

 Departamento de Lenguajes y
Sistemas Informáticos

 Universitat d'Alacant
Universidad de Alicante

Servicios web

Programación en Internet
Curso 2007-2008

Programación en Internet – Curso 2007-2008

Contenido

- Introducción
- Los pilares (SOAP, WSDL, UDDI)
- Desarrollo de un servicio web

Programación en Internet – Curso 2007-2008

Introducción

Programación en Internet – Curso 2007-2008

Introducción

- Servicios web (*web services*) son un nuevo modelo de desarrollo de aplicaciones distribuidas basadas en Internet
- Un servicio web es un componente al que se puede acceder mediante protocolos Web estándar:
 - Los mensajes para invocar el servicio se codifican en XML
 - Los mensajes se pueden transportar utilizando HTTP o cualquier otro protocolo de transporte estándar
- Estándar de W3C
 - XML
 - SOAP
 - WSDL

Introducción

- Algunas características:
 - Permiten que las aplicaciones compartan información de forma remota
 - Permiten integrar sistemas heterogéneos de manera barata y fiable
 - Son unidades de código discretas, cada una de las cuales se encarga de un conjunto limitado de tareas
 - Están basados en XML, el lenguaje estándar de intercambio de información en Internet
 - Pueden utilizarse en cualquier plataforma o sistema operativo, independientemente del lenguaje de programación utilizado

Introducción

- ¿Qué se puede hacer?
 - Los primeros servicios web eran únicamente informativos
 - Se están explorando posibles usos
- Posible ejemplo:
 - Un servicio de reserva de paquetes de viaje
 - Un servicio de agenda compartido
 - Consulta de cotizaciones de bolsa, cambio de moneda, datos meteorológicos, etc.

Programación en Internet – Curso 2007-2008

Introducción

- Aunque los servicios web son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada → Orquestación de servicios web (*web services orchestration*)

Programación en Internet – Curso 2007-2008

Introducción

- Ventajas de servicios web:
 - Menos complejo: sencillos de implementar, probar y usar
 - Basado en protocolos abiertos de la Web (XML, HTTP y TCP/IP) ampliamente aceptados → Intoreparabilidad total
 - Al emplear protocolos de comunicación estándar (HTTP), normalmente atraviesa *firewalls* y *routers* sin problemas
 - Se puede utilizar tanto de forma anónima como con autenticación (nombre/clave) y a través de HTTPS

Programación en Internet – Curso 2007-2008

Introducción

- ¿Es algo totalmente nuevo?
- Tecnologías integración:
 - RPC
 - CORBA
 - DCE
 - RMI
 - ...

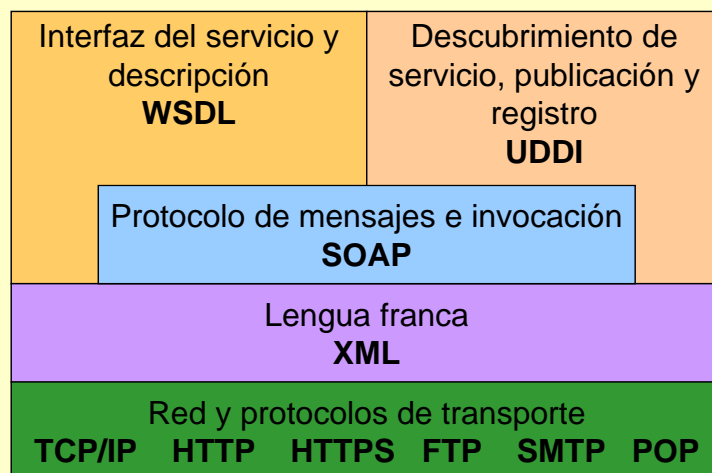
Programación en Internet – Curso 2007-2008

