

Aplicaciones de Internet: Servicios Web

Grupo ARCOS

Desarrollo de Aplicaciones Distribuidas

Ingeniería Informática

Universidad Carlos III de Madrid



Contenidos

1. **Introducción a los servicios Web**
2. **Elementos en un Servicio Web:**
 1. XML
 2. SOAP
 3. WSDL
 4. UDDI
3. **Ejemplo de aplicación**
 - ▶ Bolsa (servicio público)

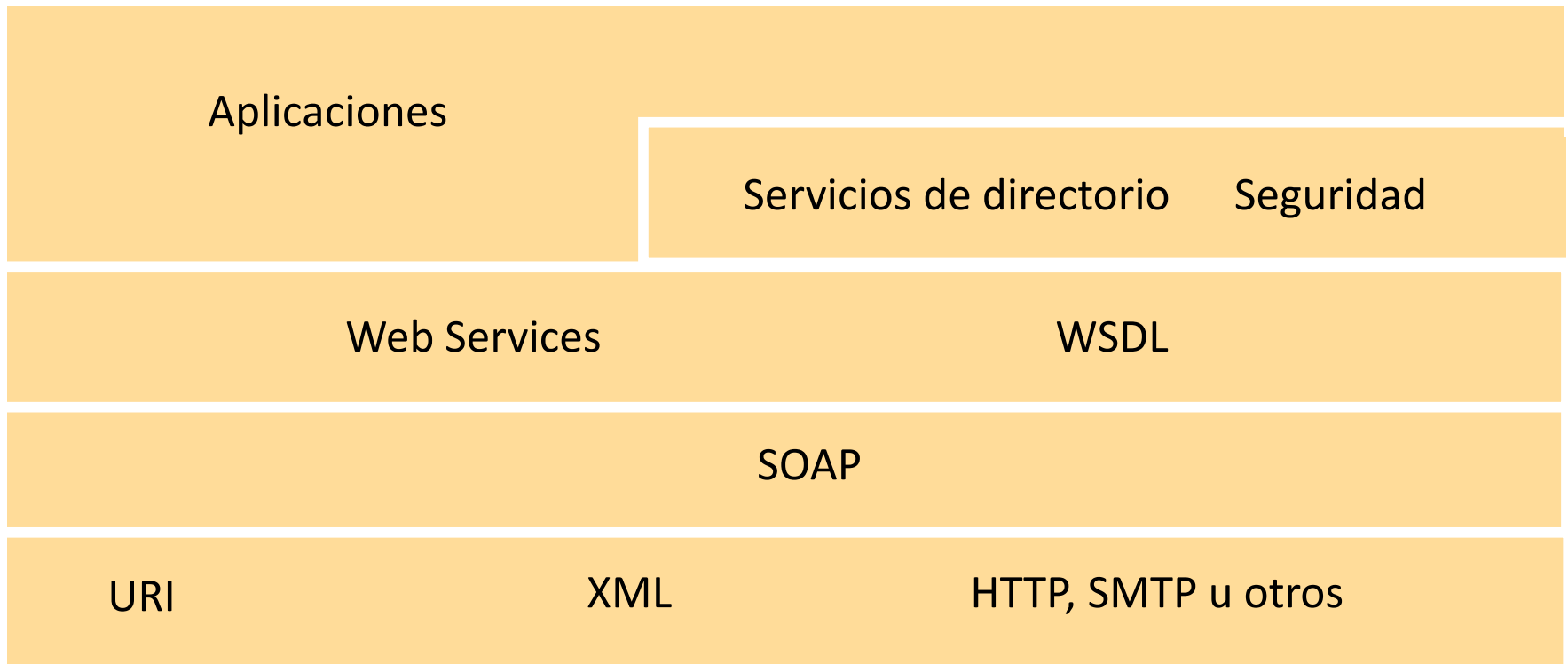
Contenidos

1. **Introducción a los servicios Web**
2. **Elementos en un Servicio Web:**
 1. XML
 2. SOAP
 3. WSDL
 4. UDDI
3. **Ejemplo de aplicación**
 - ▶ Bolsa (servicio público)

