



Por cortesía de **Pedro A. Castillo**
Valdivieso para
ProgramadorPHP.Org – **Juan Belón**

¿Qué es PHP?

Un lenguaje de "código abierto"
interpretado, de alto nivel,
embebido en páginas HTML y
ejecutado en el servidor.



¿Qué se puede hacer con PHP?

- Scripts en el lado del servidor
- Scripts en la línea de comandos
- Aplicaciones de interfaz gráfica



¿Qué características tiene PHP?

Versiones para la mayoría de los S.O.

Soporta la mayoría de los servidores web

Tipos de programación:

- Procedimental
- Orientada a objetos

Generación de múltiples formatos (HTML, XML, PDF, imágenes, ...)

Múltiples extensiones

Bases de datos:

- Dbase
- Hyperwave
- IBM DB2
- Direct MS-SQL
- MySQL
- ODBC
- Oracle
- PostgreSQL

Protocolos:

- LDAP
- IMAP
- SNMP
- NNTP
- POP3
- HTTP

Instalación de PHP

Un buen **tutorial** para la instalación:

<http://php.net/manual/en/install.php>

La mejor opción (para usuarios no avanzados) es usar **XAMPP**.



Incluye **Apache, PHP, Perl y MySQL**

<http://apachefriends.org/en/xampp.html>

El primer script

```
<html>
<head>
    <title>primer script</title>
</head>
<body>
    <?php
        echo "<p>Hola mundo</p>";
    ?>
    <p>Más contenido HTML</p>
</body>
</html>
```

The diagram illustrates the structure of a PHP script within an HTML document. It shows the following code blocks:

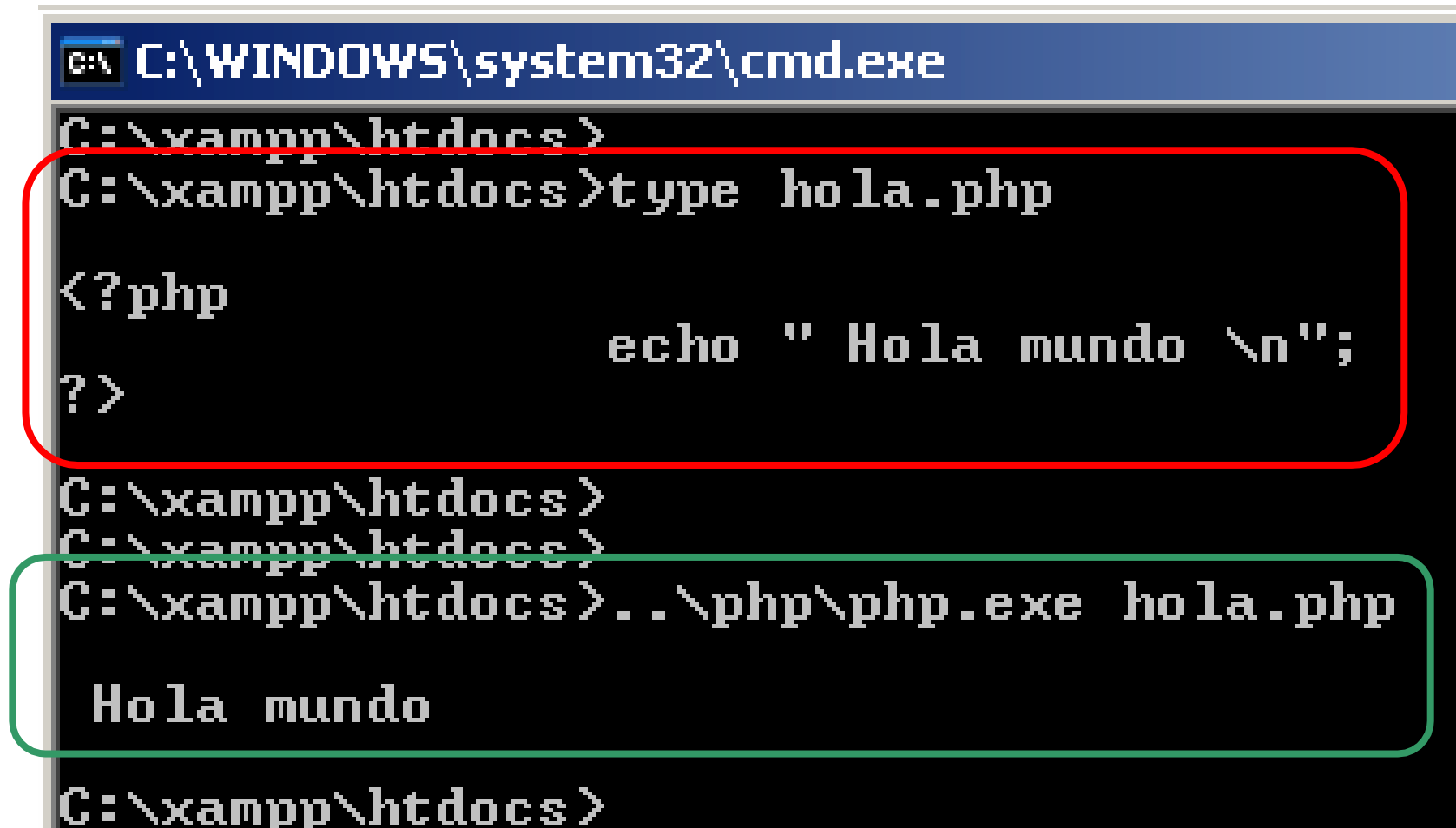
- Etiqueta de inicio**: A callout pointing to the `<body>` tag.
- Orden a ejecutar**: A callout pointing to the PHP code block `<?php echo "<p>Hola mundo</p>"; ?>`.
- Etiqueta de fin**: A callout pointing to the `</body>` tag.

Ejercicio: probar el primer script

- + Crea un documento de texto con extensión .php en el espacio web de tu servidor.
- + Copia el siguiente contenido.
- + Accede desde el navegador a este script alojado en tu servidor
<http://127.0.0.1/hola.php>

```
<html>
<body>
    <?php
        echo "<p>Hola mundo</p>";
    ?>
</body>
</html>
```


Ejecutar en la ventana de comandos



```
C:\WINDOWS\system32\cmd.exe
C:\xampp\htdocs>
C:\xampp\htdocs>type hola.php

<?php
                                echo "Hola mundo \n";
?>

C:\xampp\htdocs>
C:\xampp\htdocs>
C:\xampp\htdocs>..\php\php.exe hola.php

Hola mundo

C:\xampp\htdocs>
```

The image shows a Windows command prompt window with a blue title bar. The title bar text is "C:\WINDOWS\system32\cmd.exe". The command prompt shows the following sequence of commands and output:

- The prompt is at "C:\xampp\htdocs>".
- The command "type hola.php" is entered and executed.
- The output of the command is displayed, enclosed in a red rounded rectangle: `<?php`, `echo "Hola mundo \n";`, and `?>`.
- The prompt returns to "C:\xampp\htdocs>".
- The command "C:\xampp\htdocs>..\php\php.exe hola.php" is entered and executed.
- The output of the command is displayed, enclosed in a green rounded rectangle: "Hola mundo".
- The prompt returns to "C:\xampp\htdocs>".

