



Sistemas Distribuidos

Módulo 10

Servicios Web



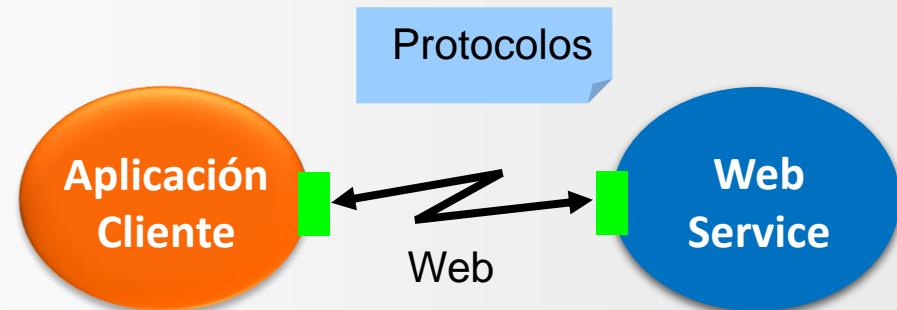
Agenda

- Servicios Web: Estructura y Funcionalidades
- Protocolo de Descubrimiento UDDI
- Protocolo de Comunicación: Soap y Rest
- Lenguaje Descriptor de Servicios WSDL

Servicios Web

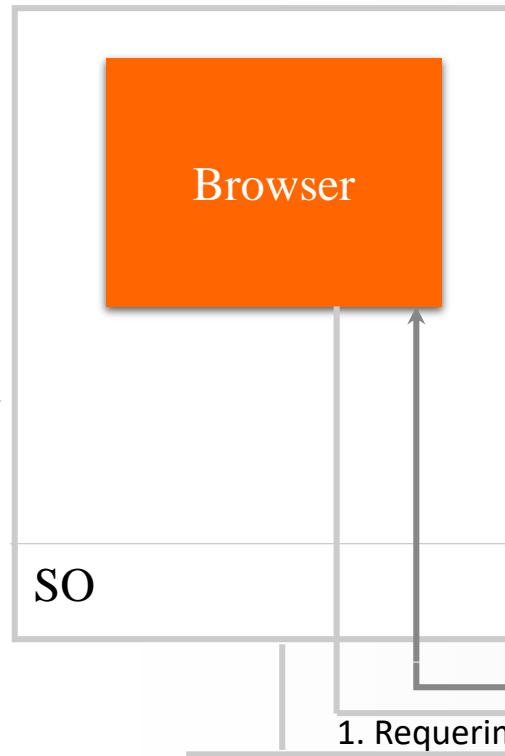
Servicio es una función que está bien definida, autocontenido y no depende del contexto o el estado de otros servicios

*“un **Servicio Web** es cualquier servicio disponible sobre Internet, utiliza un sistema de mensajería standard (XML) y no está ligado a un sistema operativo o lenguaje de programación”*

