

# Todo PHP7

## De novato a experto



# Todo PHP7

## De novato a experto

### Capítulo 1: Introducción a PHP

**Objetivo:** El alumno conocerá los inicios de PHP, su forma básica de trabajo y las herramientas necesarias para instalar PHP y MySQL en la computadora.

#### Introducción:

Con **PHP** se han hecho poderosos imperios, como **Yahoo** y **Facebook** (también Udemey). Comenzaremos desde lo básico, conociendo la historia del lenguaje y su estado actual. Revisaremos la formación de variables, los ciclos, las sentencias condicionales, los arreglos (o vectores), así como la forma de validar formas de HTML. Analizaremos las poderosas funciones para manejar fechas y cadenas. Todo de forma fácil y amena. Los temas que veremos en esta sección son:

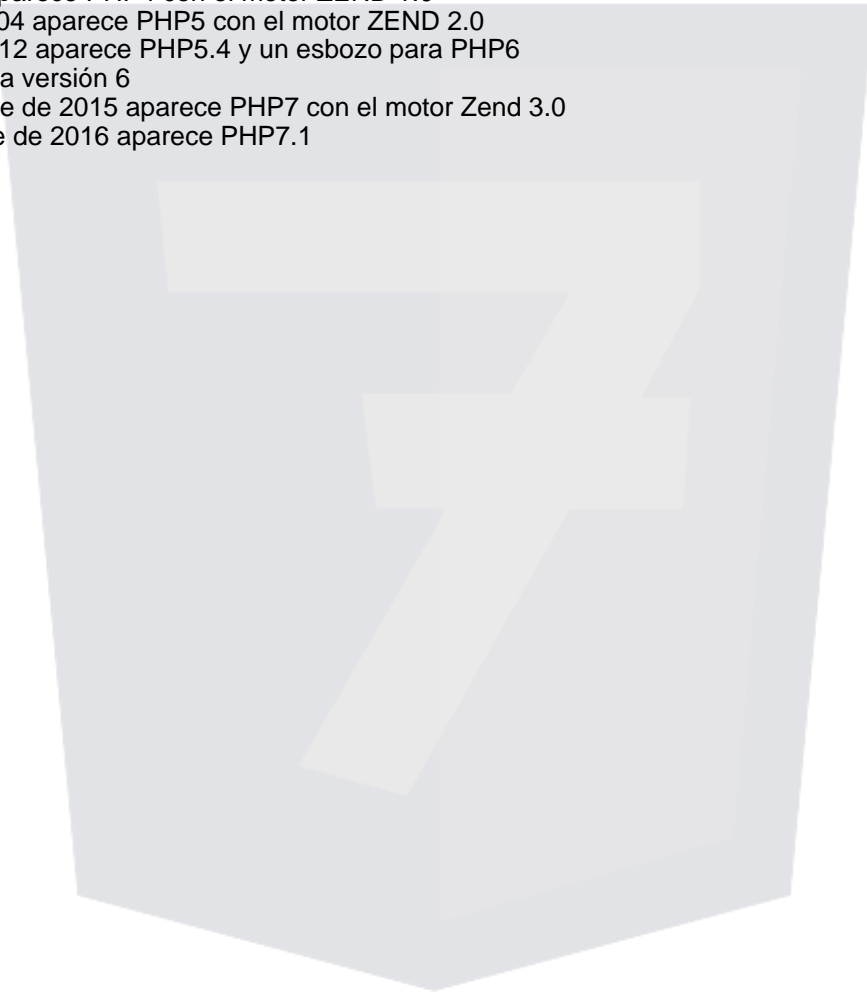
- Breve historia de PHP
- ¿Cómo funciona PHP?
- ¿Qué necesitamos para usar PHP?
- Instalar el Apache + PHP + MySQL
- Instalar MAMP en una computadora con Windows
- Instalar MAMP en una Mac
- Nuevas características de PHP7
- Nuestra primera página: Hola mundo
- Mezclar HTML y PHP
- Comentarios y otras cosas

[www.pacoarce.com](http://www.pacoarce.com)

## 1.1. Breve historia de PHP

PHP o Personal Home Pages fue desarrollado por Rasmus Lerdorf en 1995 como una serie de programas auxiliares para internet. Posteriormente los agrupó y dejó libre, por lo que hoy el lenguaje es libre y no tenemos que pagar un centavo a nadie.

- Zeev Suraski y Andi Gutmans : En 1997 crearon PHP 3
- Mayo del 2000 aparece PHP4 con el motor ZEND 1.0
- 13 de julio de 2004 aparece PHP5 con el motor ZEND 2.0
- 8 de mayo de 2012 aparece PHP5.4 y un esbozo para PHP6
- Nunca apareció la versión 6
- El 3 de noviembre de 2015 aparece PHP7 con el motor Zend 3.0
- El 1 de diciembre de 2016 aparece PHP7.1



*www.pacoarce.com*

## 1.2. ¿Cómo funciona PHP?

Un paso importante para poder manejar una tecnología es comprender su funcionamiento.

1. El cliente realiza una solicitud al servidor de un recurso .php
2. El servidor recibe la solicitud y llama al lenguaje PHP.
3. Donde el servidor encuentre las etiquetas PHP, lo ejecutará.
4. PHP realiza el programa y regresa como etiquetas HTML.
5. El servidor regresa el código HTML (nunca código PHP).
6. El cliente toma el HTML y el navegador lo interpreta.



*www.pacoarce.com*

## 1.3. ¿Qué necesitamos para usar PHP?

- Las rutinas de **PHP** se crean en cualquier editor de texto, ya que es un lenguaje "*interpretado*" (se lee, válida e interpreta hasta que es llamada una rutina por el servidor).
- Puede utilizar un procesador de código gratuito como Aptana, SublimeText o Brackets, o utilizar un procesador pagado como Dreamweaver (cualquier versión).
- Debe de instalar en su computadora Apache-MySQL-PHP con los paquetes **XAMPP**, **MAMP**, **WAMP** o **LAMP**.
- Puede utilizar directamente un servidor remoto. Puede utilizar un FTP o Cpanel para subir sus archivos al servidor.
- Todo archivo que contenga rutinas e instrucciones de **PHP** debe estar contenido entre los símbolos "`<?php`" y "`?>`" y tener la extensión **".php"** para ser ejecutado correctamente.
- Así, un ejemplo de un archivo de **PHP**, que podría ser guardado con el nombre **"hola.php"**.

[www.pacoarce.com](http://www.pacoarce.com)

## 1.4. Instalar el Apache + PHP + MySQL: XAMPP

El paso fundamental es instalar PHP. Para ello necesitamos instalar un sistema operativo web que soporte el lenguaje. Para ello la herramienta básica es el sistema operativo de web Apache. Tanto PHP+Apache+MySQL se les conoce como la "Santa Triada" en el desarrollo en Internet.



Imagen 1.4.1. Entrar a la pagina de ApacheFriends.org

*www.pacoarce.com*

## 1.5. Instalar MAMP en una computadora con Windows

Instalar MAMP en una computadora con Windows es muy sencillo.

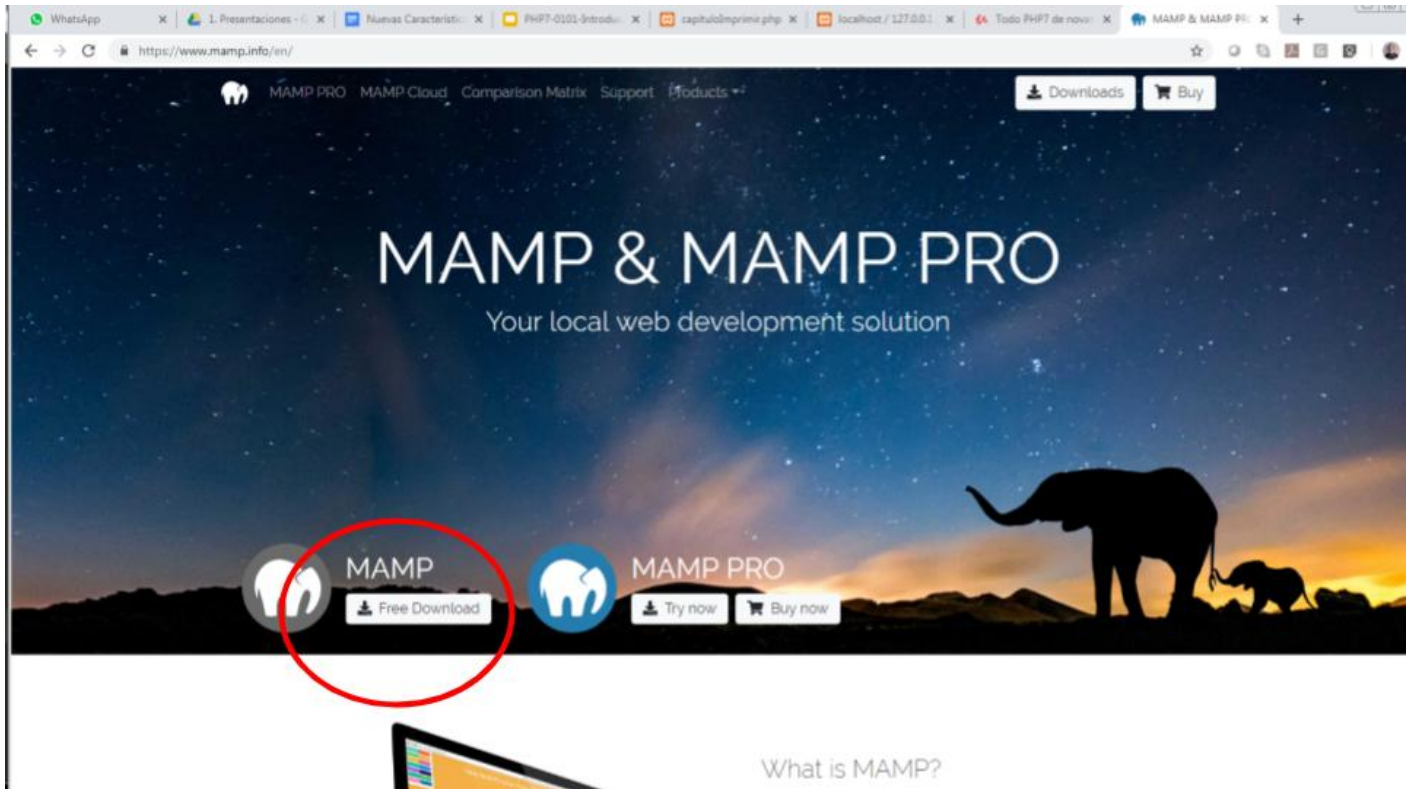


Imagen 1.5.1.

*www.pacoarce.com*

# Todo PHP7

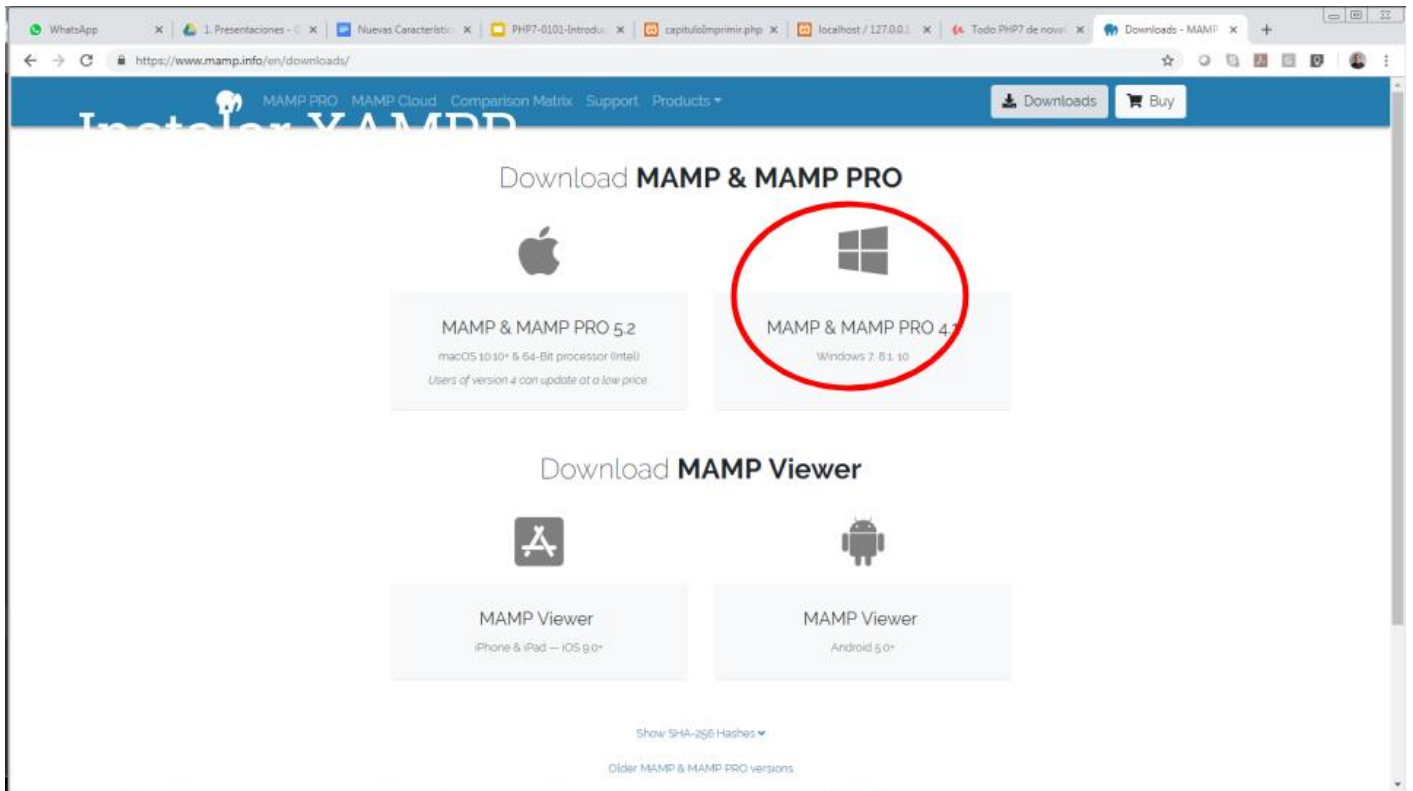


Imagen 1.5.2.

[www.pacoarce.com](http://www.pacoarce.com)



## 1.6. Instalar MAMP en una Mac

Instalar MAMP en una Mac

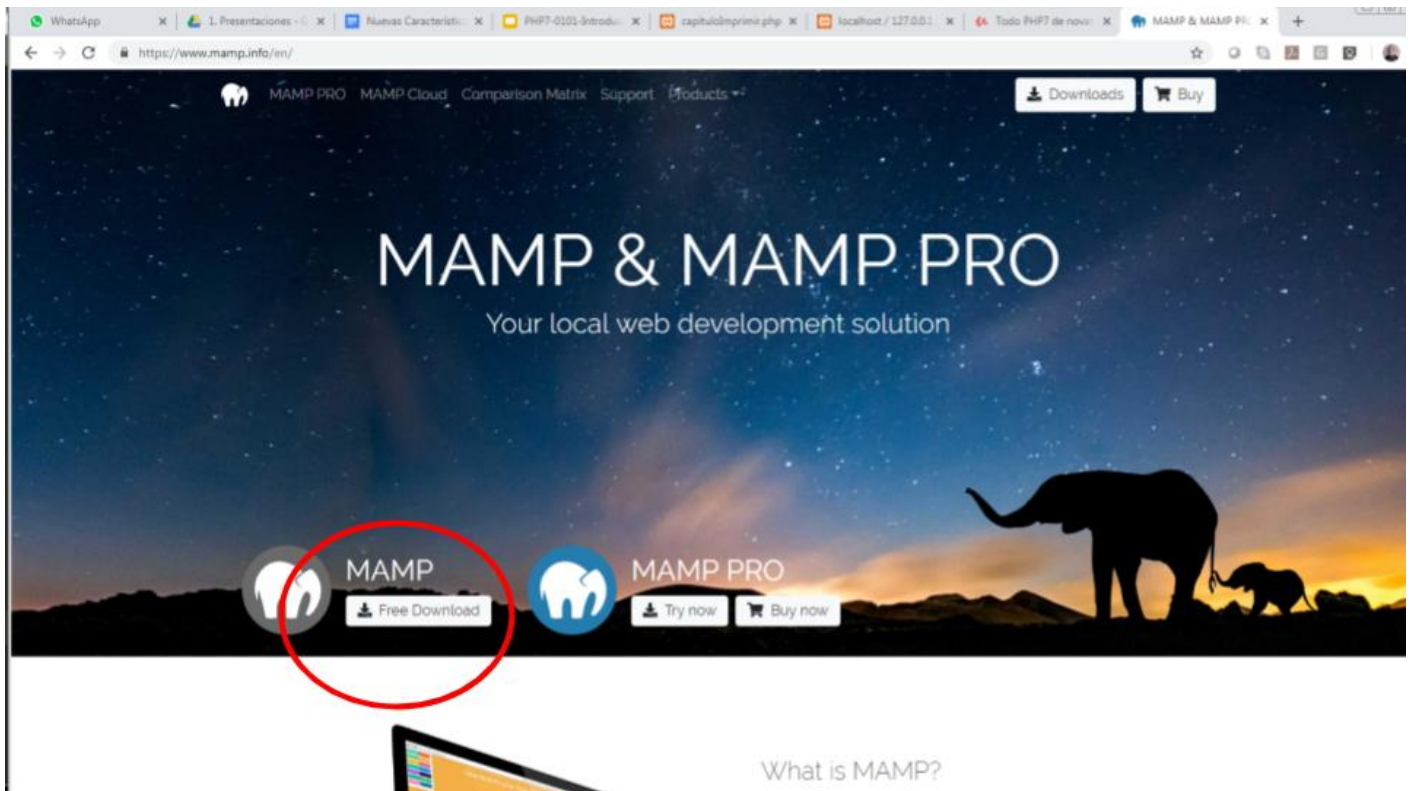


Imagen 1.6.1.

*www.pacoarce.com*

# Todo PHP7

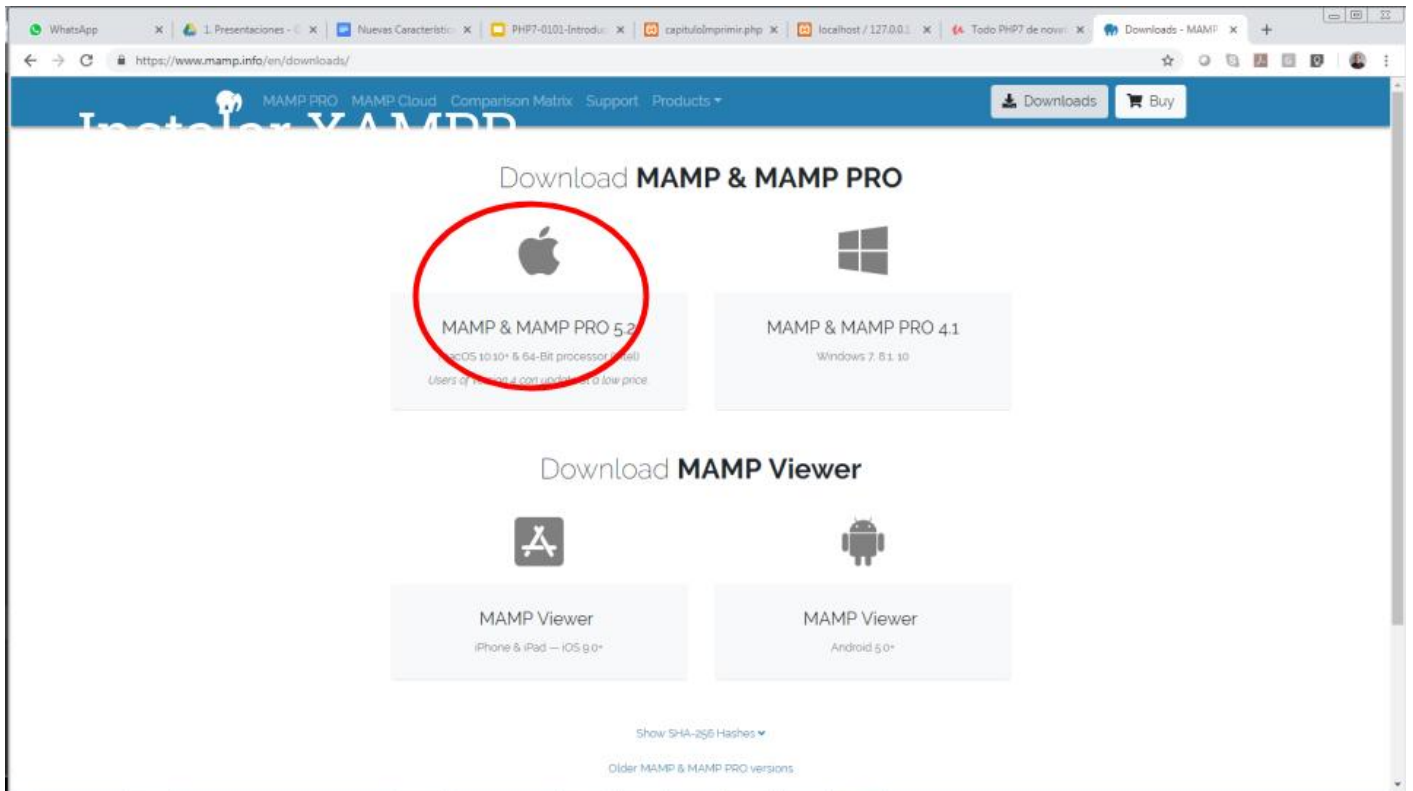
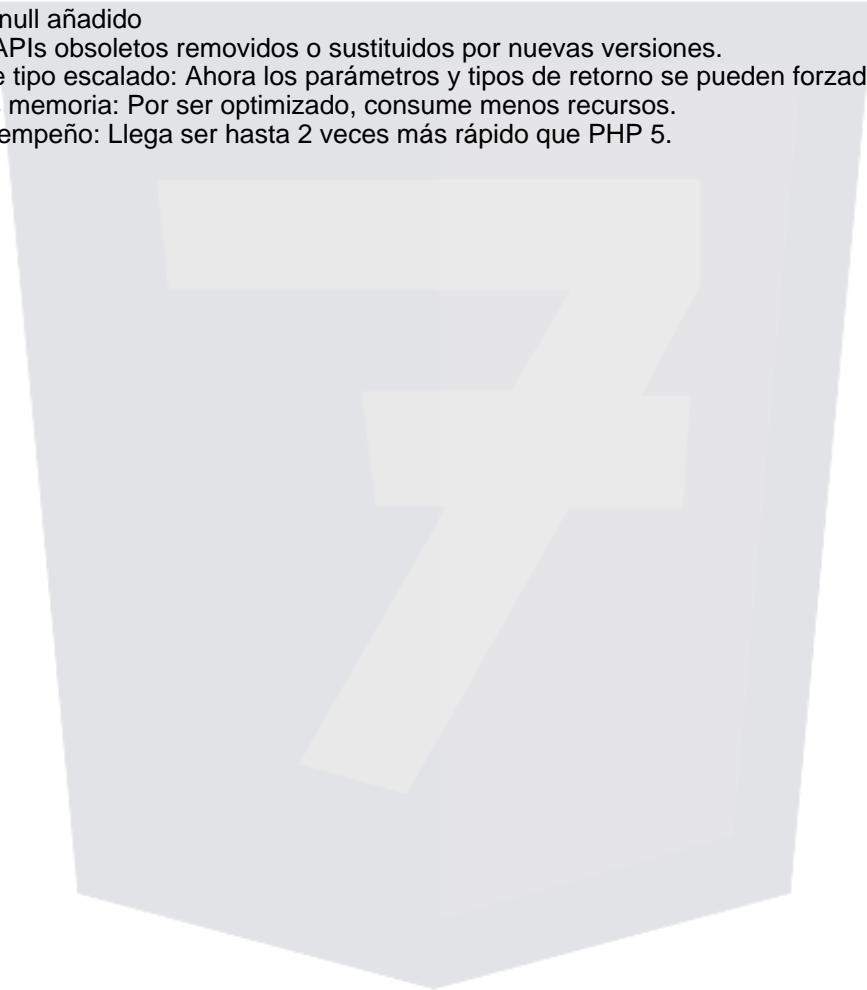


Imagen 1.6.2. Seleccionar la plataforma para MAC.

*www.pacoarce.com*

## 1.7. Nuevas características de PHP7

- Nuevo generador de números aleatorios.
- Algunos “errores fatales” se convirtieron en “excepciones”.
- La jerarquía de excepciones mejorada.
- Soporte a 64 bit: Soporte consistente a computadoras con arquitectura de 64 bits.
- Soporte a “zero cost assert”.
- Creación de clases anónimas.
- Nuevo operador null añadido
- Extensiones y SAPIs obsoletos removidos o sustituidos por nuevas versiones.
- Declaraciones de tipo escalado: Ahora los parámetros y tipos de retorno se pueden forzar.
- Consume menos memoria: Por ser optimizado, consume menos recursos.
- Mejora en el desempeño: Llega a ser hasta 2 veces más rápido que PHP 5.



*www.pacoarce.com*

## 1.8. Nuestra primera página: Hola mundo

Así, un ejemplo de un archivo de PHP, que podría ser guardado con el nombre "hola.php", es:

```
<?php
print "Hola mundo!";
?>
```

Para ejecutar un archivo de php con extensión ".php", debe "subirse" en el servidor junto con las páginas de HTML, y escribir en un navegador el nombre del archivo:

localhost/hola.php

Para saber si un servidor de páginas web tiene instalado el intérprete de PHP se puede crear este sencillo archivo, subirlo y ejecutarlo desde un navegador (debe de usarse un servidor de paginas web, no funciona si se abre un archivo guardado en un disco duro local desde un navegador):

```
<?php
print phpinfo();
?>
```

Este archivo debe producir un resultado similar a esto:

---

```
1 <?php
2 print "Hola mundo desde PHP<br>";
3 echo "PHP no regresa retorno de carro";
4 ?>
```

**Listado 1.8.1.** holaMundo.php

[www.pacoarce.com](http://www.pacoarce.com)

## 1.9. Mezclar HTML y PHP

Es posible mezclar páginas de HTML y código PHP, escribiendo en la parte correspondiente del archivo HTML el código de PHP, siempre delimitado por las etiquetas "<?php" y ">".

Una desventaja de esta mezcla es que no se puede separar el diseño del código, lo cual hace difícil el desarrollo y el mantenimiento de un sitio en el que se involucran diseñadores y programadores (o al menos varias personas a la vez). Más adelante se describirán técnicas para mantener separados el diseño y el código.

---

```
1  <?php print "hola<br>"; ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>Mezclar HTML y PHP</title>
6    <?php print "hola desde head<br>"; ?>
7  </head>
8  <body>
9    <?php print "hola desde body<br>"; ?>
10  <input type="text" value="<?php print 'hola, cara de bola'; ?>">
11 </body>
12 </html>
```

**Listado 1.9.1.** mezcla.php

*www.pacoarce.com*

## 1.10. Comentarios y otras cosas

En todo lenguaje de programación es conveniente tener alguna forma de incluir comentarios, letreros que indican alguna nota acerca del código o su implementación.

En el PHP hay tres maneras de crear comentarios en el código:

- Usando los caracteres `/*` y `*/`: se usa para incluir varias líneas de comentarios
- Usando el carácter `#` (gatito, numeral, almohadilla): Funciona igual que las dos diagonales.
- Usando los caracteres `//`: se usa para incluir una sola línea de comentarios

```
1  <?php
2  //Podemos escribir fuera de las etiquetas html
3  #print "hola<br>";
4  ?>
5  <!DOCTYPE html>
6  <html>
7  <head>
8    <title>Mezclar HTML y PHP</title>
9    <?php
10   /*****
11   comentario de bloque
12   cualquier cosa...
13   print "hola desde head<br>";
14   *****/?>
15 </head>
16 <body>
17 <?php print "hola desde body<br>";
18 print "adios"; ?>
19 <input type="text" value="<?php print 'hola, cara de bola'; ?>">
20 </body>
21 </html>
```

**Listado 1.10.1.** comentarios.php

[www.pacoarce.com](http://www.pacoarce.com)

## Cuestionario:

**1) ¿En qué lugar se ejecuta el código PHP?**

- 1) Servidor
- 2) Cliente

**2) ¿Cuáles de estas son marcas para la inserción del código PHP en las páginas HTML?**

- 1) /\* y \*/
- 2) < php >< /php >
- 3) <# y #>
- 4) < ?php y ? >

**3) ¿Cuál de estas instrucciones PHP es correcta?**

- 1) print < strong >Hola Mundo < /strong>;
- 2) print (< strong>Hola Mundo< /strong>);
- 3) print ("< strong>Hola Mundo < /strong>");

**4) PHP es un lenguaje de etiquetas**

- 1) Verdadero
- 2) Falso

**5) PHP es un lenguaje que es interpretado por el servidor**

- 1) Verdadero
- 2) Falso

**6) ¿Cuál función es equivalente a print?**

- 1) imprime
- 2) echo
- 3) out
- 4) document.write()

**7) Para configurar el funcionamiento de PHP se emplea el archivo:**

- 1) php.conf
- 2) php.ini
- 3) php.setup
- 4) Las anteriores respuestas no son correctas

**8) ¿Qué significa PHP?**

- 1) PHP: Hypertext Preprocessor

- 2) Private Home Page
- 3) Personal Hypertext Processor
- 4) Personal Home Page

**9) En PHP, ¿cómo se inserta un comentario que ocupa una línea?**

- 1)
- 2) Comentario
- 3) // Comentario
- 4) Las anteriores respuestas no son correctas

**10) En PHP, ¿cuál es el carácter que se emplea para finalizar una instrucción?**

- 1) .
- 2) \$
- 3) %
- 4) ;

**11) ¿Quién inició el desarrollo de PHP?**

- 1) Brendan Eich
- 2) Marc Andreessen
- 3) Rasmus Lerdorf
- 4) Las anteriores respuestas no son correctas

**12) La sintaxis de PHP es parecida a**

- 1) VBScript
- 2) C
- 3) Pascal
- 4) Fortran

*www.pacoarce.com*



# Todo PHP7

## De novato a experto

### Capítulo 2: Tipos de datos y variables

**Objetivo:** El alumno aplicará los diferentes tipos de datos que se manejan en PHP y las reglas generales para nombrar las variables.

#### Introducción:

En esta sección veremos:

1. Tipos de datos
2. Tipos de datos: cadenas
3. Concatenación de cadenas
4. Tipos de datos: números enteros
5. Tipos de datos: punto flotante
6. Tipo de dato: Booleano
7. Variables de variables y constantes
8. Secuencias de escape para caracteres Unicode (PHP7)
9. Los tipos de datos compuestos y especiales
10. Forzar el tipo de dato
11. Eliminar una variable con unset()

[www.pacoarce.com](http://www.pacoarce.com)

## 2.1. Variables

Los tipos de datos son fundamentales en cualquier lenguaje de programación. En PHP se considera un lenguaje "debilmente tipado", es decir, va a ser flexible en el manejo de datos, En este video te indicamos cuáles son los tipos de datos primitivos que utilizamos en PHP.

### Variables



**Imagen 2.1.1.** Las variables son como cajas.

*www.pacoarce.com*

## 2.2. Tipos de datos: cadenas

Cadenas de caracteres o "**String**" - Datos alfanuméricos para representar letras y el lenguaje escrito de los humanos. Estos valores van siempre "encerrados" o delimitados entre las comillas dobles o comillas sencillas (apóstrofes).

**PHP** solo admite un conjunto de 256 caracteres, y de ahí que no ofrezca soporte nativo para Unicode. En PHP7 ya soporta Unicode como secuencias de escape.

### Cadenas

0-	1-	2-	3-	4-	5-	6-	7-	8-	9-
0-	1-	2-	3-	4-	5-	6-	7-	8-	9-
10-	11-	12-	13-	14-	15-	16-	17-	18-	19-
20-	21-	22-	23-	24-	25-	26-	27-	28-	29-
30-	31-	32-	33-	34-	35-	36-	37-	38-	39-
40-	41-	42-	43-	44-	45-	46-	47-	48-	49-
50-	51-	52-	53-	54-	55-	56-	57-	58-	59-
60-	61-	62-	63-	64-	65-	66-	67-	68-	69-
70-	71-	72-	73-	74-	75-	76-	77-	78-	79-
80-	81-	82-	83-	84-	85-	86-	87-	88-	89-
90-	91-	92-	93-	94-	95-	96-	97-	98-	99-
100-	101-	102-	103-	104-	105-	106-	107-	108-	109-
110-	111-	112-	113-	114-	115-	116-	117-	118-	119-
120-	121-	122-	123-	124-	125-	126-	127-	128-	129-
130-	131-	132-	133-	134-	135-	136-	137-	138-	139-
140-	141-	142-	143-	144-	145-	146-	147-	148-	149-
150-	151-	152-	153-	154-	155-	156-	157-	158-	159-
160-	161-	162-	163-	164-	165-	166-	167-	168-	169-
170-	171-	172-	173-	174-	175-	176-	177-	178-	179-
180-	181-	182-	183-	184-	185-	186-	187-	188-	189-
190-	191-	192-	193-	194-	195-	196-	197-	198-	199-
200-	201-	202-	203-	204-	205-	206-	207-	208-	209-
210-	211-	212-	213-	214-	215-	216-	217-	218-	219-
220-	221-	222-	223-	224-	225-	226-	227-	228-	229-
230-	231-	232-	233-	234-	235-	236-	237-	238-	239-
240-	241-	242-	243-	244-	245-	246-	247-	248-	249-
250-	251-	252-	253-	254-	255-				

Imagen 2.2.1. Tabla ASCII

```

1  <?php
2  $nombre = "James Bond";
3  $agente = '007';
4  $titulo = "Agente al servicio de su 'magestad'";
5  //print $nombre;
6  //print $agente;
7  //print $titulo;
8
9  $str = <<<'EOD'
10 Mi nombre es: $nombre
11 mi número de agente es: $agente
12 mi lema es: $titulo
13 EOD;
14
15 print $str;

```

16  
17 ?>

**Listado 2.2.1.** PHP7-0202-Cadenas.php



*www.pacoarce.com*

## 2.3. Concatenación de cadenas

**PHP** es famoso por sus formas de concatenar cadenas. Por su propia naturaleza de ser un lenguaje intermedio para HTML, tiene poderosas herramientas para el manejo de cadenas, entre ellas su concatenación.

---

```
1  <?php
2  $nombre = "James Bond";
3  $agente = '007';
4  $titulo = "Agente al servicio de su \"magestad\"";
5  $salida = $nombre." Num. Agente: ";
6  $salida .= $agente."<br>";
7  $salida .= $titulo;
8  print $salida;
9
10 ?>
```

**Listado 2.3.1.** PHP7-0203-Concatenar.php

[www.pacoarce.com](http://www.pacoarce.com)

## 2.4. Tipos de datos: números enteros

Las variables numéricas son muy flexibles en PHP. Podemos almacenar número decimales, octales y hexadecimal, enteros, negativos y enteros sin ninguna restricción.

```
1  <?php
2  $num = 12345678;
3  $num = -12345678;
4  $num = 0123456; //octal 0-7
5  $num = 0xFFAADD; //hexadecimal 0-9 a-f
6  $num = 0b1010101; //Binario 0-1
7
8  $num = 2147483647;
9  print var_dump($num);
10
11 $num = 2147483648;
12 print var_dump($num);
13
14 $num = 2147483647*2;
15 print var_dump($num)
16 ?>
```

**Listado 2.4.1.** PHP7-0204-TipoEntero.php

*www.pacoarce.com*

## 2.5. Tipos de datos: punto flotante

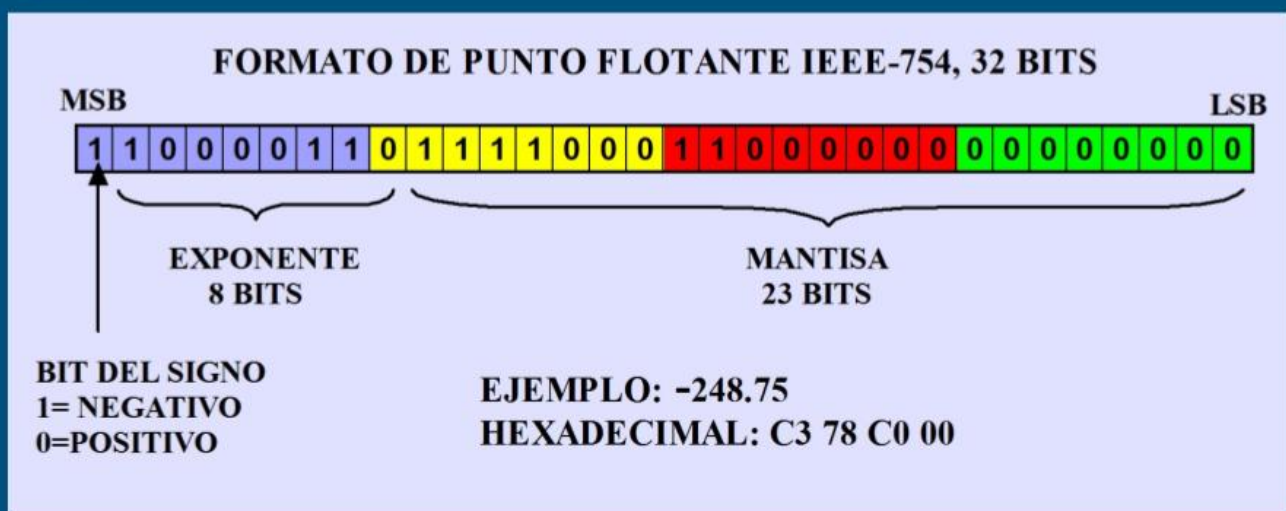
Los números de punto flotante (también conocidos como "de coma flotante" en español, y "floats" en inglés) pueden ser especificados usando cualquiera de las siguientes sintaxis:

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

El tamaño de un 'float' depende de la plataforma, aunque un valor común consiste en un máximo de  $\sim 1.8e308$  con una precisión de aproximadamente 14 dígitos decimales (el formato de 64 bit del IEEE).

Comprobar la igualdad de valores de punto flotante es problemático debido a la forma en que se representan internamente. Sin embargo, hay maneras de hacer comparaciones de los valores de punto flotante que evitan estas limitaciones. Para comprobar la igualdad de valores de punto flotante, se utiliza un límite superior en el error relativo debido al redondeo. Este valor se conoce como el  $\epsilon$  de la máquina o unidad de redondeo, y es la menor diferencia aceptable en los cálculos.

### Números de punto flotante



**Imagen 2.5.1.** Esquema de un numero flotante.

```
1  <?php
2  $a = 1234.5678;
3  $a = 12345.2e4;
4  $a = 7E-10;
5
6  $e = 1.23456789;
7  $d = 1.2345678901;
8  $epsilon = 0.000001;
9
10 if(abs($e-$d)<$epsilon) print "Son iguales";
11 else print "Son diferentes";
12 ?>
```

**Listado 2.5.1.** PHP7-0205-PuntoFlotante.php

*www.pacoarce.com*



## 2.6. Tipo de dato: Booleano

- Este es el tipo más simple. Un boolean expresa un valor que indica verdad. Puede ser TRUE (verdadero) o FALSE (falso).
- Para especificar un literal de tipo boolean se emplean las constantes TRUE o FALSE. Ambas no son susceptibles a mayúsculas y minúsculas.
- Usualmente, el resultado de un operador que devuelve un valor de tipo boolean es pasado a una estructura de control.
- Advertencia: -1 se considera TRUE, como cualquier otro número distinto de cero (ya sea negativo o positivo).

---

```
1 <?php
2 $administrador = true;
3 if ($administrador) {
4     print "Bienvenido";
5 } else {
6     print "No tiene acceso a la sistema";
7 }
8 ?>
```

**Listado 2.6.1.** PHP7-0206-Booleanos.php

*www.pacoarce.com*

## 2.7. Variables de variables y constantes

Podemos utilizar el nombre de una variable dentro del valor de una segunda variable y utilizar esta última por medio de las llaves {}. Por ejemplo

```
$df = 22000000;  
$guadalajara = 4000000;  
$monterrey = 1400000;  
$ciudad = "df";  
print "<p>La población de la ciudad $ciudad es de ${$ciudad} de habitantes</p>";  
$ciudad = "guadalajara";  
print "<p>La población de la ciudad $ciudad es de ${$ciudad} de habitantes</p>";  
$ciudad = "monterrey";  
print "<p>La población de la ciudad $ciudad es de ${$ciudad} de habitantes</p>";
```

NOTA: en PHP7 se evalúan en forma diferente las variables de variables en arreglos o arrays.

**bool define ( string \$name , mixed \$value [, bool \$case\_insensitive = false ] )**

La última bandera permite diferenciar entre mayúsculas y minúsculas, donde si es falso, la constante SALARIO es diferente a Salario.

En PHP7 se aceptan también arreglos.

Por medio de la función define() podemos definir una constante por su nombre en tiempo de ejecución:

---

```
1  <!DOCTYPE html>  
2  <html>  
3  <head>  
4    <title>Variables de variables y contantes</title>  
5    <meta charset="utf-8">  
6  </head>  
7  <body>  
8    <?php  
9      $cdmx = 22000000;  
10     $guadalajara = 4000000;  
11     $monterrey = 1400000;  
12     //variable de variables  
13     $ciudad = "cdmx";  
14     print "<p>La población de la ciudad $ciudad es de ${$ciudad}</p>";  
15     $ciudad = "guadalajara";  
16     print "<p>La población de la ciudad $ciudad es de ${$ciudad}</p>";  
17     $ciudad = "monterrey";  
18     print "<p>La población de la ciudad $ciudad es de ${$ciudad}</p>";  
19     //  
20     //Creamos una constante  
21     define("TASA_CAMBIO",18.35);  
22     $deuda = 12345;
```

```
23     print "Tu deuda en dólares es de " . ($deuda*TASA_CAMBIO);  
24     ?>  
25 </body>  
26 </html>
```

**Listado 2.7.1.** PHP7-0207-VariablesVariablesConstantes.php



*www.pacoarce.com*

## 2.8. PHP7: Secuencias de escape para caracteres Unicode

Esta sintaxis toma un punto de código de **Unicode** en forma hexadecimal e imprime ese punto de código en UTF-8 a una cadena con entrecomillado doble o a un **heredoc**. Se acepta cualquier punto de código válido, siendo los ceros iniciales opcionales.

---

```
1 <?php
2 echo "\u{2764}";
3 echo "\u{265E}";
4 echo "\u{9999}";
5 ?>
```

**Listado 2.8.1.** PHP7-0208-Unicode.php

*www.pacoarce.com*

## 2.9. Los tipos de datos compuestos y especiales

- Existen un tipo de "cajitas" que pueden contener mas de un valor independiente a la vez, y se llaman "**arreglos**" (arrays). Esto puede visualizarse como una caja de herramientas con compartimentos independientes, pero todos unidos formando una sola "caja". No se pueden separar los compartimentos de la caja principal, y ahí donde va la caja principal, van las "cajitas" que esta contiene. Se verán los arreglos en la sección 6
- **Iterable** es un seudotipo introducido en PHP 7.1. Acepta cualquier arreglo u objeto que implemente la interfaz **Traversable**. Estos dos tipos de datos (arreglos y objetos) se recorren con el ciclo foreach y se pueden emplear con la palabra reservada **yield from** dentro de un generador. Los generadores los veremos en el capítulo 7.
- Los objetos se crean con la palabra reservada **class** y podemos almacenar métodos (funciones) y propiedades (datos). Se verán los objetos en la programación orientada a objetos.
- Muchas funciones de **PHP** llaman a otras funciones para procesar sus datos. A estas funciones las llamaremos funciones de llamada de retorno o **callbacks**. Las llamadas de retorno (**Callbacks / Callables**) se verán en la sección 7.
- Un valor tipo **resource** es una variable especial, que contiene una referencia a un recurso externo. Los recursos son creados y usados por funciones especiales. Los recursos externos pueden ser leer una base de datos, leer un archivo de imágenes binarios, entre muchos otros.
- El valor especial **NULL** representa una variable sin valor. **NULL** es el único valor posible del tipo null. Una variable es considerada null si: se le ha asignado la constante **NULL**. no se le ha asignado un valor todavía. se ha destruido con **unset()**.

[www.pacoarce.com](http://www.pacoarce.com)

## 2.10. Forzar el tipo de dato

Es posible "**forzar**" a que el resultado de una operación cambie de tipo, como por ejemplo una división entre números con decimales regrese un número entero. La manera más fácil de forzar este cambio es así:

La variable \$entero sólo recibirá un valor entero, sin los decimales.

```
$entero = (integer) $subtotal / $iva;
```

En el último caso, cualquier valor que sea diferente de cero regresa "true", y cero regresa "false".

También tenemos las funciones: intval() floatval() strval() boolval() settype() gettype()

Operador	Cambia a
(integer)	numero entero
(double)	número con decimales
(string)	cadena de caracteres
(boolean)	verdadero o falso

Imagen 2.10.1. Cambiar de tipo

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Forzar tipo de dato</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <?php
9     $div = 10/3;
10    print $div."<br>" ;
```

```
11 $entero = (int) $div;
12 print $entero."<br>";
13 var_dump($div);
14 print "<br>";
15 var_dump($entero);
16 print "<br>";
17 print intval(10/3)."<br>";
18 settype($div, "integer");
19 var_dump($div);
20 print "<br>";
21 print gettype($div);
22 ?>
23 </body>
24 </html>
```

**Listado 2.10.1.** forzarTipo.php

*www.pacoarce.com*

## 2.11. Eliminar una variable con unset()

Por medio de **unset()** podemos eliminar una variable, un objeto, una instancia o un elemento de un arreglo.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Borrar variable</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $entero = 10;
10   var_dump($entero);
11   unset($entero);
12   var_dump($entero);
13   ?>
14 </body>
15 </html>
```

**Listado 2.11.1.** borrarVariable.php

*www.pacoarce.com*



## Cuestionario:

**1) PHP es un lenguaje donde no se distinguen mayúsculas de minúsculas por lo tanto \$color y \$Color son equivalentes**

- 1) Verdadero
- 2) Falso

**2) Indicar cuales de los siguientes nombres de variables en PHP NO es correcto:**

- 1) \$\_color
- 2) \$nombre\_archivo
- 3) \$1enlace
- 4) \$PESO

- 1) \$variable = True;
- 2) \$variable = true;
- 3) \$variable = TRUE;
- 4) Todas las respuestas son correctas

**4) En PHP, para comprobar si una constante existe se emplea la función**

- 1) isconstant()
- 2) defined()
- 3) isset()
- 4) Las anteriores respuestas no son correctas

**5) En PHP, todas las variables comienzan con el símbolo**

- 1) !
- 2) &
- 3) \$
- 4) #

**6) En PHP, ¿cómo NO se escriben las cadenas de caracteres?**

- 1) Entre comillas simples '
- 2) Entre comillas dobles "
- 3) Entre comillas simples ' o comillas dobles "
- 4) Entre llaves {

**7) Un valor que no ha sido definido se expresa en PHP con la palabra reservada:**

- 1) empty

- 2) null
- 3) undef
- 4) Las anteriores respuestas no son correctas

**8) ¿Cuál de los siguientes tipos no son básicos o escalares?**

- 1) boolean
- 2) integer
- 3) string
- 4) array

**9) ¿Cuál de estos tipos NO son compuestos?**

- 1) array
- 2) boolean
- 3) object
- 4) callable

**10) ¿A partir de qué versión podemos tener contenido UNICODE?**

- 1) 4.0.
- 2) 5.0.
- 3) 5.6.
- 4) 7.0.

**11) La sintaxis heredoc nos permiten definir cadenas o strings:**

- 1) Verdadero
- 2) Falso

**12) La sintaxis nowdoc nos permiten escribir números muy grandes de precisión doble.**

- 1) Verdadero
- 2) Falso

**13) Para concatenar cadenas en PHP utilizamos el comando:**

- 1) +
- 2) =
- 3) .
- 4) ;

**14) La secuencia de escape para determinar una nueva línea es:**

- 1) \n
- 2) \t

3) \\*

4) \\\

**15) Para determinar un número hexadecimal hay que anteponer al número:**

1) hex:

2) #

3) 0x

4) 0

**16) En PHP, un número con decimales se llama:**

1) dec

2) decimal

3) float

4) string

**17) En PHP, un valor -1 se considera como falso**

1) Verdadero

2) Falso

**18) Para eliminar una variable u objetos se utiliza el método:**

1) unset()

2) clear()

3) delete()

4) erase()

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 3: Operadores en PHP

**Objetivo:** El alumno aplicará los diferentes operadores en PHP como operadores aritméticos, de asignación, bit a bit, de comparación, lógicos, entre otros, así como su precedencia.

#### Introducción:

En esta sección veremos los siguientes operadores:

- Introducción a los operadores y sus conceptos en PHP
- Operadores aritméticos
- Los operadores de asignación y los operadores abreviados o atajos
- Operadores de bit a bit
- Operadores de comparación
- Operadores Lógicos
- Otros operadores en PHP
- La precedencia de los operadores
- PHP7: El operador de fusión de null
- PHP7: Operador nave espacial .

## 3.1. Introducción a los operadores y sus conceptos en PHP

- Un operador es algo que toma uno más valores (literales, variables u otros operadores) y produce otro valor (de modo que la construcción en si misma se convierte en una expresión).
- Los operadores se pueden agrupar de acuerdo con el número de valores que toman. Los operadores unarios toman sólo un valor, por ejemplo ! (el operador lógico de negación) o ++ (el operador de incremento).
- Los operadores binarios toman dos valores, como los familiares operadores aritméticos + (suma) y - (resta), y la mayoría de los operadores de PHP entran en esta categoría.
- Finalmente, hay sólo un operador ternario, ? :, el cual toma tres valores; usualmente a este se le refiere simplemente como "el operador ternario" (aunque podría llamarse operador condicional).
  - Operadores de tipo
  - Operadores para arreglos
  - Operadores para cadenas
  - Operadores lógicos
  - Operadores de incremento/decremento
  - Operadores de ejecución
  - Operadores de control de errores
  - Operadores de comparación
  - Operadores bit a bit
  - Operadores de asignación
  - Operadores aritméticos
- La precedencia de un operador indica la prioridad en que se ejecutan los operadores. Por ejemplo, en la expresión  $1 + 5 * 3$ , la respuesta es 16 y no 18 porque el operador de multiplicación ("\*") tiene una precedencia mayor que el operador de adición ("+").
- Los paréntesis pueden ser usados para forzar la precedencia, si es necesario. Por ejemplo:  $(1 + 5) * 3$  se evalúa como 18.
- Una expresión es la combinación de literales, operadores, variables y funciones, bajo la sintaxis correcta del lenguaje, que regresan un valor válido para PHP.
- Valores que no cambian, como número (66) o cadenas ("Hola mundo") o valores booleanos (true).

---

```
1 <?php
2 $a = 6;
3 print -$a ** 3;
4 ?>
```

**Listado 3.1.1.** PHP7-0302-OperadoresAritmeticos.php

[www.pacoarce.com](http://www.pacoarce.com)

## 3.2. Operadores aritméticos

Introducido en PHP 5.6.

- $\$a ** \$b$  Exponenciación Resultado de elevar  $\$a$  a la potencia  $\$b$ ésima.
- $\$a \% \$b$  Módulo Resto de  $\$a$  dividido por  $\$b$ .
- $\$a / \$b$  División Cociente de  $\$a$  y  $\$b$ .
- $\$a * \$b$  Multiplicación Producto de  $\$a$  y  $\$b$ .
- $\$a - \$b$  Sustracción Diferencia de  $\$a$  y  $\$b$ .
- $\$a + \$b$  Adición Suma de  $\$a$  y  $\$b$ .
- $-\$a$  Negación Opuesto de  $\$a$ .
- $+\$a$  Identidad Conversión de  $\$a$  a int o float según el caso.

---

```
1 <?php
2 $a = 6;
3 print -$a ** 3;
4 ?>
```

**Listado 3.2.1.** PHP7-0302-OperadoresAritmeticos.php

*www.pacoarce.com*

## 3.3. Los operadores de asignación y los operadores abreviados o atajos

- El operador básico de asignación es "=". Se podría inclinar a pensar primero que es como un "igual a". No lo es. Realmente significa que el operando de la izquierda se establece con el valor de la expresión de la derecha (es decir, "se define como").
- El valor de una expresión de asignación es el valor asignado. Es decir, el valor de "\$a = 3" es de 3.
- Además del operador básico de asignación, existen «operadores combinados» para todos los de aritmética binaria, unión de arrays y operadores de strings que permiten usar un valor en una expresión y entonces establecer su valor como el resultado de esa expresión. Se puede realizar "Asignación por valor" vs. "Asignación por referencia".
- Los "atajos" también son considerados operadores de asignación:

\$a += \$b \$a = \$a + \$b Suma

\$a -= \$b \$a = \$a - \$b Resta

\$a \*= \$b \$a = \$a \* \$b Multiplicación

\$a /= \$b \$a = \$a / \$b División

\$a %= \$b \$a = \$a % \$b Módulo

---

```
1  <?php
2  $a = 30;
3  $a = $a + 5 / 2 * 10;
4  $a += 10;
5  $a /= 2;
6  $a *= 1.5;
7  print $a;
8  print $a++;
9  print $a--;
10 ?>
```

**Listado 3.3.1.** PHP7-0303-OperadoresAsginacion.php

*www.pacoarce.com*

## 3.4. Operadores de bit a bit

Los operadores bit a bit permiten la evaluación y la manipulación de bits específicos dentro de un valor entero.

$\$a \& \$b$  And (y) Los bits que están activos en ambos  $\$a$  y  $\$b$  son activados.

$\$a | \$b$  Or (o inclusivo) Los bits que están activos ya sea en  $\$a$  o en  $\$b$  son activados.

$\$a \wedge \$b$  Xor (o exclusivo) Los bits que están activos en  $\$a$  o en  $\$b$ , pero no en ambos, son activados.

$\sim \$a$  Not (no) Los bits que están activos en  $\$a$  son desactivados, y viceversa.

$\$a \ll \$b$  Shift left(desplazamiento a izquierda) Desplaza los bits de  $\$a$ ,  $\$b$  pasos a la izquierda (cada paso quiere decir "multiplicar por dos").  $\$a \gg \$b$  Shift right (desplazamiento a derecha) Desplaza los bits de  $\$a$ ,  $\$b$  pasos a la derecha (cada paso quiere decir "dividir por dos"),

```
1  <?php
2  $values = array(0, 1, 2, 4, 8);
3  $test = 1 + 4;
4  $formato = '(%1$2d = %1$04b) = (%2$2d = %2$04b)'
5  . ' %3$s (%4$2d = %4$04b)' . "<br>";
6  echo "AND bit a bit <br>";
7  foreach ($values as $value) {
8  $result = $value & $test;
9  printf($formato, $result, $value, '&', $test);
10 }
11 ?>
```

**Listado 3.4.1.** PHP7-0304-OperadoresBitaBit.php

[www.pacoarce.com](http://www.pacoarce.com)



## 3.5. Operadores de comparación

- Los operadores de comparación, como su nombre lo indica, permiten comparar dos valores:
- `$a == $b` Igual, TRUE si \$a es igual a \$b después de la manipulación de tipos.
- `$a === $b` Idéntico, TRUE si \$a es igual a \$b, y son del mismo tipo.
- `$a != $b` Diferente, TRUE si \$a no es igual a \$b después de la manipulación de tipos.
- `$a <> $b` Diferente TRUE, si \$a no es igual a \$b después de la manipulación de tipos.
- `$a !== $b` No idéntico, TRUE si \$a no es igual a \$b, o si no son del mismo tipo.
- `$a < $b` Menor que, TRUE si \$a es estrictamente menor que \$b.
- `$a > $b` Mayor que, TRUE si \$a es estrictamente mayor que \$b.
- `$a <= $b` Menor o igual que, TRUE, si \$a es menor o igual que \$b.
- `$a >= $b` Mayor o igual que, TRUE, si \$a es mayor o igual que \$b.
- `$a ?? $b ?? $c` Fusión de null, El primer operando de izquierda a derecha que exista y no sea NULL. NULL si no hay valores definidos y no son NULL. Disponible a partir de PHP 7.
- `$a <=> $b` Nave espacial, Un integer menor que, igual a, o mayor que cero cuando \$a es respectivamente menor que, igual a, o mayor que \$b. Disponible a partir de PHP 7.

---

```
1 <?php
2 $a = 18;
3 $b = 11;
4 if($a==$b){
5     print " a es mayor que b";
6 } else {
7     print "a NO es mayor que b";
8 }
9 ?>
```

**Listado 3.5.1.** PHP7-0305-OperadoresComparacion.php

*www.pacoarce.com*

## 3.6. Operadores Lógicos

`$a and $b` And (y), TRUE si tanto `$a` como `$b` son TRUE.

`$a or $b` Or (o inclusivo), TRUE si cualquiera de `$a` o `$b` es TRUE.

`$a xor $b` Xor (o exclusivo), TRUE si `$a` o `$b` es TRUE, pero no ambos.

`! $a` Not (no), TRUE si `$a` no es TRUE.

`$a && $b` And (y), TRUE si tanto `$a` como `$b` son TRUE.

`$a || $b` Or (o inclusivo), TRUE si cualquiera de `$a` o `$b` es TRUE.

### Los operadores lógicos

AND			OR			XOR		
A	B	A	A	B	A	A	B	A
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0

Imagen 3.6.1. Los operadores logicos

```

1 <?php
2 $dinero = false;
3 $tiempo = false;
4 if ($dinero and $tiempo) {
5     print "Voy al cine";
6 } else {
7     print "Me quedo en casa";
8 }
9 ?>

```

Listado 3.6.1. PHP7-0306-OperadoresLogicos.php

## 3.7. Otros operadores en PHP

PHP soporta un operador de control de errores: el signo de arroba (@). Cuando se antepone a una expresión en PHP, cualquier mensaje de error que pueden ser generado por esa expresión será ignorado.

```
$archivo = @file ('noexiste.php') or
```

```
die ("La apertura de archivo ha fallado: el error fue '$php_errormsg');"
```

PHP soporta operadores estilo C de pre- y post-incremento y decremento.

Nota: Los operadores de incremento/decremento solamente afectan a números y strings. Los arrays, objects y resources no se ven afectados. Decrementar valores NULL tampoco tiene efecto, pero incrementarlos entonces resulta en 1.

`++$a` Pre-incremento, Incrementa \$a en uno, y luego retorna \$a.

`$a++` Post-incremento, Retorna \$a, y luego incrementa \$a en uno.

`--$a` Pre-decremento, Decrementa \$a en uno, luego retorna \$a.

`$a--` Post-decremento, Retorna \$a, luego decrementa \$a en uno.

Existen dos operadores para datos tipo string. El primero es el operador de concatenación ('.'), el cual retorna el resultado de concatenar sus argumentos derecho e izquierdo. El segundo es el operador de asignación sobre concatenación ('.='), el cual añade el argumento del lado derecho al argumento en el lado izquierdo. Por favor consulte Operadores de asignación para más información.

`$a + $b` Unión, Unión de \$a y \$b.

`$a == $b` Igualdad, TRUE si \$a y \$b tienen las mismas parejas clave/valor.

`$a === $b` Identidad, TRUE si \$a y \$b tienen las mismas parejas clave/valor en el mismo orden y de los mismos tipos.

`$a != $b` Desigualdad, TRUE si \$a no es igual a \$b.

`$a <> $b` Desigualdad, TRUE si \$a no es igual a \$b.

`$a !== $b` No-identidad, TRUE si \$a no es idéntica a \$b.

