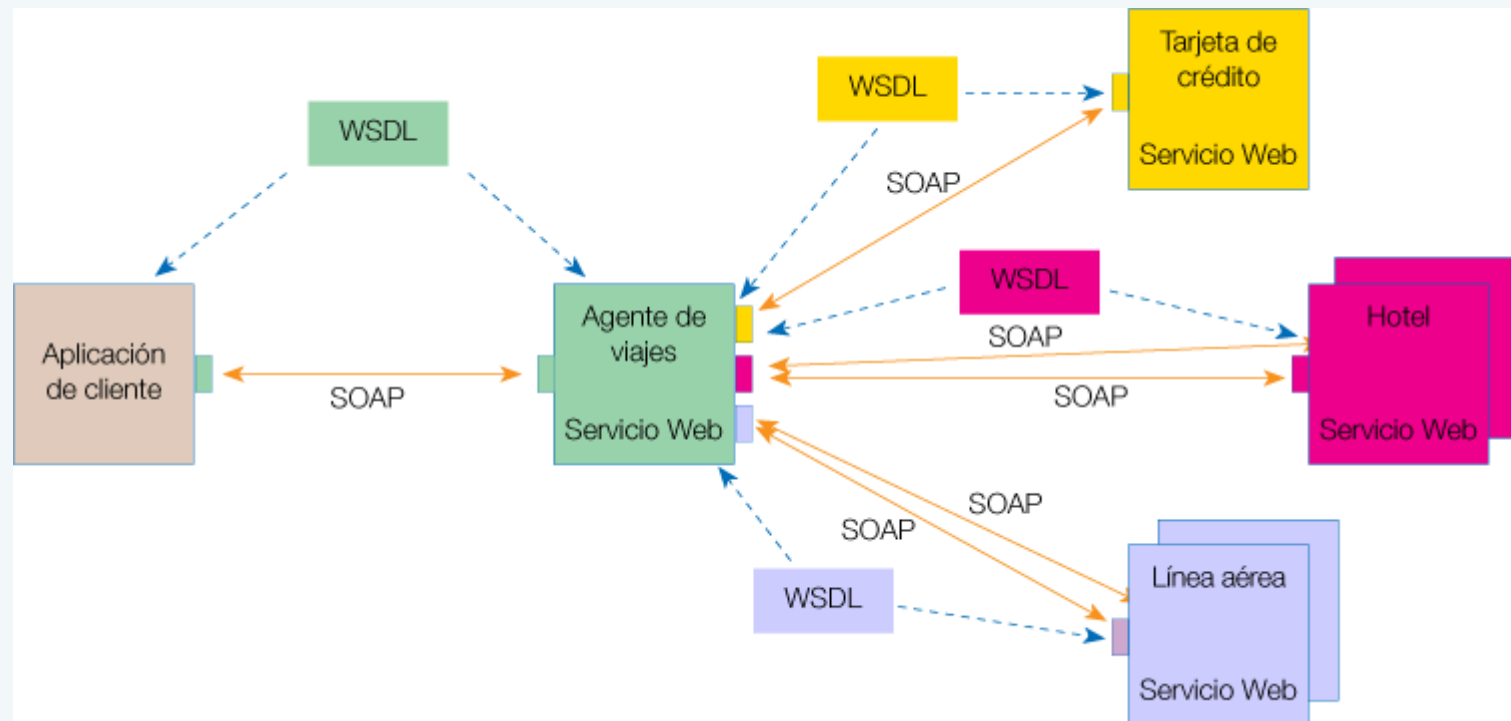
**Base de datos de  
precios de billetes**

A faint, stylized map of Europe serves as the background for the slide. The map is light blue and white, showing major landmasses and water bodies. It is positioned behind the title and the main text area.

# Interoperabilidad de los Servicios Web

- ❖ La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

# Interoperabilidad de los Servicios Web





# Servicio Web XML

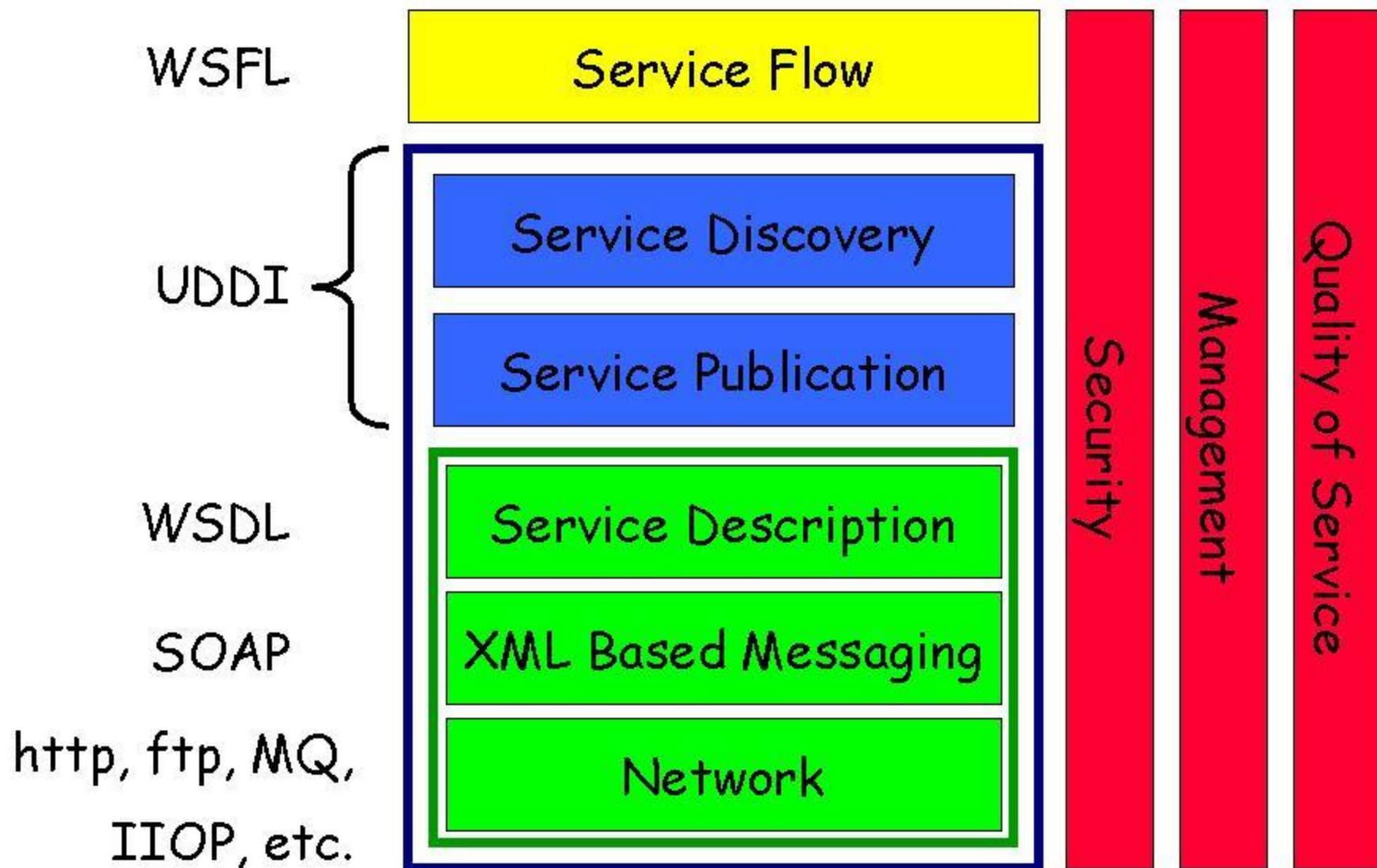
- ❖ Es un Servicio Web que usa XML para intercambiar datos entre el cliente del servicio y el servidor que lo proporciona.