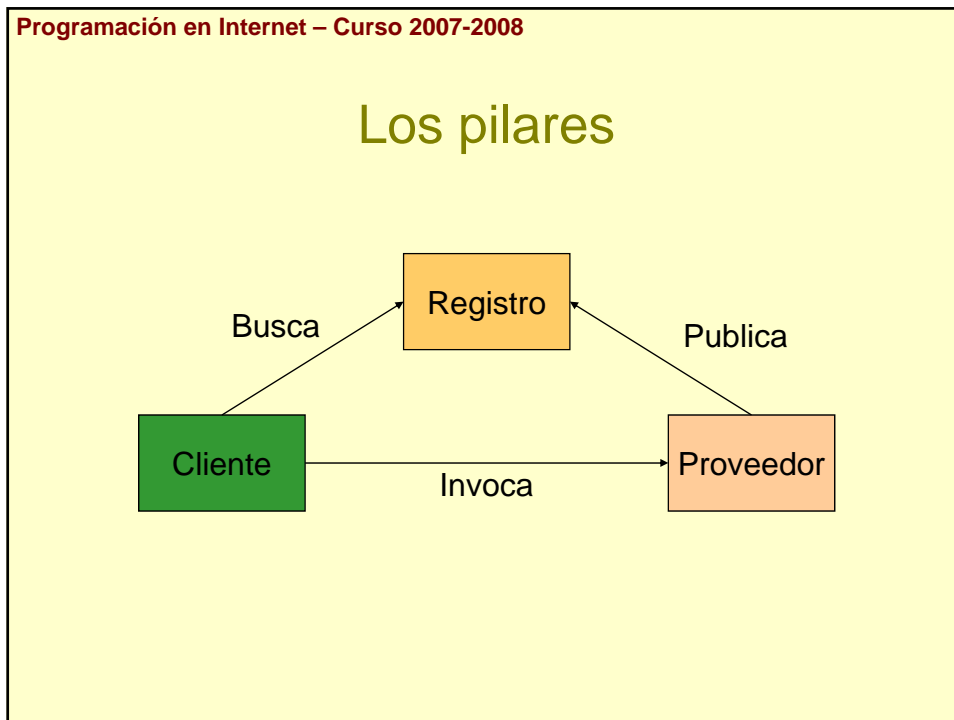
Los pilares

Los pilares

- Servicios web se sustentan sobre tres pilares:
 - SOAP: *Simple Object Access Protocol* (Protocolo de acceso a objetos simple)
 - WSDL: *Web Services Description Language* (Lenguaje de descripción de servicios web)
 - UDDI: *Universal Description, Discovery, and Integration* (Descripción, descubrimiento e integración universales)
- Servicio web: **software accesible en la Web a través de SOAP, descrito con un fichero WSDL y registrado en UDDI**

Los pilares

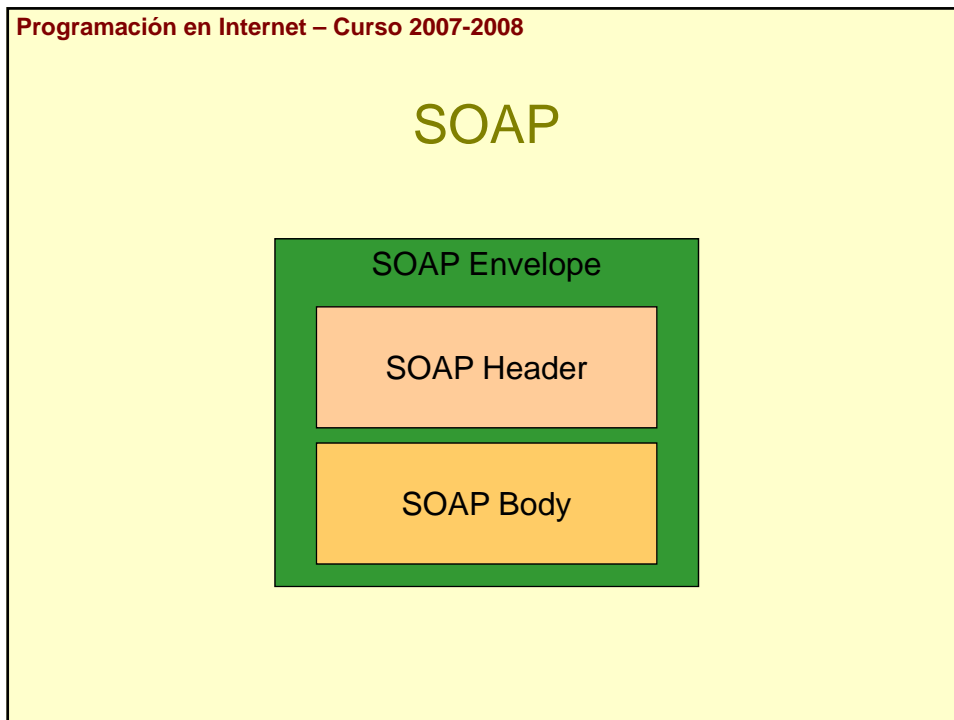




Programación en Internet – Curso 2007-2008

SOAP

- *Simple Object Access Protocol*
- Tecnología de mensajería basada en XML que especifica todas las reglas necesarias para ubicar servicios web, integrarlos en aplicaciones y establecer la comunicación entre ellos
- Recomendación: <http://www.w3.org/TR/soap>
- Implementaciones de SOAP:
 - <http://www.soapware.org/directory/4/implementations>
 - IBM Web Services Toolkit
 - Microsoft SOAP Toolkit 3.0
 - Sun Java Web Services Developer Pack 1.3
 - PHP-SOAP, NuSOAP