`instanceof` se utiliza para determinar si una variable de PHP es un objeto instanciado de una cierta clase.

```
1 <?php
2 $a = 5;
3 print --$a;
4 print $a--;
5 print $a;
6 ?>
```

**Listado 3.7.1.** PHP7-0307-OperadoresOtros.php

## 3.8. La precedencia de los operadores

- La precedencia de un operador indica qué tan "estrechamente" se unen dos expresiones juntas.
- Los paréntesis pueden ser usados para forzar la precedencia, si es necesario.
- Cuando los operadores tienen *igual precedencia* su asociatividad decide cómo se agrupan.
- Por ejemplo "-" tiene asociatividad a izquierda, así  $1 - 2 - 3$  se agrupa como  $(1 - 2) - 3$  y se evalúa a -4. "=", por otra parte, tiene asociatividad a derecha, así  $\$a = \$b = \$c$  se agrupa como  $\$a = (\$b = \$c)$ .
- Los operadores de *igual precedencia* que no son asociativos no pueden usarse unos junto a otros, por ejemplo,  $1 < 2 > 1$  es ilegal en PHP. La expresión  $1 \leq 1 == 1$ , por otro lado, es legal, ya que el operador `==` tiene *menos precedencia* que el operador `<=`.
- El uso de paréntesis, incluso cuando no es estrictamente necesario, a menudo puede aumentar la legibilidad del código haciendo grupos explícitamente en lugar de confiar en la precedencia y asociatividad implícitas del operador.
  - no asociativo clone new
  - izquierda [
  - derecha \*\*
  - derecha ++ -- ~
  - no asociativo instanceof
  - derecha !
  - izquierda \* / %
  - izquierda + - .
  - izquierda << >>
  - no asociativo < <= > >=
  - no asociativo == != === !== <> <=>
  - izquierda &
  - izquierda ^
  - izquierda |
  - izquierda &&
  - izquierda ||
  - derecha ??
  - izquierda ? :
  - derecha = += -= \*= \*\*= /= .= %= &= |= ^= <=> >=>
  - izquierda and
  - izquierda xor
  - izquierda or

```
1 <?php
2 $a = 2+(3*4)**2;
3 print $a;
4 ?>
```

Listado 3.8.1. PHP7-0308-Precedencia.php

[www.pacoarce.com](http://www.pacoarce.com)

## 3.9. PHP7: El operador de fusión de null

Versiones: PHP7

- El operador de fusión de null (??) es equivalente al operador ternario junto con isset().
- Devuelve su primer operando si existe y no es NULL; de lo contrario devuelve su segundo operando.
- `$nombre_usuario = $_GET['usuario'] ?? 'nadie';`
- `$nombre_usuario = isset($_GET['usuario']) ? $_GET['usuario'] : 'nadie';`
- `$nombre_usuario = $_GET['usuario'] ?? $_POST['usuario'] ?? 'nadie';`

---

```
1 <?php
2 $usuario = $_GET["usuario"]??$_POST["usuario"]??"Anonimo";
3 print $usuario;
4 ?>
5
```

**Listado 3.9.1.** operadorFusion.php

*www.pacoarce.com*

## 3.10. PHP7: Operador nave espacial

Versiones: PHP7

- El operador nave espacial se emplea para comparar dos expresiones.
- Devuelve -1, 0 o 1 cuando \$a es respectivamente menor, igual, o mayor que \$b.
- Las comparaciones se realizan según las reglas de comparación de tipos usuales de PHP.

// Números enteros

```
echo 1 <=> 1; // 0
```

```
echo 1 <=> 2; // -1
```

```
echo 2 <=> 1; // 1
```

---

```
1  <?php
2  print 1 <=> 1;
3  print "<br>";
4  print 1 <=> 2;
5  print "<br>";
6  print 2 <=> 1;
7  print "<br>";
8  print "<br>";
9  print 1.5 <=> 1.5;
10 print "<br>";
11 print 1.5 <=> 2.1;
12 print "<br>";
13 print 2.1 <=> 1.5;
14 print "<br>";
15 print "<br>";
16 print "1" <=> "1";
17 print "<br>";
18 print "1" <=> "2";
19 print "<br>";
20 print "2" <=> "1";
21 print "<br>";
22 print "<br>";
23 ?>
```

**Listado 3.10.1.** operadorNaveEspacial.php

*www.pacoarce.com*

## Cuestionario:

**1) Un operador es algo que toma uno más valores y produce otro valor.**

- 1) Verdadero
- 2) Falso

**2) Los operadores que toman sólo un valor se llaman:**

- 1) Unarios
- 2) Binarios
- 3) Ternarios
- 4) Aburridos

**3) Los operadores suma (+) o multiplicación son de tipo:**

- 1) Unarios
- 2) Binarios
- 3) Ternarios
- 4) Aburridos

**4) La mayoría de los operadores de PHP son ternarios:**

- 1) Verdadero
- 2) Falso

**5) La prioridad de ejecución de los operadores se llama:**

- 1) Herencia
- 2) Precedencia
- 3) Polimorfismo
- 4) Abstracción

**6) Para modificar la precedencia en los operadores utilizamos:**

- 1) Corchetes
- 2) Llaves
- 3) Dos puntos
- 4) Paréntesis

**7) Una expresión es la combinación de literales, operadores, variables y funciones, bajo la sintaxis correcta del lenguaje, que regresan un valor válido para PHP.**

- 1) Verdadero
- 2) Falso

**8) Para asignar un valor a una variable utilizamos el operador:**

- 1) Paréntesis
- 2) Llaves
- 3) Igualdad
- 4) Corchetes

**9) Para comparar dos valores utilizamos:**

- 1) Punto (.)
- 2) Igualdad (=)
- 3) Doble igualdad (==)
- 4) Símbolo de administración (!)

**10) Para compara la diferencia de dos elementos utilizamos:**

- 1) Admiración e igualdad (!=)
- 2) Igualdad (=)
- 3) Doble igualdad (==)
- 4) Símbolo de administración (!)

**11) El operador "nave espacial" se representa como:**

- 1) !=
- 2) =
- 3) ==
- 4) <=>

**12) El operador "Fusión de null" está disponible a partir de la versión:**

- 1) 4.0
- 2) 5.6
- 3) 6.0
- 4) 7.0

**13) El operador OR lo representamos como:**

- 1) !=
- 2) ||
- 3) ==
- 4) <=>

**14) El operador NEGACION lo representamos como:**

- 1) !



- 2) ||
- 3) ==
- 4) <=>

**15) Para concatenar dos cadenas en PHP utilizamos el operador:**

- 1) \*
- 2) +
- 3) ==
- 4) .

**16) El operador que se utiliza para determinar si una variable de PHP es un objeto instanciado de una cierta clase.**

- 1) class
- 2) instance
- 3) instanceof
- 4) is\_instance\_of()

**17) Cuando los operadores tienen igual precedencia el término para evaluarlos lo llamamos:**

- 1) Igualdad
- 2) Asociatividad
- 3) Precedencia
- 4) Herencia

**18) En PHP, para calcular el módulo (el resto) de una división, se emplea el operador**

- 1) !
- 2) #
- 3) %
- 4) &

**19) En PHP, ¿cuál es la forma correcta de añadir 1 a la variable \$contador?**

- 1) \$contador =+1;
- 2) \$contador++;
- 3) \$++contador;
- 4) Las anteriores respuestas no son correctas

**20) En PHP, ¿cuáles son operadores de comparación?**

- 1) ==, >=, >=, !=

- 2) &&, ||, and, or, xor
- 3) \*, /, -, +
- 4) Las anteriores respuestas no son correctas

**21) En PHP cuando dentro de una expresión de cadena queremos que se expanda el contenido de una variable, encerramos la cadena con comillas simples. Por ejemplo: 'El contenido es: \$variable';**

- 1) Verdadero
- 2) Falso



*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 4: Las sentencias condicionales

**Objetivo:** El alumno conocerá y aplicará las diferentes sentencias condicionales, su anidación, la estructura else if, el uso de switch y del operador condicional.

#### Introducción:

Los temas que veremos en este capítulo son:

- Sentencias condicionales
- Operadores condicionales anidados
- Crear rangos con operadores condicionales
- Sintaxis alterna a las sentencias condicionales
- Sintaxis alterna a las sentencias condicionales
- La estructura condicional switch

*www.pacoarce.com*

## 4.1. Sentencias condicionales

- En los lenguajes de programación existen dos grandes grupos de comandos para el control del flujo del programa, para decidir que hacer, o si se trata de un ciclo, cuantas veces hacerlo o hasta que momento dejar de hacerlo.
- El primer grupo de comandos son los "condicionales", los cuales evalúan una expresión y dependiendo del resultado de la evaluación (verdadero o falso) ejecutan o dejan de ejecutar ciertas instrucciones.

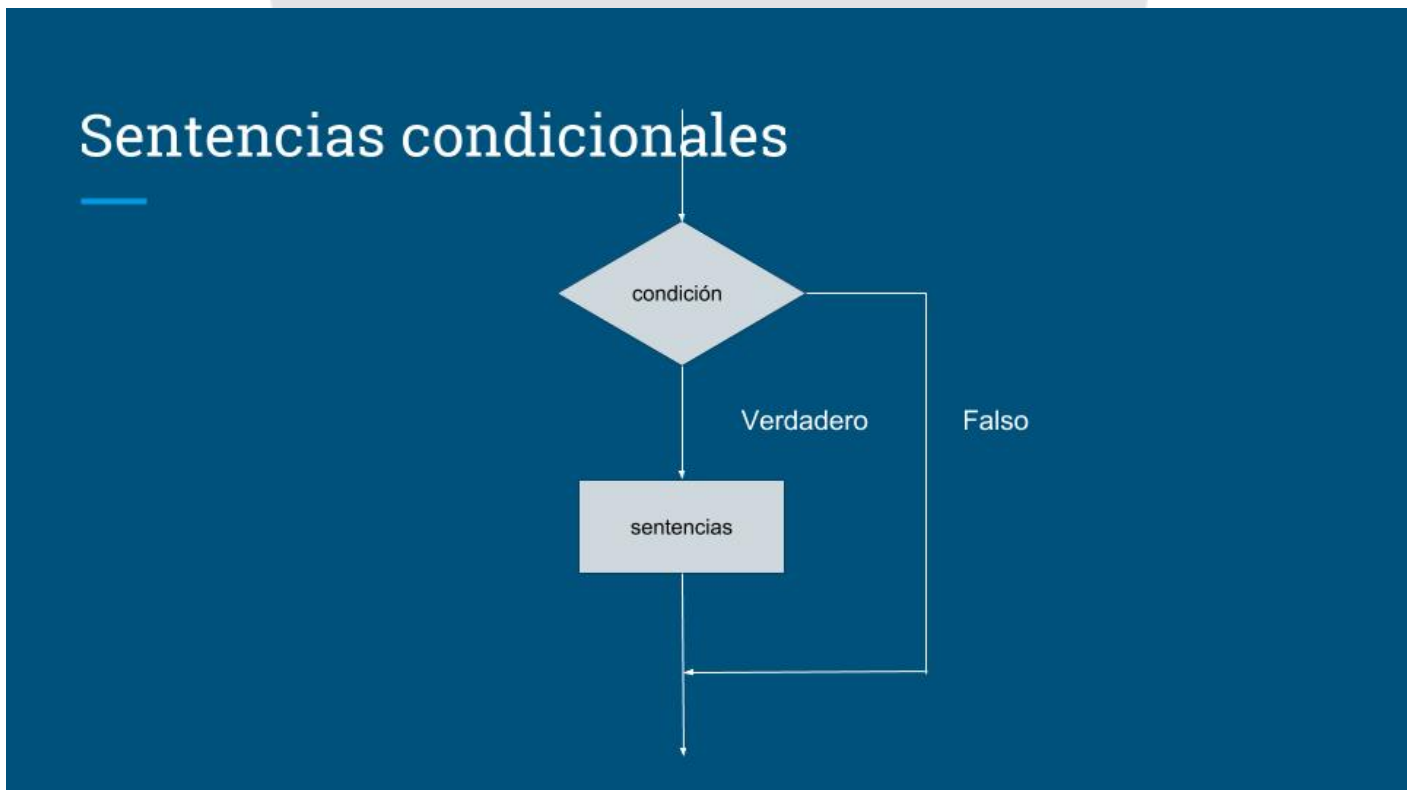


Imagen 4.1.1. La sentencia condicional if

[www.pacoarce.com](http://www.pacoarce.com)

## Sentencias condicionales

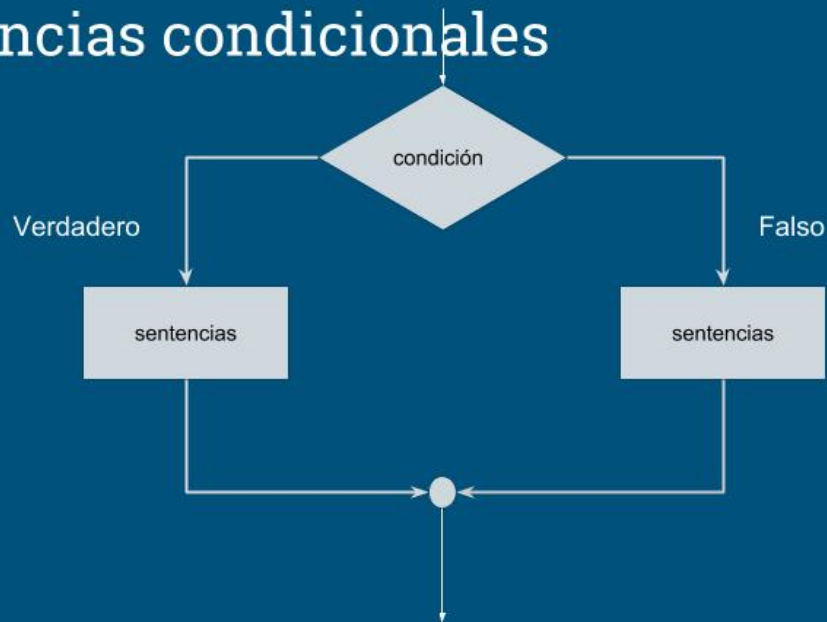


Imagen 4.1.2. La sentencia condicional if-else

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Condicionales</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <?php
9     $edad = 56;
10    //si es verdadero, true
11    if ($edad>40) {
12      print "<p>Lo sentimos, esta página puede resultar peligrosa para tu salud</p>";
13    } else if($edad>30){
14      print "<p>Bienvenido a nuestra página, donde encontrarás lo que siempre haz
15 buscado</p>";
16    } else if($edad>18){
17      print "<p>Bienvenido a nuestra página, estás en edad de comprar nuestros
18 productos</p>";
19    } else {
20      print "<p>Lo sentimos, no puedes entrar a nuestra página</p>";
21    }
22   ?>
23 </body>
24 </html>
```

Listado 4.1.1. PHP7-0401-Condicionales.php

## 4.2. Operadores condicionales anidados

- Es posible incluir un comando "if" dentro de otro comando "if". En este caso es conveniente alinear los comandos y sus "llaves" ( { y } ) para facilitar la lectura y comprensión de la lógica de la programación.
- Nota: se recomienda no "anidar" más de tres sentencias condicionales, por cuestiones de claridad.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Condicionales</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $edad = 56;
10     //si es verdadero, true
11     if ($edad>40) {
12       print "<p>Lo sentimos, esta página puede resultar peligrosa para tu salud</p>";
13     } else if($edad>30){
14       print "<p>Bienvenido a nuestra página, donde encontrarás lo que siempre haz
15 buscado</p>";
16     } else if($edad>18){
17       print "<p>Bienvenido a nuestra página, estás en edad de comprar nuestros
18 productos</p>";
19     } else {
20       print "<p>Lo sentimos, no puedes entrar a nuestra página</p>";
21     }
22   ?>
23 </body>
24 </html>
```

**Listado 4.2.1.** PHP7-0401-Condicionales.php

*www.pacoarce.com*

## 4.3. Crear rangos con operadores condicionales

A veces es necesario preguntar por una serie de "rangos" en las condiciones, y se deben escribir varios estatutos "if", ya sea independientes o unos dentro de otros.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Condicionales Rangos</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $edad = 18;
10     if ($edad <= 20) {
11       print "<p>Te regalamos boletos para el cine</p>";
12     }
13     if ($edad > 20 && $edad <=30) {
14       print "<p>Te regalamos boletos para el concierto de Katy Perry</p>";
15     }
16     if ($edad > 30 && $edad <=40) {
17       print "<p>Te regalamos boletos para el teatro</p>";
18     }
19     if ($edad > 40) {
20       print "<p>Te regalamos boletos para la Ópera</p>";
21     }
22   ?>
23 </body>
24 </html>
```

**Listado 4.3.1.** PHP7-0403-CondicionalesRangos.php

*www.pacoarce.com*

## 4.4. Sintaxis alterna a las sentencias condicionales

- No podemos combinar ambas sintaxis, con llaves y sin llaves. En términos generales esta notación no es recomendada porque es poco flexible.
- Cuando tenemos una sola sentencia tanto en la sentencia afirmativa como en la negativa (else), podemos omitir las llaves.
- También podemos sustituir las llaves por dos puntos (:), pero no podemos combinar ambas sintaxis.
- Nota: en esta sintaxis debe utilizar la sentencia "elseif" sin espacio.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Condicionales sintaxis alterna</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $edad = 45;
10     if($edad > 40) print "<p>Te regalamos boletos para la Ópera</p>";
11     else print "<p>Te regalamos boletos para el cine</p>";
12
13     $a = 18;
14     $b = 18;
15     if($a > $b):
16       print $a." es mayor a ".$b;
17     elseif ($a < $b):
18       print $a." es menor a ".$b;
19     else:
20       print $a." es igual a ".$b;
21     endif;
22   ?>
23 </body>
24 </html>
```

**Listado 4.4.1.** PHP7-0404-CondicionalesSintaxisAlterna.php

*www.pacoarce.com*



## 4.5. El operador condicional

- Un atajo de un comando "if" con solo un comando para el caso "true" y uno solo para el caso "false" lo proporciona este operador.
- Nota: Este tipo de operador no es recomendado porque resulta, por lo general, poco claro.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Operador condicional</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $total = 12345;
10     $tipoCambio = 0;
11     $granTotal = ($tipoCambio>0)? $total * $tipoCambio : $total;
12     $cadena = ($tipoCambio>0)? ", tipo de cambio ".$tipoCambio : "";
13     print "El gran total del pedido es ".$granTotal.$cadena;
14
15  ?>
16 </body>
17 </html>
```

**Listado 4.5.1.** PHP7-0405-OperadorCondicional.php

*www.pacoarce.com*

## 4.6. La estructura condicional switch

- Podemos agrupar más de una comparación, equivalente a una sentencia "or".
- Si una evaluación es verdadera se ejecutarán todas las sentencias hasta encontrar una sentencia "break". Si ninguna sentencia se cumple, podemos añadir una sentencia "default".
- Una alternativa a usar varios comandos "if" es el comando "switch". A diferencia de los comandos "if", el comando "switch" sólo puede evaluar una sola expresión.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Sentencia switch</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8  <?php
9    $estado = "soltera";
10   print "Lo invitamos a un viaje por el Amazonas ";
11   switch ($estado) {
12     case 'soltero':
13     case 'soltera':
14       print "donde podrá conocer a otras personas como usted ";
15       break;
16
17     case 'casado':
18     case 'casada':
19       print "convivir con su familia ";
20       break;
21
22     case 'divorciado':
23       print "para que olvide sus problemas ";
24       break;
25
26     case 'viudo':
27       print "donde podrá reflexionar en sus mejores recuerdos ";
28       break;
29
30     case 'es complicado':
31     case 'arreguntado':
32     case 'separado':
33       print "donde podrá simplificar su vida ";
34       break;
35
36     default:
37       print "donde podrá definir su vida ";
38   }
39   print " en una extraordinaria aventura de dos semanas, todo incluido.";
40  ?>
41 </body>
```

42 </html>

**Listado 4.6.1.** PHP7-0406-Switch.php



*www.pacoarce.com*

## Cuestionario:

**1) ¿Cuál de estas instrucciones está correctamente escrita en PHP?**

- 1) if (a=0) print a
- 3) if (a==0) { echo ok }
- 4) if (a==0): print a;

**2) Una sentencia condicional sólo puede evaluar expresiones que regresen un resultado booleano, es decir, verdadero (true) o falso (false):**

- 1) Verdadero
- 2) Falso

**3) La condición, dentro de una estructura if , va dentro de:**

- 1) Paréntesis ( )
- 2) Llaves { }
- 3) Corchetes [ ]
- 4) Comillas dobles " "

**4) Para evaluar si dos valores son iguales, en PHP, se utiliza el operador:**

- 1) i=
- 2) =
- 3) ==
- 4) .eq.

**5) Una comparación estricta en una condicional de PHP, es decir, dos valores deben tener el mismo valor y ser del mismo tipo, se representa con el símbolo:**

- 1) i=
- 2) =
- 3) ==
- 4) ===

**6) En PHP, para representar una desigualdad, se utiliza el símbolo:**

- 1) ===
- 2) !=
- 3) ==
- 4) ===

**7) En PHP, en una sentencia condicional, el bloque de sentencias que se ejecutarán si se**

**cumple la misma, se delimita mediante:**

- 1) Paréntesis ( )
- 2) Corchetes [ ]
- 3) Llaves { }
- 4) Comillas dobles " "

**8) Para ejecutar un bloque de sentencias, si la condición resulta FALSA, se utiliza la sentencia:**

- 1) else
- 2) then
- 3) FALSE
- 4) no\_true

**9) En PHP podemos tener una sentencia ELSE sin una sentencia IF**

- 1) Verdadero
- 2) Falso

**10) El operador lógico AND, se representa en PHP como:**

- 1) []
- 2) {}
- 3) ||
- 4) &&

**11) En PHP, el operador condicional OR se representa con:**

- 1) []
- 2) {}
- 3) &&
- 4) ||

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 5: Las estructuras cíclicas

**Objetivo:** El alumno conocerá y aplicará los diferentes estructuras cíclicas de PHP como el ciclo while, do...while, for y los comandos break y continue.

#### Introducción:

Los temas que veremos en este capítulo son:

- El ciclo While
- El ciclo do... while
- El ciclo for()
- Los comandos break y continue

*www.pacoarce.com*

## 5.1. El ciclo While

- En el ciclo “while” primero se hace la pregunta, y si es verdadera, se ejecuta un bloque de sentencias y se vuelve a repetir la pregunta, hasta que ésta resulte falsa.
- Los ciclos son sentencias condicionales, pero si la condición es verdadera (true) se repite (ciclo) hasta que la condición resulte falsa (false).

### El ciclo while

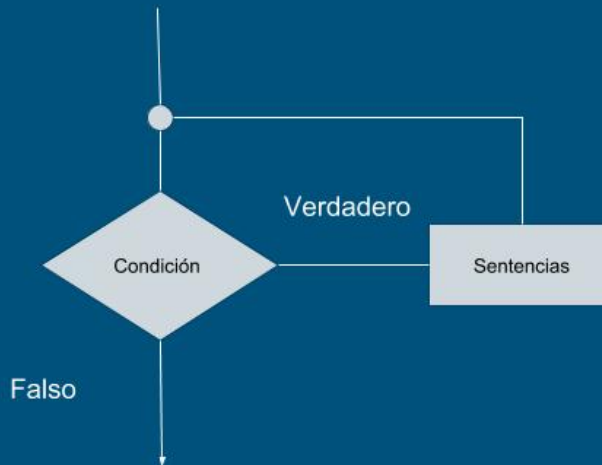


Imagen 5.1.1. El ciclo while()

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ciclo while</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9 $contador = 1;
10 $veces = 10;
11 print "<p>Se van a visualizar ".$veces." números</p>";
12 while ($contador <= $veces) {
13   print "<p>Número: ".$contador."</p>";
14   $contador++;
15 }
16 print "Fin del ciclo, el contador tiene el valor de ".$contador;
```

```
17 ?>  
18 </body>  
19 </html>
```

**Listado 5.1.1.** PHP7-0501-While.php



*www.pacoarce.com*



## 5.2. El ciclo do... while

- La sentencia do..while() primero se hace el bloque de sentencias y luego la expresión condicional.
- También aplican las reglas de las llaves de las estructuras condicionales.
- Cada repetición del bucle se llama "iteración".
- La representación del ciclo do..while es el siguiente:

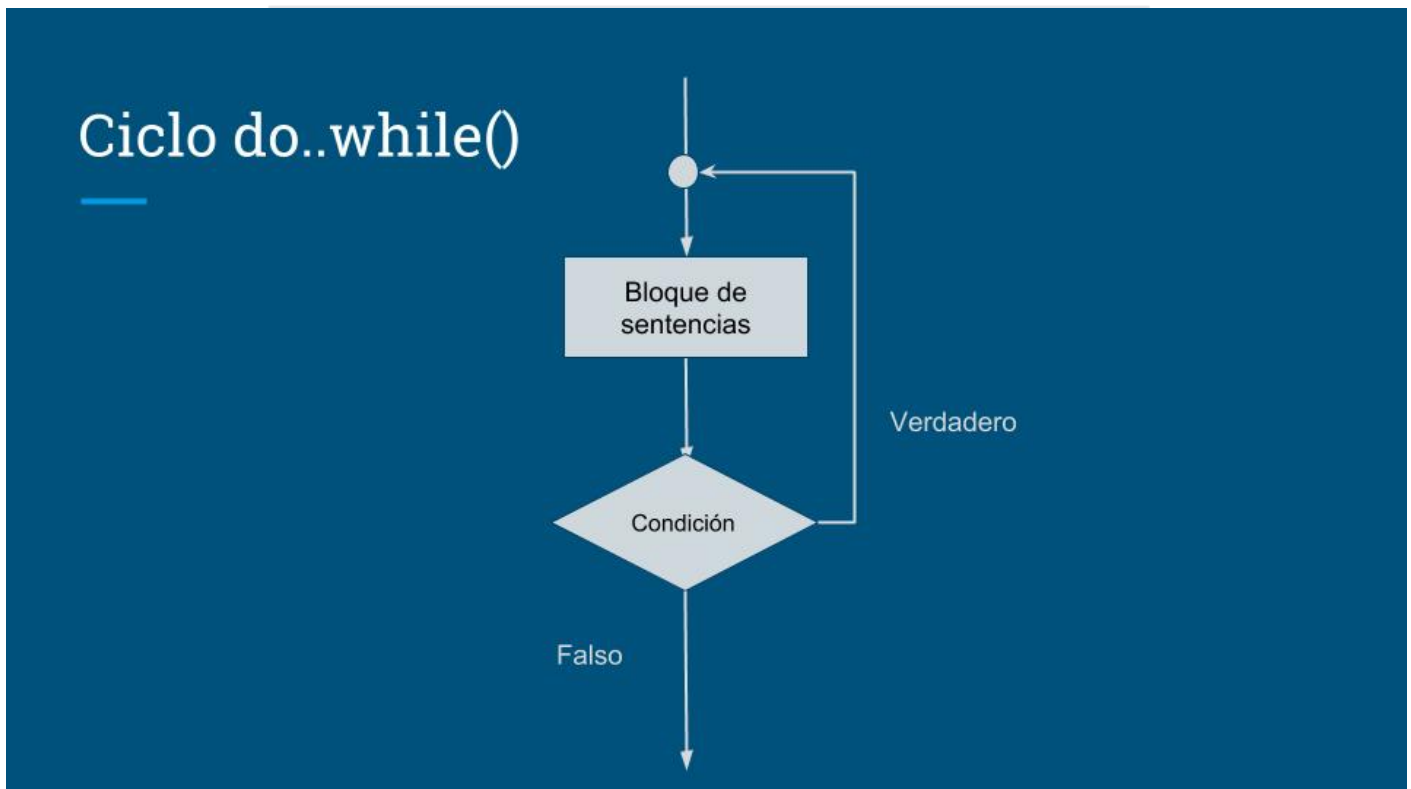


Imagen 5.2.1. El ciclo do..while()

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ciclo do while</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9 $numero = 5;
10 $binario = "";
11 do{
12   $d = $numero % 2;
13   //concatenamos la cadena
14   $binario = $d.$binario;
15   $numero = intval($numero/2);
16 } while($numero>=2);
```

```
17 $binario = $numero.$binario;  
18 print "<p>El número binario es ".$binario."</p>";  
19 ?>  
20 </body>  
21 </html>
```

**Listado 5.2.1.** PHP7-0502-DoWhile.php



*www.pacoarce.com*

## 5.3. El ciclo for

- Como se observa en los ejemplos anteriores, es muy común usar una variable "contador", la cual es inicializada en algún valor(en los ejemplos fue "cero") e incrementada en un valor a cada vuelta del ciclo (en los ejemplos se incrementó en "1").
- El comando "for" representa una manera "diferente" de hacer un ciclo de tipo "while", pero con la inicialización de la variable "contador", y su incremento, en una sola línea.

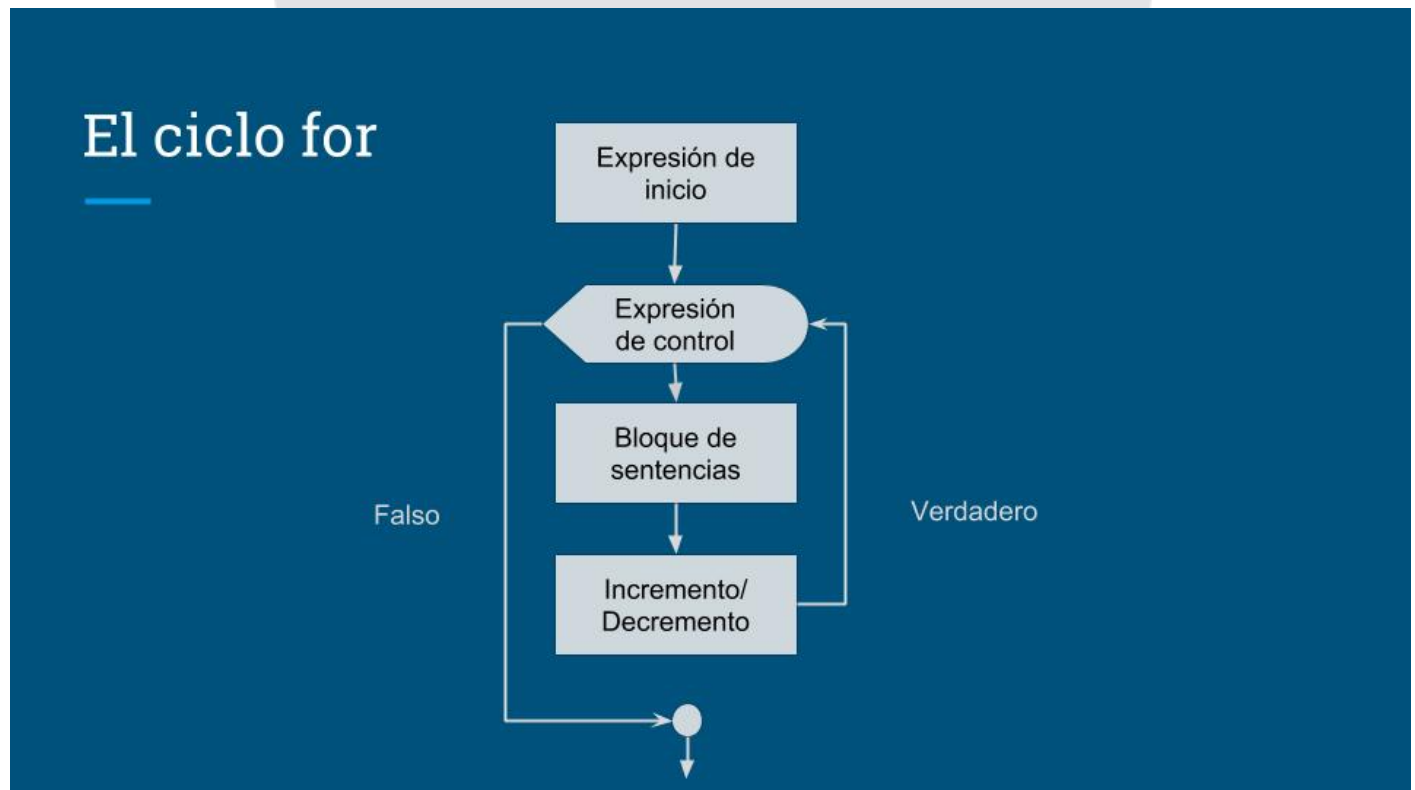


Imagen 5.3.1. El ciclo for()

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ciclo for</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9   print "<ul>";
10  for ($i=0; $i < 10 ; $i++) {
11    print "<li>El valor de i es ".$i."</li>";
12  }
13  print "</ul>";
14  print "<ul>";
15  for ($i=10; $i > 0 ; $i--) print "<li>El valor de i es ".$i."</li>";
```

```
16 print "</ul>";
17
18 $i=10;
19 $j=0;
20 print "<ul>";
21 for (; $i > 0 ; $i--, $j++) print "<li>El valor de i es ".$i.", valor de j es
22 ".$j."</li>";
23 print "</ul>";
24 ?>
25 </body>
26 </html>
```

**Listado 5.3.1.** PHP7-0503-For.php

*www.pacoarce.com*

## 5.4. Los comandos break y continue

- El comando "continue" repite el ciclo sin ejecutar las instrucciones que estén después de ésta.
- Este comando funciona dentro de cualquier comando de ciclos y condicionales (if, switch, while, do...while y for).
- El comando "break" termina el ciclo y no vuelve a evaluar la expresión, terminando definitivamente el ciclo, sin importar el valor de la expresión.
- A veces es necesario salir prematuramente de un ciclo, debido a una condición excepcional.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Sentencias Break y Continue</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    for ($i=0; $i <10 ; $i++) {
10     if ($i==7) {
11       print "<p>Este es el número mágico: ".$i;
12       break;
13     }
14     print "<p>El valor de i es: ".$i;
15   }
16   print "<p>Fin del ciclo</p>";
17   for ($i=0; $i <10 ; $i++) {
18     if($i%2==1) continue;
19     print "<p>El valor de i es: ".$i;
20   }
21   print "<p>Fin del ciclo</p>";
22   /*** tercer ejemplo ***/
23   for ($i=0; ; $i++) {
24     if($i>10) break;
25     print "<p>El valor de i es: ".$i;
26   }
27   print "<p>Fin del ciclo</p>";
28   /*** cuarto ejemplo ***/
29   $i=0;
30   for ( ; ; ) {
31     if($i>10) break;
32     print "<p>El valor de i es: ".$i;
33     $i++;
34   }
35   print "<p>Fin del ciclo</p>";
36   ?>
37   ?>
38 </body>
39 </html>
```

**Listado 5.4.1.** PHP7-0504-BreakContinue.php

## Cuestionario:

### 1) En PHP, ¿qué hace la sentencia break?

- 1) Finaliza la ejecución del método actual
- 2) Finaliza la ejecución del código
- 3) Salta a la siguiente iteración en el bucle for, foreach, while o do-while actual
- 4) Finaliza la ejecución del bucle for, foreach, while o do-while actual

### 2) En PHP, ¿qué sentencia de control está destinada a ser usada con arrays?

- 1) for
- 2) foreach
- 3) switch
- 4) while

### 3) En PHP, ¿qué hace la sentencia continue?

- 1) Finaliza la ejecución del método actual
- 2) Finaliza la ejecución del código
- 3) Salta a la siguiente iteración en el bucle for, foreach, while o do-while actual
- 4) Finaliza la ejecución del bucle for, foreach, while o do-while actual

### 4) El ciclo while hace la verificación de la condicional:

- 1) Al inicio del ciclo
- 2) Al final del ciclo
- 3) No hace validación
- 4) Al inicio y al final

### 5) El ciclo do-while hace la verificación de la condicional:

- 1) Al inicio del ciclo
- 2) Al final del ciclo
- 3) No hace validación
- 4) Al inicio y al final

### 6) El ciclo que podemos utilizar cuando sabemos el inicio y final del mismo, utilizamos:

- 1) for
- 2) while
- 3) do..while
- 4) else if

**7) La instrucción para hacer que se repita un ciclo y no se ejecute las instrucciones inferiores es:**

- 1) break
- 2) loop
- 3) next
- 4) continue

**8) La instrucción para salir de un ciclo y ya no ejecuta más instrucciones del mismo es:**

- 1) continue
- 2) loop
- 3) break
- 4) next

**9) Los elementos de un ciclo "for" se separan con:**

- 1) Dos puntos (:)
- 2) Punto y coma (;)
- 3) Puntos (.)
- 4) Dobles comillas ("")

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 6: Creación y manejo de arreglos

**Objetivo:** El alumno aprenderá a crear los arreglos, la estructura cíclica foreach y las funciones para añadir y eliminar elementos, así como el ordenamiento de arreglos numéricos y asociativos.

#### Introducción:

En esta sección veremos los siguientes temas:

- Crear arreglos
- El ciclo foreach
- Arreglos asociativos
- Algunas funciones de los arreglos
- Ordenamiento de arreglos
- Ordenar los arreglos asociativos
- Desordenar un arreglo y selección aleatoria de elementos de un arreglo
- Funciones para ordenar arreglos en forma ascendente y descendente
- Funciones de navegación de elemento en un arreglo
- PHP7: Arreglos de constantes
- La función `var_dump()` para mostrar el contenido de una variable
- La función `print_r()` para visualizar una variable, arreglo u objeto
- Arreglos multidimensionales
- Arreglos globales
- El comando `List()` para asignar valores de arreglos
- Funciones para arreglos asociativos



## 6.1. Crear arreglos

- Si las variables son como cajas, los arreglos o arrays serían como cajoneras:
- En la gran mayoría de los lenguajes, la numeración de los arreglos inician en cero:
- En PHP los arreglos inician en cero y pueden contener diferente tipo de datos:

### Arreglos

Si las variables son como cajas, los arreglos o arrays serían como cajoneras:



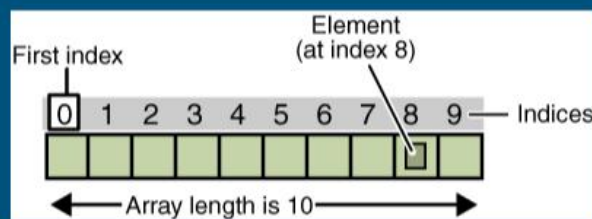
**Imagen 6.1.1.** Los arreglos son como anaqueles de muchas cajas

[www.pacoarce.com](http://www.pacoarce.com)

## Arreglos

En la gran mayoría de los lenguajes, la numeración de los arreglos inician en cero:

0	1	2	3	4
---	---	---	---	---



**Imagen 6.1.2.** Los arreglos en PHP inician en cero

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Condicionales</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $edad = 56;
10     //si es verdadero, true
11     if ($edad>40) {
12       print "<p>Lo sentimos, esta página puede resultar peligrosa para tu salud</p>";
13     } else if($edad>30){
14       print "<p>Bienvenido a nuestra página, donde encontrarás lo que siempre haz
15 buscado</p>";
16     } else if($edad>18){
17       print "<p>Bienvenido a nuestra página, estás en edad de comprar nuestros
18 productos</p>";
19     } else {
20       print "<p>Lo sentimos, no puedes entrar a nuestra página</p>";
21     }
22   ?>
23 </body>
24 </html>

```

**Listado 6.1.1.** PHP7-0401-Condicionales.php

## 6.2. El ciclo foreach

Para evitar todo este manejo de contadores, tamaño del arreglo y el uso de los caracteres "[" y "]" se puede usar el comando "foreach" para recorrer todo un arreglo.

La sintaxis básica para este comando es:

```
foreach( nombre-del-arreglo as elemento ) {  
    sentencias;  
}
```

---

```
1  <!DOCTYPE html>  
2  <html>  
3  <head>  
4    <title>ARREGLOS</title>  
5    <meta charset="utf-8">  
6  </head>  
7  <body>  
8    <?php  
9    $meses = array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",  
10   "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre");  
11   $dias[] = "Lunes";  
12   $dias[] = "Martes";  
13   $dias[] = "Miércoles";  
14   $dias[] = "Jueves";  
15   $dias[] = "Viernes";  
16   $dias[] = "Sábado";  
17   $dias[] = "Domingo";  
18  
19   $alumnos[0] = "Alejandro";  
20   $alumnos[1] = "Michell";  
21   $alumnos[2] = "Jessica";  
22   $alumnos[3] = "Laura";  
23   $alumnos[4] = "Pedro";  
24   $alumnos[5] = "Pablo";  
25  
26   foreach ($meses as $key => $value) {  
27     print "<p>El campo ".$key." tiene un valor de ".$value."</p>";  
28   }  
29  
30   foreach ($dias as $key => $value) {  
31     print "<p>El campo ".$key." tiene un valor de ".$value."</p>";  
32   }  
33   ?>  
34 </body>  
35 </html>
```

**Listado 6.2.1.** PHP7-0602-ArreglosForeach.php

## 6.3. Arreglos asociativos

- PHP permite crear arreglos cuyos elementos tienen nombres en vez de números.
- No siempre es conveniente usar números para identificar los elementos de un arreglo. Para el humano es más fácil usar "letreros" o "nombres" para identificar las cosas.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Arreglos asociados</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $empleado = array(
10     "nombre" => "James",
11     "apellidos" => "Bond",
12     "edad" => 40,
13     "num" => "007"
14   );
15   print "Nombre: ".$empleado["nombre"]." ".$empleado["apellidos"];
16   foreach ($empleado as $llave => $valor) {
17     print "<p>El valor de la propiedad ".$llave." es de: ".$valor."</p>";
18   }
19   $alumno["nombre"] = "Pedro";
20   $alumno["apellido"] = "Picapiedra";
21   $alumno["puesto"] = "Montacargas";
22   foreach ($alumno as $llave => $valor) {
23     print "<p>El valor de la propiedad <b>".$llave."</b> es de: ".$valor."</p>";
24   }
25   print "El nombre del empleado es $empleado[nombre], ".$empleado['apellidos'];
26   ?>
27 </body>
28 </html>
```

**Listado 6.3.1.** PHP7-0603-ArreglosAsociados.php

*www.pacoarce.com*

## 6.4. Algunas funciones de los arreglos

- Buscar por un valor exacto en todo el arreglo, con el comando "in\_array".
- Quitar cualquier elemento de un arreglo, con el comando "unset"s
- Quitar el primer elemento de un arreglo, con el comando "array\_shift"
- Agregar varios elementos al principio del arreglo, con el comando "array\_push"
- Mezclar dos o más arreglos en uno solo, con el comando "array\_merge".

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>ARREGLOS</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8  <?php
9  $meses = array("Enero","Febrero","Marzo","Abril","Mayo","Junio",
10 "Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre");
11 $dias[] = "Lunes";
12 $dias[] = "Martes";
13 $dias[] = "Miércoles";
14 $dias[] = "Jueves";
15 $dias[] = "Viernes";
16 $dias[] = "Sábado";
17 $dias[] = "Domingo";
18
19 $alumnos[0] = "Alejandro";
20 $alumnos[1] = "Michell";
21 $alumnos[2] = "Jessica";
22 $alumnos[3] = "Laura";
23 $alumnos[4] = "Pedro";
24 $alumnos[5] = "Pablo";
25
26 array_push($meses,"Treceavo mes","Catorceavo mes");
27 array_unshift($dias, "primero","primerisimo");
28 $mesPrimero = array_shift($meses);
29 $mesesNuevos = array_slice($meses,11,2);
30 $diasEliminados = array_splice($dias, 5, 2);
31
32 $mesesDias = array_merge($meses,$dias);
33 foreach ($mesesDias as $key => $value) {
34   print "<p>Elemento ".$key." con valor: ".$value."</p>";
35 }
36 print "<hr>";
37 foreach ($mesesNuevos as $key => $value) {
38   print "<p>Elemento ".$key." con valor: ".$value."</p>";
39 }
40 print "<hr>";
41 foreach ($diasEliminados as $key => $value) {
```

```
42 print "<p>Elemento ".$key." con valor: ".$value."</p>";
43 }
44 ?>
45 </body>
46 </html>
```

**Listado 6.4.1.** PHP7-0604-ArreglosFunciones.php



*www.pacoarce.com*

## 6.5. Ordenamiento de arreglos

- A veces es necesario ordenar los elementos de un arreglo. Esto se logra fácilmente con el comando "sort", el cual ordena todos los elementos de un arreglo de manera "ascendente".

`bool sort ( $array [, int $sort_flags = SORT_REGULAR ] );`

- SORT\_REGULAR - compara elementos normalmente (no cambia los tipos)
- SORT\_NUMERIC - compara elementos de forma numérica
- SORT\_STRING - compara elementos como cadenas
- SORT\_LOCALE\_STRING - compara elementos como cadenas, basándose en la configuración regional en uso. Utiliza la configuración regional, la cual puede cambiarse usando `setlocale()`
- .SORT\_FLAG\_CASE - se puede combinar (OR a nivel de bits) con SORT\_STRING o SORT\_NATURAL para ordenar cadenas de forma insensible a mayúsculas/minúsculas.
- SORT\_NATURAL - compara elementos como cadenas usando el "orden natural" de la misma forma que `natsort()`.

Advertencia: Se ha de tener cuidado cuando se ordenen arreglos con valores de tipos mixtos ya que `sort()` puede producir resultados impredecibles.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>ARREGLOS</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $meses = array("Enero","Febrero","Marzo","Abril","Mayo","Junio",
10 "Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre");
11 $dias[] = "Lunes";
12 $dias[] = "Martes";
13 $dias[] = "Miércoles";
14 $dias[] = "Jueves";
15 $dias[] = "Viernes";
16 $dias[] = "Sábado";
17 $dias[] = "Domingo";
18
19 $alumnos[0] = "miguel";
20 $alumnos[1] = "Michell";
21 $alumnos[2] = "luisa";
22 $alumnos[3] = "Laura";
23 $alumnos[4] = "pedro";
24 $alumnos[5] = "Pablo";
25
26 $cal = array(2,3,5,4,60,40,20,30,200,100,25);
27 /*
28 sort($meses);
29 foreach ($meses as $key => $value) {
30   print "<p>Meses[".$key."] = ".$value."</p>";
```

```
31 }
32
33 sort($alumnos, SORT_NATURAL | SORT_FLAG_CASE);
34 foreach ($alumnos as $key => $value) {
35     print "<p>Alumnos[\".$key.\"] = ".$value."</p>";
36 }
37 */
38 sort($cal);
39 foreach ($cal as $key => $value) {
40     print "<p>Calificación[\".$key.\"] = ".$value."</p>";
41 }
42 ?>
43 </body>
44 </html>
```

**Listado 6.5.1.** PHP7-0605-ArreglosSort.php

*www.pacoarce.com*



## 6.6. Ordenar los arreglos asociativos

`asort` — Ordena un array y mantiene la asociación de índices

`bool asort ( array &$array [, int $sort_flags = SORT_REGULAR ] );`

Esta función ordena un array manteniendo la correlación de los índices del array con los elementos con los que están asociados. Esta función se utiliza principalmente para ordenar arrays asociativos en los que el orden es importante.

- `SORT_LOCALE_STRING` - compara elementos como cadenas, basándose en la configuración regional en uso. Utiliza la configuración regional, la cual puede cambiarse usando `setlocale()`.
- `SORT_STRING` - compara elementos como cadenas
- `SORT_NUMERIC` - compara elementos de forma numérica
- `SORT_REGULAR` - compara elementos normalmente (no cambia los tipos)
- `SORT_FLAG_CASE` - se puede combinar (OR a nivel de bits) con `SORT_STRING` o `SORT_NATURAL` para ordenar cadenas de forma insensible a mayúsculas/minúsculas.
- `SORT_NATURAL` - compara elementos como cadenas usando el "orden natural" de la misma forma que `natsort()`.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Arreglos asociados</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8  <?php
9    $empleado = array(
10     "nombre" => "James",
11     "apellidos" => "Bond",
12     "edad" => 40,
13     "num" => "007"
14   );
15   print "Nombre: ".$empleado["nombre"]." ".$empleado["apellidos"];
16   foreach ($empleado as $llave => $valor) {
17     print "<p>El valor de la propiedad ".$llave." es de: ".$valor."</p>";
18   }
19   print "<hr>";
20   asort($empleado);
21   foreach ($empleado as $llave => $valor) {
22     print "<p>El valor de la propiedad ".$llave." es de: ".$valor."</p>";
23   }
24   ?>
```

**Listado 6.6.1.** PHP7-0606-ArreglosAsociadosSort.php

## 6.7. Desordenar un arreglo y selección aleatoria de elementos de un arreglo

- La función *shuffle()* mezcla un arreglo (crea un orden aleatorio de sus elementos). Utiliza un generador de números seudo aleatorios que no es apto para fines criptográficos.

**bool shuffle ( array &\$arreglo )**

- La función *array\_rand()* selecciona uno o más valores aleatorios de un arreglo y devuelve la clave (o claves) de dichos valores aleatorios.
- Utiliza un generador de números seudo aleatorios que no es apto para fines criptográficos.

**mixed array\_rand ( array \$array [, int \$num = 1 ] )**

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>ARREGLOS</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $meses = array("Enero","Febrero","Marzo","Abril","Mayo","Junio",
10 "Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre");
11 $dias[] = "Lunes";
12 $dias[] = "Martes";
13 $dias[] = "Miércoles";
14 $dias[] = "Jueves";
15 $dias[] = "Viernes";
16 $dias[] = "Sábado";
17 $dias[] = "Domingo";
18
19 $alumnos[0] = "Alejandro";
20 $alumnos[1] = "Michell";
21 $alumnos[2] = "Jessica";
22 $alumnos[3] = "Laura";
23 $alumnos[4] = "Pedro";
24 $alumnos[5] = "Pablo";
25
26 shuffle($alumnos);
27 print "<ul>";
28 foreach ($alumnos as $key => $value) {
29   print "<li>".$value."</li>";
30 }
31 print "</ul>";
32
33 $examen = array_rand($alumnos,2);
34 print "<ul>";
35 foreach ($examen as $key => $value) {
36   print "<li>".$value." ".$alumnos[$value]."</li>";
```

```
37 }
38 print "</ul>";
39
40 print "<ul>";
41 foreach ($alumnos as $key => $value) {
42     print "<li>".$value."</li>";
43 }
44 print "</ul>";
45 ?>
46 </body>
47 </html>
```

**Listado 6.7.1.** arreglosDesordenar.php



*www.pacoarce.com*

## 6.8. Funciones para ordenar arreglos en forma ascendente y descendente

- `sort()` - Ordena un arreglo en orden ascendente.
- `rsort()` - Ordena un arreglo en orden descendente.
- `asort()` - Ordena un arreglo asociado en orden ascendente según un valor.
- `ksort()` - Ordena un arreglo asociado en orden ascendente según una llave.
- `arsort()` - Ordena un arreglo asociado en orden descendente según un valor.
- `krsort()` - Ordena un arreglo asociado en orden descendente según una llave.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Arreglos | Otras funciones de ordenado</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8  <?php
9    $scal = array(2,3,5,4,60,40,20,30,200,100,25);
10   rsort($scal);
11   var_dump($scal);
12   foreach ($scal as $key => $value) {
13     print "<p>Calificación[\".$key.\"] = ".$value."</p>";
14   }
15   print "<hr>";
16   $empleado = array(
17     "nombre" => "James",
18     "apellidos" => "Bond",
19     "edad" => 40,
20     "num" => "007"
21   );
22   print "<hr>";
23   arsort($empleado);
24   foreach ($empleado as $llave => $valor) {
25     print "<p>El valor de la propiedad ".$llave." es de: ".$valor."</p>";
26   }
27   print "<hr>";
28   ksort($empleado);
29   foreach ($empleado as $llave => $valor) {
30     print "<p>El valor de la propiedad ".$llave." es de: ".$valor."</p>";
31   }
32   print "<hr>";
33   krsort($empleado);
34   foreach ($empleado as $llave => $valor) {
35     print "<p>El valor de la propiedad ".$llave." es de: ".$valor."</p>";
36   }
37   ?>
38 </body>
```

39 </html>

**Listado 6.8.1.** PHP7-0608-ArreglosFuncionesOrdenado.php



*www.pacoarce.com*

## 6.9. Funciones de navegación de elemento en un arreglo

- `current()` - Devuelve el elemento actual en un arreglo
- `each()` - Devolver el par clave/valor actual de un arreglo y avanzar el cursor del arreglo
- `prev()` - Rebobina el puntero interno del arreglo
- `reset()` - Establece el puntero interno de un arreglo a su primer elemento
- `next()` - Avanza el puntero interno de un arreglo
- `end()` - Establece el puntero interno de un arreglo a su último elemento

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Arreglos | Funciones de ubicación</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $meses = array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio",
10 "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre",
11 "treceavo");
12 print current($meses). "<br>";
13 print next($meses). "<br>";
14 print end($meses). "<br>";
15 print prev($meses). "<br>";
16 print next($meses). "<br>";
17 print next($meses). "<br>";
18 print current($meses). "<br>";
19 ?>
20 </body>
21 </html>
```

**Listado 6.9.1.** PHP7-0609-ArreglosFuncionesUbicacion.php

*www.pacoarce.com*

## 6.10. PHP7: Arreglos de constantes

Ahora se pueden definir constantes de array con define().

En PHP 5.6, solamente se podían definir con const.

---

```
1 <?php
2 define('FRUTAS',["manzana","pera","uvas","sandia"]);
3 print "<pre>";
4 var_dump(FRUTAS);
5 print "</pre>";
6 print FRUTAS[1];
7 FRUTAS[1]="Zanahoria";
8 ?>
```

**Listado 6.10.1.** arreglosConstantes.php

*www.pacoarce.com*

## 6.11. La función var\_dump() para mostrar el contenido de una variable

**var\_dump** — Muestra información sobre una variable.

**void var\_dump ( mixed \$expression [, mixed \$... ] )**

- Esta función muestra información estructurada sobre una o más expresiones incluyendo su tipo y valor.
- Los arreglos y los objetos son explorados recursivamente con valores sangrados para mostrar su estructura.

---

```
1  <?php
2   $arreglo = array("lunes","martes",
3   array("manzana","pera", "uvas",
4   array(1,2,3,4,5)),true
5   );
6   var_dump($arreglo);
7   print "<br>";
8   $hola = true;
9   var_dump($hola);
10 ?>
```

**Listado 6.11.1.** var\_dump.php

*www.pacoarce.com*



## 6.12. La función print\_r() para visualizar una variable, arreglo u objeto

print\_r() — Muestra información sobre una variable.

mixed print\_r ( mixed \$expression [, bool \$return = false ] );

---

```
1  <?php
2  $arreglo = array("lunes","martes",
3  array("manzana","pera", "uvas",
4  array(1,2,3,4,5)),true);
5  print "<pre>";
6  print_r($arreglo);
7  $hola = true;
8  $salida = print_r($hola, true);
9  print "</pre>";
10 print $salida;
11 ?>
```

**Listado 6.12.1.** print\_r.php

*www.pacoarce.com*

## 6.13. Arreglos multidimensionales

Podemos combinar arreglos con índices numéricos o asociativos.

En PHP podemos hacer arreglos de dos o más dimensiones.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Arreglos | Multimiensionales</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $ciudades = array(
10     array("México", "CDMX", "Guadalajara", "Monterrey"),
11     array("España", "Madrid", "Barcelona", "Bilbao"),
12     array("Colombia", "Bogotá", "Bucaramanga", "Medellin"),
13     array("Perú", "Lima", "Cuzco", "Arequipa", "Chiclayo", "Ayacucho")
14 );
15
16 for($i = 0; $i< count($ciudades); $i++){
17     print "<ul>";
18     for($j=0; $j<count($ciudades[$i]); $j++){
19         print "<li>".$ciudades[$i][$j]."</li>";
20     }
21     print "</ul>";
22 }
23 ?>
24 </body>
25 </html>
```

**Listado 6.13.1.** PHP7-0613-ArreglosMultidimensionales.php

*www.pacoarce.com*

## 6.14. Arreglos globales o "superglobals"

Cada vez que se crea una página en PHP, se generan automáticamente varios arreglos que se les conoce como “arreglos globales” o “arreglos de ambiente”.

No es necesario crearlos, pero pueden crearse vacíos.

**Las variables globales son:**

`$_SESSION`  
`$_COOKIE`  
`$_ENV`  
`$_FILES`  
`$_GET`  
`$_POST`  
`$_REQUEST`  
`$_SERVER`  
`$GLOBALS`

El arreglo `$GLOBALS` nos permite utilizar una variable en cualquier parte del programa.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Arreglos | Superglobals</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $nombre = "James";
10     $apellido = "Bond";
11
12     function saludo(){
13       return "Hola ".$GLOBALS['nombre']." ".$GLOBALS['apellido'];
14     }
15
16     print saludo();
17     print "<hr>";
18     print $_SERVER["PHP_SELF"]."<br>";
19     print $_SERVER["SERVER_NAME"]."<br>";
20     print $_SERVER["HTTP_HOST"]."<br>";
21     print $_SERVER["HTTP_REFERER"]."<br>";
22     print $_SERVER["SCRIPT_NAME"]."<br>";
23     print $_SERVER["HTTP_USER_AGENT"]."<br>";
24   ?>
25 </body>
26 </html>
```

**Listado 6.14.1.** PHP7-0614-Superglobals.php

## 6.15. El comando List() para asignar valores de arreglos

*list* — Asignar variables como si fueran un arreglo.

**array *list* ( mixed \$var1 [, mixed \$... ] )**

Al igual que *array()*, no es realmente una función, es un constructor del lenguaje. *list()* se utiliza para asignar una lista de variables en una sola operación.