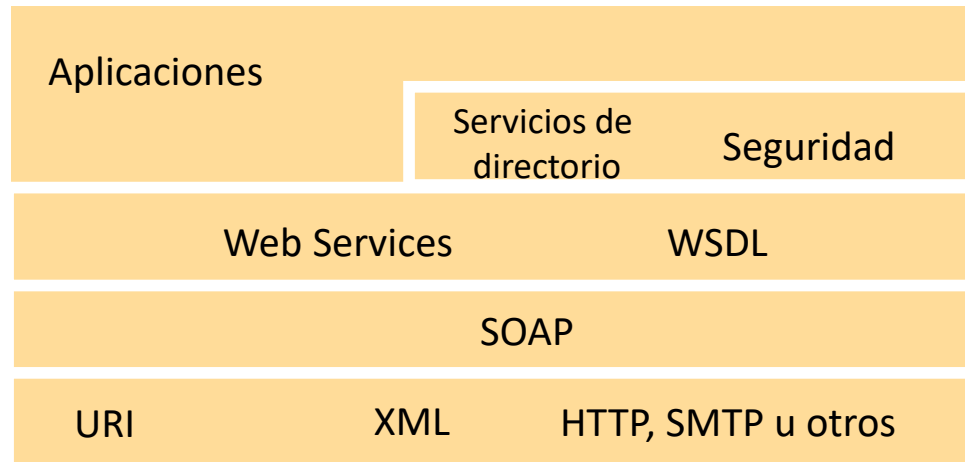
Servicios Web

- ▶ Adaptar el modelo de programación Web (débilmente acoplado) para su uso en aplicaciones no basadas en navegador.
- ▶ El objetivo es ofrecer una plataforma para construir aplicaciones distribuidas utilizando software:
 - ▶ Que ejecuta en diferentes sistemas operativos y arquitecturas
 - ▶ Escrito utilizando diferentes lenguajes y herramientas de programación
 - ▶ Desarrollado de forma independiente