- Programación en Internet – Curso 2007-2008
- ## SOAP
- **Envelope:** define los espacios de nombres (*namespaces*) que serán empleados en el resto del mensaje. Los valores típicos son:
 - xmlns:SOAP-ENV → SOAP Envelope namespace
 - xmlns:xsi → XML Schema for Instances
 - xmlns:xsd → XML Schema for DataTypes
 - **Header:** elemento opcional que proporciona información auxiliar para los procesos de autenticación, las transacciones y los pagos
 - **Body:** es la parte principal del mensaje y contiene el nombre del método y los parámetros necesarios para invocar el servicio web

SOAP

- Con SOAP podemos intercambiar cualquier documento XML, pero no otro tipo (imágenes, documentos PDF, etc.)
- *SOAP with Attachment (SwA) → SOAP Message Transmission Optimization Mechanism:*
 - Permite añadir datos que no sean XML al mensaje
 - Parte adjunta (*attachment*)

SOAP

```
POST /soap/servlet/rpcrouter HTTP/1.0
Host: localhost:8070
Content-Type: text/xml
Content-Length: 461
SOAPAction: ""

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getRate xmlns:ns1="urn:demo1:exchange"
      SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding
      /">
      <country1 xsi:type="xsd:string">USA</country1>
      <country2 xsi:type="xsd:string">japan</country2>
    </ns1:getRate>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Programación en Internet – Curso 2007-2008

SOAP

```
<Envelope>
  <Body>
    <getRate>
      <country1>USA</country1>
      <country2>japan</country2>
    </getRate>
  </Body>
</Envelope>
```

Programación en Internet – Curso 2007-2008

SOAP

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=UTF-8
Content-Length: 425
Set-Cookie2:
  JSESSIONID=4x1b3dqoc1;Version=1;Discard;Path="/soap"
Set-Cookie: JSESSIONID=4x1b3dqoc1;Path=/soap
Servlet-Engine: Tomcat Web Server/3.2 (final) (JSP 1.1;
Servlet 2.2; Java 1.3.0; Windows 2000 5.0 x86;
java.vendor=Sun Microsystems Inc.)

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getRateResponse xmlns:ns1="urn:demo1:exchange"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding
/">
    <return xsi:type="xsd:float">144.52</return>
  </ns1:getRateResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Programación en Internet – Curso 2007-2008

SOAP

```
<Envelope>
  <Body>
    <getRateResponse>
      <return>144.52</return>
    </getRateResponse>
  </Body>
</Envelope>
```

Programación en Internet – Curso 2007-2008

SOAP

```
HTTP/1.0 500 Internal Server Error
Content-Type: text/xml; charset=UTF-8
Content-Length: 411
Set-Cookie2:
  JSESSIONID=ha5k5ds6h1;Version=1;Discard;Path="/soap"
Set-Cookie: JSESSIONID= ha5k5ds6h1;Path=/soap
Servlet-Engine: Tomcat Web Server/3.2 (final) (JSP 1.1;
  Servlet 2.2; Java 1.3.0; Windows 2000 5.0 x86;
  java.vendor=Sun Microsystems Inc.)
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>cannot calculate rate</faultstring>
      <faultactor>/soap/servlet/rpcrouter</faultactor>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Programación en Internet – Curso 2007-2008

SOAP

```
<Envelope>
  <Body>
    <Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>cannot calculate rate</faultstring>
      <faultactor>/soap/servlet/rpcrouter</faultactor>
    </Fault>
  </Body>
</Envelope>
```

Programación en Internet – Curso 2007-2008

WSDL

- *Web Services Description Language*
- Lenguaje para describir los servicios web como una serie de operaciones que operan sobre mensajes
- Las operaciones y mensajes se describen de forma abstracta y no están ligados a ningún lenguaje de programación concreto
- Recomendación: <http://www.w3.org/TR/wsd/>