# Estándares empleados en los Servicios Web

- ❖ [Web Services Protocol Stack](#): conjunto de servicios y protocolos de los servicios Web.
- ❖ [XML](#) (Extensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar.
- ❖ [SOAP](#) (Simple Object Access Protocol) o [XML-RPC](#) (XML Remote Producer Call): Protocolos sobre los que se establece el intercambio.
- ❖ Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como [HTTP](#) (Hypertext Transfer Protocol), [FTP](#) (File Transfer Protocol), o [SMTP](#) (Simple Mail Transfer Protocol).
- ❖ [WSDL](#) (Web Services Description Languages): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- ❖ [UDDI](#) (Universal Description, Discovery and Integration): Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones comprobar qué servicios web están disponibles.
- ❖ [WS-Security](#) (Web Service Security): Protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

# The Web Services Stack



A faint, stylized map of the United States is visible in the background, showing state boundaries and major cities. The map is rendered in a light blue and white color scheme, with the title text overlaid on a dark blue banner at the top.

# Protocolos para acceder a los Servicios Web

- ❖ HTTP-GET.
- ❖ HTTP-POST.
- ❖ SOAP.

A background map of Mexico is visible at the top of the slide, showing the country's borders and major cities. The title 'HTTP-GET' is overlaid on this map in a large, white, sans-serif font.

# HTTP-GET

- ❖ Forma en que normalmente se solicita una página Web, ejemplo:

<http://www.acme.com/inicio.aspx?id=David&equipo=mexico>

- ❖ ASP.NET puede obtener los valores con:
  - `Request.QueryString("id")`
  - `Request.QueryString("equipo")`

Cuando se usa el método “GET” para el envío de datos en un formulario.

# HTTP-POST

- ❖ En vez de adjuntar los parámetros al URL, se colocan en el mensaje de petición HTTP. Es la manera común que usan los formularios HTML.

```
<form method="post">
```

```
  <input type="Text" id="id">
```

```
  ...
```

```
</form>
```

- ❖ ASP.NET los puede recuperar con:

```
Request.Form("id")
```





# SOAP

- ❖ Basado en XML para transmitir información.
- ❖ Además de soportar pares nombre/valor, soporta datos complejos como datos enriquecidos, clases y objetos.
- ❖ SOAP también transmite sobre HTTP pero no está limitado al modelo de petición/respuesta.

# SOAP

## Ejemplos de mensajes SOAP

Como ejemplo se muestra la forma en que un cliente solicitaría información de un producto a un proveedor de servicios Web:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productId>827635</productId>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

# SOAP

Y esta sería la respuesta del proveedor:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productId>827635</productId>
        <description>3-Piece luggage set.  Black Polyester.</description>
        <price>96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```





# Ventajas de los servicios Web

- ❖ Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- ❖ Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- ❖ Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- ❖ Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.



# Inconvenientes de los Servicios Web

- ❖ Para realizar [transacciones](#) no pueden compararse en su grado de desarrollo con los estándares abiertos de [computación distribuida](#) como [CORBA](#) (Common Object Request Broker Architecture).
- ❖ Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como [RMI](#) (Remote Method Invocation), [CORBA](#), o [DCOM](#) (Distributed Component Object Model). Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de [XML](#) no se encuentra la concisión ni la eficacia de procesamiento.
- ❖ Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en [firewall](#) cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.
- ❖ Existe poca información de Servicios Web para algunos lenguajes de programación.





# Razones para crear Servicios Web

- ❖ Se basan en [HTTP](#) sobre [TCP](#) (Transmission Control Protocol) en el [puerto](#) 80.
- ❖ Antes de que existiera [SOAP](#), no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red.
- ❖ Pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio.

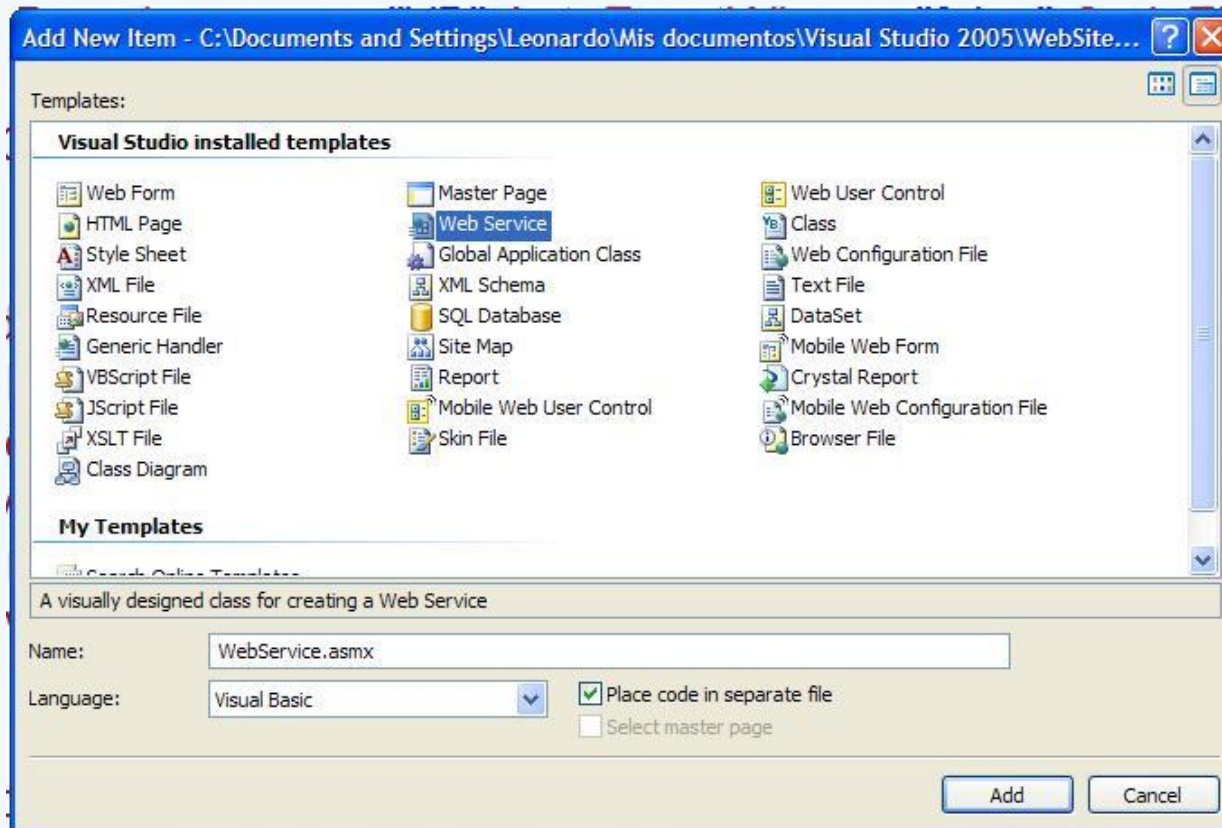


# Servidores de aplicaciones para Servicios Web

- ❖ [Axis](#) y el servidor [Jakarta Tomcat](#) (de Apache)
- ❖ [ColdFusion](#) MX de [Macromedia](#)
- ❖ [Java Web Services Development Pack](#) (JWS DP) de [Sun Microsystems](#) (basado en [Jakarta Tomcat](#))
- ❖ [JOnAS](#) (parte de *ObjectWeb* una iniciativa de código abierto)
- ❖ [Microsoft .NET](#)
- ❖ [Novell exteNd](#) (basado en la plataforma [J2EE](#))
- ❖ [WebLogic](#)
- ❖ [WebSphere](#)
- ❖ [Zope](#) es un servidor de aplicaciones Web [orientado a objetos](#) desarrollado en el [lenguaje de programación Python](#)
- ❖ [VERASTREAM](#) de AttachmateWRQ para modernizar o integrar aplicaciones host IBM y VT
- ❖ [Mono](#)