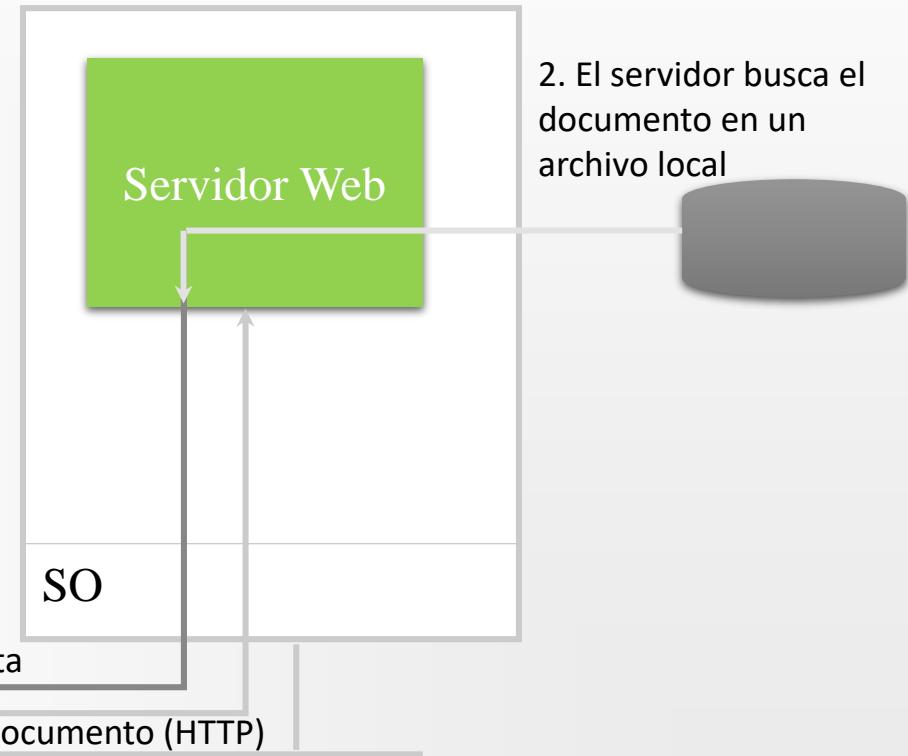


Sistema Tradicional Basado en la Web

Máquina Cliente

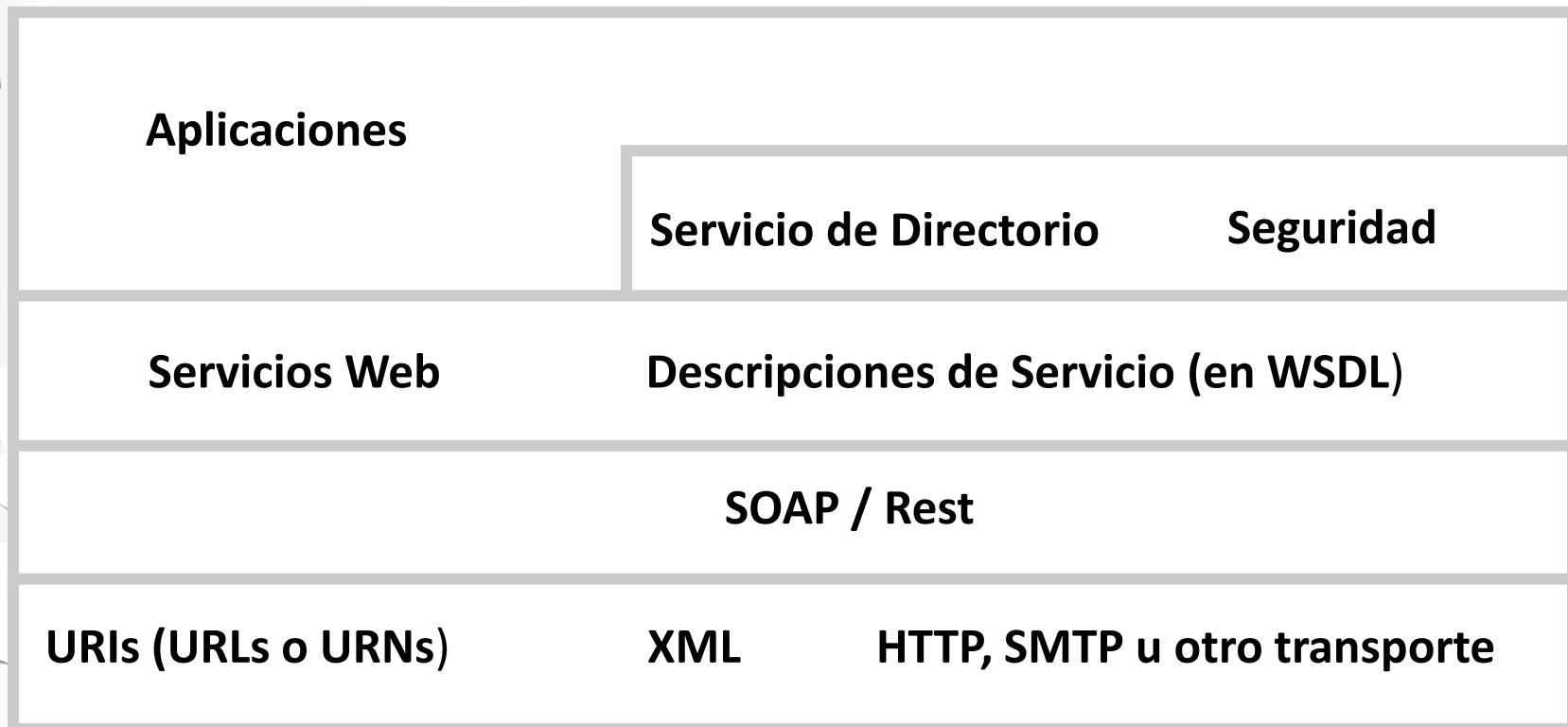


Máquina Servidor





Infraestructura de Servicios Web





Servicios Web

- Los Servicios Web son la evolución de tecnologías como RPC, ORPC (DCOM, CORBA, y JAVA RMI).
- Los Servicios Web se originaron para resolver tres problemas principales:
 1. Interoperatividad.
 2. Atravesar firewalls.
 3. Complejidad.



Servicios Web

INTEROPERABILIDAD

- Los primeros sistemas distribuidos tenían problemas de interoperatividad, cada proveedor implementaba sus propios formatos para el envío de mensajes.
 - Aplicaciones DCOM ligadas a Windows
 - Aplicaciones RMI ligadas a Java

ATRAVESAR FIREWALLS

- Proyectos de cooperación entre corporaciones: difícil. CORBA vs. DCOM.
- Los Servicios Web emplean HTTP como protocolo de transporte y la mayoría de los firewalls permite acceso a través del puerto 80 facilitando la colaboración.



Servicios Web

COMPLEJIDAD

- Las tecnologías para Servicios Web son *amigables* a los desarrolladores.
- La mayoría de las tecnologías antes mencionadas (RMI, COM, CORBA) involucran una curva completa de aprendizaje.
- Deben aprenderse nuevas tecnologías y lenguajes para implementar estos servicios.