WSDL

- WSDL permite que un servicio web se describa a sí mismo:
 - Proporciona:
 - URL del *endpoint* (punto de conexión al servicio)
 - Nombre de la función a llamar
 - Nombre y tipo de los parámetros
 - Tipo del valor de retorno
 - ...
 - De esta forma una aplicación podrá conocer cuál es la interfaz del servicio, y podrá integrarlo y utilizarlo de forma automática

WSDL

- Un WSDL se crea:
 - A mano
 - Clases → Generación automática del WSDL
- Un WSDL se utiliza:
 - A mano
 - WSDL → Generación automática de un cliente (clases)

WSDL

- Un servicio web se define por medio de:
 - <definitions> :
 - El elemento raíz del documento
 - Suele incluir la declaración de espacios de nombres
 - <types> :
 - Proporciona definiciones de tipos de datos que se emplean para definir los mensajes empleados en la comunicación
 - Podemos emplear XML Schema

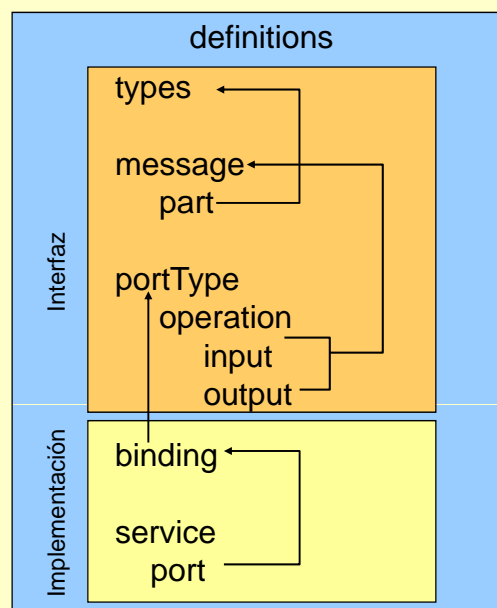
WSDL

- Un servicio web se define por medio de:
 - <message> :
 - Representación abstracta de los mensajes que se intercambian en la comunicación
 - Un mensaje está compuesto de parámetros <part> con un tipo (de los definidos en <types>)
 - Cada operación tendrá un mensaje de entrada (petición) y uno de salida (respuesta)
 - <portType> :
 - Define un conjunto abstracto de operaciones (<operation>) que ofrece el servicio
 - Cada operación contiene un mensaje de entrada (<input>) y un mensaje de salida (<output>) de los anteriores (referencia a <message>)
 - Se necesita un puerto abstracto por cada protocolo soportado: por conveniencia se coloca como sufijo el nombre del protocolo, pero no es necesario (se especifica en <binding>)

WSDL

- Un servicio web se define por medio de:
 - <binding>:
 - Conexión entre interfaz e implementación, específica:
 - Estilo de llamada (rpc, document, etc.)
 - » Consultar: <http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/>
 - Protocolo utilizado y especificaciones de formatos de datos para las operaciones y mensajes definidos en un *portType* particular
 - Método de transporte de datos (HTTP, SMTP, etc.)
 - Localización del servicio
 - <service>:
 - Define un servicio mediante una colección de puertos a los que acceder (agrega un conjunto de <port> relacionados)
 - <port>:
 - Especifica una dirección (URL) para un <binding>, es decir, define un punto de comunicación concreto (*endpoint*)
 - Contiene documentación en lenguaje natural sobre el servicio