Componentes e infraestructura

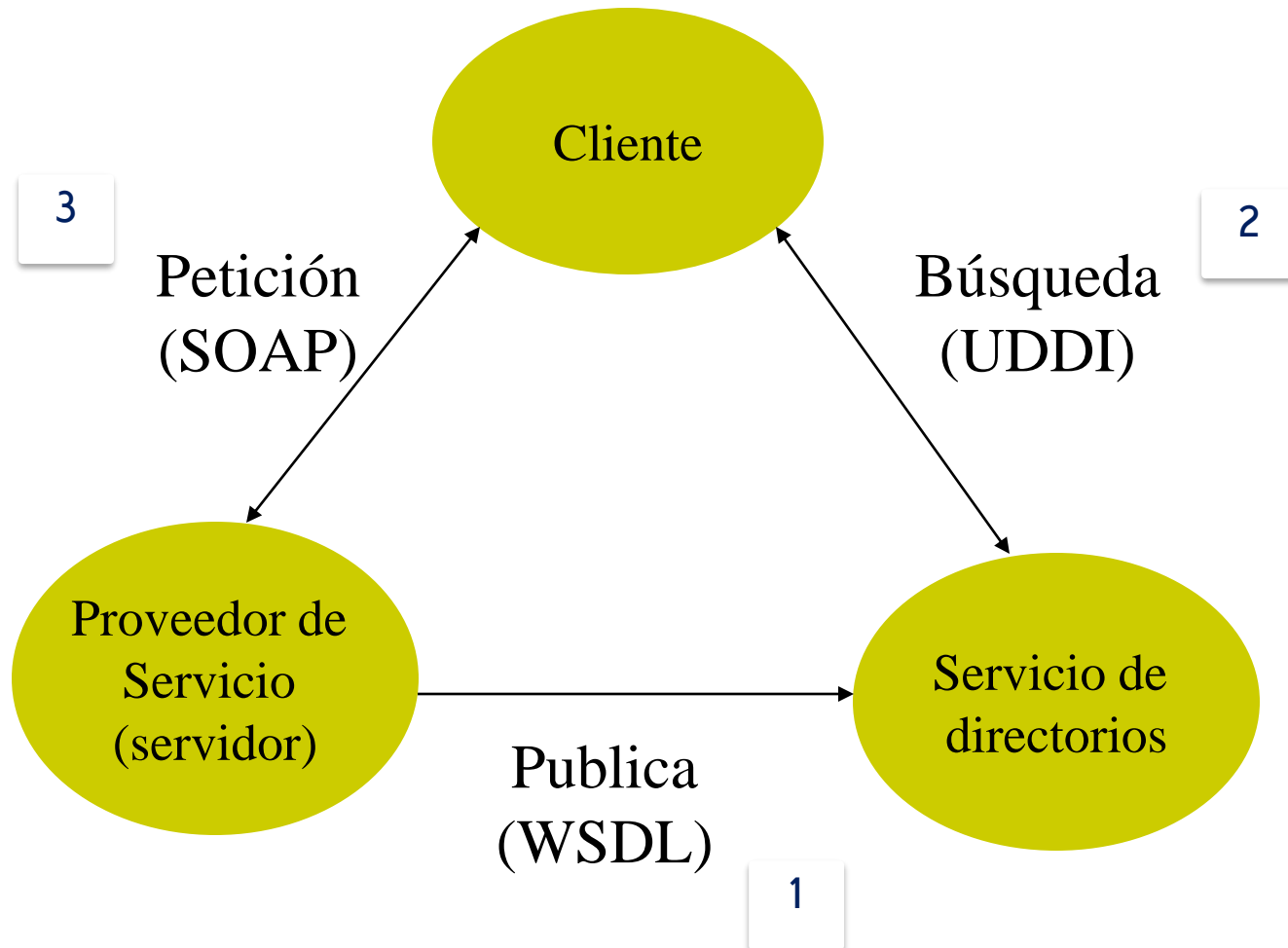


Componentes e infraestructura

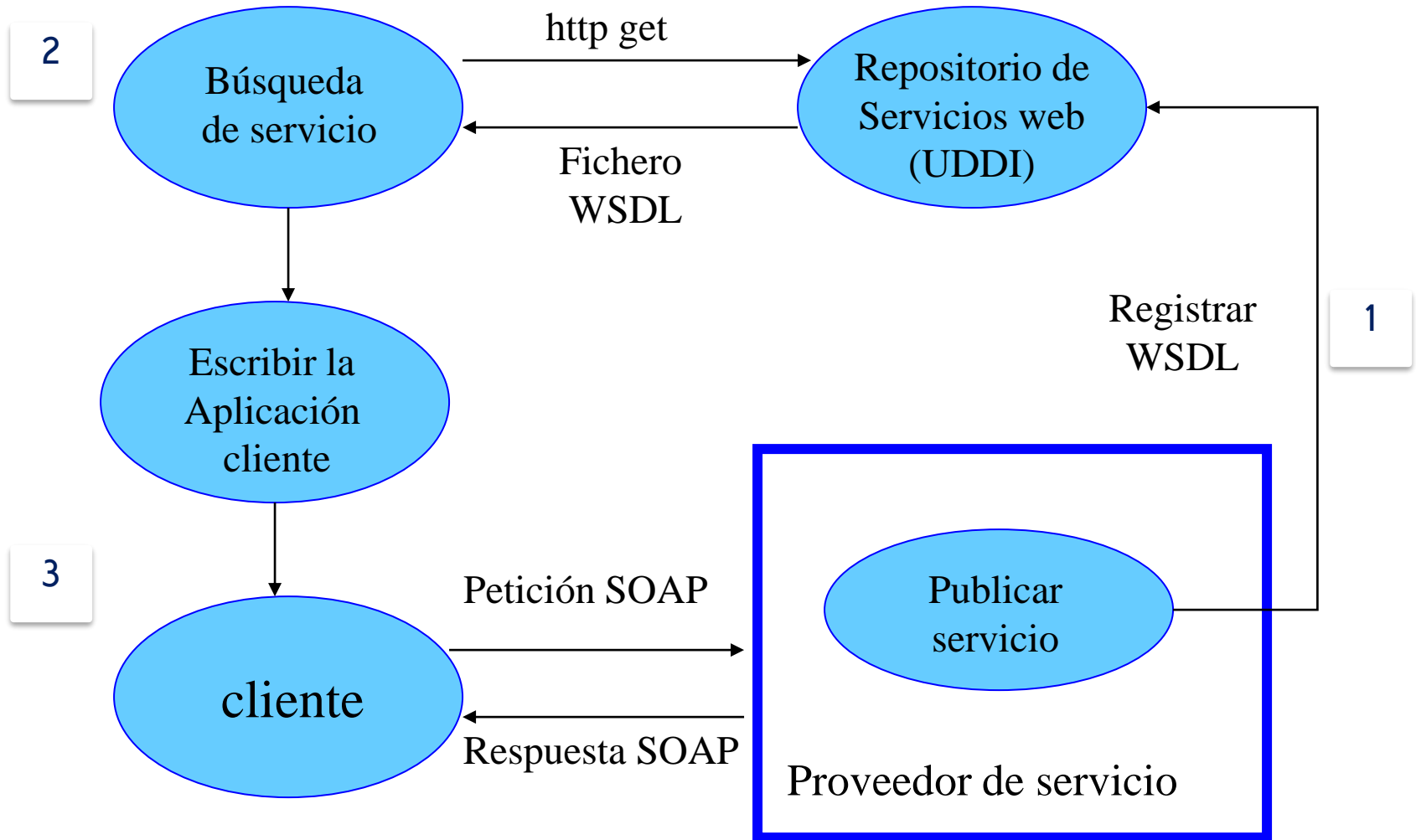


- ▶ HTTP: transporte utilizado
- ▶ SOAP: empaqueta la información y la transmite entre el cliente y el proveedor del servicio
- ▶ XML: describe la información, los mensajes
- ▶ UDDI: lista de servicios disponibles
- ▶ WSDL: descripción del servicio