Sintaxis básica

Separación de instrucciones con ;

La etiqueta de fin ?> también equivale a fin de instrucción

Comentarios:

// ...

/*...*/

Insertar código PHP

```
<?php  
    echo "Hola Mundo"  
?>
```



Las tres son
equivalentes

```
<? echo "Hola Mundo" ?>
```

```
<script language="php">  
    echo "Hola Mundo"  
</script>
```

Variables

Comienzan por \$

Son sensibles a mayúsculas y minúsculas

Empezar con una letra o un carácter de subrayado (después del \$)

Pueden contener cualquier número de:

- letras: a-zA-Z
- números: 0-9
- caracteres de subrayado: _

Tipos de datos

Tipos **escalares**:

- Boolean, integer, float, string

Tipos **compuestos**:

- Array, object

En PHP **no** se define el **tipo** de las variables

Datos escalares. Valores numéricos

Podemos trabajar de forma similar a C/C++

```
<html><body>
```

```
<?php
```

```
    $num = 7;
```

```
    $cont = 9;
```

```
    $suma = $num + $cont;
```

```
    echo "<p>SUMA = $suma</p>";
```

```
?>
```

```
</body></html>
```

Ejercicio: probar el script anterior

- + Crea un documento de texto con extensión .php en el espacio web de tu servidor.
- + Copia el siguiente contenido.
- + Accede desde el navegador a este script
<http://127.0.0.1/suma.php>

```
<html><body>
<?php
    $num = 7;
    $cont = 9;
    $suma = $num + $cont;
    echo "<p>SUMA = $suma</p>";
?>
</body></html>
```

Datos escalares. Cadenas

Pueden estar delimitadas por “ o por ‘

```
<html><body>
```

```
<?php
```

```
    $cad = "una cadena de caracteres";
```

```
    $str = "otra cadena";
```

```
    $concat = $cad . $str;
```

```
    echo "<p>CONCAT = $concat</p>";
```

```
?>
```

```
</body></html>
```


Datos escalares. Cadenas

Para generar una fecha (con hora, min, seg):

```
$fecha = date("Y\-n\-j H\:i\:s");  
echo "$fecha";
```

Para trocear una cadena según un carácter:

```
$trocitos = explode("#", $cadena);  
echo "primer trozo: $trocitos[0] ";
```

Ejercicio: mostrar la fecha

+ Haz un script que cada vez que se acceda a él muestre la fecha y la hora.

+ Accede desde el navegador al script

<http://127.0.0.1/fecha.php>

```
<?php
    $fecha = date("Y\-n\-j H\:i\:s");
    echo "$fecha";
?>
```

Funciones de manejo de cadena

En la siguiente dirección encontraremos la guía completa de las funciones de cadenas de PHP:

http://www.w3schools.com/PHP/php_ref_string.asp



Arrays indexados

1. Crear el array automáticamente, comenzando en el índice 0:

```
$nombres=array("pepe","juan","luisa","pepa");
```

2. Asignar los índices manualmente:

```
$nombres[0]="pepe";  
$nombres[1]="juan";  
$nombres[2]="luisa";  
$nombres[3]="pepa";
```

Arrays indexados

3. Acceder a los elementos:

```
<?php
```

```
$nombres[0]="pepe";  
$nombres[1]="juan";  
$nombres[2]="luisa";  
$nombres[3]="pepa";
```

```
echo $nombres[0] ." quiere a " . $nombres[3] ;
```

```
?>
```

Ejercicio: Arrays

- + Crea un documento de texto con extensión .php en el espacio web de tu servidor.
- + Copia el siguiente contenido.
- + Accede desde el navegador a este script.

```
<html><body>
<?php
    $nombres[0]="pepe";
    $nombres[1]="juan";
    $nombres[2]="luisa";
    $nombres[3]="pepa";

    echo $nombres[0] ." quiere a " . $nombres[3] ;
?>
</body></html>
```

Eliminar elementos de un array

Existe la función **unset()** para eliminar la definición de una variable escalar o un array:

```
<?php
```

```
$nombres=array("pepe","juan","luisa","pepa");
```

```
unset( $nombres[0] );
```

```
unset( $nombres );
```

```
?>
```

Arrays multidimensionales

Crearla con array; acceso con la sintaxis de corchetes:

```
<?php
```

```
$matriz = array(  
    "mdim"=>array(0=>5,1=>9,2=>35)  
);
```

```
echo $matriz["mdim"][0] ;  
echo $matriz["mdim"][2] ;
```

```
?>
```


Ejemplo. Arrays multidimensionales

C:\WINDOWS\system32\cmd.exe

C:\xampp\htdocs>type array.php

```
<?php
$matriz = array(
    0 => array(5,9,35) ,
    1 => array(4,6,48)
);

echo $matriz[0][0] ;
echo "\n";
echo $matriz[1][2] ;

?>
```

C:\xampp\htdocs>..\php\php.exe array.php

5
48

C:\xampp\htdocs>

Arrays asociativos

1. Definamos un array asociativo para guardar datos de varias personas:

```
$edad=array ("Pepe"=>32, "Juan"=>30, "Luis"=>34) ;
```

2. Creamos un array idéntico al anterior, pero usando una sentencia para cada elemento:

```
$edad['Pepe'] = "32" ;  
$edad['Juan'] = "30" ;  
$edad['Luis'] = "34" ;
```

Arrays asociativos

3. La forma de usar estos arrays en un script es:

```
<?php
```

```
$edad['Pepe'] = "32";
```

```
$edad['Juan'] = "30";
```

```
$edad['Luis'] = "34";
```

```
echo "Pepe tiene " . $edad['Pepe'] . " años.";
```

```
?>
```

Guía completa de funciones de arrays

En la siguiente dirección encontraremos la guía completa de las funciones de manejo de arrays de PHP:

http://www.w3schools.com/PHP/php_ref_array.asp



Constantes

Las constantes en PHP tienen que ser definidas por la función *define()* y además no pueden ser redefinidas con otro valor

```
<?php
    define("CONSTANTE", "hello world.");
    echo CONSTANTE;
?>
```

Expresiones y operadores

Se parecen mucho a los de C/C++

Los incrementos, asignaciones, etc, son iguales.

