

Servicios Web (I)

Jesús Arias, Norberto Fernández
Departamento de Ingeniería Telemática

<http://www.it.uc3m.es/jaf/>
<http://www.it.uc3m.es/berto/>



Tecnologías de Distribución de
Contenidos - UC3M



1

Motivación

- IDC: servicios de análisis de mercado y asesoría a empresas sobre TIC
 - Año 2004: 2.300 millones \$ gastados en software para servicios Web
 - Se esperaba una inversión de 15.000 millones de \$ para 2009
 - Fuente: <http://www.internetnews.com/dev-news/article.php/3520271>
- Usuarios y/o comercializadores: SAP, Oracle, IBM, Microsoft, BEA Systems, Sun Microsystems

Tecnologías de Distribución de
Contenidos - UC3M

2

Servicios Web: ¿Qué son?

- Los servicios Web son un mecanismo de **comunicación distribuida** que permiten que las aplicaciones compartan información y que además **invoquen funciones** de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas.

Servicios Web: ¿Qué son?

- Componentes que ejecutan procesos o funciones de negocio significativas...
- ... con una interfaz claramente definida y accesible a través de Internet...
 - Normalmente usando protocolo HTTP
- ... basada en el intercambio de documentos electrónicos en formato XML...
 - MIME para datos binarios
- ... y que pueden ser combinados entre sí.

Servicios Web ¿Para qué sirven?

- Permiten que varias aplicaciones
 - Compartan información
 - Invoquen funciones de otras aplicaciones independientemente de:
 - Cómo hayan sido creadas (lenguaje de programación)
 - Cómo se ejecutan (sistema operativo y plataforma)
 - Dispositivos utilizados para acceder a ellas
- Crean una especie de WWW paralela para máquinas
 - WWW humana (personas accediendo a páginas Web)
 - WWW cibernética (aplicaciones accediendo a servicios Web)

Pila de especificaciones de servicios Web

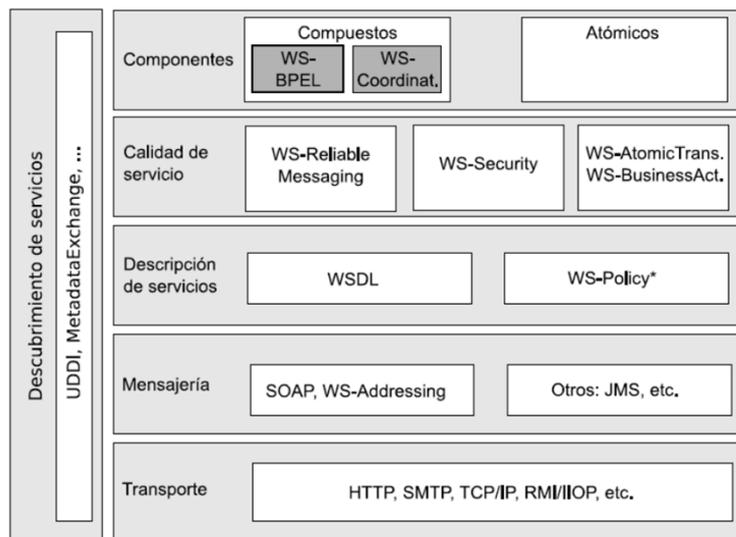
Introducción

- Los servicios Web se basan en un conjunto de especificaciones coherentes y complementarias entre sí
- Unas especificaciones se apoyan en otras para proporcionar la funcionalidad deseada
- Cada especificación resuelve una posible necesidad en el uso de servicios Web

Tecnologías de Distribución de
Contenidos - UC3M

7

Pila de especificaciones de servicios Web



Fuente: Web Services Platform Architecture

8

Capa de transporte

- Protocolos sobre los cuales se realiza el envío de mensajes
- Servicios Web no están ligados a un único protocolo en esta capa:
 - En principio, cualquier protocolo es válido
 - Los más utilizados actualmente para servicios Web son HTTP y HTTPS

Capa de mensajería

- Protocolos específicos para el envío de mensajes
- Los servicios Web no están ligados a un único protocolo
- Las principales especificaciones en esta capa son:
 - **SOAP**
 - Formato de mensajes, modelo procesamiento de mensajes, etc.
 - **WS-Addressing**
 - Mecanismo independiente de la capa de transporte para identificar a los emisores y receptores de mensajes

Capa de descripción de servicios

- Mecanismos para proporcionar descripciones funcionales y no funcionales de los servicios
- Las principales especificaciones en esta capa son:
 - **WSDL** (*Web Service Description Language*)
 - **WS-Policy***: conjunto de especificaciones para la descripción de propiedades no funcionales asociadas a servicios
 - Calidad de servicio (seguridad, fiabilidad, etc)