Servicios Web y SOA

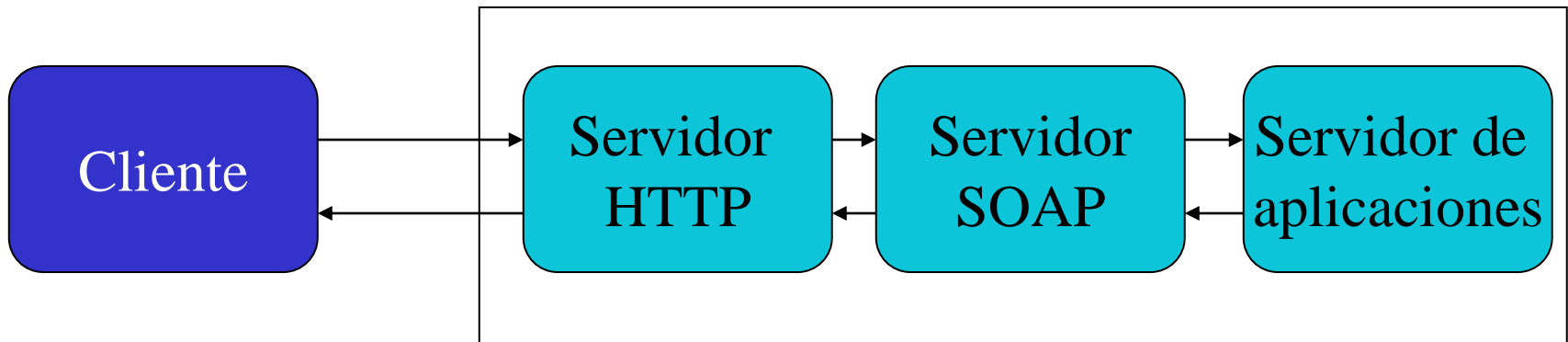


Escenario de uso



Ejemplo de implantación

Proveedor del servicio Web



Respecto a la implantación

- ▶ Una interfaz de servicio web consta de un **conjunto de operaciones utilizables por un cliente en Internet**
 - ▶ Los servicios Web no son específicos de HTTP.
- ▶ Las operaciones en un servicio Web **pueden ser ofrecidas por:**
 - ▶ Programas, objetos, bases de datos
- ▶ Un servicio Web **puede ser manejado por:**
 - ▶ Un servidor web tradicional
 - ▶ Un servidor independiente

Más información

- Sobre protocolos:
 - SOAP, XML, etc.: <http://www.w3.org/>
 - UDDI: <http://www.uddi.org/>
- Cursos sobre SOAP, WSDL y otras tecnologías web:
 - <http://www.w3schools.com/>
- Repositorios de servicios Web:
 - <http://www.xmethods.com/>
 - Google: <http://www.google.com/apis>
 - Para aplicaciones que realizan búsquedas en Internet

Entornos de desarrollo

- ▶ Número creciente de entornos de desarrollo
- ▶ Algunas implementaciones de interés:
 - ▶ gSOAP
 - ▶ .Net de Microsoft
 - ▶ *Web Services Project* de Apache
 - ▶ *Java Web Services Developer Pack*
 - ▶ *IBM WebSphere SDK for Web services (WSDK)*
 - ▶ WASP de Systinet
 - ▶ JOnAS
 - ▶ AXIS

Contenidos

1. Introducción a los servicios Web
2. **Elementos en un Servicio Web:**
 1. XML
 2. SOAP
 3. WSDL
 4. UDDI
3. Ejemplo de aplicación
 - ▶ Bolsa (servicio público)

XML

- *Extensible markup language*
 - Definido por W3C (<http://www.w3c.org>)
- XML es extensible, permite a los usuarios definir sus propias etiquetas (diferente a HTML)
- Componentes:
 - Elementos y atributos
 - `<tag attr=valor/>`
 - `<tag>valor</tag>`
 - Espacios de nombres
 - `xmlns="http://www.w3.org/1999/xhtml"`
 - Esquemas
 - Elementos y atributos que pueden aparecer en un documento

Ejemplo de XML

- Ej: *float ObtenerPrecio(string item);*

Petición:

```
<ObtenerPrecio>  
  <item>mesa</item>  
</ObtenerPrecio>
```

Respuesta:

```
<ObtenerPrecioResponse>  
  <Precio>134.5</Precio>  
</ObtenerPrecioResponse>
```

Ejemplo de XML

- Ej: *float ObtenerPrecio(string item);*

Petición:

```
<ObtenerPrecio>
  <item>mesa</item>
</ObtenerPrecio>
```

Respuesta:

```
<ObtenerPrecioResponse>
  <Precio>134.5</Precio>
</ObtenerPrecioResponse>
```

Esquema:

```
<element name="ObtenerPrecio">
  <complexType><all>
    <element name="item" type="string"/>
  </all></complexType>
</element>
<element name="ObtenerPrecioResponse">
  <complexType><all>
    <element name="Precio" type="float"/>
  </all></complexType>
</element>
```


Contenidos

1. Introducción a los servicios Web
2. **Elementos en un Servicio Web:**
 1. XML
 2. **SOAP**
 3. WSDL
 4. UDDI
3. Ejemplo de aplicación
 - ▶ Bolsa (servicio público)