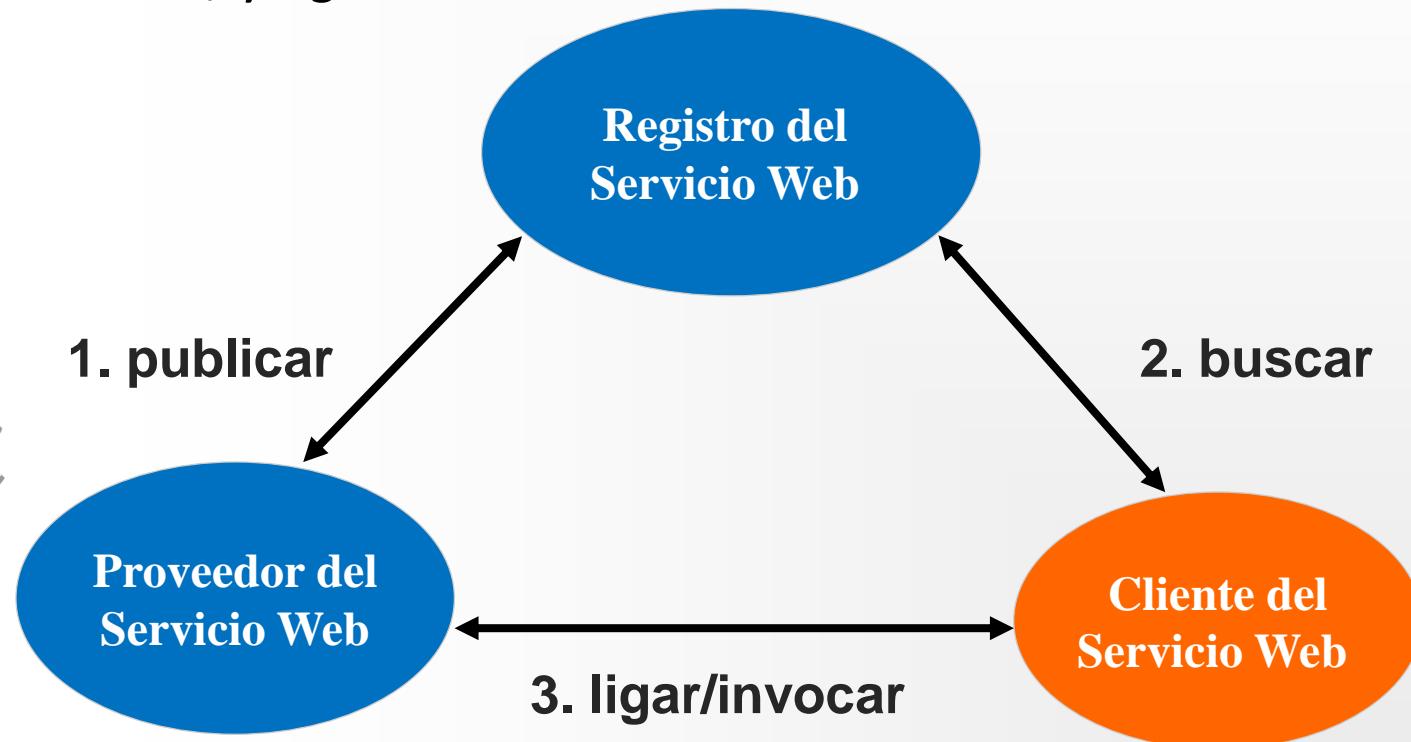


Servicios Web - Modelo

- La arquitectura de Servicios Web se basa en estos tres componentes:
 - Registro del servicio
 - Proveedor del servicio
 - Solicitante del servicio
- La interacción entre estos componentes involucra:
 - Operaciones de publicación
 - Operación de búsqueda
 - Operaciones de ligado (*binding*)/invocación

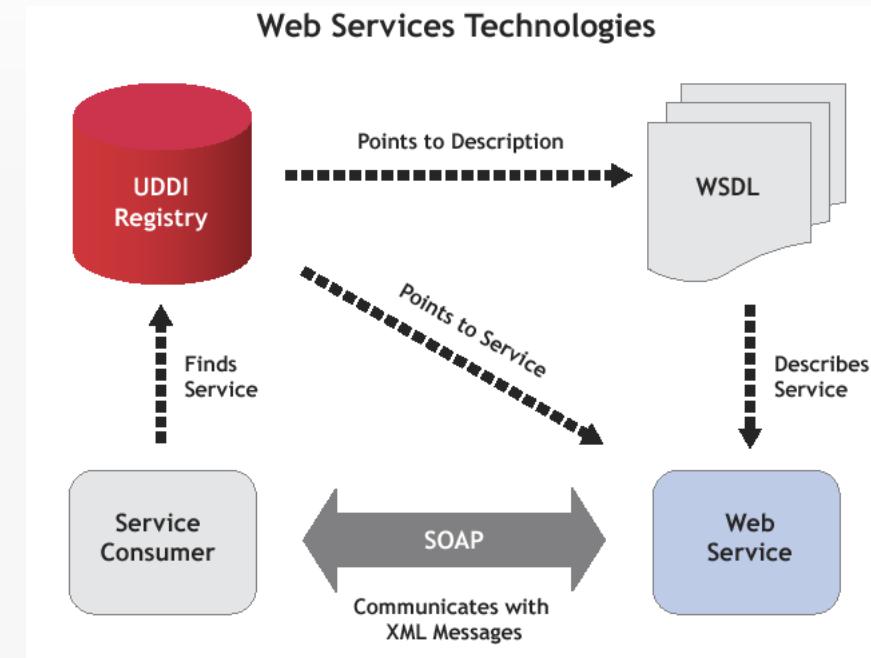
Servicios Web - Modelo

- El modelo Servicio Web sigue el paradigma de *publicar*, *buscar*, y *ligar*.

