Ingeniería de Aplicaciones para la Web Semántica

Clase 02

SOAP y XML

Mg. A. G. Stankevicius

Segundo Cuatrimestre

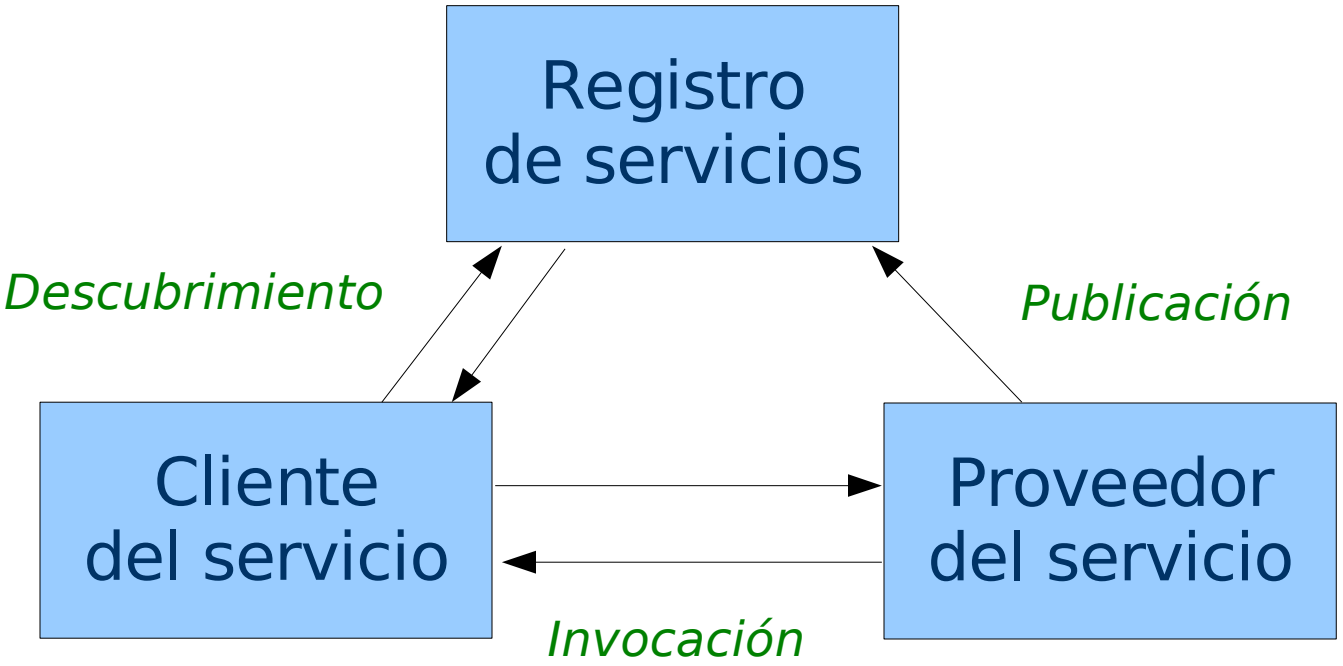
2005

Copyright

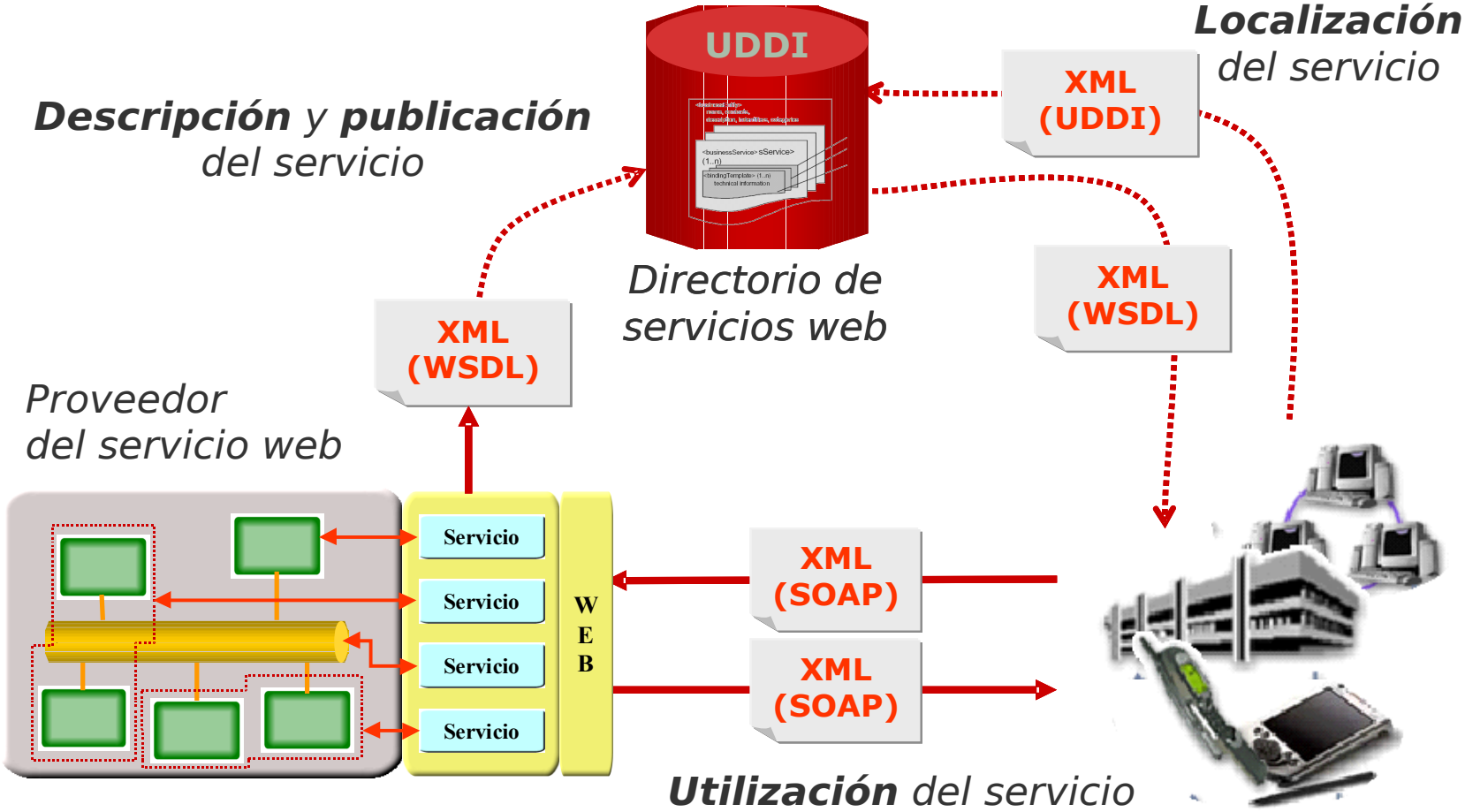
Contenidos

- El rol de SOAP y de XML en los SW.
- Componentes de un mensaje SOAP.
- Introducción a XML.
- Estructura de un documento XML.
- Espacios de nombres XML.
- Encabezamiento y cuerpo de mensajes.
- Manejo de fallas en SOAP.
- Modelos de intercambio de mensajes.
- SOAP bindings.

Anatomía de un servicio web



El rol de SOAP y de XML en los servicios web



¿Qué es SOAP?

- En pocas palabras:
  - SOAP es un protocolo para acceder a los servicios web.
- Definición un poco más extensa:
  - SOAP es un protocolo sencillo basado en XML, diseñado con el objeto de permitir que las aplicaciones puedan intercambiar información empleado HTTP.

Características

- SOAP...
  - ...es un protocolo de comunicación.
  - ...permite que las aplicaciones puedan comunicarse entre si.
  - ...es un formato para estructurar mensajes.
  - ...está diseñado para funcionar sobre internet.
  - ...se basa en XML.
  - ...puede atravezar firewalls.

