

Servicios web



Esta lección está en elaboración.

Qué es un servicio web

Un servicio web es un programa disponible en Internet y que mediante HTTP puede recibir peticiones y entregar información.

Podemos entender un servicio web como una biblioteca de funciones remota. De la misma manera que un programa que quiere realizar una tarea hace uso de una función de una biblioteca, un programa puede recurrir a un servicio web que le proporciona el resultado de una tarea.

Las ventajas de ofrecer un servicio web son muchas: puede ser universal (si se permite el acceso abierto, cualquiera puede aprovecharlo o nosotros aprovechar servicios ya existentes), su actualización es automática (los clientes acceden siempre a la única versión disponible), es más escalable (una aplicación compleja se puede descomponer en muchos servicios web coordinados y dedicar a cada uno la infraestructura que necesite), etc.

Los servicios web también ofrecen desventajas: el retraso debido a la comunicación entre el programa y el servicio, la centralización (si el servicio es inaccesible, el programa no podría funcionar correctamente), etc.

De un programa monolítico a un servicio web

Los siguientes ejemplos muestran la evolución desde una aplicación monolítica hasta una aplicación distribuida (con un servicio web) pasando por el uso de funciones o bibliotecas. El ejemplo es trivial y no justificaría el uso de servicios web, pero sirve como ejemplo.

- La aplicación muestra un número al azar entre 1 y 10.
- Una aplicación monolítica contiene todo el código en un único lugar, en este caso, en la propia página web.

```
<?php
$numero = rand(1, 10);

print "    <p>Número al azar (sin funciones):
```

Número al azar del 1 al 10 (sin funciones): 8

```
<strong>$numero</strong></p>\n";  
?>
```

- Si una parte de la aplicación se utiliza en varios sitios, puede ser conveniente encapsularla como función y las partes repetidas se sustituirían por llamadas a la función.

```
<?php  
function genera_numero() {  
    return rand(1, 10);  
}  
  
$numero = genera_numero();  
  
print "    <p>Número al azar del 1 al 10 (con  
funciones): <strong>$numero</strong></p>\n";  
?>
```

Número al azar del 1 al 10 (con funciones): 5

- Si una función se puede utilizar en varios programas, puede ser conveniente incluirla en una biblioteca. La función ya no está definida en el programa y cada parte (el programa y la biblioteca) puede desarrollarse por separado.

```
<?php  
// biblioteca.php  
function genera_numero() {  
    return rand(1, 10);  
}  
?>
```

```
<?php  
include "biblioteca.php";  
  
$numero = genera_numero();  
  
print "    <p>Número al azar del 1 al 10 (con
```

Número al azar del 1 al 10 (con biblioteca): 8

```
biblioteca): <strong>$numero</strong></p>\n";  
?>
```

Normalmente la biblioteca y el programa se encuentran en el mismo servidor. En principio, la biblioteca podría encontrarse en otro servidor, pero en ese caso el servidor que aloje la biblioteca no realiza ningún cálculo. Los cálculos los realiza siempre el servidor en el que se encuentra el programa.

- La biblioteca puede convertirse en un servicio web que funciona como programa independiente. Con esta arquitectura, se puede distribuir la carga de trabajo entre distintos servidores, puesto que tanto el programa como el servicio web realizan cálculos.

```
<?php  
# webservice.php  
print rand(1, 10);  
?>
```

```
<?php  
$numero =  
file_get_contents("http://.../webservice.php");  
  
print "    <p>Número al azar del 1 al 10 (con  
servicio web): <strong>$numero</strong></p>\n";  
?>
```

Número al azar del 1 al 10 (con servicio web): 6

Es importante señalar que:

- Para llamar al servicio web hay que conocer su URL.
- Este servicio web lo único que hace es contestar un valor al azar cuando se le llama. La funcionalidad podría estar encapsulada en una función o en una biblioteca, en el ejemplo está en el programa.
- Para recoger el dato contestado por el servicio web se puede utilizar la función [file_gets_content\(\)](#)

Argumentos y valores devueltos

En el ejemplo de servicio web anterior, la llamada al servicio no incluye parámetros y la respuesta es un simple dato, que no necesita ninguna estructura, pero es posible tanto enviar parámetros al servicio web como recibir datos compuestos.

- El ejemplo anterior del servicio web anterior era el siguiente:

```
<?php
# webservice.php
print rand(1, 10);
?>
```

```
<?php
$numero =
file_get_contents("http://.../webservice.php");

print "    <p>Número al azar del 1 al 10 (con
servicio web): <strong>$numero</strong></p>\n";
?>
```

Número al azar del 1 al 10 (con servicio web): **3**

- En caso de querer enviar parámetros al servicio web, estos se deben añadir a la URL, como los controles de un formulario. Para ello se puede utilizar la función [http_build_query](#), cuyo argumento es una matriz cuyos índices son los nombres de los controles y cuyos valores los valores de los controles.

En el ejemplo siguiente, se proporcionan al servicio web los límites inferior y superior entre los que el servicio elige un valor al azar:

```
<?php
# webservice.php
$minimo = $_REQUEST["min"];
$maximo = $_REQUEST["max"];

print rand($minimo, $maximo);
?>
```

```
<?php
$minimo = 10;
$maximo = 20;

$consulta = http_build_query(["min" => $minimo, "max" =>
```

Número al azar del 10 al 20 (con servicio web): **12**

```

$maximo]);
$numero = file_get_contents("http://.../webservice.php?
$consulta");

print "    <p>Número al azar del $minimo al $maximo (con
servicio web): <strong>$numero</strong></p>\n";
?>

```

- En caso de querer recibir varios datos del servicio web, el servicio puede imprimir por ejemplo una matriz. Pero como los datos se reciben como cadena, para reconstruir una variable de tipo matriz en el programa se puede la función [json_decode](#), cuyo argumento es la cadena a convertir en matriz.

En el ejemplo siguiente, se proporcionan al servicio web los límites inferior y superior entre los que el servicio elige los valores al azar y la cantidad de valores a generar:

```

<?php
# webservice.php
$minimo = $_REQUEST["min"];
$maximo = $_REQUEST["max"];
$valores = $_REQUEST["n"];

print "[";
for ($i = 0; $i < $valores - 1; $i++) {
    print rand($minimo, $maximo) . ", ";
}
print rand($minimo, $maximo);
print "];
?>

```

```

<?php
$minimo = 10;
$maximo = 20;
$cantidad = 5;

$consulta = http_build_query(["min" => $minimo, "max" => $maximo, "n"

```

Número al azar del 10 al 20 (con servicio web):
12

Número al azar del 10 al 20 (con servicio web):
20

Número al azar del 10 al 20 (con servicio web):
14

Número al azar del 10 al 20 (con servicio web):
19

Número al azar del 10 al 20 (con servicio web):
17

```
=> $cantidad]);  
$numeros = json_decode(file_get_contents("http://.../webservice.php?  
$consulta"));  
  
foreach ($numeros as $valor) {  
    print "    <p>Número al azar del $minimo al $maximo (con servicio  
web): <strong>$valor</strong></p>\n";  
}  
?>
```

Última modificación de esta página: 5 de diciembre de 2018



Esta página forma parte del curso [Programación web en PHP](#) por [Bartolomé Sintés Marco](#) que se distribuye bajo una [Licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#).