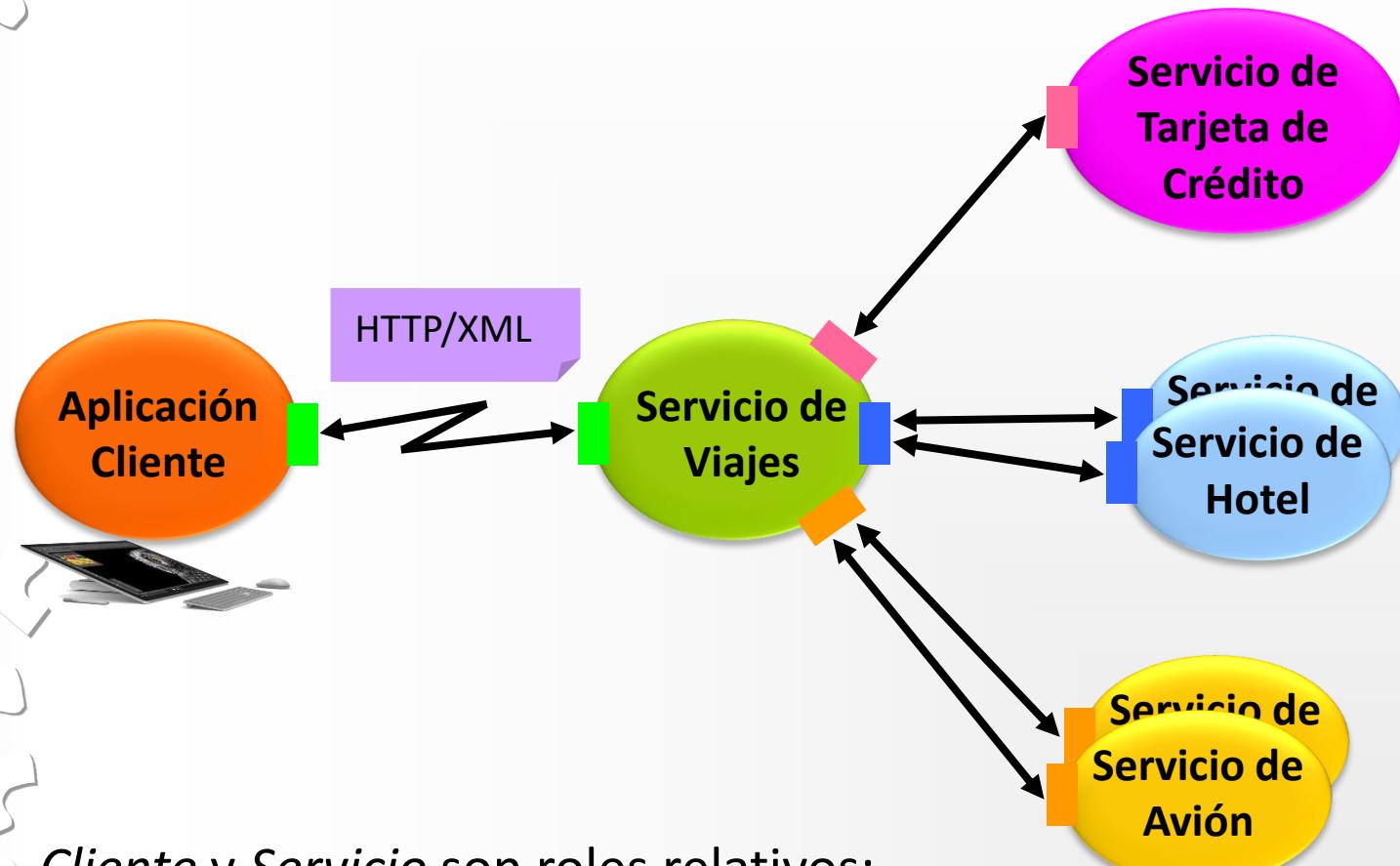


Servicios Web - Componentes

- **XML – eXtensible Markup Language** – Un mecanismo uniforme de representación e intercambio de datos.
- **SOAP – Simple Object Access Protocol** – Un estándar de comunicación.
- **UDDI – Universal Description, Discovery and Integration specification** – Un mecanismo para registrar y localizar una aplicación WS.
- **WSDL – Web Services Description Language** – Un metalenguaje estándar para describir los servicios ofrecidos.