Características

- SOAP...
  - ...próximamente será un estandar de la W3C.
  - ...es independiente de las arquitecturas en las que se lo utilice.
  - ...es independiente de los lenguajes de programación que hagan uso de él.

Estructura de un mensaje SOAP

- Un mensaje SOAP es un documento XML conteniendo los siguientes elementos:
  - (requerido) Un Envelope que identifica al documento XML como un mensaje SOAP.
  - (opcional) Un Header que contiene el encabezamiento del mensaje.
  - (requerido) Un Body que contiene la información comunicada en el mensaje.
  - (opcional) El elemento Fault proveyendo información sobre los errores encontrados durante el procesamiento del mensaje.

A tener en cuenta

- Consideraciones a tener en cuenta al construir mensajes SOAP:
  - Deben estar codificados usando XML.
  - Deben hacer uso del espacio de nombres *SOAP Envelope*.
  - Deben hacer uso del espacio de nombre *SOAP Encoding*.
  - No deben hacer referencia a DTD alguno.
  - No deben hacer referencia a directiva de procesamiento alguna.

Esqueleto de un mensaje SOAP codificado en XML

```
<?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:Header>
  ...
</soap:Header>

<soap:Body>
  ...
  <soap:Fault>
    ...
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```

Introducción a XML

- EXtensible Markup Language.
- ¿Qué es XML?
  - Es un lenguaje de marcado similar a HTML.
  - Fue diseñado para poder describir información.
  - Los tags no están predefinidos. Cada usuario debe definir sus propios tags.
  - Es decir, sería aún más correcto hablar de un metalenguaje de marcado.
  - Es un estandar soportado por la W3C.

HTML vs. XML

- Consideremos el siguiente fragmento válido de código HTML:

```
<h2>Nonmonotonic Reasoning:
Context-Dependent Reasoning</h2>
<i>by <b>V. Marek</b> and
<b>M. Truszczynski</b></i><br>
Springer 1993<br>
ISBN 0387976892
```

HTML vs. XML

- Ese mismo fragmento reformulado en XML puede resultar:

```
<book>
   <title>Nonmonotonic Reasoning:
    Context-Dependent Reasoning</title>
   <author>V. Marek</author>
   <author>M. Truszczynski</author>
   <publisher>Springer</publisher>
   <year>1993</year>
   <ISBN>0387976892</ISBN>
</book>
```

¿HTML ó XML?

- ¿Cuál conviene usar? ¡Los dos!
- HTML describe la forma en la cual visualizar la información contenida.
  - Meta: cómo se ve.
- XML describe la estructura de la información contenida.
  - Meta: qué es.
- Es posible combinar ambos lenguajes:
  - CSS (Cascading Stylesheet)
  - XSL (EXtensible Stylesheet Language)

El lenguaje XML

- Todo documento XML se compone de las siguientes secciones:
  - Un prólogo.
  - Un conjunto de elementos.
  - Un epílogo.
- El prólogo consiste de dos partes:
  - Una declaración XML.
  - Una refencia opcional a documentos externos describiendo las convenciones de estructura adoptadas.

Elementos XML

- Los elementos XML son los “objetos” que el documento XML referencia:
  - Libros, autores, ISBNs, etc.
- Todo elemento XML debe contener:
  - Un tag inicial (opening tag).
  - El contenido propiamente dicho.
  - Un tag final (closing tag).
- Por ejemplo:

```
<gangster>Carlos S. Mendez</gangster>
```

Elementos XML

- Se puede usar prácticamente cualquier identificador como tag.
- El primer carácter debe ser una letra, un guión bajo o un dos puntos.
- Ningún nombre puede comenzar con la cadena “XML”, en ninguna de sus posibles mayusculizaciones.
  - XML es inválido,
  - xML también,
  - etc.

Contenido de los Elementos XML

- Un elemento XML puede contener texto, otros elementos XML, o bien nada:

```
<gangster>
    <nombre>Carlos S. Mendez</nombre>
    <edad>92</edad>
</gangster>
```

- Los elementos XML sin contenido puede abreviarse de la siguiente forma:

`<edad/>` equivale a `<edad></edad>`

Atributos XML

- Un elemento XML vacío no necesariamente carece de significado:
  - Puede contar con algunas propiedades productos de sus atributos.
- Un atributo es un par nombre-valor apareciendo en el tag inicial de un dado elemento XML.