**Nota:** *list()* solo funciona con arrays numéricos y supone que los índices numéricos empiezan en 0.

### **Advertencia**

**: En PHP 5, *list()* asigna los valores empezando desde el parámetro más a la derecha. En PHP 7, *list()* empieza desde el parámetro más a la izquierda.**

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Arreglos | list</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $info = array("guayaba","Psidium","Myrtaceae","del Caribe");
10     list($nombre,$cientifico,$familia,$origen) = $info;
11
12     print "La ".$nombre." (".$cientifico.") son un género de unas
13     cien especies de árboles tropicales y árboles pequeños en la
14     familia ".$familia.", nativas ".$origen."<hr>";
15
16     list($nombre,$cientifico,, $origen) = $info;
17     print "La ".$nombre." (".$cientifico.") son un género de unas
18     cien especies de árboles tropicales y árboles pequeños,
19     nativas ".$origen."<hr>";
20
21     list($nombre,$cientifico,,) = $info;
22     print "La ".$nombre." (".$cientifico.") es una fruta deliciosa<hr>";
23     list($nombre,,, ) = $info;
24     print "La ".$nombre." es la fruta con mayor cantidad de vitamina C<hr>";
25     list($nombre) = "guayaba";
26     print "La ".$nombre." es la fruta con mayor cantidad de vitamina C<hr>";
27   ?>
28 </body>
29 </html>
```

**Listado 6.15.1.** PHP7-0615-ArreglosList.php

## 6.16. Funciones para arreglos asociativos

- **array\_search()** - Busca un valor determinado en un arreglo y devuelve la primera llave correspondiente en caso de encontrarla.
- **array\_key\_exists()** - Verifica si el índice o llave dada existe en el arreglo.
- **array\_combine()** - Crea un nuevo arreglo, usando un arreglo para las llaves y otra para los valores.
- **array\_keys()** - Devuelve todas las llaves de un arreglo o un subconjunto de llaves de un arreglo.
- **array\_values()** - Devuelve todos los valores como un arreglo indexado.

---

```
1  <?php
2  print "1. array_values() toma un arreglo asociativo y lo convierte en
   indexado<br>";
3  $arreglo = array("cielo"=>"azul", "pasto"=>"verde", "manzana"=>"rojo");
4  print_r(array_values($arreglo));
5  print "<br><br>";
6
7  print "2. array_keys() Devuelve todas las llaves de un arreglo o un subconjunto de
   llaves de un arreglo.<br>";
8  $arreglo = array("cielo"=>"azul", "pasto"=>"verde", "manzana"=>"rojo");
9  print_r(array_keys($arreglo));
10 print "<br><br>";
11
12 print "2. array_keys() Regresa todas las coincidencias en un arreglo.<br>";
13 $arreglo = array("azul", "verde", "rojo", "verde", "rojo", "azul");
14 print_r(array_keys($arreglo, "rojo"));
15 print "<br><br>";
16
17 print "3. array_combine: Crea un nuevo arreglo, usando un arreglo para las llaves y
   otra para los valores.<br>";
18 $llaves = array("cielo", "pasto", "manzana");
19 $valores = array("azul", "verde", "rojo");
20 print_r(array_combine($llaves, $valores));
21 print "<br><br>";
22
23 print "4. array_key_exists() Verifica si el índice o llave dada existe en el
   arreglo..<br>";
24 $arreglo = array("cielo"=>"azul", "pasto"=>"verde", "manzana"=>"rojo");
25 print_r(array_key_exists("perico", $arreglo));
26 print "<br><br>";
27
28 print "5. array_search() Busca un valor determinado en un arreglo y devuelve la
   primera llave correspondiente en caso de encontrarla.<br>";
29 $arreglo = array("cielo"=>"azul", "pasto"=>"verde", "manzana"=>"rojo");
30 print_r(array_search("perico", $arreglo));
31 print "<br>";
32 print_r(array_search("azul", $arreglo));
33 print "<br>";
34 print_r(array_search("rojo", $arreglo));
35 print "<br>";
```

36 ?>

**Listado 6.16.1.** funcionesArreglosAsociativos.php



*www.pacoarce.com*

## Cuestionario:

**1) Un arreglo o array es:**

- 1) Un conjunto de caracteres alfanuméricos
- 2) Un sistema para convertir una variable de texto en un número
- 3) Un conjunto de elementos

**2) En PHP, ¿qué información se almacena en la variable superglobal \$\_SERVER['HTTP\_HOST']?**

- 1) La dirección IP del servidor
- 2) El nombre del servidor
- 3) La versión de HTTP del servidor
- 4) Las anteriores respuestas no son correctas

**3) En PHP, para unir todos los elementos de un array en una cadena se emplea la función**

- 1) concat()
- 2) implode()
- 3) union()
- 4) Las anteriores respuestas no son correctas

**4) En PHP, para buscar un elemento en un array se puede emplear:**

- 1) find()
- 2) in\_array()
- 3) search()
- 4) sort()

**5) En PHP, ¿qué información se almacena en la variable superglobal \$\_SERVER['REMOTE\_ADDR']?**

- 1) La dirección IP del servidor
- 2) La dirección IP del visitante
- 3) La dirección IP del servidor de DNS
- 4) Las anteriores respuestas no son correctas

**6) En PHP, la sentencia de control foreach se emplea para**

- 1) Recorrer los elementos de un arreglo o array
- 2) Recorrer las propiedades de un objeto
- 3) Recorrer los elementos de un array y las propiedades de un objeto
- 4) Las anteriores respuestas no son correctas

**7) En PHP, \$language['php'] es un ejemplo de**

- 1) Array asociativo
- 2) Array multidimensional
- 3) Array superglobal
- 4) Las anteriores respuestas no son correctas

**8) En PHP, ¿cómo se obtiene la información de un formulario que ha sido enviado mediante el método "get"?**

- 1) \$\_URL[]
- 2) \$\_GET[]
- 3) \$\_FORM[]
- 4) Las anteriores respuestas no son correctas

**9) En PHP, ¿cuál es el nombre del tipo de dato que permite almacenar múltiples valores separados?**

- 1) array
- 2) float
- 3) string
- 4) Las anteriores respuestas no son correctas

**10) En PHP, para aplicar una función a cada elemento de un array se emplea la función**

- 1) array\_reset()
- 2) array\_apped()
- 3) foreach()
- 4) Las anteriores respuestas no son correctas

**11) El ciclo adecuado para recorrer un arreglo es:**

- 1) for()
- 2) while()
- 3) list()
- 4) foreach()

**12) Mezclar dos o más arreglos en uno solo, con el comando:**

- 1) array\_push
- 2) array\_merge
- 3) array\_shift
- 4) unset



**13) Agregar varios elementos al principio del arreglo, con el comando:**

- 1) array\_push
- 2) array\_merge
- 3) array\_shift
- 4) unset

**14) Quitar el primer elemento de un arreglo, con el comando:**

- 1) array\_push
- 2) array\_merge
- 3) array\_shift
- 4) unset

**15) Quitar cualquier elemento de un arreglo, con el comando:**

- 1) array\_push
- 2) array\_merge
- 3) array\_shift
- 4) unset

**16) Buscar por un valor exacto en todo el arreglo, con el comando:**

- 1) in\_array
- 2) array\_merge
- 3) array\_shift
- 4) unset

**17) Si deseas ordenar un arreglo con indice numérico en forma ascendente utilizamos:**

- 1) asort()
- 2) sort()
- 3) merge()
- 4) ascendent()

**18) Para ordenar un arreglo asociativo utilizamos:**

- 1) asort()
- 2) sort()
- 3) merge()
- 4) ascendent()

**19) Para desordenar un arreglo utilizamos:**

- 1) join()

- 2) sort()
- 3) merge()
- 4) shuffle()

**20) Ordena un arreglo en orden descendente.**

- 1) sort()
- 2) rsort()
- 3) ksort()
- 4) arsort()

**21) Ordena un arreglo asociado en orden ascendente según una llave.**

- 1) sort()
- 2) rsort()
- 3) ksort()
- 4) arsort()

**22) Ordena un arreglo asociado en orden descendente según un valor.**

- 1) sort()
- 2) rsort()
- 3) ksort()
- 4) arsort()

**23) Ordena un arreglo asociado en orden descendente según una llave.**

- 1) krsort()
- 2) rsort()
- 3) ksort()
- 4) arsort()

**24) Devuelve el elemento actual en un arreglo**

- 1) prev()
- 2) each()
- 3) current()
- 4) reset()

**25) Devolver el par clave/valor actual de un arreglo y avanzar el cursor del arreglo**

- 1) prev()
- 2) each()
- 3) current()

4) reset()

## 26) Rebobina el puntero interno del arreglo

- 1) prev()
- 2) each()
- 3) current()
- 4) reset()

## 27) Establece el puntero interno de un arreglo a su primer elemento

- 1) prev()
- 2) each()
- 3) current()
- 4) reset()

## 28) Muestra información sobre una variable.

- 1) var()
- 2) var\_dump()
- 3) print()
- 4) echo()

## 29) Asignar variables como si fueran un arreglo.

- 1) list()
- 2) var\_dump()
- 3) merge()
- 4) arsort()

## 30) La función que devuelve todos los valores de un arreglo asociativo como un arreglo indexado es:

- 1) array\_keys()
- 2) array\_values()
- 3) array\_combine()
- 4) array\_assoc()

## 31) La función que devuelve todas las llaves de un arreglo o un subconjunto de llaves de un arreglo es:

- 1) array\_keys()
- 2) array\_values()
- 3) array\_combine()

4) array\_assoc\_keys()

**32) La función que crea un nuevo arreglo asociativo, usando un arreglo para las llaves y otra para los valores es:**

- 1) array\_keys()
- 2) array\_values()
- 3) array\_create()
- 4) array\_combine()

**33) La función que verifica si el índice o llave dada existe en el arreglo es:**

- 1) array\_keys()
- 2) array\_values()
- 3) array\_key\_exists()
- 4) array\_combine()

**34) La función que busca un valor determinado en un arreglo y devuelve la primera llave correspondiente en caso de encontrarla es:**

- 1) array\_search()
- 2) array\_keys()
- 3) array\_key\_exists()
- 4) array\_combine()

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 7: Creación de funciones personalizadas

**Objetivo:** El alumno creará funciones personalizadas con parámetros y la palabra reservada 'return', así como las nuevas características en PHP7 de los valores predeterminados, declaraciones de tipo.

#### Introducción:

En esta sección trataremos los siguientes temas:

- Crear y llamar a una función
- El alcance de las variables
- Regresar un valor de una función
- Argumentos o parámetros por referencia o por valor
- PHP7: Valores predeterminados de argumentos
- PHP7: Declaraciones de tipo en las funciones
- PHP7: Declaraciones de tipo de devolución
- PHP7: Tipificación estricta
- PHP7: Listas de argumentos de longitud variable
- Crear generadores en PHP
- Utilizar yield para facilitar valores con clave desde un generador
- PHP7: El uso de la sentencia Return en los generadores
- PHP7: Delegación de generadores
- Las funciones anónimas en PHP
- Tomar valores externos desde una función anónima

[www.pacoarce.com](http://www.pacoarce.com)

## 7.1. Crear y llamar a una función

Una **función** es un bloque de código de **PHP** que no es ejecutado inmediatamente, sino que puede ser "llamado" o ejecutado desde el código "principal" varias veces. Una función es la definición de una "rutina", que es probable que se vaya a ejecutar muchas veces.

Una analogía típica para una **función** es una receta de cocina.

La sintaxis general para crear o definir una **función** es la siguiente:

```
function nombre_de_la_receta ( ingredientes ) {  
    comando;  
    comando;  
}
```

El nombre de las funciones sigue las reglas de los nombres de las variables, con una excepción: el símbolo "\$" al principio del nombre

---

```
1  <!DOCTYPE html>  
2  <html>  
3  <head>  
4    <title>Funciones</title>  
5    <meta charset="utf-8">  
6  </head>  
7  <body>  
8    <?php  
9    function saludo($nombre){  
10     print "<h2>Hola " . $nombre . "</h2>";  
11   }  
12   saludo("Pedro Picapiedra");  
13   saludo("Pedro Mármol");  
14   saludo("Juanito Pérez");  
15   ?>  
16 </body>  
17 </html>
```

**Listado 7.1.1.** PHP7-0701-Funciones.php

www.pacoarce.com

## 7.2. El alcance de las variables

- Las variables que se definen fuera de una función se les considera “globales”.
- Las variables que se definen dentro de una función se le considera “local”.
- Los parámetros son variables locales.
- Para que podamos “ver” una variable desde una función, debemos “pedir permiso” con la palabra reservada “global”.

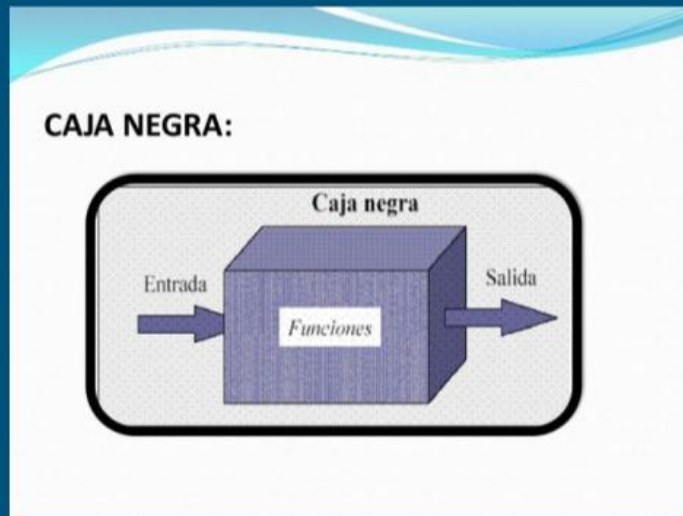
### El alcance de las variables



Imagen 7.2.1. El paradigma de la caja negra

[www.pacoarce.com](http://www.pacoarce.com)

## El alcance de las variables



**Imagen 7.2.2.** Paradigma GIGO: Basura entra, basura sale.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Funciones</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9 /*****
10 Variables
11 *****/
12 $empleado = "Pedro Picapiedra";
13 $mensaje = "buenos días";
14 /*****
15 Funciones
16 *****/
17 function saludo($nombre, $msg){
18   print "<h2>Hola ".$nombre.", ".$msg."</h2>";
19 }
20 /*****
21 Inicio
22 *****/
23 saludo($empleado, $mensaje);
24 ?>
25 </body>
```



26 </html>

**Listado 7.2.1.** PHP7-0702-FuncionesAlcance.php



*www.pacoarce.com*

## 7.3. Regresar un valor de una función

- Las funciones pueden regresar un valor después de ser llamadas, pero solo pueden regresar un valor directamente por medio de la palabra reservada "return".
- Si se desea "simular" el regreso de varios valores, se declaran variables globales y se modifican dentro de la función o puede regresar un objeto o un arreglo.
- Así, al terminar la función dejará una "herencia" de variables globales modificadas.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones Return</title>
5  </head>
6  <body>
7    <?php
8      /*****
9      Variables
10     *****/
11     $subtotal = 1500;
12     $tasa = 0.16;
13     $iva = 0;
14     $edad = 999999;
15     /*****
16     FUNCIONES
17     *****/
18     function calculaIva($sub, $porcien){
19       $ivaCalculo = $sub * (1+$porcien);
20       return $ivaCalculo;
21     }
22     function validaEdad($edad)
23     {
24       $valida = ($edad<=0 || $edad >120);
25       return !$valida;
26     }
27     /*****
28     INICIO
29     *****/
30     print "<p>Subtotal : ".$subtotal."</p>";
31     //calcula iva
32     $iva = calculaIva($subtotal, $tasa);
33     //
34     print "<p>Total con IVA: ".$iva;
35     if (validaEdad($edad)) {
36       print "<p>Edad válida</p>";
37     } else {
38       print "<p>Edad NO válida</p>";
39     }
40     ?>
41 </body>
```

42 </html>

**Listado 7.3.1.** PHP7-0703-FuncionesReturn.php



*www.pacoarce.com*

## 7.4. Argumentos o parámetros por referencia o por valor

- Por defecto, los argumentos o parámetros de las funciones son pasados por valor (así, si el valor del argumento dentro de la función cambia, este no cambia fuera de la función).
- Para permitir a una función modificar sus argumentos, éstos deben pasarse por referencia.

```
1  <?php
2  function saludo(&$nombre)
3  {
4      $nombre .= ", buenos dias";
5  }
6  $nombre = "Pedro Picapiedra";
7  saludo($nombre);
8  print $nombre;
9  //& pasamos por referencia (si se modifica)
10 //por omision se pasa el valor (no se modifica)
11 $fruta = array("manzana","pera");
12 function frutas(&$f)
13 {
14     array_push($f,"uvas");
15 }
16 frutas($fruta);
17 var_dump($fruta);
18 ?>
```

**Listado 7.4.1.** PHP7-0704-ValorReferencia.php

[www.pacoarce.com](http://www.pacoarce.com)

## 7.5. PHP7: Valores predeterminados de argumentos

- Una función puede definir valores predeterminados al estilo de C++ para argumentos escalares.
- PHP también permite el uso de arreglos y del tipo especial NULL como valores predeterminados.
- El valor predeterminado debe ser una expresión constante, no una variable, un miembro de una clase o una llamada a una función.

```
1  <?php
2  function pastel($costo, $sabor=array("limon"))
3  {
4  $salida = "<p>Esto es un pastel sabor ".join(",",$sabor);
5  $salida.= ", con un costo de $".$costo."</p>";
6  return $salida;
7  }
8  $sabores = array("fresa","chocolate");
9  print pastel(100,$sabores);
10 print pastel(200);
11 ?>
```

**Listado 7.5.1.** PHP7-0705-ValoresPredeterminados.php

*www.pacoarce.com*

## 7.6. PHP7: Declaraciones de tipo en las funciones

- Las **declaraciones de tipo** permiten a las funciones requerir que los parámetros sean de cierto tipo durante una llamada.
- Si el valor dado es de un tipo incorrecto, se generará un error: en PHP 5, este error es un error fatal recuperable, mientras que PHP 7 lanzará una excepción TypeError.

```
1 <?php
2 declare(strict_types=1);
3 function hola(bool $nombre)
4 {
5     print "hola " . $nombre;
6 }
7 hola(12345);
8 ?>
```

**Listado 7.6.1.** PHP7-0706-DeclaracionesdeTipo.php

[www.pacoarce.com](http://www.pacoarce.com)

## 7.7. PHP7: Declaraciones de tipo de devolución

- PHP 7 añade soporte para las **declaraciones de tipo de devolución** de las funciones.
- De forma similar a las declaraciones de tipo de argumento, las **declaraciones de tipo de devolución** especifican el tipo del valor que serán devuelto desde una función.
- Están disponibles los mismos tipos para las declaraciones de tipo de devolución que para las declaraciones de tipo de argumento.

```
1  <?php
2  /*
3  declare(strict_types=1);
4  function suma($n1, $n2):bool
5  {
6      return $n1+$n2;
7  }
8  var_dump(suma(10,5));
9  */
10 class Gato{};
11 class Perro{};
12 function regresaGato(): Gato {
13     return new Perro;
14 }
15 var_dump(regresaGato());
16 ?>
```

**Listado 7.7.1.** PHP7-0707-DeclaracionesdeTipoDevolucion.php

[www.pacoarce.com](http://www.pacoarce.com)

## 7.8. PHP7: Tipificación estricta

- Por defecto, **PHP** fuerza a los valores de un tipo erróneo a ser del tipo esperado si es posible.
- Por ejemplo, una función a la que se le pasa un entero o **integer** para un parámetro que se prevé sea una cadena o **string** obtendrá una variable de tipo cadena o **string**.

```
1  <?php
2  declare(strict_types=1);
3  function suma(int $n1, int $n2):float{
4      return $n1+$n2;
5  }
6  //
7  try {
8      var_dump(suma(10,5));
9      var_dump(suma(10.6,5.8));
10 } catch (TypeError $e) {
11     print "<p>Error: ".$e->getMessage();
12 }
13 ?>
```

**Listado 7.8.1.** PHP7-0708-TipificacionEstricta.php

[www.pacoarce.com](http://www.pacoarce.com)



## 7.9. PHP7: Listas de argumentos de longitud variable

- Se puede especificar **argumentos posicionales** normales antes del token .... En este caso, solamente los argumentos al final que no coincidan con un **argumento posicional** serán añadidos al arreglo generado por ...
- También es posible añadir una **declaración de tipo** antes del símbolo ...
- Si está presente, todos los argumentos capturados por ... deben ser objetos de la **clase** implicada.

```
1 <?php
2 function suma($simbolo, int ...$numeros)
3 {
4     $total = 0;
5     foreach ($numeros as $num) {
6         $total += $num;
7     }
8     return $simbolo.$total;
9 }
10 echo suma("euros ", "manzana", 4, 3, 4, 5, 6, 7, 8, 7, 56, 54, 4, 3, 3, 3, 6);
11 ?>
```

**Listado 7.9.1.** PHP7-0709-ListaArgumentos.php

[www.pacoarce.com](http://www.pacoarce.com)

## 7.10. Crear generadores en PHP

- Los generadores son funciones que en lugar de utilizar la palabra reservada “return”, con la cual acaba la función, utilizamos la palabra reservada “yield”, con la cual “entregamos” el valor, pero NO se termina la función, la podemos volver a llamar.
- El uso de generadores nos ayudan a ahorrar mucha memoria RAM en procesos muy grandes.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones | generadores</title>
5  </head>
6  <body>
7  <?php
8  function rango($inicio, $fin, $incremento=1)
9  {
10   for($i=$inicio; $i<=$fin; $i+=$incremento){
11     yield $i;
12   }
13 }
14 foreach (range(1,10) as $key => $value) {
15   print $value." ";
16 }
17 print "<hr>";
18 foreach (rango(1,10) as $key => $value) {
19   print $value." ";
20 }
21 print "<hr>";
22
23 ?>
24 </body>
25 </html>
```

**Listado 7.10.1.** generadores.php

[www.pacoarce.com](http://www.pacoarce.com)

## 7.11. Utilizar yield para facilitar valores con clave desde un generador

- En la palabra reservada “yield” indicamos cuál es la llave.
- Los generadores podemos regresar valores como arreglos asociativos.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones | generador asociativo</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $input = <<<'EOF'
10 1;Sor Juana;Hombres necios
11 2;López Velarde;La suave Patria
12 3;Becquer;Vuelven oscuras golondrinas
13 EOF;
14
15 function poemas($input)
16 {
17     foreach (explode("\n",$input) as $value) {
18         $r = explode(";", $value);
19         $id = array_shift($r);
20         yield $id=>$r;
21     }
22 }
23 foreach (poemas($input) as $id => $poema) {
24     print $id."<br>";
25     print $poema[0]."<br>";
26     print $poema[1]."<br>";
27     print "<hr>";
28 }
29 ?>
30 </body>
31 </html>
```

**Listado 7.11.1.** yieldLlaves.php

## 7.12. PHP7: El uso de la sentencia Return en los generadores

- Esta característica se basa en la funcionalidad de los **generadores**. Habilita a la sentencia “**return**” para utilizarla dentro de un generador para que pueda **devolver** una expresión final (la devolución por referencia no está permitida).
- Este valor se puede obtener empleando el nuevo método **Generator::getReturn()**, el cual solamente se puede utilizar una vez que el generador ha finalizado de producir valores.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Funciones | Gerneradores return</title>
5 </head>
6 <body>
7   <?php
8   $gen = (function(){
9     yield 1;
10    yield 2;
11    return 3;
12  })();
13  foreach ($gen as $value) {
14    print $value."<br>";
15  }
16  print $gen->getReturn()."<br>";
17  ?>
18 </body>
19 </html>
```

**Listado 7.12.1.** generadoresReturn.php

[www.pacoarce.com](http://www.pacoarce.com)

## 7.13. PHP7: Delegación de generadores

Los generadores ahora pueden “**delegar**” a otro generador de forma muy sencilla por medio de la construcción “**yield from**”.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones | Generadores delegados</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    function gen()
10   {
11     yield 1;
12     yield 2;
13     yield from gen2();
14   }
15
16   function gen2()
17   {
18     yield 3;
19     yield 4;
20   }
21
22   foreach (gen() as $val) {
23     print $val." ";
24   }
25   print "<hr>";
26   function americaNorte()
27   {
28     yield "Canadá";
29     yield "México";
30   }
31   function americaCentral()
32   {
33     yield "Costa Rica";
34     yield "Honduras";
35   }
36   function americaSur()
37   {
38     yield "Colombia";
39     yield "Perú";
40   }
41   function capitalesAmerica()
42   {
43     yield from americaNorte();
44     yield from americaCentral();
```

```
45 yield from americaSur();
46 }
47 foreach (capitalesAmerica() as $val) {
48     print $val."<br>";
49 }
50 print "<hr>";
51 ?>
52 </body>
53 </html>
```

**Listado 7.13.1.** yieldFrom.php



*www.pacoarce.com*

## 7.14. Las funciones anónimas en PHP

### Versiones: PHP 5 y 7

- También se utilizan como valores en variables y PHP las convierte en instancias de la clase *Closure*.
- Las **funciones anónimas** están implementadas utilizando la clase *Closure*.
- Son muy útiles como *parámetros* en otras funciones.
- Las **funciones anónimas**, también conocidas como cierres (*closures*), permiten la creación de funciones que no tienen un nombre especificado.

---

```
1  <?php
2  print preg_replace_callback('~-([a-z])~', function($coincidencias){
3      return strtoupper($coincidencias[1]);
4  }, "-hola-mundo");
5  print "<hr>";
6  $saludo = function($nombre){
7      printf("Hola %s<br>", $nombre);
8  };
9  $saludo("Mundo");
10 $saludo("cara de bola");
11 ?>
```

**Listado 7.14.1.** funcionAnonimaParametro.php

[www.pacoarce.com](http://www.pacoarce.com)

## 7.15. Tomar valores externos desde una función anónima

Versiones: PHP 5 y 7

- Desde PHP 7.1 las variables **superglobals**, \$this o variables con el mismo nombre de un parámetro de la función anónima, **NO** pueden pasarse.
- El valor debe ser pasado por medio del constructor “**use**”.
- Las **funciones anónimas** o “cierres” pueden tomar o heredar una variable fuera de su ámbito.

```
1  <?php
2  $mensaje = "hola";
3
4  //Sin el "use"
5  $ejemplo = function(){
6      var_dump($mensaje);
7  };
8  $ejemplo();
9
10 print "<hr>";
11 //Utilizando "use"
12 $ejemplo = function() use ($mensaje){
13     var_dump($mensaje);
14 };
15 $ejemplo();
16
17 print "<hr>";
18 //Utilizando "use" por referencia
19 $mensaje = "adios";
20 $ejemplo = function() use (&$mensaje){
21     var_dump($mensaje);
22 };
23 $ejemplo();
24
25 print "<hr>";
26 //Utilizando "use" por referencia y argumentos
27 $mensaje = "buenas tardes";
28 $ejemplo = function($arg) use (&$mensaje){
29     var_dump($arg, ".$mensaje");
30 };
31 $ejemplo("Pedro Picapiedra");
32
33 ?>
```

**Listado 7.15.1.** funcionesAnonimasUse.php



## Cuestionario:

**1) En PHP, ¿cuál es la forma correcta de declarar una función?**

- 1) function miFuncion()
- 2) create function miFuncion()
- 3) sub miFuncion()
- 4) method miFuncion()

**2) El nombre de las funciones sigue las reglas de los nombres de las variables. También inician con el símbolo "\$".**

- 1) Verdadero
- 2) Falso

**3) Las variables que se definen fuera de una función se les considera:**

- 1) estáticas
- 2) locales
- 3) globales
- 4) naturales

**4) Las variables que se definen dentro de una función se le considera:**

- 1) estáticas
- 2) locales
- 3) globales
- 4) naturales

**5) Los parámetros son variables locales.**

- 1) Verdadero
- 2) Falso

**6) Para acceder a una variable externa desde una funcion debemos pedir permiso con la palabra reservada:**

- 1) public
- 2) private
- 3) local
- 4) global

**7) Las funciones pueden regresar un valor después de ser llamadas, pero solo pueden regresar un valor directamente por medio de la palabra reservada:**

- 1) echo
- 2) print
- 3) return
- 4) yield

**8) Si se desea "simular" el regreso de varios valores, se declaran variables globales y se modifican dentro de la función o puede regresar un objeto o un arreglo.**

- 1) Verdadero
- 2) Falso

**9) Por defecto, los argumentos o parámetros de las funciones son pasados por:**

- 1) notificación
- 2) valor
- 3) referencia
- 4) apuntador

**10) Para permitir a una función modificar sus argumentos, éstos deben pasarse por:**

- 1) notificación
- 2) valor
- 3) referencia
- 4) apuntador

**11) Una función puede definir valores predeterminados al estilo de C++ para argumentos escalares.**

- 1) Verdadero
- 2) Falso

**12) PHP NO permite el uso de arreglos y del tipo especial NULL como valores predeterminados.**

- 1) Verdadero
- 2) Falso

**13) El valor predeterminado de un argumento de una función debe ser:**

- 1) una expresión constante
- 2) una variable
- 3) un miembro de una clase
- 4) una llamada a una función

**14) Las declaraciones de tipo de los parámetros de una función permiten:**

- 1) permiten que las variables sean globales dentro de la función.
- 2) Declaran que las variables sean anónimas.
- 3) requerir que los parámetros sean de cierto tipo durante una llamada.
- 4) declaran a las variables como privadas

**15) Si el valor dado aun parámetro en una función es de un tipo incorrecto, se generará un error recuperable en la versión de PHP:**

- 1) 4.0
- 2) 5.0
- 3) 6.0.
- 4) 7.0.

**16) PHP 7 añade soporte para las declaraciones de tipo de devolución de las funciones:**

- 1) Verdadero
- 2) Falso

**17) Las declaraciones de tipo de argumento son extrictamente diferentes a las declaraciones de tipo de devolución:**

- 1) Verdadero
- 2) Falso

**18) El comando para regresar la información de una función, utilizamos el comando:**

- 1) regresa
- 2) back
- 3) value
- 4) return

**19) Una función en PHP siempre debe tener al menos un parámetro:**

- 1) Verdadero
- 2) Falso

**20) El comando "global" permite leer el contenido de una variable declarada fuera de una función, dentro de una función y modificar el mismo:**

- 1) Verdadero
- 2) Falso

**21) Los parámetros, dentro de la definición de una función, se separan por medio de:**

- 1) Comillas ("")
- 2) Punto y coma (;)

- 3) Dos puntos (:)
- 4) Comas (,)

**22) Las variables locales solo son válidas mientras se ejecuta la función. Cuando la función termina, las variables locales desaparecen:**

- 1) Verdadero
- 2) Falso

**23) En la estructura de la función, los parámetros se encuentran entre:**

- 1) Paréntesis ()
- 2) Corchetes []
- 3) Llaves {}
- 4) Comillas ""

**24) Cuando definimos a una función con la palabra reservada "function", automáticamente se ejecuta la misma...**

- 1) Verdadero
- 2) Falso

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 8: Validación de datos de los formularios

**Objetivo:** El alumno validará los datos de un formulario de HTML con PHP, desde datos escalares como datos múltiples, redireccionamiento a otras páginas con el comando header() y el uso de plantillas.

#### Introducción:

En esta sección veremos los siguientes temas:

- Comunicación de un Formulario con PHP
- Recibir variables simples
- Recibir variables múltiples
- Validación de cadenas
- Validación de arreglos con isset()
- Recibir campos ocultos de formas HTML
- Redireccionar a otras páginas con el comando header
- Uso de plantillas PHP

## 8.1. Comunicación de un Formulario con PHP

- Al usar el PHP7 mediante un servidor de páginas web, el único medio de comunicación "*interactiva*" con los usuarios es a través de formas de HTML.
- En ellas, el usuario escribe datos que son evaluados y procesados por una rutina de PHP7 y se genera un resultado.
- Los métodos para enviar los datos de un formulario a otra página son **GET** y **POST**.
- El método **GET** envía los datos en la URL del navegador, por lo cual son visibles a un posible observador directo o a quedar almacenado en el historial del navegador. No podemos enviar datos binarios, como imágenes.
- El método **POST** envía los valores dentro de los paquetes **HTTP** por lo cual no es visible y podemos enviar datos binarios, como imágenes.
- En ambos casos, al crearse la página PHP, se generan automáticamente los arreglos de ambiente **\$\_GET** y **\$\_POST**.
- También se genera un arreglo **\$\_REQUEST** que contiene ambos métodos, pero no es aconsejado.

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formulario</title>
5 </head>
6 <body>
7   <form action="PHP7-0801-Valida.php" method="POST">
8     <input type="text" name="nombre" id="nombre"/><br>
9     <input type="submit" value="Enviar datos"/>
10  </form>
11 </body>
12 </html>
```

**Listado 8.1.1.** PHP7-0801-Formulario.php

---

```
1 <?php
2 $nombre = $_POST["nombre"];
3 print "<h1>Bienvenido, ".$nombre." a nuestra página</h1>";
4 ?>
```

**Listado 8.1.2.** PHP7-0801-Valida.php

www.pacoarce.com

## 8.2. Recibir variables simples

Los siguientes elementos los recibimos del formulario como cadenas:

- Input (de todo tipo)
- Radio button
- Select sencillo
- Textarea

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formulario</title>
5  </head>
6  <body>
7    <form action="PHP7-0801-Valida.php" method="POST">
8      <input type="text" name="nombre" id="nombre"/><br>
9      <input type="submit" value="Enviar datos"/>
10   </form>
11 </body>
12 </html>
```

**Listado 8.2.1.** PHP7-0801-Formulario.php

---

```
1  <?php
2  $nombre = $_POST["nombre"];
3  $clave = $_POST["clave"];
4  $comentarios = $_POST["comentarios"];
5  $estado = $_POST["estado"];
6  $idioma = $_POST["idioma"];
7  print "<h1>Bienvenido, ".$nombre." a nuestra página</h1>";
8  print "<p>Clave de usuario: ".$clave."</p>";
9  print "<p>Comentarios : ".$comentarios."</p>";
10 print "<p>Estado Civil : ".$estado."</p>";
11 print "<p>Idioma : ".$idioma."</p>";
12 ?>
```

**Listado 8.2.2.** PHP7-0802-Valida.php

*www.pacoarce.com*

## 8.3. Recibir variables múltiples

Los siguientes elementos los recibimos del formulario como arreglos:

- Select múltiple
- Cajas de verificación

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formulario</title>
5  </head>
6  <body>
7    <form action="PHP7-0803-Valida.php" method="POST">
8      <label for="nombre">Nombre:</label><br>
9      <input type="text" name="nombre" id="nombre"/><br>
10     <label for="clave">Clave:</label><br>
11     <input type="password" name="clave" id="clave"/><br>
12     <label for="comentarios">Comentarios:</label><br>
13     <textarea id="comentarios" name="comentarios"></textarea><br>
14     <label for="idioma">Idioma:</label><br>
15     <select id="idioma" name="idioma">
16       <option value="es">Español</option>
17       <option value="en">Inglés</option>
18       <option value="fr">Francés</option>
19     </select><br>
20     <label for="estado">Estado:</label><br>
21     <input type="radio" name="estado" id="soltero" value="soltero"><label
for="soltero">Soltero</label><br>
22     <input type="radio" name="estado" id="casado" value="casado"><label
for="casado">Casado</label><br>
23     <input type="radio" name="estado" id="viudo" value="viudo"><label
for="viudo">Viudo</label><br>
24     <label>Pasatiempos:</label><br>
25     <label><input type="checkbox" name="pasatiempos[]" value="leer"
id="leer">Leer</label><br>
26     <label><input type="checkbox" name="pasatiempos[]" value="dormir"
id="dormir">Dormir</label><br>
27     <label><input type="checkbox" name="pasatiempos[]" value="ajedrez"
id="ajedrez">Ajedrez</label><br><br>
28     <label for="pasteles">Sabor de pasteles preferidos:</label><br>
29     <select multiple="multiple" name="pasteles[]" id="pasteles">
30       <option value="chocolate">Chocolate</option>
31       <option value="fresa">Fresa</option>
32       <option value="vainilla">Vainilla</option>
33       <option value="coco">Coco</option>
34       <option value="napolitano">Napolitano</option>
35     </select>
36     <input type="submit" value="Enviar datos"/>
37   </form>
```



```
38 </body>
39 </html>
```

## Listado 8.3.1. PHP7-0803-Formulario.php

---

```
1  <?php
2  $nombre = $_POST["nombre"];
3  $clave = $_POST["clave"];
4  $comentarios = $_POST["comentarios"];
5  $estado = $_POST["estado"];
6  $idioma = $_POST["idioma"];
7  $pasatiempos = $_POST["pasatiempos"];
8  $pasteles = $_POST["pasteles"];
9  print "<h1>Bienvenido, ".$nombre." a nuestra página</h1>";
10 print "<p>Clave de usuario: ".$clave."</p>";
11 print "<p>Comentarios : ".$comentarios."</p>";
12 print "<p>Estado Civil : ".$estado."</p>";
13 print "<p>Idioma : ".$idioma."</p>";
14 print "<p>Num. pasatiempos: ".count($pasatiempos)."</p>";
15 print "<ol>";
16 foreach ($pasatiempos as $key => $value) {
17     print "<li>".$value."</li>";
18 }
19 print "</ol>";
20 print "<p>Sabor de pasteles: ".count($pasteles)."</p>";
21 print "<ol>";
22 foreach ($pasteles as $key => $value) {
23     print "<li>".$value."</li>";
24 }
25 print "</ol>";
26 ?>
```

## Listado 8.3.2. PHP7-0803-Valida.php

[www.pacoarce.com](http://www.pacoarce.com)

## 8.4. Validación de cadenas

- El primer ejemplo compara la variable `$nombre` contra una cadena "vacío", representado por dos comillas.
- El segundo usa una función integrada que obtiene la longitud de un string (`strlen`) y la compara contra el número 0.
- En ambos casos, si esto llega a ser verdadero (`true`), se ejecutará el código para manejar ese error.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formulario</title>
5  </head>
6  <body>
7    <form action="PHP7-0804-Valida.php" method="POST">
8      <label for="nombre">* Nombre:</label><br>
9      <input type="text" name="nombre" id="nombre"/><br>
10     <label for="clave">* Clave:</label><br>
11     <input type="password" name="clave" id="clave"/><br>
12     <label for="comentarios">Comentarios:</label><br>
13     <textarea id="comentarios" name="comentarios"></textarea><br>
14     <label for="idioma">Idioma:</label><br>
15     <select id="idioma" name="idioma">
16       <option value="es">Español</option>
17       <option value="en">Inglés</option>
18       <option value="fr">Francés</option>
19     </select><br>
20     <label for="estado">Estado:</label><br>
21     <input type="radio" name="estado" id="soltero" value="soltero"><label
for="soltero">Soltero</label><br>
22     <input type="radio" name="estado" id="casado" value="casado"><label
for="casado">Casado</label><br>
23     <input type="radio" name="estado" id="viudo" value="viudo"><label
for="viudo">Viudo</label><br>
24     <label>Pasatiempos:</label><br>
25     <label><input type="checkbox" name="pasatiempos[]" value="leer"
id="leer">Leer</label><br>
26     <label><input type="checkbox" name="pasatiempos[]" value="dormir"
id="dormir">Dormir</label><br>
27     <label><input type="checkbox" name="pasatiempos[]" value="ajedrez"
id="ajedrez">Ajedrez</label><br><br>
28     <label for="pasteles">Sabor de pasteles preferidos:</label><br>
29     <select multiple="multiple" name="pasteles[]" id="pasteles">
30       <option value="chocolate">Chocolate</option>
31       <option value="fresa">Fresa</option>
32       <option value="vainilla">Vainilla</option>
33       <option value="coco">Coco</option>
34       <option value="napolitano">Napolitano</option>
35     </select><br>
36     <input type="submit" value="Enviar datos"/>
```

```
37 </form>
38 </body>
39 </html>
```

## Listado 8.4.1. PHP7-0804-Formulario.php

```
1 <?php
2 $nombre = $_POST["nombre"];
3 $clave = $_POST["clave"];
4 $comentarios = $_POST["comentarios"];
5 $estado = $_POST["estado"];
6 $idioma = $_POST["idioma"];
7 $pasatiempos = $_POST["pasatiempos"];
8 $pasteles = $_POST["pasteles"];
9 /**/ Valicación ***/
10 $error = array();
11 if($nombre==""){
12     array_push($error,"Error: el nombre del usuario no puede estar vacío");
13 }
14 if($clave==""){
15     array_push($error,"Error: la clave de acceso no puede ser vacía");
16 }
17 if($comentarios==""){
18     array_push($error,"Error: los comentarios son requeridos");
19 }
20 if($estado==""){
21     array_push($error,"Error: el estado civil es requerido");
22 }
23 if($idioma==""){
24     array_push($error,"Error: el idioma es requerido");
25 }
26 if(count($error)>0){
27     print "<ul>";
28     foreach ($error as $key => $value) {
29         print "<li>".$value."</li>";
30     }
31     print "</ul>";
32 } else {
33     print "<h1>Bienvenido, ".$nombre." a nuestra página</h1>";
34     print "<p>Clave de usuario: ".$clave."</p>";
35     print "<p>Comentarios : ".$comentarios."</p>";
36     print "<p>Estado Civil : ".$estado."</p>";
37     print "<p>Idioma : ".$idioma."</p>";
38     print "<p>Num. pasatiempos: ".count($pasatiempos)."</p>";
39     print "<ol>";
40     foreach ($pasatiempos as $key => $value) {
41         print "<li>".$value."</li>";
42     }
43     print "</ol>";
44     print "<p>Sabor de pasteles: ".count($pasteles)."</p>";
45     print "<ol>";
46     foreach ($pasteles as $key => $value) {
```

```
47     print "<li>".$value."</li>";  
48 }  
49 print "</ol>";  
50 }  
51  
52 ?>
```

**Listado 8.4.2.** PHP7-0804-Valida.php



*www.pacoarce.com*

## 8.5. Validación de arreglos con isset()

- Para las variables de tipo arreglo, que se manejan en los combos de lista y los cuadros de verificación, lo primero a validar es si existe el arreglo, ya que si el usuario no selecciona ninguna opción, no se genera un arreglo vacío, sino que la variable ni siquiera existe.
- En el caso de existir, es bueno saber cuántos elementos tiene el arreglo.
- Con la función `isset()` nos regresa un valor verdadero si la variable está definida (si existe) y un valor falso si esta variable **NO** está definida.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formulario</title>
5  <?php
6  if(isset($_POST["nombre"])){
7    $nombre = $_POST["nombre"];
8    $clave = $_POST["clave"];
9    $comentarios = $_POST["comentarios"];
10   $estado = $_POST["estado"];
11   $idioma = $_POST["idioma"];
12   $pasatiempos = $_POST["pasatiempos"];
13   $pasteles = $_POST["pasteles"];
14   /** Validación ***/
15   $error = array();
16   if($nombre==""){
17     array_push($error,"Error: el nombre del usuario no puede estar vacío");
18   }
19   if($clave==""){
20     array_push($error,"Error: la clave de acceso no puede ser vacía");
21   }
22   if($comentarios==""){
23     array_push($error,"Error: los comentarios son requeridos");
24   }
25   if($estado==""){
26     array_push($error,"Error: el estado civil es requerido");
27   }
28   if($idioma==""){
29     array_push($error,"Error: el idioma es requerido");
30   }
31   if(isset($pasatiempos)){
32     if(count($pasatiempos)==0){
33       array_push($error,"Error: debe de seleccionar al menos un pasatiempo");
34     }
35   } else {
36     array_push($error,"Error: debe de seleccionar al menos un pasatiempo");
37   }
38   if(isset($pasteles)){
39     if(count($pasteles)==0){
40       array_push($error,"Error: debe de seleccionar al menos un sabor de pastel");
41     }
42   }
```

# Todo PHP7

```
41     }
42     } else {
43         array_push($error,"Error: debe de seleccionar al menos un sabor de pastel");
44     }
45     if(count($error)>0){
46         print "<ul>";
47         foreach ($error as $key => $value) {
48             print "<li>".$value."</li>";
49         }
50         print "</ul>";
51     } else {
52         print "<h1>Bienvenido, ".$nombre." a nuestra página</h1>";
53         print "<p>Clave de usuario: ".$clave."</p>";
54         print "<p>Comentarios : ".$comentarios."</p>";
55         print "<p>Estado Civil : ".$estado."</p>";
56         print "<p>Idioma : ".$idioma."</p>";
57         print "<p>Num. pasatiempos: ".count($pasatiempos)."</p>";
58         print "<ol>";
59         foreach ($pasatiempos as $key => $value) {
60             print "<li>".$value."</li>";
61         }
62         print "</ol>";
63         print "<p>Sabor de pasteles: ".count($pasteles)."</p>";
64         print "<ol>";
65         foreach ($pasteles as $key => $value) {
66             print "<li>".$value."</li>";
67         }
68         print "</ol>";
69     }
70 }
71
72 ?>
73 </head>
74 <body>
75     <form action="PHP7-0805-Formulario.php" method="POST">
76         <label for="nombre">* Nombre:</label><br>
77         <input type="text" name="nombre" id="nombre"/><br>
78         <label for="clave">* Clave:</label><br>
79         <input type="password" name="clave" id="clave"/><br>
80         <label for="comentarios">Comentarios:</label><br>
81         <textarea id="comentarios" name="comentarios"></textarea><br>
82         <label for="idioma">Idioma:</label><br>
83         <select id="idioma" name="idioma">
84             <option value="es">Español</option>
85             <option value="en">Inglés</option>
86             <option value="fr">Francés</option>
87         </select><br>
88         <label for="estado">Estado:</label><br>
89         <input type="radio" name="estado" id="soltero" value="soltero"><label
for="soltero">Soltero</label><br>
90         <input type="radio" name="estado" id="casado" value="casado"><label
for="casado">Casado</label><br>
```

```
91     <input type="radio" name="estado" id="viudo" value="viudo"><label  
for="viudo">Viudo</label><br>  
92     <label>Pasatiempos:</label><br>  
93     <label><input type="checkbox" name="pasatiempos[]" value="leer"  
id="leer">Leer</label><br>  
94     <label><input type="checkbox" name="pasatiempos[]" value="dormir"  
id="dormir">Dormir</label><br>  
95     <label><input type="checkbox" name="pasatiempos[]" value="ajedrez"  
id="ajedrez">Ajedrez</label><br><br>  
96     <label for="pasteles">Sabor de pasteles preferidos:</label><br>  
97     <select multiple="multiple" name="pasteles[]" id="pasteles">  
98         <option value="chocolate">Chocolate</option>  
99         <option value="fresa">Fresa</option>  
100        <option value="vainilla">Vainilla</option>  
101        <option value="coco">Coco</option>  
102        <option value="napolitano">Napolitano</option>  
103    </select><br>  
104    <input type="submit" value="Enviar datos"/>  
105 </form>  
106 </body>  
107 </html>
```

**Listado 8.5.1.** PHP7-0805-Formulario.php

*www.pacoarce.com*

## 8.6. Recibir campos ocultos de formas HTML

- A veces es necesario incluir valores que sean asignados a una variable al mandar una forma, pero sin que el usuario se entere. Para esto se usan los campos ocultos.
- Supongamos que se desea pasar los nombres de las plantillas de HTML que se van a mostrar si hubo un error en el llenado de la forma, y otra cuando el llenado fue exitoso.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formulario</title>
5    <style>
6      .error{ color:red; margin:0; padding: 0;}
7    </style>
8  <?php
9  if(isset($_POST["bandera"])){
10    //recibimos los valores y los almacenamos en variables de trabajo
11    $nombre = $_POST["nombre"];
12    $clave = $_POST["clave"];
13    $comentarios = $_POST["comentarios"];
14    $estado = $_POST["estado"];
15    $idioma = $_POST["idioma"];
16    $pasatiempos = $_POST["pasatiempos"];
17    $pasteles = $_POST["pasteles"];
18    /** Validación **/
19    $error = array();
20    if($nombre==""){
21      $error[0] = "Error: el nombre del usuario no puede estar vacío";
22    }
23    if($clave==""){
24      $error[1] = "Error: la clave de acceso no puede ser vacía";
25    }
26    if($comentarios==""){
27      $error[2] = "Error: los comentarios son requeridos";
28    }
29    if($estado==""){
30      $error[3] = "Error: el estado civil es requerido";
31    }
32    if($idioma==""){
33      $error[4] = "Error: el idioma es requerido";
34    }
35    if(isset($pasatiempos)){
36      if(count($pasatiempos)==0){
37        $error[5] = "Error: debe de seleccionar al menos un pasatiempo";
38      }
39    } else {
40      $error[6] = "Error: debe de seleccionar al menos un pasatiempo";
41    }
}
```



```
42  if(isset($pasteles)){
43      if(count($pasteles)==0){
44          $error[7] = "Error: debe de seleccionar al menos un sabor de pastel";
45      }
46  } else {
47      $error[8] = "Error: debe de seleccionar al menos un sabor de pastel";
48  }
49  }
50
51  ?>
52  </head>
53  <body>
54      <?php
55      if(count($error)==0 && isset($_POST["bandera"])){
56          print "<h1>Bienvenido, ".$nombre." a nuestra página</h1>";
57          print "<p>Clave de usuario: ".$clave."</p>";
58          print "<p>Comentarios : ".$comentarios."</p>";
59          print "<p>Estado Civil : ".$estado."</p>";
60          print "<p>Idioma : ".$idioma."</p>";
61          print "<p>Num. pasatiempos: ".count($pasatiempos)."</p>";
62          print "<ol>";
63          foreach ($pasatiempos as $key => $value) {
64              print "<li>".$value."</li>";
65          }
66          print "</ol>";
67          print "<p>Sabor de pasteles: ".count($pasteles)."</p>";
68          print "<ol>";
69          foreach ($pasteles as $key => $value) {
70              print "<li>".$value."</li>";
71          }
72          print "</ol>";
73      } else {
74          ?>
75      <form action="PHP7-0806-Formulario.php" method="POST">
76          <label for="nombre">* Nombre:</label><br>
77          <input type="text" name="nombre" id="nombre"/><br>
78          <?php
79          if(isset($error[0])) print "<p class='error'>".$error[0]."</p>";
80          ?>
81          <label for="clave">* Clave:</label><br>
82          <input type="password" name="clave" id="clave"/><br>
83          <?php
84          if(isset($error[1])) print "<p class='error'>".$error[1]."</p>";
85          ?>
86          <label for="comentarios">Comentarios:</label><br>
87          <textarea id="comentarios" name="comentarios"></textarea><br>
88          <?php
89          if(isset($error[2])) print "<p class='error'>".$error[2]."</p>";
90          ?>
91          <label for="idioma">Idioma:</label><br>
92          <select id="idioma" name="idioma">
93              <option value="es">Español</option>
```

```
94     <option value="en">Inglés</option>
95     <option value="fr">Francés</option>
96 </select><br>
97 <?php
98 if(isset($error[3])) print "<p class='error'>".$error[3]."</p>";
99 ?>
100 <label for="estado">Estado:</label><br>
101 <input type="radio" name="estado" id="soltero" value="soltero"><label
102 for="soltero">Soltero</label><br>
103 <input type="radio" name="estado" id="casado" value="casado"><label
104 for="casado">Casado</label><br>
105 <input type="radio" name="estado" id="viudo" value="viudo"><label
106 for="viudo">Viudo</label><br>
107 <?php
108 if(isset($error[4])) print "<p class='error'>".$error[4]."</p>";
109 ?>
110 <label>Pasatiempos:</label><br>
111 <label><input type="checkbox" name="pasatiempos[]" value="leer"
112 id="leer">Leer</label><br>
113 <label><input type="checkbox" name="pasatiempos[]" value="dormir"
114 id="dormir">Dormir</label><br>
115 <label><input type="checkbox" name="pasatiempos[]" value="ajedrez"
116 id="ajedrez">Ajedrez</label><br><br>
117 <?php
118 if(isset($error[5])) print "<p class='error'>".$error[5]."</p>";
119 if(isset($error[6])) print "<p class='error'>".$error[6]."</p>";
120 ?>
121 <label for="pasteles">Sabor de pasteles preferidos:</label><br>
122 <select multiple="multiple" name="pasteles[]" id="pasteles">
123 <option value="chocolate">Chocolate</option>
124 <option value="fresa">Fresa</option>
125 <option value="vainilla">Vainilla</option>
126 <option value="coco">Coco</option>
127 <option value="napolitano">Napolitano</option>
128 </select><br>
129 <?php
130 if(isset($error[7])) print "<p class='error'>".$error[7]."</p>";
131 if(isset($error[8])) print "<p class='error'>".$error[8]."</p>";
132 ?>
133 <input type="hidden" name="bandera" id="bandera" value="bandera">
134 <input type="submit" value="Enviar datos"/>
135 </form>
136 <?php } ?>
137 </body>
138 </html>
```

**Listado 8.6.1.** PHP7-0806-Formulario.php

## 8.7. Redireccionar a otras páginas con el comando header

- Por medio del comando *HEADER* podemos desviar el flujo de la página a otra, pero no debe hacer ninguna salida a la pantalla, ya sea de PHP o HTML.

```
1  <?php
2  if (isset($_GET["quien"])) {
3      $quien = $_GET["quien"];
4      if ($quien=="perro") {
5          header("location:PHP7-0807-FormulariosPerro.php");
6      } else {
7          header("location:PHP7-0807-FormulariosGato.php");
8      }
9  }
10 ?>
11 <!DOCTYPE html>
12 <html>
13 <head>
14     <title>Formularios Header</title>
15     <meta charset="utf-8">
16 </head>
17 <body>
18 <form action="PHP7-0807-FormulariosHeader.php" method="GET">
19     <label>¿Qué soy?</label>
20     <input type="radio" name="quien" id="perro" value="perro"><label
for="perro">Perro</label>
21     <input type="radio" name="quien" id="gato" value="gato"><label
for="gato">Gato</label>
22     <input type="submit" value="Enviar">
23 </form>
24 </body>
25 </html>
```

**Listado 8.7.1.** PHP7-0807-FormulariosHeader.php

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Soy un gato</title>
5  </head>
6  <body>
7      <h1>Soy un gato</h1>
8      
9  </body>
10 </html>
```

**Listado 8.7.2.** PHP7-0807-FormulariosGato.php

```
1  <!DOCTYPE html>
```

```
2 <html>
3 <head>
4   <title>Perro</title>
5 </head>
6 <body>
7   <h1>Soy un perro</h1>
8   
9 </body>
10 </html>
```

**Listado 8.7.3.** PHP7-0807-FormulariosPerro.php



*www.pacoarce.com*

## 8.8. Uso de plantillas PHP

- Por medio de pasar un sólo parámetro podemos utilizar **una plantilla** para reducir el número de archivos necesarios.
- También podemos pasar un parámetro en un **hipervínculo**.

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Animales Plantilla</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <h2>¿Quién soy?</h2>
9 <a href="PHP7-0808-Animales.php?animal=1">Perro</a><br>
10 <a href="PHP7-0808-Animales.php?animal=2">Gato</a><br>
11 <a href="PHP7-0808-Animales.php?animal=3">Gallo</a><br>
12 <a href="PHP7-0808-Animales.php?animal=4">Manati</a><br>
13 </body>
14 </html>
```

**Listado 8.8.1.** PHP7-0808-FormulariosPlantilla.php

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Plantilla animales</title>
5   <meta charset="utf-8">
6   <?php
7     if(isset($_GET["animal"])){
8       $animal = $_GET["animal"];
9       if ($animal==1) {
10         $titulo = "Perro";
11         $imagen = "perro.jpg";
12       }
13       if ($animal==2) {
14         $titulo = "Gato";
15         $imagen = "gato.jpg";
16       }
17       if ($animal==3) {
18         $titulo = "Gallo";
19         $imagen = "gallo.jpg";
20       }
21       if ($animal==4) {
22         $titulo = "Manati";
23         $imagen = "manati.jpg";
24       }
25     }
26
```

```
27  ?>
28  </head>
29  <body>
30  <h1>Soy un <?php print $titulo; ?></h1>
31  
32  </body>
33  </html>
```

**Listado 8.8.2.** PHP7-0808-Animales.php



*www.pacoarce.com*

## Cuestionario:

**1) Dos de las formas de pasar los parámetros entre páginas PHP son:**

- 1) Require e Include
- 2) Get y Put
- 3) Post y Get
- 4) Into e Include

**2) ¿Para qué sirve el siguiente código?: if (isset(\$variable)){}**

- 1) Recorre un array de nombre \$variable
- 2) Crea una variable de nombre "\$variable"
- 3) Verifica si la variable "\$variable" está definida y tiene un valor no nulo
- 4) Ninguna de las anteriores

**3) Cuando tenemos que enviar los datos de un formulario a una página php para que los procese. ¿En qué la propiedad del formulario indicaremos el nombre de la página?**

- 1) ninguna es correcta
- 2) action
- 3) method
- 4) name

**4) Cuando la propiedad action de un formulario se establece como una cadena vacía o no se indica, significa que:**

- 1) Los datos se tienen que procesar en la misma página donde se define el formulario
- 2) Ninguna es correcta
- 3) Es indiferente, puesto que el navegador nos preguntará a donde deseamos enviar los datos
- 4) No queremos que el formulario sea procesado

**5) Cuando se emplea el método "post" de envío de un formulario, ¿los datos del formulario se muestran en la URL?**

- 1) Siempre
- 2) Sí, si así se ha indicado en el formulario
- 3) Nunca
- 4) Sólo si transmitimos una imagen

**6) En PHP, ¿qué función se emplea para realizar una redirección?**

- 1) header()
- 2) redirect()

3) send()

4) Las anteriores respuestas no son correctas

**7) En PHP, ¿cómo se obtiene la información de un formulario que ha sido enviado mediante el método "get"?**

1) \$\_URL[]

2) \$\_GET[]

3) \$\_FORM[]

4) Las anteriores respuestas no son correctas

**8) Cuando un cliente web envía la información de un formulario al servidor**

1) Las distintas parejas campo=valor se separan por espacios en blanco

2) Las distintas parejas campo=valor se separan por "&"

3) Las distintas parejas campo=valor se separan por "+"

4) Las distintas parejas campo=valor se separan por "%"

**9) PHP, ¿cómo se obtiene la información que se envía a través de la URL (query string)?**

1) \$\_GET

2) \$\_POST

3) \$\_QUERY

4) \$\_URL En

**10) El método que envía el valor de las variables dentro de los bloques MIME y que por tanto no se ve en la URL, es el método:**

1) GET

2) POST

3) HTML

4) http

**11) Es una buena práctica nombrar los campos, botones, combos y listas de HTML usando las mismas reglas de nombres de variables de PHP, pero sin el signo de pesos o dólar (\$):**

1) Verdadero

2) Falso

**12) Para crear una variable de tipo arreglo en PHP, cuando se recibe un valor múltiple de una formulario HTML, en el nombre del campo en HTML se deben incluir los signos:**

1) Comillas ""



- 2) Llaves {}
- 3) Paréntesis ()
- 4) Corchetes []

**13) El valor del parametro para hacer oculto un campo input en HTML es:**

- 1) occult
- 2) unset
- 3) hidden
- 4) novalue

**14) isset() tiene como funcion:**

- 1) Verificar que existe una variable
- 2) Verificar que el tipo de variable es el correcto
- 3) Eliminar una variable
- 4) Vaciar una variable

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 9: Manejo de fechas

**Objetivo:** El alumno aprenderá a validar fechas con PHP, crear nuevos objetos de fecha y cambiar la zona horaria, así como convertir una cadena a fecha.