Existen operadores para concatenar cadenas (.) y para comprobar la identidad y tipo (===)

Expresiones y operadores

Ejemplos:

`x=2; y=x+3;`

Operadores aritméticos

`x=5; x++; x--;`

`x=y; x=x/y;`

Operadores de asignación

`5==8` es falso
`5<=8` es cierto

Operadores de comparación

`x=6; z=3;`
`(x<10 && z>1)` es cierto

Operadores lógicos

Estructuras de control

De cuando en cuando, debemos ejecutar diferentes porciones de código en función de ciertas **condiciones**. En estos casos debemos usar las **sentencias condicionales**.

Para ejecutar diferentes porciones de código de forma **repetida**, en lugar de replicar varias veces el mismo código, debemos usar las **sentencias de tipo bucle**.

Estructuras de control condicional

```
<html>
<body>

<?php

$d=date("D");
if ($d=="Fri") {
    echo "Buen finde!";
}
?>

</body>
</html>
```

```
<html><body>

<?php

$d=date("D");
if ($d=="Fri"){
    echo "Buen finde!";
}else{
    echo "Buenos dias!";
}
?>

</body></html>
```

Estructuras de control iterativas

```
<html><body>

<?php
$i=1;
while($i<=5) {
    echo "Iteración " . $i ;
    echo "<br />";
    $i++;
}
?>

</body></html>
```

```
<html><body>

<?php
$i=1;
do {
    $i++;
    echo "Iteración " . $i ;
    echo "<br />";
}
while ($i<=5);
?>

</body></html>
```

Estructuras de control iterativas

```
<html><body>
```

```
<?php
for ($i=1; $i<=5; $i++) {
    echo " Iteración " . $i ;
    echo "<br />";
}
?>
```

```
</body></html>
```

```
<html><body>
```

```
<?php
$x=array("uno","dos","3");
foreach ($x as $valor) {
    echo $valor . "<br />";
}
?>
```

```
</body></html>
```

```
<?php
$x=array("uno"=>1,"dos"=>2);
foreach ($x as $clave => $valor) {
    echo " $clave= $valor ";
}
?>
```

Ejercicio: estructuras de control

- + Haz un script que repita 10 veces el siguiente proceso:
 - + mostrar el número de iteración
 - + si la iteración es menor que 5, que muestre el mensaje anterior en negrita
 - + si la iteración es mayor que 5, que muestre el mensaje en cursiva
- + Accede desde el navegador a este script
<http://127.0.0.1/bucle.php>

Funciones en PHP

En la siguiente dirección encontraremos la guía completa de las funciones incluidas en PHP:

<http://www.w3schools.com/PHP/default.asp>



Funciones en PHP. Sintaxis

```
<html><body>
```

```
<?php
```

```
    function escribe() {  
        echo "Hola";  
    }
```

```
    echo "Vamos a saludar."  
    escribe() ;
```

```
?>
```

```
</body></html>
```

Funciones en PHP. Parámetros

```
<html><body>
```

```
<?php
```

```
function escribe($nombre,$edad) {  
    echo $nombre . " tiene ";  
    echo $edad . " años <br />";  
}
```

```
escribe("Angel","78");  
escribe("Pepa",47);
```

```
?>
```

```
</body></html>
```

Ejercicio: Funciones

- + Crea un documento de texto con extensión .php en el espacio web de tu servidor.
- + Copia el siguiente contenido.
- + Accede desde el navegador a este script.

```
<html><body>
<?php
    function escribe($nombre,$edad) {
        echo $nombre . " tiene ";
        echo $edad . " años <br />";
    }

    escribe("Angel","78");
    escribe("Pepa","47");

?>
</body></html>
```


Funciones en PHP. Devolver valores

```
<html>
```

```
<body>
```

```
<?php
```

```
    function suma($x,$y) {  
        $total=$x+$y;  
        return $total;  
    }
```

```
    $retorno = suma(10,13);  
    echo "1 + 16 = " . suma(1,16);
```

```
?>
```

```
</body>
```

```
</html>
```

E/S con ficheros

Estarán alojados en el servidor.

Necesitaremos:

abrir, leer, escribir, cerrar, comprobar que existe el fichero

```
fopen (archivo, modo)
```

```
fgets (manejador_archivo, longitud)
```

```
fputs (manejador_archivo, cadena)
```

```
fclose (manejador_archivo)
```

```
file_exists (fichero)
```

E/S con ficheros. Apertura y lectura

```
<?php
    $archivo = fopen("fichero.txt" , "r");


    if ($archivo) {
        while (!feof($archivo)) {
            $linea = fgets($archivo, 255);
            echo $linea;
        }
    }

    fclose ($archivo);

?>
```

E/S con ficheros. Escritura

```
<?php
    $archivo = fopen("fichero.txt" , "w");
    fputs ($archivo, "Hola Mundo");
    fclose ($archivo);
?>
```



Comprobar la existencia

```
<?php
    if ( file_exists("fichero.txt") ) {
        echo "El fichero existe";
    } else {
        echo "El fichero NO existe";
    }
?>
```

Ejercicio: E/S con ficheros

- + Haz un programa que abra un fichero de texto y muestre el contenido.
- + Al mismo tiempo, que haga una copia sobre otro fichero (vaya escribiendo cada línea leída en el segundo fichero).
- + Accede desde el navegador a este script
<http://127.0.0.1/fichero.php>

include() / require()

Ambos incluyen y evalúan un archivo:

```
require ("libreria.inc") ;  
include ("libreria.inc") ;
```

En caso de error en ese archivo:

- **require** produce un error fatal
- **include** sólo produce un *warning* (y sigue)

Existe **require_once()** e **include_once()**

Usando require()

```
<?php

function mifuncion(){
    echo "mensaje \n";
}

function miSuma($a,$b){
    $a = $a + $b;
    echo "suma = $a \n";
}

?>
```

libreria.php

```
<?php
require('libreria.php');

mifuncion();

echo "\n otro mensaje \n";

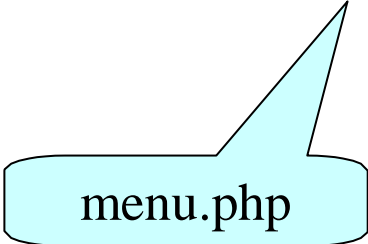
$sum1=7;
$sum2=9;
miSuma ($sum1, $sum2);

?>
```

ejemplo.php

Usando include()

```
<a href=http...>opción 1</a>
<a href=http...>opción 2</a>
<a href=http...>opción 3</a>
<a href=http...>opción 4</a>
<a href=http...>opción 5</a>
```



menu.php

```
<html><body>
```

```
<?php
include ("menu.php") ;
?>
```

```
<p>contenido html</p>
```

```
</body></html>
```



ejemplo.php

Formularios HTML

Proceso:

- El cliente pide una **página HTML** que contiene un **formulario**
- Lo **rellena** y **envía** los datos.
- La petición se hace a un **script PHP** que los recibe, los **procesa** y genera una página HTML que se envía al cliente.