```
<gangster estado="prófugo">
   Carlos S. Mendez
</gangster>
```

Atributos vs. Elementos XML

- Los atributos pueden ser fácilmente reemplazados por elementos.
- Cuándo conviene usar uno y cuándo conviene usar otro termina siendo una cuestión de preferencia personal.
- No obstante, los atributos no pueden anidarse.
- En contraste, los elementos son fácilmente extendibles.

Otros componentes de un documento XML

- Comentarios:
  - Definen porciones del documento que deberán ser ignoradas al procesarlo:

```
<!-- esto es un comentario -->
```

- Directivas de procesamiento:
  - Definen otros documentos y procedimientos adicionales a tener en cuenta:

```
<?stylesheet
   type="text/css"
      href="mystyle.css"?>
```

Documentos XML bien formados

- Deben ser sintácticamente correctos:
  - Por fuera debe haber un único elemento XML (denominado elemento raíz).
  - Para cada elemento, el tag de inicio se debe corresponder con el tag de finalización.
  - Los tags no se puede solapar:

    ```
    <barrio><calle>Donado</barrio><calle>
    ```

  - Los atributos dentro de un mismo elemento deben tener nombre únicos.
  - Los elementos y los tags deben respetar las reglas antes especificadas.

Estructurando a los documentos XML

- En los documentos XML es posible dejar constancia de su propia estructura.
- Para lograrlo, se deben definir todos los elementos y nombres de atributos que serán usados.
  - Indicando los valores que un cierto atributo puede tomar.
  - Indicando cuáles elementos pueden aparecer anidados dentro de otros elementos.

Estructurando a los documentos XML

- Un documento XML se dice válido:
  - Si se trata de un documento bien formado,
  - Y además, respecta la información acerca de su estructura suministrada.
- Existen dos formas de dotar de estructura a un documento XML:
  - Apelando a las DTD.
  - Apelando a los Esquemas XML.

Espacios de nombres XML

- Un documento XML puede usar más de un DTD o Esquema XML, lo que puede conducir a conflictos de nombres.
- Los tag elegidos pueden entrar en conflicto con otros tags.

```
<table> ... </table>
```

- La solución es muy simple: emplear distintos prefijos para cada nombre que genere conflicto:

```
prefijo:nombre
```

Declaración de un espacio de nombres XML

- Los espacios de nombres se declaran dentro de elementos XML.
- El alcance abarca el elemento en el cual se define y todos sus hijos, sean estos elementos o atributos.
- La declaración guarda la forma:

