

## Tarea:

### Web service:

Un web service es un método de comunicación entre dos aparatos electrónicos en una red. Es una colección de protocolos abiertos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet.

Como sistema de mensajes se utiliza XML estandarizado. El protocolo más simple para el intercambio de información entre ordenadores es XML-RPC, que emplea XML para llevar a cabo RPCs. RPC, (Remote Procedure Call), es un protocolo de red que permite a un programa a ejecutar código en una máquina remota. Los XML-RPC "requests" son una combinación entre XML y headers HTTP. La simpleza de los XML-RPC hizo que el estándar evolucionase a SOAP, uno de los componentes básicos de los web-services.

La base de comunicación entre web services es por tanto el lenguaje XML y el protocolo HTTP. Se clasifican como contenedores que encapsulan funciones específicas y hace que estas funciones puedan ser utilizadas en otros servidores.

### Componentes:

Los web services estandarizados funcionan con los siguientes componentes:

- SOAP (Simple Object Access Protocol). Es un protocolo escrito en XML para el intercambio de información entre aplicaciones. Es un formato para enviar mensajes, diseñado especialmente para servir de comunicación en Internet.
- WSDL (Web Services Description Language). Es un lenguaje basado en XML para describir los servicios web y cómo acceder a ellos. Es el formato estándar para describir un web service.
- UDDI (Universal Description, Discovery and Integration). Es un estándar XML para describir, publicar y encontrar servicios web, un directorio donde las compañías pueden registrar y buscar servicios web.

### Arquitectura:

En la arquitectura de los web services se identifican las siguientes partes:

- Service Discovery. Es el responsable de centralizar servicios web en un directorio común de registro y proveer una funcionalidad sencilla para publicar y buscar. UDDI se encarga del Service Discovery.
- Service Description. Uno de los aspectos más característicos de los web services es que se autodescriben. Esto significa que una vez que se ha localizado un Web Service nos proporcionará información sobre que operaciones soporta y cómo activarlo. Esto se realiza a través del WSDL.
- Service Invocation. Invocar a un Web Service implica pasar mensajes entre el cliente y el servidor. SOAP especifica cómo deberíamos formatear los mensajes request para el servidor, y cómo el servidor debería formatear sus mensajes de respuesta.
- Transport. Todos estos mensajes han de ser transmitidos de alguna forma entre el servidor y el cliente. El protocolo elegido para ello es HTTP. Se pueden utilizar otros protocolos, pero HTTP es actualmente el más usado.

### Funcionamiento:

1. El Service Provider genera el WSDL describiendo el Web Service y registra el WSDL en el directorio UDDI o Service Registry.
2. El Service Requestor o la aplicación del cliente requiere un Web Service y se pone en contacto con el UDDI para localizar el Web Service.
3. El cliente, basándose en la descripción descrita por el WSDL, envía un request para un servicio particular al Web Service Listener, que se encarga de recibir y enviar los mensajes en formato SOAP.
4. El Web Service analiza el mensaje SOAP del request e invoca una operación particular en la aplicación para procesar el request. El resultado se escribe de nuevo en SOAP en forma de respuesta y se envía al cliente.
5. El cliente analiza el mensaje de respuesta SOAP y lo interpreta o genera un error si ha habido alguno.

### Ventajas:

- Al estar basados en XML, los servicios web son fáciles de adaptar, extender y personalizar.
- Independencia del lenguaje de programación. El servidor y el cliente no necesitan estar escritos en el mismo lenguaje.
- Independencia del modo de transporte. Pueden funcionar sobre distintos protocolos de transporte, por ejemplo: HTTP, HTTPS, SMTP, FTP, etc.
- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Múltiples estilos de comunicación, los servicios web soportan tanto comunicación síncrona (RPC) como asíncrona (mensajería).
- Fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permite que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

### Desventajas:

- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear.
- Existe poca información de los servicios web para algunos lenguajes de programación.
- Es una tecnología relativamente nueva y en plena fase de crecimiento, por tanto, todavía existen algunos estándares que aún no han sido definidos en su totalidad o que podrían ser modificados.
- Existe dependencia de la disponibilidad de servidores y comunicaciones.
- La seguridad de los servicios, puede ser una tarea muy difícil de implementarla correctamente.
- El canal puede sufrir sobrecarga por el uso de XML en el transporte de mensajes produciendo lentitud en la velocidad de la comunicación.

## Codecademy:

Learn SQL | Codecademy

Es seguro | <https://www.codecademy.com/learn/learn-sql>

Aplicaciones Microsoft Word - 2d Bleach: Can't Fear Y Potencial eléctrico d programación neur Background Wallpap Assassination 2015 Cursos Área de Esta Otros marcadores

codecademy Upgrade to Pro Learn Community Catalog

Dashboard

Courses

- Learn Git 100%
- Learn the Command Li... 25%
- Learn SQL 100%

**Learn SQL**

Great job completing Learn SQL

COMPLETED

[Reset Progress](#)

**INTENSIVE**

Learn an Introduction to Data Analysis by taking this accelerated 10-week program.

[LEARN FASTER](#)

Overview Syllabus

- Learn SQL - Manipulation
- Learn SQL - Queries