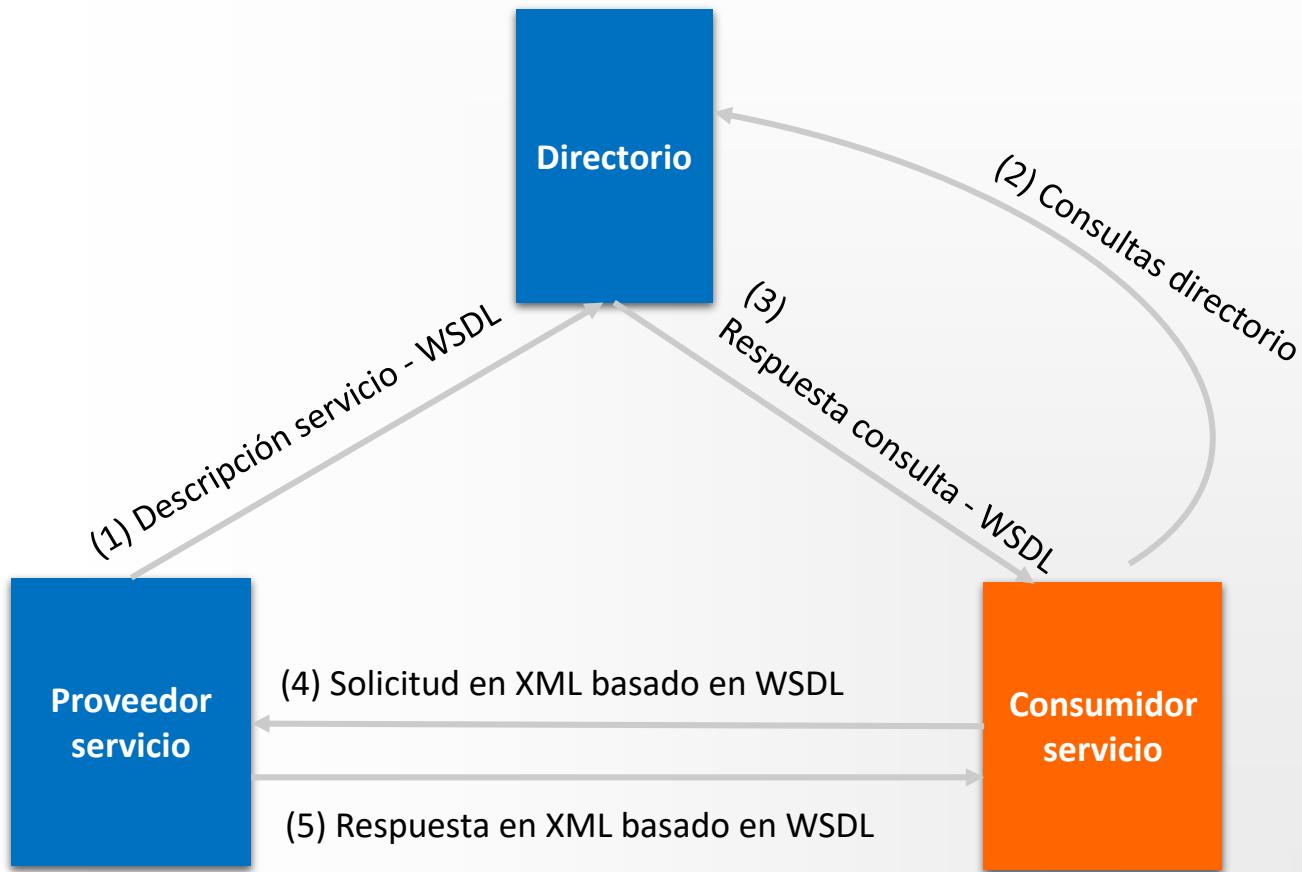


Servicios Web: Ejemplo Servicio de Viajes



Cliente y *Servicio* son roles relativos:
Servicio podría ser Cliente de otros Web Services.

Servicios Web – Ejemplo usando Mensajes SOAP





Servicios Web - SOAP

- **SOAP** es un vocabulario estándar XML que permite que los programas interactúen en una red. SOAP es un lenguaje simple para describir mensajes entre aplicaciones. HTTP es uno de los protocolos de comunicación de transportes más utilizados.

CARACTERÍSTICAS

- Extensibilidad – seguridad y ruteo de servicios web están entre las extensiones bajo desarrollo.
- Neutralidad – SOAP puede ser usado sobre cualquier protocolo de transporte tales como HTTP, SMTP o aún TCP.
- Independiente – SOAP se puede utilizar en cualquier modelo de programación.



Servicios Web – SOAP

Un mensaje SOAP es un documento común XML conteniendo los siguientes elementos:

- Se requiere un elemento **Envelope** que identifica el documento XML como mensaje SOAP.
- Un elemento opcional **Header** que contiene información de encabezamiento.
- Un elemento **Body** que contiene información de llamada y respuesta.
