**Clase del día 27/04/2021**

En la clase de hoy veremos los conceptos básicos de los servicios web (Web Services) así como los elementos de servicios web SOAP y REST.

**Conceptos básicos de Servicios web**

En el documento [Web Services Architecture](http://www.w3.org/TR/ws-arch/) del World Wide Web Consortium (W2C) define un servicio web como:

“Un sistema de software diseñado para soportar la interacción interoperable de maquina-a-máquina sobre una red. Este cuenta con una interface descrita en un formato el cual puede ser procesado por una computadora (especificamente WSDL). Otros sistemas interactúan con el servicio web en una manera prescrita por su descripción usando mensajes SOAP, tipicamente transportados usando HTTP con una serialización XML en conjunción con otros estándares relativos a la Web”.

Un servicio web es un concepto abstracto que debe ser implementado mediante un agente concreto.

Un ***agente*** es el software o hardware que envía y recibe mensajes. El ***servicio*** es el recurso caracterizado por un conjunto abstracto de la funcionalidad que se provee. Un servicio web no cambia aún cuando cambie el agente, es decir, la funcionalidad es independiente de la implementación de ésta.

El propósito de servicio web es proveer cierta funcionalidad a nombre de su propietario (una persona o una organización). La ***entidad proveedora*** es aquella persona u organización que provee un agente que implementa un determinado servicio.

Una ***entidad solicitante*** es una persona u organización que desea hacer uso del servicio mediante un ***agente solicitante*** (también llamado *solicitante del servicio*) que intercambia mensajes con el ***agente proveedor*** (también llamado *proveedor del servicio*).

En la mayoría de los casos el agente solicitante es el que inicia la comunicación con el agente proveedor, aunque no siempre es así, no obstante se sigue llamando agente solicitante aunque no sea el que inicia la comunicación.

La ***semántica*** de un servicio web es la expectativa compartida sobre el comportamiento del servicio, en particular el comportamiento en respuesta a los mensajes que recibe.

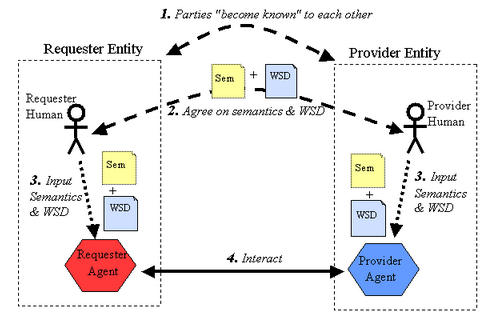
Se le llama **contrato** al acuerdo entre la entidad solicitante y la entidad proveedora. Un contrato puede ser explícito o implícito, escrito u oral, establecido entre las personas y/o las computadoras, legal o informal.

Hay dos tipos de contratos: 1) la *descripción del servicio* es el contrato que gobierna la mecánica de interacción con un servicio en particular y 2) la *semántica* del servicio es el contrato que gobierna el significado y propósito de la interacción. Sin embargo, puede haber contratos “híbridos” que incluyan elementos de descripción y elementos de semántica.

**Participación en un servicio web**

Una entidad solicitante puede participar de un servicio web de diferentes maneras. La siguiente figura muestra el proceso general de participación en un servicio web.

1. Las entidades solicitante y proveedora se conocen una a la otra, o por lo menos una conoce a la otra.  
2. Las entidades acuerdan la descripción (WSD: Web Service Description) y semántica del servicio.  
3. La descripción y la semántica son implementadas por el agente solicitante y el agente proveedor.  
4. Los agentes solicitante y proveedor intercambian mensajes.



Fuente: Web Services Architecture, W3C

**Servicios web basados en SOAP**

SOAP (Simple Object Access Protocol) define un protocolo de RPC (Remote Procedure Call) basado en XML, para la interacción cliente-servidor a través de la red utilizando: 1) HTTP como la base de transporte, y 2) documentos XML para la codificación de requerimientos y respuestas.

SOAP permite la comunicación entre aplicaciones ejecutando en diferentes sistemas operativos, con diferentes tecnologías y lenguajes de programación.

**Componentes de un mensaje SOAP**

Un mensaje SOAP es un documento XML compuesto por los siguientes elementos:

* Envelope (sobre). Identifica el documento XML como un mensage SOAP.
* Header (encabezado). Contiene información de encabezado.
* Body (cuerpo). Contiene información del requerimiento y la respuesta.
* Fault (falla). Contiene errores e información de estatus.

Para implementar servicios web en Java se puede utilizar la API JAX-WS.