Documento WSDL



Programación en Internet – Curso 2007-2008

WSDL

- Ejemplo:
 - Google Search Web Service
 - <http://api.google.com/GoogleSearch.wsdl>

```
<?xml version="1.0" ?>
- <!--
  WSDL description of the Google Web APIs.
  The Google Web APIs are in beta release. All interfaces are subject to
  change as we refine and extend our APIs. Please see the terms of use
  for more information.
-->
- <!-- Revision 2002-08-16 -->
- <definitions name="GoogleSearch" targetNamespace="urn:GoogleSearch"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <!-- Types for search - result elements, directory categories -->
  + <types>
  <!-- Messages for Google Web APIs - cached page, search, spelling. -->
  + <message name="doGetCachedPage">
  + <message name="doGetCachedPageResponse">
  + <message name="doSpellingSuggestion">
  + <message name="doSpellingSuggestionResponse">
  <!-- note, ie and oe are ignored by server; all traffic is UTF-8. -->
  + <message name="doGoogleSearch">
  + <message name="doGoogleSearchResponse">
  <!-- Port for Google Web APIs, "GoogleSearch" -->
  | <portType name="GoogleSearchPort">
  <!-- Binding for Google Web APIs - RPC, SOAP over HTTP -->
  + <binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
  <!-- Endpoint for Google Web APIs -->
  + <service name="GoogleSearchService">
  </definitions>
```



```

<?xml version="1.0" ?>
<!--
  WSDL description of the Google Web APIs.
  The Google Web APIs are in beta release. All interfaces are subject to
  change as we refine and extend our APIs. Please see the terms of use
  for more information.
-->
<!-- Revision 2002-08-16 -->
<definitions name="GoogleSearch" targetNamespace="urn:GoogleSearch"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  >
  <!-- Types for search - result elements, directory categories -->
  <types>
    <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:GoogleSearch">
      <xsd:complexType name="GoogleSearchResult">
        <xsd:all>
          <xsd:element name="documentFiltering" type="xsd:boolean" />
          <xsd:element name="searchComments" type="xsd:string" />
          <xsd:element name="estimatedTotalResultsCount" type="xsd:int" />
          <xsd:element name="estimateIsExact" type="xsd:boolean" />
          <xsd:element name="resultElements" type="typens:ResultElementArray" />
          <xsd:element name="searchQuery" type="xsd:string" />
          <xsd:element name="startIndex" type="xsd:int" />
          <xsd:element name="endIndex" type="xsd:int" />
          <xsd:element name="searchTips" type="xsd:string" />
          <xsd:element name="directoryCategories" type="typens:DirectoryCategoryArray" />
          <xsd:element name="searchTime" type="xsd:double" />
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="ResultElement">
        <xsd:all>
          <xsd:element name="summary" type="xsd:string" />
          <xsd:element name="URL" type="xsd:string" />
          <xsd:element name="snippet" type="xsd:string" />
          <xsd:element name="title" type="xsd:string" />
          <xsd:element name="cachedSize" type="xsd:string" />
        </xsd:all>
      </xsd:complexType>
    </xsd:schema>
  </types>
</definitions>
  
```

Programación en Internet – Curso 2007-2008

WSDL

- <service>
 - Nombre: GoogleSearchService
 - Dirección: <http://api.google.com/search/beta2>
 - No ofrece documentación

```

<!-- Endpoint for Google Web APIs -->
<service name="GoogleSearchService">
  <port name="GoogleSearchPort" binding="typens:GoogleSearchBinding">
    <soap:address location="http://api.google.com/search/beta2" />
  </port>
</service>
  
```

Programación en Internet – Curso 2007-2008

WSDL

- <binding>
 - Ofrece tres operaciones:
 - doGetCachedPage
 - doSpellingSuggestion
 - doGoogleSearch
 - El método de llamada es RPC
 - El protocolo utilizado es SOAP
 - El método de transporte de datos es HTTP

Programación en Internet – Curso 2007-2008

```
<!-- Binding for Google Web APIs - RPC, SOAP over HTTP -->
<binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="doGetCachedPage">
    <soap:operation soapAction="urn:GoogleSearchAction" />
    <input>
      <soap:body use="encoded" namespace="urn:GoogleSearch" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:GoogleSearch" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="doSpellingSuggestion">
    <soap:operation soapAction="urn:GoogleSearchAction" />
    <input>
      <soap:body use="encoded" namespace="urn:GoogleSearch" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:GoogleSearch" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="doGoogleSearch">
    <soap:operation soapAction="urn:GoogleSearchAction" />
    <input>
      <soap:body use="encoded" namespace="urn:GoogleSearch" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:GoogleSearch" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
</binding>
```

WSDL

- `<portType>`
 - Define las operaciones, con una referencia al mensaje de entrada y al mensaje de salida

```
<!-- Port for Google Web APIs, "GoogleSearch" -->
- <portType name="GoogleSearchPort">
- <operation name="doGetCachedPage">
  <input message="typens:doGetCachedPage" />
  <output message="typens:doGetCachedPageResponse" />
</operation>
- <operation name="doSpellingSuggestion">
  <input message="typens:doSpellingSuggestion" />
  <output message="typens:doSpellingSuggestionResponse" />
</operation>
- <operation name="doGoogleSearch">
  <input message="typens:doGoogleSearch" />
  <output message="typens:doGoogleSearchResponse" />
</operation>
</portType>
```

WSDL

- `<message>`
 - Define los parámetros de los mensajes de entrada y salida
 - Para cada parámetro se indica su tipo de dato