#### Introducción:

PHP tienes muchas formas de crear, manipular y validar fechas ya que, por lo general, la forma de comunicarse el usuario es por medio de formularios y tenemos que garantizar que la información sea correcta antes de guardarla en la base de datos. Los temas que vamos a tratar en esta sección son:

- Uso de fechas con `time()` y `date()`
- La función `getdate()`
- La función `mktime()`
- Verificar una fecha con `checkdate()`
- Cambiar la zona horaria: `date_default_timezone_set`
- Convertir una cadena a fecha con `strtotime()`

## 9.1. Uso de fechas con time() y date()

- El manejo de hora y fecha es una herencia del Unix, y aunque el manejo interno es muy sencillo, hay que entender el formato "nativo".
- En PHP los datos de hora y fecha se representan exclusivamente como "segundos": los segundos transcurridos desde el 1 de enero de 1970 (fecha "oficial" de la era del Unix). Así es que cualquier hora o fecha es representada en "segundos".

a	"am" o "pm", en minúsculas	am
A	"AM" o "PM", en mayúsculas	PM
d	número de día del mes, con ceros antes cuando es menor a 10	02 , 25
D	día de la semana, abreviado a tres letras en inglés	Sat , Mon
F	nombre del mes	January , June
h	hora del día, formato de 12 horas, con ceros antes cuando es menor a 10	
H	hora del día, formato de 24 horas, con ceros antes cuando es menor a 10	
g	hora del día, formato de 12 horas, sin ceros antes cuando es menor a 10	
G	hora del día, formato de 24 horas, sin ceros antes cuando es menor a 10	
i	minutos	
j	número de día del mes, sin ceros antes cuando es menor a 10	
l	día de la semana	Sunday, Thursday
L	año bisiesto (0 para no, 1 para si)	1
m	número del mes del año, con ceros antes cuando es menor a 10	
M	mes del año, abreviado a tres letras	
n	número del mes del año, con ceros antes cuando es menor a 10	
s	segundos	
U	número de segundos desde 1 enero 1970	
y	año, dos dígitos	
Y	año, cuatro dígitos	
z	número de día del año, entre 0 y 365	
Z	desfase en segundos de la hora GMT (meridiano de Greenwich)	

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Fechas: time() y date()</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9   $t = time();
10  print "<p>".$t."</p>";
```

```
11 $fecha = date("d/m/Y",$t);
12 print "<p>".$fecha."</p>";
13 $hora = date("h:i:s",$t);
14 print "<p>".$hora."</p>";
15 //
16 $diaSemana = date("l",$t);
17 $diaMes = date("j",$t);
18 $mes = date("F",$t);
19 $anio = date("Y",$t);
20 $hora = date("H",$t);
21 $ampm = date("A",$t);
22 $min = date("i",$t);
23 print "Hola es ".$diaSemana." del mes ".$mes." a ".
24 $diaMes." del año ".$anio." y son las ".$hora." y ".
25 $min." minutos.";
26 ?>
27 </body>
28 </html>
```

**Listado 9.1.1.** PHP7-0901-FechasTimeDate.php

*www.pacoarce.com*

## 9.2. La función getdate()

- Existe una manera de obtener los datos más comunes de hora y fecha, mediante la función integrada "getdate".
- La sintaxis básica para este comando es:  
`$arreglo = getdate( segundos );`
- Si el parámetro "segundos" se omite, se usa el resultado de la función "time".
- Este comando regresa un arreglo asociativo con los siguientes datos:
  - seconds segundos del minuto, entre 0 y 59 23
  - minutes minutos de la hora, entre 0 y 60 15
  - hours horas del día, entre 0 y 23 22
  - mday día del mes, entre 1 y 31 10
  - wday día de la semana, entre 0 y 6 2
  - mon mes del año, entre 1 y 12 2
  - year año, cuatro dígitos 2001
  - yday día del año, entre 0 y 365 43
  - weekday nombre del día de la semana Tuesday
  - month nombre del mes del año February
  - 0 segundos 983671286

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Fechas Getdate()</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9 $meses = array("Enero","Febrero","Marzo","Abril",
10 "Mayo","Junio","Julio","Agosto","Septiembre","Octubre",
11 "Noviembre","Diciembre");
12 $dias[] = "Domingo";
13 $dias[] = "Lunes";
14 $dias[] = "Martes";
15 $dias[] = "Miércoles";
16 $dias[] = "Jueves";
17 $dias[] = "Viernes";
18 $dias[] = "Sábado";
19
20 $fecha_array = getdate();
21
22 $dia = $dias[$fecha_array["wday"]];
23 $mes = $meses[$fecha_array["mon"]-1];
24 print "hoy es ".$dia." ".$fecha_array["mday"]." de ".
25 $mes." de ".$fecha_array["year"].", y son las ".
26 $fecha_array["hours"]." con ".$fecha_array["minutes"].
27 " minutos.";
28
29
30 ?>
31 </body>
```

32 </html>

**Listado 9.2.1.** PHP7-0902-FechasGateDate.php



*www.pacoarce.com*

## 9.3. La función mktime()

- Cuando se desea saber qué hora y fecha es, todo va muy bien hasta aquí. Pero qué pasa cuando se desean hacer operaciones con fechas? El PHP solo entiende de segundos, y nada más. ¿Y que pasa si el usuario escribe una fecha, o una hora?
- El comando mktime es el primer paso para solucionar los problemas descritos, convirtiendo hora y fecha de formato "humano" a segundos.
- La sintaxis básica para este comando es:

`$segundos = mktime( hora, minuto, segundo, mes, dia, anio );`

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Fechas | mktime()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $cumple = mktime(0,0,0,8,27,1999);
10   $hoy = time();
11   $edad = $hoy-$cumple;
12
13   print "<p>Tu naciste un ".date("l",$cumple)."</p>";
14   print "<p>Haz vivido aproximadamente " . ($edad/60/60/24). " días</p>";
15   print "<p>Haz vivido aproximadamente " . ($edad/60/60/24/365). " años</p>";
16
17   ?>
18 </body>
19 </html>
```

**Listado 9.3.1.** PHP7-0903-FechasMktime.php

*www.pacoarce.com*

## 9.4. Verificar una fecha con checkdate()

- Comprueba la validez de una fecha formada por los argumentos. Una fecha se considera válida si cada parámetro está propiamente definido.

`bool checkdate ( int $mes , int $dia , int $anio );`

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Fecha Checkdate</title>
5 <meta charset="utf-8">
6 <?php
7     if (isset($_GET["fecha"])) {
8         $fecha = $_GET["fecha"];
9         $fecha = trim($fecha);
10        if ($fecha=="") {
11            print "La fecha es requerida";
12        } else {
13            //dd/mm/yyyy
14            $fecha_array = explode("/", $fecha);
15            //mm/dd/yyyy
16            if (checkdate($fecha_array[1], $fecha_array[0], $fecha_array[2])) {
17                print "Fecha correcta: ".$fecha;
18            } else {
19                print "Error en la fecha: ".$fecha;
20            }
21        }
22    }
23 ?>
24 </head>
25 <body>
26 <form action="PHP7-0904-FechasCheckdate.php" method="GET">
27     <input type="text" name="fecha" id="fecha" placeholder="DD/MM/YYYY"/>
28     <input type="submit" value="Enviar">
29 </form>
30 </body>
31 </html>
```

**Listado 9.4.1.** PHP7-0904-FechasCheckdate.php



## 9.5. Cambiar la zona horaria: date\_default\_timezone\_set

date\_default\_timezone\_set

Establece la zona horaria predeterminada usada por todas las funciones de fecha/hora en un script.

Establece la zona horaria predeterminada usada por todas las funciones de fecha/hora en un script.

ini\_set('date.timezone','America/Mexico\_City');

<http://php.net/manual/es/timezones.php>

phpinfo()

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Fechas | Timezone</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9   //print phpinfo();
10  print date("D, d M Y H:i:s")."<br>";
11  //date_default_timezone_set("America/Mexico_City");
12  ini_set("date.timezone", "America/Mexico_City");
13  print date("D, d M Y H:i:s");
14  ?>
15 </body>
16 </html>
```

**Listado 9.5.1.** PHP7-0905-FechasTimezone.php

*www.pacoarce.com*

## 9.6. Convertir una cadena a fecha con strtotime()

Esta función espera que se proporcione una cadena que contenga un formato de fecha en Inglés US e intentará convertir ese formato a una fecha Unix (el número de segundos desde el 1 de Enero del 1970 00:00:00 UTC), relativa a la marca de tiempo dada en “now”, o la marca de tiempo actual si “now” no se proporciona.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Fecha | strtotime()</title>
5  </head>
6  <body>
7    <?php
8      print strtotime("now")."<br>";
9      print strtotime("10 September 2015")."<br>";
10     print strtotime("+1 day")."<br>";
11     print strtotime("+1 week")."<br>";
12     print strtotime("+1 week 2 days 4 hours 2 seconds")."<br>";
13     print strtotime("next Thursday")."<br>";
14     print strtotime("last MMonday")."<br>";
15   ?>
16
17 </body>
18 </html>
```

**Listado 9.6.1.** PHP7-0906-FechasStrtotime.php

[www.pacoarce.com](http://www.pacoarce.com)

## Cuestionario:

### 1) La función "time()" regresa:

- 1) Los segundos transcurridos desde el 1 de enero de 1970.
- 2) La fecha y hora del servidor.
- 3) La fecha y hora de la computadora.
- 4) El número de días transcurridos desde el primero de enero de 1970.

### 2) Con el comando date() podemos representar una fecha de muchas formas, dependiendo de comandos en letras:

- 1) Verdadero
- 2) Falso

### 3) La función "date()" te puede traer el nombre de los días de la semana y los meses en español o cualquier otro idioma:

- 1) Verdadero
- 2) Falso

### 4) La función "mktime()" nos regresará:

- 1) El número de segundos transcurridos del 1 de enero de 1970 a la fecha señalada en los parámetros.
- 2) El número de segundos transcurridos del 1 de enero de 1970 a la fecha.
- 3) Un arreglo con la fecha actual.
- 4) Un arreglo con los meses, días y horas del año a partir del 1 de enero de 1970.

### 5) El comando "checkdate()" nos regresará:

- 1) Un valor booleano, indicando si la fecha es correcta o no.
- 2) Una cadena indicándonos si la fecha es correcta o no.
- 3) Un arreglo con las fechas válidas.
- 4) Una cadena con la fecha en formato mes - día - año.

[www.pacoarce.com](http://www.pacoarce.com)

# Todo PHP7

## De novato a experto

### Capítulo 10: Formateo de datos

**Objetivo:** El alumno aprenderá a formatear los datos, tanto las cadenas como los números, realizar una serialización de los mismos y crear filtros para la serialización con PHP7.

#### Introducción:

En este capítulo veremos las funciones para cambiar el formato a nuestros valores numéricos y cadenas, lo cual puede causar muchos dolores de cabeza cuando desarrollamos aplicaciones web. Los comandos que revisaremos son:

- Formato de un número con **printf()**
- Salida a una cadena con **sprintf()**
- Formato de un número con **number\_format()**
- Modificar la configuración local con **setlocale()**
- Serialización de objetos y arreglos
- Filtros para **unserialize()**

## 10.1. Formato de un número con printf()



*[www.pacoarce.com](http://www.pacoarce.com)*

# Todo PHP7

- Para "formatear" un valor numérico de modo que sea más fácil su lectura, se usa la función integrada "printf".
- La sintaxis básica para este comando es:

`printf ( control_del_formato, lista_de_numeros );`

- El primer parámetro es una "cadena" en el cual se indica el formato a dar al segundo parámetro, que puede ser desde un solo número (o variable numérica), hasta una lista de números separada por comas.
- El formato a aplicar son una combinación de caracteres, que comienzan siempre con el carácter "%", seguidos de algún carácter especial.
- Un especificador de signo opcional que fuerza a usar un signo (- o +) en un número. Por defecto, sólo el signo - se utiliza en un número si es negativo. Esta especificación fuerza números positivos para que también tengan adjunto el signo + (agregado en PHP 4.3.0).
- Un especificador de relleno opcional que indica qué carácter se utiliza para rellenar el resultado hasta el tamaño justo de la cadena. Este puede ser un carácter de espacio o un 0 (el carácter cero). El valor por defecto es rellenar con espacios. Un carácter de relleno alternativo se puede especificar prefijándolo con una comilla simple (').
- Un especificador de alineación opcional que indica si el resultado debe ser alineado a la izquierda o a la derecha. El valor por defecto es justificado a la derecha, un carácter - lo justificará a la izquierda.
- Un número opcional, un especificador de ancho que indica de cuántos caracteres (mínimo) resultará esta conversión.
- Un especificador de precisión opcional en la forma de un punto (.) seguido de un string opcional de dígitos decimales que indica cuántos dígitos decimales deben mostrarse para los números de punto flotante.
- Cuando se utiliza este especificador con una cadena, actúa como un punto de corte, estableciendo un límite máximo de caracteres a la cadena. Además, el carácter para empleado cuando se rellena un número podría especificarse opcionalmente entre el punto y el dígito.
- Un especificador de tipo que indica con qué tipo deben ser tratados los datos del argumento. Los tipos posibles son:

% - un carácter de porcentaje literal. No se requiere argumento.

b - el argumento es tratado como un valor de tipo integer y presentado como un número binario.

c - el argumento es tratado como un valor de tipo integer y presentado como el carácter con ese valor ASCII.

d - el argumento es tratado como un valor de tipo integer y presentado como un número decimal (con signo).

e - el argumento es tratado con notación científica (e.g. 1.2e+2). El especificador de precisión indica el número de dígitos después del punto decimal a partir de PHP 5.2.1. En versiones anteriores, se tomó como el número de dígitos significativos (menos uno).

E - como %e pero utiliza la letra mayúscula (e.g. 1.2E+2).

f - el argumento es tratado como un valor de tipo float y presentado como un número de punto flotante (considerando la configuración regional).

F - el argumento es tratado como un valor de tipo float y presentado como un número de punto flotante (no considerando la configuración regional). Disponible desde PHP 4.3.10 y PHP 5.0.3.

g - lo mismo que %e y %f.

G - lo mismo que %E y %f.

o - el argumento es tratado como un valor de tipo integer y presentado como un número octal.

s - el argumento es tratado y presentado como un string.

u - el argumento es tratado como un valor de tipo integer y presentado como un número decimal sin signo.

```
1  <?php
2  $monto = 1500;
3  $tasaIVA = 0.16;
4  $montoIVA = $monto * $tasaIVA;
5  $granTotal = $monto + $montoIVA;
6
7  $retISR = 0.1;
8  $retIVA = 0.166667;
9
10 $montoRetIVA = $granTotal * $retIVA;
11 $montoRetISR = $granTotal * $retISR;
12
13 $total = $granTotal - $montoRetISR - $montoRetIVA;
14
15 printf("<p>Monto: $%6.2f</p>", $monto);
16 printf("<p>IVA : $%6.2f</p>", $montoIVA);
17 printf("<p>Gran total: $%6.2f</p>", $granTotal);
18 printf("<p>-----</p>");
19 printf("<p>Retención IVA: $%6.2f</p>", $montoRetIVA);
20 printf("<p>Retención ISR: $%6.2f</p>", $montoRetISR);
21 printf("<p>-----</p>");
22 printf("<p>Total: $%6.2f</p>", $total);
23 ?>
```

**Listado 10.1.1.** PHP7-1001-FormatoPrintf.php

[www.pacoarce.com](http://www.pacoarce.com)

## 10.2. Salida a una cadena con sprintf()

- Esta variante del comando "printf" no manda el resultado al navegador del usuario, sino que crea un "string" para ser guardado y usado más adelante.
- La sintaxis básica para este comando es:

`$cadena = sprintf ( control_del_formato, lista_de_numeros );`

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formatos | sprintf()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $monto = 1500;
10   $tasaIVA = 0.16;
11   $montoIVA = $monto * $tasaIVA;
12   $granTotal = $monto + $montoIVA;
13
14   $retISR = 0.1;
15   $retIVA = 0.166667;
16
17   $montoRetIVA = $granTotal * $retIVA;
18   $montoRetISR = $granTotal * $retISR;
19
20   $total = $granTotal - $montoRetISR - $montoRetIVA;
21
22   $sTotal = sprintf("<p>Monto: $%6.2f, IVA : $%6.2f,
23   Gran total: $%6.2f</p>", $monto, $montoIVA, $granTotal);
24   $sRet = sprintf("<p>Retención IVA: $%6.2f,
25   Retención ISR: $%6.2f</p>", $montoRetIVA, $montoRetISR);
26   $sTotalFinal = sprintf("<p>Total: $%6.2f</p>", $total);
27   print $sTotal;
28   print "<p>-----</p>";
29   print $sRet;
30   print "<p>-----</p>" ;
31   print $sTotalFinal;
32   ?>
33 </body>
34 </html>
```

**Listado 10.2.1.** PHP7-1002-FormatoSprintf.php



## 10.3. Formato de un número con number\_format()

- Este comando es muy usado para "formatear" valores numéricos de una manera más sencilla, y muy orientado a mostrar cantidades en formato de "moneda". Solo incluye comas para separar miles, millones, etc, y el número de decimales indicado.

```
string number_format ( float $number , int $decimals = 0 , string $dec_point = "." , string $thousands_sep = "," )
```

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formatos | sprintf()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $monto = 1500;
10     $tasaIVA = 0.16;
11     $montoIVA = $monto * $tasaIVA;
12     $granTotal = $monto + $montoIVA;
13
14     $retISR = 0.1;
15     $retIVA = 0.166667;
16
17     $montoRetIVA = $granTotal * $retIVA;
18     $montoRetISR = $granTotal * $retISR;
19
20     $total = $granTotal - $montoRetISR - $montoRetIVA;
21
22     $sMonto = number_format($monto,2);
23     $sMontoIVA = number_format($montoIVA,2);
24     $sMontoTotal = number_format($granTotal,2);
25     $sMontoRetIVA = number_format($montoRetIVA,2);
26     $sMontoRetISR = number_format($montoRetISR,2);
27     $sMontoTotalFinal = number_format($total,2);
28     print "<p>Monto: $".$sMonto."</p>";
29     print "<p>IVA: $".$sMontoIVA."</p>";
30     print "<p>Total parcial: $".$sMontoTotal."</p>";
31     print "<hr>";
32     print "<p>Retención IVA: $".$sMontoRetIVA."</p>";
33     print "<p>Retención ISR: $".$sMontoRetISR."</p>";
34     print "<hr>";
35     print "<p>Total: $".$sMontoTotalFinal."</p>";
36     $totalFrances = number_format($total,2," "," ");
37     $totalEspania = number_format($total,2," ",".");
38     print "<p>Total Francés: $".$totalFrances."</p>";
39     print "<p>Total Europeo: $".$totalEspania."</p>";
40     ?>
```

```
41 </body>
42 </html>
```

**Listado 10.3.1.** PHP7-1003-FormatoNumberFormat.php



*www.pacoarce.com*

## 10.4. Modificar la configuración local con setlocale()

- La función setlocale() modifica la configuración local, como el formato de la fecha, formato de la hora, la moneda, etc.
- Tiene dos sintaxis:

string setlocale ( int \$category , string \$locale [, string \$... ] )

string setlocale ( int \$category , array \$locale )

- category es una constante con nombre que especifica la categoría de las funciones afectadas por la configuración local:

LC\_ALL para establecer todas las siguientes

LC\_COLLATE para la comparación de cadenas; véase strcoll()

LC\_CTYPE para la clasificación y conversión de caracteres, por ejemplo strtoupper()

LC\_MONETARY para localeconv()

LC\_NUMERIC para el separador decimal (véase también localeconv())

LC\_TIME para el formato de fecha y hora con strftime()

LC\_MESSAGES para las respuestas del sistema (disponible si PHP fue compilado con libintl)

- Si locale es NULL o el string vacío "", los nombres de los localismos se establecerán a partir de los valores de las variables de entorno con los mismos nombres que las categorías anteriores, o desde "LANG".
- Si locale es "0", el localismo no se ve afectado, solamente devuelve la configuración actual.
- Si locale es un array o le siguen parámetros adicionales, cada elemento del array o parámetro se intenta establecer como nuevo localismo hasta que se tenga éxito. Esto es útil si un localismo se conoce con diferentes nombres en diferentes sistemas, o para proporcionar un respaldo para un localismo posiblemente no disponible.

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formato setlocale()</title>
5 </head>
6 <body>
7   <?php
8     $loc_de = setlocale(LC_ALL, 'de_DE@euro', 'de_DE', 'deu_deu');
9     print "La configuración local en alemán es ".$loc_de;
10  ?>
11 </body>
12 </html>
```

**Listado 10.4.1.** PHP7-1004-FormatoSetlocale.php

## 10.5. Serialización de objetos y arreglos

- Genera una representación apta para el almacenamiento de un valor

`string serialize ( mixed $value )`

- Esto es útil para el almacenamiento de valores en PHP sin perder su tipo y estructura.
- Para recuperar el valor PHP a partir de la cadena seriada, utilice `unserialize()`.
- El valor a ser serializado. `serialize()` maneja todos los tipos.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formatos serialize()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8  <?php
9    $arreglo = array(
10     "fruta"=>"manzana",
11     "equipo"=>"MacPro",
12     "animales" => array(
13       "perro",
14       "gato",
15       "ratón"
16     );
17   $cadena = serialize($arreglo);
18   print $cadena."<br>";
19   $arreglo2 = unserialize($cadena);
20   print var_dump($arreglo2)."<br>";
21   print $arreglo2["animales"][1];
22 ?>
23 </body>
24 </html>
```

**Listado 10.5.1.** PHP7-1005-FormatoSerialize.php

[www.pacoarce.com](http://www.pacoarce.com)

## 10.6. Filtros para unserialize()

Por medio de los filtros crearemos “listas blancas” de las clases que si podemos deserializar de datos de entrada no seguros. Todas las demás clases serán bloqueadas para evitar la inyección de código.

Los filtros son exclusivos de PHP7.x.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Formato | filtros unserialize (PHP7)</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    class Objeto1{
10     public $valor1;
11   }
12   class Objeto2{
13     public $valor2;
14   }
15   $obj1 = new Objeto1();
16   $obj1->valor1 = "gato";
17
18   $obj2 = new Objeto2();
19   $obj2->valor2 = "perro";
20
21   $obj1Serializado = serialize($obj1);
22   $obj2Serializado = serialize($obj2);
23
24   //deserializarlo
25   // "allowed_classes"=>true acepta todas las clases
26   $data = unserialize($obj1Serializado,["allowed_classes"=>true]);
27   var_dump($data);
28   print "<hr>";
29   // "allowed_classes"=>false no acepta ninguna las clases
30   $data = unserialize($obj1Serializado,["allowed_classes"=>false]);
31   var_dump($data);
32   print "<hr>";
33   // "allowed_classes"=>[] (lista blanca)
34   $data = unserialize($obj1Serializado,["allowed_classes"=>["Objeto2","Objeto1"]]);
35   var_dump($data);
36   print "<hr>";
37   // "allowed_classes"=>[] (lista blanca)
38   $data = unserialize($obj2Serializado,["allowed_classes"=>["Objeto2"]]);
39   var_dump($data);
40
41  ?>
```

```
42 </body>  
43 </html>
```

**Listado 10.6.1.** filtrosUnserialize.php



*www.pacoarce.com*

## Cuestionario:

### 1) El comando "printf()" sirve para:

- 1) Formatear los números en una cadena.
- 2) Concatenar las cadenas con los arreglos
- 3) Mandar una cadena a impresora.
- 4) Mandar una cadena a un archivo.

### 2) El comando "sprintf()" sirve para:

- 1) Formatea un conjunto de número en una cadena y la manda a una variable.
- 2) Formatea un conjunto de número en una cadena y la manda a una impresora.
- 3) Formatea un conjunto de número en una cadena y la manda a la pantalla.
- 4) Formatea un conjunto de número en una cadena y la manda a un archivo.

### 3) El comando "printf()" cuenta con cuatro parámetros:

- 1) Verdadero
- 2) Falso

### 4) El comando que formatea con ceros y como un número es:

- 1) number\_format
- 2) printf
- 3) scanf
- 4) sprintf

### 5) Con el comando "printf()" el formato a aplicar son una combinación de caracteres, que comienzan siempre con el caracter:

- 1) #
- 2) &
- 3) %
- 4) @

### 6) Con el comando "printf()" el argumento que es tratado como un valor de tipo integer y presentado como un número decimal (con signo) es:

- 1) a
- 2) i
- 3) u
- 4) d

**7) La función que modifica la configuración local, como el formato de la fecha, formato de la hora, la moneda, etc. es:**

- 1) local()
- 2) changeLocale()
- 3) config()
- 4) setlocale()

**8) La función que genera una representación apta para el almacenamiento de un valor es:**

- 1) setlocale()
- 2) serialize()
- 3) printf()
- 4) echo

**9) Para recuperar el valor PHP a partir de la cadena seriada se utiliza:**

- 1) local()
- 2) serialize()
- 3) printf()
- 4) unserialize()

**10) Los filtros para unserialize() son exclusivos a partir de la versión:**

- 1) 4.0.
- 2) 5.0.
- 3) 6.0.
- 4) 7.0.

*www.pacoarce.com*



# Todo PHP7

## De novato a experto

### Capítulo 11: Manejo de cadenas y funciones matemáticas

**Objetivo:** El alumno aprenderá a modificar, cortar y sanear cadenas con PHP, así como aplicar las funciones matemáticas.

#### Introducción:

En esta sección estudiaremos dos grupos de funciones muy importantes en PHP: Las funciones que modifican las cadenas y las funciones matemáticas:

- Longitud de una cadena con `strlen()`
- Buscar una subcadena en una cadena: el comando `strstr()`
- Buscar una subcadena en una cadena: el comando `strpos()`
- Extraer una subcadena: el comando `substr()`
- Pasar una cadena a mayúsculas minúsculas: `strtolower()` y `strtoupper()`
- Reemplazar partes de una cadena: `str_replace()`
- Sanear una cadena con PHP, la función `str_replace()`
- Crear una cadena aleatoria con `str_shuffle()`
- Convertir caracteres especiales de texto a HTML
- Ajusta la longitud de una cadena: `wordwrap()`
- Dividir una cadena en diferentes cadenas: `explode()`
- Cambiar la primera letra a mayúsculas
- Verificar si es un número con `is_number`
- Redondear valores
- Funciones trigonométricas
- La función `range()`
- Generar números aleatorios
- Exponentes y raíz cuadrada
- Cambio de bases
- Recuperar la parte entera y la parte decimal de una división
- Funciones de CSPRNG

[www.pacoarce.com](http://www.pacoarce.com)

## 11.1. Longitud de una cadena con strlen()

- Existen muchas operaciones que aplican a cadenas o "strings". Aquí solo se mencionan las más usadas.
- Obtener la longitud de una cadena, con el comando "strlen". Ya se ha visto su uso en las validaciones.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Manejo de cadenas | strlen()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $cadena1 = "En el agua clara, que brota en la fuente, un lindo pescado sale de
10     repente";
11     $cadena2 = "";
12     for ($i=0; $i < strlen($cadena1); $i++) {
13       $cadena2 = $cadena1[$i].$cadena2;
14     }
15     print "<p>".$cadena1."</p>";
16     print "<p>".$cadena2."</p>";
17     print "<p>La longitu de la cadena es de ".strlen($cadena1)." caracateres</p>";
18   ?>
19 </body>
20 </html>
```

**Listado 11.1.1.** PHP7-1101-CadenaStrlen.php

*www.pacoarce.com*

## 11.2. Buscar una subcadena en una cadena: el comando strstr()

- Buscar una subcadena en una cadena: el comando strstr
- -Buscar y extraer un "sub-string" dentro de otro string, con el comando "strstr".
- Primero busca el "sub-string" dentro del "string".
- Si no lo encuentra, regresa "false".
- Si lo encuentra, regresa el "sub-string" encontrado.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Manejo de cadenas | strstr()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8  <?php
9    $cadena1 = "En el agua clara, que brota en la fuente, un lindo pescado sale de
    repente";
10   $cadena2 = "pescado";
11   if(strstr($cadena1,$cadena2)){
12     print "Si se encontró la subcadena '". $cadena2.'" en la cadena '". $cadena1.'"';
13   } else {
14     print "No se encontró la subcadena '". $cadena2.'" en la cadena '". $cadena1.'"';
15   }
16   ?>
17 </body>
18 </html>
```

**Listado 11.2.1.** PHP7-1102-CadenaStrstr.php

*www.pacoarce.com*

## 11.3. Buscar una subcadena en una cadena: el comando strpos()

- Buscar una **subcadena** en una cadena: el comando strpos
- Buscar una "**subcadena**" dentro de otra cadena y reportar la posición donde comienza, con la función "**strpos()**".
- Primero busca la "**subcadena**" dentro de la "cadena".
- Si no lo encuentra, regresa "**false**".
- Si lo encuentra, regresa un entero indicando la posición donde comienza.

mixed strpos ( string \$haystack , mixed \$needle [, int \$offset = 0 ] )

- Buscar una **subcadena** en una cadena: el comando **strpos**
- Devuelve la posición donde la **subcadena** (aguja) existe, en relación al inicio de la cadena (pajar) (independiente del desplazamiento u *offset*). También tener en cuenta que las posiciones de inicio de las cadenas empiezan en 0 y no 1.
- Devuelve **false** si no fue encontrada la aguja.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Manejo de cadenas | strpos()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    //
10   //Iniciamos las variables
11   //
12   $cadena1 = "En el agua clara, que brota en la fuente, un lindo pescado sale de
13   repente";
14   $cadena2 = "la";
15   $pos = 0;
16   $pos_array = array();
17   //
18   //Buscamos las coincidencias de la subcadena
19   //
20   while ($pos = strpos($cadena1, $cadena2, $pos)) {
21     array_push($pos_array, $pos);
22     $pos++;
23   }
24   //Imprimimos las ocurrencias de la cadena
25   //
26   print "<p>El número de ocurrencias de la cadena '". $cadena2;
27   print "' en la '". $cadena1. "' es de ".count($pos_array). " veces.</p>";
28   //
29   //Desplegamos las posiciones donde se encontró la coincidencia
30   //
31   foreach ($pos_array as $key => $value) {
32     print "<p>Posición ".$key." => ".$value."</p>";
```

```
33 }  
34 ?>  
35 </body>  
36 </html>
```

**Listado 11.3.1.** PHP7-1103-CadenaStrpos.php



*www.pacoarce.com*

## 11.4. Extraer una subcadena: el comando substr()

- Se especifica la cadena de donde extraer, el caracter donde se va a comenzar a extraer y opcionalmente la longitud o número de caracteres a extraer.
- Si no se especifica la cantidad de caracteres a extraer, se extrae hasta el final de la cadena. (PHP 4, PHP 5, PHP 7)

`string substr ( string $string , int $start [, int $length ] )`

- Si "start" es negativo, la cadena devuelta empezará en "start" contando desde el final de la cadena.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Cadenas | substr</title>
5  </head>
6  <body>
7    <?php
8      $mail = "estesmiemail@hotmail.com";
9      $arroba = strpos($mail, "@");
10     $arroba++;
11     $punto = strpos($mail, ".", $arroba);
12     $lon = $punto - $arroba;
13     $servicio = substr($mail, $arroba, $lon);
14     $com = substr($mail, -3);
15     print "<p>La arroba se encuentra en: ".$arroba."</p>";
16     print "<p>El punto se encuentra en: ".$punto."</p>";
17     print "<p>Las longitud es: ".$lon."</p>";
18     print "<p>El servicio es: ".$servicio."</p>";
19     print "<p>La extensión del correo es: ".$com."</p>";
20
21
22  ?>
23 </body>
24 </html>
```

**Listado 11.4.1.** PHP7-1104-cadenasSubstr.php

[www.pacoarce.com](http://www.pacoarce.com)

## 11.5. Pasar una cadena a mayúsculas minúsculas: strtolower() y strtoupper()

- La función **strtolower** convierte una cadena a minúsculas, tomando como “alfabeto” la configuración local del servidor.
- También podemos utilizar **mb strtolower** que convierte una cadena de caracteres a minúsculas
- La función **strtoupper** convierte una cadena a mayúsculas, tomando como “alfabeto” la configuración local del servidor.
- También podemos utilizar **mb strtoupper** que convierte una cadena de caracteres a mayúsculas.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Cadenas | Mayúsculas y minúsculas</title>
5  </head>
6  <body>
7  <?php
8    $cadena = "En el águá clÄra, quÉ brota en la fuente...";
9    $mayusculas = strtoupper($cadena);
10   $minusculas = strtolower($cadena);
11   $mb_mayusculas = mb_strtoupper($cadena,"UTF-8");
12   $mb_minusculas = mb_strtolower($cadena,"UTF-8");
13   print "<p>".$cadena."</p>";
14   print "<p>".$mayusculas."</p>";
15   print "<p>".$minusculas."</p>";
16   print "<p>".$mb_mayusculas."</p>";
17   print "<p>".$mb_minusculas."</p>";
18
19  ?>
20 </body>
21 </html>
```

**Listado 11.5.1.** PHP7-1105-CadenasMayusculasMinusculas.php

www.pacoarce.com

## 11.6. Reemplazar partes de una cadena: str\_replace()

- Reemplazar partes de una cadena: **str\_replace()**
- La sintaxis básica para este comando es:

`mixed str_replace ( mixed $search , mixed $replace , mixed $subject [, int &$count ] );`

- Esta función devuelve una cadena o un arreglo con todas las apariciones de **search** en **subject** reemplazadas con el valor dado de **replace**.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Cadenas | str_replace</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $salario = "$56,500.07";
10     $cadena = str_replace("$", "", $salario);
11     $cadena = str_replace(",", "", $cadena);
12     $nuevoSalario = (double) $cadena;
13     $nuevoSalario *= 1.12;
14     $nuevoSalarioCadena = number_format($nuevoSalario,2);
15
16     print "<p>Tu salario de ".$salario." más tu bono anual del %12 da un total de
17     $".$nuevoSalarioCadena.", felicidades!</p>";
18   ?>
19 </body>
20 </html>
```

**Listado 11.6.1.** PHP7-1106-CadenasStrreplace.php

[www.pacoarce.com](http://www.pacoarce.com)



## 11.7. Sanear una cadena con PHP, la función str\_replace()

Por medio de la sentencia **str\_replace** podemos evitar que el usuario introduzca caracteres que nos podrían dañar nuestra información, sobre todo en la base de datos.

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="UTF-8">
5  <title>Sanear cadena</title>
6  <?php
7  function sanear($string){
8      $string = trim($string);
9      //
10     $string = str_replace(
11     array('à', 'â', 'â', 'ä', 'Á', 'À', 'Â', 'Ä'),
12     array('a', 'a', 'a', 'a', 'A', 'A', 'A', 'A'),
13     $string
14     );
15     //
16     $string = str_replace(
17     array('è', 'é', 'ê', 'É', 'È', 'Ê', 'Ë'),
18     array('e', 'e', 'e', 'E', 'E', 'E', 'E'),
19     $string
20     );
21
22     $string = str_replace(
23     array('ì', 'í', 'î', 'Í', 'Ì', 'Î', 'Ï'),
24     array('i', 'i', 'i', 'I', 'I', 'I', 'I'),
25     $string
26     );
27
28     $string = str_replace(
29     array('ò', 'ó', 'ô', 'Ó', 'Ò', 'Ö', 'Ô'),
30     array('o', 'o', 'o', 'O', 'O', 'O', 'O'),
31     $string
32     );
33
34     $string = str_replace(
35     array('ù', 'ü', 'û', 'Ú', 'Ù', 'Û', 'Ü'),
36     array('u', 'u', 'u', 'U', 'U', 'U', 'U'),
37     $string
38     );
39     $string = str_replace(
40     array('ç', 'Ç'),
41     array('c', 'C'),
42     $string
43     );
44     $string = str_replace(
```

```
45 array("\\", "\"", "°", "-", "~",
46 "#", "@", "|", "!", "\\\"",
47 ".", "$", "%", "&", "/",
48 "(", ")", "?", "'", "i",
49 "¿", "[", "^", "`", "]",
50 "+", "}", "{", "``", "´",
51 ">", "< ", ";", ", ", ":",
52 "."),
53 ',
54 $string
55 );
56 return $string;
57 }
58 $cadena =
59 "áàãäåäÀÃÄÅdoéèëêÊËËËEreíîïîíîïîmióðöôöÔöfaúûüüÜÜÜÜsolñÑçÇlasi\\\"°-~#@|!\\\",·$%&/()?)¿¡¿[^^]
60 ;,:. ";
61 print $cadena;
62 print "<hr>";
63 print sanear($cadena);
64 ?>
65 </head>
66 <body>
67 </body>
68 </html>
```

**Listado 11.7.1.** PHP7-1107-CadenasSanearcadenas.php

[www.pacoarce.com](http://www.pacoarce.com)

## 11.8. Crear una cadena aleatoria con `str_shuffle()`

Con la función **`str_shuffle()`** podemos hacer una nueva cadena de otra, desordenando sus caracteres, sean numéricos, símbolos o letras.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Cadenas | str_shuffle</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $cadena = "23456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ-_" ;
10     $clave = substr(str_shuffle($cadena),0,12);
11     print "<p>Tu nueva clave de usuario es: ".$clave."</p>";
12
13
14   ?>
15
16 </body>
17 </html>
```

**Listado 11.8.1.** PHP7-1108-CadenasStrShuffle.php

[www.pacoarce.com](http://www.pacoarce.com)

## 11.9. Convertir caracteres especiales de texto a HTML

- Para lograr esto de una manera rápida y sencilla se usa el comando "get\_html\_translation\_table()".
- La sintaxis básica para este comando es:

```
$tabla = get_html_translation_table( HTML_ENTITIES );
```

- Convertir caracteres especiales de texto a HTML

**strtr** — Convierte caracteres o reemplaza substrings

string strtr ( string \$str , string \$from , string \$to )

string strtr ( string \$str , array \$replace\_pairs )

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Cadenas | strtr</title>
5  </head>
6  <body>
7    <?php
8      $tabla = get_html_translation_table(HTML_ENTITIES);
9      $cadena = "La etiqueta que efectúa un salto de línea en HTML es: <br><hr>";
10     print $cadena;
11     print "<br>";
12     print strtr($cadena, $tabla);
13   ?>
14 </body>
15 </html>
```

**Listado 11.9.1.** PHP7-1109-CadenasStrtr.php

*www.pacoarce.com*

## 11.10. Ajusta la longitud de una cadena: wordwrap()

- La función wordwrap() ajusta una cadena hasta un número dado de caracteres

string **wordwrap** ( string \$str [, int \$width = 75 [, string \$break = "\n" [, bool \$cut = false ]]] )

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Cadena | wordwrap</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8  <?php
9    $cadena = "En el agua clara, que brota en la fuente, un lindo pescado, sale de
repente";
10 $salida = wordwrap($cadena,70,"\r\n",true);
11 print $salida;
12
13
14 ?>
15 </body>
16 </html>
```

**Listado 11.10.1.** PHP7-1110-Wordwrap.php

[www.pacoarce.com](http://www.pacoarce.com)

## 11.11. Dividir una cadena en diferentes cadenas: explode()

- La función explode() divide una cadena en diferentes cadenas por medio de un “separador” y los regresa como un arreglo.

explode( string \$separador, string \$cadena [, int \$limite = PHP\_INT\_MAX]);

- Si \$limite es positivo, regresará el número de elementos señalados en él.
- Si \$limite es negativo, regresará todos los elementos indicados, menos el -\$limite (de atrás a adelante).
- Si \$limite es cero, se considerará como 1.

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Cadenas | explode</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9 $frutas = "manzana pera guayaba piña sandía";
10 $frutas_array = explode(" ", $frutas);
11 foreach ($frutas_array as $key => $value) {
12   print $value."<br>";
13 }
14 //asignamos valores
15 list($manzana, $pera, $guayaba, $pinia, $sandia) = explode(" ", $frutas);
16 print "La fruta con más vitamina C es la ".$guayaba."<br>";
17 print "<hr>El uso de los limites<br>";
18 $frutas_array = explode(" ", $frutas,2);
19 foreach ($frutas_array as $key => $value) {
20   print $value."<br>";
21 }
22 print "<hr>El uso de los limites negativos<br>";
23 $frutas_array = explode(" ", $frutas,-2);
24 foreach ($frutas_array as $key => $value) {
25   print $value."<br>";
26 }
27 ?>
28 </body>
29 </html>
```

www.pacoarce.com

**Listado 11.11.1.** PHP7-1111-CadenasExplode.php

## 11.12. Limpiar cadenas: trim(), ltrim() y rtrim()

Versión (PHP 4, PHP 5, PHP 7)

- trim — Elimina los espacios en blanco (u otro tipo de caracteres) al inicial y al final de la cadena.
- rtrim - Elimina los espacios en blanco a la derecha de la cadena.
- ltrim - Elimina los espacios en blanco a la izquierda de la cadena.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Limpiar cadenas: trim(), rtrim() y ltrim()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $texto = "\t\tHola, cara de bola\t\t";
10     $binaria = "\x09Hola, cara de bola\x0A";
11     $saludo = "www.google.com/";
12     var_dump($texto, $binaria, $saludo);
13     print "<br>";
14
15     $cadena = rtrim($texto);
16     var_dump($cadena);
17     print "<br>";
18
19     $cadena = trim($texto);
20     var_dump($cadena);
21     print "<br>";
22
23     $cadena = ltrim($binaria, "\x00..\x1F");
24     var_dump($cadena);
25     print "<br>";
26
27     $cadena = rtrim($saludo, "/");
28     var_dump($cadena);
29     print "<br>";
30
31   ?>
32 </body>
33 </html>
```

**Listado 11.12.1.** trim.php

## 11.13. Cambiar la primera letra a mayúsculas

(PHP 4, PHP 5, PHP 7)

ucwords — Convierte a mayúsculas el primer caracter de cada palabra de una cadena

ucfirst() - Convierte el primer caracter de una cadena a mayúsculas.

string ucwords ( string \$str [, string \$delimiters = " \t\r\n\f\v" ] )

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Cambiar la primer letra a mayúcula: ucwords(), ucfirst()</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <?php
9     $nombre = "juan peréz lópez";
10    print ucwords($nombre)."<br>";
11    print ucfirst($nombre)."<br>";
12    $saludo = "HOLA MUNDO";
13    print $saludo."<br>";
14    $saludo = strtolower($saludo);
15    print $saludo."<br>";
16    $saludo = ucfirst($saludo);
17    print $saludo."<br>";
18  ?>
19 </body>
20 </html>
```

**Listado 11.13.1.** ucwords.php

*www.pacoarce.com*



## 11.50. Verificar si es un número con is\_number

Cambiar la primera letra a mayúscula

is\_numeric: verifica si es un número.

is\_finite: es un número finito.

is\_infinite: es un valor infinito.

Is\_nan: no es un número.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones Matemáticas | is_numeric</title>
5  </head>
6  <body>
7  <?php
8    print is_numeric(5)?"Verdadero":"Falso";
9    print "<br>";
10   print is_numeric("cinco")?"Verdadero":"Falso";
11   print "<br>";
12   print is_numeric("5")?"Verdadero":"Falso";
13   print "<br>";
14   print is_numeric("$5.00")?"Verdadero":"Falso";
15   print "<hr>";
16   print is_nan(acos(8))?"No es un número (NaN)": "Si es un número";
17   print "<br>";
18   print is_nan(acos(1))?"No es un número (NaN)": "Si es un número";
19   print "<hr>";
20   print is_finite(log(0))?"Es un número finito": "No es un número finito";
21   print "<br>";
22   print is_finite(1)"Es un número finito": "No es un número finito";
23   print "<hr>";
24   print is_infinite(10/0)"Es un número infinito": "No es un número infinito";
25   print "<br>";
26   print is_infinite(10/1)"Es un número infinito": "No es un número infinito";
27   print "<hr>";
28  ?>
29 </body>
30 </html>
```

**Listado 11.50.1.** funcionesMatematicasIsNumeric.php

## 11.51. Redondear valores

abs: regresa el valor absoluto.

ceil: regresa el valor decimal hacia arriba.

round: redondea de .5 hacia arriba, menor a .5, trunca los decimales.

floor: trunca los decimales.

max: valor máximo de una lista de valores o de un arreglo.

min: valor mínimo de una lista de valores o de un arreglo.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones Matemáticas | abs ceil floor</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      print "<h2>Valor absoluto abs()</h2>";
10     print abs(-5);
11     print "<br>";
12     print abs(5);
13     print "<br>";
14     print abs(-5.6);
15     print "<br>";
16     print abs(5.5);
17     print "<hr>";
18     print "<h2>Redondea hacia arriba ceil()</h2>";
19     print ceil(-5);
20     print "<br>";
21     print ceil(5);
22     print "<br>";
23     print ceil(-5.6);
24     print "<br>";
25     print ceil(5.1);
26     print "<hr>";
27     print "<h2>Redondea hacia 'abajo' floor()</h2>";
28     print floor(-5);
29     print "<br>";
30     print floor(5);
31     print "<br>";
32     print floor(-5.6);
33     print "<br>";
34     print floor(5.1);
35     print "<hr>";
36     print "<h2>Redondea round()</h2>";
37     print round(-5);
38     print "<br>";
39     print round(5);
```

```
40 print "<br>";
41 print round(-5.6);
42 print "<br>";
43 print round(5.1);
44 print "<br>";
45 print round(5.6);
46 print "<hr>";
47 print "<h2>Valor máximo max()</h2>";
48 print max(5,2,8,6,7,9);
49 print "<br>";
50 print max(array(8,3,5,1,8,6,9));
51 print "<hr>";
52 print "<h2>Valor mínimo min()</h2>";
53 print min(5,2,8,6,7,9);
54 print "<br>";
55 print min(array(8,3,5,1,8,6,9));
56 print "<hr>";
57 ?>
58 </body>
59 </html>
```

**Listado 11.51.1.** funcionesMatematicasAbsCeilRound.php

*www.pacoarce.com*

## 11.52. Funciones trigonométricas

cos: coseno

sin: seno

tan: tangente

acos: arco coseno

acosh: arco coseno hiperbólico

deg2rad: convierte los grados a radianes

atan2: Arco tangente de dos variables

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones matemáticas | Trigonometría</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      print cos(2.11122)."<br>";
10     print atan(1.22)."<br>";
11     print atanh(0.7112)."<br><hr>";
12     print 1/sin(0.701)."<br>";
13     print 1/cos(0.701)."<br>";
14     print 1/tan(0.701)."<br><hr>";
15     print tan(deg2rad(60))."<br>";
16   ?>
17 </body>
18 </html>
```

**Listado 11.52.1.** funcionesMatematicastrigonometria.php

*www.pacoarce.com*

## 11.53. La función range()

La función range()

Por medio de la función range() podemos crear un arreglo con un rango de números enteros o de letras. Su sintaxis es:

array range ( mixed \$start , mixed \$end [, number \$step = 1 ] )

Podemos variar el valor \$step para variar el incremento del intervalo.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones numéricas | range()</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    foreach (range(0,30) as $key => $value) {
10     print $value.", ";
11   }
12   print "<hr>";
13   foreach (range(30,0) as $key => $value) {
14     print $value.", ";
15   }
16   print "<hr>";
17   foreach (range(30,0,5) as $key => $value) {
18     print $value.", ";
19   }
20   print "<hr>";
21   foreach (range("a","z") as $key => $value) {
22     print $value.", ";
23   }
24   print "<hr>";
25   foreach (range("z","a") as $key => $value) {
26     print $value.", ";
27   }
28   print "<hr>";
29   $numeros = range(0,100,5);
30   shuffle($numeros);
31   foreach ($numeros as $key => $value) {
32     print $value.", ";
33   }
34   print "<hr>";
35   ?>
36 </body>
37 </html>
```

**Listado 11.53.1.** range.php

## 11.54. Generar números aleatorios

Con la función `rand()` podemos crear un número semi-aleatorio entero. Su sintaxis es:

*int rand ( void )*

*int rand ( int \$min , int \$max )*

Generar número aleatorios

Otras funciones que pueden generar números aleatorios son:

**`srand()`** - Genera un número aleatorio a partir de una semilla

**`getrandmax()`** - Mostrar el mayor valor aleatorio posible

**`mt_rand()`** - Genera un mejor número entero aleatorio

**`random_int()`** - Genera números enteros pseudoaleatorios criptográficamente seguros

**`random_bytes()`** - Genera bytes pseudoaleatorios criptográficamente seguros

**`openssl_random_pseudo_bytes()`** - Genera una cadena de bytes pseudo-aleatoria

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones matematicas | aleatorios</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      print rand()."<br>";
10     print rand()."<br>";
11     print rand(200,650)."<br>";
12     print getrandmax()."<hr>";
13   //
14   list($param1, $param2) = explode(" ", microtime());
15   $semilla = $param2 + $param1 * 100000;
16   print "Semilla ".$semilla."<br>";
17   srand($semilla);
18   $r = rand();
19   print $r;
20   print "<hr>";
21
22   list($param1, $param2) = explode(" ", microtime());
23   $semilla = $param2 + $param1 * 100000;
24   print "Semilla ".$semilla."<br>";
25   mt_srand($semilla);
26   $r = mt_rand();
27   print $r;
28   print "<hr>";
29
30  ?>
31 </body>
32 </html>
```

**Listado 11.54.1.** aleatorios.php



*[www.pacoarce.com](http://www.pacoarce.com)*

## 11.55. Exponentes y raíz cuadrada

- Contamos con la función **exp()** elevamos a “e” a una potencia, donde 'e' es la base del sistema natural de logaritmos, y su valor aproximado es 2.718282.
- Con la función **pow()** podemos elevar un número a cualquier potencia.
- La función **sqrt()** obtenemos la raíz cuadrada de un número.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones matemáticas | Raiz cuadrada y exponentes</title>
5  </head>
6  <body>
7    <?php
8      print exp(12). "<br>";
9      print exp(7.2). "<br>";
10     print "<hr>";
11     print pow(2,8). "<br>";
12     print pow(3, 3). "<br>";
13     print pow(10,-2.3). "<br>";
14     print "<hr>";
15     print sqrt(9). "<br>";
16     print sqrt(10). "<br>";
17     print "<hr>";
18
19  ?>
20 </body>
21 </html>
```

**Listado 11.55.1.** exponentesRaizCuadrada.php

[www.pacoarce.com](http://www.pacoarce.com)



## 11.56. Cambio de bases

Contamos con muchas funciones en PHP para cambiar de base de un número:

***base\_convert()***— Convertir un número entre bases arbitrarias.

***bindec()*** - Binario a decimal

***octdec()*** - Octal a decimal

***hexdec()*** - Hexadecimal a decimal

***decbin()*** - Decimal a binario

***decoct()*** - Decimal a octal

***dechex()*** - Decimal a hexadecimal

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones matemáticas | Cambio de base</title>
5  </head>
6  <body>
7    <?php
8      $hexadecimal = "FF6600";
9      print base_convert($hexadecimal, 16, 2);
10     print "<br>";
11     print base_convert($hexadecimal, 16, 8);
12     print "<br>";
13     print base_convert($hexadecimal, 16, 10);
14     print "<hr>";
15     print hexdec("ABCD");
16
17  ?>
18 </body>
19 </html>
```

**Listado 11.56.1.** cambioBases.php

[www.pacoarce.com](http://www.pacoarce.com)

## 11.57. Recuperar la parte entera y la parte decimal de una división

- Por medio de la función `intdiv()` (Sólo en PHP 7) podemos realizar una división y regresa la parte entera de la misma.
- Con la función `fmod()` regresamos la parte decimal de una división.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones matemáticas | intdiv y fmode</title>
5  </head>
6  <body>
7    <?php
8      print intdiv(3, 2)."<br>";
9      print intdiv(35, 7)."<br>";
10     print intdiv(75, 7)."<br>";
11     print "<hr>";
12     //4.3846153846
13     //4*1.3 = 5.2 (5.7-5.2) => 0.5
14     print fmod(5.7,1.3);
15
16
17
18
19  ?>
20 </body>
21 </html>
```

**Listado 11.57.1.** divisionEntera.php

*www.pacoarce.com*

## 11.58. Funciones de CSPRNG

- *random\_int* — Crea números enteros pseudo-aleatorios criptográficamente seguros.
- *random\_bytes* — Crea bytes pseudo-aleatorios criptográficamente seguros

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Funciones matemáticas | Funciones CSPRNG</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $bytes = random_bytes(10);
10   print bin2hex($bytes);
11   print "<hr>";
12   print random_int(0, 1024);
13   print "<hr>";
14   print random_int(-1024, 1024);
15
16  ?>
17 </body>
18 </html>
```

**Listado 11.58.1.** funcionesCSPRNG.php

[www.pacoarce.com](http://www.pacoarce.com)

## Cuestionario:

**1) ¿Cómo se define una variable de tipo string en PHP?**

- 1) char str;
- 2) string str;
- 3) En PHP no se define el tipo de las variables explícitamente

**2) Tenemos el siguiente código: \$a="10"; \$b=\$a + 2; ¿Cuál será el valor de \$b?**

- 1) "12"
- 2) 12
- 3) "102"
- 4) Ninguno (no se puede sumar un número a una cadena)

**3) En PHP, ¿qué función se emplea para convertir la primera letra de cada palabra en mayúscula?**

- 1) upper()
- 2) strtoupper()
- 3) ucwords()
- 4) Las anteriores respuestas no son correctas

**4) En PHP, ¿cómo se calcula el máximo de los números 2 y 4?**

- 1) max(2, 4)
- 2) max(array(2, 4))
- 3) max(array(0 => 2, 1 => 4))
- 4) Todas las respuestas son correctas

**5) En PHP, ¿qué función se emplea para convertir una cadena a minúsculas?**

- 1) lower()
- 2) tolower()
- 3) strtolower()
- 4) str2lower()

**6) En PHP, para eliminar los espacios en blanco al final de una cadena se emplea**

- 1) ltrim()
- 2) rtrim()
- 3) strip()
- 4) trim()

**7) En PHP, ¿qué función se emplea para invertir una cadena?**

- 1) inverse()
- 2) reverse()
- 3) strrev()
- 4) Las anteriores respuestas no son correctas

**8) En PHP, ¿cómo se genera un número aleatorio?**

- 1) rnd()
- 2) round()
- 3) rand()
- 4) Las anteriores respuestas no son correctas

**9) La función "strlen" tiene como finalidad**

- 1) Indica el número de caracteres de una cadena
- 2) Indica el número de elementos de un arreglo.
- 3) Indica el número de elementos de un objeto.
- 4) Indica el número de elementos de una tabla

**10) La función "trim()" sirve para:**

- 1) Regresa el número de caracteres de una cadena
- 2) Limpia las cadenas de espacios en blanco.
- 3) Regresa una subcadena
- 4) Verifica que exista una variable.

**11) El comando "explode()" nos permite dividir una cadena en diferentes elementos, según un separador, por ejemplo "/":**

- 1) Verdadero
- 2) Falso

**12) El comando para encontrar una subcadena en una cadena es:**

- 1) len()
- 2) substring()
- 3) strstr()
- 4) find()

**13) El comando que nos indica la posición donde se encuentra una subcadena dentro de otra cadena es:**

- 1) find()

- 2) substr()
- 3) strstr()
- 4) strpos()

**14) El comando para convertir una cadena a letras mayúsculas es:**

- 1) strtoupper()
- 2) strtolower()
- 3) upper()
- 4) strstr()

**15) El comando para sustraer una subcadena de una cadena, a partir de la posición inicial y hasta la posición final, es:**

- 1) substr()
- 2) substr()
- 3) strpos()
- 4) strlen()

**16) El comando para convertir una cadena a letras minúsculas es:**

- 1) strstr()
- 2) strtolower()
- 3) strtoupper()
- 4) lower()

**17) El comando para sustituir una subcadena dentro de otra cadena es:**

- 1) substr()
- 2) replace()
- 3) strstr()
- 4) str\_replace()

**18) El comando para determinar la longitud de una cadena en PHP es:**

- 1) strstr()
- 2) len()
- 3) strlen()
- 4) lenght()

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 12: Manejo de archivos

**Objetivo:** Manipulará los archivos y directorios del servidor por medio de PHP.

#### Introducción:

Una característica muy importante de **PHP** es que podemos manipular a nuestro antojo los archivos que se encuentran en el servidor (**file system**), con lo cual están desarrollados muchos manejadores de archivos de la web. En este capítulo veremos los siguientes comandos:

- Verificar si existe un archivo: **file\_exists()**
- Determinar si es un archivo o un directorio
- Propiedades de un archivo
- Leer el contenido de un directorio.
- Constantes predefinidas
- Leer el contenido de una carpeta o directorio con **scandir()**
- Conocer la ruta de un archivo con **dirname()**, **basename()**, **realpath()**

## 12.1. Verificar si existe un archivo: file\_exists()

Antes de hacer alguna operación sobre algún archivo, es buena idea asegurarse que éste exista, de lo contrario se genera un error que el usuario final verá en su navegador. Para esto se usa el comando "file\_exists".

La sintaxis básica para este comando es:

`$existe = file_exists( nombreArchivo );`

El parámetro "nombreArchivo" es de tipo cadena.

El resultado es de tipo booleano, verdadero o falso.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | file_exists</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $archivo = "datos.txt";
10     $carpeta = "datos/";
11     if (file_exists($carpeta.$archivo)) {
12       print "<p>Si existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
13     } else {
14       print "<p>NO existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
15     }
16
17
18
19   ?>
20 </body>
21 </html>
```

**Listado 12.1.1.** PHP7-1201-ArchivosFileExists.php

[www.pacoarce.com](http://www.pacoarce.com)



## 12.2. Determinar si es un archivo o un directorio

Determinar si es un archivo o un directorio

Al estar trabajando con los archivos de un directorio, es posible "toparse" con uno o varios directorios dentro del directorio que se está examinando. Para saber de qué tipo es un "archivo", se usan los comandos "is\_file" y "is\_dir".

La sintaxis básica para estos comandos es:

```
$archivo = is_file(nombre_del_archivo);
```

```
$carpeta = is_dir( nombre_del_archivo );
```

El parámetro "nombre\_del\_archivo" es de tipo cadena. El resultado es de tipo booleano.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | file_exists</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $archivo = "prueba";
10   $carpeta = "datos/";
11   if (file_exists($carpeta.$archivo)) {
12     if (is_file($carpeta.$archivo)) {
13       print "<p>Si existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
14     } else if(is_dir($carpeta.$archivo)){
15       print "<p>Si existe la carpeta '". $archivo.'"</p>";
16     } else {
17       print "<p>No se pudo determinar el tipo de archivo</p>";
18     }
19
20
21   } else {
22     print "<p>NO existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
23   }
24
25
26
27  ?>
28 </body>
29 </html>
```

**Listado 12.2.1.** PHP7-1202-ArchivosIsFileIsDir.php

## 12.3. Propiedades de un archivo

### Propiedades de un archivo

Todos los archivos poseen ciertas propiedades, como tamaño y fechas de creación y modificación. Estas cualidades se pueden obtener con los comandos "filesize", "filectime" y "filemtime".

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | file_exists</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $archivo = "perro.jpg";
10   $carpeta = "imagenes/";
11   if (file_exists($carpeta.$archivo)) {
12     if (is_file($carpeta.$archivo)) {
13       print "<p>Si existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
14       $size = filesize($carpeta.$archivo);
15       $creado = filectime($carpeta.$archivo);
16       $modificado = filemtime($carpeta.$archivo);
17       if($size < 1024){
18         $medida = "b";
19       } else if ($size >=1024 && $size < 1024000) {
20         $medida = "Kb";
21         $size /= 1000;
22       } else {
23         $medida = "Mb";
24         $size /= 1000000;
25       }
26       $sCreado = date("d/m/Y H:i:s",$creado);
27       $sModificado = date("d/m/Y H:i:s",$modificado);
28       print "<ul>";
29       print "<li>Tamaño: '". $size.'" ". $medida."</li>";
30       print "<li>Creado: '". $sCreado."</li>";
31       print "<li>Modificado: '". $sModificado."</li>";
32       print "</ul>";
33     } else if(is_dir($carpeta.$archivo)){
34       print "<p>Si existe la carpeta '". $archivo.'"</p>";
35     } else {
36       print "<p>No se pudo determinar el tipo de archivo</p>";
37     }
38
39
40   } else {
41     print "<p>NO existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
```

```
42 }  
43  
44  
45  
46 ?>  
47 </body>  
48 </html>
```

**Listado 12.3.1.** PHP7-1203-ArchivosFileSize.php



*www.pacoarce.com*

## 12.4. Leer el contenido de un directorio

Leer el contenido de un directorio

Por medio de una sencilla función podemos leer los archivos dentro de un directorio. De hecho los principales manejadores de archivos en la web están escritos con PHP por su enorme facilidad para manipular los archivos y directorios del sistema operativo.

`resource opendir ( string $path [, resource $context ] )`

Leer el contenido de un directorio

`readdir` — Lee una entrada desde un gestor de directorio. Devuelve el nombre de la siguiente entrada del directorio. Las entradas son devueltas en el orden en que fueron almacenadas por el sistema de archivos.

`string readdir ([ resource $dir_handle ] );`

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | Lee el directorio</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $archivo = ".";
10   $carpeta = "";
11   if (file_exists($carpeta.$archivo)) {
12     if (is_file($carpeta.$archivo)) {
13       print "<p>Si existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
14       $size = filesize($carpeta.$archivo);
15       $creado = filectime($carpeta.$archivo);
16       $modificado = filemtime($carpeta.$archivo);
17       if($size < 1024){
18         $medida = "b";
19       } else if ($size >=1024 && $size < 1024000) {
20         $medida = "Kb";
21         $size /= 1000;
22       } else {
23         $medida = "Mb";
24         $size /= 1000000;
25       }
26       $sCreado = date("d/m/Y H:i:s",$creado);
27       $sModificado = date("d/m/Y H:i:s",$modificado);
28       print "<ul>";
29       print "<li>Tamaño: ". $size." ". $medida."</li>";
30       print "<li>Creado: ". $sCreado."</li>";
31       print "<li>Modificado: ". $sModificado."</li>";
32       print "</ul>";
33     } else if(is_dir($carpeta.$archivo)){
```

```
34  print "<p>Si existe la carpeta '". $archivo.'"</p>";
35  $dirID = opendir($carpeta.$archivo);
36  while ($nombre = readdir($dirID)) {
37      print $nombre;
38      if(is_file($archivo."/".$nombre)) print " > archivo";
39      if(is_dir($archivo."/".$nombre)) print " > dir";
40      print "<br>";
41  }
42  } else {
43      print "<p>No se pudo determinar el tipo de archivo</p>";
44  }
45
46
47  } else {
48      print "<p>NO existe el archivo '". $archivo.'" en la carpeta '". $carpeta.'"</p>";
49  }
50
51
52
53  ?>
54  </body>
55  </html>
```

**Listado 12.4.1.** PHP7-1204-ArchivosLeeDirectorio.php

*www.pacoarce.com*

## 12.5. Constantes predefinidas

\_\_LINE\_\_ Número de línea actual en el archivo.

\_\_FILE\_\_ Ruta completa y nombre del archivo. Si se usa dentro de un include, devolverá el nombre del fichero incluido.

\_\_DIR\_\_ Directorio del archivo. Si se utiliza dentro de un include, devolverá el directorio del archivo incluido.

\_\_TRAIT\_\_ El nombre del trait. El nombre del trait incluye el espacio de nombres en el que fue declarado.

\_\_METHOD\_\_ Nombre del método de la clase.

\_\_NAMESPACE\_\_ Nombre del espacio de nombres actual.

\_\_FUNCTION\_\_ Nombre de la función.

\_\_CLASS\_\_ Nombre de la clase.

---

```
1  <?php
2  function prueba(){
3  print __LINE__." desde dentro de la funcion<br>";
4  print __FUNCTION__."<br>";
5  }
6  print __LINE__." desde fuera de la funcion<br>";
7  print __FILE__."<br>";
8  print __DIR__."<br>";
9  prueba()
10 ?>
```

**Listado 12.5.1.** constantesPredifinidas.php

*www.pacoarce.com*

## 12.6. Leer el contenido de una carpeta o directorio con scandir()

Versiones: (PHP 5, PHP 7)

Regresa los archivos en un vector numérico.

scandir() — Enumera los archivos y directorios ubicados en la ruta especificada.

0 forma ascendente.

1 forma descendente.

Versiones: (PHP 5, PHP 7)

getcwd() — Obtiene el directorio actual en donde se está trabajando

Versiones: (PHP 4, PHP 5, PHP 7)

chdir — Cambia de directorio

---

```
1  <?php
2  $dir = "img/";
3  $archivos1 = scandir($dir);
4  $archivos2 = scandir($dir,1);
5
6  foreach ($archivos1 as $key => $value) {
7  print "[".$key."] => ".$value."<br>";
8  }
9  print "<br><br>";
10 foreach ($archivos2 as $key => $value) {
11 print "[".$key."] => ".$value."<br>";
12 }
13 print "<br><br>";
14 print getcwd();
15 print "<br>";
16 chdir($dir);
17 print getcwd();
18 print "<br>";
19 chdir("../");
20 print getcwd();
21 print "<br>";
22 mkdir("tmp");
23 chdir("tmp");
24 print getcwd();
25 print "<br>";
26 chdir("../");
27 rmdir("tmp");
28 ?>
```

**Listado 12.6.1.** leeContenidoDirectorio.php

## 12.7. Conocer la ruta de un archivo con dirname(), basename(), realpath()

- **dirname()** — Devuelve la ruta de un directorio padre.
- **basename()** - Devuelve el último componente de nombre de una ruta.
- **pathinfo()** - Devuelve información acerca de la ruta de un archivo.
- **realpath()** - Devuelve el nombre de la ruta absoluta.