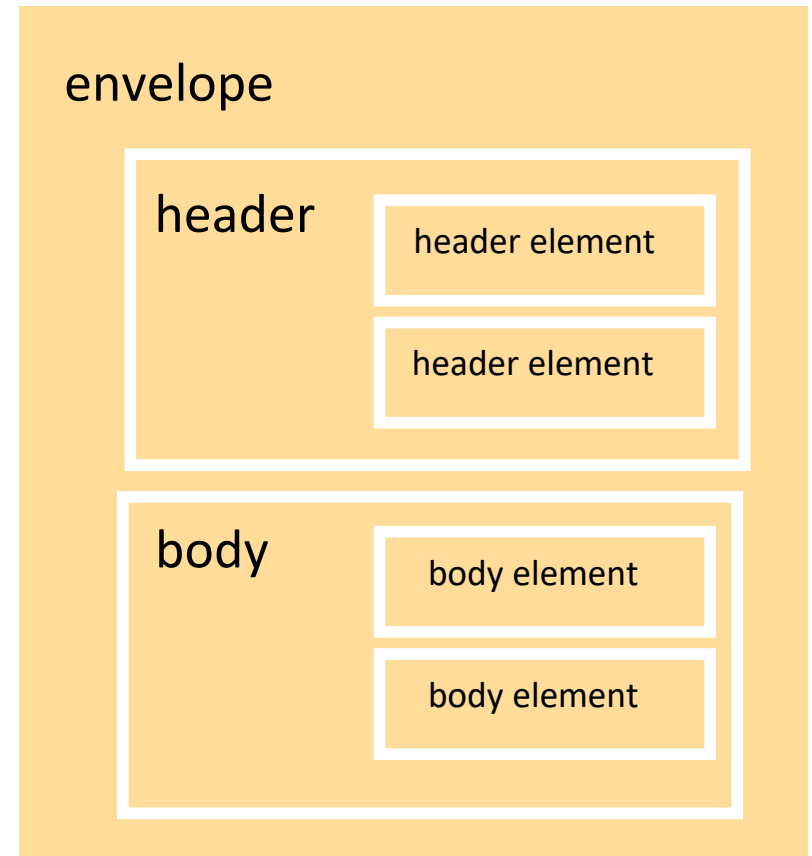


SOAP

- ▶ *Simple Object Access Protocol*
 - ▶ <http://www.w3.org>
- ▶ **SOAP específica:**
 - ▶ Cómo representar los mensajes en XML
 - ▶ Como combinar mensajes SOAP para un modelo petición-respuesta
 - ▶ Cómo procesar los elementos de los mensajes
 - ▶ Cómo utilizar el transporte (HTTP, SMTP, ...)
para enviar mensajes SOAP

Mensaje SOAP

- ▶ Unidad básica de comunicación entre nodos SOAP
- ▶ El mensaje es transportado en un *envelope*
 - ▶ Encabezado opcional
 - ▶ Cuerpo
- ▶ Los elementos XML anteriores son definidos como un esquema en el espacio de nombres XML
 - ▶ Esquema definido en <http://www.w3.org>



Ejemplo de petición/respuesta

“
<ObtenerPrecio>
 <item>mesa</item>
</ObtenerPrecio>
”



POST /StockQuote HTTP/1.1

.....
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
 <SOAP-ENV:Body>
 <m:ObtenerPrecio xmlns:m="http://example.com/stockquote.xsd">
 <item>mesa</item>
 </m:ObtenerPrecio>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Ejemplo de petición/*respuesta*

“
<ObtenerPrecioResponse>
 <precio>134.5</precio>
</ObtenerPrecioResponse>
”



HTTP/1.1 200 OK

.....

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
  <SOAP-ENV:Body>
    <m:ObtenerPrecioResponse xmlns:m="http://example.com/stockquote.xsd">
      <Precio>134.5</Precio>
    </m:ObtenerPrecioResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Contenidos

1. Introducción a los servicios Web
2. **Elementos en un Servicio Web:**
 1. XML
 2. SOAP
 3. **WSDL**
 4. UDDI
3. Ejemplo de aplicación
 - ▶ Bolsa (servicio público)

WSDL

- *WSDL: Web Services Description Language*
 - IDL para servicios Web en XML
- Se utiliza para:
 - Describir servicios Web
 - Especifica las operaciones y métodos del servicio
 - Localizar servicios Web
- WSDL es un documento XML
 - Escrito en XML
 - Estándar descrito por la W3C
 - <http://www.w3.org/TR/wsdl>
 - <http://www.w3.org/TR/wsdl20>

Ejemplo de WSDL

string NumberToWords (unsignedLong ubiNum)