- Un elemento **Fault** que provee información acerca de errores que ocurrieron durante el procesamiento del mensaje.

Servicios Web – SOAP Mensaje

envelope

header

elemento header

elemento header

body

elemento body

elemento body

Servicios Web – SOAP – Ejemplo sin headers

- Cada elemento XML está representado por una caja sombreada con su nombre en *italica* seguido por algún atributo y su contenido

env:envelope xmlns:env =namespace URI for SOAP envelopes

env:body

m:exchange

xmlns:m = namespace URI of the service description

m:arg1

Hello

m:arg2

World

Servicios Web – SOAP – Ejemplo Respuesta

env:envelope *xmlns:env* = namespace URI for SOAP envelope

env:body

m:exchangeResponse

xmlns:m = namespace URI for the service description

m:res1
World

m:res2
Hello

Servicios Web – SOAP – Ejemplo

Requerimiento

POST /InStock HTTP/1.1

Host: www.stock.org

Content-Type: application/soap+xml; charset=utf-8 Content-Length: 150

```
<?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.stock.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Servicios Web – SOAP – Ejemplo Respuesta

HTTP/1.1 200 OK

Content-Type: application/soap; charset=utf-8

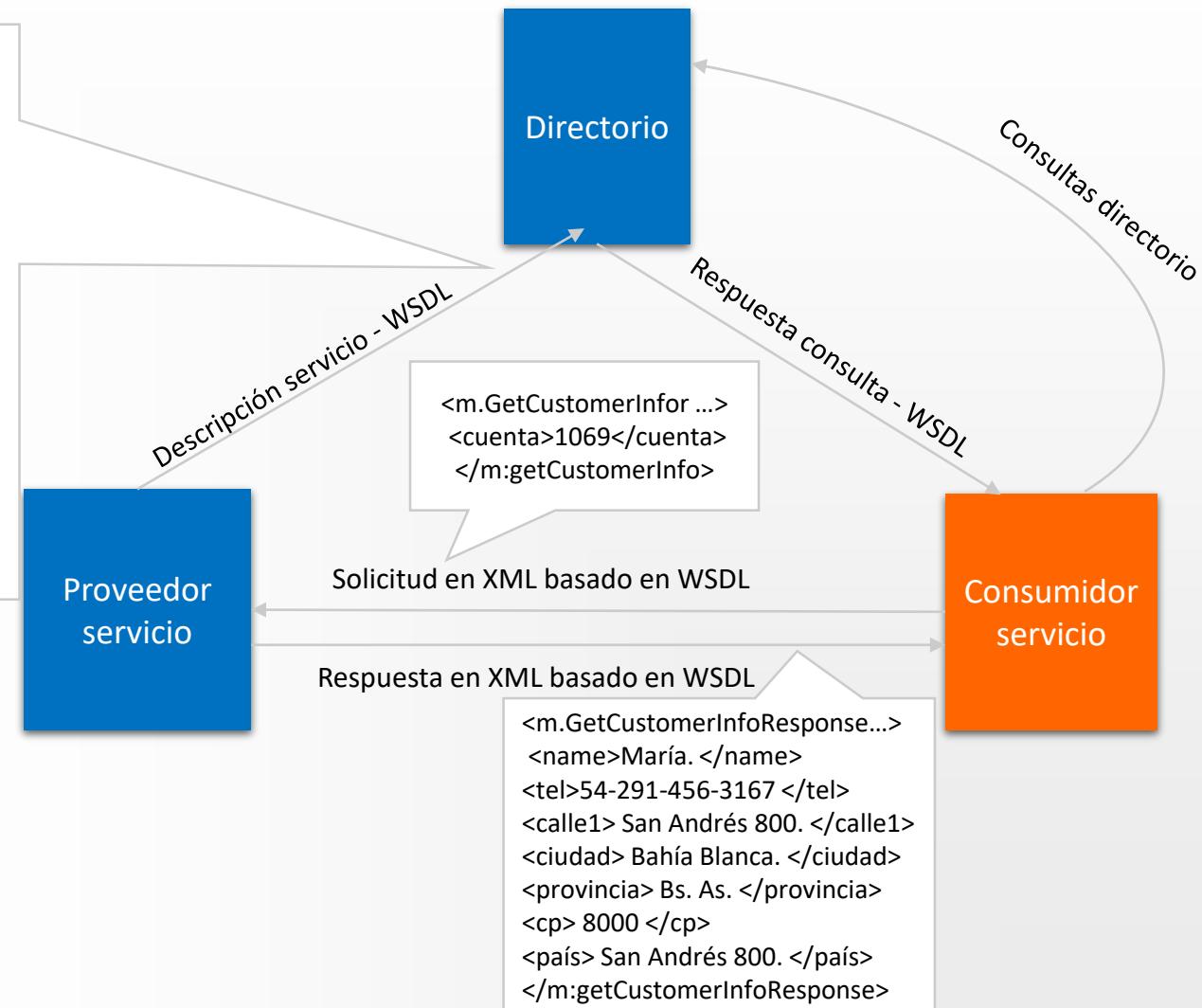
Content-Length: 126