Formularios HTML

El formulario puede ser como el siguiente

```
<html><body>
```

```
<form action="procesar.php" method="post">
```

```
Nombre: <input type="text" name="nombre" />
```

```
Edad: <input type="text" name="edad" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body></html>
```



formulario.html 50

Formularios HTML

El script que recibe y procesa los datos:

```
<html><body>
Hola
    <?php
        echo $_POST["nombre"] ;
    ?>
Tienes
    <?php
        echo $_POST["edad"] ;
    ?>
años.
</body></html>
```



procesar.php 51

Ejercicio: formularios

- + Crea un formulario en un archivo HTML que envíe los datos introducidos a un script PHP que reciba los datos y los muestre.
- + Escribir los datos de cada envío en un archivo de texto (cada envío en una línea nueva).
- + Probar el funcionamiento:

<http://127.0.0.1/formulario.html>

La función \$_GET

Se usa para recoger valores enviados usando **method="get"**

Al enviar los datos, se ven en la URL:

<http://www.servidor.com/procesar.php?nombre=Pepe>

Hay un límite máximo de 100 caracteres

La función \$_POST

Se usa para recoger valores enviados usando **method="post"**

El envío de los datos es invisible:

<http://www.servidor.com/procesar.php>

No hay límite en la cantidad de datos

Cookies

Resolver el **problema de la falta de estado en la navegación** a través de las páginas web.

Almacenar información que queda registrada en el navegador, permitiendo identificarlo después.

El manejo de *cookies* en PHP se realiza mediante el uso de la función *setcookie()*

Cookies

La función define una *cookie* que es enviada junto con el resto de la información de la cabecera.

Deben ser enviadas antes de cualquier etiqueta de HTML, por lo tanto deberemos realizar la llamada a estas funciones antes de cualquier etiqueta `<HTML>` o `<HEAD>`

Esta es una restricción de las *cookies*, no de PHP.

Cookies. Ejemplo (I)

Primer paso: desde un formulario enviamos información a un script que la recibirá y establecerá la *cookie*

```
<html><body>
```

Introduzca su nombre:

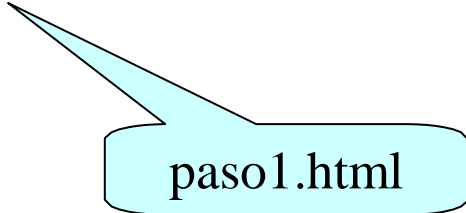
```
<FORM ACTION="paso2.php" METHOD="GET">
```

```
<INPUT TYPE="text" NAME="nombre"><BR>
```

```
<INPUT TYPE="submit" VALUE="Enviar">
```

```
</FORM>
```

```
</body></html>
```



paso1.html

Cookies. Ejemplo (II)

Segundo paso: el script PHP establece la *cookie* (leyendo el nombre enviado desde el formulario)

```
<?php
    setcookie("ejemusuario", $_GET['nombre'],
time()+10,"/","");
?>
```

```
<html><body>
```

Se ha establecido la cookie `ejemusuario` con el valor:

```
<? print $_GET['nombre']; ?>
```

que será válida durante 10 segundos.

```
<a href=paso3.php>Ir al paso de comprobar el valor </a>
</body> </html>
```

paso2.php

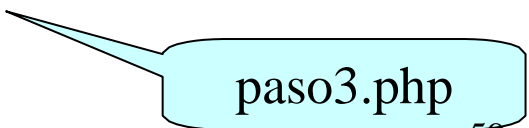
Cookies. Ejemplo (III)

Tercer paso: para comprobar, accedemos al nombre de la *cookie* para ver si aún está activa o ya ha caducado

```
<html><body>
```

```
La cookie de nombre  ejemusuario
```

```
<?php
if( $_COOKIE['ejemusuario'] ) {
    echo "tiene el valor: ";
    echo $_COOKIE['ejemusuario'] ;
}else{
    echo "No existe o ha caducado.";
}
?>
```



paso3.php

59

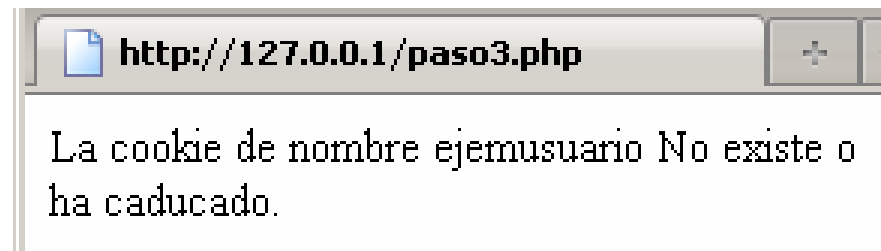
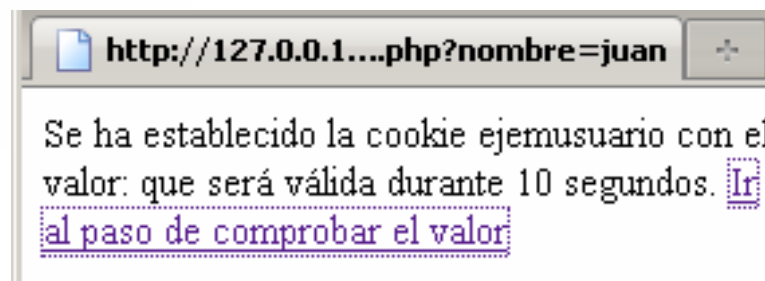
```
</body></html>
```

Ejercicio: cookies

- + Usa el sistema de tres archivos anteriores para comprobar el funcionamiento de las cookies.
- + Establece como tiempo de vida de la cookie 5 segundos solamente



Introduzca su nombre:



Clases y objetos

Una clase es una colección de variables y métodos.

Las variables miembro se definen con *var*

Los métodos (funciones) se definen con *function*

Clases y objetos

```
<?php
class Carrito {
    var $items;

    function agregar_item($tipo, $num) {
        $this->items[$tipo] += $num;
    }

    function retirar_item($tipo, $num) {
        if ($this->items[$tipo] > $num) {
            $this->items[$tipo] -= $num;
            return true;
        } elseif ($this->items[$tipo] == $num) {
            unset($this->items[$tipo]);
            return true;
        } else {
            return false;
        }
    }
}

?>
```

Clases y objetos

Como ejemplo creamos dos carritos. Al primero se le añade 1 item del artículo ordenador. Al segundo se le añaden 3 del artículo disco:

```
<?php
    $carrito = new Carrito;
    $carrito->agregar_item("ordenador", 1);

    $otro_carrito = new Carrito;
    $otro_carrito->agregar_item("disco", 3);
?>
```

Ejercicio: Clases

- + Ampliar la clase carrito con una función miembro para consultar el número de elementos de cada tipo que hay en el carrito.
- + Crear un carrito y añadir varios elementos
- + Posteriormente mostrar el número de elementos que hay de cada tipo en el carrito.