```
xmlns:prefijo="ubicación"
```

- La declaración hace referencia a la ubibación del DTD o esquema XML en cuestión.

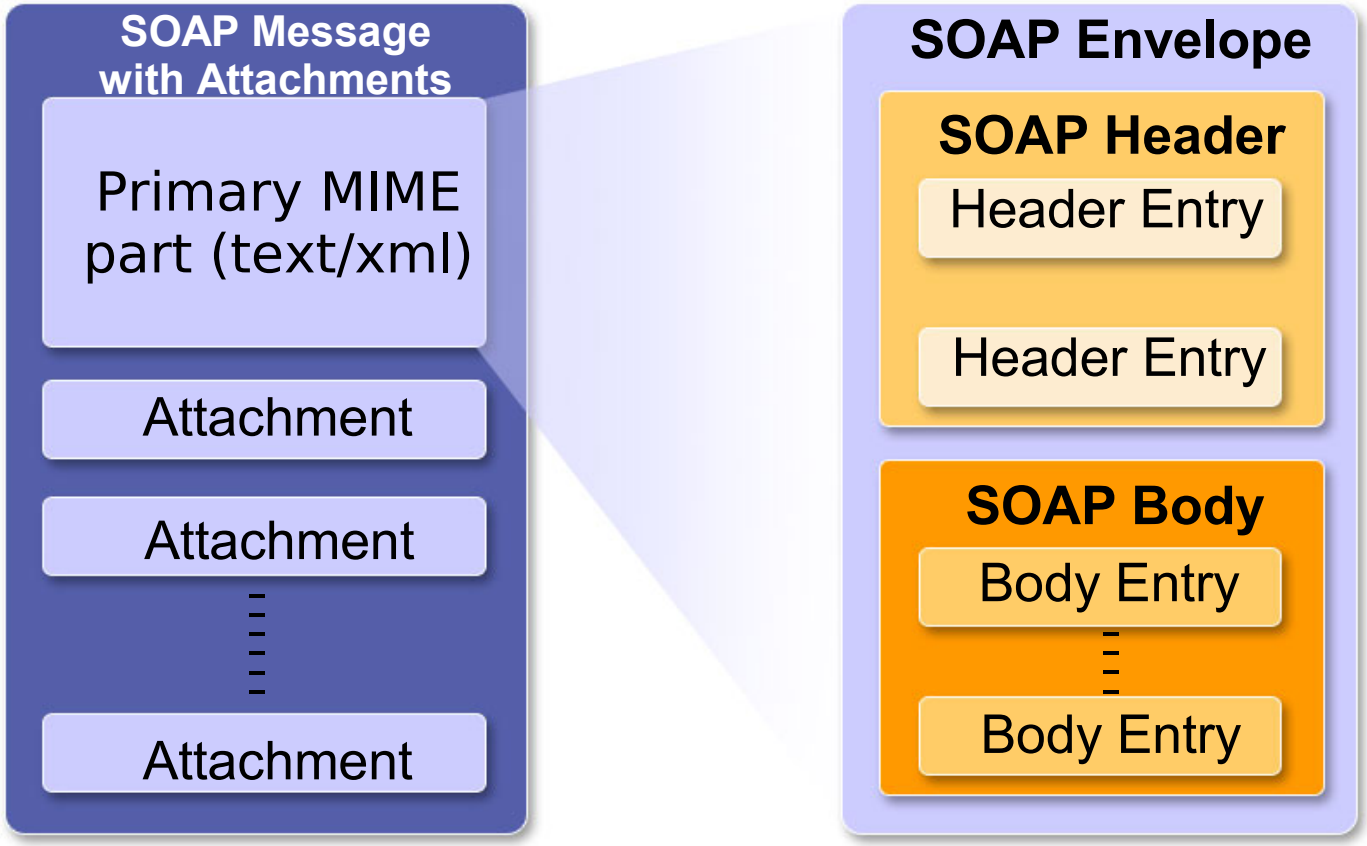
Un ejemplo en concreto

```
<instructors
   xmlns:uns="http://www.uns.edu.ar/empDTD"
   xmlns:uba="http://www.uba.ar/empDTD"
<uns:faculty
   uns:title="Asistente de Docencia"
   uns:name="Juan Perez"
   uns:department="DCIC"/>
<uba:academicStaff
   uba:title="Jefe de Trabajos Prácticos"
   uba:name="Pepa Flores"
   uba:school="Ciencias Exáctas"/>
</instructors>
```

Resumen

- XML es un metalenguaje que permite a los usuarios definir un lenguaje de marcado para la información.
- XML separa contenido y estructura de la presentación visual.
- XML es el estandar de facto para representar e intercambiar información con estructura en la web.
- Entendiendo un poco de XML, podemos retomar nuestro análisis de SOAP.

Estructura de un mensaje SOAP



SOAP Envelope

- El SOAP Envelope caracteriza al documento XML como un mensaje SOAP.