# Crear un Servicio Web

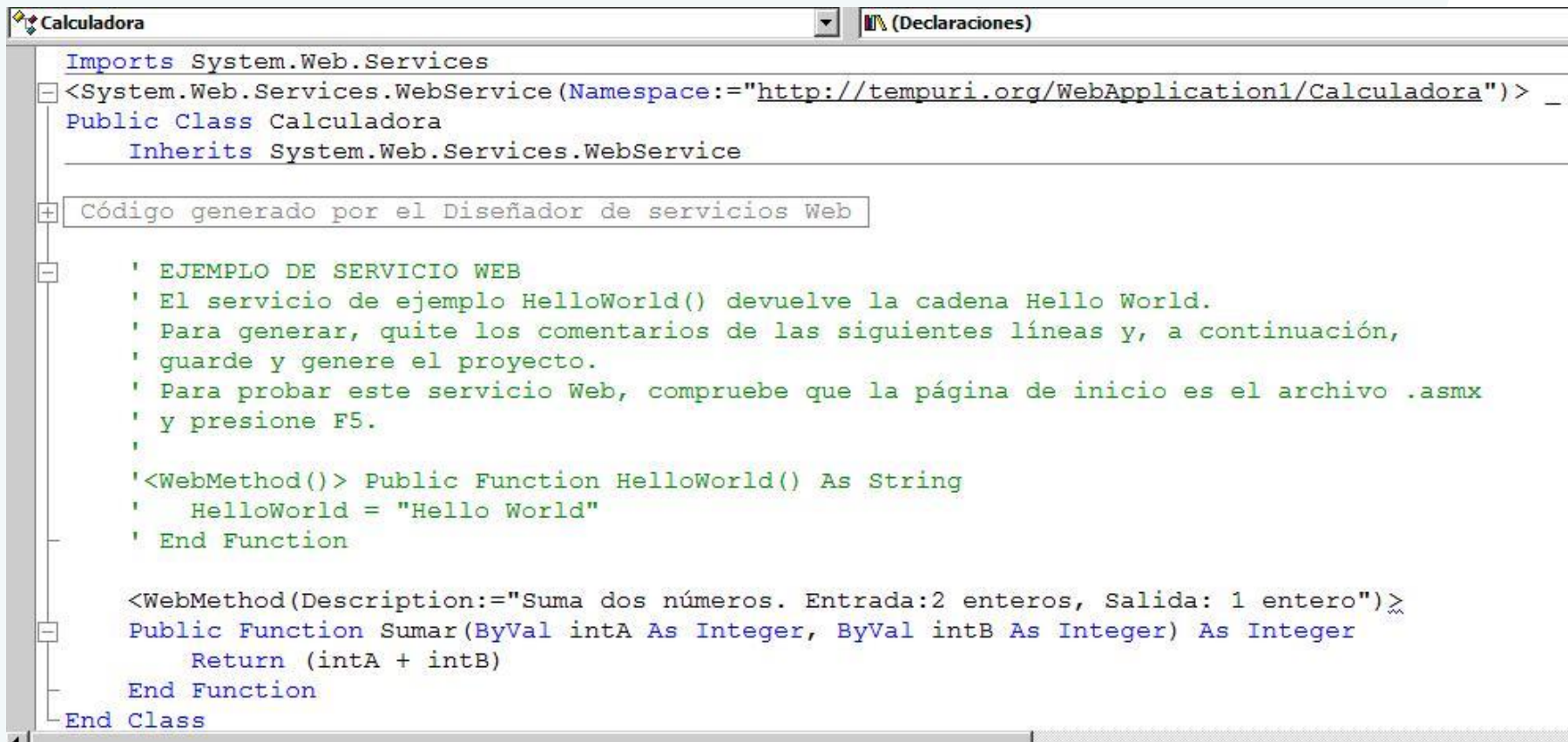
- ❖ 1. Agregar un nuevo elemento tipo “Servicio Web” al proyecto Web. Nombrar al proyecto: Calculadora.asmx





# Crear un Servicio Web

## ❖ 2. Escribir el código que define la funcionalidad del servicio.



```
Imports System.Web.Services

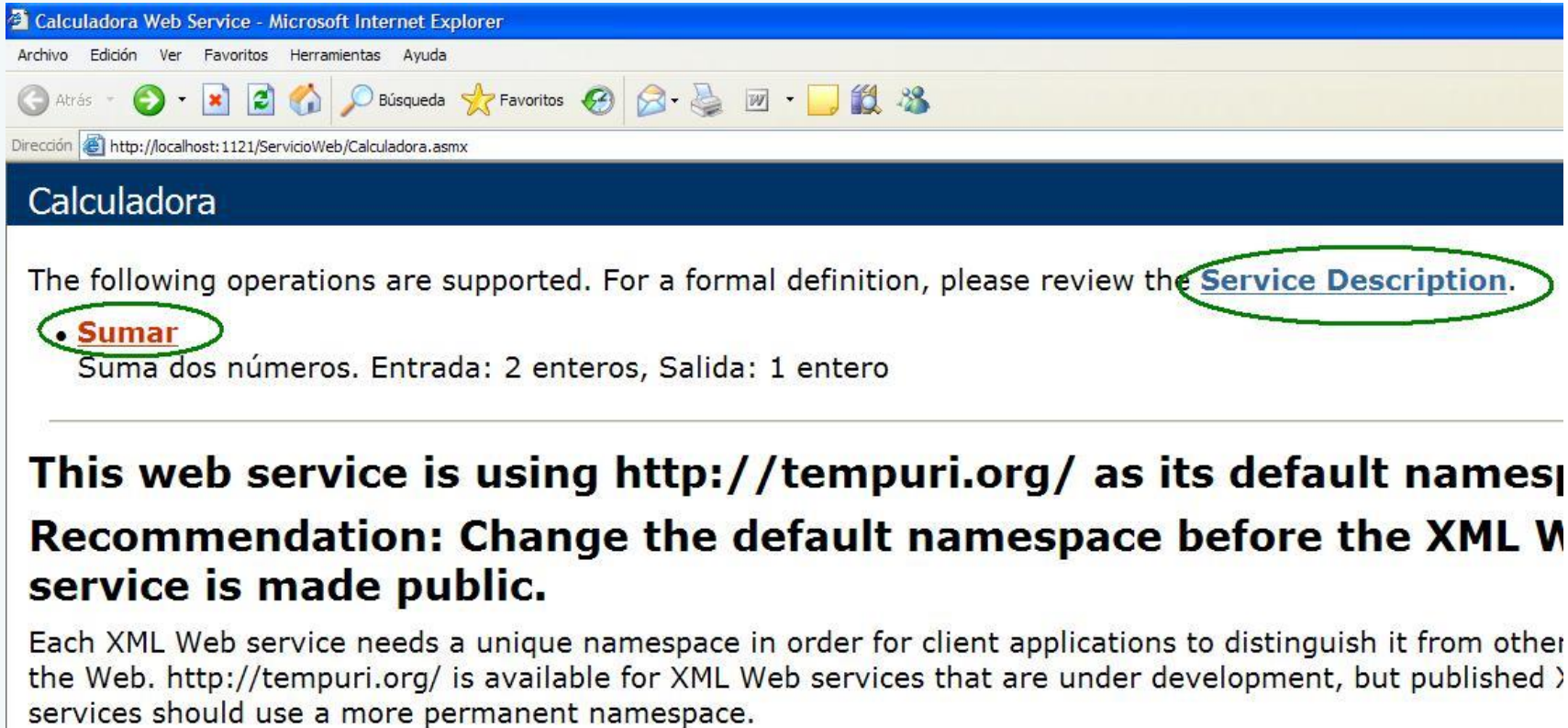
<System.Web.Services.WebService(Namespace:="http://tempuri.org/WebApplication1/Calculadora")> _
Public Class Calculadora
    Inherits System.Web.Services.WebService

    ' EJEMPLO DE SERVICIO WEB
    ' El servicio de ejemplo HelloWorld() devuelve la cadena Hello World.
    ' Para generar, quite los comentarios de las siguientes líneas y, a continuación,
    ' guarde y genere el proyecto.
    ' Para probar este servicio Web, compruebe que la página de inicio es el archivo .asmx
    ' y presione F5.
    '
    ' <WebMethod()> Public Function HelloWorld() As String
    '     HelloWorld = "Hello World"
    ' End Function

    <WebMethod(Description:="Suma dos números. Entrada:2 enteros, Salida: 1 entero")>
    Public Function Sumar(ByVal intA As Integer, ByVal intB As Integer) As Integer
        Return (intA + intB)
    End Function
End Class
```

# Crear un Servicio Web

## 3. “Ejecutar” archivo “.asmx” en el explorador.



The screenshot shows a Microsoft Internet Explorer window with the title 'Calculadora Web Service - Microsoft Internet Explorer'. The address bar displays 'http://localhost:1121/ServicioWeb/Calculadora.asmx'. The page content includes a header 'Calculadora' and a paragraph stating 'The following operations are supported. For a formal definition, please review the [Service Description](#).' Below this, a bulleted list contains the item '• **Sumar**', which is circled in green. Underneath the list item, it says 'Suma dos números. Entrada: 2 enteros, Salida: 1 entero'. A horizontal line separates this from a bold warning: 'This web service is using http://tempuri.org/ as its default namespace. Recommendation: Change the default namespace before the XML Web service is made public.' Below the warning, a paragraph explains that each XML Web service needs a unique namespace and that http://tempuri.org/ is only for development.