```
1  <?php
2  print "La constantes __FILE__"."<br>";
3  print __FILE__;
4  print "<br><br>";
5
6  print "real_path(__FILE__)"."<br>";
7  print realpath(__FILE__);
8  print "<br><br>";
9
10 print "dirname()"."<br>";
11 print dirname(__FILE__);
12 print "<br><br>";
13
14 print "dirname()"."<br>";
15 print dirname(dirname(__FILE__));
16 print "<br><br>";
17
18 print "dirname(__FILE__,3) unicamente con PHP7"."<br>";
19 print dirname(__FILE__,3);
20 print "<br><br>";
21
22 print "basename(__FILE__)"."<br>";
23 print basename(__FILE__);
24 print "<br><br>";
25
26 $path_array = pathinfo(__FILE__);
27
28 print "dirname: ".$path_array["dirname"]."<br>";
29 print "basename: ".$path_array["basename"]."<br>";
30 print "extension: ".$path_array["extension"]."<br>";
31 print "filename: ".$path_array["filename"]."<br>";
32 ?>
```

**Listado 12.7.1.** funcionesRutaArchivos.php



## Cuestionario:

**1) En PHP, para obtener el tamaño de un fichero se emplea la función:**

- 1) file()
- 2) filesize()
- 3) fsize()
- 4) f\_size()

**2) En PHP, para comprobar que un archivo es un fichero normal se emplea**

- 1) is\_archive()
- 2) is\_file()
- 3) is\_link()
- 4) is\_normal()

**3) El comando para verificar si un archivo existe o no, es:**

- 1) file()
- 2) file\_exist()
- 3) touch()
- 4) file\_exists()

**4) El comando para verificar si un archivo es un archivo del sistema (no una carpeta o directorio) es:**

- 1) is\_dir()
- 2) is\_file()
- 3) file\_exists()
- 4) not\_dir()

**5) El comando que determina si un archivo del sistema es un directorio o carpeta es:**

- 1) is\_dir()
- 2) is\_file()
- 3) touch()
- 4) file\_exists()

**6) El comando para saber la fecha de creación de un archivo del sistema es:**

- 1) createTime()
- 2) filetime()
- 3) filectime()
- 4) filemtime()

**7) El comando para obtener la fecha de modificación de un archivo del sistema es:**

- 1) filetime()
- 2) filemtime()
- 3) filectime()
- 4) touch()

**8) Para abrir un directorio con PHP se utiliza el comando:**

- 1) dir()
- 2) open()
- 3) open\_dir()
- 4) opendir()

**9) Para borrar físicamente un archivo del sistema operativo desde PHP se utiliza el comando:**

- 1) unlink()
- 2) delete()
- 3) erase()
- 4) delete\_file()

**10) Cuando borramos un archivo del sistema por medio de PHP, podemos recuperarlo de la papelera:**

- 1) Verdadero
- 2) Falso

**11) El comando en PHP para abrir un directorio dentro del servidor es:**

- 1) file()
- 2) dir()
- 3) open()
- 4) opendir()

**12) El comando en PHP para leer cada uno de los archivos del directorio o carpeta del servidor es:**

- 1) readdir()
- 2) dir()
- 3) read()
- 4) file()

**13) El archivo oculto nombrado como ".." significa:**

- 1) Que hay una carpeta anterior.
- 2) Es la carpeta raíz.
- 3) Hay más carpetas adelante.
- 4) Cuando hay un error en la lectura de los archivos.

**14) La función que devuelve la ruta de un directorio padre es:**

- 1) basename()
- 2) dirname()
- 3) pathinfo()
- 4) realpath()

**15) La función que devuelve el último componente de nombre de una ruta es:**

- 1) realpath()
- 2) pathinfo()
- 3) dirname()
- 4) basename()

**16) La función que devuelve información acerca de la ruta de un archivo como un arreglo es:**

- 1) pathinfo()
- 2) realpath()
- 3) basename()
- 4) dirname()

**17) La función que devuelve el nombre de la ruta absoluta es:**

- 1) dirname()
- 2) basename()
- 3) realpath()
- 4) pathinfo()

**18) La función que enumera los archivos y directorios ubicados en la ruta especificada es:**

- 1) basename()
- 2) dirname()
- 3) scandir()
- 4) dir\_files()

# Todo PHP7

## De novato a experto

### Capítulo 13: Archivos de datos en el servidor

**Objetivo:** El alumno aprenderá a abrir, leer y escribir en los archivos del servidor.

#### Introducción:

Una característica muy importante de **PHP** es que podemos crear, leer y borrar archivos planos de datos en el servidor, lo cual nos permitirá manipular información inmediata de forma mucha más directa que el acceso a una base de datos, como se verá en los siguientes capítulos. los temas que veremos en esta sección son:

- El comando touch()
- Borrar un archivo, el comando unlink()
- Modos de abrir un archivo
- Leer un archivo de datos
- Escribir en un archivo de datos
- Añadir datos a un archivo
- Proteger los archivos con flock() .
- Copiar y renombrar archivos
- Leer un archivo CSV con fgetcsv()
- Crear un archivo con formato CSV fputcsv()
- Leer todo un archivo en una cadena: file\_get\_contents

## 13.1. El comando touch()

La manera más segura de crear un archivo es usando el comando "touch()", el cual recibe el nombre de un archivo y primero verifica si existe.

Si el archivo existe, no le hace nada a sus contenidos y solo cambia la fecha de modificación.

Si el archivo NO existe, crea un archivo vacío.

Al ejecutar este comando ya no es necesario usar el comando "file\_exists()". Recuerde tomar en cuenta los permisos para crear archivos en el directorio de trabajo o directorio especificado.

El comando touch()

La sintaxis básica para este comando es:

touch( nombre\_del\_archivo );

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | Touch</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $contador = "contador.txt";
10   if (touch($contador)) {
11     print "No hubo error con el touch()";
12   } else {
13     print "Existió un error con el touch()";
14   }
15   ?>
16 </body>
17 </html>
```

**Listado 13.1.1.** PHP7-1301-ArchivosTouch.php

*www.pacoarce.com*

## 13.2. Borrar un archivo, el comando unlink()

Borrar un archivo, el comando unlink()

Para borrar un archivo, se usa el comando "unlink()". Si el archivo no existe, no pasa nada. Si el archivo existe, es borrado inmediatamente. NO SE TRASLADA EL ARCHIVO A LA PAPELERA (TRASH), NI SE PIDE CONFIRMACIÓN.

La sintaxis básica para este comando es:

unlink( nombre\_del\_archivo );

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | Unlink</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $contador = "contador.txt";
10   if (touch($contador)) {
11     print "No hubo error con el touch()";
12   } else {
13     print "Existió un error con el touch()";
14   }
15   print "<br>";
16   if(unlink($contador)){
17     print "El archivo ".$contador." se borró exitosamente";
18   } else {
19     print "Error al borrar el archivo ".$contador;
20   }
21   ?>
22 </body>
23 </html>
```

**Listado 13.2.1.** PHP7-1302-ArchivosUnlink.php

*www.pacoarce.com*

## 13.3. Modos de abrir un archivo

Esta operación establece un "canal de comunicación directo" entre la rutina de PHP y el archivo.

La sintaxis básica para este comando es:

**\$apuntador = fopen( nombre\_del\_archivo, "modo" );**

El parámetro "nombre\_del\_archivo" es de tipo cadena. El parámetro "modo" es una letra de la tabla anterior. El resultado es un número entero único que identifica ese canal de comunicación.

'r' Apertura para sólo lectura; coloca el puntero al archivo al principio del archivo.

'r+' Apertura para lectura y escritura; coloca el puntero al archivo al principio del archivo.

'w' Apertura para sólo escritura; coloca el puntero al archivo al principio del archivo y trunca el archivo a longitud cero. Si el archivo no existe se intenta crear.

'w+' Apertura para lectura y escritura; coloca el puntero al archivo al principio del archivo y trunca el archivo a longitud cero. Si el archivo no existe se intenta crear.

'a' Apertura para sólo escritura; coloca el puntero del archivo al final del mismo. Si el archivo no existe, se intenta crear. En este modo, fseek() solamente afecta a la posición de lectura; las lecturas siempre son pospuestas.

'a+' Apertura para lectura y escritura; coloca el puntero del archivo al final del mismo. Si el archivo no existe, se intenta crear. En este modo, fseek() no tiene efecto, las escrituras siempre son pospuestas.

'x' Creación y apertura para sólo escritura; coloca el puntero del archivo al principio del mismo. Si el archivo ya existe, la llamada a fopen() fallará devolviendo FALSE y generando un error de nivel E\_WARNING. Si el archivo no existe se intenta crear. Esto es equivalente a especificar las banderas O\_EXCL|O\_CREAT para la llamada al sistema de open(2) subyacente.

'x+' Creación y apertura para lectura y escritura; de otro modo tiene el mismo comportamiento que 'x'.

'c' Abrir el archivo para sólo escritura. Si el archivo no existe, se crea. Si existe no es truncado (a diferencia de 'w'), ni la llamada a esta función falla (como en el caso con 'x'). El puntero al archivo se posiciona en el principio del archivo. Esto puede ser útil si se desea obtener un bloqueo asistido (véase flock()) antes de intentar modificar el archivo, ya que al usar 'w' se podría truncar el archivo antes de haber obtenido el bloqueo (si se desea truncar el archivo, se puede usar truncate() después de solicitar el bloqueo).

'c+' Abrir el archivo para lectura y escritura; de otro modo tiene el mismo comportamiento que 'c'.

'e' Establecer la bandera 'close-on-exec' en el descriptor de archivo abierto. Disponible solamente en PHP compilado en sistemas que se ajustan a POSIX.1-2008.

Al terminar de trabajar con un archivo, se recomienda cerrar el canal de comunicación, con el

comando "fclose()". La sintaxis básica para este comando es:

**fclose( apuntador\_al\_archivo );**

El parámetro "apuntador\_al\_archivo" es un entero que se obtuvo al abrir el archivo.



*www.pacoarce.com*



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | fopen() y fclose()</title>
5  </head>
6  <body>
7    <?php
8      $archivo = "datos.txt";
9      if (touch($archivo)) {
10         //Abrimos con "a" append
11         $id = fopen($archivo, "a");
12         var_dump($id);
13         //cerramos
14         fclose($id);
15     } else {
16         print "Error en el touch del archivo";
17     }
18     ?>
19 </body>
20 </html>
```

**Listado 13.3.1.** PHP7-1303-ArchivosFopen.php

*www.pacoarce.com*

## 13.4. Leer un archivo de datos

Cuando se abre un archivo en modo de lectura, los contenidos del archivo se accesan con el comando "fgets()". Este comando lee línea por línea del archivo, donde una línea termina con el caracter "newline" o "\n".

Los archivos de texto creados en los sistemas Windows, Unix, Linux y algunos programas de Mac OS X incluyen este caracter para separar las líneas. Pero los programas del sistema Mac OS 9 o menor, y algunos programas del sistema Mac OS X, no generan este caracter para separar las líneas. Así que debe tener cuidado al leer un archivo de texto creado bajo estas circunstancias.

La sintaxis básica para este comando es:

**\$línea = fgets( apuntador\_al\_archivo, max\_caracteres );**

El parámetro "apuntador\_al\_archivo" es un entero que se obtuvo al abrir el archivo. A veces no se encuentra un carácter "newline", y para cubrir estos casos, el segundo parámetro indica cuántos caracteres máximo deben leerse si no se encuentra un "newline". El comando regresa una cadena.

El comando "feof()" permite preguntar si ya se llegó al final de un archivo.

La sintaxis básica para este comando es:

**\$r = feof( apuntador\_al\_archivo );**

El parámetro "apuntador\_al\_archivo" es un entero que se obtuvo al abrir el archivo. El comando regresa un booleano indicando si ya se llegó al final del archivo: mientras no se llegue al final del archivo, este comando regresa "false".

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivo | Lee Archivo</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $archivo = "datos/datos.txt";
10   if (touch($archivo)) {
11     $id = fopen($archivo, "r");
12     //ciclo de lectura del archivo
13     while (!feof($id)) {
14       //leemos un registro a la vez "newline" \n \r
15       $línea = fgets($id,1024);
16       //
17       print "<p>".$línea."</p>";
18     }
19     fclose($id);
```

```
20 } else {  
21     print "<p>Error al acceder al archivo</p>";  
22 }  
23 ?>  
24 </body>  
25 </html>
```

**Listado 13.4.1.** PHP7-1304-ArchivosLeeArchivo.php



*www.pacoarce.com*

## 13.5. Escribir en un archivo de datos

Para agregar datos en un archivo existente, se abre el archivo en modo "w" y se usa el comando "fwrite". La sintaxis básica para este comando es:

**fwrite( apuntador\_al\_archivo, dato\_a\_grabar );**

El parámetro "apuntador\_al\_archivo" es un entero que se obtuvo al abrir el archivo. El parámetro "dato\_a\_grabar" es un "string". Si se está escribiendo un archivo de texto, hay que incluir el carácter "\n" al final de cada línea.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | Guardar Datos</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $archivo = "datos/datos.txt";
10   if (touch($archivo)) {
11     $linea1 = "Lindo pescadito,\n";
12     $linea2 = "no quieres salir,\n";
13     $linea3 = "a jugar con mi aro,\n";
14     $linea4 = "vamos al jardin.\n";
15     //
16     $id = fopen($archivo,"w");
17     //
18     fwrite($id,$linea1);
19     fwrite($id,$linea2);
20     fwrite($id,$linea3);
21     fwrite($id,$linea4);
22     //
23     fclose($id);
24     //
25     //leemos el archivo
26     //
27     $id = fopen($archivo, "r");
28     //
29     while (!feof($id)) {
30       $linea = fgets($id,1024);
31       print "<p>".$linea."</p>";
32     }
33     fclose($id);
34
35   } else {
36     print "Error al manipular el archivo";
37   }
38   ?>
39 </body>
```

```
40 </html>
```

**Listado 13.5.1.** PHP7-1305-ArchivosGuardarDatos.php



*www.pacoarce.com*

## 13.6. Añadir datos a un archivo

Para agregar datos al final de un archivo existente, se abre el archivo en modo "a" y se usa el comando "fwrite()". La sintaxis básica para este comando es:

```
fwrite( apuntador_al_archivo, dato_a_grabar );
```

El parámetro "apuntador\_al\_archivo" es un entero que se obtuvo al abrir el archivo. El parámetro "dato\_a\_grabar" es una cadena. Si se está escribiendo un archivo de texto, hay que incluir el carácter "\n" al final de cada línea.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Archivos | Guardar Datos</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    $archivo = "datos/datos.txt";
10   if (touch($archivo)) {
11     $linea1 = "Lindo pescadito,\n";
12     $linea2 = "no quieres salir,\n";
13     $linea3 = "a jugar con mi aro,\n";
14     $linea4 = "vamos al jardin.\n";
15     //
16     $id = fopen($archivo,"a");
17     //
18     fwrite($id,$linea1);
19     fwrite($id,$linea2);
20     fwrite($id,$linea3);
21     fwrite($id,$linea4);
22     //
23     fclose($id);
24     //
25     //leemos el archivo
26     //
27     $id = fopen($archivo, "r");
28     //
29     while (!feof($id)) {
30       $linea = fgets($id,1024);
31       print "<p>".$linea."</p>";
32     }
33     fclose($id);
34   } else {
35     print "Error al manipular el archivo";
36   }
37   ?>
38 </body>
39 </html>
```

**Listado 13.6.1.** PHP7-1306-ArchivosAnadirDatos.php



*[www.pacoarce.com](http://www.pacoarce.com)*

## 13.7. Proteger los archivos con flock()

Para evitar que varios usuarios escriban sobre un archivo al mismo tiempo, se usan los comandos "flock" y "is\_writable". Estos comandos evitan que otra rutina pueda escribir en un archivo mientras se está actualizando.

La sintaxis básica para estos comandos es:

```
$si_no=is_writable(nombre_de_archivo);
```

```
flock( apuntador_al_archivo, entero );
```

Proteger los archivos con flock()

El parámetro "nombre\_del\_archivo" es un "string". El parámetro "apuntador\_al\_archivo" es un entero que se obtuvo al abrir el archivo. El parámetro "entero" es un número de la siguiente tabla:

- 1 de "seguro" compartido, permite a otras rutinas leer el archivo, pero no escribir en él.
- 2 exclusivo, evita que otras rutinas lean y escriban el archivo.
- 3 liberar, quita el seguro al archivo.

Proteger los archivos con flock()

El comando "flock" siempre se usa en pares, uno para proteger el archivo y otro para liberarlo.

Estas operaciones se ejecutan justo después de abrir el archivo y justo antes de cerrarlo.

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Archivos | Proteger los Datos</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9 $archivo = "datos/datos.txt";
10 if (touch($archivo)) {
11   $bandera = false;
12   $maximo = 100000;
13   $contador = 0;
14   //
15   while (!$bandera) {
16     if (is_writable($archivo)) {
17       $bandera = true;
18       break;
19     }
20     $contador++;
21     if ($contador > $maximo) {
22       break;
23     }
24   }
25   if ($bandera) {
```



```
26 //primera parte
27 $linea1 = "En el agua clara,\n";
28 $linea2 = "que brota en la fuente,\n";
29 $linea3 = "un lindo pescado,\n";
30 $linea4 = "sale de repente.\n";
31 //
32 $id = fopen($archivo,"w");
33 //
34 //Hacemos el archivo exclusivo
35 //
36 flock($id, 2);
37 //
38 fwrite($id,$linea1);
39 fwrite($id,$linea2);
40 fwrite($id,$linea3);
41 fwrite($id,$linea4);
42 //
43 fclose($id);
44 //
45 $linea1 = "Lindo pescadito,\n";
46 $linea2 = "no quieres salir,\n";
47 $linea3 = "a jugar con mi aro,\n";
48 $linea4 = "vamos al jardin.\n";
49 //
50 $id = fopen($archivo,"a");
51 //
52 fwrite($id,$linea1);
53 fwrite($id,$linea2);
54 fwrite($id,$linea3);
55 fwrite($id,$linea4);
56 //
57 fclose($id);
58 //
59 //liberamos el archivo
60 //
61 flock($id, 3);
62 }
63
64 //
65 //leemos el archivo
66 //
67 $id = fopen($archivo, "r");
68 //
69 while (!feof($id)) {
70     $linea = fgets($id,1024);
71     print "<p>".$linea."</p>";
72 }
73 fclose($id);
74 } else {
75     print "Error al manipular el archivo";
76 }
77 ?>
```

```
78 </body>
79 </html>
```

**Listado 13.7.1.** PHP7-1307-ArchivosFlock.php



*www.pacoarce.com*

## 13.8. Copiar y renombrar archivos

(PHP 4, PHP 5, PHP 7)

copy() — Copia un fichero

Sintaxis

bool copy ( string \$origen , string \$destino )

Si el archivo destino ya existe, será sobrescrito.

(PHP 4, PHP 5, PHP 7)

rename() — Renombra un archivo o directorio

Sintaxis:

bool rename ( string \$nombre , string \$nuevo )

---

```
1  <?php
2
3  touch("datos.txt");
4
5  if (copy("datos.txt", "datos1.txt")) {
6  print "Copia de archivo correcta";
7  } else {
8  print "Error al copiar el archivo";
9  }
10 print "<br><br>";
11
12 if (@copy("datos.txt", "respaldo/datos1.txt")) {
13 print "Copia de archivo a respaldo correcta";
14 } else {
15 print "Error al copiar a respaldo el archivo";
16 }
17 print "<br><br>";
18
19 if (@rename("datos1.txt", "datos2.txt")) {
20 print "Renombrar de archivo correcto";
21 } else {
22 print "Error al renombrar el archivo";
23 }
24 print "<br><br>";
25
26 if (@rename("datos2.txt", "respaldo/datos3.txt")) {
27 print "Renombrar de archivo correcto";
28 } else {
29 print "Error al renombrar el archivo";
30 }
31 print "<br><br>";
32
33 if (@rename("datos.txt", "respaldo/datos.txt")) {
34 print "Mover el archivo correcto";
```

```
35 } else {  
36 print "Error al mover el archivo";  
37 }  
38 print "<br><br>";  
39  
40 ?>
```

**Listado 13.8.1.** archivosCopiarRenombrar.php



*www.pacoarce.com*

## 13.9. Leer un archivo CSV con fgetcsv()

(PHP 4, PHP 5, PHP 7)

Por medio de la función fgetcsv() podemos leer un archivo con el formato CVS.

Regresa un arreglo indexado con la información.

array fgetcsv ( resource \$handle [, int \$length = 0 [, string \$delimiter = "," [, string \$enclosure = ""  
[, string \$escape = "\" ]]] )

```
1  <?php
2  $archivo = fopen("prueba.csv","r");
3
4  $inicio = true;
5
6  while (!feof($archivo)) {
7
8      //Leemos el registro en formato CSV
9      $data = fgetcsv($archivo);
10
11     if ($inicio) {
12         $inicio = false;
13         continue;
14     }
15     print $data[0].") ".$data[1]."<br>";
16     print $data[2]."<br>";
17     print $data[3]."<br>";
18     print $data[4]."<br>";
19     print $data[5]."<br>";
20     print "Respuesta correcta: ".$data[6]."<br>";
21     print "<br>";
22 }
23 fclose($archivo);
24 ?>
```

**Listado 13.9.1.** archivosCSV.php

[www.pacoarce.com](http://www.pacoarce.com)

## 13.10. Crear un archivo con formato CSV

(PHP 5 >= 5.1.0, PHP 7)

fputcsv — Toma un arreglo y lo guarda como formato CSV.

Sintaxis:

int fputcsv ( resource \$manejador , array \$campos [, string \$delimitador = "," [, string \$comillas = "" [, string \$caracteresControl = "\" ]]] )

```
1  <?php
2
3  $data = array(
4  array('1','Pedro Picapiedra','3000.00','Operador'),
5  array('2','Pablo Marmol','5000.00','Gerente'),
6  array('3','Vilma Picapiedra','10000.00','Directora General')
7  );
8
9  touch("data.csv");
10 $archivo = fopen("data.csv","w");
11
12 foreach ($data as $empleado) {
13 fputcsv($archivo, $empleado);
14 }
15
16 fclose($archivo);
17
18 ?>
```

**Listado 13.10.1.** guardarCSV.php

[www.pacoarce.com](http://www.pacoarce.com)

## 13.11. Leer todo un archivo en una cadena: file\_get\_contents

(PHP 4 >= 4.3.0, PHP 5, PHP 7)

file\_get\_contents — Lee un archivo completo, o a partir del “desplazamiento”

Sintaxis:

string file\_get\_contents ( string \$archivo [, bool \$incluidoPath = FALSE [, int \$desplazamiento = 0 [, int \$maxCaracteresLeidos ]]] )

---

```
1 <?php
2 $cadena = file_get_contents("poema.txt",false,null,-20);
3 print $cadena;
4 ?>
```

**Listado 13.11.1.** leeArchivo.php

*www.pacoarce.com*

## Cuestionario:

**1) En PHP, la función unlink() se emplea para**

- 1) Crear un fichero normal
- 2) Crear un enlace simbólico entre dos ficheros
- 3) Contar el número de segmentos que componen un fichero
- 4) Eliminar un archivo

**2) En PHP, ¿cuál es la forma correcta de abrir el archivo "datos.txt" para sólo lectura?**

- 1) open("datos.txt");
- 2) fopen("datos.txt", "r");
- 3) open("datos.txt", "read");
- 4) fopen("datos.txt", "r+");

**3) En PHP, para detectar el final de archivo, se emplea la función**

- 1) end()
- 2) eof()
- 3) feof()
- 4) file\_end()

**4) En PHP, para leer un número de bytes de un fichero abierto se emplea:**

- 1) fgets()
- 2) fread()
- 3) fscanf()
- 4) ftell()

**5) En PHP, para leer completamente un fichero en un array se emplea**

- 1) fgets()
- 2) file()
- 3) file\_get\_contents()
- 4) Las anteriores respuestas no son correctas

**6) En PHP, cuando se abre un archivo, ¿qué opción abrirá el archivo para sólo escritura y colocará el puntero al final del fichero?**

- 1) r+
- 2) w
- 3) w+
- 4) a



**7) En PHP, ¿qué función lee un archivo abierto línea a línea?**

- 1) fgets()
- 2) fread()
- 3) fscanf()
- 4) Las anteriores respuestas no son correctas

**8) La función que nos permite escribir en un archivo abierto para escritura es:**

- 1) fgets()
- 2) fread()
- 3) fwrite()
- 4) print\_file()

**9) La función que nos permite evitar que varios usuarios escriban sobre un archivo al mismo tiempo es:**

- 1) protected()
- 2) flock()
- 3) private()
- 4) banned()

**10) La función que verifica si un archivo no se encuentra bloqueado es:**

- 1) is\_free()
- 2) is\_available()
- 3) is\_append()
- 4) is\_writable()

**11) La función que libera un archivo protegido contra escritura es:**

- 1) free()
- 2) available()
- 3) flock()
- 4) is\_writable()

**12) Para mover un archivo de una carpeta a otra, dentro del servidor es:**

- 1) rename()
- 2) move()
- 3) copy()
- 4) file\_move()

**13) Por medio de que función podemos leer un archivo con el formato CVS:**

- 1) cvs()
- 2) fgetcsv()
- 3) read()
- 4) No se puede leer una archivo CVS en PHP.

**14) Por medio de que función podemos tomar un arreglo y lo guardarlo como formato CSV:**

- 1) putCSVs()
- 2) cvs()
- 3) copyFile()
- 4) Ninguna de las anteriores.



*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 14: Crear, modificar y eliminar las Galletas o cookies

**Objetivo:** El alumno aprenderá a crear, modificar y eliminar las galletas o cookies.

#### Introducción:

Las **cookies** o galletas son un medio tradicional de guardar una cadena en la computadora del cliente. Esta cadena debe ser no muy grande y no debe guardar datos confidenciales como números de tarjetas de créditos o claves de acceso importantes. En esta sección estudiaremos como crear, acceder y destruir las **cookies** desde **PHP**.

- Crear una cookie
- Leer una cookie
- Borrar una cookie
- Crear un contador de visitas con cookies .
- Almacenar datos de configuración del usuario con cookies.
- Almacenar los datos del usuario en un login con cookies

[www.pacoarce.com](http://www.pacoarce.com)

## 14.1. Crear una cookie

Para crear una galleta desde una rutina de PHP se usa el comando "setcookie".

La sintaxis básica para este comando es:

**setcookie( nombre\_de\_la\_galleta, valor\_de\_la\_galleta, expiración, ruta );**

Las galletas tienen esta información:

**Nombre de la galleta:** un string que se adhiera a las reglas de los nombres de las variables. El nombre de la galleta va a crear una variable global con el mismo nombre.

**Valor de la galleta:** un string que contenga los datos a guardar. Es común usar un caracter "separador" si se desea guardar datos con una estructura.

**Expiración de la galleta:** toda galleta debe tener un tiempo de vigencia, medido en segundos. Después de este tiempo, la galleta no es válida. Si se pasa un valor 0 (cero), la galleta solo es válida mientras el usuario tiene abierto su navegador.

**Ruta:** define el "área" del sitio donde esta galleta será mandada al servidor, cada vez que se accesa una página dentro de esta ruta. Si se desea mandar la galleta para todo el sitio, se usa "/".

**Dominio:** define los dominios a los cuales se va a mandar la galleta. Como esto restringe el mandar la galleta sólo desde ciertos dominios, generalmente se omite al crear galletas de uso "universal". Si no se desea especificar un dominio, se usa unstring vacío ("") o se omite el parámetro.

**Transmisión segura:** si se especifica el valor 1 (tipo entero, NO string), sólo se transmite la galleta si la conexión es segura (https). De lo contrario hay que especificar el valor 0 (cero). Se puede omitir el parámetro.

---

```
1  <?php
2  $nombre = "datos";
3  $valor = "pacoarce@hotmail.com|Francisco|Arce|20180411";
4  $fecha = time() + (60*60*24);
5  setcookie($nombre, $valor, $fecha);
6  ?>
7  <!DOCTYPE html>
8  <html>
9  <head>
10   <title>Galletas | Crear</title>
11   <meta charset="utf-8">
12 </head>
13 <body>
14
15 </body>
16 </html>
```

**Listado 14.1.1.** PHP7-1401-GalletasCrear.php



*[www.pacoarce.com](http://www.pacoarce.com)*

## 14.2. Leer una cookie

Para leer una o varias galletas guardadas, es necesario que el usuario ejecute una rutina de **PHP**. Esto hace que todas las galletas activas generen variables globales con el mismo nombre de la galleta. Si la galleta ya no es válida, no se crea la variable.

```
1  <?php
2  $nombre = "datos";
3  $valor = "pacoarce@hotmail.com|Francisco|Arce|20180411";
4  $fecha = time() + (60*60*24);
5  setcookie($nombre, $valor, $fecha);
6  ?>
7  <!DOCTYPE html>
8  <html>
9  <head>
10   <title>Galletas | Crear</title>
11   <meta charset="utf-8">
12 </head>
13 <body>
14
15 </body>
16 </html>
```

**Listado 14.2.1.** PHP7-1401-GalletasCrear.php

[www.pacoarce.com](http://www.pacoarce.com)

## 14.3. Borrar una cookie

Para borrar una galleta, la manera más segura es crear la galleta con el mismo nombre, con un valor "vacío" y una fecha de **expiración** "vencida".

```
1  <?php
2  $nombre = "datos";
3  $valor = "pacoarce@hotmail.com|Francisco|Arce|20180411";
4  $fecha = time() - 1;
5  setcookie($nombre, $valor, $fecha);
6  ?>
7  <!DOCTYPE html>
8  <html>
9  <head>
10   <title>Galletas | Crear</title>
11   <meta charset="utf-8">
12 </head>
13 <body>
14 <?php
15 if (isset($_COOKIE["datos"])) {
16   $datos = $_COOKIE["datos"];
17   $datos_array = explode("|", $datos);
18   $email = $datos_array[0];
19   $nombre = $datos_array[1];
20   $apellido = $datos_array[2];
21   $fecha = $datos_array[3];
22   print "<p>Hola <b>".$nombre." ".$apellido."</b>, recibimos tu mensaje del correo
23   ".$email." en la fecha ".$fecha." de en breve nos comunicaremos contigo.</p>";
24 } else {
25   # code...
26 }
27
28
29 ?>
30 </body>
31 </html>
```

Listado 14.3.1. PHP7-1403-GalletasBorrar.php

www.pacoarce.com

## 14.4. Crear un contador de visitas con cookies

En esta clase crearemos un **control de visitas** del usuario a una página, y almacenaremos la fecha y hora de su última visita. Si es la primera vez que nos visita (no está creada la *cookie*), mandaremos a pantalla un mensaje correspondiente.

```
1  <?php
2  $fecha = date("d/m/Y H:i:s");
3
4  if (isset($_COOKIE["visita"])) {
5  $data = explode("|", $_COOKIE["visita"]);
6  print "Hola de nuevo, es tu visita no. " .(++$data[0])." y tu anterior visita fue en
7  la fecha ".$data[1].", bienvenid@"."  
>";
8  setcookie("visita",$data[0]."|".$fecha);
9
10 } else {
11 print "Hola, es la primera vez que ingresa, bienvenid@";
12 setcookie("visita","1|".$fecha);
13 }
14 ?>
```

**Listado 14.4.1.** contadorVisitas.php

[www.pacoarce.com](http://www.pacoarce.com)



## 14.5. Almacenar datos de configuración del usuario con cookies.

En esta clase almacenaremos en la computadora del usuario por medio de *cookies* **parámetros de configuración**. En este ejemplo almacenaremos la imagen de fondo de pantalla que seleccione el usuario.

```
1  <?php
2  $imagen_array = array("crisantemo.jpg","desierto.jpg","faro.jpg");
3  if (isset($_POST["imagen"])) {
4  $numImagen = $_POST["imagen"];
5  setcookie("imagen",$numImagen,time()+60*60*24*7);
6  $imagen = $imagen_array[$numImagen];
7  } else {
8  if(isset($_COOKIE["imagen"])){
9  $imagen = $imagen_array[$_COOKIE["imagen"]];
10 } else {
11 $imagen = "";
12 }
13 }
14 ?>
15 <!DOCTYPE html>
16 <html>
17 <head>
18 <title>Imágenes con cookies</title>
19 </head>
20 <body>
21 <?php
22 if($imagen!=""){
23 print "style='background:url(img/".$imagen."); background-size:cover;';";
24 }
25 ?>
26 >
27 <form method="POST" action="<?php print $_SERVER['PHP_SELF']; ?>">
28 <label for="imagen">Selecciona una imagen:</label>
29 <select name="imagen" id="imagen">
30 <option value="0">Crisantemos</option>
31 <option value="1">Desierto</option>
32 <option value="2">Faro</option>
33 </select>
34 <input type="submit" value="Cambiar foto de fondo">
35 </form>
36 </body>
37 </html>
```

**Listado 14.5.1.** imagenFondo.php

## 14.6. Almacenar los datos del usuario en un login con cookies

En esta clase almacenaremos los **datos de entrada** del usuario al sistema (*login*) por medio de *cookies* a solicitud de una casilla de verificación (*checkbox*).

```
1  <?php
2  $recordar = "";
3  if (isset($_POST["usuario"])) {
4  $usuario = $_POST["usuario"] ?? "";
5  $clave = $_POST["clave"] ?? "";
6  $recordar = $_POST["recordar"] ?? "";
7  if($recordar=="on"){
8  setcookie("datos",$usuario."|".$clave, time()+60*60*24*7);
9  } else {
10 $usuario = "";
11 $clave = "";
12 $recordar = "";
13 }
14 }
15 if (isset($_COOKIE["datos"])) {
16 $data = explode("|",$_COOKIE["datos"]);
17 $usuario = $data[0];
18 $clave = $data[1];
19 $recordar = "on";
20 }
21 ?>
22 <!DOCTYPE html>
23 <html>
24 <head>
25 <title>Login</title>
26 </head>
27 <body>
28 <form action="<?php print $_SERVER['PHP_SELF']; ?>" method="POST">
29 <label for="usuario">Usuario:</label>
30 <input type="text" name="usuario"
31 <?php
32 if ($recordar=="on") {
33 print "value='".$usuario.'"';
34 }
35 ?>
36 >
37 <label for="clave">Clave:</label>
38 <input type="password" name="clave"
39 <?php
40 if ($recordar=="on") {
41 print "value='".$clave.'"';
42 }
43 ?>
```

```
44
45 >
46 <input type="checkbox" name="recordar"
47 <?php
48 if ($recordar=="on") {
49 print "checked";
50 }
51 ?>
52 >
53 <label for="recordar">Recordar</label>
54 <input type="submit" value="Entrar">
55 </form>
56
57 </body>
58 </html>
```

**Listado 14.6.1.** recordarDatos.php

*www.pacoarce.com*

## Cuestionario:

### 1) Cómo se crea una cookie en PHP?

- 1) Con document.cookie
- 2) Con \$\_COOKIE
- 3) Con setcookie()
- 4) Con doCookie()

### 2) El lugar para escribir una galleta o cookie en nuestro programa PHP es:

- 1) Al inicio, antes que cualquier etiqueta HTML.
- 2) Después de la etiqueta head
- 3) Dentro de la etiqueta body
- 4) En cualquier parte del programa

### 3) La instrucción para crear una galleta con PHP es:

- 1) newCookie
- 2) cookie
- 3) setcookie()
- 4) createCookie

### 4) Con las galletas creadas en PHP, a diferencia de las galletas creadas en JavaScript, podemos almacenar información ilimitada en el disco duro del cliente:

- 1) Verdadero
- 2) Falso

### 5) Para leer una galleta, utilizamos la variable global:

- 1) \$\_COOKIE\_NEW[]
- 2) \$\_GET[]
- 3) \$\_COOKIE\_READ
- 4) \$\_COOKIE[]

### 6) Para determinar el tiempo de vida de una COOKIE en el disco duro del navegador del usuario, debemos partir de qué fecha:

- 1) El valor cero.
- 2) La hora cero de UNIX.
- 3) La fecha del servidor.
- 4) La fecha de la computadora del usuario.

**7) Si en el parámetro de expiración del tiempo de una galleta colocamos el valor cero (0), la galleta:**

- 1) Será válida mientras el navegador esté abierto.
- 2) El navegador marca un error.
- 3) Tendrá una vigencia ilimitada.
- 4) La galleta será guardada en el disco del servidor, en lugar del disco del cliente.

**8) Una vez que expira la vigencia de una galleta, el navegador procede a:**

- 1) Mandar a la galleta a un archivo histórico por si el usuario desea volver a utilizar.
- 2) Mandar un mensaje para actualizar la galleta.
- 3) Eliminar la galleta.
- 4) Dar un día más de vigencia a la galleta en forma automática.

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 15: Conexión con MySQL

**Objetivo:** El alumno aprenderá a conectarse a MySQL, así como a insertar, modificar, seleccionar y eliminar datos de las tablas de la misma.

#### Introducción:

Una de las razones por lo que es tan popular PHP es por su conexión con MySQL. En esta sección estudiaremos la forma de conectarnos a MySQL, que prácticamente han crecido juntas. Los temas que revisaremos son:

- Crear la base de datos y una tabla de prueba
- Conectar una base de datos MySQL con PHP
- Las sentencias include y require
- Insertar un registro en la base de datos
- Borrar un registro de una tabla
- Modificar un registro de la tabla
- Seleccionar archivos de una tabla, el comando SELECT
- Establecer el conjunto de caracteres para la base de datos
- Recuperar el último "id" insertado
- Leer un archivo CSV e insertarlo en una tabla MySQL
- Verificar que los datos no se dupliquen.
- Bajar la información de MySQL a CSV .

## 15.1. Crear la base de datos y una tabla de prueba

Las rutinas de **PHP** pueden acceder varias bases de datos, incluyendo las basadas en SQL. La más popular es la base de datos **MySQL**, y es la que se va a usar para estos ejemplos básicos.

```
1  -- phpMyAdmin SQL Dump
2  -- version 4.4.15.5
3  -- http://www.phpmyadmin.net
4  --
5  -- Host: localhost:3308
6  -- Generation Time: Apr 13, 2018 at 03:26 AM
7  -- Server version: 5.5.49-log
8  -- PHP Version: 7.0.9
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8mb4 */;
18
19 --
20 -- Database: `quiz`
21 --
22
23 -- -----
24
25 --
26 -- Table structure for table `preguntas`
27 --
28
29 CREATE TABLE IF NOT EXISTS `preguntas` (
30   `id` int(11) NOT NULL,
31   `pregunta` varchar(200) COLLATE utf8_spanish_ci NOT NULL,
32   `op1` varchar(200) COLLATE utf8_spanish_ci NOT NULL,
33   `op2` varchar(200) COLLATE utf8_spanish_ci NOT NULL,
34   `op3` varchar(200) COLLATE utf8_spanish_ci NOT NULL,
35   `op4` varchar(200) COLLATE utf8_spanish_ci NOT NULL,
36   `buena` char(1) COLLATE utf8_spanish_ci NOT NULL,
37   `examen` char(5) COLLATE utf8_spanish_ci NOT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
39
40 --
41 -- Indexes for dumped tables
42 --
43
44 --
```

```
45 -- Indexes for table `preguntas`
46 --
47 ALTER TABLE `preguntas`
48 ADD PRIMARY KEY (`id`);
49
50 --
51 -- AUTO_INCREMENT for dumped tables
52 --
53
54 --
55 -- AUTO_INCREMENT for table `preguntas`
56 --
57 ALTER TABLE `preguntas`
58 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
59 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
60 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
61 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
62
```

**Listado 15.1.1.** quiz.sql

*www.pacoarce.com*



## 15.2. Conectar una base de datos MySQL con PHP

Por medio de `mysqli_connect()` conectamos a la base de datos y a la vez seleccionamos una base de datos:

**`mysqli_connect(host,usuario,clave,db,puerto,socket);`**

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>MySQL | Connect</title>
5    <meta charset="utf-8">
6    <?php
7      $host = "localhost";
8      $usuario = "root";
9      $clave = "root";
10     $db = "quiz";
11     $puerto = "3308";
12     //
13     $conn = mysqli_connect($host, $usuario, $clave, $db, $puerto) or die("Error al
conectar la base de datos");
14     print "Conexión exitosa";
15     ?>
16 </head>
17 <body>
18
19 </body>
20 </html>
```

**Listado 15.2.1.** MySQLConnect.php

[www.pacoarce.com](http://www.pacoarce.com)

## 15.3. Las sentencias include y require

- Las sentencias *include()* y *require()* incluyen y evalúan el archivo especificado.
- Contamos con una ruta predefinida para las funciones *include*, *require*, *fopen()*, *file()*, *readfile()* y *file\_get\_contents()* donde buscarán sus archivo, la cual es *include\_path*.
- Podemos cambiar esta ruta con la sentencia:

`string set_include_path ( string $nuevo_include_path );`

- La diferencia entre *include()* y *require()* es que la primera, si no encuentra el archivo señalado manda un mensaje de “warning” y continúe, y la sentencia *require()* manda un mensaje de error fatal y termina el script.
- También se tienen las sentencias *include\_once()* y *require\_once()* donde se asegura que sólo se cargue el archivo una vez.

---

```
1  <?php require_once("php/conn.php"); ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>MySQL | Include y require</title>
6    <meta charset="utf-8">
7
8  </head>
9  <body>
10   <?php print "Se accedió correctamente a la base de datos."; ?>
11 </body>
12 </html>
```

**Listado 15.3.1.** MySQLRequireInclude.php

*www.pacoarce.com*

## 15.4. Insertar un registro en la base de datos

Por medio de la función `mysqli_query()` ejecuta un query:

*`mysqli_query(conexión,query);`*

Por medio del comando INSERT añadimos los registros a la tabla.

```
1  <?php require_once("php/conn.php"); ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>MySQL | Insertar un registro</title>
6    <meta charset="utf-8">
7    <?php
8      $pregunta = "¿Cuál es la capital de España?";
9      $op1 = "Barcelona";
10     $op2 = "Madrid";
11     $op3 = "Caracas";
12     $op4 = "Valladolid";
13     $res = "2";
14     $examen = "GEO01";
15     //armamos el query
16     $q = "INSERT INTO preguntas(id,pregunta,op1,op2,op3,op4,buena,examen) ";
17     $q.= "VALUES(0,'" . $pregunta . "','" . $op1 . "','" . $op2 . "','" . $op3 . "','" . $op4 . "','" . $res . "','" . $examen . "')";
18     //print $q;
19     $r = mysqli_query($conn,$q);
20     if($r){
21       print "<p>Se insertó correctamente el registro</p>";
22     } else {
23       print "<p>Error al insertar el registro</p>";
24     }
25   ?>
26 </head>
27 <body>
28
29 </body>
30 </html>
```

**Listado 15.4.1.** MySQLInsert.php

[www.pacoarce.com](http://www.pacoarce.com)

## 15.5. Borrar un registro de una tabla

- Por medio del comando DELETE podemos eliminar los registros de una tabla. Una vez borrados, no se pueden recuperar.
- Siempre hay que utilizar el comando WHERE, de lo contrario se borrarán todos los registros de la tabla.

DELETE FROM tabla

WHERE columna= valor

```
1  <?php require_once("php/conn.php"); ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>MySQL | Delete </title>
6    <meta charset="utf-8">
7    <?php
8      $pregunta = "¿Cuál es la capital de Argentina?";
9      $op1 = "Buenos Aires";
10     $op2 = "Lima";
11     $op3 = "Caracas";
12     $op4 = "Los Ángeles";
13     $res = "1";
14     $examen = "GEO01";
15     //armamos el query
16     $q = "INSERT INTO preguntas(id,pregunta,op1,op2,op3,op4,buena,examen) ";
17     $q.= "VALUES(0,'" . $pregunta . "','" . $op1 . "','" . $op2 . "','" . $op3 . "','" . $op4 . "','" . $res . "','" . $examen . "')";
18     //print $q;
19     if(mysqli_query($conn,$q)){
20       print "<p>Se insertó correctamente el registro</p>";
21     } else {
22       print "<p>Error al insertar el registro</p>";
23     }
24     //
25     $q = "DELETE FROM preguntas";
26     if(mysqli_query($conn,$q)){
27       print "<p>Se eliminó correctamente el registro</p>";
28     } else {
29       print "<p>Error al eliminar el registro</p>";
30     }
31     ?>
32 </head>
33 <body>
34
35 </body>
36 </html>
```

**Listado 15.5.1.** MySQLDelete.php

## 15.6. Modificar un registro de la tabla

Por medio de la instrucción UPDATE podemos actualizar un registro ya existente.

**UPDATE** tabla

**SET** columna1 = valor1, columna2 = valor2, ...

**WHERE** condición;

```
1  <?php require_once("php/conn.php"); ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>MySQL | Update </title>
6    <meta charset="utf-8">
7    <?php
8      $pregunta = "¿Cuál es la capital de Perú?";
9      $op1 = "Buenos Aires";
10     $op2 = "Lima";
11     $op3 = "Bogotá";
12     $op4 = "Los Ángeles";
13     $res = "2";
14     $examen = "GEO01";
15     //armamos el query
16     $q = "INSERT INTO preguntas(id,pregunta,op1,op2,op3,op4,buena,examen) ";
17     $q.= "VALUES(0,'" . $pregunta . "','" . $op1 . "','" . $op2 . "','" . $op3 . "','" . $op4 . "','" . $res . "','" . $examen . "')";
18     //print $q;
19     if(mysqli_query($conn,$q)){
20       print "<p>Se insertó correctamente el registro</p>";
21     } else {
22       print "<p>Error al insertar el registro</p>";
23     }
24     //
25     $q = "UPDATE preguntas SET op4='Nueva York' WHERE id=6";
26     if(mysqli_query($conn,$q)){
27       print "<p>Se modificó correctamente el registro</p>";
28     } else {
29       print "<p>Error al modificar el registro</p>";
30     }
31     ?>
32 </head>
33 <body>
34
35 </body>
36 </html>
```

**Listado 15.6.1.** PHP7-1506-MySQLUpdate.php

## 15.7. Seleccionar archivos de una tabla, el comando SELECT

- La sentencia SELECT nos permitirá seleccionar los registros de una tabla. En PHP los registros seleccionados se almacenarán en un "recurso".
- La sintaxis de la sentencia SELECT es:

SELECT \* FROM nombre\_tabla;

- El cual selecciona todas las columnas y todos los registros.
- Para hacer una selección más precisa, utilizamos:

SELECT columna1, columna2,... FROM nombre\_tabla  
WHERE una\_columa=valor

---

```
1  <?php require_once("php/conn.php"); ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>MySQL | Select </title>
6    <meta charset="utf-8">
7    <?php
8      $q = "SELECT * FROM preguntas";
9      $r = mysqli_query($conn,$q);
10     $n = mysqli_num_rows($r);
11   ?>
12 </head>
13 <body>
14   <?php
15     print "<h2>Número de registros: ".$n."</h2>";
16     print "<table border='1'>";
17     while ($data = mysqli_fetch_assoc($r)) {
18       print "<tr>";
19       print "<td>".$data["pregunta"]."</td>";
20       print "<td>".$data["op1"]."</td>";
21       print "<td>".$data["op2"]."</td>";
22       print "<td>".$data["op3"]."</td>";
23       print "<td>".$data["op4"]."</td>";
24       print "<td>".$data["buena"]."</td>";
25       print "<td>".$data["examen"]."</td>";
26       print "</tr>";
27     }
28     print "</table>";
29   ?>
30 </body>
31 </html>
```

**Listado 15.7.1.** MySQLSelect.php

## 15.8. Establecer el conjunto de caracteres para la base de datos

**Versiones: (PHP 5 >= 5.0.5, PHP 7)**

**bool mysqli\_set\_charset ( mysqli \$link , string \$charset );**

**bool mysqli::set\_charset ( string \$charset )**

También tenemos la versión orientado a objetos:

Podemos establecer el conjunto de caracteres desde y hacia la base de datos con la instrucción:

No se recomienda utilizar el comando con:

**query("SET NAMES 'utf8'")**

---

```
1  <?php require_once("php/conn.php"); ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>MySQL | Select </title>
6    <meta charset="utf-8">
7    <?php
8      $q = "SELECT * FROM preguntas";
9      $r = mysqli_query($conn,$q);
10     $n = mysqli_num_rows($r);
11   ?>
12 </head>
13 <body>
14   <?php
15     print "<h2>Número de registros: ".$n."</h2>";
16     print "<table border='1'>";
17     while ($data = mysqli_fetch_assoc($r)) {
18       print "<tr>";
19       print "<td>".$data["pregunta"]."</td>";
20       print "<td>".$data["op1"]."</td>";
21       print "<td>".$data["op2"]."</td>";
22       print "<td>".$data["op3"]."</td>";
23       print "<td>".$data["op4"]."</td>";
24       print "<td>".$data["buena"]."</td>";
25       print "<td>".$data["examen"]."</td>";
26       print "</tr>";
27     }
28     print "</table>";
29   ?>
30 </body>
31 </html>
```

**Listado 15.8.1. set\_charset.php**

## 15.9. Recuperar el último “id” insertado

(PHP 5, PHP 7)

**mysqli\_insert\_id** — Devuelve el último valor generado con atributo AUTO\_INCREMENT.

- También regresa el último “id” modificado con UPDATE.
- La columna no necesariamente debe llamarse “id”.
- Si no hay un campo AUTO\_INCREMENT regresará el valor de cero.

```
1  <?php
2  require "php/conn.php";
3
4  $archivo = fopen("prueba.csv","r");
5
6  $inicio = true;
7
8  while (!feof($archivo)) {
9
10 //Leemos el registro en formato CSV
11 $data = fgetcsv($archivo);
12
13 if ($inicio) {
14 $inicio = false;
15 continue;
16 }
17 $sql = "INSERT INTO preguntas VALUES(0,";
18 $sql.= "'".$data[1]."', ";
19 $sql.= "'".$data[2]."', ";
20 $sql.= "'".$data[3]."', ";
21 $sql.= "'".$data[4]."', ";
22 $sql.= "'".$data[5]."', ";
23 $sql.= $data[6].", ";
24 $sql.= "'".$data[7]."'");
25
26 //DML
27 if (mysqli_query($conn, $sql)) {
28 $id = mysqli_insert_id($conn);
29 print "<p>Se insertó correctamente el registro con el id ".$id."</p>";
30 } else {
31 print "<p>Error al insertar el registro</p>";
32 }
33 }
34 fclose($archivo);
35 ?>
```

**Listado 15.9.1.** MySQLArchivosCSV.php



## 15.10. Leer un archivo CSV e insertarlo en una tabla MySQL

Leemos el archivo CSV con `fgetcsv($archivo)`; y guardamos la información con `INSERT`.

```
1  <?php
2  require "php/conn.php";
3
4  $archivo = fopen("prueba.csv","r");
5
6  $inicio = true;
7
8  while (!feof($archivo)) {
9
10 //Leemos el registro en formato CSV
11 $data = fgetcsv($archivo);
12
13 if ($inicio) {
14 $inicio = false;
15 continue;
16 }
17 $sql = "INSERT INTO preguntas VALUES(0,";
18 $sql.= "'".$data[1]."', ";
19 $sql.= "'".$data[2]."', ";
20 $sql.= "'".$data[3]."', ";
21 $sql.= "'".$data[4]."', ";
22 $sql.= "'".$data[5]."', ";
23 $sql.= $data[6].", ";
24 $sql.= "'".$data[7]."'");
25
26 //DML
27 if (mysqli_query($conn, $sql)) {
28 $id = mysqli_insert_id($conn);
29 print "<p>Se insertó correctamente el registro con el id ".$id."</p>";
30 } else {
31 print "<p>Error al insertar el registro</p>";
32 }
33 }
34 fclose($archivo);
35 ?>
```

**Listado 15.10.1.** MySQLarchivosCSV.php

## 15.11. Verificar que los datos no se dupliquen.

Siempre debemos tener una llave única “natural” a parte de la llave primaria de la tabla (AUTO\_INCREMENT).

```
1  <?php
2  require "php/conn.php";
3
4  $archivo = fopen("prueba.csv","r");
5
6  $inicio = true;
7
8  while (!feof($archivo)) {
9
10 //Leemos el registro en formato CSV
11 $data = fgetcsv($archivo);
12
13 if ($inicio) {
14 $inicio = false;
15 continue;
16 }
17 $sql = "SELECT id FROM preguntas WHERE pregunta='".$data[1]."'";
18
19 //DQL Data Query Language
20 $r = mysqli_query($conn, $sql);
21 $num = mysqli_num_rows($r);
22
23 //Si el numero de renglones es cero => no existe la pregunta
24 if($num==0){
25 $sql = "INSERT INTO preguntas VALUES(0,";
26 $sql.= "'".$data[1]."', ";
27 $sql.= "'".$data[2]."', ";
28 $sql.= "'".$data[3]."', ";
29 $sql.= "'".$data[4]."', ";
30 $sql.= "'".$data[5]."', ";
31 $sql.= $data[6].", ";
32 $sql.= "'".$data[7]."'");
33
34 //DML
35 if (mysqli_query($conn, $sql)) {
36 $id = mysqli_insert_id($conn);
37 print "<p>Se insertó correctamente el registro con el id ".$id."</p>";
38 } else {
39 print "<p>Error al insertar el registro</p>";
40 }
41 } else {
42 print "<p>La pregunta está duplicada: ".$data[1]."</p>";
43 }
44
```

```
45 }  
46 fclose($archivo);  
47 ?>
```

**Listado 15.11.1.** MySQLArchivosCSV.php



*www.pacoarce.com*

## 15.12. Bajar la información de MySQL a CSV

Podemos bajar la información de una tabla a un archivo CSV para su manipulación fuera del sistema.