```
<?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.stock.org/stock">
        <m:GetStockPriceResponse>
            <m:Price>34.5</m:Price>
        </m:GetStockPriceResponse>
    </soap:Body>
</soap:Envelope>
```

Servicios Web - SOAP

```
<element name="CustomerInfoRequest">
...
<element name="cuenta" type="string"/>
...
</element>
<element name="CustomerInfoResponse">
...
<element name="name" type="string"/>
<element name="tel" type="string"/>
<element name="calle1" type="string"/>
<element name="ciudad" type="string"/>
<element name="provincia" type="string"/>
<element name="cp" type="string"/>
<element name="pais" type="string"/>
...
</element>
```





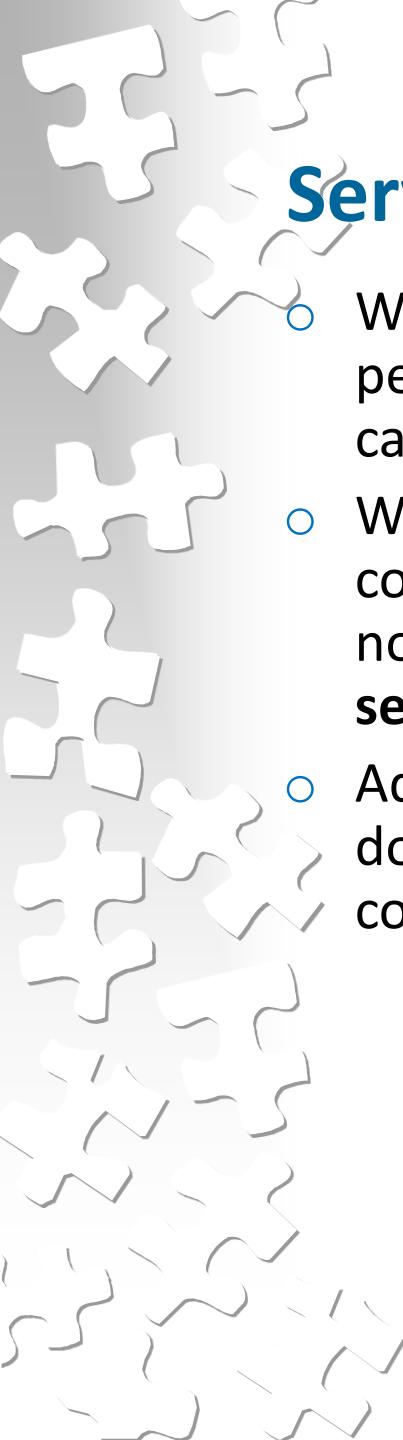
Servicios Web - SOAP

WS-ReliableMessaging: Comunicación fiable

La entrega de un mensaje puede tener la siguiente semántica:

Semántica	Característica
Al menos una vez	Un error es reportado sino es entregado
A lo sumo una vez	No reporta error sino es entregado
Exactamente una vez	Un error es reportado sino es entregado

Para cualquier semántica ordenamiento de los mensajes.



Servicios Web - WSDL

- WSDL es un vocabulario XML para describir servicios web. Les permite a los desarrolladores describir servicios web y sus capacidades de manera standard.
- WSDL especifica lo que un mensaje de requerimiento debe contener y como debe lucir un mensaje de respuesta con una notación no ambigua. En otras palabras, es un **contrato** entre **servicio web XML** y el **cliente** que desea utilizar ese servicio.
- Además de describir el contenido del mensaje, WSDL define donde está disponible el circuito y que protocolo de comunicación es usado para conectarse con el servicio.



Servicios Web – WSDL: Estructura de Documento

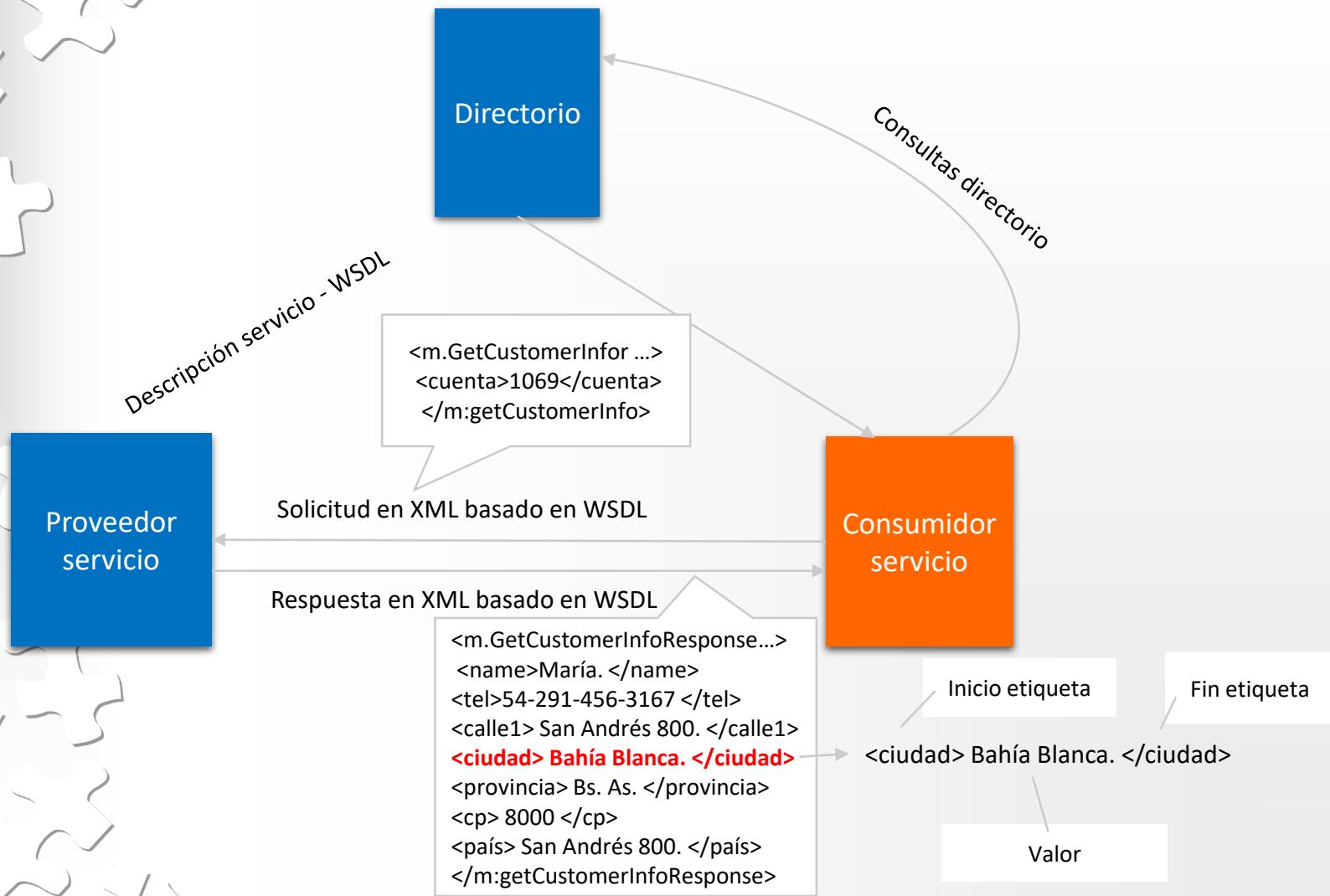
- Un documento WSDL es simplemente un documento XML.
- El mismo define un servicio web usando los siguientes elementos:
 - **port type** - Las operaciones realizadas por el servicio web.
 - **message** – El mensaje usado por el servicio web.
 - **types** – El tipo de dato usado por el servicio web.
 - **binding** - los protocolos de comunicación usados por el servicio web.

Servicios Web – WSDL - Documento