Calculadora

The following operations are supported. For a formal definition, please review the [Service Description](#).

- **Sumar**  
Suma dos números. Entrada: 2 enteros, Salida: 1 entero

---

**This web service is using http://tempuri.org/ as its default namespace.  
Recommendation: Change the default namespace before the XML Web service is made public.**

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other the Web. http://tempuri.org/ is available for XML Web services that are under development, but published ) services should use a more permanent namespace.

Nota: observar las opciones para ver la descripción de servicios (Service Description) y probar la función del Servicio Web (Sumar).



- ❖ Al hacer clic en el vínculo “Sumar” se puede ejecutar y probar el Servicio Web.

Calculadora Web Service - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección <http://localhost:1121/ServicioWeb/Calculadora.asmx?op=Sumar>

## Calculadora

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

### Sumar

Suma dos números. Entrada: 2 enteros, Salida: 1 entero

#### Test

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

Parameter	Value
intA:	<input type="text" value="45"/>
intB:	<input type="text" value="11"/>

#### SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

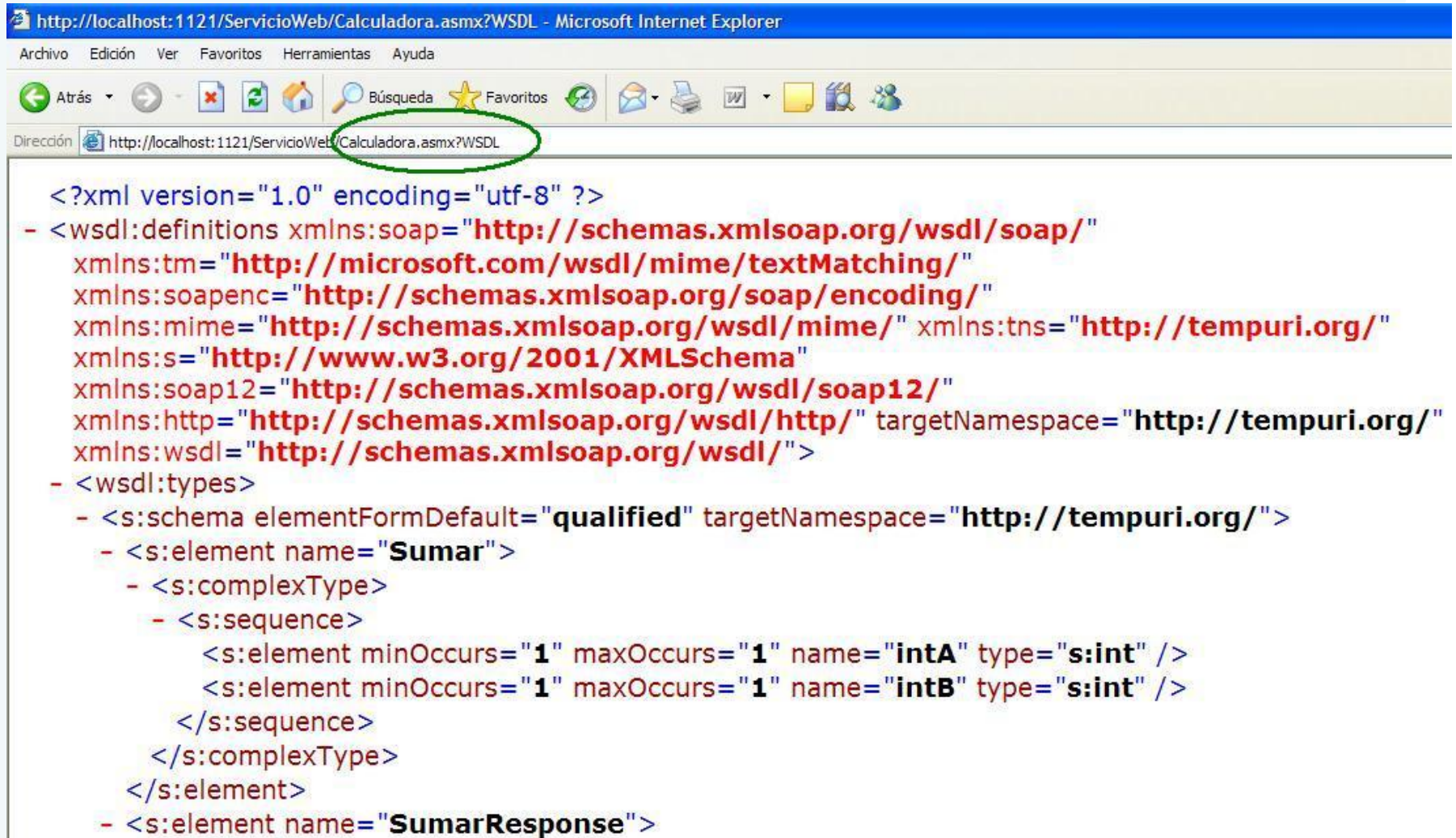
```
POST /ServicioWeb/Calculadora.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Sumar"
```

1. Escribir valores e invocar función.

2. Resultado de la función.

`<?xml version="1.0" encoding="utf-8" ?>`  
`<int xmlns="http://tempuri.org/">56</int>`

Al hacer un clic en la opción “Service Description” se muestra el archivo XML que escribe el servicio (protocolos, parámetros, etc.)



http://localhost:1121/ServicioWeb/Calculadora.asmx?WSDL - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://localhost:1121/ServicioWeb/Calculadora.asmx?WSDL

```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://tempuri.org/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
- <wsdl:types>
  - <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    - <s:element name="Sumar">
      - <s:complexType>
        - <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="intA" type="s:int" />
          <s:element minOccurs="1" maxOccurs="1" name="intB" type="s:int" />
        </s:sequence>
      </s:complexType>
    </s:element>
    - <s:element name="SumarResponse">
```



A faint, stylized map of the Iberian Peninsula (Spain and Portugal) serves as the background for the slide. It shows major cities, rivers, and geographical features in a light blue and grey color scheme.

## 6.4 Consumo de un Servicio Web.

- ❖ El consumo consiste en realizar las tareas necesarias para acceder a un Servicio Web y usar su funcionalidad en una aplicación.



# Usar un Servicio Web

1. Identificar o descubrir los Servicios Web existentes en un servidor.

Herramienta: disco.exe

Ejemplo:

```
C:\> disco http://www.acme.com /username:xxxx /password: yyyy
```

Nota:

- La herramienta se ejecuta en la consola de Visual Studio.NET. Tienen que tenerse permisos en el servidor.
- Opcional si se conoce la ubicación del Servicio Web.
- Ayuda: disco /?



## Usar un Servicio Web

2. Crear una clase Proxy.

Esta sirve como intermediaria entre el Servicio Web y el consumidor. Se puede generar con la herramienta: `wsdl.exe`.