```
1  <?php
2  require "php/conn.php";
3
4  //DQL
5  $sql = "SELECT * FROM preguntas";
6  $r = mysqli_query($conn, $sql);
7  $num = mysqli_num_rows($r);
8
9  touch("data.csv");
10 $archivo = fopen("data.csv","w");
11
12 while ($data =mysqli_fetch_array($r,MYSQLI_NUM)) {
13 fputcsv($archivo, $data);
14 }
15 print "Se tranfirieron ".$num." registros";
16 fclose($archivo);
17 mysqli_close($conn);
18 ?>
```

**Listado 15.12.1.** MySQLcreaCSV.php

*www.pacoarce.com*

## Cuestionario:

**1) ¿Cuál de estas instrucciones se utiliza para realizar una consulta a una base de datos MySQL?**

- 1) mysqli\_query
- 2) mysqli\_access
- 3) mysqli\_db\_access

**2) Cuando deseamos establecer el valor de un campo de tipo timestamp en MySQL, utilizamos la constante:**

- 1) Por medio de CURRENT\_TIMESTAMP podemos recuperar la hora del servidor para ser insertado en un campo de la tabla.
- 2) TIME\_STAMP
- 3) CURRENT\_TIMESTAMP
- 4) CURRENT\_DATE

**3) PHP nos permite conectarnos a base de datos de MySQL utilizando métodos y/o objetos**

- 1) Verdadero
- 2) Falso

**4) Cuando deseamos incluir un archivo PHP externo en nuestra página PHP, NO utilizamos la siguiente instrucción:**

- 1) insert
- 2) include
- 3) include\_once
- 4) require

**5) PHP sólo puede trabajar con la base de datos de MySQL**

- 1) Verdadero
- 2) Falso

**6) ¿Cuál es la forma correcta de abrir una conexión con una base de datos MySQL?**

- 1) connect\_mysqli();
- 2) dbopen();
- 3) mysqli\_open();
- 4) mysqli\_connect();

**7) Cuando se incluye un archivo con include o requiere:**

- 1) El archivo incluido debe tener la extensión ".inc"
- 2) El archivo incluido debe tener únicamente código PHP
- 3) El archivo incluido debe estar situado en el mismo directorio que el archivo desde donde se incluye
- 4) Las anteriores respuestas no son correctas

**8) En PHP y MySQL, para liberar la memoria utilizada por un resultado se emplea la función:**

- 1) mysqli\_clean()
- 2) mysqli\_free()
- 3) mysqli\_free\_result()
- 4) mysqli\_garbage\_collect()

**9) En PHP y MySQL, la función mysqli\_error():**

- 1) Activa la gestión de los mensajes de error
- 2) Reinicia la cola de mensajes de error
- 3) Devuelve el código de error de la operación anterior
- 4) Devuelve el texto del mensaje de error de la operación anterior

**10) ¿Cuál es la forma correcta de incluir un archivo?**

- 1) include "archivo";
- 2) include("archivo");
- 3) require "archivo";
- 4) Todas las respuestas son correctas

**11) ¿Qué funciones se emplean para abrir y cerrar una conexión en PHP?**

- 1) mysqli\_open() y mysqli\_close()
- 2) mysqli\_open() y mysqli\_disconnect()
- 3) mysqli\_connect() y mysqli\_close()
- 4) mysqli\_connect() y mysqli\_disconnect()

**12) En PHP y MySQL, para obtener el número de filas devueltas por un resultado, se utiliza la función:**

- 1) mysqli\_affected\_rows()
- 2) mysqli\_list\_fields()
- 3) mysqli\_num\_rows()
- 4) Las anteriores respuestas no son correctas

**13) ¿Qué función de PHP devuelve el resultado exclusivamente como un arreglo asociativo?**

- 1) mysqli\_fetch\_array()
- 2) mysqli\_fetch\_assoc()
- 3) mysqli\_fetch\_row()
- 4) Las anteriores respuestas no son correctas

**14) En PHP y MySQL, para obtener el número de filas que han sido borradas con una sentencia DELETE, se utiliza la función**

- 1) mysqli\_affected\_rows()
- 2) mysqli\_list\_fields()
- 3) mysqli\_num\_rows()
- 4) Las anteriores respuestas no son correctas

**15) Una tecnología hermana de PHP en el manejo de base de datos es:**

- 1) MySQL
- 2) ADABAS
- 3) ORACLE
- 4) SQLserver

**16) phpMyAdmin es la única forma de administrar la base de datos MySQL:**

- 1) Verdadero
- 2) Falso

**17) En una tabla de MySQL pueden existir muchos campos de tipo auto increment:**

- 1) Verdadero
- 2) Falso

**18) Señala cual tipo de campo tiene una longitud variable y su longitud debe ser indicada por el usuario.**

- 1) varchar
- 2) date
- 3) int
- 4) double

# Todo PHP7

## De novato a experto

### Capítulo 16: Manejo de imágenes con PHP

**Objetivo:** El alumno creará una galería de fotos tomadas desde una carpeta del servidor, optimizar las imágenes así como aplicar diferentes filtros a las mismas y subir una imagen de la máquina del usuario.

#### Introducción:

PHP, por medio de la librería **GD**, es una poderosa herramienta para procesar imágenes. Estudiaremos cómo subirlas de la máquina del usuario (cliente) a una carpeta en el servidor, así como subir una imagen a un campo BLOB en la base de datos, visualizarla y bajarla a un archivo jpg.

- Galería de fotos desde una carpeta
- Tabla de imágenes tomándola de una carpeta
- Advertencia para una imagen
- Borrar físicamente la imagen
- Parámetros para la optimización de una imagen
- Optimiza la imagen por un porcentaje
- Optimizar una imagen por su ancho en forma proporcional
- El uso de filtros a una imagen
- Subir un archivo desde la computadora al servidor
- Validar el tipo de archivo subido
- Subir una imagen de PHP a la base de datos MySQL
- Guardar la imagen y desplegar una tabla
- Desplegar una columna binaria de MySQL desde PHP
- de BLOB a un archivo de imagen

[www.pacoarce.com](http://www.pacoarce.com)



## 16.1. Galería de fotos desde una carpeta

Una operación básica será leer de un directorio los datos de los archivos que se encuentran en él, y hacer algo con ellos. Con esto podremos hacer un sencillo manejador de archivos, pero en este caso desplegaremos las imágenes que encontremos. Este sería el paso para realizar un album de fotos dinámicos, así como un manejador de imágenes que nos ayudaría en nuestra aplicación.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Imagenes | Leer la carpeta</title>
5  </head>
6  <body>
7  <?php
8    /*****
9    F U N C I O N E S
10   *****/
11   function validaFoto($imagen){
12     $patron = "%\.(gif|jpe?g|png)$%i";
13     // Ejemplo de visualización del resultado
14     $bandera = preg_match($patron, $imagen) == 1 ? true : false;
15     return $bandera;
16   }
17   $dir = opendir("fotos");
18   $ren = 4;
19   $i = 0;
20   if ($dir) {
21     print "<table border='1'>";
22     print "<tr>";
23     while ($foto=readdir($dir)) {
24       if ($foto != "." && $foto != ".." && validaFoto($foto)) {
25         if ($i == $ren) {
26           $i=0;
27           print "</tr>";
28           print "<tr>";
29         }
30         $i++;
31         print "<td>";
32         print "<img src='fotos/' . $foto . "' width='100' height='100' />";
33         print "</td>";
34       }
35     }
36     print "</tr>";
37     print "</table>";
38   } else {
39     print "Error al abrir la carpeta fotos";
40   }
```

```
41
42
43
44 ?>
45 </body>
46 </html>
```

**Listado 16.1.1.** PHP7-1601-ImagenesCarpeta.php



*www.pacoarce.com*

## 16.2. Tabla de imágenes tomándola de una carpeta

- En este clase iniciaremos con el armado de una tabla que nos permitirá visualizar y seleccionar si deseamos borrar u optimizar las imágenes.
- Una tarea muy importante cuando hacemos páginas auto-administrables, es que podamos darle mantenimiento a las imágenes del sitio, y ello consiste en poder subir y borrar archivos, así como optimizarlos.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Imagenes | Tabla de imágenes</title>
5  </head>
6  <body>
7  <?php
8    /*****
9    F U N C I O N E S
10   *****/
11   function validaFoto($imagen){
12     $patron = "%\.(gif|jpe?g|png)$%i";
13     // Ejemplo de visualización del resultado
14     $bandera = preg_match($patron, $imagen) == 1 ? true : false;
15     return $bandera;
16   }
17   /*****
18   V A R I A B L E S
19   *****/
20   $dir = opendir("fotos");
21   $ren = 1;
22   $i = 0;
23   /*****
24   I N I C I O
25   *****/
26   if ($dir) {
27     print "<table border='1'>";
28     print "<tr>";
29     print "<th>Imagen</th>";
30     print "<th>Archivo</th>";
31     print "<th>Alto</th>";
32     print "<th>Ancho</th>";
33     print "<th>Tamaño kb</th>";
34     print "<th>Borrar</th>";
35     print "<th>Optimizar</th>";
36     print "</tr>";
37     print "<tr>";
38     while ($foto=readdir($dir)) {
39       if ($foto != "." && $foto != ".." && validaFoto($foto)) {
40         if ($i == $ren) {
41           $i=0;
```

```
42     print "</tr>";
43     print "<tr>";
44 }
45 $i++;
46 $archivo = "fotos/".$foto;
47 $info = getimagesize($archivo);
48 print "<td>";
49 print "<img src='fotos/".$foto."' width='100' />";
50 print "</td>";
51 print "<td>";
52 print $foto;
53 print "</td>";
54 print "<td>";
55 print $info[0];
56 print "</td>";
57 print "<td>";
58 print $info[1];
59 print "</td>";
60 print "<td>";
61 print number_format(filesize($archivo)/1024,2);
62 print "</td>";
63 print "<td>";
64 print "Borrar";
65 print "</td>";
66 print "<td>";
67 print "Optimizar";
68 print "</td>";
69 }
70 }
71 print "</tr>";
72 print "</table>";
73 } else {
74     print "Error al abrir la carpeta fotos";
75 }
76
77
78
79 ?>
80 </body>
81 </html>
```

**Listado 16.2.1.** ImagenesTabla.php

www.pacoarce.com

## 16.3. Advertencia para una imagen

Borra un archivo con PHP es definitivo, por lo cual es una buena práctica el generar una pantalla de advertencia previa. En este video desarrollamos una pantalla que advierte al usuario.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Imagenes | Borrar</title>
5  </head>
6  <body>
7  <?php
8    /*****
9    F U N C I O N E S
10   *****/
11   function validaFoto($imagen){
12     $patron = "%\.(gif|jpe?g|png)$%i";
13     // Ejemplo de visualización del resultado
14     $bandera = preg_match($patron, $imagen) == 1 ? true : false;
15     return $bandera;
16   }
17   /*****
18   V A R I A B L E S
19   *****/
20   $dir = opendir("fotos");
21   $ren = 1;
22   $i = 0;
23   /*****
24   I N I C I O
25   *****/
26   if ($dir) {
27     print "<table border='1'>";
28     print "<tr>";
29     print "<th>Imagen</th>";
30     print "<th>Archivo</th>";
31     print "<th>Alto</th>";
32     print "<th>Ancho</th>";
33     print "<th>Tamaño kb</th>";
34     print "<th>Borrar</th>";
35     print "<th>Optimizar</th>";
36     print "</tr>";
37     print "<tr>";
38     while ($foto=readdir($dir)) {
39       if ($foto != "." && $foto != ".." && validaFoto($foto)) {
40         if ($i == $ren) {
41           $i=0;
42           print "</tr>";
43           print "<tr>";
```

```
44     }
45     $i++;
46     $archivo = "fotos/".$foto;
47     $info = getimagesize($archivo);
48     print "<td>";
49     print "<img src='fotos/".$foto."' width='100' />";
50     print "</td>";
51     print "<td>";
52     print $foto;
53     print "</td>";
54     print "<td>";
55     print $info[0];
56     print "</td>";
57     print "<td>";
58     print $info[1];
59     print "</td>";
60     print "<td>";
61     print number_format(filesize($archivo)/1024,2);
62     print "</td>";
63     print "<td>";
64     print "<a href='PHP7-1603-borrarFoto.php?foto=".$foto."'>Borrar</a>";
65     print "</td>";
66     print "<td>";
67     print "Optimizar";
68     print "</td>";
69 }
70 }
71 print "</tr>";
72 print "</table>";
73 } else {
74     print "Error al abrir la carpeta fotos";
75 }
76
77
78
79 ?>
80 </body>
81 </html>
```

## Listado 16.3.1. imagenesAdvertencia.php

```
1 <?php
2 if (isset($_GET["foto"])) {
3     $foto = $_GET["foto"];
4 } else {
5     header("location:PHP7-1603-ImagenesAdvertencia.php");
6 }
7 ?>
8 <!DOCTYPE html>
9 <html>
10 <head>
11     <title>Imagenes | Advertencia Borrar</title>
```

```
12 <script>
13     function regresar() {
14         window.open("PHP7-1603-ImagenesAdvertencia.php", "_self");
15         return false;
16     }
17     function borrar(foto) {
18         window.open("", "_self");
19         return false;
20     }
21
22 </script>
23 </head>
24 <body>
25 <?php
26     print "<table border='1'>";
27     print "<tr>";
28     print "<th>Imagen</th>";
29     print "<th>Archivo</th>";
30     print "<th>Alto</th>";
31     print "<th>Ancho</th>";
32     print "<th>Tamaño kb</th>";
33     print "</tr>";
34     print "<tr>";
35     $archivo = "fotos/".$foto;
36     $info = getimagesize($archivo);
37     print "<td>";
38     print "<img src='fotos/".$foto."' width='300' />";
39     print "</td>";
40     print "<td>";
41     print $foto;
42     print "</td>";
43     print "<td>";
44     print $info[0];
45     print "</td>";
46     print "<td>";
47     print $info[1];
48     print "</td>";
49     print "<td>";
50     print number_format(filesize($archivo)/1024,2);
51     print "</td>";
52     print "</tr>";
53     print "</table>";
54 ?>
55 <form>
56     <input type="button" value="Borrar" onclick="borrarImagen();">
57     <input type="button" value="Regresar" onclick="regresar();">
58 </form>
59 </body>
60 </html>
```

**Listado 16.3.2.** borrarFoto.php

## 16.4. Borrar físicamente la imagen

Cuando borramos un archivo en PHP, ya sea una imagen o un archivo de datos, éste se borra en forma definitiva, por lo que es muy importante estar seguro de la acción.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Imágenes | Borrar</title>
5  </head>
6  <body>
7  <?php
8    /*****
9    F U N C I O N E S
10   *****/
11   function validaFoto($imagen){
12     $patron = "%\.(gif|jpe?g|png)$%i";
13     // Ejemplo de visualización del resultado
14     $bandera = preg_match($patron, $imagen) == 1 ? true : false;
15     return $bandera;
16   }
17   /*****
18   V A R I A B L E S
19   *****/
20   $dir = opendir("fotos");
21   $ren = 1;
22   $i = 0;
23   /*****
24   I N I C I O
25   *****/
26   if ($dir) {
27     print "<table border='1'>";
28     print "<tr>";
29     print "<th>Imagen</th>";
30     print "<th>Archivo</th>";
31     print "<th>Alto</th>";
32     print "<th>Ancho</th>";
33     print "<th>Tamaño kb</th>";
34     print "<th>Borrar</th>";
35     print "<th>Optimizar</th>";
36     print "</tr>";
37     print "<tr>";
38     while ($foto=readdir($dir)) {
39       if ($foto != "." && $foto != ".." && validaFoto($foto)) {
40         if ($i == $ren) {
41           $i=0;
42           print "</tr>";
43           print "<tr>";
44         }
```



```
45     $i++;
46     $archivo = "fotos/".$foto;
47     $info = getimagesize($archivo);
48     print "<td>";
49     print "<img src='fotos/".$foto."' width='100' />";
50     print "</td>";
51     print "<td>";
52     print $foto;
53     print "</td>";
54     print "<td>";
55     print $info[0];
56     print "</td>";
57     print "<td>";
58     print $info[1];
59     print "</td>";
60     print "<td>";
61     print number_format(filesize($archivo)/1024,2);
62     print "</td>";
63     print "<td>";
64     print "<a href='PHP7-1604-borrarFoto.php?foto=".$foto."'>Borrar</a>";
65     print "</td>";
66     print "<td>";
67     print "Optimizar";
68     print "</td>";
69 }
70 }
71 print "</tr>";
72 print "</table>";
73 } else {
74     print "Error al abrir la carpeta fotos";
75 }
76
77
78
79 ?>
80 </body>
81 </html>
```

## Listado 16.4.1. PHP7-1604-ImagenesAdvertencia.php

```
1 <?php
2 if ($_GET["foto"]) {
3     $archivo = "fotos/".$_GET["foto"];
4     touch($archivo);
5     unlink($archivo);
6 }
7 header("location:PHP7-1604-ImagenesAdvertencia.php");
8 ?>
```

## Listado 16.4.2. PHP7-1604-Borrarmagen.php

```
1 <?php
2 if (isset($_GET["foto"])) {
```

```
3  $foto = $_GET["foto"];
4  } else {
5  header("location:PHP7-1604-ImagenesAdvertencia.php");
6  }
7  ?>
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <title>Imagenes | Advertencia Borrar</title>
12 <script>
13 function regresar() {
14     window.open("PHP7-1604-ImagenesAdvertencia.php", "_self");
15     return false;
16 }
17 function borrarImagen(foto) {
18     window.open("PHP7-1604-BorraImagen.php?foto="+foto, "_self");
19     return false;
20 }
21
22 </script>
23 </head>
24 <body>
25 <?php
26 print "<table border='1'>";
27 print "<tr>";
28 print "<th>Imagen</th>";
29 print "<th>Archivo</th>";
30 print "<th>Alto</th>";
31 print "<th>Ancho</th>";
32 print "<th>Tamaño kb</th>";
33 print "</tr>";
34 print "<tr>";
35 $archivo = "fotos/".$foto;
36 $info = getimagesize($archivo);
37 print "<td>";
38 print "<img src='fotos/".$foto."' width='300' />";
39 print "</td>";
40 print "<td>";
41 print $foto;
42 print "</td>";
43 print "<td>";
44 print $info[0];
45 print "</td>";
46 print "<td>";
47 print $info[1];
48 print "</td>";
49 print "<td>";
50 print number_format(filesize($archivo)/1024,2);
51 print "</td>";
52 print "</tr>";
53 print "</table>";
54 ?>
```

```
55 <p>Advertencia: una vez borrado el archivo, no se podrá recuperar</p>
56 <form>
57   <input type="button" value="Borrar" onclick="borrarImagen('<?php print $foto;
   ?>');">
58   <input type="button" value="Regresar" onclick="regresar();">
59 </form>
60 </body>
61 </html>
```

## Listado 16.4.3. PHP7-1604-BorrarFoto.php



*www.pacoarce.com*

## 16.5. Parámetros para la optimización de una imagen

Antes de pasar al script que optimizará la imagen, necesitamos capturar algunos parámetros. Para ello crearemos una forma que nos indique el porcentaje que desea reducir la página, así como el nombre del nuevo archivo.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Imágenes | Tabla fotos</title>
5  </head>
6  <body>
7  <?php
8    /*****
9    F U N C I O N E S
10   *****/
11  function validaFoto($imagen){
12    $patron = "%\.(gif|jpe?g|png)$%i";
13    // Ejemplo de visualización del resultado
14    $bandera = preg_match($patron, $imagen) == 1 ? true : false;
15    return $bandera;
16  }
17  /*****
18  V A R I A B L E S
19   *****/
20  $dir = opendir("fotos");
21  $ren = 1;
22  $i = 0;
23  /*****
24  I N I C I O
25   *****/
26  if ($dir) {
27    print "<table border='1'>";
28    print "<tr>";
29    print "<th>Imagen</th>";
30    print "<th>Archivo</th>";
31    print "<th>Alto</th>";
32    print "<th>Ancho</th>";
33    print "<th>Tamaño kb</th>";
34    print "<th>Borrar</th>";
35    print "<th>Optimizar</th>";
36    print "</tr>";
37    print "<tr>";
38    while ($foto=readdir($dir)) {
39      if ($foto != "." && $foto != ".." && validaFoto($foto)) {
40        if ($i == $ren) {
41          $i=0;
42          print "</tr>";
43          print "<tr>";
```

```
44     }
45     $i++;
46     $archivo = "fotos/".$foto;
47     $info = getimagesize($archivo);
48     print "<td>";
49     print "<img src='fotos/".$foto."' width='100' />";
50     print "</td>";
51     print "<td>";
52     print $foto;
53     print "</td>";
54     print "<td>";
55     print $info[0];
56     print "</td>";
57     print "<td>";
58     print $info[1];
59     print "</td>";
60     print "<td>";
61     print number_format(filesize($archivo)/1024,2);
62     print "</td>";
63     print "<td>";
64     print "<a href='PHP7-1605-borrarFoto.php?foto=".$foto."'>Borrar</a>";
65     print "</td>";
66     print "<td>";
67     print "<a href='PHP7-1605-opcionesFoto.php?foto=".$foto."'>Opciones</a>";
68     print "</td>";
69 }
70 }
71 print "</tr>";
72 print "</table>";
73 } else {
74     print "Error al abrir la carpeta fotos";
75 }
76
77
78
79 ?>
80 </body>
81 </html>
```

## Listado 16.5.1. PHP7-1605-ImagenesAdvertencia.php

```
1 <?php
2 if (isset($_GET["foto"])) {
3     $foto = $_GET["foto"];
4 } else {
5     header("location:PHP7-1605-ImagenesAdvertencia.php");
6 }
7 ?>
8 <!DOCTYPE html>
9 <html>
10 <head>
11     <title>Imagenes | Opciones foto</title>
```

```
12 <script>
13     function regresar() {
14         window.open("PHP7-1605-ImagenesAdvertencia.php", "_self");
15         return false;
16     }
17     function opcionesImagen(foto) {
18         var op = document.getElementById("op").value;
19         var nombre = document.getElementById("nombre").value;
20         window.open("PHP7-1605-optimizaImagen.php?foto="+foto+"&op="+op+"&nombre="+nombre,
21 "_self");
22         return false;
23     }
24 </script>
25 </head>
26 <body>
27 <?php
28     print "<table border='1'>";
29     print "<tr>";
30     print "<th>Imagen</th>";
31     print "<th>Archivo</th>";
32     print "<th>Alto</th>";
33     print "<th>Ancho</th>";
34     print "<th>Tamaño kb</th>";
35     print "</tr>";
36     print "<tr>";
37     $archivo = "fotos/".$foto;
38     $info = getimagesize($archivo);
39     print "<td>";
40     print "<img src='fotos/".$foto."' width='300' />";
41     print "</td>";
42     print "<td>";
43     print $foto;
44     print "</td>";
45     print "<td>";
46     print $info[0];
47     print "</td>";
48     print "<td>";
49     print $info[1];
50     print "</td>";
51     print "<td>";
52     print number_format(filesize($archivo)/1024,2);
53     print "</td>";
54     print "</tr>";
55     print "</table>";
56 ?>
57 <p>Selecciona una opción para redir la foto en forma proporcional</p>
58 <form>
59     <select id="op">
60         <option value="75">75%</option>
61         <option value="50">50%</option>
62         <option value="25">25%</option>
```

```
63     <option value="10">10%</option>
64     <option value="5">5%</option>
65 </select>
66 <br><br>
67 <label for="nombre">Nombre de la nueva imagen:</label>
68 <br><br>
69 <input type="text" name="nombre" id="nombre"/>
70 <br><br>
71
72 <input type="button" value="Optimizar" onclick="opcionesImagen('<?php print $foto;
?>');">
73 <input type="button" value="Regresar" onclick="regresar();">
74 </form>
75 </body>
76 </html>
```

**Listado 16.5.2.** PHP7-1605-OpcionesFoto.php

*www.pacoarce.com*

## 16.6. Optimiza la imagen por un porcentaje

Por medio de la librería *GD* podemos optimizar una imagen, en este caso cambiaremos la imagen en forma proporcional por medio de un porcentaje.

El resultado es un nuevo archivo de imagen que mantiene la proporción y que es optimizado al 80% de calidad del original, reduciendo así considerablemente su tamaño.

**getimagesize** — Obtener el tamaño de una imagen

array **getimagesize** ( string \$filename [, array &\$imageinfo ] )

Devuelve un arreglo de hasta 7 elementos. No todos los tipos de imagen incluirán los elementos “channels” y “bits”.

Los índices 0 y 1 contienen el ancho y el alto de la imagen, respectivamente.

Nota: Algunos formatos podrían no contener imágenes o contener múltiples imágenes. En estos casos,

**getimagesize** no será capaz de determinar apropiadamente el tamaño de la imagen.

**getimagesize** devolverá cero para el ancho y el alto en estos casos.

El índice 2 es una de las constantes `IMAGETYPE_XXX` que indican el tipo de imagen.

El índice 3 es una cadena de texto con la cadena `height="yyy" width="xxx"` que puede ser usada directamente en una etiqueta `IMG`.

**mime** es el tipo MIME correspondiente de la imagen.

Esta información se puede usar para emitir imágenes con la cabecera HTTP Content-type correcta.

**imagecreatefromjpeg** — Crea una nueva imagen a partir de un fichero o de una URL

resource **imagecreatefromjpeg** ( string \$filename )

Devuelve un identificador de imagen que representa la imagen obtenida desde el nombre de archivo dado.

**imagecreatetruecolor** — Crear una nueva imagen de color verdadero

resource **imagecreatetruecolor** ( int \$width , int \$height )

**imagecreatetruecolor** () devuelve un identificador de imagen que representa una imagen en negro del tamaño especificado.

**imagecopyresampled** () copia una porción rectangular de una imagen a otra imagen, interpolando de manera suave los valores de los píxeles por lo que, particularmente, al reducir el tamaño de una imagen, ésta todavía conserva mucha nitidez.

**imagecopyresampled** — Copia y cambia el tamaño de parte de una imagen redimensionándola

bool **imagecopyresampled** ( resource \$dst\_image , resource \$src\_image , int \$dst\_x , int \$dst\_y , int \$src\_x , int \$src\_y , int \$dst\_w , int \$dst\_h , int \$src\_w , int \$src\_h )

En otras palabras, `imagecopyresampled()` tomará un área rectangular de `src_image` de ancho



src\_w y alto src\_h en la posición (src\_x,src\_y) y la coloca en un área rectangular de dst\_image de ancho dst\_w y alto dst\_h en la posición (dst\_x,dst\_y).

Si las coordenadas del origen y destino y los anchos y altos difieren, se llevará a cabo la ampliación o reducción del fragmento de la imagen. Las coordenadas se refieren a la esquina superior izquierda. Esta función se puede usar para copiar regiones dentro de la misma imagen (si dst\_image es el mismo que src\_image) pero si las regiones se traslapan el resultado será impredecible.

**imagejpeg** — Exportar la imagen al navegador o a un archivo

bool **imagejpeg** ( resource \$image [, mixed \$to [, int \$quality ]] )

**imagejpeg()** crea un archivo JPEG desde image.

---

```
1  <?php
2  if (isset($_GET["foto"])==false) {
3      header("location:index.php")
4  }
5
6  $foto = $_GET["foto"];
7  $porcien = $_GET["op"];
8  $nombre = $_GET["nombre"];
9
10 $archivo = "fotos/".$foto;
11
12 $info = getimagesize($archivo);
13 $ancho = $info[0];
14 $alto = $info[1];
15 $tipo = $info["mime"];
16
17 $nuevoAncho = $ancho * ($porcien/100);
18 $nuevoAlto = $alto * ($porcien/100);
19
20
21 ?>
```

**Listado 16.6.1.** optimizarImagen.php

## 16.7. Optimizar una imagen por su ancho en forma proporcional

En este video te mostraremos cómo optimizar una imagen dándole sólo un parámetro, en este caso, el ancho, y calculando el alto, con la finalidad de mantener la proporcionalidad de la imagen. Esto lo realizaremos por medio de la librería GD que viene instalada con PHP.

```
1  <?php
2  if (isset($_GET["foto"])==false) {
3      header("location:index.php");
4  }
5
6  $foto = $_GET["foto"];
7  $porcien = $_GET["op"];
8  $nombre = $_GET["nombre"];
9
10 $archivo = "fotos/".$foto;
11
12 $info = getimagesize($archivo);
13 $ancho = $info[0];
14 $alto = $info[1];
15 $tipo = $info["mime"];
16
17 $nuevoAncho = $ancho * $porcien / 100;
18 $nuevoAlto = $alto * $porcien / 100;
19
20 switch ($tipo) {
21     case 'image/jpg':
22     case 'image/jpeg':
23         $imagen = imagecreatefromjpeg($archivo);
24         break;
25
26     case 'image/png':
27         $imagen = imagecreatefrompng($archivo);
28         break;
29
30     case 'image/gif':
31         $imagen = imagecreatefromgif($archivo);
32         break;
33 }
34
35 //creamos el lienzo donde vaciaremos la nueva imagen
36 $lienzo = imagecreatetruecolor($nuevoAncho, $nuevoAlto);
37
38 //optimizamos el tamaño GD
39 imagecopyresampled($lienzo, $imagen, 0, 0, 0, 0, $nuevoAncho, $nuevoAlto, $ancho,
40 $alto);
```

```
41 //Vaciamos de la memoria RAM al disco
42 imagejpeg($lienzo, "fotos/". $nombre, 80);
43
44 //regresmos
45 header("location:index.php");
46
47
48 ?>
```

### Listado 16.7.1. optimizarImagen.php



*www.pacoarce.com*

## 16.8. El uso de filtros a una imagen

Antes de aplicar los filtros, debemos de analizar su entorno y sus parámetros.

En este video prepararemos y revisaremos la librería GD con la cual procesamos las imágenes en PHP.

**imagefilter** — Aplica un filtro a una imagen

bool **imagefilter** ( resource \$image , int \$filtertype [, int \$arg1 [, int \$arg2 [, int \$arg3 [, int \$arg4 ]]] ] )

El uso de filtros a una imagen

**IMG\_FILTER\_NEGATE**: Invierte todos los colores de la imagen.

**IMG\_FILTER\_GRAYSCALE**: Convierte la imagen a escala de grises.

**IMG\_FILTER\_BRIGHTNESS**: Cambia el brillo de la imagen. Use arg1 para establecer el nivel de brillo. El rango para el brillo es de -255 a 255.

**IMG\_FILTER\_CONTRAST**: Cambia el contraste de la imagen. Use arg1 para establecer el nivel de contraste.

El uso de filtros a una imagen

**IMG\_FILTER\_COLORIZE** : Como **IMG\_FILTER\_GRAYSCALE**, excepto que se puede especificar el color. Use arg1, arg2 y arg3 en la forma red, green, blue y arg4 para el canal alpha. El rango de cada color es de 0 a 255.

**IMG\_FILTER\_EDGEDETECT**: Utiliza detección de borde para resaltar los bordes de la imagen.

**IMG\_FILTER\_EMBOSS**: Pone en relieve la imagen.

**IMG\_FILTER\_GAUSSIAN\_BLUR**: Pone borrosa la imagen usando el método Gaussiano.

El uso de filtros a una imagen

**IMG\_FILTER\_SELECTIVE\_BLUR**: Pone borrosa la imagen.

**IMG\_FILTER\_MEAN\_REMOVAL**: Utiliza eliminación media para lograr un efecto "superficial".

**IMG\_FILTER\_SMOOTH**: Suaviza la imagen. Use arg1 para establecer el nivel de suavidad.

**IMG\_FILTER\_PIXELATE**: Aplica el efecto de pixelación a la imagen, use arg1 para establecer el tamaño de bloque y arg2 para establecer el modo de efecto de pixelación.

El uso de filtros a una imagen

arg1

**IMG\_FILTER\_BRIGHTNESS**: Nivel de brillo.

**IMG\_FILTER\_CONTRAST**: Nivel de contraste.

**IMG\_FILTER\_COLORIZE**: Valor del componente rojo.

**IMG\_FILTER\_SMOOTH**: Nivel de suavidad.

**IMG\_FILTER\_PIXELATE**: Tamaño de bloque en píxeles.

El uso de filtros a una imagen

arg2

**IMG\_FILTER\_COLORIZE:** Valor del componente verde.

**IMG\_FILTER\_PIXELATE:** Si usar o no el efecto de pixelación avanzado (por defecto a FALSE).

El uso de filtros a una imagen

arg3

**IMG\_FILTER\_COLORIZE:** Valor del componente azul.

arg4

**IMG\_FILTER\_COLORIZE:** Canal alfa. Un valor entre 0 y 127. 0 indica completamente opaco mientras que 127 indica completamente transparente.

---

```
1  <?php
2  if (isset($_GET["foto"])==false) {
3      header("location:index.php");
4  }
5  //Recibimos parámetros
6  $foto = $_GET["foto"];
7  $nuevoAncho = $_GET["ancho"];
8  $nombre = $_GET["nombre"];
9
10 $archivo = "fotos/".$foto;
11
12 $info = getimagesize($archivo);
13 $ancho = $info[0];
14 $salto = $info[1];
15 $tipo = $info["mime"];
16
17 //calculamos el nuevo alto
18 $factor = $nuevoAncho / $ancho;
19 $nuevoAlto = $salto * $factor;
20
21 //Leyendo el archivo a un objeto en la RAM
22 switch ($tipo) {
23     case 'image/jpg':
24     case 'image/jpeg':
25         $imagen = imagecreatefromjpeg($archivo);
26         break;
27
28     case 'image/png':
29         $imagen = imagecreatefrompng($archivo);
30         break;
31
32     case 'image/gif':
33         $imagen = imagecreatefromgif($archivo);
```

```
34     break;
35 }
36
37 //creamos el lienzo donde vaciaremos la nueva imagen
38 $lienzo = imagecreatetruecolor($nuevoAncho, $nuevoAlto);
39
40 //optimizamos el tamaño GD
41 imagecopyresampled($lienzo, $imagen, 0, 0, 0, 0, $nuevoAncho, $nuevoAlto, $ancho,
42 $alto);
43 //Vaciamos de la memoria RAM al disco
44 imagejpeg($lienzo, "fotos/".$nombre, 80);
45
46 //regresmos
47 header("location:index.php");
48
49
50 ?>
```

**Listado 16.8.1.** optimizarImagenAncho.php

*www.pacoarce.com*

## 16.9. Subir un archivo desde la computadora al servidor

En nuestras páginas nos será de mucha utilidad que el usuario suba sus propios archivos a nuestro sitio. En este video aprenderemos a realizar un programa en PHP para que el usuario pueda subir archivos de imágenes al sitio.

```
1  <?php
2  if (is_uploaded_file($_FILES['foto']['tmp_name'])) {
3      $nombre = "foto".date("YmdHis");
4      copy($_FILES["foto"]["tmp_name"], "fotos/".$nombre.".jpg");
5      header("location:index.php");
6  }
7  ?>
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <title>Subir Archivo</title>
12 </head>
13 <body>
14 <form enctype="multipart/form-data" action="subirArchivo.php" method="POST">
15   Subir foto: <input type="file" name="foto">
16   <input type="submit" value="Enviar">
17 </form>
18 </body>
19 </html>
```

**Listado 16.9.1.** subirArchivo.php

*www.pacoarce.com*

## 16.10. Validar el tipo de archivo subido

Por medio de la opción “type” del arreglo superglobal \$\_FILES[ ] podemos saber el tipo de archivo.

`$_FILES['file']['tmp_name']`

`$_FILES['file']['name']`

`$_FILES['file']['size']`

`$_FILES['file']['type']`

`$_FILES['file']['error']`

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Imágenes | Tabla fotos</title>
5  </head>
6  <body>
7  <?php
8    /*****
9    F U N C I O N E S
10   *****/
11   function validaFoto($imagen){
12     $patron = "%\.(gif|jpe?g|png)$%i";
13     // Ejemplo de visualización del resultado
14     $bandera = preg_match($patron, $imagen) == 1 ? true : false;
15     return $bandera;
16   }
17   /*****
18   V A R I A B L E S
19   *****/
20   $dir = opendir("fotos");
21   $ren = 1;
22   $i = 0;
23   /*****
24   I N I C I O
25   *****/
26   if ($dir) {
27     print "<a href='subirArchivo.php'>Subir imagen</a>";
28     print "<table border='1'>";
29     print "<tr>";
30     print "<th>Imagen</th>";
31     print "<th>Archivo</th>";
32     print "<th>Ancho</th>";
33     print "<th>Alto</th>";
34     print "<th>Tamaño kb</th>";
35     print "<th>Borrar</th>";
36     print "<th>Optimizar (%)</th>";
37     print "<th>Optimizar (ancho)</th>";
```



```
38 print "<th>Filtros</th>";
39 print "</tr>";
40 print "<tr>";
41 while ($foto=readdir($dir)) {
42     if ($foto != "." && $foto != ".." && validaFoto($foto)) {
43         if ($i == $ren) {
44             $i=0;
45             print "</tr>";
46             print "<tr>";
47         }
48         $i++;
49         $archivo = "fotos/".$foto;
50         $info = getimagesize($archivo);
51         print "<td>";
52         print "<img src='fotos/".$foto."' width='100' />";
53         print "</td>";
54         print "<td>";
55         print $foto;
56         print "</td>";
57         print "<td>";
58         print $info[0];
59         print "</td>";
60         print "<td>";
61         print $info[1];
62         print "</td>";
63         print "<td>";
64         print number_format(filesize($archivo)/1024,2);
65         print "</td>";
66         print "<td>";
67         print "<a href='borrarFoto.php?foto=".$foto."'>Borrar</a>";
68         print "</td>";
69         print "<td>";
70         print "<a href='opcionesFoto.php?foto=".$foto."'>Opciones</a>";
71         print "</td>";
72         print "<td>";
73         print "<a href='optimizarAncho.php?foto=".$foto."'>Opciones Ancho</a>";
74         print "</td>";
75         print "<td>";
76         print "<a href='opcionesFiltro.php?foto=".$foto."'>Filtros</a>";
77         print "</td>";
78     }
79 }
80 print "</tr>";
81 print "</table>";
82 } else {
83     print "Error al abrir la carpeta fotos";
84 }
85 ?>
86 </body>
87 </html>
```

**Listado 16.10.1.** index.php

## 16.11. Subir una imagen de PHP a la base de datos MySQL

Necesitamos crear una columna en la tabla de tipo longblob.

El archivo debemos recibirlo dentro del arreglo \$\_FILES[ ].

Recibimos los datos de la imagen con la función file\_get\_contents(). Transmite un archivo completo a una cadena.

Aplicamos la función addslashes() la cual devuelve una cadena con barras invertidas delante de los caracteres que necesitan ser escapados. Estos caracteres son la comilla simple ('), comilla doble ("), barra invertida (\) y NUL (el byte NULL).

Subir una imagen de PHP a la base de datos MySQL

```
CREATE TABLE `imagenes` (  
  `id` smallint(6) NOT NULL,  
  `ancho` smallint(6) NOT NULL,  
  `altura` smallint(6) NOT NULL,  
  `tipo` char(15) COLLATE utf8_spanish_ci NOT NULL,  
  `imagen` mediumblob NOT NULL,  
  `nombre` varchar(100) COLLATE utf8_spanish_ci NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

---

```
1  <?php  
2  if (isset($_FILES['imagen']['tmp_name']) &&  
3  is_uploaded_file($_FILES['imagen']['tmp_name'])) {  
4  
5  //verificamos que se haya subido una imagen  
6  $formato_array = new array("image/jpeg","image/gif","image/png");  
7  if(in_array($_FILES['imagen']['type'],$formato_array)){  
8  $imagen_array = getimagesize($_FILES['imagen']['tmp_name']);  
9  
10 $imagenEscapada = file_get_contents($_FILES['imagen']['tmp_name']);  
11 $imagenEscapada = addslashes($imagenEscapada);  
12  
13 }  
14 }  
15  
16 ?>  
17 <!DOCTYPE html>  
18 <html>  
19 <head>  
20 <title>Subir imagen a MySQL</title>  
21 </head>  
22 <body>  
23 <form action="<?php print $_SERVER['PHP_SELF']; ?>" method="POST"  
  enctype="multipart/form-data">
```

```
24 <input type="file" name="imagen" id="imagen">
25 <input type="submit" value="Enviar imagen">
26 </form>
27
28 </body>
29 </html>
```

**Listado 16.11.1.** subirlmgenMySQL.php



*www.pacoarce.com*

## 16.12. Guardar la imagen y desplegar una tabla

Subir una imagen de PHP a la base de datos MySQL: Guardamos la información en la tabla, pero no podemos desplegarla directamente.

```
1  <?php
2  $conn = mysqli_connect("localhost","root","","quiz") or die("Error al abrir la base
   de datos");
3  if (isset($_FILES['imagen']['tmp_name']) &&
   is_uploaded_file($_FILES['imagen']['tmp_name'])) {
4
5  //verificamos que se haya subido una imagen
6  $formato_array = array("image/jpeg","image/gif","image/png");
7  if(in_array($_FILES['imagen']['type'],$formato_array)){
8  $imagen_array = getimagesize($_FILES['imagen']['tmp_name']);
9
10 $imagenEscapada = file_get_contents($_FILES['imagen']['tmp_name']);
11 $imagenEscapada = addslashes($imagenEscapada);
12
13 //Crear el sql
14 $sql = "INSERT INTO imagenes VALUES(0,";
15 $sql.= $imagen_array[0].", "; //ancho
16 $sql.= $imagen_array[1].", "; //altura
17 $sql.= "'".$_FILES['imagen']['type']."'"; //tipo
18 $sql.= "'".$imagenEscapada."'"; //imagen
19 $sql.= "'".$_FILES['imagen']['name']."'"; //nombre del archivo
20
21 if (mysqli_query($conn, $sql)) {
22 print "Se insertó correctamente la imagen.<br>";
23 } else {
24 print "Error al insertar la imagen.<br>";
25 }
26 }
27 }
28
29 ?>
30 <!DOCTYPE html>
31 <html>
32 <head>
33 <title>Subir imagen a MySQL</title>
34 </head>
35 <body>
36 <form action="<?php print $_SERVER['PHP_SELF']; ?>" method="POST"
   enctype="multipart/form-data">
37 <input type="file" name="imagen" id="imagen">
38 <input type="submit" value="Enviar imagen">
39 </form>
40
41 <p>Selecciona una imagen</p>
```

```
42 <div>
43 <?php
44 $r = mysqli_query($conn,"SELECT * FROM imagenes");
45
46 if ($r) {
47 while($row = mysqli_fetch_assoc($r)){
48 print "<a href='verImagen.php?id=".$row['id']."'>".$row['nombre']."
49 ( ".$row['ancho']."'x".$row['altura']."' )</a><br>";
50 }
51 ?>
52 </div>
53
54 </body>
55 </html>
```

**Listado 16.12.1.** subirlmgenMySQL.php

*www.pacoarce.com*

## 16.13. Desplegar una columna binaria de MySQL desde PHP

Desplegar una columna binaria de MySQL desde PHP

Necesitamos indicar el encabezado del tipo de imagen: data:image/jpeg;base64.

Usar la función base64\_encode() para tratar los datos binarios:

```
'';
```

```
header("Content-type:".$row['tipo']);]
```

---

```
1  <?php
2  $conn = mysqli_connect("localhost","root","","quiz") or die("Error al abrir la base
   de datos");
3
4  $r = mysqli_query($conn, "SELECT * FROM imagenes WHERE id=".$_GET['id']);
5  $row = mysqli_fetch_assoc($r);
6
7  //mostramos la imagen
8  header("Content-type:".$row["tipo"]);
9  print $row["imagen"];
10
11 ?>
```

**Listado 16.13.1.** verImagen.php

*www.pacoarce.com*

## 16.14. de BLOB a un archivo de imagen

De BLOB a un archivo de imagen

Para crear un archivo de imagen, sólo hay que guardar el contenido del campo Blob a un archivo.

`fwrite();`

---

```
1  <?php
2  $conn = mysqli_connect("localhost","root","","quiz") or die("Error al abrir la base
3  de datos");
4  $r = mysqli_query($conn, "SELECT * FROM imagenes WHERE id=".$_GET['id']);
5  $row = mysqli_fetch_assoc($r);
6
7  //Guardar la imagen
8  $f = fopen($row['nombre'], "w+");
9
10 fwrite($f, $row['imagen']);
11
12 print "<img src='".$_row['nombre']."' />";
13
14 ?>
```

**Listado 16.14.1.** guardarImagen.php

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 17: Ejemplos y ejercicios

**Objetivo:** El alumno aprenderá a enviar un correo electrónico desde el servidor, calcular edades de una persona, paginar la consulta de una tabla y crear un captcha para sus formularios.

#### Introducción:

En esta sección realizaremos el primer bloque de ejercicios sencillos para aplicar lo que hemos aprendido hasta este punto:

- Utilizar un mismo archivo con estados en PHP
- Mandar un mail desde PHP
- Mandar un mail en formato HTML desde PHP
- Calcular las edades con PHP
- Leer y mostrar una tabla de la base de datos
- Realizar la paginación de una tabla con PHP
- Crear los botones HTML para la paginación con PHP
- Crear botones inicio-fin en la paginación con PHP
- Crear la imagen de un captcha con PHP
- Generar el texto aleatorio en el Captcha con PHP
- Llamar al captcha y verificarlo desde un formulario con PHP
- Conocer la IP, el browser y el sistema operativo del usuario

[www.pacoarce.com](http://www.pacoarce.com)



## 17.1. Utilizar un mismo archivo con estados en PHP

- En muchos casos con PHP crecen los archivos en forma exponencial, ya que vamos saltando entre ellos para realizar procesos o verificaciones.
- Una forma de evitar este crecimiento es utilizar un mismo archivo con diferentes resultados, es decir, el mismo archivo abrirá diferentes zonas de la misma página, dependiendo el parámetro que nosotros le pasemos.
- En esta clase ejemplifica cómo tener en el mismo archivo, una zona para el formulario y otra para la respuesta del mismo.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Ejemplo | Estados</title>
5    <meta charset="utf-8">
6    <?php
7      if (isset($_POST["estado"])) {
8        $tc = 18.90;
9        $cantidad = $_POST["cantidad"];
10       $tipo = $_POST["tipo"];
11       if ($tipo=="1") {
12         $r = $cantidad * $tc;
13         $rc = number_format($r,2);
14         print "La cantidad es $".$rc." de pesos mexicanos por ";
15         print number_format($cantidad,2);
16         print " de dólares americanos al tipo de cambio de ";
17         print number_format($tc,2);
18       }
19       if ($tipo=="2") {
20         $tcd = 1 / $tc;
21         $r = $cantidad / $tc;
22         $rc = number_format($r,2);
23         print "La cantidad es $".$rc." de dólares americanos por ";
24         print number_format($cantidad,2);
25         print " de pesos mexicanos al tipo de cambio de ";
26         print number_format($tcd,2);
27       }
28     }
29     print "<br><br>";
30   ?>
31 </head>
32 <body>
33 <form method="post" action="PHP7-1701-Estados.php">
34   <label>Introduzca la cantidad a convertir:</label>
35   <input type="text" name="cantidad" size="10"/>
36   <br><br>
37   <input type="radio" name="tipo" value="1" checked/>Dólares<br>
38   <input type="radio" name="tipo" value="2"/>Pesos Mexicanos<br>
39   <input type="hidden" name="estado" value="1"/>
40   <input type="submit" value="Convertir">
```

```
41 </form>
42 </body>
43 </html>
```

**Listado 17.1.1.** estados.php



*www.pacoarce.com*

## 17.2. Mandar un mail desde PHP

Un ejercicio clásico (y muy necesario) con PHP es enviar un mail desde el manejador de mensajes del servidor, y no desde la computadora, por ejemplo, con outlook, que la mayoría de las personas lo tenemos desactivado. En este video realizaremos el envío y la forma en el mismo archivo.

Sintaxis

`bool mail ( string $to , string $subject , string $mensaje [, string $encabezados [, string $parametros ] ] );`

**to:** Destinatario/s del correo. El formato de este string debe cumplir con la » RFC 2822.

**Subject:** Título del correo electrónico a enviar. Precaución: El título debe cumplir con la » RFC 2047.

**Mensaje:** Mensaje a enviar. Cada línea debería separarse con un CRLF (`\r\n`). Las líneas no deberían ocupar más de 70 caracteres.

**Encabezados** (opcional): String a insertar al final de la cabecera del correo. Se usa normalmente para añadir cabeceras extra (From, Cc y Bcc). Las cabeceras múltiples adicionales deberían separarse con un CRLF(`\r\n`). Si se usan datos externos para componer esta cabecera, los datos deberían ser saneados para que no se inyecten cabeceras no deseadas.

**Parametros** (opcional) El parámetro `additional_parameters` puede usarse para indicar opciones adicionales como opciones de línea de comandos al programa que está configurado para usarse cuando se envía correo, definido por la opción de configuración `sendmail_path`. Por ejemplo, puede usarse para establecer la dirección de correo del remitente usando `sendmail` con su opción `-f`.

**Nota:** Es importante indicar que la función `mail()` no es conveniente para grandes volúmenes de correo en bucle. Esta función abre y cierra un socket SMTP para cada correo, algo que no es muy eficiente.

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Enviar un correo</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <?php
9   $mensaje = "Hola, cara de bola\r\n¿Cómo estás?";
10  $mensaje = wordwrap($mensaje, 70, "\r\n");
11  $para = "micorreo@misitio.com";
12  $titulo = "Tienes un correo";
```

```
13 $cabecera = "From: info@misitio.com\r\n".
14 "Reply-To: info@misitio.com\r\n".
15 "X-Mailer: PHP/" . phpversion();
16
17 if(mail($para, $titulo, $mensaje)){
18     print "Mensaje enviado";
19 } else {
20     print "Error al enviar el mensaje";
21 }
22
23 ?>
24 </body>
25 </html>
```

**Listado 17.2.1.** enviarCorreo.php

*www.pacoarce.com*

## 17.3. Mandar un mail en formato HTML desde PHP

// Para enviar un correo HTML, debe establecerse la cabecera Content-type

```
$cabeceras = 'MIME-Version: 1.0' . "\r\n";
```

```
$cabeceras .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
```

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Enviar correo HTML</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $para = "correo1@misitio.com, correo2@misitio.com";
10     $subject = "Recordatorio de cumpleaños del mes";
11     $mensaje = "<html>
12       <head><title>Recordatorio de cumpleaños del mes de Abril</title></head>
13       <body><p>Estas son las personas que cumplen años en Abril</p>
14       <table><tr><th>Empleado</th><th>Día</th><th>Mes</th><th>Año</th></tr>
15       <tr><td>Juanita Pérez</td><td>1</td><td>Abril</td><td>1990</td></tr>
16       <tr><td>Pedro Picapiedra</td><td>5</td><td>Abril</td><td>1987</td></tr>
17       <tr><td>Pablo Mármol</td><td>10</td><td>Abril</td><td>1985</td></tr>
18     </table></body></html>";
19
20     //Cabeceras
21     $cabeceras = "MIME-Version: 1.0" . "\r\n";
22     $cabeceras .= "Content-type: text/html; charset=iso-8859-1" . "\r\n";
23
24     //Otras cabeceras
25     $cabeceras .= "To: Mary <recursoshumanos@misitio.com>, Direccion
26     <correo2@misitio.com>" . "\r\n";
27     $cabeceras .= "From: Recordatorio <cumples@misitio.com>" . "\r\n";
28     $cabeceras .= "Cc: pepito@misitio.com" . "\r\n";
29     $cabeceras .= "Bcc: juanito@misitio.com" . "\r\n";
30
31     mail($para, $subject, $mensaje, $cabeceras);
32   ?>
33 </body>
34 </html>
```

**Listado 17.3.1.** enviarMensajeHTML.php

## 17.4. Calcula las edades con PHP

Por medio de un algoritmo muy sencillo, calcularemos la edad exacta de una fecha en formato ISO, simplemente comparando meses y días del mes con la fecha actual.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Calcula edades</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    function edad($fecha){
10      $dia = date("j");
11      $mes = date("n");
12      $anio = date("Y");
13
14      $aNacimiento = substr($fecha,0,4);
15      $mNacimiento = substr($fecha,5,2);
16      $dNacimiento = substr($fecha,8,2);
17
18      if ($mNacimiento > $mes) {
19        $edad = $anio - $aNacimiento - 1;
20      } else if($mes==$mNacimiento && $dNacimiento > $dia){
21        $edad = $anio - $aNacimiento - 1;
22      } else {
23        $edad = $anio - $aNacimiento;
24      }
25      return $edad;
26    }
27    print "Tenemos ".edad("1941-12-14")." años";
28  ?>
29  </body>
30  </html>
```

**Listado 17.4.1.** calculaEdades.php

[www.pacoarce.com](http://www.pacoarce.com)

## 17.5. Leer y mostrar una tabla de la base de datos

- Una tarea muy común, es leer los datos de una tabla en una base de datos, generalmente MySQL y desplegarla por medio de PHP.
- Por lo general es el primer paso para una aplicación ABC o CRUD.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Leer tabla de MySQL con PHP</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $host = "localhost";
10     $usuario = "root";
11     $clave = "root"; //en XAMPP la clave es vacía
12     $db = "escuela";
13     $puerto = "3308"; //en XAMPP no se necesita el puerto
14     //
15     $conn = mysqli_connect($host, $usuario, $clave, $db, $puerto)
16     or die("Error al conectar la base de datos");
17     $q = "SELECT * FROM alumnos";
18     $r = mysqli_query($conn,$q);
19     print "<table border='1'>";
20     print "<tr>";
21     print "<th>Num.</th>";
22     print "<th>Nombres</th>";
23     print "<th>Apellidos</th>";
24     print "<th>Fecha de Nacimiento</th>";
25     print "<th>Promedio</th>";
26     print "<th>Género</th>";
27     print "</tr>";
28     while ($data = mysqli_fetch_assoc($r)) {
29       print "<tr>";
30       print "<td>".$data["id"]."</td>";
31       print "<td>".$data["nombres"]."</td>";
32       print "<td>".$data["apellidos"]."</td>";
33       print "<td>".$data["fechaNacimiento"]."</td>";
34       print "<td>".$data["promedio"]."</td>";
35       print "<td>".$data["genero"]."</td>";
36       print "</tr>";
37     }
38     print "</table>";
39     ?>
40   </body>
41 </html>
```

**Listado 17.5.1.** leerTabla.php

## 17.6. Realizar la paginación de una tabla con PHP

Una tarea muy común, es leer los datos de una tabla en una base de datos, generalmente MySQL y desplegarla por medio de PHP, pero frecuentemente debemos llevar el control de la paginación para no saturar el ancho de banda.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Paginar contenido de una tabla</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9      $host = "localhost";
10     $usuario = "root";
11     $clave = "root";
12     $db = "escuela";
13     $puerto = "3308";
14     $conn = mysqli_connect($host, $usuario, $clave, $db, $puerto)
15     or die("Error al conectar la base de datos");
16     //
17     //Variables y constantes
18     //
19     $TAMANO_PAGINA = 10;
20     $PAGINAS_MAXIMAS = 5;
21     //
22     if (isset($_GET["p"])) {
23       $pagina = $_GET["p"];
24     } else {
25       $pagina = 1;
26     }
27     //
28     //Calculamos el inicio de la pagina
29     //
30     $inicio = ($pagina-1)*$TAMANO_PAGINA;
31     //
32     $q = "SELECT COUNT(*) as reg FROM alumnos";
33     $r = mysqli_query($conn,$q);
34     $data = mysqli_fetch_assoc($r);
35     $num = $data["reg"];
36     //
37     //Total de páginas
38     //
39     $totalPaginas = ceil($num/$TAMANO_PAGINA);
40     //
41     //Lectura de la página
42     //
43     $q = "SELECT * FROM alumnos LIMIT ".$inicio.", ".$TAMANO_PAGINA;
```



```
44 $r = mysqli_query($conn,$q);
45 print "<h2>Tenemos en total ".$num." alumnos</h2>";
46 print "<table border='1'>";
47 print "<tr>";
48 print "<th>Num.</th>";
49 print "<th>Nombres</th>";
50 print "<th>Apellidos</th>";
51 print "<th>Fecha de Nacimiento</th>";
52 print "<th>Promedio</th>";
53 print "<th>Género</th>";
54 print "</tr>";
55 while ($data = mysqli_fetch_assoc($r)) {
56     print "<tr>";
57     print "<td>".$data["id"]."</td>";
58     print "<td>".$data["nombres"]."</td>";
59     print "<td>".$data["apellidos"]."</td>";
60     print "<td>".$data["fechaNacimiento"]."</td>";
61     print "<td>".$data["promedio"]."</td>";
62     print "<td>".$data["genero"]."</td>";
63     print "</tr>";
64 }
65 print "</table>";
66 ?>
67 </body>
68 </html>
```

**Listado 17.6.1.** paginacion.php

[www.pacoarce.com](http://www.pacoarce.com)

## 17.7. Crear los botones HTML para la paginación con PHP

En esta clase creamos los botones dependiendo del número de elementos y del tamaño de la página.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Paginar contenido de una tabla</title>
5    <meta charset="utf-8">
6    <style>
7      .activo{
8        background-color: orange;
9      }
10   </style>
11   <script>
12     function cambiaPagina(p){
13       window.open("paginacionHTML.php?p="+p,"_self");
14     }
15   </script>
16 </head>
17 <body>
18 <?php
19 $host = "localhost";
20 $usuario = "root";
21 $clave = "root";
22 $db = "escuela";
23 $puerto = "3308";
24 $conn = mysqli_connect($host, $usuario, $clave, $db, $puerto)
25 or die("Error al conectar la base de datos");
26 //
27 //Variables y constantes
28 //
29 $TAMANO_PAGINA = 10;
30 $PAGINAS_MAXIMAS = 5;
31 //
32 if (isset($_GET["p"])) {
33   $pagina = $_GET["p"];
34 } else {
35   $pagina = 1;
36 }
37 //
38 //Calculamos el inicio de la pagina
39 //
40 $inicio = ($pagina-1)*$TAMANO_PAGINA;
41 //
42 $q = "SELECT COUNT(*) as reg FROM alumnos";
43 $r = mysqli_query($conn,$q);
44 $data = mysqli_fetch_assoc($r);
```

```
45 $num = $data["reg"];
46 //
47 //Total de páginas
48 //
49 $totalPaginas = ceil($num/$TAMANO_PAGINA);
50 //
51 //Lectura de la página
52 //
53 $q = "SELECT * FROM alumnos LIMIT ".$inicio.", ".$TAMANO_PAGINA;
54 $r = mysqli_query($conn,$q);
55 print "<h2>Tenemos en total ".$num." alumnos</h2>";
56 print "<table border='1'>";
57 print "<tr>";
58 print "<th>Num.</th>";
59 print "<th>Nombres</th>";
60 print "<th>Apellidos</th>";
61 print "<th>Fecha de Nacimiento</th>";
62 print "<th>Promedio</th>";
63 print "<th>Género</th>";
64 print "</tr>";
65 while ($data = mysqli_fetch_assoc($r)) {
66     print "<tr>";
67     print "<td>".$data["id"]."</td>";
68     print "<td>".$data["nombres"]."</td>";
69     print "<td>".$data["apellidos"]."</td>";
70     print "<td>".$data["fechaNacimiento"]."</td>";
71     print "<td>".$data["promedio"]."</td>";
72     print "<td>".$data["genero"]."</td>";
73     print "</tr>";
74 }
75 print "</table>";
76 if($totalPaginas > $PAGINAS_MAXIMAS){
77     //pagina actual es la ultima
78     if ($pagina== $totalPaginas) {
79         $inicio = $pagina-$PAGINAS_MAXIMAS;
80         $fin = $totalPaginas;
81     } else {
82         $inicio = $pagina;
83         $fin = $inicio-1 + $PAGINAS_MAXIMAS;
84     }
85 }
86 } else {
87     $inicio = 1;
88     $fin = $totalPaginas;
89 }
90 for($i=$inicio; $i<=$fin; $i++){
91     print "<button type='button' ";
92     if($i==$pagina) print " class='activo' ";
93     print " onclick='cambiaPagina(\".$i.\")'>".$i."</button>";
94 }
95 ?>
96 </body>
```

97 </html>

**Listado 17.7.1.** paginacionHTML.php



*www.pacoarce.com*

## 17.8. Crear botones inicio-fin en la paginación con PHP

Necesitamos detectar el inicio y el fin de la paginación:

```
if ($inicio!=1) {  
    print "<button type='button' onclick='cambiaPagina(1)'>Primera</button>";  
    print "<button type='button' onclick='cambiaPagina(\".$pagina-1.\")>Anterior</button>";  
}
```

Crear botones inicio-fin en la paginación con PHP

```
if ($totalPaginas>$PAGINAS_MAXIMAS && $pagina!=$totalPaginas) {  
    print "<button type='button' onclick='cambiaPagina(\".$pagina+1.\")>Siguiente</button>";  
    print "<button type='button' onclick='cambiaPagina(\".$totalPaginas.\")>Ultima</button>";  
}
```

---

```
1  <!DOCTYPE html>  
2  <html>  
3  <head>  
4      <title>Paginar contenido de una tabla</title>  
5      <meta charset="utf-8">  
6      <style>  
7          .activo{  
8              background-color: orange;  
9          }  
10     </style>  
11     <script>  
12         function cambiaPagina(p){  
13             window.open("paginacionInicioFin.php?p="+p,"_self");  
14         }  
15     </script>  
16 </head>  
17 <body>  
18 <?php  
19 $host = "localhost";  
20 $usuario = "root";  
21 $clave = "root";  
22 $db = "escuela";  
23 $puerto = "3308";  
24 $conn = mysqli_connect($host, $usuario, $clave, $db, $puerto)  
25 or die("Error al conectar la base de datos");  
26 //  
27 //Variables y constantes  
28 //  
29 $TAMANO_PAGINA = 10;  
30 $PAGINAS_MAXIMAS = 5;  
31 //  
32 if (isset($_GET["p"])) {  
33     $pagina = $_GET["p"];
```

```
34 } else {
35     $pagina = 1;
36 }
37 //
38 //Calculamos el inicio de la pagina
39 //
40 $inicio = ($pagina-1)*$TAMANO_PAGINA;
41 //
42 $q = "SELECT COUNT(*) as reg FROM alumnos";
43 $r = mysqli_query($conn,$q);
44 $data = mysqli_fetch_assoc($r);
45 $num = $data["reg"];
46 //
47 //Total de páginas
48 //
49 $totalPaginas = ceil($num/$TAMANO_PAGINA);
50 //
51 //Lectura de la página
52 //
53 $q = "SELECT * FROM alumnos LIMIT ".$inicio.", ".$TAMANO_PAGINA;
54 $r = mysqli_query($conn,$q);
55 print "<h2>Tenemos en total ".$num." alumnos</h2>";
56 print "<table border='1'>";
57 print "<tr>";
58 print "<th>Num.</th>";
59 print "<th>Nombres</th>";
60 print "<th>Apellidos</th>";
61 print "<th>Fecha de Nacimiento</th>";
62 print "<th>Promedio</th>";
63 print "<th>Género</th>";
64 print "</tr>";
65 while ($data = mysqli_fetch_assoc($r)) {
66     print "<tr>";
67     print "<td>".$data["id"]."</td>";
68     print "<td>".$data["nombres"]."</td>";
69     print "<td>".$data["apellidos"]."</td>";
70     print "<td>".$data["fechaNacimiento"]."</td>";
71     print "<td>".$data["promedio"]."</td>";
72     print "<td>".$data["genero"]."</td>";
73     print "</tr>";
74 }
75 print "</table>";
76 if($totalPaginas > $PAGINAS_MAXIMAS){
77     //pagina actual es la ultima
78     if ($pagina== $totalPaginas) {
79         $inicio = $pagina-$PAGINAS_MAXIMAS;
80         $fin = $totalPaginas;
81     } else {
82         $inicio = $pagina;
83         $fin = $inicio-1 + $PAGINAS_MAXIMAS;
84     }
85     if ($inicio!=1) {
```

```
86     print "<button type='button' onclick='cambiaPagina(1) '>Inicio</button>";
87     print "<button type='button'
88 onclick=' cambiaPagina(\".($pagina-1).\") '>Anterior</button>";
89 }
90 } else {
91     $inicio = 1;
92     $fin = $totalPaginas;
93 }
94 for($i=$inicio; $i<=$fin; $i++){
95     print "<button type='button' ";
96     if($i==$pagina) print " class='activo' ";
97     print " onclick='cambiaPagina(\".$i.\") '>\".$i.\"</button>";
98 }
99 if ($totalPaginas > $PAGINAS_MAXIMAS && $pagina!=$totalPaginas) {
100     print "<button type='button'
101 onclick=' cambiaPagina(\".($pagina+1).\") '>Siguiete</button>";
102     print "<button type='button'
103 onclick=' cambiaPagina(\".$totalPaginas.\") '>Fin</button>";
104 }
105 ?>
106 </body>
107 </html>
```

**Listado 17.8.1.** paginacionInicioFin.php

[www.pacoarce.com](http://www.pacoarce.com)

## 17.9. Crear la imagen de un captcha con PHP

- Crear la imagen de un captcha con PHP
- Crear la imagen
- Crear colores
- Crear mensaje aleatorio
- Desplegar texto
- Desplegar la imagen
- Guardar la cadena en una sesión

```
1  <?php
2  session_start();
3  //
4  //Establecer el contenido
5  //
6  header("Content-Type: image/png");
7  //
8  //Crear el objeto de la imagen
9  //
10 $img = imagecreatetruecolor(159, 30);
11 //
12 //Crear colores
13 //
14 $blanco = imagecolorallocate($img, 255, 255, 255);
15 $gris = imagecolorallocate($img, 128, 128, 128);
16 $negro = imagecolorallocate($img, 0, 0, 0);
17 $amarillo = imagecolorallocate($img, 255, 255, 0);
18 //
19 imagefilledrectangle($img, 0, 0, 399, 29, $blanco);
20 //
21 //Desplegar el rectángulo
22 //
23 imagepng($img);
24 imagedestroy($img);
25 ?>
```

**Listado 17.9.1.** captcha.php

[www.pacoarce.com](http://www.pacoarce.com)



## 17.10. Generar el texto aleatorio en el Captcha con PHP

Generar el texto aleatorio en el Captcha con PHP

**rand(\$min, \$max)** Número aleatorio entre \$min y \$max.

**srand(\$semilla)** Crea un número aleatorio a partir de una semilla.