**Web Services Description Language (WSDL)**

Un documento WSDL es un documento XML que contiene la descripción de un servicio web SOAP. Este especifica la localización del servicio y los métodos del servicio.

Un cliente puede hacer un requerimiento HTTP a un servicio web SOAP para obtener el WSDL que describe el servicio web.

Los elementos de un documento WSDL son los siguientes:

--------------------------------------------------------------------------------------------------------------------  
Elemento                    Descripción  
--------------------------------------------------------------------------------------------------------------------  
<types>                      Define los tipos de dato usados por el servicio web  
<message>                Define los elementos de datos para cada operación  
<servicioportType>    Describe las operaciones que pueden ser ejecutadas y los mensajes involucrados  
<binding>                  Define el protocolo y el formato de los datos para cada portType

Un ejemplo de WSDL es el siguiente (fuente www.w3schools.com/xml/xml\_wsdl.asp):

<message name="getTermRequest">  
  <part name="term" type="xs:string"/>  
</message>  
<message name="getTermResponse">  
  <part name="value" type="xs:string"/>  
</message>  
<portType name="glossaryTerms">  
  <operation name="getTerm">  
    <input message="getTermRequest"/>  
    <output message="getTermResponse"/>  
  </operation>  
</portType>  
<binding type="glossaryTerms" name="b1">  
   <soap:binding style="document"  
   transport="http://schemas.xmlsoap.org/soap/http" />  
   <operation>  
     <soap:operation soapAction="http://example.com/getTerm"/>  
     <input><soap:body use="literal"/></input>  
     <output><soap:body use="literal"/></output>  
  </operation>  
</binding>

El siguiente código, es un ejemplo de un servicio web escrito en Java utilizando el API JAX-WS:

package negocio;

import javax.jws.WebMethod;  
import javax.jws.WebParam;  
import javax.jws.WebService;  
import javax.naming.InitialContext;  
import javax.sql.DataSource;  
import java.sql.Connection;  
  
@WebService  
public class ServicioSOAP  
{  
    static DataSource pool;  
    static  
    {  
        pool = null;  
        try  
        {  
            pool = (DataSource)new InitialContext().lookup("java:comp/env/jdbc/prueba");  
        }  
        catch (Exception e)  
        {  
            e.printStackTrace();  
        }  
    }  
    @WebMethod  
    public Integer suma(@WebParam(name = "a") Integer a, @WebParam(name = "b") Integer b) throws Exception  
    {  
        return a + b;  
    }  
    @WebMethod  
    public String mayusculas(@WebParam(name = "s") String s) throws Exception  
    {  
        return s.toUpperCase();  
    }  
    @WebMethod  
    public void prueba\_conexion\_bd() throws Exception  
    {  
        Connection conexion = pool.getConnection();  
        conexion.close();  
    }  
}

En este ejemplo podemos ver que la anotación @WebService define la clase correspondiente al servicio web. Cada operación del servicio web se implementa como un método de Java incluyendo la anotación @WebMethod. La anotación @WebParam se utiliza para definir los parámetros de cada operación del servicio web.

En este caso el servicio web accede a una base de datos llamada "prueba" mediante JDBC.

**Servicios web estilo REST**

REST define un conjunto de principios arquitectónicos para la creación de servicios web. REST fue presentado por Roy Fielding el año 2000 en su disertación doctoral "Architectural Styles and the Design of Network-based Software Architectures".

El diseño de servicios web estilo REST sigue cuatro principios:

* Utilizar métodos HTTP de forma explícita.
  + Un servicio web utiliza los métodos de HTTP para crear un recurso (POST), leer (GET), cambiar el estado o actualizar un recurso (PUT), y borrar un recurso (DELETE).
* Los servicios son sin estado (stateless).
  + Los clientes de servicios web estilo REST deben enviar peticiones completas e independientes, es decir, las peticiones deben incluir todos los datos que permitan completar el servicio, sin la necesidad de guardar un estado entre peticiones.
* Los URIs representan una estructura de directorios.
  + Los URIs (Uniform Resource Identifier) deben ser intuitivos y auto-explicados. Un URI es una jerarquía que corresponde a la estructura de los servicios web definidos en la empresa.
* Se transfiere XML, JSON o ambos.
  + Los recursos que provee un servicio web pueden ser documentos, imágenes, videos y en general objetos. La representación de objetos mediante XML o JSON es fácil e independiente de la plataforma.

**Actividades individuales a realizar**

Ver el video:

<https://www.youtube.com/watch?v=UAwQFrCtpJ0&t=1s>