Programación en Internet – Curso 2007-2008

```
<!-- Messages for Google Web APIs - cached page, search, spelling. -->
- <message name="doGetCachedPage">
  <part name="key" type="xsd:string" />
  <part name="url" type="xsd:string" />
</message>
- <message name="doGetCachedPageResponse">
  <part name="return" type="xsd:base64Binary" />
</message>
- <message name="doSpellingSuggestion">
  <part name="key" type="xsd:string" />
  <part name="phrase" type="xsd:string" />
</message>
- <message name="doSpellingSuggestionResponse">
  <part name="return" type="xsd:string" />
</message>
<!-- note, ie and oe are ignored by server; all traffic is UTF-8. -->
- <message name="doGoogleSearch">
  <part name="key" type="xsd:string" />
  <part name="q" type="xsd:string" />
  <part name="start" type="xsd:int" />
  <part name="maxResults" type="xsd:int" />
  <part name="filter" type="xsd:boolean" />
  <part name="restrict" type="xsd:string" />
  <part name="safeSearch" type="xsd:boolean" />
  <part name="lr" type="xsd:string" />
  <part name="ie" type="xsd:string" />
  <part name="oe" type="xsd:string" />
</message>
- <message name="doGoogleSearchResponse">
  <part name="return" type="typens:GoogleSearchResult" />
</message>
```

Programación en Internet – Curso 2007-2008

WSDL

- <types>
 - Define los tipos de datos de los parámetros de los mensajes

```

Pri <!-- Types for search - result elements, directory categories -->
<types>
- <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:GoogleSearch">
- <xsd:complexType name="GoogleSearchResult">
- <xsd:all>
<xsd:element name="documentFiltering" type="xsd:boolean" />
<xsd:element name="searchComments" type="xsd:string" />
<xsd:element name="estimatedTotalResultsCount" type="xsd:int" />
<xsd:element name="estimateIsExact" type="xsd:boolean" />
<xsd:element name="resultElements" type="typens:ResultElementArray" />
<xsd:element name="searchQuery" type="xsd:string" />
<xsd:element name="startIndex" type="xsd:int" />
<xsd:element name="endIndex" type="xsd:int" />
<xsd:element name="searchTips" type="xsd:string" />
<xsd:element name="directoryCategories" type="typens:DirectoryCategoryArray" />
<xsd:element name="searchTime" type="xsd:double" />
</xsd:all>
</xsd:complexType>
- <xsd:complexType name="ResultElement">
- <xsd:all>
<xsd:element name="summary" type="xsd:string" />
<xsd:element name="URL" type="xsd:string" />
<xsd:element name="snippet" type="xsd:string" />
<xsd:element name="title" type="xsd:string" />
<xsd:element name="cachedSize" type="xsd:string" />
<xsd:element name="relatedInformationPresent" type="xsd:boolean" />
<xsd:element name="hostName" type="xsd:string" />
<xsd:element name="directoryCategory" type="typens:DirectoryCategory" />
<xsd:element name="directoryTitle" type="xsd:string" />
</xsd:all>
</xsd:complexType>
+ <xsd:complexType name="ResultElementArray">
+ <xsd:complexType name="DirectoryCategoryArray">
+ <xsd:complexType name="DirectoryCategory">
</xsd:schema>
</types>