```
1  <?php
2  session_start();
3  //
4  //establecer el contenido
5  //
6  header("Content-Type: image/png");
7  //
8  //Crear el objeto de la imagen
9  //
10 $img = imagecreatetruecolor(150, 30);
11 //
12 //Crear colores
13 //
14 $blanco = imagecolorallocate($img, 255, 255, 255);
15 $gris = imagecolorallocate($img, 128, 128, 128);
16 $negro = imagecolorallocate($img, 0, 0, 0);
17 $blanco = imagecolorallocate($img, 255, 255, 0);
18 //
19 imagefilledrectangle($img, 0, 0, 399, 29, $blanco);
20 //
21 //Crear una cadena aleatoria
22 //
23 $l = rand(5,7);
24 $c = "abcdefghijklmnopqrstuvwxyz23456789";
25 $str = "";
26 $i = 0;
27 srand((double)microtime()*100000000);
28 while ($i <= $l) {
29     $num = rand() % 32;
30     $car = substr($c, $num, 1);
31     $str .= $car;
32     $i++;
33 }
34 //
35 $texto = $str;
36 //
37 //Archivo de la fuente
38 //
39 $fuente = "arial.ttf";
40 //
41 imagettftext($img, 20, 0, 15, 25, $gris, $fuente, $texto);
42 //
43 //sombra
```

www.pacoarce.com

```
44 //
45 imagettftext($img, 20, 0, 17, 27, $negro, $fuente, $texto);
46 //
47 $_SESSION["captcha"] = $texto;
48 //
49 //desplegar el rectángulo
50 //
51 imagepng($img);
52 imagedestroy($img);
53 ?>
```

## Listado 17.10.1. captcha.php



*www.pacoarce.com*

## 17.11. Llamar al captcha y verificarlo desde un formulario con PHP

Llamar al captcha y verificarlo desde un formulario con PHP

- Pasos para crear un captcha
- Crear la imagen
- Crear colores
- Crear mensaje aleatorio
- Desplegar texto
- Desplegar la imagen
- Guardar la cadena en una sesión

```
1  <?php
2  session_start();
3  //capturamos los datos
4  if (!$_POST["submit"]) {
5      print "<form method='POST' action='formulario.php'>";
6      print "<p>Escriba las letras que aparecen</p>";
7      print "<p><img src='captcha.php' /></p>";
8      print "<p><input type='text' name='imagen'></p>";
9      print "<p><input type='submit' name='submit' value='valida'></p>";
10     print "</form>";
11 } else {
12     $imagen = $_POST["imagen"];
13     if ($imagen==$_SESSION["captcha"]) {
14         print "<p>Excelente</p>";
15     } else {
16         print "<p>No coincide el captcha</p>";
17     }
18 }
19 ?>
```

**Listado 17.11.1.** formulario.php

*www.pacoarce.com*

## 17.12. Conocer la IP, el browser y el sistema operativo del usuario

Un punto muy importante para cualquier sitio son sus estadísticas. Para ello podemos saber cierta información del usuario por medio de comandos en PHP.

Con esta información podremos conocer a nuestros visitantes y tomar decisiones.

```
1  <?php
2  session_start();
3  //capturamos los datos
4  if (!$_POST["submit"]) {
5      print "<form method='POST' action='formulario.php'>";
6      print "<p>Escriba las letras que aparecen</p>";
7      print "<p><img src='captcha.php' /></p>";
8      print "<p><input type='text' name='imagen'></p>";
9      print "<p><input type='submit' name='submit' value='valida'></p>";
10     print "</form>";
11 } else {
12     $imagen = $_POST["imagen"];
13     if ($imagen==$_SESSION["captcha"]) {
14         print "<p>Excelente</p>";
15     } else {
16         print "<p>No coincide el captcha</p>";
17     }
18 }
19 ?>
```

**Listado 17.12.1.** formulario.php

[www.pacoarce.com](http://www.pacoarce.com)

# Todo PHP7

## De novato a experto

### Capítulo 18: Trabajar con XML y JSON desde PHP

**Objetivo:** El alumno aprenderá a leer y escribir en archivos con formato XML y JSON.

#### Introducción:

En esta sección veremos la forma de leer y escribir archivos en formato XML y JSON:

- Conceptos generales para leer un documento XML desde PHP
- Crear el programa que lee un documento XML con PHP
- Escribir un documento XML desde PHP
- Leer un archivo JSON
- Escribir un archivo JSON .

## 18.1. Conceptos generales para leer un documento XML desde PHP

El método **XMLReader::open**, utilizado como método estático, establece el URL de entrada para el contenido XML que se procesará.

La función **XMLReader::read** mueve al siguiente nodo en el documento.

Por su parte la función **XMLReader::next** mueve el cursor al siguiente nodo saltándose todos los subnodos.

Para obtener el valor de un atributo se utiliza la función **XMLReader::getAttribute(name)**.

Para obtener el valor (texto) de un nodo se puede utilizar el atributo **XMLReader::value** y para conocer el nombre del elemento se utiliza el atributo **XMLReader::name**.

*www.pacoarce.com*

## 18.2. Crear el programa que lee un documento XML con PHP

El método ***XMLReader::open***, utilizado como método estático, establece el URL de entrada para el contenido XML que se procesará.

```
1  <?php
2  //
3  //Archivo a leer
4  //
5  $path = "libros.xml";
6  //
7  //Verifica que exista el archivo
8  //
9  if (!file_exists($path)) exit("Archivo no encontrado");
10 //
11 //Abre el archivo
12 //
13 $xml = XMLReader::open($path);
14 //
15 //Crea las primeras etiquetas de la tabla
16 //
17 print "<html><body><table border=1>";
18 print "<tr><th>Titulo</th><th>Autor</th><th>Editorial</th>";
19 print "<th>Precio</th><th>Fecha</th></tr>";
20 //
21 //Inicia ciclo para leer el archivo
22 //
23 while ($xml->read()){
24     if ($xml->nodeType == XMLReader::ELEMENT && $xml->name == 'libro') {
25         $nombre = $xml->getAttribute('nombre');
26         $autor = $xml->getAttribute('autor');
27         $editroial = $xml->getAttribute('editorial');
28         $precio = $xml->getAttribute('precio');
29         $fecha = $xml->getAttribute('fecha');
30         print "<tr><td>".$nombre."</td><td>".$autor."</td>";
31         print "<td>".$editroial."</td>";
32         print "<td>".$precio."</td>";
33         print "<td>".$fecha."</td></tr>";
34     }
35 }
36 //
37 //Etiquetas de cierre
38 //
39 print "</table></body></html>";
40 //
41 //Cerramos la conexión
42 //
43 $xml->close();
44 ?>
```

**Listado 18.2.1.** libros.php

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <libros>
3   <libro paginas="600" nombre="El Quijote de la Mancha" autor="Miguel de Cervantes"
4     editorial="Oveja Negra" precio="200.00" fecha="2005"/>
5   <libro paginas="500" nombre="Cien años de soledad" autor="Gabriel García Márquez"
6     editorial="Oveja Negra" precio="300.00" fecha="2000"/>
7   <libro paginas="600" nombre="La guerra del fin del mundo" autor="Mario Vargas
8     Llosa" editorial="El barco de vapor" precio="350.00" fecha="2010"/>
9 </libros>
```

**Listado 18.2.2.** libros.xml



*www.pacoarce.com*



## 18.3. Escribir un documento XML desde PHP

La clase **XMLWriter** crea un objeto de escritura XML.

El método **XMLWriter::openURI(path)** establece el URI de salida para el contenido XML que se generará.

**XMLWriter::flush** escribe dicha entrada.

La función **XMLWriter::startDocument** crea el nodo principal de un documento xml.

El método **XMLWriter::endDocument** finaliza el documento.

Por su parte la función **XMLWriter::startElement** crea un elemento XML.

El método **XMLWriter::endElement** lo finaliza.

Para escribir el valor de un atributo se utiliza la función **XMLWriter::writeAttribute(name,value)**.

Para obtener el valor (texto) de un nodo se puede utilizar el atributo **XMLWriter::value**.

---

```
1  <?php
2
3  $path = "libros2.xml";
4
5  $writer = new XMLWriter();
6  $writer->openURI($path);
7  $writer->startDocument('1.0');
8
9  $writer->startElement('libros');
10
11 $writer->startElement('libro');
12 $writer->writeAttribute('nombre', 'El llano en llamas');
13 $writer->writeAttribute('autor', 'Juan Rulfo');
14 $writer->writeAttribute('editorial', 'Era');
15 $writer->writeAttribute('fecha', '1999');
16 $writer->writeAttribute('paginas', '200');
17 $writer->writeAttribute('precio', '200.00');
18 $writer->endElement();
19
20 $writer->startElement('libro');
21 $writer->writeAttribute('nombre', 'La región más transparente');
22 $writer->writeAttribute('autor', 'Carlos Fuentes');
23 $writer->writeAttribute('editorial', 'Diana');
24 $writer->writeAttribute('fecha', '1990');
25 $writer->writeAttribute('paginas', '350');
26 $writer->writeAttribute('precio', '300.00');
27 $writer->endElement();
28
29 $writer->endElement();
30
31 $writer->endDocument();
32
```

```
33 $writer->flush();  
34  
35 ?>
```

**Listado 18.3.1.** librosGuarda.php



*www.pacoarce.com*

## 18.4. Leer un archivo JSON

El método utilizado por **PHP** para tratar datos **JSON** es simplemente convertir cadenas (string) en formato **JSON** a arreglos de **PHP**.

Para ello se utiliza la función ***json\_decode(string)*** la cual recibe dicha cadena y retorna en arreglo.

```
1  <?php
2
3  $path = "alumnos2.json";
4
5  if (!file_exists($path))
6  exit("File not found");
7
8  $data = file_get_contents($path);
9  $json = json_decode($data, true);
10
11 echo "<html><body><table border=1>";
12 echo "<tr><th>Alumno</th><th>Calificación</th></tr>";
13 for ($i=0; $i < count($json); $i++) {
14   $n = $json[$i]['nombre'];
15   $c = $json[$i]['cali'];
16   echo "<tr><td>".$n."</td><td>".$c."</td></tr>";
17 }
18 echo "</table></body></html>";
19 ?>
```

**Listado 18.4.1.** librosJSONlee.php

[www.pacoarce.com](http://www.pacoarce.com)

## 18.5. Escribir un archivo JSON

De igual forma para escribir datos en formato **JSON**, **PHP** utiliza la función **json\_encode(array)** la cual recibe un arreglo de PHP y retorna una hilera de texto en formato **JSON**.

---

```
1 [{"nombre":"Juan","cali":"9.5"}, {"nombre":"Mar\u00eda","cali":"8.6"}]
```

**Listado 18.5.2.** alumnos2.json

---

```
1 <?php
2 //Archivo
3 $path = "alumnos2.json";
4 //Crear el archivo
5 $file = fopen($path, "w");
6 //Crea el arreglo
7 $alumnos = array(
8     array("nombre"=>"Juan","cali"=>"9.5"),
9     array("nombre"=>"María","cali"=>"8.6")
10 );
11 //Convertirlo en JSON
12 $json = json_encode($alumnos);
13 //Guardarlo
14 fwrite($file, $json);
15 //Cerrar el archivo
16 fclose($file);
17 ?>
```

**Listado 18.5.1.** librosJSONescribe.php

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 19: Introducción a Google Chart con PHP y MySQL

**Objetivo:** El alumno aprenderá las operaciones básicas para crear gráficas con Google Chart tomando la información de una base MySQL por medio de PHP.

**Introducción:**



*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 20: Crear un archivo PDF desde PHP

**Objetivo:** El alumno utilizará una biblioteca gratuita para crear un PDF desde MySQL con PHP.

#### Introducción:

Por medio de la librería FPDF podemos crear reportes dinámicos en este formato por medio de PHP. Los temas que veremos en esta sección son:

- Crear un PDF desde PHP con la librería FPDF
- Crear un encabezado y un pie de página con FPDF
- Leer textos externos y crear un archivo PDF con FPDF
- Manejo de columnas en texto con FPDF
- Crear una tabla dinámica con FPDF y PHP

*www.pacoarce.com*

## 20.1. Crear un PDF desde PHP con la librería FPDF



*[www.pacoarce.com](http://www.pacoarce.com)*

- Una de las muchas librerías para generar PDFs es FPDF (la F es de free).
- Hay que llamar la librería con require.
- Creamos un objeto:

```
$pdf = new FPDF();
```

## Añadir una página

**AddPage():** Añade una nueva página al documento. Si la página ya está presente, el método **Footer()** es invocado primero para producir el pie de página.

- Entonces es añadida la página, la posición actual se establece en la esquina superior izquierda de acuerdo a los márgenes izquierda y superior, y el **Header()** es invocado para mostrar el encabezado.
- La fuente que fue fijada antes de ser invocada es restaurada automáticamente. No es necesario invocar nuevamente **SetFont()** si desea continuar con la misma fuente. Esto también se aplica para colores y ancho de líneas.
- El origen del sistema de coordenadas es la esquina superior izquierda y se incrementan hacia abajo.
- Ejemplo:

```
$pdf->AddPage();
```

## Establecer una fuente:

```
$pdf->SetFont('Arial','B',16);
```

- Establece la fuente usada para imprimir cadenas de caracteres.
- Es obligatorio invocar este método por lo menos una vez antes de imprimir texto.
- La fuente puede ser una estándar o una adicionada vía el método **AddFont()**.
- Las fuentes estándar usan codificación cp1252 (Western Europe) de Windows.
- El método puede ser invocado antes que la primera página sea creada y es mantenido de página a página.
- Si sólo desea cambiar el tamaño de la fuente actual, es más simple invocar **SetFontSize()**.

## Imprime una celda

**Cell(float w [, float h [, string txt [, mixed border [, int ln [, string align [, boolean fill [, mixed link]]]]]]])**

- Imprime una celda (de área rectangular) bordes opcionales, color de fondo y secuencia de caracteres. La esquina superior izquierda de la celda corresponde a la posición actual.
- El texto puede ser alineado o centrado. Después de invocar, la posición actual se desplaza a la derecha o la siguiente línea.
- Es posible poner una referencia en el texto.
- Si está el salto de página automático habilitado y la celda está por fuera del límite, es realizado un salto de página antes de producir la salida.
- Ejemplo

```
$pdf->Cell(40,10,'¡Hola, Mundo!');
```

## Enviar el documento

Sitaxis:

```
string Output([string dest [, string name [, boolean isUTF8]])
```

- Envía el documento a un destino dado: una cadena, un fichero local o al navegador.
- En el último caso, puede utilizarse la extensión -plug in- (si existe) o forzarse un cuadro de diálogo de descarga.
- El método invoca ante todo a **Close()** si es necesario cerrar el documento.
- Le damos salida al documento:

```
$pdf->Output();
```



```
2
3 require('fpdf181/fpdf.php');
4
5 //Creamos el objeto
6 $pdf = new FPDF();
7
8
9 $pdf->AddPage();
10
11
12 $pdf->SetFont('Arial','B',16);
13
14 //Creamos contenidos
15
16
17 //Le damos la salida al documento
18 $pdf->Output();
19 ?>
20
```

**Listado 20.1.1.** fpdf01.php

*www.pacoarce.com*

## 20.2. Crear un encabezado y un pie de página con FPDF



*www.pacoarce.com*

## Añadir imágenes

`Image(string file [, float x [, float y [, float w [, float h [, string type [, mixed link]]]]])`

Imprime una imagen en la página. Las dimensiones pueden establecerse de diferentes maneras:

- mediante la especificación explícita de ancho y alto (en unidades definidas por el usuario o en ppp)
- mediante la indicación de una sola de las dimensiones: la otra se calculará automáticamente para mantener la proporción original sin indicar ninguna dimensión explícita.

En este caso, la imagen se imprime a 96 puntos por pulgada

Los formatos admitidos son JPEG, PNG y GIF. La extensión GD es necesaria para GIF.

Para el JPEG, se admiten todas sus versiones:

- escala de grises
- color verdadero (24 bits)
- CMYK (32 bits)
- Para el PNG se permiten:
  - escala de grises de 8 bits como máximo (256 niveles)
  - color indexado
  - color verdadero (24 bits)

Para el GIF: en el caso de un GIF animado, sólo será mostrado el primer fotograma.

- Se admite transparencias.
- color indexado
- color verdadero (24 bits)

El formato puede ser especificado explícitamente o deducido a partir de la extensión del fichero.

Es posible asociar un enlace a la imagen.

**Nota:** si una imagen se utiliza varias veces, sólo una copia será incrustada en el archivo.

## Salto de línea

`Ln([float h])`

Ejecuta un salto de línea. La abscisa actual regresa al margen izquierdo y la ordenada aumenta en la cantidad pasada como parámetro.

## Coordenada en Y

`SetY(float y [, boolean resetX])`

Establece la ordenada y de forma opcional mueve la abscisa al margen izquierdo.

Si el valor pasado es negativo, esta es relativa a la parte inferior de la página.

## Alias páginas

`AliasNbPages([string alias])`

Define un alias para el número total de páginas. Se sustituirá en el momento que el documento se cierre.

## Número páginas

`int PageNo()`

Devuelve el número de página actual.

```
2
3 require('fpdf181/fpdf.php');
4
5 class PDF extends FPDF
6 {
7
8     function Header()
9     {
10         // Logo
11         $this->Image('logo.png',10,8,33);
12         // Arial bold 15
13         $this->SetFont('Arial','B',15);
14         // Movernos a la derecha
15         $this->Cell(80);
16
17
18
19         $this->Ln(20);
20     }
21
22
23     function Footer()
24     {
25
26         $this->SetY(-15);
27         // Arial italic 8
28         $this->SetFont('Arial','I',8);
29
30
31     }
32 }
33
34 $pdf = new PDF();
35
36 $pdf->AliasNbPages();
37
38 $pdf->AddPage();
39 //Establece la fuente
40 $pdf->SetFont('Times','',12);
41 //Imprime celdas
42 for($i=1;$i<=40;$i++)
43
44
45 $pdf->Output();
46 ?>
47
```

**Listado 20.2.1.** fpdf02.php

### 20.3. Leer textos externos y crear un archivo PDF con FPDF



*[www.pacoarce.com](http://www.pacoarce.com)*

## Longitud de una cadena

float GetStringWidth(string s)

- Devuelve la longitud de una cadena en la unidad del usuario. Una fuente debe ser seleccionada.

## Coordena horizontal

SetX(float x)

- Define la abscisa de la posición actual.
- Si el valor pasado es negativo, esta es relativa a la derecha de la página.

## Color del borde

SetDrawColor(int r [, int g, int b])

- Define el color usado para las operaciones de graficación (líneas, rectángulos, y bordes de celdas ).
- Este puede ser expresado en componentes RGB o en escala de grises.
- El método puede ser invocado antes de que la primera página sea creada y el valor sea conservado de página a página.

## Color de fondo

SetFillColor(int r [, int g, int b])

- Define el color usado por todas las operaciones de relleno (rectángulos rellenos y fondos de celdas).

## Color del borde

SetFillColor(int r [, int g, int b])

- Este puede ser expresado en componentes RGB o escala de grises.
- El método puede ser invocado antes que la primera página sea creada y el valor es conservado de página a página.

## Color del texto

SetTextColor(int r [, int g, int b])

- Define el color usado por el texto.
- Este puede ser expresado en componentes RGB o escala de grises.

## Ancho de la línea

SetLineWidth(float width)

- Por defecto, el valor es igual a 0.2 mm.
- El método puede ser llamado antes de la creación de la primera página y el valor es retenido de página a página.

## Imprimir texto

MultiCell(float w, float h, string txt [, mixed border [, string align [, boolean fill]]])

- Este método permite imprimir texto con saltos de línea.
- Estos pueden ser automáticos (tan pronto como el texto alcanza el borde derecho de la celda) o explícito (vía el carácter \n).
- Tantas celdas como sean necesarias son creadas, uno debajo de otra.
- El texto puede ser alineado, centrado o justificado.
- El bloque de celda puede ser enmarcado y el fondo impreso.

## Título del documento

SetTitle(string title [, boolean isUTF8])

## Autor del documento

SetAuthor(string author [, boolean isUTF8])



*[www.pacoarce.com](http://www.pacoarce.com)*

```
1  <?php
2  require('fpdf181/fpdf.php');
3
4  class PDF extends FPDF
5  {
6  function Header()
7  {
8      global $title;
9
10     // Arial bold 15
11     $this->SetFont('Arial','B',15);
12
13     $w = $this->GetStringWidth($title)+6;
14     $this->SetX((210-$w)/2);
15     // Colores de los bordes, fondo y texto
16     $this->SetDrawColor(0,80,180);
17     $this->SetFillColor(230,230,0);
18     $this->SetTextColor(220,50,50);
19     // Ancho del borde (1 mm)
20     $this->SetLineWidth(1);
21
22     $this->Cell($w,9,$title,1,1,'C',true);
23
24     $this->Ln(10);
25 }
26
27 function Footer()
28 {
29
30     $this->SetY(-15);
31
32     $this->SetFont('Arial','I',8);
33     // Color del texto en gris
34     $this->SetTextColor(128);
35
36
37 }
38
```

www.pacoarce.com



```
39 function ChapterTitle($num, $label)
40 {
41     // Arial 12
42     $this->SetFont('Arial','',12);
43     // Color de fondo
44     $this->SetFillColor(200,220,255);
45
46
47
48     $this->Ln(4);
49 }
50
51 function ChapterBody($file)
52 {
53     // Leemos el fichero
54     $txt = file_get_contents($file);
55     // Times 12
56     $this->SetFont('Times','',12);
57     // Imprimimos el texto justificado
58     $this->MultiCell(0,5,$txt);
59
60     $this->Ln();
61
62     $this->SetFont('','I');
63     $this->Cell(0,5,'(fin del extracto)');
64 }
65
66 function PrintChapter($num, $title, $file)
67 {
68     $this->AddPage();
69     $this->ChapterTitle($num,$title);
70     $this->ChapterBody($file);
71 }
72 }
73
74 $pdf = new PDF();
75 $title = '20000 Leguas de Viaje Submarino';
76 $pdf->SetTitle($title);
77 $pdf->SetAuthor('Julio Verne');
78 $pdf->PrintChapter(1,'UN RIZO DE HUIDA','20k_c1.txt');
79 $pdf->PrintChapter(2,'LOS PROS Y LOS CONTRAS','20k_c2.txt');
80 $pdf->Output();
81 ?>
```

## Listado 20.3.1. fpdf03.php

## 20.4. Manejo de columnas en texto con FPDF

### Controlar el salto de línea

boolean AcceptPageBreak()

- En cualquier momento que un salto de página es encontrado, es invocado el método y el salto es emitido o no dependiendo del valor retornado.
- La implementación por defecto retorna un valor de acuerdo a el modo seleccionado por SetAutoPageBreak().
- Este método es invocado automáticamente y no debería ser llamado directamente por la aplicación.

### Definir el margen izquierdo

SetLeftMargin(float margin)

- Define el margen izquierdo.
- El método puede ser invocado antes de la creación de la primera página.
- Si la abscisa actual está fuera del rango de la página, esta es traída a la margen.

### Recuperar coordenada X

float GetX()

Retorna la abscisa de la posición actual.

---

```
1  <?php
2  require('fpdf181/fpdf.php');
3
4  class PDF extends FPDF
5  {
6      protected $col = 0; // Columna actual
7      protected $y0; // Ordenada de comienzo de la columna
8
9      function Header(){
10         // Cabecera
11         global $title;
12
13         $this->SetFont('Arial','B',15);
14         $w = $this->GetStringWidth($title)+6;
15         $this->SetX((210-$w)/2);
16         $this->SetDrawColor(0,80,180);
17         $this->SetFillColor(230,230,0);
18         $this->SetTextColor(220,50,50);
19         $this->SetLineWidth(1);
20         $this->Cell($w,9,$title,1,1,'C',true);
21         $this->Ln(10);
22         // Guardar ordenada (y)
23         $this->y0 = $this->GetY();
24     }
25
26     function Footer(){
27
28         $this->SetY(-15);
29         $this->SetFont('Arial','I',8);
30         $this->SetTextColor(128);
```

```
31
32 }
33
34 function SetCol($col){
35
36     $this->col = $col;
37
38     $x = 10+$col*65;
39     $this->SetLeftMargin($x);
40     $this->SetX($x);
41 }
42
43 function AcceptPageBreak(){
44
45     // Tres columnas
46     if($this->col<2){
47         // Ir a la siguiente columna
48         $this->SetCol($this->col+1);
49         // Establecer la ordenada al principio
50         $this->SetY($this->y0);
51
52         return false;
53     } else {
54         // Volver a la primera columna
55         $this->SetCol(0);
56
57         return true;
58     }
59 }
60
61 function ChapterTitle($num, $label){
62
63     $this->SetFont('Arial','',12);
64     $this->SetFillColor(200,220,255);
65
66     $this->Ln(4);
67     // Guardar ordenada
68     $this->y0 = $this->GetY();
69 }
70
71 function ChapterBody($file){
72     // Abrir fichero de texto
73     $txt = file_get_contents($file);
74     // Fuente
75     $this->SetFont('Times','',12);
76     // Imprimir texto en una columna de 6 cm de ancho
77     // MultiCell(float w, float h, string txt [, mixed border [, string align [,
boolean fill]])
78     $this->MultiCell(60,5,$txt);
79
80     $this->Ln();
81
```

```
82     $this->SetFont('', 'I');
83     $this->Cell(0,5,'(fin del extracto)');
84     // Volver a la primera columna
85     $this->SetCol(0);
86 }
87
88 function PrintChapter($num, $title, $file){
89
90     $this->AddPage();
91     $this->ChapterTitle($num,$title);
92     $this->ChapterBody($file);
93 }
94 }
95
96 $pdf = new PDF();
97 $title = '20000 Leguas de Viaje Submarino';
98 $pdf->SetTitle($title);
99 $pdf->SetAuthor('Julio Verne');
100 $pdf->PrintChapter(1,'UN RIZO DE HUIDA','20k_c1.txt');
101 $pdf->PrintChapter(2,'LOS PROS Y LOS CONTRAS','20k_c2.txt');
102 $pdf->Output();
103 ?>
```

**Listado 20.4.1.** fpdf04.php

*www.pacoarce.com*

## 20.5. Crear una tabla dinámica con FPDF y PHP

Crear tablas

```
Cell(float w  
[, float h  
[, string txt  
[, mixed border  
[, int ln  
[, string align  
[, boolean fill  
[, mixed link]]]]]]))
```

### **border**

Indica si los bordes deben ser dibujados alrededor de la celda. El valor puede ser un número:

0: sin borde

1: marco

o una cadena que contenga una o una combinación de los siguientes caracteres (en cualquier orden):

L: izquierda

T: superior

R: derecha

B: inferior

### **ln**

Indica donde la posición actual debería ir antes de invocar. Los valores posibles son:

0: a la derecha

1: al comienzo de la siguiente línea

2: debajo

Poner 1 es equivalente a poner 0 y llamar justo después Ln(). Valor por defecto: 0.

### **align**

Permite centrar o alinear el texto. Los posibles valores son:

L o una cadena vacía: alineación izquierda (valor por defecto)

C: centro

R: alineación derecha

```
1  <?php  
2  require( 'fpdf181/fpdf.php' );  
3  
4  class PDF extends FPDF  
5  {
```

```
6 // Cargar los datos
7 function CargaDatos($file){
8
9     $lines = file($file);
10    $data = array();
11    //
12    foreach($lines as $line){
13        $data[] = explode(';',trim($line));
14    }
15    //
16    return $data;
17 }
18
19 // Tabla simple
20 function BasicTable($header, $data){
21     // Cabecera
22     foreach($header as $col){
23         $this->Cell(40,7,$col,1);
24     }
25     //
26     $this->Ln();
27     // Datos
28     foreach($data as $row)
29     {
30         foreach($row as $col){
31             $this->Cell(40,6,$col,1);
32         }
33         $this->Ln();
34     }
35 }
36
37
38 function ImprovedTable($header, $data){
39     // Anchuras de las columnas
40     $w = array(40, 35, 45, 40);
41     // Cabeceras
42     for($i=0;$i<count($header);$i++){
43         $this->Cell($w[$i],7,$header[$i],1,0,'C');
44     }
45     //
46     $this->Ln();
47     // Datos
48     foreach($data as $row)
49     {
50         $this->Cell($w[0],6,$row[0],'LR');
51         $this->Cell($w[1],6,$row[1],'LR');
52         $this->Cell($w[2],6,number_format($row[2]),'LR',0,'R');
53         $this->Cell($w[3],6,number_format($row[3]),'LR',0,'R');
54         $this->Ln();
55     }
56
57     $this->Cell(array_sum($w),0,'','T');
```

```
58     }
59
60     // Tabla coloreada
61     function FancyTable($header, $data){
62
63         $this->SetFillColor(255,0,0); //Rojo
64         $this->SetTextColor(255); //Blanco
65         $this->SetDrawColor(128,0,0); //otro rojo
66         $this->SetLineWidth(.3);
67         $this->SetFont('', 'B');
68         // Cabecera
69         $w = array(40, 35, 45, 40);
70         for($i=0;$i<count($header);$i++){
71             $this->Cell($w[$i],7,$header[$i],1,0,'C',true);
72         }
73         $this->Ln();
74
75         $this->SetFillColor(224,235,255);
76         $this->SetTextColor(0);
77         $this->SetFont('');
78         // Datos
79         $fill = false;
80         foreach($data as $row)
81         {
82             $this->Cell($w[0],6,$row[0],'LR',0,'L',$fill);
83             $this->Cell($w[1],6,$row[1],'LR',0,'L',$fill);
84             $this->Cell($w[2],6,number_format($row[2]),'LR',0,'R',$fill);
85             $this->Cell($w[3],6,number_format($row[3]),'LR',0,'R',$fill);
86             $this->Ln();
87             $fill = !$fill;
88         }
89
90         $this->Cell(array_sum($w),0,'','T');
91     }
92 }
93
94 $pdf = new PDF();
95
96
97 // Carga de datos
98 $data = $pdf->CargaDatos('países.txt');
99 $pdf->SetFont('Arial','',14);
100 //
101 $pdf->AddPage();
102 $pdf->BasicTable($header,$data);
103 //
104 $pdf->AddPage();
105 $pdf->ImprovedTable($header,$data);
106 //
107 $pdf->AddPage();
108 $pdf->FancyTable($header,$data);
109 //
```

```
110 $pdf->Output();  
111 ?>
```

**Listado 20.5.1.** fpdf05.php



*www.pacoarce.com*



# Todo PHP7

## De novato a experto

### Capítulo 21: Manejar sesiones con PHP

**Objetivo:** El alumno creará y eliminará una sesión en PHP7.

#### Introducción:

Las **sesiones** son una excelente herramienta para transmitir información de una página a otra, mucha más segura que las galletas o cookies. Los temas que veremos en esta sección son:

- Crear páginas privadas con `session_start()`
- Validar y cerrar una sesión
- Limitar el tiempo de las sesiones con el comando `time()`
- PHP7: El arreglo en la función `session_start()` .

*www.pacoarce.com*

## 21.1. Crear páginas privadas con session\_start()

Crear páginas privadas con **session\_start()**

Una herramienta fundamental que nos brinda **PHP** para hacer sitios interactivos profesionales, es el uso de sesiones. Una de las muchas aplicaciones que tenemos con las sesiones, es poder hacer páginas privadas.

En este video te mostramos cómo hacer páginas privadas utilizando las sesiones de **PHP**.

En la versión 7.0.0 se añadió el parámetro **options**.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Acceso</title>
5  </head>
6  <body>
7    <table>
8      <form action="acceso.php" method="post">
9        <tr>
10         <td>Usuario:</td>
11         <td><input type="text" name="usuario"></td>
12       </tr>
13       <tr>
14         <td>Clave de acceso:</td>
15         <td><input type="password" name="clave"></td>
16       </tr>
17       <tr>
18         <td>&nbsp;</td>
19         <td><input type="submit" value="Enviar"></td>
20       </tr>
21     </form>
22   </table>
23 </body>
24 </html>
```

**Listado 21.1.1.** index.php

```
1  <?php
2  $usuario = $_POST["usuario"];
3  $clave = $_POST["clave"];
4  if (strcmp($usuario, "pedro")==0 && strcmp($clave, "picapiedra")==0) {
5    session_start();
6    $_SESSION["acceso"]="1";
7    print "<p>Acceso aprobado</p>";
8    print "<a href='privado.php'>Acceso a información confidencial</a>";
9  } else {
10   print "Acceso no autorizado";
11 }
12 ?>
```

**Listado 21.1.2.** acceso.php



*[www.pacoarce.com](http://www.pacoarce.com)*

## 21.2. Validar y cerrar una sesión

Una vez creada la sesión, podemos validar si hay alguna variable de sesión creada, o hacer uso de la misma.

También podemos destruir la sesión y sus variables por seguridad, con la función **session\_destroy()**

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Acceso</title>
5  </head>
6  <body>
7  <table>
8    <form action="acceso.php" method="post">
9      <tr>
10       <td>Usuario:</td>
11       <td><input type="text" name="usuario"></td>
12     </tr>
13     <tr>
14       <td>Clave de acceso:</td>
15       <td><input type="password" name="clave"></td>
16     </tr>
17     <tr>
18       <td>&nbsp;</td>
19       <td><input type="submit" value="Enviar"></td>
20     </tr>
21   </form>
22 </table>
23 </body>
24 </html>
```

**Listado 21.2.1.** index.php

---

```
1  <?php
2  $usuario = $_POST["usuario"];
3  $clave = $_POST["clave"];
4  if (strcmp($usuario, "pedro")==0 && strcmp($clave, "picapiedra")==0) {
5    session_start();
6    $_SESSION["acceso"]="1";
7    print "<p>Acceso aprobado</p>";
8    print "<a href='privado.php'>Acceso a información confidencial</a>";
9  } else {
10   print "Acceso no autorizado";
11  }
12  ?>
```

**Listado 21.2.2.** acceso.php

```
1  <?php
2  session_start();
3  if ($_SESSION["acceso"]!="1") {
4      header("location:index.php");
5      exit;
6  }
7  ?>
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <title>Información ultra confidencial</title>
12 </head>
13 <body>
14 <h1>Bienvenido a la información ultra confidencial</h1>
15 <a href="index.php">Regresar</a>
16 <a href="salida.php">Salir de la sesión</a>
17 </body>
18 </html>
```

## Listado 21.2.3. privado.php

---

```
1  <?php
2  session_start();
3  if (isset($_SESSION["acceso"])) {
4      session_destroy();
5      unset($_SESSION["acceso"]);
6      print utf8_encode("Haz abandonado la sesión");
7  } else {
8      print utf8_encode("Lo siento, no hay sesión abierta.");
9  }
10 print "<br>";
11 print "<a href='index.php'>Regresar</a>";
12 ?>
```

## Listado 21.2.4. salida.php

[www.pacoarce.com](http://www.pacoarce.com)

## 21.3. Limitar el tiempo de las sesiones con el comando time()

Limitar el tiempo de las sesiones con el comando **time()**

Una vez que tenemos nuestras páginas con sesiones, nos será muy fácil el poder limitarlas por tiempo sin actividad por medio del comando **time()**.

---

```
1  <?php
2  include "libs/sesion.php";
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7    <title>Información ultra confidencial</title>
8  </head>
9  <body>
10 <h1>Las fotos prohibidas</h1>
11 <a href="index.php">Regresar</a>
12 <a href="privado.php">Información ultra secreta.</a>
13 <a href="salida.php">Salir de la sesión</a>
14 </body>
15 </html>
```

**Listado 21.3.1.** fotos.php

---

```
1  <?php
2  $limite = 5;
3  session_start();
4  if (time()-$_SESSION["acceso"]>$limite) {
5    header("location:index.php?error=true");
6    exit;
7  } else if (isset($_SESSION["acceso"])==false) {
8    header("location:index.php");
9    exit;
10 } else {
11   $_SESSION["acceso"] = time();
12 }
13 ?>
```

**Listado 21.3.2.** sesion.php

www.pacoarce.com

## 21.4. PHP7: El arreglo en la función `session_start()`

**`session_start()`** ahora acepta un arreglo de opciones que sobrescriben las directivas de configuración de sesiones establecidas normalmente en *php.ini*.

Estas opciones también se han ampliado para admitir **`session.lazy_write`**, la cual está activada de forma predeterminada y causa que **PHP** solamente sobre escriba cualquier fichero de sesión si los datos de sesión han cambiado, y **`read_and_close`**, la cual es una opción que se puede pasar solo a **`session_start()`** para indicar que los datos de sesión deberían ser leídos y luego la sesión debería ser cerrada inmediatamente sin cambios.

Por ejemplo, para establecer **`session.cache_limiter`** a **`private`** e inmediatamente cerrar la sesión después de leerla.

---

```
1 <?php
2 session_start([
3 'cache_limiter' => 'private',
4 'read_and_close' => true,
5 ]);
6 ?>
```

**Listado 21.4.1.** sessionStartArreglo.php

[www.pacoarce.com](http://www.pacoarce.com)

# Todo PHP7

## De novato a experto

### Capítulo 22: Las expresiones regulares en PHP

**Objetivo:** El alumno realizará expresiones regulares en PHP.

#### Introducción:

- Expresiones regulares y PHP
- Validar un correo electrónico con expresiones regulares
- PHP7: `preg_replace_callback_array()`



*[www.pacoarce.com](http://www.pacoarce.com)*



## 22.1. Expresiones regulares y PHP

Las expresiones regulares son una poderosa herramienta para validar cadenas.

En esta clase veremos cuáles tipos de expresiones regulares y la forma de utilizarlas en PHP.

int **preg\_match** ( string \$pattern , string \$subject [, array &\$matches [, int \$flags = 0 [, int \$offset = 0 ]]] );

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Expresiones regulares</title>
5  </head>
6  <body>
7    <?php
8      $cadena1 = "679094495987123";
9      $cadena2 = "$678,,,7,8.9....00";
10     $regex = "/^[[:digit:]]+$/";
11
12     if (preg_match($regex, $cadena1)) {
13       print "<p>La cadena ".$cadena1." sólo contiene números</p>";
14     } else {
15       print "<p>La cadena ".$cadena1." contiene caracteres no numéricos</p>";
16     }
17
18     if (preg_match($regex, $cadena2)) {
19       print "<p>La cadena ".$cadena2." sólo contiene números</p>";
20     } else {
21       print "<p>La cadena ".$cadena2." contiene caracteres no numéricos</p>";
22     }
23
24     ?>
25   </body>
26 </html>
```

**Listado 22.1.1.** regex.php

www.pacoarce.com

## 22.2. Validar un correo electrónico con expresiones regulares

Una de las tareas más importantes y más tediosas de la programación es la validación de los datos de entrada.

Una forma de aligerarnos ese trabajo es por medio de las **expresiones regulares**.

En esta clase te decimos cómo validar un correo electrónico con **expresiones regulares**.

---

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Validar un correo electrónico con expresiones regulares</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <?php
9    function validaCorreo($correo){
10      $r =
10 preg_match('/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,5})$/',$
    $correo);
11    return $r;
12  }
13  $correo = "francisco.arce@yahoo.com.mx";
14  if(validaCorreo($correo)){
15    print "El correo '". $correo.'" es correcto";
16  } else {
17    print "El correo '". $correo.'" es incorrecto";
18  }
19  ?>
20 </body>
21 </html>
```

**Listado 22.2.1.** regexCorreo.php

www.pacoarce.com

## 22.3. PHP7: preg\_replace\_callback\_array()

Con la nueva función **preg\_replace\_callback\_array()** , sustituye a la función **preg\_**

Su sintaxis es:

```
mixed preg_replace_callback_array (
array $patterns_and_callbacks ,
mixed $subject
[, int $limit = -1
[, int &$amp;count ]] )
```

**preg\_replace\_callback\_array()**

**patterns\_and\_callbacks** : Un arreglo asociativo con correspondencias entre patrones regex (claves) y funciones de callback (valores).

**subject**: Cadena o arreglo con varias cadenas para la búsqueda y sustitución.

**limit**: Las sustituciones máximas posibles para cada patrón en cada string de subject. Su valor predeterminado es de -1 (sin límite).

**count**: Si se especifica, esta variable contendrá el número de sustituciones.

**preg\_replace\_callback\_array()** devuelve un arreglo si el parámetro "**subject**" es un arreglo, o una cadena de lo contrario. En caso de error devuelve *NULL*.

---

```
1  <?php
2  $cadena = "AaaaAa BBBbbbbbb CCCccccc";
3
4  preg_replace_callback_array([
5      '~[a]+~i' => function($match){
6          print strlen($match[0])." letras 'a' en la cadena."<br>";
7      },
8      '~[b]+~i' => function($match){
9          print strlen($match[0])." letras 'b' en la cadena."<br>";
10     },
11     '~[c]+~i' => function($match){
12         print strlen($match[0])." letras 'c' en la cadena."<br>";
13     }
14 ], $cadena);
15 ?>
```

**Listado 22.3.1.** preg\_replace\_callback\_array.php

# Todo PHP7

## De novato a experto

### Capítulo 23: Depurar nuestro código con Aserciones

**Objetivo:** El alumno aprenderá a depurar su código con las aserciones y sus nuevas características con PHP7.

#### Introducción:

Por medio de las aserciones podemos depurar nuestro código de forma muy eficiente. En esta sección estudiaremos los siguientes temas:

- Aserciones
- `assert_options()` .
- Ejemplos de `assert` PHP 5 y 7
- Ejemplos de `assert` PHP 7

*www.pacoarce.com*

## 23.1. Aserciones

### Versiones: PHP4, PHP5 y PHP7:

- Las aserciones son una herramienta de depuración. No debe ser utilizada para mensajes al usuario.
- Regresan un mensaje de error cuando la sentencia es FALSA y continúa si la condición es verdadera sin mandar ningún mensaje.
- Como regla general nunca debería mandar mensajes de error o de advertencia si el código está bien programado.
- Podemos configurar la función `assert()` por medio de la función `assert_options()` o con el archivo `php.ini`.
- La directiva `ASSERT_CALLBACK` establece una función de callback para manejar las aserciones fallidas.
- Las funciones de callback de las aserciones deben aceptar los siguientes parámetros:
  - El archivo en el cual se presenta la aserción fallida.
  - La línea de la aserción fallida.
  - La expresión que falló.
  - Descripción personalizada (versión 5.4.8)

### Expectativas (solamente PHP7)

- En PHP 7 las aserciones son un constructor de clase que permite la definición de expectativas: las podemos utilizar en un ambiente de prueba (test) o desarrollo, pero desactivarlas ya en producción (costo cero o “zero cost assert”).
- En PHP 7 aún podemos utilizar la función `assert_options()`, pero podemos configurar todo desde `assert()` con los nuevos parámetros.
- Se recomienda pasar el parámetro a evaluar como cadena, ya que la podemos visualizar en la salida del `assert`.

---

```
1 <?php
2 error_reporting(E_ALL);
3 ini_set("display_errors", 1);
4
5 assert(true==false,"<<<Mi error>>>");
6
7 ?>
```

**Listado 23.1.1.** `assert.php`

*www.pacoarce.com*

## 23.2. assert\_options()

**Versiones: PHP4, PHP5 y PHP7:**

- Establecemos o recuperamos las directivas relacionadas con las aserciones.
- *Sintaxis:*
- mixed assert\_options ( int \$what [, mixed \$value ] )

**Opción: ASSERT\_ACTIVE**

- Configuración inicial: assert.active
- Valor predeterminado: 1
- Descripción: Activa la función assert()

**Opción: ASSERT\_WARNING**

- Configuración inicial: assert.warning
- Valor predeterminado: 1
- Descripción: Cada vez que una aserción falla se genera una advertencia

**Opción: ASSERT\_BAIL**

- Configuración inicial: assert.bail
- Valor predeterminado: 0
- Descripción: Termina la ejecución cuando falla una aserción

**Opción: ASSERT\_QUIET\_EVAL**

- Configuración inicial: assert.quiet\_eval
- Valor predeterminado: 0
- Descripción: Desactiva la directiva error\_reporting durante la evaluación de la aserción

**Opción: ASSERT\_CALLBACK**

- Configuración inicial: assert.callback
- Valor predeterminado: (NULL)
- Descripción: Función que se encargará de gestionar las aserciones cuando fallen

**Nota:** Paquetes de instalación, por ejemplo MAMP, tienen desactivados los mensajes de error. Los podemos activar con:

```
error_reporting(E_ALL);  
ini_set("display_errors", 1);
```

---

```
1 <?php  
2 error_reporting(E_ALL);  
3 ini_set("display_errors", 1);  
4  
5  
6 ?>
```

**Listado 23.2.1.** assert\_options.php

---

```
1 <?php  
2 error_reporting(E_ALL);  
3 ini_set("display_errors", 1);
```

```
4
5 assert_options(ASSERT_ACTIVE,1);
6 assert("true==false","***Verdadero nunca será falso***");
7 print "Ok";
8 ?>
```

**Listado 23.2.2.** assert\_options\_assert\_active.php

---

```
1 <?php
2 error_reporting(E_ALL);
3 ini_set("display_errors", 1);
4
5 assert_options(ASSERT_BAIL,1);
6 assert("true==false","***Verdadero nunca será falso***");
7 print "Ok";
8 ?>
```

**Listado 23.2.3.** assert\_options\_assert\_bail.php

---

```
1 <?php
2 error_reporting(E_ALL);
3 ini_set("display_errors", 1);
4
5 $valor = "name";
6 function x($valor){
7     is_numeric($valor);
8 }
9 function assertCallback(){
10    print "Error en la función";
11 }
12 $call = "x($valor)";
13 assert_options(ASSERT_CALLBACK,"assertCallback");
14 assert($call);
15 print "Ok";
16 ?>
```

**Listado 23.2.4.** assert\_options\_assert\_callback.php

---

```
1 <?php
2 error_reporting(E_ALL);
3 ini_set("display_errors", 1);
4
5 // $valor = "name";
6 function x($valor){
7     is_numeric($valor);
8 }
9 $call = "x($valor)";
10 assert_options(ASSERT_QUIET_EVAL,0);
11 assert($call,"***Error al procesar la funcion***");
12 print "Ok";
13 ?>
```

**Listado 23.2.5.** assert\_options\_assert\_quiet\_eval.php

---

```
1 <?php
2 error_reporting(E_ALL);
3 ini_set("display_errors", 1);
4
5 assert_options(ASSERT_WARNING,0);
6 assert("true==false","***Verdadero nunca será falso***");
7 print "Ok";
8 ?>
```

**Listado 23.2.6.** assert\_options\_assert\_warning.php



*www.pacoarce.com*



## 23.3. Ejemplos de assert PHP 5 y 7

La función **assert()** recibe un segundo parámetro en forma de cadena en la versión 5 y es “retrocompatible” en la versión 7.

Combinaremos las opciones de **assert\_options()** para obtener diferentes resultados.  
archivos/0408-PHP71. ejerciciosPHP7-2303-AssertTradicional/assertTradicionalMensaje.php no existe

---

```
1  <?php
2  assert_options(ASSERT_ACTIVE,1);
3  assert_options(ASSERT_WARNING,0);
4  assert_options(ASSERT_QUIET_EVAL,1);
5  assert_options(ASSERT_CALLBACK,"miAssertCallback");
6
7  function miAssertCallback($file, $linea, $code){
8      print "<br>Error: <br>archivo ".$file."<br>Linea: ".$linea."<br>Code: ".$code;
9  }
10 assert('mysqli_query("")');
11 ?>
```

**Listado 23.3.1.** assertTradicional.php

*www.pacoarce.com*

## 23.4. Ejemplos de assert PHP 7

### Expectativas (solamente PHP7)

- En PHP 7 las aserciones son un constructor de clase que permite la definición de expectativas: las podemos utilizar en un ambiente de prueba (test) o desarrollo, pero desactivarlas ya en producción (costo cero o "zero cost assert").
- En PHP 7 aún podemos utilizar la función **assert\_options()**, pero podemos configurar todo desde **assert()** con los nuevos parámetros.
- Se recomienda pasar el parámetro a evaluar como cadena, ya que la podemos visualizar en la salida del assert.

### zend.assertions 1

- 1: genera y ejecuta el código (development mode)
- 0: genera el código pero lo salta en tiempo de ejecución
- -1: no genera el código (production mode)

### assert.exception 0

- 1: se lanza cuando falla la aserción, ya sea arrojando el objeto proporcionado como la excepción o lanzando un nuevo objeto AssertionError si no se proporcionó la excepción
- 0: usa o genera un objeto Throwable como se describe arriba, pero solo genera una advertencia (warning) basada en ese objeto en lugar de lanzarlo (compatible con el comportamiento de PHP 5)

---

```
1  <?php
2  error_reporting(E_ALL);
3  ini_set("display_errors", 1);
4  //esta asserts activos =>1, Apagado => 0
5  ini_set("zend.assertions",1);
6  //warning => 0 fatal error => 1
7  ini_set("assert.exception",1);
8
9  class SalidaErrores extends AssertionError{}
10
11 assert("true==false", new SalidaErrores("Lo verdadero no es falso"));
12
13 print "OK";
14 ?>
```

**Listado 23.4.1.** assertPHP7.php

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 24: Crear un filtro para sanar y evaluar datos de entrada

**Objetivo:** El alumno aplicará filtros para sanar y evaluar cadenas externas, como los formularios, bases de datos, galletas, etc.

#### Introducción:

Los filtros son una excelente herramienta para sanear (quitar los caracteres "raros" o peligrosos) de una cadena y para validar los datos que recibimos en nuestro programa. Los temas que revisaremos en esta sección son:

- Introducción a los filtros: `filter_list()` y `filter_id()`
- La función `filter_var` para sanear y validar
- La función `filter_input()` para filtrar o validar variables externas
- La función `filter_var_array()` para sanear o filtrar varias variables en arreglo
- Uso de banderas en filtros
- El uso del filtro `FILTER_CALLBACK`

## 24.1. Introducción a los filtros: filter\_list() y filter\_id()

### PHP 5 >= 5.2.0, PHP 7

- Principalmente hay dos tipos de filtrado: *validación* y *saneamiento*.
- La validación se usa para validar o comprobar si los datos cumplen ciertos requisitos.
- Por ejemplo, pasándole FILTER\_VALIDATE\_EMAIL determinará si los datos son una dirección de correo válida, pero no realizará ningún cambio en los datos.

### Los filtros sirven para:

- Las entradas que recibimos de un formulario
- La información de las galletas o cookies
- Datos recibidos de un web service
- Servidor de variables
- Los datos recibidos de una base de datos

El saneamiento limpiará los datos, de modo que los modificará **eliminando los caracteres no deseados**. Por ejemplo, pasándole FILTER\_SANITIZE\_EMAIL eliminará los caracteres que no son apropiados para una dirección de correo electrónico. Sin embargo, no valida los datos.

Las banderas se usan opcionalmente tanto con la **validación** como con el **saneamiento** para adaptar el comportamiento según las necesidades.

- **filter\_has\_var** — Comprueba si existe una variable de un tipo concreto existe
- **filter\_id** — Indica el ID del nombre de filtro que se indique
- **filter\_input\_array** — Obtiene variables externas y opcionalmente las filtra
- **filter\_input** — Toma una variable externa concreta por su nombre y opcionalmente la filtra
- **filter\_list** — Devuelve una lista de todos los filtros soportados
- **filter\_var\_array** — Retorna múltiple variables y opcionalmente las filtra
- **filter\_var** — Filtra una variable con el filtro que se indique

---

```
1  <?php
2  print "<table>";
3  print "<tr>";
4  print "<th>Nombre</th>";
5  print "<th>ID</th>";
6  print "</tr>";
7  foreach (filter_list() as $id => $filtro) {
8      print "<tr>";
9      print "<td>".$filtro."</td>";
10     print "<td>".filter_id($filtro)."</td>";
11     print "</tr>";
12 }
13 print "</table>";
14 ?>
```

Listado 24.1.1. filterList.php

## 24.2. La función filter\_var para sanear y validar

### PHP 5 >= 5.2.0, PHP 7

- La función filter\_var() nos permite validar y sanar a la vez una cadena.

mixed filter\_var ( mixed \$variable [, int \$filter = FILTER\_DEFAULT [, mixed \$options ]] )

- **variable:** Valor a filtrar
- **filter:** El ID del filtro a aplicar. Si se omite, se utilizará **FILTER\_DEFAULT**, que es equivalente a **FILTER\_UNSAFE\_RAW**. Esto resultará en que no se realice ningún tipo de filtro de forma predeterminada.
- **options:** Arreglo asociativo de opciones o disyunción lógica de banderas. Si un filtro acepta opciones, se pueden indicar banderas en el campo "flags" del arreglo.

```
1  <?php
2  $correo = "fj_****(br)arce@hotmail.com";
3  print $correo."<br>";
4  $correo = filter_var($correo, FILTER_SANITIZE_EMAIL);
5  print $correo."<br>";
6  if (filter_var($correo,FILTER_VALIDATE_EMAIL)) {
7    print "El correo ".$correo." es correcto<br>";
8  } else {
9    print "El correo ".$correo." NO es correcto<br>";
10 }
11 print "<hr>";
12 $saludo = "<h1>Hola, cara de bola</h1>";
13 print $saludo."<br>";
14 $cadena = filter_var($saludo,FILTER_SANITIZE_STRING);
15 print $cadena."<br>";
16
17 print "<hr>";
18 $url = "http : // www . paco . com ";
19 $url = filter_var($url,FILTER_SANITIZE_URL);
20 print $url."<br>";
21 if (filter_var($url,FILTER_VALIDATE_URL)) {
22   print "La url ".$url." es correcta<br>";
23 } else {
24   print "La url ".$url." NO es correcta<br>";
25 }
26 print "<hr>";
27
28 ?>
```

Listado 24.2.1. filter\_var.php

## 24.3. La función `filter_input()` para filtrar o validar variables externas

**PHP 5 >= 5.2.0, PHP 7**

La función **`filter_input()`** toma una variable externa concreta por su nombre y opcionalmente la filtra.

### Descripción:

`mixed filter_input ( int $type , string $variable_name [, int $filter = FILTER_DEFAULT [, mixed $options ]] )`

### Parámetros:

- **type**: Uno de los siguientes: `INPUT_GET`, `INPUT_POST`, `INPUT_COOKIE`, `INPUT_SERVER` o `INPUT_ENV`.
- **variable\_name**: Nombre de la variable a filtrar.
- **filter**: El ID del filtro a aplicar.
- **options**: Arreglo asociativo de opciones o disyunción lógica de flags. Si el filtro acepta opciones, se pueden añadir flags en el arreglo asociativo bajo la clave "flags".

Valores devueltos: En caso de éxito, valor de la variable pedida, `FALSE` si el filtro falla o `NULL` si la variable `variable_name` no está definida.

---

```
1  <?php
2  $titulo = filter_input(INPUT_GET, "buscar", FILTER_SANITIZE_SPECIAL_CHARS);
3  $buscar = filter_input(INPUT_GET, "buscar", FILTER_SANITIZE_ENCODED);
4  print "Tu búsqueda: ".$titulo."<br>";
5  print "<a href='http://www.google.com?q=".$buscar."'>Buscar</a>";
6
7
8
9
10 ?>
```

**Listado 24.3.1.** filterInput.php

[www.pacoarce.com](http://www.pacoarce.com)

## 24.4. La función `filter_var_array()` para sanear o filtrar varias variables en arreglo

### PHP 5 >= 5.2.0, PHP 7

- La función **`filter_var_array()`** retorna múltiple variables y opcionalmente las filtra. Su sintaxis es:

`mixed filter_var_array ( array $data [, mixed $definition [, bool $add_empty = true ]] )`

- Esta función es útil para recuperar muchos valores sin llamar repetidamente a **`filter_var()`**.

### Parámetros:

**data:** Un arreglo asociativo de claves en formato cadena que contiene los datos a filtrar.

**definition:** Un arreglo definiendo los argumentos. Una clave válida será aquella que contiene una cadena con el nombre de una variable y un valor válido aquel que o bien es un tipo filter type o un arreglo especificando opcionalmente el filtro, flags y opciones.

Si el valor es un arreglo, las claves válidas serán: **filter** que especifica el tipo filter type, **flags** que define cualquier bandera que deba aplicarse a los filtros, y **options** que establece cualquier opción que se deba aplicar al filtro.

Este parámetro puede ser también un entero indicando una constante de filtro. Entonces, todos los valores en el arreglo de entrada son filtrados por este filtro.

**add\_empty:** Añade claves faltantes como NULL al valor devuelto.

### Valores devueltos:

En caso de éxito un arreglo que contiene los valores de las variables que se han pedido o FALSE en caso de fallo.

El valor del arreglo será FALSE si el filtro falla o NULL si la variable no está definida.

**FILTER\_SANITIZE\_ENCODED:**

Cadena URL-encode, opcionalmente elimina o codifica caracteres especiales.

### Banderas:

**FILTER\_FLAG\_STRIP\_LOW, FILTER\_FLAG\_STRIP\_HIGH, FILTER\_FLAG\_ENCODE\_LOW, FILTER\_FLAG\_ENCODE\_HIGH**

**FILTER\_VALIDATE\_INT:**

Valida un valor como entero, opcionalmente desde el rango especificado, y lo convierte a int en caso de éxito.