```
<message name="GetStockPriceRequest">
  <part name="stock" type="xs:string"/>
</message>
<message name="GetStockPriceResponse">
  <part name="value" type="xs:string"/>
</message>

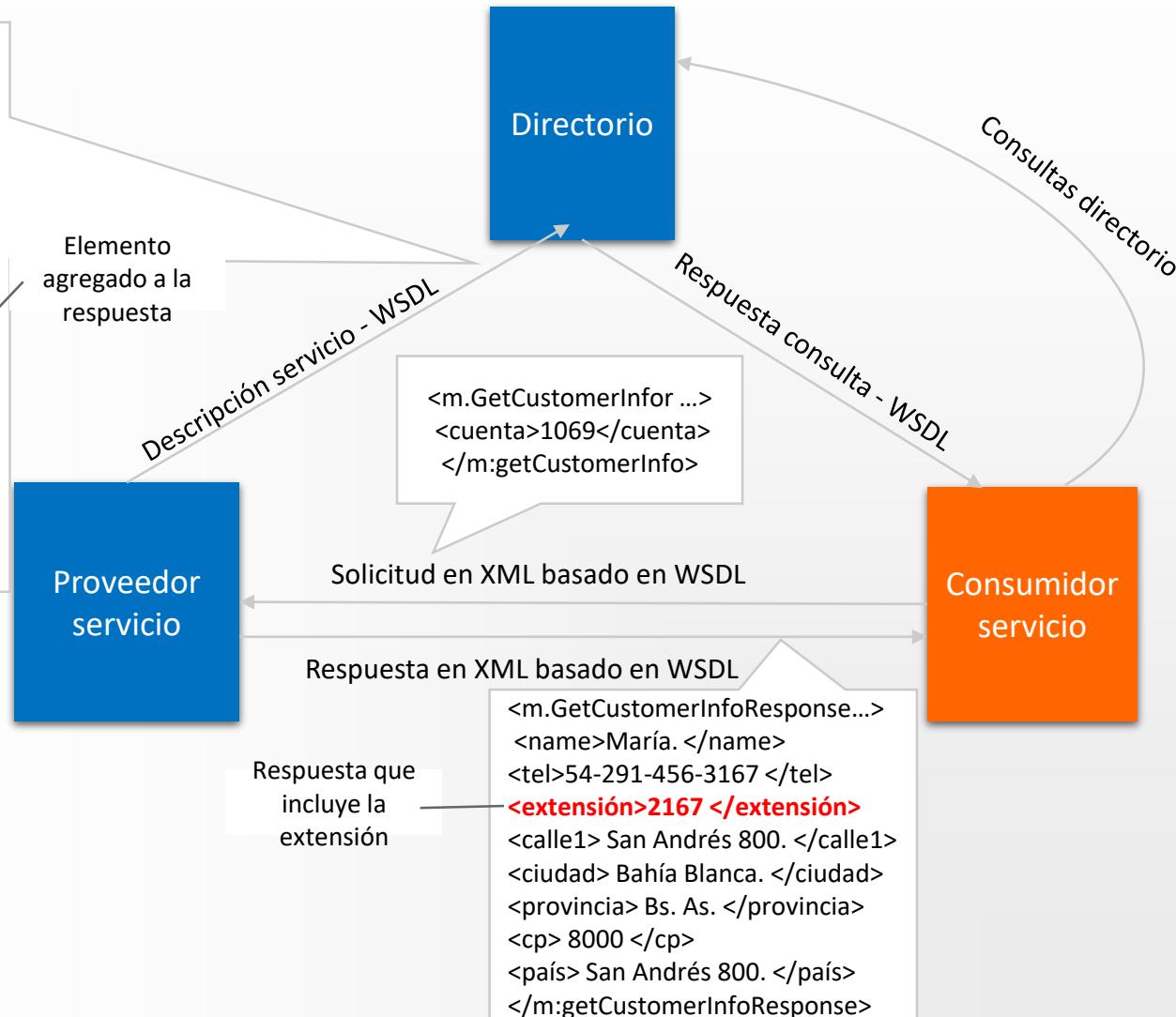
<portType name="StocksRates">
  <operation name="GetStockPrice">
    <input message="GetStockPriceRequest"/>
    <output message="GetStockPriceResponse"/>
  </operation>
</portType>
```

Servicios Web – XML con WSDL



Servicios Web – XML con WSDL

```
<element name="CustomerInfoRequest">  
...  
<element name="cuenta" type="string"/>  
...  
</element>  
<element name="CustomerInfoResponse">  
...  
<element name="name" type="string"/>  
<element name="tel" type="string"/>  
<element name="extensión" type="string"/>  
<element name="calle1" type="string"/>  
<element name="ciudad" type="string"/>  
<element name="provincia" type="string"/>  
<element name="cp" type="string"/>  
<element name="país" type="string"/>  
...  
</element>
```





Servicios Web - Rest

- El protocolo utiliza las siguientes operaciones: HTTP Get, HTTP Post, HTTP Put y HTTP Del.
- Escalable.
- Escaso consumo de recursos.
- XML autodescriptivo.
- No mantiene estado.

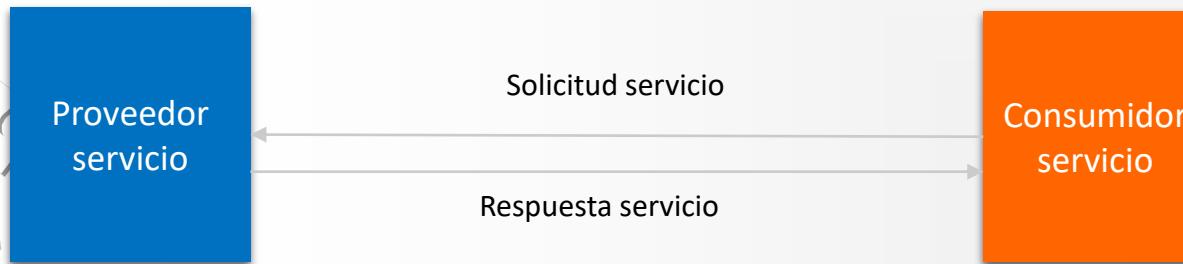


Rest y SOAP

Ventajas REST	Ventajas SOAP
<ul style="list-style-type: none">• Pocas operaciones con muchos recursos• Se centra en la escalabilidad y rendimiento a gran escala para sistemas distribuidos hipermédia• HTTP Get, HTTP Post, HTTP Put, HTTP Del• XML auto descriptivo• Síncrono• HTTPS• Comunicación punto a punto y segura• Sin estado	<ul style="list-style-type: none">• Muchas operaciones con pocos recursos• Se centra en el diseño de aplicaciones distribuidas• SMTP, HTTP POST, MQ• Tipado fuerte, XML Schema• Síncrono y Asíncrono• WS Security• Comunicación origen a destino segura• Sin estado por defecto pero se puede configurar con estado

Servicios Web - SOA

Una arquitectura orientada al servicio (Service-oriented Architecture – SOA) es esencialmente una colección de servicios. Estos servicios se comunican entre sí.





Servicios Web - Resumen

- Están basados en estándares que tienen amplio apoyo de la industria.
- Emplean tecnologías simples y probadas, e.g. HTTP y XML.
- Logran interoperatividad entre sistemas que difieren en software y hardware.



Bibliografía:

- Coulouris, G.F.; Dollimore, J. y T. Kindberg; "Distributed Systems: Concepts and Design". 5th Edition Addison Wesley, 2011.
- Barry, Douglas; "Web Services and Service-Oriented Architectures and Cloud Computing (The Savvy Manager's Guides)", 1er. Ed Morgan Kaufmann, 2003, 2da. Ed. 2013.
- Tanenbaum, A.S.; van Steen, Maarten; "Distributed Systems: Principles and Paradigms". 2nd Edition, Prentice Hall, 2007 and 1st Edition 2002.