Ejemplo:

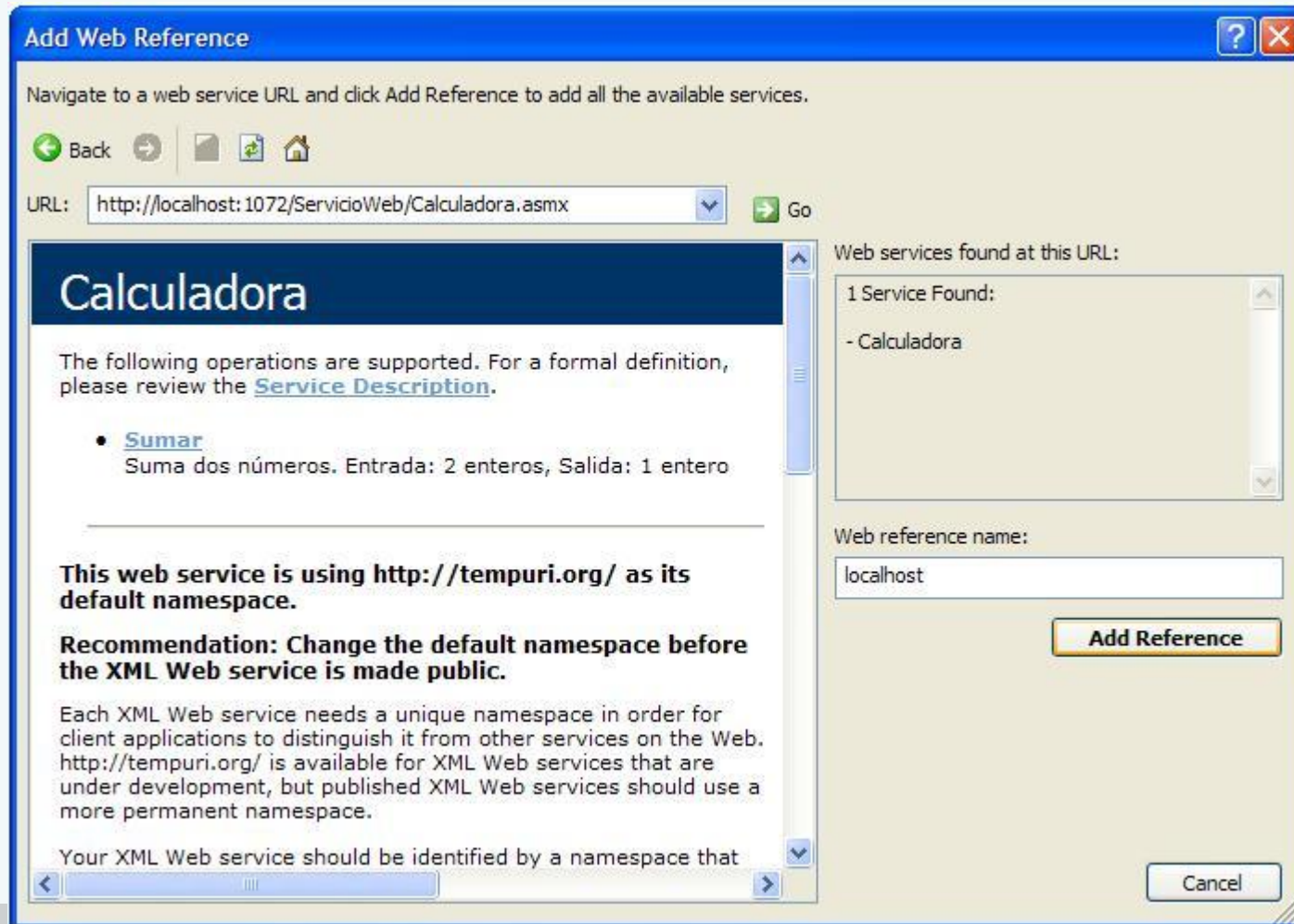
`wsdl /language:VB http://localhost/WebAPP1/calculadora.asmx?WSDL`

Nota: observar en el proyecto el archivo `Calculadora.vb` generado.



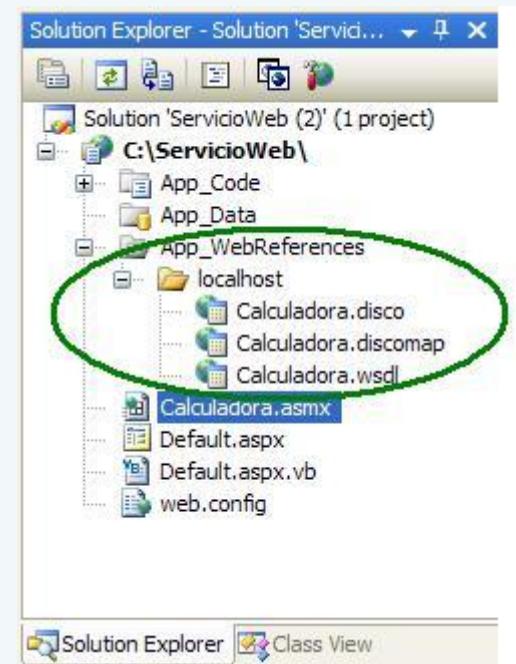
# Usar un Servicio Web

- ❖ Otra manera de crear la clase “proxy” es agregando una referencia Web en el proyecto que usará el servicio. Para esto seleccionar el proyecto y la opción “Agregar referencia Web”, buscar el servicio y seleccionarlo. Como se muestra:



# Nuevos archivos.

- ❖ Al hacer lo anterior se agregan al proyecto que usará el servicio Web, los siguientes archivos:



- ❖ Con lo anterior, el cliente del Servicio Web, puede usarlo. Se crea un objeto de la clase y se invocan los métodos.
- ❖ El Servicio Web se usará en una página .aspx con el siguiente diseño (todos los controles son del lado del servidor).

## Se usa un Servicio Web.

El Servicio Web es "Calculadora" y la función es "sumar".

Número 1:	<input type="text"/>	<input del="" servicio="" sumar"="" type="button" value="Invocar el método " web"=""/>
Número 2:	<input type="text"/>	
Resultado:	<input type="text"/>	



❖ El código del lado del servidor para el botón es:

```
Protected Sub Button1_Click(ByVal sender As Object,  
    Dim cal As Calculadora  
    cal = New Calculadora  
    Dim num1, num2, resul As Integer  
    num1 = Integer.Parse(TextBox1.Text)  
    num2 = Integer.Parse(TextBox2.Text)  
    resul = cal.Sumar(num1, num2)  
    TextBox3.Text = resul  
  
End Sub
```

Ejecutar la página y probar el funcionamiento.





{ REST }

A background image of a map of Europe, showing various countries and cities. The map is in a light blue and white color scheme. The title 'API' is centered over the map.

# API

Las API son la clave para crear nuevos canales digitales. Facilitan el intercambio de datos internos y externos, permiten aplicaciones móviles y crean nuevas oportunidades de monetización.