Capa de calidad de servicio

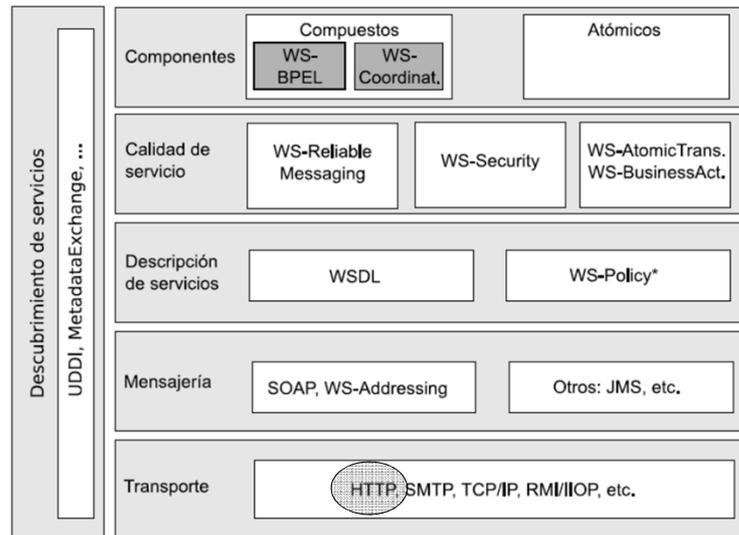
- Algunas de las especificaciones de esta capa son:
 - **WS-ReliableMessaging**: proporciona envío fiable de mensajes
 - **WS-Security**: conjunto de especificaciones acerca de cómo proporcionar distintas propiedades de seguridad en entornos basados servicios Web
 - **WS-AtomicTransaction** y **WS-BusinessActivity**: protocolos que permiten proporcionar soporte a transacciones en entornos basados en servicios Web

Capa de componentes

- La principal especificación en esta capa es:
 - **WS-BPEL**, que permite la definición, en forma de procesos, de nuevos servicios a partir de otros ya existentes. La ejecución de las definiciones de procesos se puede automatizar

HTTP

Pila de especificaciones de Servicios Web



Fuente: Web Services Platform Architecture

15

HTTP

- *HyperText Transfer Protocol*
- Desarrollado por W3C e IETF
 - Estándar actual: HTTP 1.1, RFC 2616 (1999)
- Protocolo solicitud/respuesta
 - Cliente solicitud a servidor y este responde
- Basado en texto
- Carente de estado
 - Galletitas (*cookies*) para control sesión
 - Datos del servidor que se guardan en el cliente

Tecnologías de Distribución de
Contenidos - UC3M

16

HTTP: Solicitud



Tecnologías de Distribución de
Contenidos - UC3M

17

HTTP: Solicitud

- Métodos
 - 8 en HTTP 1.1
 - Los más importantes:
 - GET: Solicitar un recurso al servidor
 - POST: Envío de datos al servidor (ej.: formularios)
 - HEAD: Lectura de las cabeceras (mismo que GET pero no devuelve el contenido del recurso)