```

```

<!-- Endpoint for Google Web APIs -->
- <service name="GoogleSearchService">
- <port name="GoogleSearchPort" binding="typens:GoogleSearchBinding">
<soap:address location="http://api.google.com/search/beta2"/>
</port>
</service>

- <operation name="doGoogleSearch">
<input message="typens:doGoogleSearch" />
<output message="typens:doGoogleSearchResponse" />
</operation>

- <message name="doGoogleSearch">
<part name="key" type="xsd:string" />
<part name="q" type="xsd:string" />
<part name="start" type="xsd:int" />
<part name="maxResults" type="xsd:int" />
<part name="filter" type="xsd:boolean" />
<part name="restrict" type="xsd:string" />
<part name="safeSearch" type="xsd:boolean" />
<part name="lr" type="xsd:string" />
<part name="ie" type="xsd:string" />
<part name="oe" type="xsd:string" />
</message>

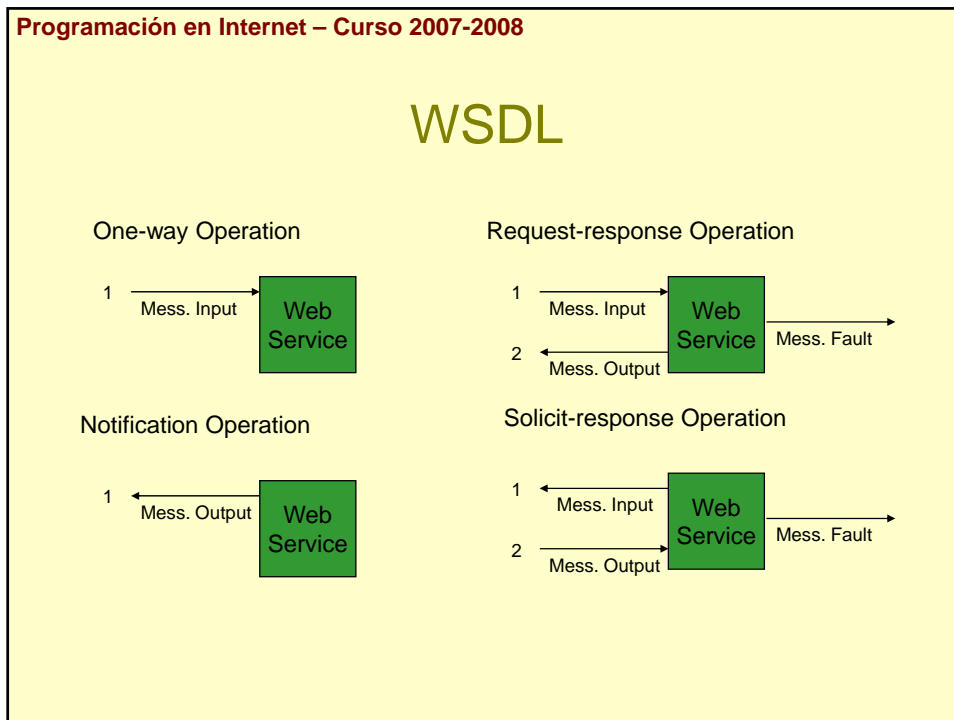
- <message name="doGoogleSearchResponse">
<part name="return" type="typens:GoogleSearchResult" />
</message>

- <xsd:complexType name="GoogleSearchResult">
- <xsd:all>
<xsd:element name="documentFiltering" type="xsd:boolean" />
<xsd:element name="searchComments" type="xsd:string" />
<xsd:element name="estimatedTotalResultsCount" type="xsd:int" />
<xsd:element name="estimateIsExact" type="xsd:boolean" />
<xsd:element name="resultElements" type="typens:ResultElementArray" />
<xsd:element name="searchQuery" type="xsd:string" />
<xsd:element name="startIndex" type="xsd:int" />
<xsd:element name="endIndex" type="xsd:int" />
<xsd:element name="searchTips" type="xsd:string" />
<xsd:element name="directoryCategories" type="typens:DirectoryCategoryArray" />
<xsd:element name="searchTime" type="xsd:double" />
</xsd:all>
</xsd:complexType>

```

• ¿Qué operaciones proporciona Google?

GoogleSearchResult doGoogleSearch(key as string, q as string, start as int, maxResults as int, filter as boolean, restrict as string, safeSearch as boolean, lr as string, ie as string, oe as string)



- Programación en Internet – Curso 2007-2008
- ## UDDI
- *Universal Description, Discovery, and Integration*
 - Es un registro público gratuito en el que se pueden publicar y enviar consultas acerca de los servicios web
 - Es similar a unas “páginas amarillas de servicios web”
 - Estándar: <http://uddi.org/>

Programación en Internet – Curso 2007-2008

UDDI

- Búsqueda automática o manual
- Un usuario podrá buscar un servicio web en base a distintos criterios:
 - Precio
 - Rendimiento (tiempo de respuesta)
 - Fiabilidad (24x7)
 - ...
- Registros:
 - Públicos
 - Privados (empresa u organización): para compartir y reutilizar código internamente