API es el acrónimo de Application Programming Interface, que es un intermediario de software que permite que dos aplicaciones se comuniquen entre sí. Cada vez que utiliza una aplicación como Facebook, envía un mensaje instantáneo o verifica el clima en su teléfono, está usando una API. API: el pegamento que lo une todo ... Las API son un habilitador central de nuevos modelos de negocios

*Mulesoft*

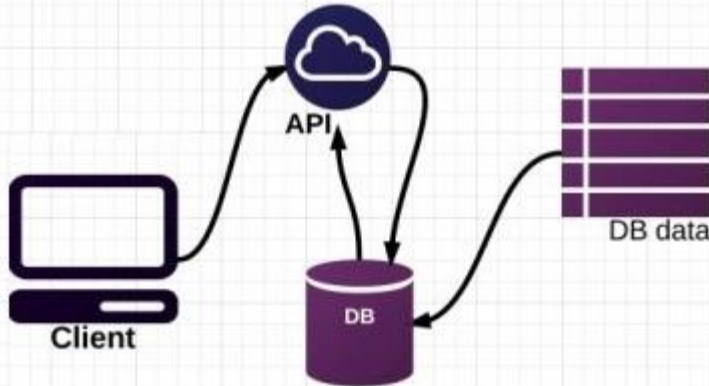
APIs como una forma de servir a sus clientes

*freecodecamp*

Video (3 min): <https://youtu.be/s7wmiS2mSXY>

# API

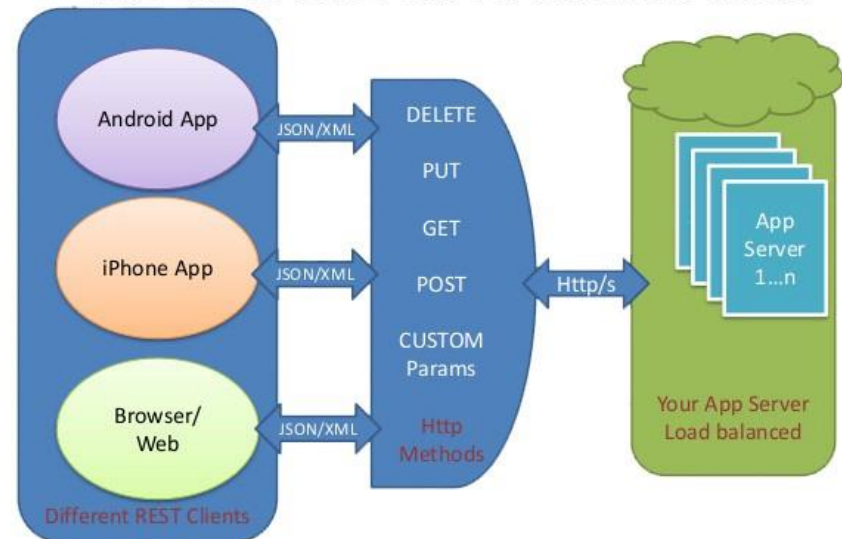
## Web Services



Una API es simplemente una interfaz para una aplicación de TI o fuente de datos que permite que las aplicaciones o máquinas autorizadas accedan fácilmente a ella. Una API bien definida es el pegamento que une los datos con la organización del futuro.



## REST API Architecture



# API y SDK

Una API es simplemente una interfaz que permite que el software interactúe con otro software.

Esto es parte de su nombre, API, interfaz de programación de aplicaciones, y es fundamental para su funcionalidad.

Las API web permiten la interacción entre sistemas dispares, a menudo para casos de uso específicos en la web.

SDK significa "Kit de desarrollo de software", que es una gran manera de pensar en ello: un kit.

Piensa en armar un auto modelo o un avión.

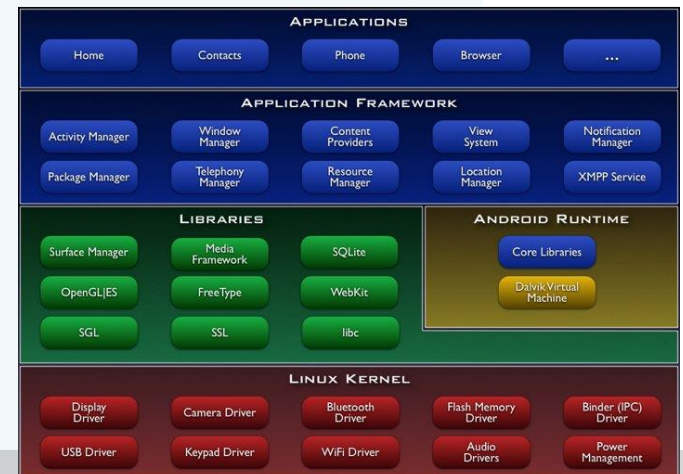
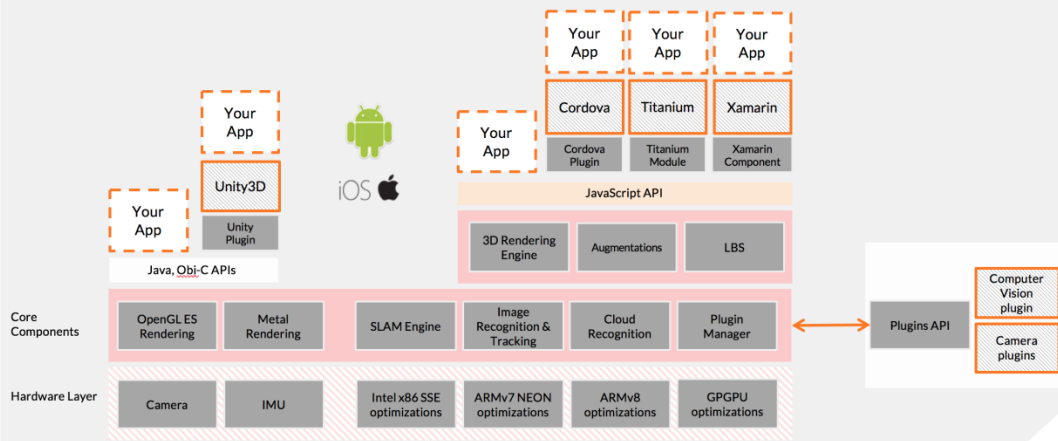
Al construir este modelo, se necesita un conjunto completo de elementos, incluidas las piezas del kit, las herramientas necesarias para armarlos, las instrucciones de montaje, etc.





# API y SDK

## Wikitude SDK Architecture



# API y SDK

*Ejemplo:*

*Facebook API (comunicacion con FB):*

```
/me?  
fields=albums.limit(5){name, photos.limit(2){name, picture, tags.limit(2)}},posts.limit(5)
```

*Facebook SDK (desarrollar aplicaciones):*

```
1 // AppDelegate.m  
2 #import <FBSDKCoreKit/FBSDKCoreKit.h>  
3 - (void)applicationDidBecomeActive:(UIApplication *)application {  
4     [FBSDKAppEvents activateApp];  
5 }
```

El SDK es el componente básico de la aplicación, mientras que el API es el idioma de sus solicitudes

SDK generalmente API; ninguna API contiene SDK.

Los SDK permiten la creación de aplicaciones, ya que una fundación permite la creación de una casa;

Las API permiten el funcionamiento de las aplicaciones dentro de los parámetros definidos del SDK, como las líneas telefónicas de una casa.





# API

API significa 'Interfaz de programación de aplicaciones'.

Técnicamente, una API describe cómo conectar un conjunto de datos o un proceso comercial con algún tipo de aplicación para el consumidor u otro proceso comercial. Las API permiten que una empresa se convierta en una plataforma.

Las API proporcionan una forma para que las empresas aprovechen nuevos mercados.





# API

DESIGNED BY  
Mehdi Medjaoui

## The API Landscape

Last Update: March, 2017

POWERED BY  
spoke

### Business Processes as an API/API-as a Product/Transactional APIs (151)



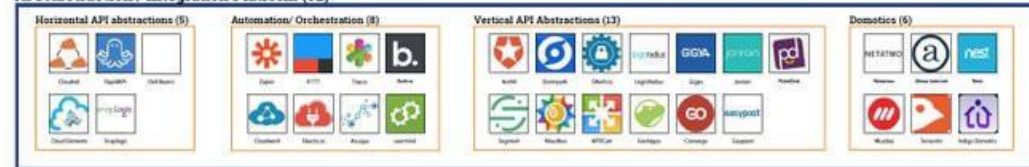
### API Lifecycle platform (78)