Tecnologías de Distribución de
Contenidos - UC3M

18

HTTP: Respuesta

- Protocolo y versión
- Código de respuesta

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
Contenido del recurso (si procede)
```

■ Cabeceras

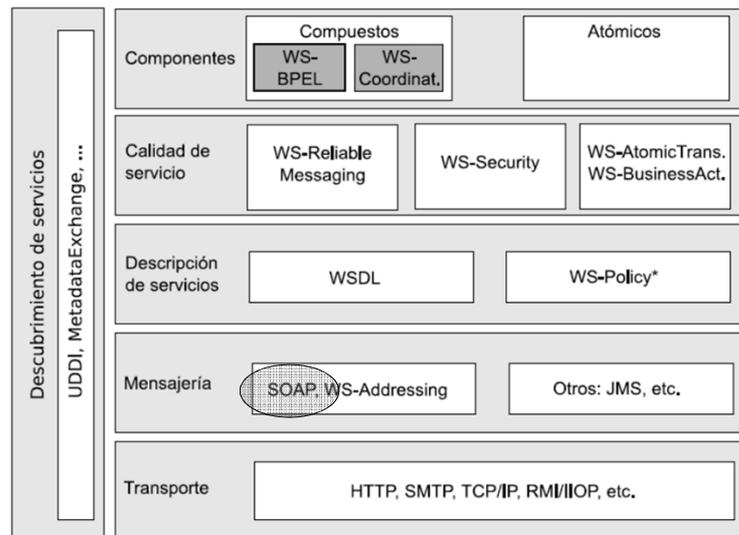
■ Contenido

HTTP: Respuesta

- Códigos de respuesta:
 - 2XX → Éxito
 - Ej.: 200 *OK*
 - 3XX → Redirecciones
 - Ej.: 301 *Moved Permanently*
 - 4XX → Errores provocados por el cliente
 - Ej.: 404 *Not Found*
 - 5XX → Errores provocados por el servidor
 - Ej.: 500 *Internal Server Error*

SOAP

Pila de especificaciones de Servicios Web



Fuente: Web Services Platform Architecture

Introducción

- SOAP es el protocolo de envío de mensajes fundamental en servicios Web
 - Originalmente: *Simple Object Access Protocol*
 - Actualmente SOAP 1.2
 - Recomendación W3C abril 2007
 - Ya no es acrónimo: nada que ver con objetos
 - Algunos leen *SOA protocol*; SOA = *Service Oriented Architecture*

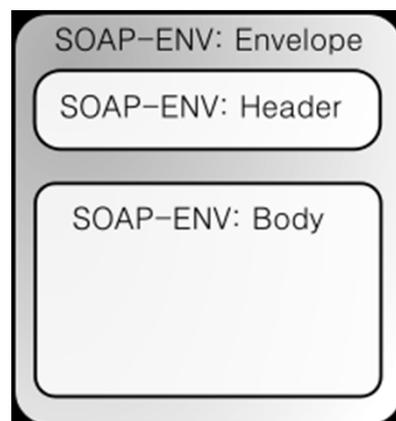
Introducción

- Proporciona cuatro funciones principales:
 - Estructura estandarizada de mensajes basada en XML
 - Modelo de procesamiento que describe cómo un servicio debe procesar los mensajes
 - Un mecanismo para ligar mensajes SOAP a distintos protocolos de comunicación
 - Un mecanismo para el envío en mensajes SOAP de datos no codificados en XML

Nodos SOAP

- Entidades capaces de hablar el protocolo SOAP para transmitir, recibir, procesar o reenviar mensajes SOAP:
 - Nodos SOAP receptores
 - Nodos SOAP remitentes
 - Nodos SOAP intermediarios
- SOAP message path
 - Conjunto de nodos SOAP por los que pasa un mensaje

Mensajes SOAP



Fuente: Wikipedia (Junio 2012)

Mensaje SOAP: Ejemplo

```
<?xml version="1.0" ?>
<e nv:Envelope xmlns:env="http://www.w3.org/2001/09/soap-envelope">
  <env:Header>
    <ca:CacheControl xmlns:ca="http://example.org/2001/06/cache">
      <ca:CacheKey>ABC-1234</ca:CacheKey>
      <ca:Expires>2001-03-09T08:00:00Z</ca:Expires>
    </ca:CacheControl>
  </env:Header>
  <env:Body>
    <c:CatalogPriceResponse xmlns:c="http://example.org/2001/06/catalog">
      <c:PartNumber>ABC-1234</c:PartNumber>
      <c:PartPrice c:currency="USD">120.37</c:PartPrice>
    </c:CatalogPriceResponse>
  </env:Body>
</env:Envelope>
```

Fuente: <http://www.w3.org/TR/xmlp-scenarios/>

Bloques de cabecera

- Un mensaje SOAP puede contener opcionalmente una cabecera, compuesta por uno o más bloques de cabecera
- Contienen información útil para el envío y procesado de mensajes:
 - Identificador y número de secuencia de mensaje (fiabilidad a nivel de aplicación)
 - A quién contestar el mensaje (Ej.: servicios asíncronos)
 - Seguridad (Ej.: Firma del contenido)
- Atributos *role* (a quién va dirigida la cabecera) y *mustUnderstand* (procesar la cabecera es o no obligatorio)

Cuerpo

- Un mensaje SOAP debe contener un cuerpo
- El cuerpo del mensaje transmite información específica de la aplicación desde el primer nodo remitente al nodo receptor final
- No es procesado por los nodos intermedios, sólo por el destinatario
- En caso de error, la información sobre el mismo se codifica en el cuerpo con un elemento *Fault*

Patrones de intercambio de mensajes

- En principio, cada mensaje no guarda correlación con otros mensajes
- En interacciones que requieren más de un mensaje, es útil representar explícitamente la correlación entre los mismos
- Algunos patrones de intercambio de mensajes (MEPs):
 - MEP de petición/respuesta
 - MEP de conversación de larga duración

Ligaduras SOAP

- Un mensaje SOAP se transmite sobre otros protocolos de comunicación subyacentes
- SOAP no limita qué protocolos se pueden usar para el envío de mensajes
- Algunos ejemplos:
 - HTTP / HTTPS
 - Correo electrónico (SMTP)
 - MOM: IBM WebSphereMQ, etc.

Mensaje SOAP ejemplo

