



## Tema 6: Programación de servicios web

### ¿Qué aprenderás?

- Qué son los servicios web y cómo crearlos.
- Los protocolos que intervienen en la comunicación con un servicio web.
- Creación de servicios web con la librería NuSOAP de PHP.

### ¿Sabías que...?

- A la hora de implementar servicios web, es habitual el uso de entornos de desarrollo integrado (IDE) como Eclipse, NetBeans o Visual Studio.
- OASIS y W3C son entidades que establecen normas y estándares para los servicios web.



## 1. Servicios web

---

### 1.1. ¿Qué es un servicio web?

---

Un servicio web es un conjunto de tecnologías, protocolos y estándares que hacen posible la comunicación entre varios sistemas o aplicaciones a través de la red.

La necesidad de los servicios web surgió de forma inevitable debido al rápido crecimiento de Internet. La red se iba poblando de diversos sistemas y aplicaciones, con diferentes arquitecturas, lenguajes de programación, incluso hardware. Así pues, con la necesidad de estos sistemas de intercambiar información, se vio la necesidad de crear herramientas con unos estándares definidos para hacer posible la comunicación.

Piensa en una agencia de viajes online, que ofrece la posibilidad de reservar avión y hotel. El cliente solicita un servicio a la agencia online, que actuaría como publicador. Ésta solicitaría datos a través de servicios web de los hoteles y las líneas aéreas. Además, se incluiría el método de pago, otro servicio web ofrecido por una entidad financiera.

### 1.2. Protocolos y estándares que intervienen en un servicio web

---

Como ya hemos dicho, los servicios web hacen uso de un conjunto de protocolos y estándares para que diversas aplicaciones intercambien información. A este conjunto se le conoce como pila de protocolos de servicios web, o Web Services Protocol Stack.

Los protocolos incluidos en esta pila se usan para definir, localizar, implementar y hacer que un servicio web interactúe con un cliente o con otro servicio web. Los podemos agrupar en las siguientes áreas:

1. Servicio de transporte.
2. Servicio de mensajería.
3. Servicio de descripción.
4. Servicio de descubrimiento.

A continuación, vamos a ver en qué consiste cada uno de estos servicios.

#### 1.2.1. Servicio de transporte

El servicio de transporte se encarga de enviar los mensajes entre las aplicaciones. Usa protocolos como HTTP, SMTP (protocolo usado en los correos electrónicos), FTP (protocolo usado para el envío de ficheros), o BEEP (Blocks Extensible Exchange Protocol).



### 1.2.2. Servicio de mensajería

El servicio de mensajería se encarga de la codificación de los mensajes que van a intercambiar las aplicaciones. Se usa el lenguaje XML para construir mensajes que sean entendibles por las partes que intervienen la comunicación.

En este servicio se usan protocolos como XML-RPC (Remote Procedure Call mediante XML), SOAP (Simple Object Access Protocol), o REST (Representational State Transfer).

SOAP es el protocolo que vamos a usar durante el curso. Utiliza XML para la comunicación, igual que el protocolo XML-RPC del cual deriva.

Con el protocolo SOAP podemos comunicar aplicaciones desarrolladas en diferentes lenguajes y que incluso se ejecutan en diferentes sistemas operativos. Para ello se usa un lenguaje intermedio parecido a XML mediante el cual el cliente crea un mensaje que contiene los métodos a invocar y los parámetros que se necesitan. El servidor recibe el mensaje y lo convierte a su lenguaje de programación. Ejecuta los métodos solicitados con los parámetros enviados y el resultado obtenido lo convierte al lenguaje intermedio de nuevo. Finalmente, se envía al cliente.

### 1.2.3. Servicio de descripción

El servicio de descripción es usado por cliente y servidor para establecer los aspectos técnicos de la comunicación que van a establecer, como son el método de transporte de los mensajes y el contenido (sintaxis) de éstos. De esta forma se describe la interfaz pública de un servicio web.

El protocolo más utilizado en el servicio de descripción es WSDL (Web Services Description Language) que usa el lenguaje XML para la descripción del servicio web.

WSDL describe los protocolos que se van a usar y el formato de los mensajes que van a intercambiar cliente y servidor para el correcto uso de los métodos que proporciona el servidor.

Con el WSDL un cliente puede leer los métodos que puede ejecutar en el servidor, así como tipos de datos que se pueden usar. Además, se puede ver otra información, como son los mensajes, los tipos de puertos, los protocolos de comunicación usados (bindings) y los servicios. En ningún momento sabemos qué hace el servidor (implementación), pero tenemos la información suficiente para interactuar con él.