### Backend Building Tools/MBaaS (14)



### API Abstraction / Integration Platform (32)

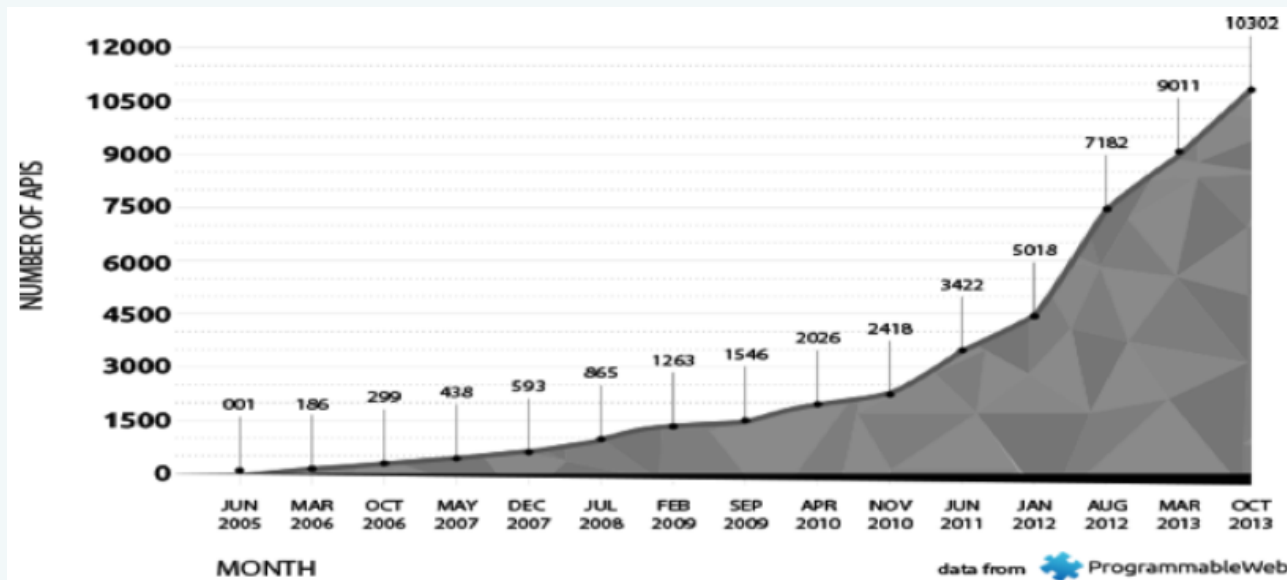




# WEB API

Una API web, también conocida como Servicio web, proporciona una interfaz para aplicaciones web o aplicaciones que necesitan conectarse entre sí a través de Internet para comunicarse.

Hasta 2013, aparecieron 13000 API públicas.





# WEB API - SOAP

SOAP fue diseñado en 1998 por Dave Winer, Don Box, Bob Atkinson y Mohsen Al-Ghosein para Microsoft Corporation.

Fue diseñado para ofrecer un nuevo protocolo y un marco de mensajería para la comunicación de aplicaciones a través de la web.

Si bien SOAP se puede usar en diferentes protocolos, requiere que un cliente SOAP cree y reciba las diferentes solicitudes, y depende en gran medida del Lenguaje de definición de servicios web (WSDL) y XML.

# WEB API - SOAP

## SOAP Request:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap--envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap---
encoding">
  <soap:Body xmlns:m="http://www.example.com/weather">
    <m:GetWeather>
      <m:ZipCode>94108</m:ZipCode>
    </m:GetWeather>
  </soap:Body>
</soap:Envelope>
```

# WEB API – XML|JSON-RPC

Llamadas de procedimiento remoto, o API de RPC, son mucho más rápidas y fáciles de implementar que SOAP.

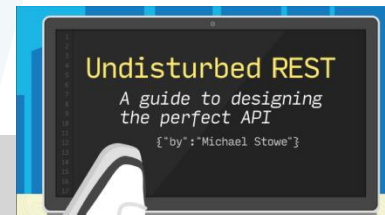
XML-RPC fue la base de SOAP, aunque muchos continuaron usándolo en su forma más genérica, realizando llamadas simples a través de HTTP con los datos formateados como XML.

Las llamadas RPC están estrechamente conectadas y requieren que el usuario no solo conozca el nombre del procedimiento, sino también el orden de los parámetros. Introducido en 2002, la notación de objetos JavaScript (JSON) fue desarrollada por State Software, Inc. y más conocida por Douglas Crawford. El formato se diseñó originalmente para aprovechar la capacidad de JavaScript para actuar como un sistema de mensajería entre el cliente y el navegador (piense en AJAX), JSON-RPC.