```
POST /axis/services/SQAPS HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, [...]
User-Agent: Axis/1.3
Host: bach.gast.it.uc3m.es:8180
Cache-Control: no-cache
Content-Length: 436

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <findConcepts xmlns="http://ws.SQAPS">
      <query>dijkstra</query>
      <authToken>...</authToken>
    </findConcepts>
  </soapenv:Body>
</soapenv:Envelope>
```

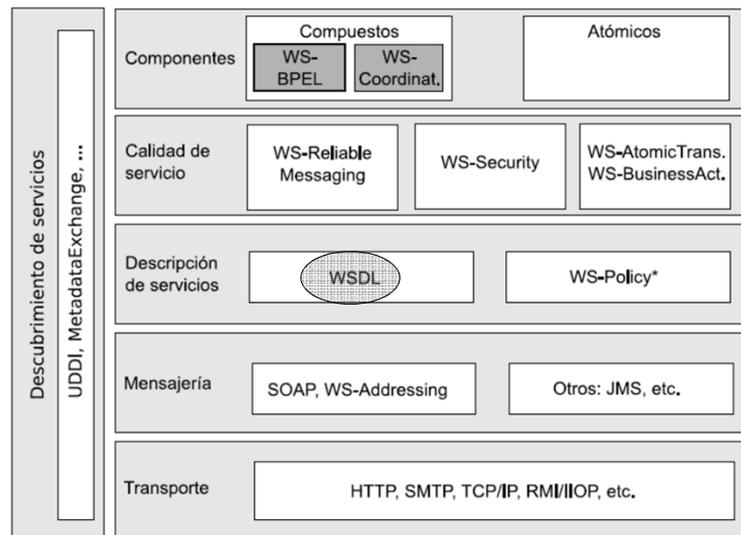
WSDL

Web Service Description Language

Tecnologías de Distribución de
Contenidos - UC3M

33

Pila de especificaciones de Servicios Web



Fuente: Web Services Platform Architecture

34

Introducción

- Vocabulario XML para describir la interfaz de un servicio Web:
 - Qué operaciones proporciona un servicio Web y qué tipos de mensajes de entrada y salida admite cada operación
 - Sobre qué protocolo se envían los mensajes y cómo se codifican los mismos
 - Dónde se accede al servicio
- Veremos la versión WSDL 1.1
 - Ya hay nuevas (WSDL 2.0)
 - Las dos recomendaciones del W3C

Estructura de un documento WSDL

- Definiciones de tipos de datos
- Definiciones de tipos de mensajes
- Definiciones de tipos de puertos
- Definiciones de ligaduras
- Definiciones de servicios

Estructura de un documento WSDL

```
<definitions>
  <!-- definiciones de tipos de datos -->
  <types>...</types>

  <!-- definiciones de tipos de mensajes -->
  <message name="...">...</message>

  <!-- definiciones de tipos de puerto -->
  <portType name="...">...</portType>

  <!-- definiciones de ligaduras -->
  <binding name="...">...</binding>

  <!-- definiciones de servicios -->
  <service name="...">...</service>
</definitions>
```

Definiciones de tipos de datos

- Todos los tipos de datos utilizados en los mensajes deben ser declarados en esta sección
- Se puede utilizar cualquier sistema de definición de tipos de datos, aunque normalmente se utiliza XML Schema

Ejemplo: definiciones de tipos de datos

```
<wsdl:types>
  <element name="login">
    <complexType>
      <sequence>
        <element name="login" type="xsd:string"/>
        <element name="password" type="xsd:string"/>
      </sequence>
    </complexType>
  </element>
  <element name="loginResponse">
    <complexType>
      <sequence>
        <element name="loginReturn" type="xsd:string"/>
      </sequence>
    </complexType>
  </element>
</wsdl:types>
```

Tipos de mensajes

- Todos los mensajes de entrada y salida de operaciones deben declararse en esta sección
- A cada mensaje se le asocia un nombre
- Un mensaje se compone de una o más partes, y para cada parte se indica su nombre y su tipo de datos

Ejemplo: tipos de mensajes