- ▶ Normalmente generado automáticamente, da lugar a:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.dataaccess.com/webservicesserver/" name="Conversions"
  targetNamespace="http://www.dataaccess.com/webservicesserver/">
  <types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://www.dataaccess.com/webservicesserver/">
      <xs:element name="NumberToWords">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ubiNum" type="xs:unsignedLong"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="NumberToWordsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="NumberToWordsResult" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </types>
  <message name="NumberToWordsSoapRequest">
    <part name="parameters" element="tns:NumberToWords"/>
  </message>
  <message name="NumberToWordsSoapResponse">
    <part name="parameters" element="tns:NumberToWordsResponse"/>
  </message>
  <portType name="ConversionsSoapType">
    <operation name="NumberToWords">
      <documentation>Returns the word corresponding to the positive number passed as parameter. Limited to quadrillions.</documentation>
      <input message="tns:NumberToWordsSoapRequest"/>
      <output message="tns:NumberToWordsSoapResponse"/>
    </operation>
  </portType>
  <binding name="ConversionsSoapBinding" type="tns:ConversionsSoapType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="NumberToWords">
      <soap:operation soapAction="" style="document"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="Conversions">
    <documentation>The Conversion Visual DataPlex Web Service will provide different conversion functions. The function currently available will help you converting numbers into words.</documentation>
    <port name="ConversionsSoap" binding="tns:ConversionsSoapBinding">
      <soap:address location="http://www.dataaccess.com/webservicesserver/conversions.wso"/>
    </port>
  </service>
</definitions>
```



Ejemplo de WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.dataaccess.com/webservicesserver/"
  name="Conversions" targetNamespace="http://www.dataaccess.com/webservicesserver/">
  <types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://www.dataaccess.com/webservicesserver/">
      <xs:element name="NumberToWords">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ubiNum" type="xs:unsignedLong"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="NumberToWordsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="NumberToWordsResult" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </types>
```

```

<message name="NumberToWordsSoapRequest">
  <part name="parameters" element="tns:NumberToWords"/>
</message>
<message name="NumberToWordsSoapResponse">
  <part name="parameters" element="tns:NumberToWordsResponse"/>
</message>
<portType name="ConversionsSoapType">
  <operation name="NumberToWords">
    <documentation>Returns the word corresponding to the positive number passed as parameter. Limited to
      quadrillions.</documentation>
    <input message="tns:NumberToWordsSoapRequest"/>
    <output message="tns:NumberToWordsSoapResponse"/>
  </operation>
</portType>
<binding name="ConversionsSoapBinding" type="tns:ConversionsSoapType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="NumberToWords">
    <soap:operation soapAction="" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="Conversions">
  <documentation>The Conversion Visual DataFlex Web Service will provide different conversion functions. The function currently
    available will help you converting numbers into words.</documentation>
  <port name="ConversionsSoap" binding="tns:ConversionsSoapBinding">
    <soap:address location="http://www.dataaccess.com/webserviceserver/conversions.wso"/>
  </port>
</service>
</definitions>

```

Estructura de un documento WSDL

```
<definitions>  
  <types>  
    definición de tipos (independientes del lenguajes)  
  </types>  
  <message>  
    definición de mensajes (a intercambiar)  
  </message>  
  <interfaz>  
    definición de puertos (interfaz de funciones, incluyendo parámetros, etc.)  
  </portType>  
  <binding>  
    definición de enlaces (formato de los mensajes y datos a usar)  
  </binding>  
  <services>  
    definición de servicios (nombre de servicio y 1 ó más puertos donde se dá)  
  </services>  
</definitions>
```

Espacio de nombres

- ▶ Definido por el W3C:
<http://www.w3.org/2001/XMLSchema>
- ▶ **Objetivo:** evitar conflictos
 - ▶ Dos servicios web distintos A y B que tienen un elemento común f.
- ▶ Cada instancia de f se puede referir como A:f o B:f

Contenidos

1. Introducción a los servicios Web
2. **Elementos en un Servicio Web:**
 1. XML
 2. SOAP
 3. WSDL
 4. **UDDI**
3. Ejemplo de aplicación
 - ▶ Bolsa (servicio público)

UDDI

- ▶ *Universal Description, Discovery, and Integration*
 - ▶ No estándar: Propuesta inicial de Microsoft, IBM y Ariba
- ▶ **Registro distribuido de servicios web** ofrecidos por empresas
- ▶ Información clasificada en 3 categorías (guías):
 - ▶ Páginas blancas: Datos de la empresa
 - ▶ Páginas amarillas: Clasificación por tipo de actividades
 - ▶ Páginas verdes: Descripción de servicios web (WSDL)
- ▶ **Se accede a su vez como un servicio web**
- ▶ Puede consultarse en tiempo de desarrollo o incluso dinámicamente en tiempo de ejecución
- ▶ Permite búsquedas por distintos criterios
 - ▶ Tipo de actividad, tipo de servicio, localización geográfica

URI, URL y URN

- ▶ Cada servicio Web tiene una **URI** (*Uniform Resource Identifier*):
 - ▶ **URL** (*uniform resource locator*)
 - ▶ Incluyen la localización del recurso (*hostname+pathname*)
 - ▶ **URN** (*uniform resource name*)
 - ▶ Nombres de recursos que no incluyen localización
- ▶ Los clientes usan la **URI** para referenciar el servicio
- ▶ Existen servicios de localización de servicios (directorios)
 - ▶ **URN -> URL**
 - ▶ Permiten transparencia de localización

Contenidos

1. **Introducción a los servicios Web**
2. **Elementos en un Servicio Web:**
 1. XML
 2. SOAP
 3. WSDL
 4. UDDI
3. **Ejemplo de aplicación**
 - ▶ **Bolsa (servicio público)**