```
{ "zipCode": "94108" }
```



*Undisturbed REST*





# WEB API – REST

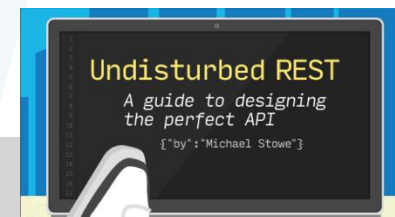
Ahora, la opción más popular para el desarrollo de API, REST o RESTful APIs se diseñó para aprovechar los protocolos existentes.

Si bien REST se puede utilizar en casi cualquier protocolo, generalmente aprovecha HTTP cuando se usa para API web. Esto significa que los desarrolladores no necesitan instalar bibliotecas o software adicional para aprovechar una API REST.

Según lo definido por el Dr. Roy Fielding en su disertación de doctorado de 2000, REST también proporciona una increíble capa de flexibilidad. Dado que los datos no están vinculados a métodos y recursos, REST tiene la capacidad de manejar múltiples tipos de llamadas, devolver diferentes formatos de datos e incluso cambiar estructuralmente con la implementación correcta de hipermedia.



*Undisturbed REST*

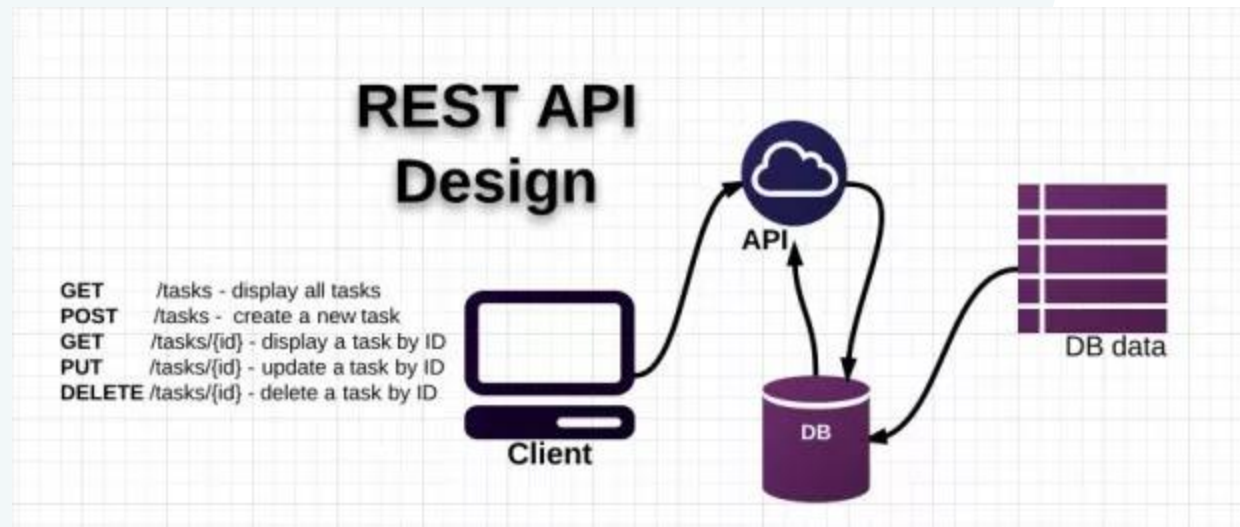


# WEB API – XML|JSON-RPC

SOAP	RPC	REST
Requires a SOAP library on the end of the client	Tightly coupled	No library support needed, typically used over HTTP
Not strongly supported by all languages	Can return back any format, although usually tightly coupled to the RPC type (ie JSON-RPC)	Returns data without exposing methods
Exposes operations/ method calls	Requires user to know procedure names	Supports any content-type (XML and JSON used primarily)
Larger packets of data, XML format required	Specific parameters and order	Single resource for multiple actions
All calls sent through POST	Requires a separate URI/ resource for each action/ method.	Typically uses explicit HTTP Action Verbs (CRUD)
Can be stateless or stateful	Typically utilizes just GET/ POST	Documentation can be supplemented with hypermedia
WSDL - Web Service Definitions	Requires extensive documentation	Stateless
Most difficult for developers to use.	Stateless	More difficult for developers to use.
	Easy for developers to get started	

# WEB API – REST

REST es un enfoque de las comunicaciones que a menudo se utiliza en el desarrollo de servicios web. El uso de REST a menudo se prefiere sobre el estilo SOAP más pesado (Protocolo simple de acceso a objetos) porque REST no aprovecha todo el ancho de banda, lo que lo hace más adecuado para su uso en Internet.





A faint, stylized map of Europe serves as the background for the top half of the slide. The map is in shades of blue and white, showing major landmasses and some internal borders. It is partially obscured by a dark blue horizontal band that contains the title.

# WEB API – REST

REST => Transferencia de estado representacional

Definido como un estilo arquitectónico para sistemas hipermedia distribuidos.

Resource-Orientation => cada cosa interesante que maneja su aplicación debe estar expuesta como un recurso: cualquier cosa que un cliente desee vincular.

- Cada recurso debe tener al menos un nombre.
- Un cliente no puede acceder a los recursos directamente.
- Un servicio web sirve representaciones de un recurso: documentos en formatos de datos específicos que contienen información sobre el recurso.
- La distinción entre un recurso y su representación adquiere mayor importancia cuando el recurso es una fila en una base de datos, un objeto físico, un concepto abstracto o un evento real en progreso.

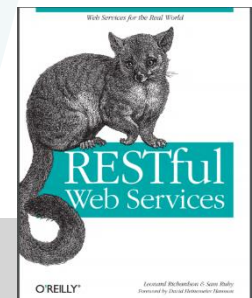
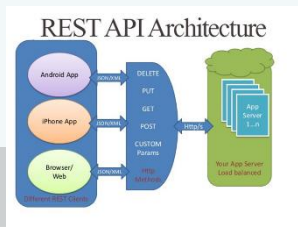
**¡REST es un estilo arquitectónico y no un estándar estricto!**



# WEB API – REST

## Tipos de recursos:

- Recursos únicos predefinidos, algo que solo tiene unos pocos: tal vez una clase en un sistema orientado a objetos, o una tabla de base de datos en un sistema orientado a bases de datos.
- Un gran número (posiblemente infinito) de recursos correspondientes a elementos de datos individuales: un objeto en un sistema orientado a objetos o una fila de base de datos en un sistema orientado a bases de datos.
- Un gran número (probablemente infinito) de recursos correspondientes a las posibles salidas de un algoritmo: los resultados de una consulta en un sistema orientado a la base de datos. Las listas de resultados de búsqueda y las listas filtradas de recursos se incluyen en esta categoría



# WEB API – REST

REST  
(Representational  
State Transfer)

Roy Fielding  
(2000)  
Architecture for  
the web



Hay seis restricciones clave para REST que debe tener en cuenta al decidir si este es el tipo de API correcto para usted:

- Interfaz uniforme
  - Existe una interfaz entre clientes y servidores.
- Sin Estado
  - El estado de la sesión se mantiene completamente en el cliente.
- Cacheable
  - Los clientes pueden almacenar respuestas en caché.
- Cliente-Servidor
  - Los clientes están separados de los servidores
- Sistema de capas
  - Un cliente normalmente no puede decir si está conectado directamente al servidor final, o a un intermediario en el camino



- Código bajo demanda (opcional)
  - Los servidores pueden ampliar o personalizar temporalmente la funcionalidad de un cliente transfiriéndole lógica que puede ejecutar (¿recuerda SOFEA?)

A faint, stylized world map is visible in the background of the slide, showing continents and major cities. The map is rendered in a light blue and white color scheme, with a darker blue overlay at the top where the title is located.

# REST

- ❖ REST se define como una interfaz entre sistemas que usan el protocolo HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.
- ❖ Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como REST.



A background image showing a map of the Americas, with North and South America visible. The map is in a light blue and white color scheme. Overlaid on the map is a dark blue horizontal band containing the word "REST" in white capital letters.

# REST

**Cliente:** Es la persona o el software que utiliza la API. Podría ser:

- 🔗 Un desarrollador que consumirá alguna API de otro programa para interactuar.
- 🔗 Otro software
- 🔗 Un navegador web.

**Recurso:** Es cualquier objeto sobre el que la API puede proporcionar información. Por ejemplo, en la API de Instagram, un recurso puede ser un usuario, una foto, un hashtag.

Cada recurso tiene un identificador único. El identificador puede ser un nombre o un número.



# Metodos Request

## GET

- Se utiliza para recuperar información del servidor dado usando un URI dado. Las solicitudes que utilizan GET solo deberían recuperar datos y no deberían tener otro efecto sobre los datos

## POST

- Se utiliza para enviar datos al servidor, por ejemplo, información del cliente, carga de archivos, etc. mediante formularios HTML

## PUT

- Reemplace todas las representaciones actuales del recurso de destino con el contenido cargado

## DELETE

- Elimine todas las representaciones actuales del recurso de destino proporcionadas por URI.

## PATCH

- Aplica modificaciones parciales a un recurso.



# Status code

1

- 1xx: informativo
- Esto significa solicitud recibida y proceso continuo.

2

- 2xx: éxito
- Esto significa que la acción se recibió, entendió y aceptó con éxito.

3

- 3xx: redirección
- Esto significa que se deben tomar medidas adicionales para completar la solicitud.

4

- 4xx: error del cliente
- Esto significa que la solicitud contiene una sintaxis incorrecta o no se puede cumplir

5

- 5xx: error del servidor
- El servidor no pudo cumplir una solicitud aparentemente válida

A background image of a map of Europe, showing various countries and geographical features. The map is in shades of blue and green, with white lines indicating borders and major cities. The word "Headers" is overlaid on the map in a large, white, sans-serif font.

# Headers

General Header: estos campos de encabezado tienen aplicabilidad general para los mensajes de solicitud y respuesta.

Request Header: estos campos de encabezado son aplicables solo para mensajes de solicitud.

Server Header: estos campos de encabezado son aplicables solo para mensajes de respuesta.

Entity Header: estos campos de encabezado definen Metadatos sobre el cuerpo de la entidad o, si no hay ningún cuerpo presente



The background of the slide features a stylized, light blue map of Europe, showing major landmasses and surrounding waters. The map is partially obscured by a dark blue horizontal band at the top and a light blue diagonal band on the right side.

# General Header

- 🔗 Cache-control
- 🔗 Connection
- 🔗 Date
- 🔗 Pragma
- 🔗 Trailer
- 🔗 Transfer-Encoding
- 🔗 Upgrade
- 🔗 Via
- 🔗 Warning

A background image of a map, possibly of Europe, with a blue overlay at the top.

# Client Request Headers

- Accept-Charset
- Accept-Encoding
- Accept-Language
- Authorization
- Expect
- From
- Host
- If-Match
- If-Modified-Since

- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Proxy-Authorization
- Range
- Referer
- TE
- User-Agent

# Entity Header

- ⌘ Allow
- ⌘ Content-Encoding
- ⌘ Content-Language
- ⌘ Content-Length
- ⌘ Content-Location
- ⌘ Content-MD5
- ⌘ Content-Range
- ⌘ Content-Type
- ⌘ Expires
- ⌘ Last-Modified

The background of the slide features a topographic map of the Pacific Northwest, showing the coastline of Washington and Oregon, with major cities like Seattle, Portland, and Vancouver visible. The map is rendered in a light blue and green color scheme, with a darker blue overlay at the top where the title is located.

# SOAP Vs REST

## SOAP

- Publica un mensaje en una URL
- Utiliza respuestas XML
- Acciones definidas por un nombre en el WSDL
- Solo una URL para el servicio

## REST

- Puede GET (obtener) recursos, POST mensajes, PUT recursos, DELETE recursos
- Puede usar respuestas XML, JSON
- Acciones definidas por la URL y el protocolo: Multiples URL