```
<?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">
  <!-- resto del mensaje -->
</soap:Envelope>
```

SOAP Header

- El SOAP Header contiene información específica del mensaje SOAP:

```
<soap:Header>
<m:Transaction
   xmlns:m="http://cs.uns.edu.ar/trans/"
   soap:mustUnderstand="1">
      234
</m:Transaction>
</soap:Header>
```

El atributo mustUnderstand

- Si el atributo *mustUnderstad* aparece en alguno de los elementos del encabezamiento, el receptor de ese mensaje debe asegurarse tener en cuenta al citado elemento.
- Si el receptor no comprende el rol del elemento conteniendo este atributo, deberá fallar su procesamiento del encabezamiento.

SOAP Body

- El SOAP Body contiene la información que se desea comunicar con el mensaje SOAP en cuestión.
  - Puede tratarse de la invocación a un servicio web.
  - Puede contener el resultado producto de esa invocación, si la misma fue exitosa.
  - O bien, puede contener información acerca de por qué no pudo cumplir con la invocación solicitada.

El elemento Fault

- Este elemento permite que un mensaje SOAP comunique que se observó un error en el procesamiento de otros mensajes.
- Este elemento puede aparece sólo una única vez dentro de un determinado mensaje.

Elementos anidados dentro del elemento Fault

- El elemento Fault contiene diversos elementos anidados:
  - El elemento `<faultcode>` describiendo la falla observada.
  - El elemento `<faultstring>` conteniendo una explicación entendibles por humanos del error observado.
  - El elemento `<faultfactor>` indicando quién observó la falla en cuestión.
  - El elemento `<detail>` conteniendo información relacionado al SOAP Body.

Códigos de falla SOAP

- Existen cuatro posibles códigos de falla:
  - VersionMismatch: denota que el SOAP Envelope hace referencia a un espacio de nombres inválido.
  - MustUnderstand: indica que algún elemento del SOAP Header que debía ser entendido, no pudo ser comprendido.
  - Client: El mensaje recibido tiene algún problema.
  - Server: El servidor no pudo procesar el mensaje enviado.

Modelos de Comunicación

- SOAP es un protocolo de comunicación simple, sin estado y de sentido único.
- Los patrones de comunicación más elaborados son responsabilidad de la aplicación que haga uso del mismo.
- No obstante, SOAP soporta diversos patrones:
  - RPC (*se intercambian mensajes*).
  - De sentido único (*los mensajes solo van*).

Estilos de Mensajes

- Otro aspecto que suele crear confusión es el estilo de mensaje SOAP a ser usado en una dada comunicación.
- SOAP admite esencialmente dos estilos de codificación de la información contenida en el SOAP Body:
  - RPC (que nada tiene que ver con el modelo de comunicación de igual nombre).
  - Documento.

Codificación del Mensaje SOAP

- Un tercer aspecto que agrega incluso más confusión son las alternativas a la hora de codificar el propio mensaje SOAP.
- Esencialmente existen dos altenativas:
  - Literal (indica que se adopta un esquema XML definiendo los elementos del mensaje).
  - SOAP Encoded (indica que se adoptan las reglas sugeridas en la definición del estandar).

Combinaciones frecuentes

- Estas características suelen combinarse en general de la misma manera:
  - RPC/Encoded.
  - Document/Literal.
- No obstante, ambas combinaciones pueden ser adoptadas para implementar los distintos modelos de comunicación antes reseñados.
  - Por ejemplo, basar RPC en Document/Literal.

SOAP bindings

- SOAP es independiente del protocolo de bajo nivel en el cual se codifiquen sus mensajes.
- Usualmente se suele utilizar HTTP, pero otras alternativas también son posibles:
  - SOAP sobre SMTP.
  - SOAP sobre TCP.

SOAP sobre HTTP

- El mecamino de comunicación simil RPC se implementa usando al método POST:
  - Un mensaje SOAP es enviado junto con el HTTP POST y un mensaje SOAP es recibido como respuesta.
- El mecamino de comunicación de sentido único se implementa usando al método GET:
  - Un mensaje SOAP es recibido como respuesta al pedido efectuado mendiante el HTTP GET.

Ejemplos prácticos

- Acceso a la información contenida en las bases de datos de Amazon.com mediante servicios web:
  - http://www.amazon.com/webservices
  - http://www.awszone.com/
- Acceso a distintos servicios web a través de SOAP sobre HTTP:
  - http://www.soapclient.com