Ejercicio: Clases

```
C:\WINDOWS\system32\cmd.exe
C:\xampp\htdocs>type clase.php
<?php
class Carrito {
    var $items;

    function agregar_item($tipo, $num) {
        $this->items[$tipo] += $num;
    }

    function retirar_item($tipo, $num) {
        if (<$this->items[$tipo] > $num) {
            $this->items[$tipo] -= $num;
            return true;
        } elseif (<$this->items[$tipo] == $num) {
            unset(<$this->items[$tipo]);
            return true;
        } else {
            return false;
        }
    }

    function consultar($tipo) {
        return < $this->items[$tipo] >;
    }
}
?>

<?php
    $carrito = new Carrito;
    $carrito->agregar_item("ordenador", 2);

    $otro_carrito = new Carrito;
    $otro_carrito->agregar_item("disco", 3);

    echo $carrito->consultar("ordenador" );
    echo "\n";
    echo $otro_carrito->consultar("disco" );
?>

C:\xampp\htdocs>..\php\php.exe clase.php

2
3

C:\xampp\htdocs>
```

Clases y objetos

Para ampliar los conocimientos sobre desarrollo orientado a objetos en PHP, podemos visitar:

<http://www.php-es.com/language.oop5.html>



Servicios Web en PHP

¿Qué vamos a aprender?

1. Crear un servidor de Web Services en PHP
2. Hacer llamadas a WS desde PHP
3. Crear el WSDL automáticamente a partir de una clase PHP

WS en PHP. Introducción

PHP5 soporta WS de forma nativa.

Permite conexión con WSDL o sin él.

Sin embargo tiene algunas limitaciones, ya que no genera automáticamente el documento WSDL.

WS en PHP. Introducción

Existen varios frameworks de WS para PHP, pero los más extendidos son:

- El nativo de PHP5 (el que vamos a usar)
- NuSOAP *<http://nusoap.sourceforge.net>*

Creación de un servidor

```
<?php
```

```
function suma($a,$b) {  
    return $a+$b;  
}
```

```
//Sin WSDL -> uri es obligatorio  
$server = new SoapServer(null,  
array('uri'=>""));  
$server->addFunction("suma");  
$server->handle();
```

```
?>
```



simple/server.php

Creación de un cliente

```
<?php
$url="http://127.0.0.1/simple/server.php";

//Como no hay WSDL añadimos 'uri'
$cliente = new
SoapClient(null,array('location'=>$url,'uri'=>""));

//Llamamos a la operación suma
//Problema: tenemos que saber nosotros que existe!
$suma = $cliente->suma(2,3);
print("La suma es ".$suma);

?>
```



simple/client.php

Ejercicio: WS en PHP

+ Modificar el WS anterior y añadir una operación nueva que reciba como parámetro una cadena de caracteres y devuelva un saludo.

```
C:\xampp\htdocs\simple>type client.php
<?php
$url="http://127.0.0.1/simple/server.php";

//Como no hay WSDL añadimos 'uri'
$cliente = new SoapClient(null,array('location'=>$url,'uri'=>""));

//Llamamos a la operación suma
//Problema: tenemos que saber nosotros que existe!
$suma = $cliente->suma(2,3);
print("La suma es ".$suma);

?>

C:\xampp\htdocs\simple>
C:\xampp\htdocs\simple>
C:\xampp\htdocs\simple>..\..\php\php.exe server.php

C:\xampp\htdocs\simple>
C:\xampp\htdocs\simple>..\..\php\php.exe client.php
La suma es 5
C:\xampp\htdocs\simple>
```


Crear un servicio con WSDL


Pasos:

- 1) Crear la clase documentada
- 2) Generar automáticamente el WSDL
- 3) Crear el servidor
- 4) Comprobar con el cliente

1) Crear la clase

```
<?php
/**
 * Ejemplo de clase para el ejemplo
 */
class EjemploWS{
/**
 * Dice hola
 * @param string $nombre persona
 * @return string saludo
 */
public function diHola($nombre){
return "Hola ".$nombre."!";
}
/**
 * Dice adios
 * @param string $otro persona
 * @return string saludo
 */
public function diAdios($otro){
return "Adios, señor ".$otro."!";
}
```

```
/**
 * Sumar
 * @param int $a Primer integer
 * @param int $b Segundo integer
 * @return int Result. de la suma
 */
public function suma($a, $b)
{
    return $a + $b;
}
?>
```



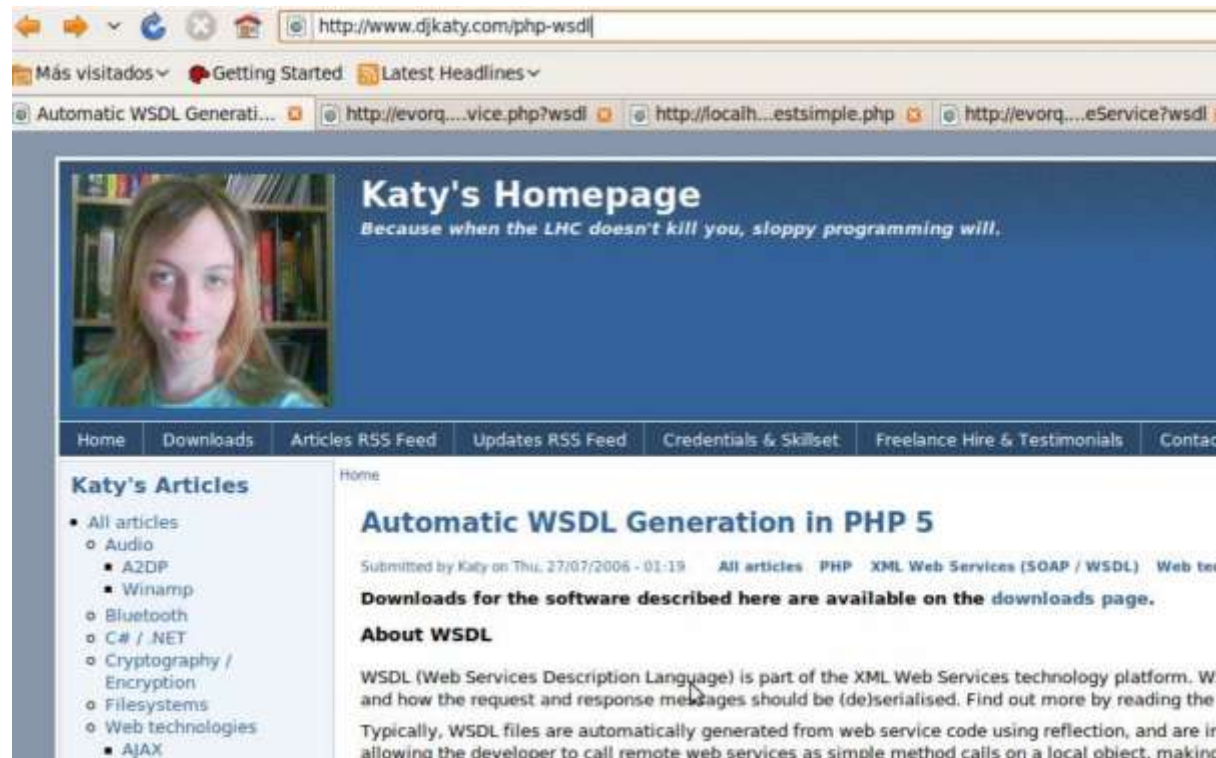
complejo/ejemplo/EjemploWS.php

¿Y el WSDL?