```
<wsdl:message name="loginRequest">
  <wsdl:part element="tns1:login" name="parameters"/>
</wsdl:message>

<wsdl:message name="findConceptsRequest">
  <wsdl:part element="tns1:findConcepts" name="parameters"/>
</wsdl:message>

<wsdl:message name="SQAPSWebServiceException">
  <wsdl:part element="impl:fault" name="fault"/>
</wsdl:message>
```

Tipos de puertos y operaciones (I)

- Una operación es una agrupación lógica de un conjunto de mensajes intercambiados
- Un tipo de puerto es una agrupación lógica de operaciones relacionadas entre sí

Tipos de puertos y operaciones (II)

- Una operación puede definir, como mucho, un tipo de mensaje de entrada y otro de salida:
 - En un sentido: llega un mensaje al servicio, y éste no responde (*input*)
 - Petición/respuesta iniciada por el cliente: llega un mensaje al servicio, y éste responde con otro mensaje (*input/output*)
 - Petición/respuesta iniciada por el servicio: el servicio envía un mensaje al cliente, y éste responde (*output/input*)
 - Notificación: el servicio envía un mensaje al cliente y éste no responde (*output*)

Ejemplo: tipos de puertos y operaciones

```
<wsdl:portType name="SQAPSWebService">
  <wsdl:operation name="query">
    <wsdl:input message="impl:queryRequest" name="queryRequest"/>
    <wsdl:output message="impl:queryResponse" name="queryResponse"/>
    <wsdl:fault message="impl:SQAPSWebServiceException"
      name="SQAPSWebServiceException"/>
  </wsdl:operation>
  <wsdl:operation name="login">
    <wsdl:input message="impl:loginRequest" name="loginRequest"/>
    <wsdl:output message="impl:loginResponse" name="loginResponse"/>
    <wsdl:fault message="impl:SQAPSWebServiceException"
      name="SQAPSWebServiceException"/>
  </wsdl:operation>
</wsdl:portType>
```

Ligaduras

- Una ligadura especifica cómo formatear los mensajes para interactuar con un servicio dado
- A cada tipo de puerto se le asigna una o más ligaduras
- WSDL es extensible en cuanto a la definición de nuevos tipos de ligaduras. Algunos predefinidos:
 - Ligadura SOAP: especifica cómo codificar los mensajes en mensajes SOAP
 - Ligadura HTTP/MIME: especifica cómo codificar los mensajes directamente en HTTP con MIME (sin SOAP)

Ejemplo: ligaduras

```
<wsdl:binding name="SQAPSSoapBinding" type="impl:SQAPSWebService">
  <wsdlsoap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="login">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="loginRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="loginResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="SQAPSWebServiceException">
      <wsdlsoap:fault name="SQAPSWebServiceException" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  (...)
</wsdl:binding>
```

Servicios

- Un servicio representa un tipo de puerto, con una ligadura concreta, accesible a través de una dirección dada
- Puede haber más de un servicio asociado al mismo tipo de puerto y misma ligadura, cambiando únicamente su dirección

Ejemplo: servicios

```
<wsdl:service name="SQAPSWebServiceService">  
  <wsdl:port binding="impl:SQAPSSoapBinding" name="SQAPS">  
    <wsdlsoap:address location=  
      "http://bach.gast.it.uc3m.es:8180/axis/services/SQAPS"/>  
  </wsdl:port>  
</wsdl:service>
```

Referencias

■ Especificaciones:

- XML 1.0
 - <http://www.w3.org/TR/xml/>
- XML Schema (Part 0: Primer)
 - <http://www.w3.org/TR/xmlschema-0/>
- SOAP Versión 1.2 (Part 0: Primer)
 - <http://www.w3.org/TR/soap12-part0/>
- WSDL Version 2.0 (Part 0: Primer)
 - <http://www.w3.org/TR/wsdl20-primer/>
- UDDI Version 3.0.2
 - http://www.uddi.org/pubs/uddi_v3.htm



Bibliografía

■ Web Services Platform Architecture

- Sanjiva Weerawarana, Francisco Curbera, Frank Leymann, Tony Storey, Donald F. Ferguson.
- Prentice Hall, 2005. ISBN: 978-0131488748

■ Professional XML Web services

- Vivek Chopra, Zaev Zoran, Gary Damschen, Chris Dix, Patrick Cauldwell, Rajesh Chawla, Kristy Saunders, Glenn Olander, Francis Norton, Tony Hong, Uche Ogbuji, Mark A. Richman
- Wrox Press, 2001. ISBN: 978-1861005090