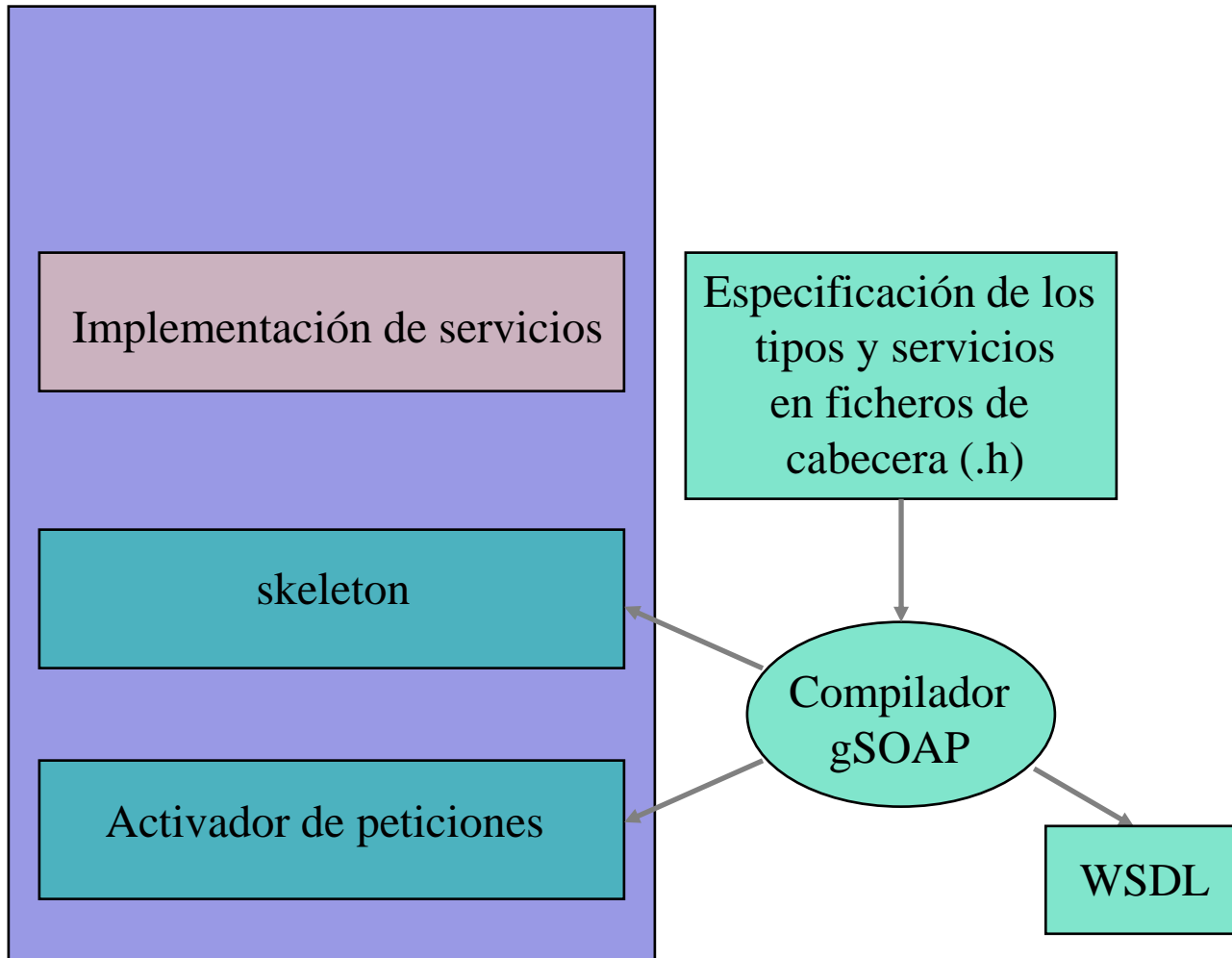
Plataforma de desarrollo

- ▶ **gSOAP**

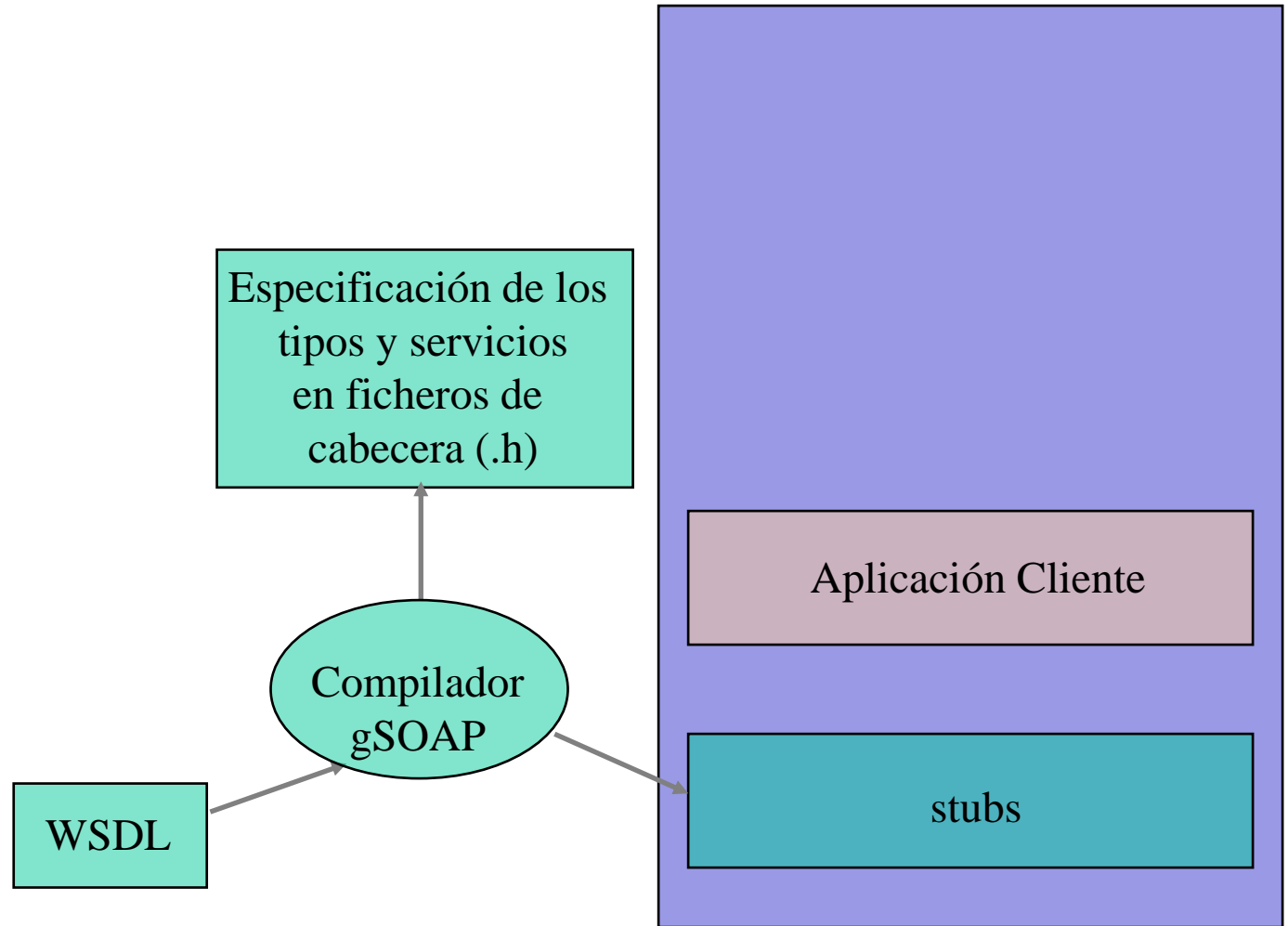
- ▶ Conjunto de herramientas para el desarrollo de aplicaciones basadas en servicios Web en C/C++

- ▶ <http://www.cs.fsu.edu/~engelen/soap.html>

Desarrollo del servidor



Desarrollo del cliente



Ejemplo: cotizaciones

- ▶ WSDL:

- ▶ <http://www.websvicex.com/stockquote.asmx?WSDL>

- ▶ Accesible desde:

- <http://www.xmethods.com/>

- ▶ Descrito en:

- <http://www.websvicex.com/stockquote.asmx?op=GetQuote>

- ▶ Tres métodos:

```
string GetQuote (char *symbol)
// Returns the last quote (and more) from the company
// represented by 'symbol'.
```

Generación de la interfaz a partir del WSDL

```
acaldero@guernika# wsd12h -c \  
                        -o quote.h \  
                        http://www.webservicex.com/stockquote.asmx?WSDL
```

Preprocesado de la interfaz a C

```
acaldero@guernika# soapcpp2 -C -c quote.h
```

Client.c

```
#include "soapH.h"
#include "StockQuoteSoap.nsmap"

int main(int argc, char **argv)
{
    struct soap soap;
    struct __ns1__GetQuote          arg1 ;
    struct __ns1__GetQuoteResponse arg2 ;

    arg1.symbol = argv[1];
    soap_init(&soap);