¿Cómo podemos generar el WSDL sin complicarnos?

Gracias a nuestra amiga Katy:

<http://www.djkaty.com/php-wsdl>



2) Generar el WSDL (I)

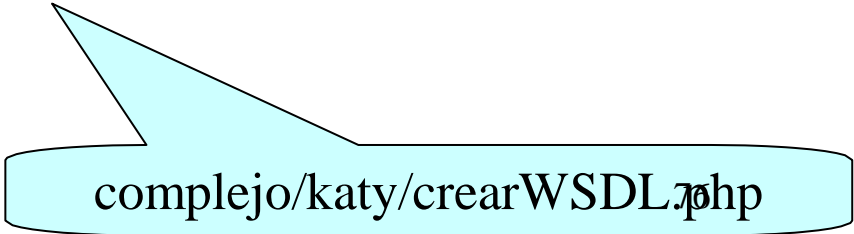
```
<?php
require_once("../classes/WsdlDefinition.php");
require_once("../classes/WsdlWriter.php");

$def = new WsdlDefinition();
$def->setDefinitionName("EjemploWS");

//Ruta de la clase del WS
$def->setClassFileName("../..ejemplo/EjemploWS.php");
$def->setWsdlFileName("EjemploWS.wsdl"); //fichero a crear

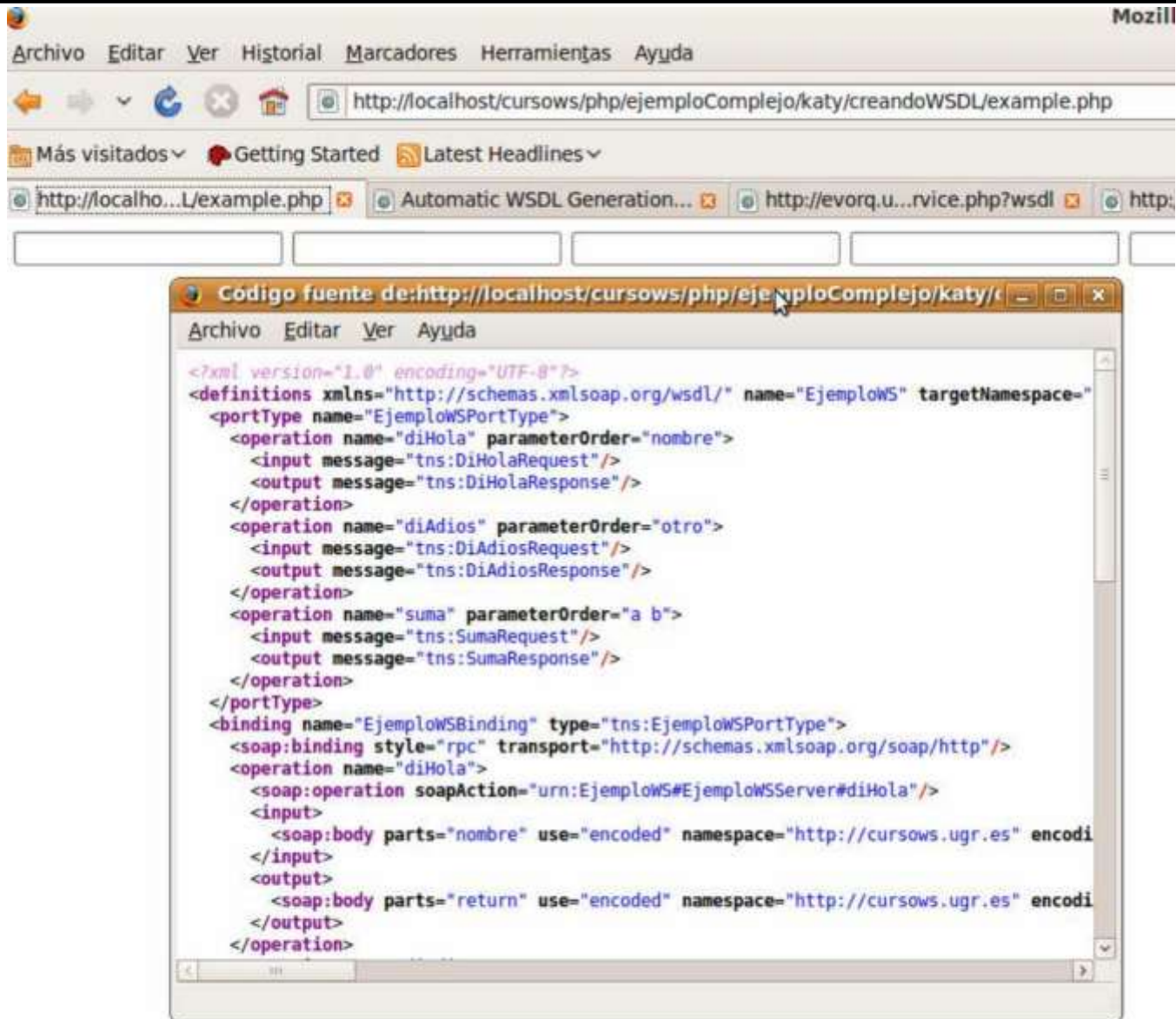
$def->setNamespace("http://cursows.ugr.es"); //Namespace
$def->setEndPoint("
http://127.0.0.1/complejo/ejemplo/service.php");
$wsdl = new WsdlWriter($def);

print $wsdl->classToWsdl();
?>
```



complejo/katy/crearWSDL.php

2) Generar el WSDL (II)



```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" name="EjemploWS" targetNamespace="
  <portType name="EjemploWSPortType">
    <operation name="diHola" parameterOrder="nombre">
      <input message="tns:DiHolaRequest"/>
      <output message="tns:DiHolaResponse"/>
    </operation>
    <operation name="diAdios" parameterOrder="otro">
      <input message="tns:DiAdiosRequest"/>
      <output message="tns:DiAdiosResponse"/>
    </operation>
    <operation name="suma" parameterOrder="a b">
      <input message="tns:SumaRequest"/>
      <output message="tns:SumaResponse"/>
    </operation>
  </portType>
  <binding name="EjemploWSBinding" type="tns:EjemploWSPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="diHola">
      <soap:operation soapAction="urn:EjemploWS#EjemploWSServer#diHola"/>
      <input>
        <soap:body parts="nombre" use="encoded" namespace="http://cursows.ugr.es" encodi
      </input>
      <output>
        <soap:body parts="return" use="encoded" namespace="http://cursows.ugr.es" encodi
      </output>
    </operation>
```