### 1.2.4. Servicio de descubrimiento

El servicio de descubrimiento se encarga de centralizar servicios web de varios servidores en un registro común (catálogo), en el que se especifica la localización y descripción de estos servicios web. De esta forma se facilita su descubrimiento para poder usarse.

El catálogo de servicios web es UDDI (Universal Description, Discovery and Integration). Se usa el lenguaje XML para el registro de los servicios.



### 1.3. Generación de servicios web en PHP. Librería NuSOAP

PHP, como lenguaje para desarrollar aplicaciones web en entorno servidor, ofrece diversas herramientas para crear servicios web. En este curso veremos cómo crear nuestros servicios con la librería llamada NuSOAP.

Hay muchas librerías para crear servicios web con PHP, incluso el propio lenguaje posee una librería interna para tal uso. NuSOAP es una librería que presenta una fase de desarrollo más avanzada que otras, por eso vamos a utilizarla.

#### 1.3.1. Instalación



Instalación de NuSOAP

#### 1.3.2. Creación y configuración del servidor

Vamos a crear un archivo en PHP llamado “servidor.php” en el que se implementarán los métodos que ofrecerá el servicio web.

Primero incluimos la librería de NuSOAP en nuestro archivo y creamos y configuramos el servidor:

```
require_once("lib/nusoap.php");  
$namespace = "http://localhost/nusoap/servidor.php";  
$server = new soap_server();  
$server->configureWSDL("MiServicioWeb", $namespace);  
$server->schemaTargetNamespace = $namespace;  
$server->soap_defencoding = 'UTF-8';
```

Ahora implementaremos los métodos que queremos incluir en el servicio web.

Haremos dos funciones:

- La típica que muestra el mensaje “Hola mundo”. No recibirá ningún parámetro de entrada y devolverá una cadena de caracteres.



- Una función que sumará dos valores que recibirá como parámetro y devolverá el resultado.

```
function holaMundo() {  
    return "Hola, mundo!!!";  
}  
  
function suma($valor1, $valor2) {  
    return $valor1 + $valor2;  
}
```

A continuación, registramos los métodos anteriores en el servicio web. Sin este paso, no podríamos usar las funciones antes implementadas.

```
$server->register(  
    'holaMundo',  
    array(),  
    array('return'=>'xsd:string'),  
    $namespace,  
    false,  
    'rpc',  
    'encoded',  
    'Método HolaMundo por excelencia'  
);  
  
$server->register(  
    'suma',  
    array('valor1'=>'xsd:int',  
        'valor2'=>'xsd:int'),  
    array('return'=>'xsd:int'),  
    $namespace,  
    false,  
    'rpc',  
    'encoded',  
    'Método que recibe dos enteros y devuelve otro  
    entero con la suma'  
);
```

Finalmente, añadimos la siguiente línea, encargada de iniciar el servicio y dejarlo en escucha de posibles peticiones:

```
$server->service(file_get_contents("php://input"));
```

### 1.3.3. Creación y configuración del cliente

Para probar el servicio web que acabamos de crear, vamos a abrir un nuevo archivo PHP con el nombre “cliente.php”. En él, implementaremos la llamada a los métodos “holaMundo” y “suma” del servicio web. El código será el siguiente:

```
require_once("lib/nusoap.php");  
$client = new  
soapclient('http://localhost/nusoap/servidor.php?wsdl');
```



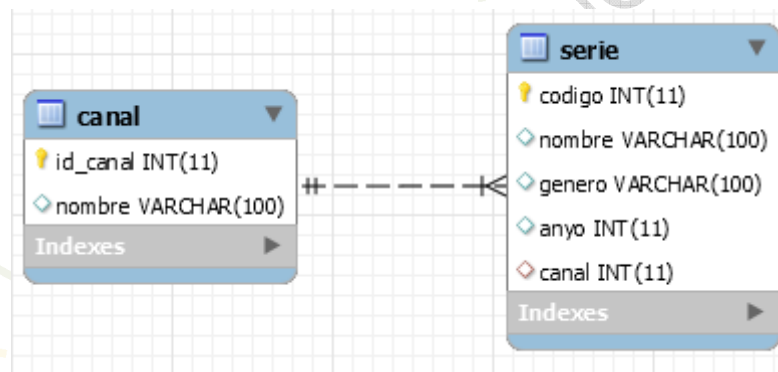
```
$result = $client->holaMundo();
echo "$result<br/>";

$result = $client->suma(6,9);
echo "$result<br/>";
```

### 1.3.4. Trabajando con tipos complejos

En el servicio web que hemos creado como ejemplo hemos visto cómo registramos dos funciones que trabajan con tipos de datos simples, como son enteros y cadenas de caracteres. Pero, obviamente, podemos hacer servicios web que incluyan métodos que trabajen con tipos de datos complejos: por ejemplo, un método puede devolvernos un array.