    if (soap_call__ns2__GetQuote(&soap, NULL, NULL, &arg1, &arg2) == 0)
        printf("Result: %s\n", arg2.GetQuoteResult);
    else soap_print_fault(&soap, stderr);
    return 0;
}
```

Compilación del ejemplo

guernika.lab.inf.uc3m.es

```
# gcc -g -c client.c -I/opt/gsoap-linux-2.7/ -o client.o
# gcc -g -c soapC.c -I/opt/gsoap-linux-2.7/ -o soapC.o
# gcc -g -c soapClient.c -I/opt/gsoap-linux-2.7/ -o soapClient.o
# gcc -Wall -g -I/opt/gsoap-linux-2.7/ \
    -c /opt/gsoap-linux-2.7/stdsoap2.c -o stdsoap2.o

# gcc -o client client.o soapClient.o soapC.o stdsoap2.o
```


Ejecución del ejemplo

guernika.lab.inf.uc3m.es

```
acaldero@guernika # ./client AAPL
```

Result:

```
<StockQuotes><Stock><Symbol>AAPL</Symbol><Last>188.50</Last><Date>10/30/2009</Date><Time>3:00pm</Time><Change>-7.85</Change><Open>196.18</Open><High>196.80</High><Low>186.0667</Low><Volume>25625842</Volume><MktCap>169.8B</MktCap><PreviousClose>196.35</PreviousClose><PercentageChange>-4.00%</PercentageChange><AnnRange>78.20 - 208.71</AnnRange><Earns>6.289</Earns><P-E>31.22</P-E><Name>Apple Inc.</Name></Stock></StockQuotes>
```



Aplicaciones de Internet: Servicios Web



Grupo ARCOS

Desarrollo de Aplicaciones Distribuidas

Ingeniería Informática

Universidad Carlos III de Madrid