3) Desplegar el WSDL

```
<?php
require_once('EjemploWS.php');

$soap = new SoapServer('EjemploWS.wsdl');
$soap->setClass('EjemploWS');
$soap->handle();

//Nunca imprimir texto en los servidores!

?>
```



complejo/ejemplo/service.php

4) Comprobar el servicio

```
<?php
```

```
$client = new SoapClient ('EjemploWS.wsdl', array('trace'=>1) ; //Creamos el cliente y le damos la posibilidad de hacer trazas
```

```
echo "OPERACIONES DEL WEBSERVICE: <p>";
```

```
var_dump($client->__getFunctions()); //Imprimimos sus funciones
```

```
echo "</p>";
```

```
$nombre = "Locke";
```

```
$resultado=$client->diHola($nombre); //llamamos al servicio
```

```
print("SALIDA DEL WEBSERVICE: ".$resultado);
```

```
//al poner array('trace'=>1) arriba, podemos mostrar los mensajes SOAP
```

```
echo '<b>Petici&oacute;n</b>';
```

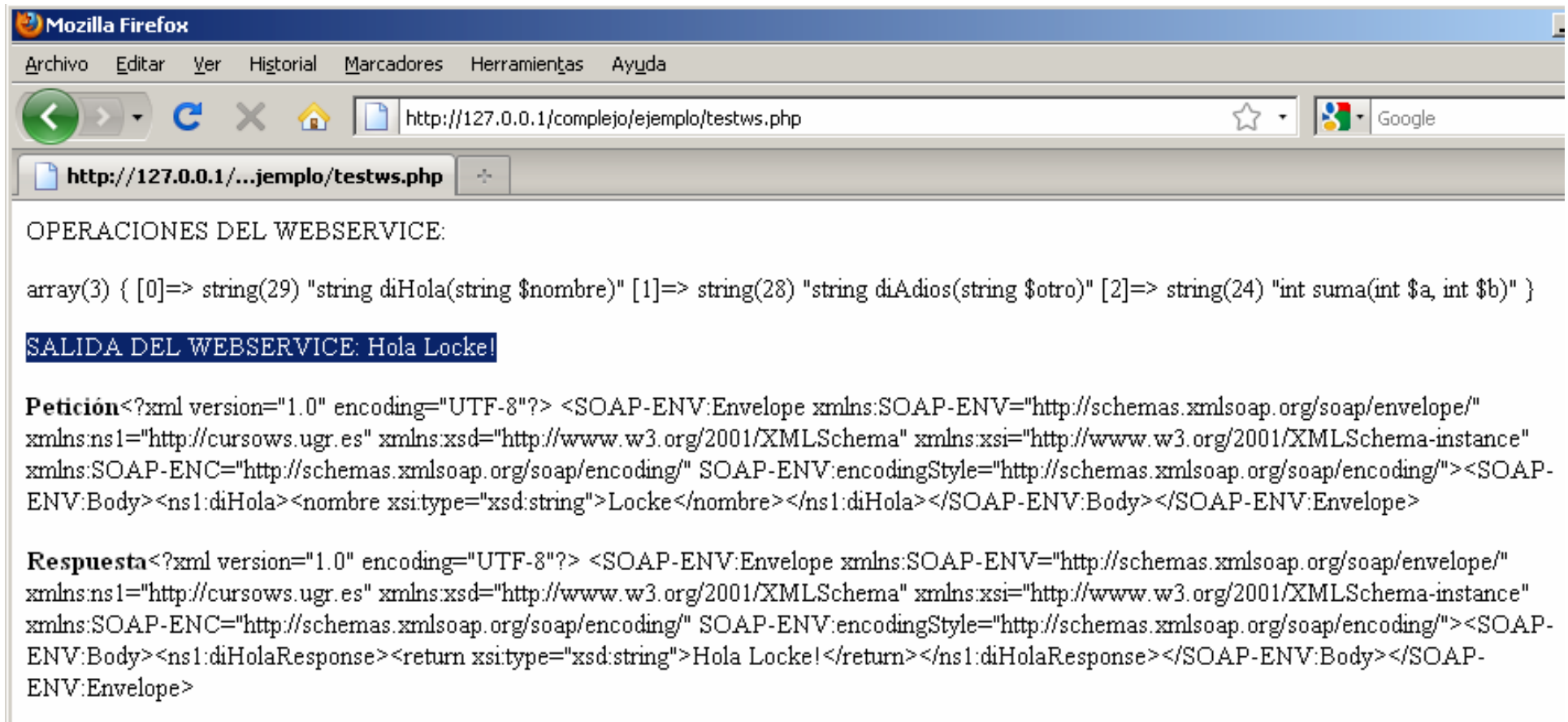
```
echo htmlspecialchars($client->__getLastRequest(), ENT_QUOTES);
```

```
echo '</p><p><b>Respuesta</b>';
```

```
echo htmlspecialchars($client->__getLastResponse(), ENT_QUOTES);
```

```
?>
```

Ejecución del cliente



Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://127.0.0.1/complejo/ejemplo/testws.php

http://127.0.0.1/...jemplo/testws.php

OPERACIONES DEL WEBSERVICE:

```
array(3) ( [0]=> string(29) "string diHola(string $nombre)" [1]=> string(28) "string diAdios(string $otro)" [2]=> string(24) "int suma(int $a, int $b)" )
```

SALIDA DEL WEBSERVICE: Hola Locke!

Petición`<?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://cursows.ugr.es" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:diHola><nombre xsi:type="xsd:string">Locke</nombre></ns1:diHola></SOAP-ENV:Body></SOAP-ENV:Envelope>`

Respuesta`<?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://cursows.ugr.es" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:diHolaResponse><return xsi:type="xsd:string">Hola Locke!</return></ns1:diHolaResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>`

Ejercicio: WS en PHP

+ Modificar la clase para añadir una nueva operación que multiplique dos números.

¿Funciona? ¿Actualiza el WSDL?