Vamos a hacer un ejemplo completo que incluya operaciones de consulta sobre una base de datos. Concretamente, supongamos que tenemos la siguiente base de datos llamada “Series”:



Implementaremos un método que nos devolverá todos los datos de la tabla “serie”. Para simplificar, no vamos a relacionar las dos tablas en la consulta, es decir, no cogeremos el nombre del canal que emite cada serie, simplemente mostraremos el identificador.

Primero creamos el método en el archivo “servidor.php”:

```
function listaSeries(){
    require_once('datos.php');
    $misPelis = array();
    $con = mysqli_connect($host, $user, $pass,
        $db_name);
    $query = "select * from serie";
    $series = mysqli_query($con, $query);
    while($serie = mysqli_fetch_assoc($series)){
        $misSeries[] = $serie;
    }
    mysqli_close($con);
    return $misSeries;
}
```

El archivo “datos.php” contiene las variables de conexión con la base de datos: \$host, \$user, \$pass y \$db\_name.



Vemos que el método “listaSeries” devuelve un array. Hay que especificar en el servidor qué datos va a devolver exactamente. Lo hacemos de la siguiente manera:

```
$server->wsdl->addComplexType(  
    'Serie',  
    'complexType',  
    'struct',  
    'all',  
    '',  
    array(  
        'codigo' =>  
            array('name'=>'codigo', 'type'=>'xsd:int'),  
        'nombre' =>  
            array('name'=>'nombre', 'type'=>'xsd:string'),  
        'genero' =>  
            array('name'=>'genero', 'type'=>'xsd:string'),  
        'anyo' =>  
            array('name'=>'anyo', 'type'=>'xsd:int'),  
        'canal' =>  
            array('name'=>'canal', 'type'=>'xsd:int')  
    )  
);  
$server->wsdl->addComplexType(  
    'ArraySeries',  
    'complexType',  
    'array',  
    '',  
    'SOAP-ENC:Array',  
    array(),  
    array(  
        array('ref'=>'SOAP-ENC:arrayType',  
            'wsdl:arrayType'=>'tns:Serie[]')),  
    'tns:Serie'  
);
```

Vemos que primero se define el tipo de datos complejo “Serie”, y luego se define el tipo “ArraySeries”, que contendrá varios tipos “Serie”.

Por último, registramos el método en el servicio web:

```
$server->register(  
    'listaSeries',  
    array(),  
    array('return'=>'tns:ArraySeries'),  
    $namespace,  
    false,  
    'rpc',  
    'encoded',  
    'Método que devuelve una array con las series de una  
    base de datos'  
);
```



En el archivo “cliente.php” añadimos las líneas encargadas de llamar al método “listaSeries” del servicio web:

```
$result = $client->listaSeries();  
print_r($result);  
echo "<br/>";
```

VERSIÓN IMPRIMIBLE ALUMNO LINKIAFP





## Test de autoevaluación

¿En qué tipo de servicio se usa el protocolo SOAP?

- a) Servicio de transporte
- b) Servicio de mensajería
- c) Servicio de descripción
- d) Servicio de descubrimiento

¿En qué tipo de servicio se usa el protocolo WSDL?

- a) Servicio de transporte
- b) Servicio de mensajería
- c) Servicio de descripción
- d) Servicio de descubrimiento

¿Qué lenguaje usa WSDL para la descripción del servicio web?

- a) XML
- b) HTML
- c) PHP
- d) Cualquier lenguaje de entorno servidor



## Recursos y enlaces

---

- [Descarga de la librería NuSOAP para PHP](#)

## Conceptos clave

---

- **Servicio web:** web es un conjunto de tecnologías, protocolos y estándares que hacen posible la comunicación entre varios sistemas o aplicaciones a través de la red.
- **Pila de protocolos de servicios web:** conjunto de protocolos y estándares para que diversas aplicaciones intercambien información. Se pueden agrupar en: servicios de transporte, servicios de mensajería, servicios de descripción y servicios de descubrimiento.



## Ponlo en práctica

### Actividad 1

Usando la base de datos que se ha usado para el ejemplo, crea un servicio web que incluya un método que reciba como parámetro el código de un canal, y devuelva un array con todas las series de ese canal. El resultado debe mostrar el nombre del canal, no el código.



SOLUCIÓN

VERSIÓN IMPRIMIBLE ALUMNO LINKIAFP