Programación en Internet – Curso 2007-2008

Desarrollo de un servicio web

Programación en Internet – Curso 2007-2008

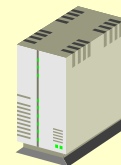
Desarrollo de un servicio web

1. El proveedor del servicio web parte de un desarrollo heredado o de un nuevo producto

Programación en Internet – Curso 2007-2008

Desarrollo de un servicio web

Sistema de información



Servidor

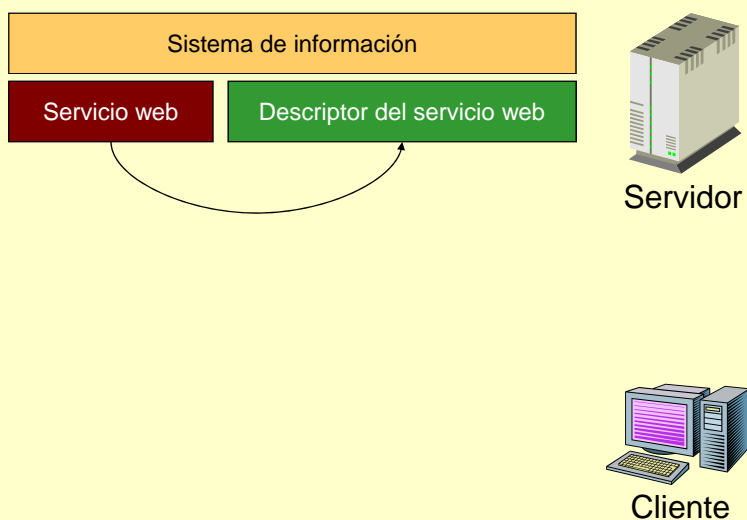


Cliente

Desarrollo de un servicio web

2. El proveedor desarrolla y publica un interfaz basado en un servicio web. Esto incluye la publicación automática, en la Web, de un descriptor del servicio (WSDL)

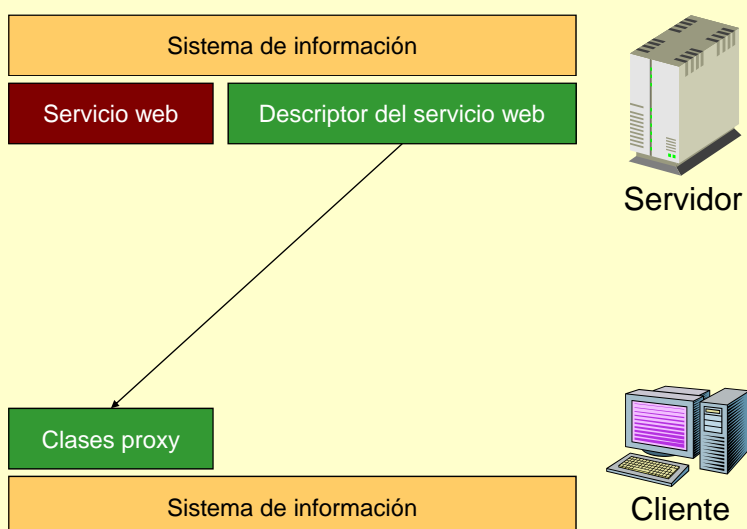
Desarrollo de un servicio web



Desarrollo de un servicio web

3. El cliente, que desea desarrollar un sistema de información propio basado en el servicio web, construye automáticamente, a partir del descriptor del servicio, las clases *proxy* (*stub*) que le permitirán acceder al servicio web desde sus aplicaciones

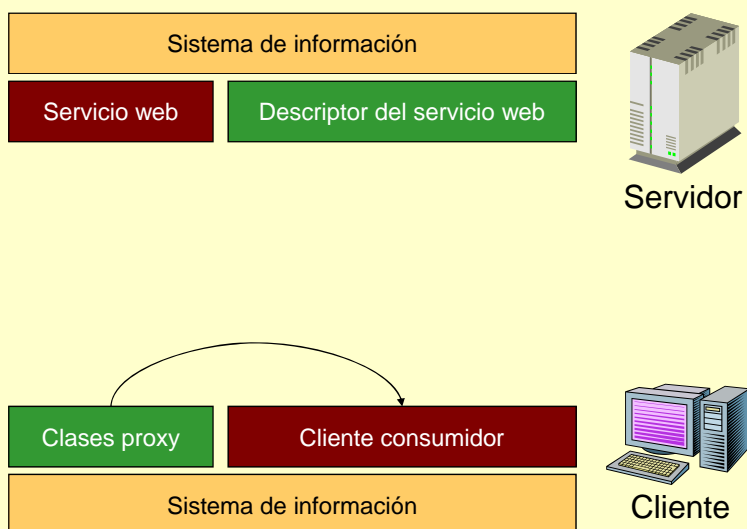
Desarrollo de un servicio web



Desarrollo de un servicio web

4. El cliente desarrolla la parte cliente que consumirá el servicio web, usando las clases *proxy*

Desarrollo de un servicio web



Desarrollo de un servicio web

5. El cliente ya está en disposición de emplear el servicio web

Desarrollo de un servicio web