Fichero xampp/php/php.ini

```
; (time to live) Sets the number of second while  
cached file will be used  
; instead of original one.  
; http://php.net/soap.wsdl-cache-ttl  
soap.wsdl_cache_ttl=86400
```

WS en PHP. Consideraciones

- 1) Si vemos el WSDL generado, las URIs de las operaciones tienen '#' => hay que quitarlas a mano, ya que no cumple el estándar y otros programas podrían quejarse
- 2) Hay que tener en cuenta el tiempo de caché de los WSDLs
- 3) Los servidores no pueden imprimir texto

Conexión con bases de datos MySQL

Obtener **registros de una base de datos** utilizando PHP es muy común.

La tarea se basa en **tres pasos**:

- Conectar con la base de datos
- Hacer una consulta SQL
- Extraer la información



Conexión con bases de datos MySQL

Conectar con la base de datos MySQL:

```
$conexion = mysql_connect("localhost", "usuario", "clave");  
  
mysql_select_db("basededatos", $conexion);
```



Conexión con bases de datos MySQL

Hacer una consulta SQL:

```
$seleccion = "SELECT * FROM tabla";
```

```
$resultado = mysql_query($seleccion, $conexion);
```

```
$numResultados = mysql_num_rows($resultado);
```



Conexión con bases de datos MySQL

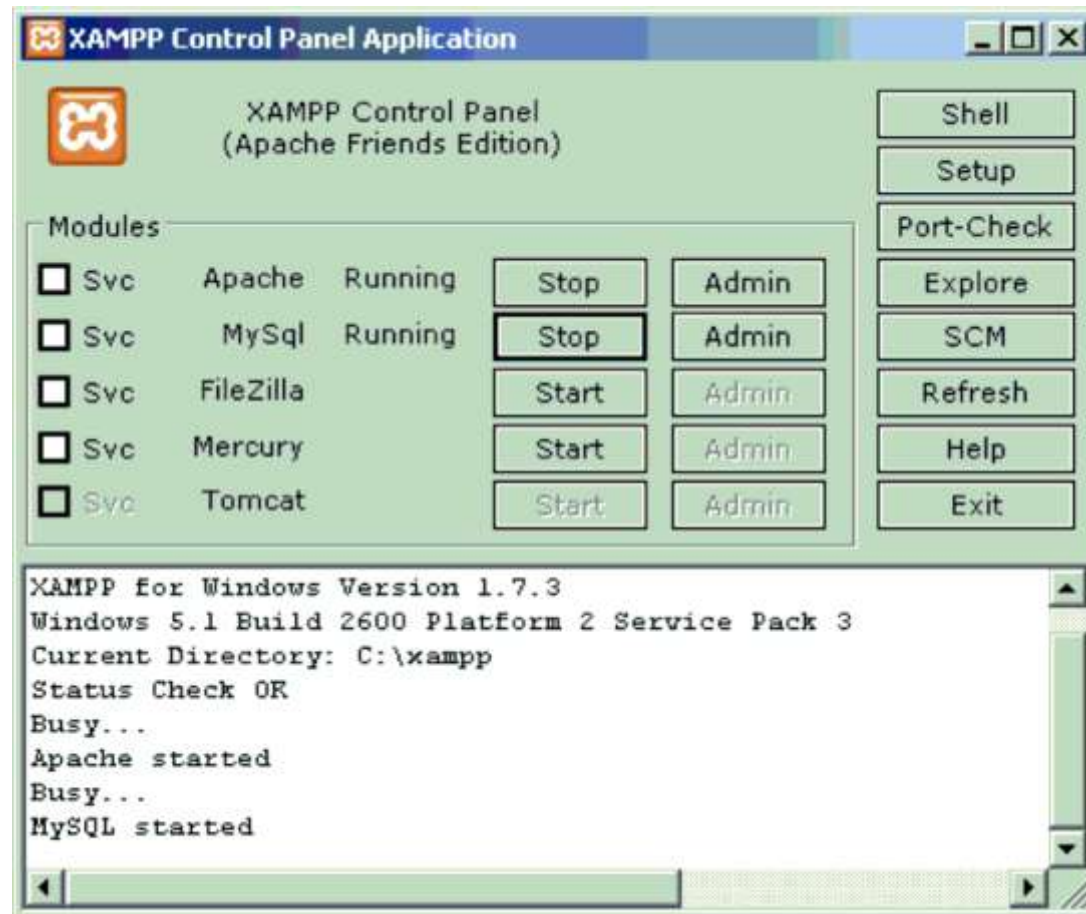
Extraer la información:

```
if ($numResultados > 0) {  
  
    while ( $fila = mysql_fetch_assoc($resultado) ) {  
  
        echo " Id " . $fila['id'] . " <br>";  
  
        echo " Nombre " . $fila['nombre'] . "<br>";  
  
    }  
  
}
```



Ejemplo: PHP + MySQL (I)

Lanzar Apache y MySQL



Ejemplo: PHP + MySQL (II)

1. Entrar en MySQL

2. Crear una base de datos

3. Crear una tabla

```
C:\xampp>.\mysql\bin\mysql.exe -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.41 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current
mysql> create database agenda;
Query OK, 1 row affected (0.03 sec)

mysql> use agenda;
Database changed
mysql> create table contactos(nombre varchar(100), tlf int);
Query OK, 0 rows affected (0.09 sec)

mysql>
mysql> show tables;
+-----+
| Tables_in_agenda |
+-----+
| contactos         |
+-----+
1 row in set (0.05 sec)

mysql>
mysql> describe contactos;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nombre | varchar(100) | YES  |     | NULL    |       |
| tlf    | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql>
```


Ejemplo: PHP + MySQL (III)

4. Insertar datos

```
mysql>
mysql> insert into contactos(nombre,tlf) values ("pepe",954845637);
Query OK, 1 row affected (0.03 sec)

mysql> insert into contactos(nombre,tlf) values ("juan",912837465);
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> select * from contactos;
+-----+-----+
| nombre | tlf      |
+-----+-----+
| pepe   | 954845637 |
| juan   | 912837465 |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Ejemplo: PHP + MySQL (IV)

```
<?php

$conexion = mysql_connect("127.0.0.1", "root", "");
mysql_select_db("agenda", $conexion);

$seleccion = "SELECT * FROM contactos;";
$resultado = mysql_query($seleccion, $conexion);
$numResultados = mysql_num_rows($resultado);

echo "<br> <br> NumResultados = $numResultados \n";
echo "<br> <br> \n";

if ($numResultados > 0) {
    while ( $fila = mysql_fetch_assoc($resultado) ) {
        echo " Nombre= " . $fila['nombre'] . " <br>";
        echo " Tlf= " . $fila['tlf'] . " <br>";
    }
}

?>
```

Ejemplo: PHP + MySQL (V)



Conectando extrayendo datos de la BD:

NumResultados = 2

Nombre= pepe

Tlf= 954845637

Nombre= juan

Tlf= 912837465

Conexión con bases de datos MySQL

Enlaces para ampliar información:

<http://es.php.net/manual/en/book.mysql.php>

<http://programadorphp.org>

<http://www.zenphp.es>