---

```
1  <?php
2  $data = array(
3      'id' => "<p>Libros</p>",
4      'cantidad' => "<h1>10</h1>",
5      'version' => "1.1.0",
6      'capitulos' => array('1','19','35'),
7      'precio' => "100.00"
```

```
8  );
9
10 $filtros = array(
11  'id' => FILTER_SANITIZE_ENCODED,
12  'cantidad' => array('filter' => FILTER_VALIDATE_INT,
13    'flags' => FILTER_FORCE_ARRAY,
14    'options' => array('min_range' => 1, 'max_range'=>15)),
15  'version' => FILTER_SANITIZE_ENCODED,
16  'noExiste' => FILTER_VALIDATE_INT,
17  'capitulos' => array('filter' => FILTER_VALIDATE_INT,
18    'flags' => FILTER_REQUIRE_ARRAY),
19  'precio' => array('filter' => FILTER_VALIDATE_INT,
20    'flags' => FILTER_REQUIRE_SCALAR)
21
22 );
23
24 $misDatos = filter_var_array($data,$filtros);
25 var_dump($misDatos);
26
27 ?>
```

**Listado 24.4.1.** filterVarArray.php

*www.pacoarce.com*



## 24.5. Uso de banderas en filtros

### Uso de banderas en filtros

- FILTER\_VALIDATE\_URL: FILTER\_FLAG\_QUERY\_REQUIRED
- FILTER\_VALIDATE\_INT: min\_range, max\_range
- FILTER\_VALIDATE\_IP: FILTER\_FLAG\_IPV6
- FILTER\_SANITIZE\_STRING: FILTER\_FLAG\_STRIP\_HIGH

```
1  <?php
2  $edad = 999;
3  $min = 1;
4  $max = 120;
5
6  if
7  (filter_var($edad,FILTER_VALIDATE_INT,array("options"=>array("min_range"=>$min,"max_range"=>$max))) {
8  } else {
9  }
10 }
11 print "<hr>";
12 $ip = "2002:GHYTG:&&%$$$53535";
13 if (filter_var($ip,FILTER_VALIDATE_IP,FILTER_FLAG_IPV6)) {
14 } else {
15 }
16 print "<hr>";
17 $url = "http://www.pacoarce.com?id=90";
18 if (filter_var($url,FILTER_VALIDATE_URL,FILTER_FLAG_QUERY_REQUIRED)) {
19 } else {
20 }
21 print "La url ".$url." es correcta, si tiene un query";
22 print "La url ".$url." NO es correcta, le falta el query";
23 }
24 print "<hr>";
25 $cadena = "<h1>Hola óíú'ñÑ;¿äÄ cara de bola";
26 $limpia = filter_var($cadena,FILTER_SANITIZE_STRING,FILTER_FLAG_STRIP_HIGH);
27 print "La cadena limpia es: ".$limpia;
28
29
30
31 ?>
```

**Listado 24.5.1.** filtrosBanderas.php

## 24.6. El uso del filtro FILTER\_CALLBACK

**FILTER\_CALLBACK:** Llamada a una función para filtrar datos. En las opciones se define la función de callback.

Como parámetro la función recibe la cadena a filtrar.

```
1  <?php
2  function validaNombre($cadena){
3      if(strpos($cadena," ")==false) return false;
4      list($apellido,$nombre) = explode(" ", $cadena,2);
5      $vacios = (empty($apellido) || empty($nombre));
6      $sonCadenas = (!is_string($apellido) || !is_string($nombre));
7      if ($vacios || $sonCadenas) {
8          return false;
9      } else {
10         return $nombre." ".$apellido;
11     }
12 }
13
14 $nombre = "Picapiedra, Pedro de Valdivia";
15 $nuevo = filter_var($nombre,FILTER_CALLBACK,array('options' => 'validaNombre'));
16 if($nuevo==false){
17     print "El nombre es incorrecto: ".$nuevo;
18 } else {
19     print "El nombre es correcto: ".$nuevo;
20 }
21
22
23
24 ?>
```

**Listado 24.6.1.** filtroCallback.php

[www.pacoarce.com](http://www.pacoarce.com)

# Todo PHP7

## De novato a experto

### Capítulo 25: Manejo de PHP, XML, MySQL y AJAX

**Objetivo:** El alumno conectará algoritmos de JavaScript, AJAX, PHP, MySQL y XML.

#### Introducción:

- Introducción a AJAX
- La instrucción XMLHttpRequest
- Crear una solicitud o
- La instrucción XMLHttpRequest
- Recuperar la información desde AJAX: responseText y responseXML
- El manejo de eventos en AJAX
- Llamar a archivos PHP desde AJAX
- Ligar el AJAX con Bases de datos (PHP+MySQL).
- Leer un archivo XML con AJAX generado desde PHP de una base de datos MySQL
- Sugerir la captura de un archivo por medio de AJAX y PHP
- Crear filtros para la lectura de una tabla en MySQL con PHP y AJAX
- Leer un archivo XML desde PHP y desplegarlo en HTML por medio de AJAX
- Crear un "livesearch" con PHP, XML y AJAX
- Crear un lector de noticias RSS con AJAX y PHP, primera parte
- Crear una encuesta con PHP y MySQL .

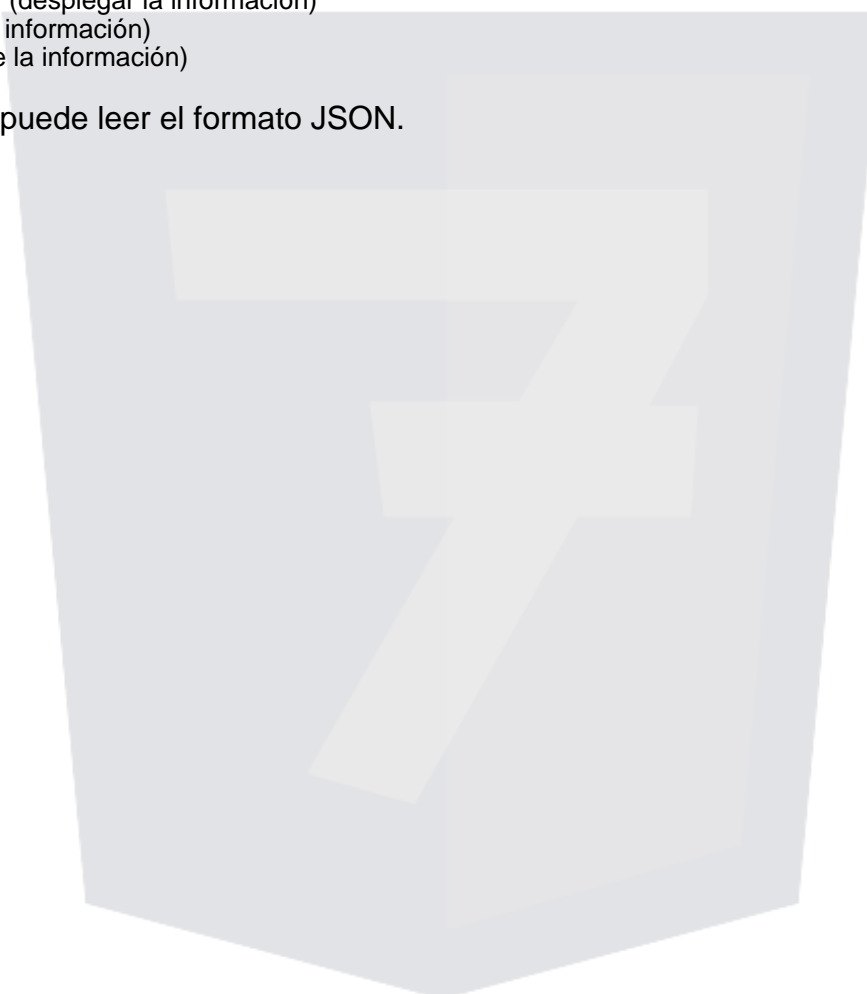
[www.pacoarce.com](http://www.pacoarce.com)

## 25.1. Introducción a AJAX

El significado de AJAX es: Asynchronous JavaScript and XML (or XHR). Para el uso de AJAX necesitamos un servidor web. Con AJAX podemos leer datos y actualizar parte de la página sin necesidad de volver a cargar toda la página. AJAX es una combinación de:

- El objeto XMLHttpRequest (leer información)
- JavaScript/DOM (desplegar la información)
- CSS (estilizar la información)
- XML (formato de la información)

**Nota:** También puede leer el formato JSON.



*www.pacoarce.com*

## 25.2. La instrucción XMLHttpRequest

Todos los navegadores modernos soportan el objeto XMLHttpRequest.

IE5 y 6 soportan ActiveXObject.

Creamos el objeto:

objeto = new XMLHttpRequest();

Para IE5 y 6 soportan

objeto = ActiveXObject("Microsoft.XMLHTTP");

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Ajax01</title>
5    <meta charset="utf-8">
6    <script>
7      window.onload = function() {
8        document.getElementById("carga").onclick = cargaArchivo;
9      }
10     function cargaArchivo(){
11       var xmlhttp;
12       if (window.XMLHttpRequest) {
13         xmlhttp = new XMLHttpRequest();
14         alert("El navegador si soporta XMLHttpRequest");
15       } else {
16         //IE 5y6
17         xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
18         alert("Tu navegador no soporta XMLHttpRequest");
19       }
20     }
21   </script>
22 </head>
23 <body>
24 <div id="salida"><h2>Pulsa el botón para leer el archivo de texto de AJAX</h2>
25 <button id="carga">Cargar archivo</button>
26 </div>
27 </body>
28 </html>
```

**Listado 25.2.1.** ajax01.html

## 25.3. Crear una solicitud o "request": xmlhttp.send()

Para enviar una solicitud al servidor necesitamos los métodos "open()" y "send()" del objeto creado anteriormente:

```
xmlhttp.open("GET","datos.txt",true);
```

```
xmlhttp.send();
```

```
open(método,url,async);
```

Método: Forma de enviar la solicitud, "GET" o "POST".

url: el archivo que abriremos.

async: true, es asíncrono, false, no asíncrono.

```
send("cadena");
```

Lanza la solicitud al servidor.

La cadena sólo se utiliza en el modo "POST".

¿GET o POST?

GET es más rápido, POST es más seguro.

POST es mejor para enviar mucha información.

Ejemplo con GET:

```
xmlhttp.open("GET","datos.php?t=" + Math.random(),true);
```

```
xmlhttp.send();
```

Ejemplo con GET:

```
xmlhttp.open("GET","datos.php?n=Ana&a=Lopez",true);
```

```
xmlhttp.send();
```

Ejemplo con POST:

```
xmlhttp.open("POST","ajax_test.asp",true);
```

```
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
```

```
xmlhttp.send("n=Ana&a=Lopez");
```

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ajax01</title>
5   <meta charset="utf-8">
6   <script>
7     window.onload = function() {
8       document.getElementById("carga").onclick = cargaArchivo;
9     }
10    function cargaArchivo(){
11      var xmlhttp;
12      if (window.XMLHttpRequest) {
13        xmlhttp = new XMLHttpRequest();
```

```
14     alert("El navegador si soporta XMLHTTP");
15 } else {
16     //IE 5y6
17     xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
18     alert("Tu navegador no soporta XMLHTTP");
19 }
20 }
21 </script>
22 </head>
23 <body>
24 <div id="salida"><h2>Pulsa el botón para leer el archivo de texto de AJAX</h2>
25 <button id="carga">Cargar archivo</button>
26 </div>
27 </body>
28 </html>
```

**Listado 25.3.1.** ajax01.html

*www.pacoarce.com*

## 25.4. La instrucción XMLHttpRequest

Como se vio en la clase anterior, los navegadores nos pueden enviar un error de **Cross Domain** si nosotros estamos trabajando en forma local tratando de acceder a AJAX, ya que se puede prestar a que nos estén "ordeñando" la máquina, entonces, por seguridad, ya casi todos los navegadores van a verificar que usted esté en un dominio de un servidor Web.

Para ello tenemos dos posibilidades. Primero podemos encontrarnos en un servidor remoto de servicio web y le van a dar su ubicación, sus claves de acceso y usted sube ahí sus datos y sus programas. en este caso nosotros vamos a trabajar en un servidor local, es decir, vamos a instalar en nuestra computadora Apache, PHP y MySQL.

Posiblemente usted ya cuente con un servidor local, pero si no lo tiene, tenemos varias opciones que son en XAMPP, que es hecho por ApacheFriends, que lo que hace es instalar Apache, PHP y MySQL (la segunda P es de Perl). Y generalmente estaban divididos por plataformas. Con XAMPP podemos instalar en las tres plataformas que son Windows, Mac y Linux. En MAMP la M es para Mac, pero ya también tiene una versión para Windows que funciona muy bien. Wamp es únicamente para Windows y LAMP para Linux. (o puede utilizar XAMPP).

Necesitamos instalar el software para que funcionen los ejercicios. En Chrome se tiene instalado el CROSS DOMAIN. Puede ser que los ejemplos que funcionen en otro navegador como **Firefox** o **Safari**, pero en **Chrome** no funcionaría. Los demás navegadores ya habían acordado en implementar este "sandbox" para segurar.

La instalación de los paquetes es muy sencillo. Búsque la dirección correspondiente para bajar el paquete de instalación. Por lo general los paquetes ya tienen la versión actualizada de PHP (7+). Posiblemente tenga problemas con los puertos. En este caso puede entrar a la configuración para cambiar los mismos.

Una vez instalado por lo general se lanza la página de configuración en su navegador por omisión. Las carpetas de servidor generalmente se llaman htdocs o www. Ahí debemos de copiar los archivos de los ejercicios y ejecutarlos desde **localhost**.

[www.pacoarce.com](http://www.pacoarce.com)



## 25.5. Recuperar la información desde AJAX:.responseText y responseXML

- Con AJAX podemos recibir los datos leídos como texto (responseText) o en formato XML (responseXML).
- También podemos leer información en formato JSON, pero debemos analizarlo o "parsearlo" manualmente.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Ajax02</title>
5    <meta charset="utf-8">
6    <script>
7      window.onload = function() {
8        document.getElementById("carga").onclick = cargaArchivo;
9      }
10     function cargaArchivo(){
11       var xmlhttp;
12       if (window.XMLHttpRequest) {
13         xmlhttp = new XMLHttpRequest();
14         alert("El navegador si soporta XMLHTTP");
15       } else {
16         //IE 5y6
17         xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
18         alert("Tu navegador no soporta XMLHTTP");
19       }
20       //GET o POST, url, true=>Asincrono false=>Sincrono
21       xmlhttp.open("GET","ajax.txt",true);
22       //Ejecutamos la lectura
23       xmlhttp.send();
24     }
25   </script>
26 </head>
27 <body>
28 <div id="salida"><h2>Pulsa el botón para leer el archivo de texto de AJAX</h2>
29 <button id="carga">Cargar archivo</button>
30 </div>
31 </body>
32 </html>
```

**Listado 25.5.1.** ajax02.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Ajax03</title>
5    <meta charset="utf-8">
6    <script>
7      window.onload = function() {
```

```
8     document.getElementById("carga").onclick = cargaArchivo;
9 }
10 function cargaArchivo(){
11     var xmlhttp;
12     if (window.XMLHttpRequest) {
13         xmlhttp = new XMLHttpRequest();
14         //alert("El navegador si soporta XMLHTTP");
15     } else {
16         //IE 5y6
17         xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
18         alert("Tu navegador no soporta XMLHTTP");
19     }
20     //GET o POST, url, true=>Asincrono false=>Sincrono
21     xmlhttp.open("GET","ajax.txt",true);
22     //Ejecutamos la lectura
23     xmlhttp.send();
24     //Vaciamos la información en salida
25     document.getElementById("salida").innerHTML = xmlhttp.responseText;
26 }
27 </script>
28 </head>
29 <body>
30 <div id="salida"><h2>Pulsa el botón para leer el archivo de texto de AJAX</h2>
31 <button id="carga">Cargar archivo</button>
32 </div>
33 </body>
34 </html>
```

## Listado 25.5.2. ajax03.html

---

1 Bienvenido al mundo de AJAX

## Listado 25.5.3. ajax.txt

*www.pacoarce.com*

## 25.6. El manejo de eventos en AJAX

### xmlhttp.onreadystatechange()

Este evento nos indica cuando el proceso inicia, su avance, su finalización y el estado del mismo, si fue correcta o existió un error.

Un evento muy importante en el control del proceso de AJAX, es el evento:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ajax03</title>
5   <meta charset="utf-8">
6   <script>
7     window.onload = function() {
8       document.getElementById("carga").onclick = cargaArchivo;
9     }
10    function cargaArchivo(){
11      var xmlhttp;
12      if (window.XMLHttpRequest) {
13        xmlhttp = new XMLHttpRequest();
14        //alert("El navegador si soporta XMLHTTP");
15      } else {
16        //IE 5y6
17        xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
18        alert("Tu navegador no soporta XMLHTTP");
19      }
20      //GET o POST, url, true=>Asincrono false=>Sincrono
21      xmlhttp.open("GET","ajax.txt",true);
22      //Ejecutamos la lectura
23      xmlhttp.send();
24      //Vaciamos la información en salida
25      document.getElementById("salida").innerHTML = xmlhttp.responseText;
26    }
27  </script>
28 </head>
29 <body>
30 <div id="salida"><h2>Pulsa el botón para leer el archivo de texto de AJAX</h2>
31 <button id="carga">Cargar archivo</button>
32 </div>
33 </body>
34 </html>
```

#### Listado 25.6.1. ajax03.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ajax04</title>
5   <meta charset="utf-8">
```

```
6 <script>
7   window.onload = function() {
8     document.getElementById("carga").onclick = cargaArchivo;
9   }
10  function cargaArchivo(){
11    var xmlhttp;
12    if (window.XMLHttpRequest) {
13      xmlhttp = new XMLHttpRequest();
14      //alert("El navegador si soporta XMLHTTP");
15    } else {
16      //IE 5y6
17      xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
18      alert("Tu navegador no soporta XMLHTTP");
19    }
20    xmlhttp.onreadystatechange = function(){
21      console.log(xmlhttp.readyState,xmlhttp.status);
22      if(xmlhttp.readyState==4){
23        if(xmlhttp.status==200){
24          //Vaciamos la información en salida
25          document.getElementById("salida").innerHTML=xmlhttp.responseText;
26        } else {
27          document.getElementById("salida").innerHTML="<p>Error en el proceso de
lectura del archivo. Error "+xmlhttp.status;
28        }
29      }
30    }
31    //GET o POST, url, true=>Asincrono false=>Sincrono
32    xmlhttp.open("GET","ajax.txt",true);
33    //Ejecutamos la lectura
34    xmlhttp.send();
35  }
36 </script>
37 </head>
38 <body>
39 <div id="salida"><h2>Pulsa el botón para leer el archivo de texto de AJAX</h2>
40 <button id="carga">Cargar archivo</button>
41 </div>
42 </body>
43 </html>
```

## Listado 25.6.2. ajax04.html

---

1 Bienvenido al mundo de AJAX

## Listado 25.6.3. ajax.txt

## 25.7. Llamar a archivos PHP desde AJAX

- Desde PHP recibimos los parámetros como GET y POST y los podemos regresar, por medio de PRINT o ECHO en formato de texto, XML o JSON.
- Desde AJAX podemos llamar un recurso como GET, pasando los parámetros en la URL, o como POST.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Ajax05 conexion con PHP</title>
5    <meta charset="utf-8">
6    <script>
7      function listaPaises(p){
8        if (p.length==0) {
9          document.getElementById("sugerencias").innerHTML = "";
10         return;
11       }
12       var xmlhttp;
13       if (window.XMLHttpRequest) {
14         xmlhttp = new XMLHttpRequest();
15         //alert("El navegador si soporta XMLHTTP");
16       } else {
17         //IE 5y6
18         xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
19         alert("Tu navegador no soporta XMLHTTP");
20       }
21       xmlhttp.onreadystatechange = function(){
22         //console.log(xmlhttp.readyState,xmlhttp.status);
23         if(xmlhttp.readyState==4){
24           if(xmlhttp.status==200){
25             //Vaciamos la información en salida
26             document.getElementById("sugerencias").innerHTML=xmlhttp.responseText;
27           } else {
28             document.getElementById("salida").innerHTML="<p>Error en el proceso de
lectura del archivo. Error "+xmlhttp.status;
29           }
30         }
31       }
32     }
33     //GET o POST, url, true=>Asincrono false=>Sincrono
34     xmlhttp.open("GET","paises.php?q="+p,true);
35     //Ejecutamos la lectura
36     xmlhttp.send();
37   }
38 </script>
39 </head>
40 <body>
41 País: <input type="text" name="pais" id="pais" onKeyUp="listaPaises(this.value)"/>
```

```
42 <p>Sugerencias: <span id="sugerencias"></span><p>
43 </div>
44 </body>
45 </html>
```

## Listado 25.7.1. ajax05.html

---

```
1 <?php
2 $cadena = $_GET["q"];
3 $paises = array (
4 'AF' => 'Afghanistan',
5 'AL' => 'Albania',
6 'DZ' => 'Algeria',
7 'AS' => 'American Samoa',
8 'AD' => 'Andorra',
9 'AO' => 'Angola',
10 'AI' => 'Anguilla',
11 'AQ' => 'Antarctica',
12 'AG' => 'Antigua and Barbuda',
13 'AR' => 'Argentina',
14 'AM' => 'Armenia',
15 'AW' => 'Aruba',
16 //
17 //El archivo completo lo puede encontrar en el curso
18 //
19 );
20 foreach($paises as $p){
21 if(strpos($p, $cadena)>-1){
22     print "<br>".$p;
23 }
24 }
25 ?>
```

## Listado 25.7.2. paises.php

[www.pacoarce.com](http://www.pacoarce.com)

## 25.8. Ligar el AJAX con Bases de datos (PHP+MySQL)

Podemos conectar AJAX a una base de datos por medio de un lenguaje intermedio o de servidor como PHP o ASP. En este caso, AJAX llama a un script de PHP, le pasa parámetros ya sea como GET o POST. PHP toma los parámetros y crea y ejecuta el query. Luego lo "imprime" con lo cual lo regresa a AJAX, ya sea como texto, XML o JSON.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Ajax06 conexion con PHP a la base de datos</title>
5 <meta charset="utf-8">
6 <script>
7     function buscaAlumnos(p){
8         if (p.length==0) {
9             document.getElementById("alumnos").innerHTML = "";
10            return;
11        }
12        var xmlhttp;
13        if (window.XMLHttpRequest) {
14            xmlhttp = new XMLHttpRequest();
15            //alert("El navegador si soporta XMLHTTP");
16        } else {
17            //IE 5y6
18            xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
19            alert("Tu navegador no soporta XMLHTTP");
20        }
21        xmlhttp.onreadystatechange = function(){
22            //console.log(xmlhttp.readyState,xmlhttp.status);
23            if(xmlhttp.readyState==4){
24                if(xmlhttp.status==200){
25                    //Vaciamos la información en salida
26                    document.getElementById("alumnos").innerHTML=xmlhttp.responseText;
27                } else {
28                    document.getElementById("alumnos").innerHTML="<p>Error en el proceso de
lectura del archivo. Error "+xmlhttp.status;
29                }
30            }
31        }
32    }
33    //GET o POST, url, true=>Asincrono false=>Sincrono
34    xmlhttp.open("GET","alumnos.php?q="+p,true);
35    //Ejecutamos la lectura
36    xmlhttp.send();
37 }
38 </script>
39 </head>
40 <body>
```

```
41 <button onclick="buscaAlumnos('A')">A</button>
42 <button onclick="buscaAlumnos('B')">B</button>
43 <button onclick="buscaAlumnos('C')">C</button>
44 <button onclick="buscaAlumnos('D')">D</button>
45 <button onclick="buscaAlumnos('E')">E</button>
46 <button onclick="buscaAlumnos('F')">F</button>
47 <p>Alumnos:<br><span id="alumnos"></span></p>
48 </div>
49 </body>
50 </html>
```

## Listado 25.8.1. ajax06.html

---

```
1 <?php
2 $q = $_GET["q"];
3 $usuario = "root";
4 $pwd = "root";
5 $servidor = "localhost";
6 $db = "escuela";
7 $puerto = "3308"; //MAMP con windows
8 //
9 $conn = mysqli_connect($servidor, $usuario, $pwd, $db, $puerto) or die ("Error en
la conexión de la base de datos");
10 $sql = "SELECT * FROM alumnos WHERE nombres regexp '^".$q."' ";
11 $r = mysqli_query($conn, $sql);
12 //
13 while ($data = mysqli_fetch_object($r)) {
14     $nombres = $data->nombres;
15     $apellidos = $data->apellidos;
16     $promedio = $data->promedio;
17     print $nombres.", ".$apellidos.", promedio: ".$promedio."<br>";
18 }
19 mysqli_close($conn);
20 ?>
```

## Listado 25.8.2. alumnos.php

---

```
1 -- phpMyAdmin SQL Dump
2 -- version 4.4.15.5
3 -- http://www.phpmyadmin.net
4 --
5 -- Host: localhost:3308
6 -- Generation Time: Jun 16, 2018 at 09:20 PM
7 -- Server version: 5.5.49-log
8 -- PHP Version: 7.0.10
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
```



# Todo PHP7

```
17 /*!40101 SET NAMES utf8mb4 */;
18
19 --
20 -- Database: `escuela`
21 --
22
23 -- -----
24
25 --
26 -- Table structure for table `alumnos`
27 --
28
29 CREATE TABLE IF NOT EXISTS `alumnos` (
30   `id` int(11) NOT NULL,
31   `nombres` varchar(200) COLLATE latin1_spanish_ci NOT NULL,
32   `apellidos` varchar(200) COLLATE latin1_spanish_ci NOT NULL,
33   `fechaNacimiento` varchar(12) COLLATE latin1_spanish_ci NOT NULL,
34   `promedio` int(11) NOT NULL,
35   `genero` char(1) COLLATE latin1_spanish_ci NOT NULL,
36   `escuela` int(11) NOT NULL,
37   `materia` int(11) NOT NULL,
38   `salon` int(11) NOT NULL
39 ) ENGINE=InnoDB AUTO_INCREMENT=106 DEFAULT CHARSET=latin1
  COLLATE=latin1_spanish_ci;
40
41 --
42 -- Dumping data for table `alumnos`
43 --
44
45 INSERT INTO `alumnos` (`id`, `nombres`, `apellidos`, `fechaNacimiento`, `promedio`,
  `genero`, `escuela`, `materia`, `salon`) VALUES
46 (1, 'Adrian Jesus', 'Escobar Diaz', '01/11/99', 4, 'M', 1, 2, 3),
47 (2, 'Adriana', 'Aguilar Tellez', '02/12/98', 5, 'F', 2, 1, 2),
48 -- El archivo completo lo encontrará en el curso
49 --
50 -- Indexes for dumped tables
51 --
52
53 --
54 -- Indexes for table `alumnos`
55 --
56 ALTER TABLE `alumnos`
57 ADD PRIMARY KEY (`id`);
58
59 --
60 -- AUTO_INCREMENT for dumped tables
61 --
62
63 --
64 -- AUTO_INCREMENT for table `alumnos`
65 --
66 ALTER TABLE `alumnos`
```

```
67 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=106;  
68 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
69 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
70 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

**Listado 25.8.3.** alumnos.sql



*www.pacoarce.com*

## 25.9. Leer un archivo XML con AJAX generado desde PHP de una base de datos MySQL 1a

Ya que hemos recibido el archivo desde el servidor en el cliente, tenemos que “analizarlo” con las herramientas propias de JavaScript, el cual cuenta con una clase para ello.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Ajax07 conexion con PHP a la base de datos y el formato XML</title>
5    <meta charset="utf-8">
6    <script>
7      function buscaAlumnos(p){
8        if (p.length==0) {
9          document.getElementById("alumnos").innerHTML = "";
10         return;
11        }
12        var xmlhttp;
13        if (window.XMLHttpRequest) {
14          xmlhttp = new XMLHttpRequest();
15          //alert("El navegador si soporta XMLHTTP");
16        } else {
17          //IE 5y6
18          xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
19          alert("Tu navegador no soporta XMLHTTP");
20        }
21        xmlhttp.onreadystatechange = function(){
22          //console.log(xmlhttp.readyState,xmlhttp.status);
23          if(xmlhttp.readyState==4){
24            if(xmlhttp.status==200){
25              //Vaciamos la información en salida
26              procesaArchivo(xmlhttp.responseXML);
27            } else {
28              document.getElementById("alumnos").innerHTML="<p>Error en el proceso de
lectura del archivo. Error "+xmlhttp.status;
29            }
30          }
31        }
32      }
33      //GET o POST, url, true=>Asincrono false=>Sincrono
34      xmlhttp.open("GET","alumnosXML.php?q="+p,true);
35      //Ejecutamos la lectura
36      xmlhttp.send();
37    }
38    function procesaArchivo(objetoXML){
39      alert(objetoXML);
40    }
41  </script>
42 </head>
```

```
43 <body>
44 <button onclick="buscaAlumnos('A')">A</button>
45 <button onclick="buscaAlumnos('B')">B</button>
46 <button onclick="buscaAlumnos('C')">C</button>
47 <button onclick="buscaAlumnos('D')">D</button>
48 <button onclick="buscaAlumnos('E')">E</button>
49 <button onclick="buscaAlumnos('F')">F</button>
50 <p>Alumnos:<br><span id="alumnos"></span></p>
51 </div>
52 </body>
53 </html>
```

## Listado 25.9.1. ajax07.html

```
1 <?php
2 $q = (isset($_GET["q"]))?$_GET["q"]:"";
3 $usuario = "root";
4 $pwd = "root"; //MAMP root y WAMP XAMPP esta vacío
5 $servidor = "localhost";
6 $db = "escuela";
7 $puerto = "3308"; //MAMP con windows
8 //
9 $conn = mysqli_connect($servidor, $usuario, $pwd, $db, $puerto) or die ("Error en
la conexión de la base de datos");
10 $sql = "SELECT * FROM alumnos WHERE nombres regexp '^".$q."' ";
11 $r = mysqli_query($conn, $sql);
12 //Generamos el archivo XML
13 //Encabezado
14 print header("Content-type:text/xml");
15 print "<?xml version='1.0' encoding='UTF-8'?>";
16 print "<alumnos>";
17 while ($data = mysqli_fetch_object($r)) {
18     print "<alumno>";
19     $nombres = $data->nombres;
20     $apellidos = $data->apellidos;
21     $promedio = $data->promedio;
22     print "<nombres>".$nombres."</nombres>";
23     print "<apellidos>".$apellidos."</apellidos>";
24     print "<promedio>".$promedio."</promedio>";
25     print "</alumno>";
26 }
27 print "</alumnos>";
28 mysqli_close($conn);
29 ?>
```

## Listado 25.9.2. alumnosXML.php

```
1 -- phpMyAdmin SQL Dump
2 -- version 4.4.15.5
3 -- http://www.phpmyadmin.net
4 --
5 -- Host: localhost:3308
6 -- Generation Time: Jun 16, 2018 at 09:20 PM
```

# Todo PHP7

```
7  -- Server version: 5.5.49-log
8  -- PHP Version: 7.0.10
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8mb4 */;
18
19 --
20 -- Database: `escuela`
21 --
22
23 -- -----
24
25 --
26 -- Table structure for table `alumnos`
27 --
28
29 CREATE TABLE IF NOT EXISTS `alumnos` (
30   `id` int(11) NOT NULL,
31   `nombres` varchar(200) COLLATE latin1_spanish_ci NOT NULL,
32   `apellidos` varchar(200) COLLATE latin1_spanish_ci NOT NULL,
33   `fechaNacimiento` varchar(12) COLLATE latin1_spanish_ci NOT NULL,
34   `promedio` int(11) NOT NULL,
35   `genero` char(1) COLLATE latin1_spanish_ci NOT NULL,
36   `escuela` int(11) NOT NULL,
37   `materia` int(11) NOT NULL,
38   `salon` int(11) NOT NULL
39 ) ENGINE=InnoDB AUTO_INCREMENT=106 DEFAULT CHARSET=latin1
  COLLATE=latin1_spanish_ci;
40
41 --
42 -- Dumping data for table `alumnos`
43 --
44
45 INSERT INTO `alumnos` (`id`, `nombres`, `apellidos`, `fechaNacimiento`, `promedio`,
46   `genero`, `escuela`, `materia`, `salon`) VALUES
47 (1, 'Adrian Jesus', 'Escobar Diaz', '01/11/99', 4, 'M', 1, 2, 3),
48 (2, 'Adriana', 'Aguilar Tellez', '02/12/98', 5, 'F', 2, 1, 2),
49 -- El archivo completo lo encontrará en el curso
50 --
51 -- Indexes for dumped tables
52 --
53 --
54 -- Indexes for table `alumnos`
55 --
56 ALTER TABLE `alumnos`
```

```
57 ADD PRIMARY KEY (`id`);
58
59 --
60 -- AUTO_INCREMENT for dumped tables
61 --
62 --
63 --
64 -- AUTO_INCREMENT for table `alumnos`
65 --
66 ALTER TABLE `alumnos`
67 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=106;
68 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
69 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
70 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
71
```

**Listado 25.9.3.** alumnos.sql

*www.pacoarce.com*

## 25.10. Leer un archivo XML con AJAX generado desde PHP de una base de datos MySQL

Ya que hemos recibido el archivo desde el servidor en el cliente, tenemos que “analizarlo” con las herramientas propias de JavaScript, el cual cuenta con una clase para ello.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Ajax07 conexion con PHP a la base de datos y el formato XML</title>
5    <meta charset="utf-8">
6    <script>
7      function buscaAlumnos(p){
8        if (p.length==0) {
9          document.getElementById("alumnos").innerHTML = "";
10         return;
11        }
12        var xmlhttp;
13        if (window.XMLHttpRequest) {
14          xmlhttp = new XMLHttpRequest();
15          //alert("El navegador si soporta XMLHTTP");
16        } else {
17          //IE 5y6
18          xmlhttp = new ActiveXObject("Microsoft.HMLHTTP");
19          alert("Tu navegador no soporta XMLHTTP");
20        }
21        xmlhttp.onreadystatechange = function(){
22          //console.log(xmlhttp.readyState,xmlhttp.status);
23          if(xmlhttp.readyState==4){
24            if(xmlhttp.status==200){
25              //Vaciamos la información en salida
26              procesaArchivo(xmlhttp.responseXML);
27            } else {
28              document.getElementById("alumnos").innerHTML="<p>Error en el proceso de
lectura del archivo. Error "+xmlhttp.status;
29            }
30          }
31        }
32      }
33      //GET o POST, url, true=>Asincrono false=>Sincrono
34      xmlhttp.open("GET","alumnosXML.php?q="+p,true);
35      //Ejecutamos la lectura
36      xmlhttp.send();
37    }
38    function procesaArchivo(objetoXML){
39      var cadena = "<table
border='1'><tr><th>Nombres</th><th>Apellidos</th><th>Promedio</th></tr>";
40      var a = objetoXML.documentElement.getElementsByTagName("alumno");
41      document.getElementById("alumnos").innerHTML = "Alumnos: "+a.length;
```

```
42     for (var i=0; i<a.length; i++) {
43         cadena += "<tr>";
44         nom = a[i].getElementsByTagName("nombres");
45         {
46             try{
47                 cadena += "<td>"+nom[0].firstChild.nodeValue+"</td>";
48             }
49             catch(err){
50                 cadena += "<td></td>";
51             }
52         }
53         ap = a[i].getElementsByTagName("apellidos");
54         {
55             try{
56                 cadena += "<td>"+ap[0].firstChild.nodeValue+"</td>";
57             }
58             catch(err){
59                 cadena += "<td></td>";
60             }
61         }
62         pro = a[i].getElementsByTagName("promedio");
63         {
64             try{
65                 cadena += "<td>"+pro[0].firstChild.nodeValue+"</td>";
66             }
67             catch(err){
68                 cadena += "<td></td>";
69             }
70         }
71         cadena += "</tr>";
72     }
73     cadena += "</table>";
74     document.getElementById("salida").innerHTML = cadena;
75 }
76 </script>
77 </head>
78 <body>
79 <button onclick="buscaAlumnos('A')">A</button>
80 <button onclick="buscaAlumnos('B')">B</button>
81 <button onclick="buscaAlumnos('C')">C</button>
82 <button onclick="buscaAlumnos('D')">D</button>
83 <button onclick="buscaAlumnos('E')">E</button>
84 <button onclick="buscaAlumnos('F')">F</button>
85 <p><span id="alumnos"></span></p>
86 </div>
87 <div id="salida"></div>
88 </body>
89 </html>
```

**Listado 25.10.1.** ajax07.html



## 25.11. Sugerir la captura de un archivo por medio de AJAX y PHP

Por medio de las rutinas desarrolladas en las clases anteriores, crearemos un programa que “sugiera” opciones a medida que el usuario teclee en un campo de entrada. A este proceso se le conoce como “liveSearch”.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Sugerencia en la captura</title>
5 <meta charset="utf-8">
6 <script>
7     function sugerencia(cadena){
8         if (cadena.length==0) {
9             document.getElementById("sugerencia").innerHTML = "";
10            return;
11        } else {
12            var xmlhttp = new XMLHttpRequest();
13            xmlhttp.onreadystatechange = function() {
14                if (this.readyState==4 && this.status==200) {
15                    document.getElementById("sugerencia").innerHTML = this.responseText;
16                }
17            }
18            xmlhttp.open("GET","países.php?q="+cadena,true);
19            xmlhttp.send();
20        }
21    }
22 </script>
23 </head>
24 <body>
25 <h2>Selecciona un país</h2>
26 <form>
27     País:<input type="text" onkeyup="sugerencia(this.value)"/>
28 </form>
29 <p>Sugerencia:<span id="sugerencia"></span></p>
30 </body>
31 </html>
```

**Listado 25.11.1.** países.html

```
1 <?php
2 $q = $_GET["q"];
3 $países = array (
4     'AF' => 'Afghanistan',
5     'AL' => 'Albania',
6     'DZ' => 'Algeria',
7     'AS' => 'American Samoa',
8     'AD' => 'Andorra',
```

```
9  'AO' => 'Angola',
10 'AI' => 'Anguilla',
11 'AQ' => 'Antarctica',
12 'AG' => 'Antigua and Barbuda',
13 'AR' => 'Argentina',
14 'AM' => 'Armenia',
15 'AW' => 'Aruba',
16 //El archivo completo en los archivos del curso...
17 );
18 $salida = "";
19 //validamos la cadena
20 if ($q != "") {
21 $q = strtolower($q);
22 $len = strlen($q);
23 foreach ($países as $pais) {
24 if (striestr($q, substr($pais, 0, $len))) {
25 if ($salida=="") {
26 $salida = $pais;
27 } else {
28 $salida .= ", ".$pais;
29 }
30 }
31 }
32 }
33 //regresamos la cadena de salida
34 print $salida === "" ? "No hay sugerencias" : $salida;
35 ?>
```

**Listado 25.11.2.** países.php

*www.pacoarce.com*

## 25.12. Crear filtros para la lectura de una tabla en MySQL con PHP y AJAX

Por medio de AJAX podemos recuperar la información filtrada de una tabla desde PHP. Simplemente enviaremos los parámetros, bajo el método GET, los cuales los tomaremos en PHP y los leeremos de la base de datos bajo el mismo ciclo.

```
1  <?php
2  $host = "localhost";
3  $usuario = "root";
4  $clave = "root";
5  $db = "escuela";
6  $puerto = "3308";
7  //
8  $conn = mysqli_connect($host, $usuario, $clave, $db, $puerto) or die("Error al
  conectar la base de datos");
9  if(isset($_GET["g"])){
10   $g = $_GET["g"];
11  }
12  if(isset($_GET["m"])){
13   $m = $_GET["m"];
14  }
15  if(isset($_GET["e"])){
16   $e = $_GET["e"];
17  }
18  $q = "SELECT * FROM alumnos WHERE escuela=".$e." AND genero='".$g.'" AND
  materia=".$m;
19  //print $q;
20  $r = mysqli_query($conn,$q);
21  if(mysqli_num_rows($r)==0){
22   print "No hay alumnos en esa selección";
23  } else {
24   print "<table>";
25   while ($data = mysqli_fetch_assoc($r)) {
26    print "<tr>";
27    print "<td>".$data["nombres"]."</td>";
28    print "<td>".$data["apellidos"]."</td>";
29    print "<td>".$data["fechaNacimiento"]."</td>";
30    print "<td>".$data["promedio"]."</td>";
31    print "</tr>";
32   }
33   print "</table>";
34  }
35  ?>
```

**Listado 25.12.1.** alumnos.php

```
1  <!DOCTYPE html>
2  <html>
```

```
3 <head>
4 <title>Filtra tabla de alumnos</title>
5 <meta charset="utf-8">
6 <script>
7 window.onload = function(){
8     document.getElementById("genero").onchange = function(){
9         genero = document.getElementById("genero").value;
10        escuela = document.getElementById("escuela").value;
11        materia = document.getElementById("materia").value;
12        //alert(genero+" "+escuela+" "+materia);
13        leeDatos(genero,escuela,materia);
14    }
15    document.getElementById("escuela").onchange = function(){
16        genero = document.getElementById("genero").value;
17        escuela = document.getElementById("escuela").value;
18        materia = document.getElementById("materia").value;
19        //alert(genero+" "+escuela+" "+materia);
20        leeDatos(genero,escuela,materia);
21    }
22    document.getElementById("materia").onchange = function(){
23        genero = document.getElementById("genero").value;
24        escuela = document.getElementById("escuela").value;
25        materia = document.getElementById("materia").value;
26        //alert(genero+" "+escuela+" "+materia);
27        leeDatos(genero,escuela,materia);
28    }
29 }
30 function leeDatos(genero,escuela,materia){
31     var xmlhttp = new XMLHttpRequest();
32     xmlhttp.onreadystatechange = function() {
33         if (this.readyState==4 && this.status==200) {
34             document.getElementById("salida").innerHTML = this.responseText;
35         }
36     }
37     xmlhttp.open("GET","alumnos.php?g="+genero+"&e="+escuela+"&m="+materia,true);
38     xmlhttp.send();
39 }
40 </script>
41 </head>
42 <body>
43 <form>
44     <select id="genero">
45         <option value="F">Femenino</option>
46         <option value="M">Masculino</option>
47     </select>
48     <select id="escuela">
49         <option value="1">Escuela 1</option>
50         <option value="2">Escuela 2</option>
51         <option value="3">Escuela 3</option>
52     </select>
53     <select id="materia">
54         <option value="1">Idiomas</option>
```

```
55     <option value="2">Deportes</option>
56     <option value="3">Historia</option>
57 </select>
58 </form>
59 <div id="salida"></div>
60 </body>
61 </html>
```

**Listado 25.12.2.** filtraTabla.php



*www.pacoarce.com*

## 25.13. Leer un archivo XML desde PHP y desplegarlo en HTML por medio de AJAX

Por medio de PHP leeremos un archivo XML, lo armaremos en un combobox y lo desplegamos por medio de AJAX.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Discos de ayer y hoy</title>
5    <meta charset="utf-8">
6    <script>
7      window.onload = function(){
8        discosCatalogo();
9
10     };
11     function discosCatalogo(){
12       var xmlhttp = new XMLHttpRequest();
13       xmlhttp.onreadystatechange = function(){
14         if (this.readyState == 4 && this.status == 200) {
15           document.getElementById("salida").innerHTML = this.responseText;
16         }
17       }
18       xmlhttp.open("GET","discosCatalogo.php", false);
19       xmlhttp.send();
20     }
21   </script>
22 </head>
23 <body>
24   <div id="salida"></div>
25   <br>
26   <div id="detalle"></div>
27 </body>
28 </html>
```

### Listado 25.13.1. discos.php

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <CATALOG>
3  <CD>
4    <TITLE>Empire Burlesque</TITLE>
5    <ARTIST>Bob Dylan</ARTIST>
6    <COUNTRY>USA</COUNTRY>
7    <COMPANY>Columbia</COMPANY>
8    <PRICE>10.90</PRICE>
9    <YEAR>1985</YEAR>
10 </CD>
11 <CD>
12 <TITLE>Hide your heart</TITLE>
```

```
13 <ARTIST>Bonnie Tyler</ARTIST>
14 <COUNTRY>UK</COUNTRY>
15 <COMPANY>CBS Records</COMPANY>
16 <PRICE>9.90</PRICE>
17 <YEAR>1988</YEAR>
18 </CD>
19 <CD>
20 <TITLE>Greatest Hits</TITLE>
21 <ARTIST>Dolly Parton</ARTIST>
22 <COUNTRY>USA</COUNTRY>
23 <COMPANY>RCA</COMPANY>
24 <PRICE>9.90</PRICE>
25 <YEAR>1982</YEAR>
26 </CD>
27 <!--El archivo completo lo encontrará en el curso-->
28 </CATALOG>
```

## Listado 25.13.2. discos.xml

---

```
1 <?php
2 $xmlDoc = new DOMDocument();
3 $xmlDoc -> load("discos.xml");
4 $x = $xmlDoc->getElementsByTagName("ARTIST");
5 $cds = array();
6 for($i=0; $i<$x->length;$i++){
7     //tipo 1 es de datos
8     if($x->item($i)->nodeType==1){
9         $artista = $x->item($i)->childNodes->item(0)->nodeValue;
10        array_push($cds, $artista);
11    }
12 }
13 sort($cds);
14 print "<select id='cds'>";
15 for($i=0; $i<count($cds); $i++){
16     print "<option value='". $cds[$i]. "'>";
17     print $cds[$i];
18     print "</option>";
19 }
20 print "</select>";
21 ?>
22
```

## Listado 25.13.3. discosCatalogo.php

## 25.14. Concluir la lectura de un catálogo XML con PHP y AJAX

- Posteriormente, por medio de AJAX, leeremos el detalle del registro solicitado por el usuario.
- Por medio de AJAX ejecutamos un archivo de PHP que nos enviará información de un archivo XML para crear un combo-box.

```
1  <?php
2  if(isset($_GET["q"])){
3      $q = $_GET["q"];
4      //Creamos el objeto XML
5      $xmlDoc = new DOMDocument();
6      $xmlDoc->load("discos.xml");
7      //Recupamos los nodos de artistas
8      $x = $xmlDoc->getElementsByTagName("ARTIST");
9      //Ciclo
10     for ($i=0; $i < $x->length ; $i++) {
11         //Buscamos el nodo
12         if ($x->item($i)->nodeType==1) {
13             $artista = $x->item($i)->childNodes->item(0)->nodeValue;
14             if ($q==$artista) {
15                 $y = $x->item($i)->parentNode;
16                 break;
17             }
18         }
19     }
20 }
21
22 $cd = $y->childNodes;
23 //Mostramos los datos del cd
24 for ($i=0; $i < $cd->length ; $i++) {
25     if ($cd->item($i)->nodeType==1) {
26         print "<b>".$cd->item($i)->nodeName."</b> ";
27         print $cd->item($i)->childNodes->item(0)->nodeValue;
28         print "<br>";
29     }
30 }
31 ?>
```

**Listado 25.14.1.** discosDetalle.php



## 25.15. Crear un "livesearch" con PHP, XML y AJAX, primera parte

- Por medio de AJAX ejecutamos un archivo de PHP que nos enviará información de un archivo XML para sugerencia de captura.
- Esta vez el usuario puede seleccionar la opción y abrir la página correspondiente.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Live Search</title>
5    <meta charset="utf-8">
6    <script>
7      window.onload = function(){
8        document.getElementById("texto").onkeyup = busca;
9      };
10     function busca(e){
11       cadena = this.value;
12       console.log(cadena);
13       if (cadena.length==0) {
14         document.getElementById("salida").innerHTML = "";
15         document.getElementById("salida").style.border = "0px";
16       } else {
17         var xmlhttp = new XMLHttpRequest();
18         xmlhttp.onreadystatechange = function(){
19           if (this.readyState == 4 && this.status == 200) {
20             document.getElementById("salida").innerHTML = this.responseText;
21           }
22         }
23         xmlhttp.open("GET","linksBusca.php?q="+cadena, false);
24         xmlhttp.send();
25       }
26     }
27   </script>
28 </head>
29 <body>
30 <form>
31   <input type="text" size="40" id="texto"/>
32   <div id="salida"></div>
33 </form>
34 </body>
35 </html>
```

### Listado 25.15.1. links.php

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <dioses>
3  <link>
```

```
4 <dios>Zeus</dios>
5 <url>https://es.wikipedia.org/wiki/Zeus</url>
6 </link>
7 <link>
8 <dios>Poseidón</dios>
9 <url>https://es.wikipedia.org/wiki/Poseidon</url>
10 </link>
11 <link>
12 <dios>Ares</dios>
13 <url>https://es.wikipedia.org/wiki/Ares</url>
14 </link>
15 <link>
16 <dios>Afrodita</dios>
17 <url>https://es.wikipedia.org/wiki/Afrodita</url>
18 </link>
19 <link>
20 <dios>Hera</dios>
21 <url>https://es.wikipedia.org/wiki/Hera</url>
22 </link>
23 <link>
24 <dios>Atenea</dios>
25 <url>https://es.wikipedia.org/wiki/Atenea</url>
26 </link>
27 <link>
28 <dios>Hermes</dios>
29 <url>https://es.wikipedia.org/wiki/Hermes</url>
30 </link>
31 <link>
32 <dios>Apolo</dios>
33 <url>https://es.wikipedia.org/wiki/Apolo</url>
34 </link>
35 <link>
36 <dios>Artemisa</dios>
37 <url>https://es.wikipedia.org/wiki/Artemisa</url>
38 </link>
39 <link>
40 <dios>Hefesto</dios>
41 <url>https://es.wikipedia.org/wiki/Hefesto</url>
42 </link>
43 </dioses>
```

## Listado 25.15.2. links.xml

```
1 <?php
2 $xmlDoc = new DOMDocument();
3 $xmlDoc->load("links.xml");
4 $x = $xmlDoc->getElementsByTagName("link");
5 $q = $_GET["q"];
6 //buscamos los links
7 if (strlen($q)>0) {
8     $salida = "";
9     for ($i=0; $i < $x->length ; $i++) {
```

```
10 $dios = $x->item($i)->getElementsByTagName("dios");
11 $url = $x->item($i)->getElementsByTagName("url");
12 if(stristr($dios->item(0)->childNodes->item(0)->nodeValue,$q)){
13
14 }
15 }
16 }
17 ?>
```

**Listado 25.15.3.** linksBusca.php



*www.pacoarce.com*

## 25.16. Crear un "livesearch" con PHP, XML y AJAX, segunda parte

- Por medio de AJAX ejecutamos un archivo de PHP que nos enviará información de un archivo XML para sugerencia de captura.
- Esta vez el usuario puede seleccionar la opción y abrir la página correspondiente.

```
1  <?php
2  $xmlDoc = new DOMDocument();
3  $xmlDoc->load("links.xml");
4  $x = $xmlDoc->getElementsByTagName("link");
5  $q = $_GET["q"];
6  //buscamos los links
7  if (strlen($q)>0) {
8      $salida = "";
9      for ($i=0; $i < $x->length ; $i++) {
10         $dios = $x->item($i)->getElementsByTagName("dios");
11         $url = $x->item($i)->getElementsByTagName("url");
12         if(stristr($dios->item(0)->childNodes->item(0)->nodeValue,$q)){
13             if($salida==""){
14                 $salida = "<a href='";
15             } else {
16                 $salida .= "<a href='";
17             }
18             $salida .= $url->item(0)->childNodes->item(0)->nodeValue;
19             $salida .= "' target='_blank'>";
20             $salida .= $dios->item(0)->childNodes->item(0)->nodeValue;
21             $salida .= "</a><br>";
22         }
23     }
24 }
25 //regresamos la salida
26 print ($salida=="")?"No hay sugerencia":$salida;
27 ?>
```

**Listado 25.16.1.** linksBusca.php

*www.pacoarce.com*

## 25.17. Crear un lector de noticias RSS con AJAX y PHP, primera parte

Por medio de AJAX ejecutamos un archivo de PHP que nos enviará información de un archivo para las noticias tipo RSS.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Lee noticias RSS-Reforma</title>
5    <meta charset="utf-8">
6    <script>
7      window.onload = function(){
8        document.getElementById("noticias").onchange = leeNoticias;
9      }
10     function leeNoticias(e){
11       seccion = this.value;
12       noticias = document.getElementById("noticias");
13       if(seccion==""){
14         noticias.innerHTML = "";
15       } else {
16         var xmlhttp = new XMLHttpRequest();
17         xmlhttp.onreadystatechange = function(){
18           if (this.readyState == 4 && this.status == 200) {
19             document.getElementById("noticias").innerHTML = this.responseText;
20           }
21         }
22         xmlhttp.open("GET", "noticiasRSS.php?q="+seccion, false);
23         xmlhttp.send();
24       }
25     }
26   </script>
27 </head>
28 <body>
29 <select id="noticias">
30   <option value="">Seleccione una sección</option>
31   <option value="portada">Portada</option>
32   <option value="ciudad">Ciudad</option>
33   <option value="negocios">Negocios</option>
34   <option value="cultura">Cultura</option>
35   <option value="gadgets">Gadgets</option>
36 </select>
37 <div id="noticias"></div>
38 </body>
39 </html>
40
41
```

**Listado 25.17.1.** noticias.php

```
1  <?php
2  if(isset($_GET["q"])){
3      $q = $_GET["q"];
4
5      if ($q=="portada") {
6          $xml = "https://www.reforma.com/rss/portada.xml";
7      } else {
8          # code...
9      }
10
11
12
13 }
14
15
16 ?>
```

**Listado 25.17.2.** noticiasRSS.php

*www.pacoarce.com*

## 25.18. Crear un lector de noticias RSS con AJAX y PHP, segunda parte

Por medio de AJAX ejecutamos un archivo de PHP que nos enviará información de un archivo plano para realizar una encuesta.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Lee noticias RSS-Reforma</title>
5    <meta charset="utf-8">
6    <script>
7      window.onload = function(){
8        document.getElementById("noticias").onchange = leeNoticias;
9      }
10     function leeNoticias(e){
11       seccion = this.value;
12       console.log(seccion);
13       noticias = document.getElementById("salida");
14       if(seccion==""){
15         noticias.innerHTML = "";
16       } else {
17         var xmlhttp = new XMLHttpRequest();
18         xmlhttp.onreadystatechange = function(){
19           if (this.readyState == 4 && this.status == 200) {
20             document.getElementById("salida").innerHTML = this.responseText;
21           }
22         }
23         xmlhttp.open("GET","noticiasRSS.php?q="+seccion, true);
24         xmlhttp.send();
25       }
26     }
27   </script>
28 </head>
29 <body>
30 <select id="noticias">
31   <option value="">Seleccione una sección</option>
32   <option value="portada">Portada</option>
33   <option value="ciudad">Ciudad</option>
34   <option value="negocios">Negocios</option>
35   <option value="cultura">Cultura</option>
36   <option value="gadgets">Gadgets</option>
37 </select>
38 <div id="salida"></div>
39 </body>
40 </html>
41
42
```

**Listado 25.18.1.** noticias.php

```
1  <?php
2  if(isset($_GET["q"])){
3      $q = $_GET["q"];
4
5      if ($q=="portada") {
6          $xml = "https://www.reforma.com/rss/portada.xml";
7      } elseif ($q=="negocios") {
8          $xml = "https://www.reforma.com/rss/negocios.xml";
9      } elseif ($q=="ciudad") {
10         $xml = "https://www.reforma.com/rss/ciudad.xml";
11     } elseif ($q=="cultura") {
12         $xml = "https://www.reforma.com/rss/cultura.xml";
13     } elseif ($q=="gadgets") {
14         $xml = "https://www.reforma.com/rss/gadgets.xml";
15     }
16     print "<h2>".$q."</h2>";
17     $xmlDoc = new DOMDocument();
18     $xmlDoc->load($xml);
19
20     //contenido en <channel>
21     $channel = $xmlDoc->getElementsByTagName("channel")->item(0);
22     $channel_title =
23     $channel->getElementsByTagName("title")->item(0)->childNodes->item(0)->nodeValue;
24     $channel_link =
25     $channel->getElementsByTagName("link")->item(0)->childNodes->item(0)->nodeValue;
26     $channel_desc =
27     $channel->getElementsByTagName("description")->item(0)->childNodes->item(0)->nodeValue;
28
29     //desplegamos los elementos
30     print "<p><a href='". $channel_link. "'>". $channel_title. "</a>";
31     print "<br>";
32     print $channel_desc. "</p>";
33
34     //diferentes noticias o items
35     $x=$xmlDoc->getElementsByTagName("item");
36     for ($i=0; $i < $x->length ; $i++) {
37         $item_title =
38         $x->item($i)->getElementsByTagName("title")->item(0)->childNodes->item(0)->nodeValue;
39         $item_link =
40         $x->item($i)->getElementsByTagName("link")->item(0)->childNodes->item(0)->nodeValue;
41         $item_desc =
42         $x->item($i)->getElementsByTagName("description")->item(0)->childNodes->item(0)->nodeValue;
43         //
44         print "<p><a href='". $item_link. "'>". $item_title. "</a>";
45         print "<br>";
46         print $item_desc. "</p>";
47     }
48 }
```

**Listado 25.18.2.** noticiasRSS.php



## 25.19. Crear una encuesta con PHP y MySQL

Crearemos una encuesta en línea con PHP, MySQL y AJAX, leyendo un formato XML.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Encuesta</title>
5    <meta charset="utf-8">
6    <script>
7      function votar(voto){
8        var xmlhttp = new XMLHttpRequest();
9        xmlhttp.onreadystatechange = function(){
10         if (this.readyState == 4 && this.status == 200) {
11           document.getElementById("salida").innerHTML = this.responseText;
12         }
13       }
14       xmlhttp.open("GET","encuestaVoto.php?q="+voto, true);
15       xmlhttp.send();
16     }
17   </script>
18 </head>
19 <body>
20 <h3>¿Crees que Alemania gane nuevamente el mundial?</h3>
21 Si:<input type="radio" name="voto" value="0" onclick="votar(this.value)">
22 <br>
23 No:<input type="radio" name="voto" value="1" onclick="votar(this.value)">
24 <div id="salida"></div>
25 </body>
26 </html>
```

**Listado 25.19.1.** encuesta.php

```
1  <?php
2  $voto = $_GET["q"];
3
4  //archivo
5  $archivo = "voto.txt";
6  touch($archivo);
7  $r = file($archivo);
8
9  //Contenido
10 $arreglo = explode("||",$r[0]);
11 $si = ($arreglo[0]=="")?0:$arreglo[0];
12 $no = ($arreglo[1]=="")?0:$arreglo[1];
13
14 if ($voto==0) {
15   $si++;
16 } else {
17   $no++;
```

```
18 }
19
20 $salida = $si."||".$no;
21 $fp = fopen($archivo,"w");
22 fputs($fp,$salida);
23 $total = $si+$no;
24 print "<h2>Resultado</h2>";
25 print "<table>";
26 print "<tr>";
27 print "<td>Si:</td>";
28 print "<td><img src='poll.png' width='".(100*round($si/($total),2))."'
  height='20' />";
29 print (100*round($si/($total),2))."%";
30 print "</td>";
31 print "</tr>";
32 print "<tr>";
33 print "<td>No:</td>";
34 print "<td><img src='poll.png' width='".(100*round($no/($total),2))."'
  height='20' />";
35 print (100*round($no/($total),2))."%";
36 print "</tr>";
37 print "</table>";
38 ?>
```

**Listado 25.19.2.** encuestaVoto.php

---

1 6||3

**Listado 25.19.3.** voto.txt

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 26: Crear una aplicación de altas, bajas y cambios (crud) de una tabla con PHP

**Objetivo:** El alumno creará una aplicación de alta, bajas y cambios de una tabla de MySQL con PHP

#### Introducción:

En esta sección aplicaremos todo lo estudiado en las clases anteriores para crear un módulo de Altas, Bajas y Cambios (ABC) o CRUD (Create, read, update, delete) por sus siglas en inglés.

- Subir una imagen a una base de datos MySQL con PHP
- Desplegar una imagen almacenada en MySQL desde PHP
- Crear la estructura de las zonas condicionales en el archivo PHP
- Crear una pantalla de advertencia antes de borrar el registro
- Borrar definitivamente el registro de la base de datos
- Integrar el proceso de Alta en el archivo PHP
- Cambiar la imagen y su información en la base de datos
- Modificar el aspecto de la aplicación de alta, bajas y cambios con CSS
- Bajar a un archivo en formato Base64 con PHP

[www.pacoarce.com](http://www.pacoarce.com)

## 26.1. Subir una imagen a una base de datos MySQL con PHP

Muchas veces es necesario subir una imagen de la máquina del cliente al servidor, pero por lo general no hay que subir la misma a la base de datos. En este caso subiremos la imagen (sus datos binarios) a una columna de MySQL, sin necesidad de transferir el archivo.?

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <form action="subirArchivo.php" method="post" enctype="multipart/form-data">
9      <input type="text" required name="nombre" placeholder="Nombre del archivo en la
      BD"/>
10     <input type="file" required name="imagen"/>
11     <input type="submit" value="Subir archivo"/>
12   </form>
13 </body>
14 </html>
```

**Listado 26.1.1.** index.php

```
1  <?php
2  $host = "localhost";
3  $usuario = "root";
4  $clave = "root";
5  $db = "imagenes";
6  $puertos = "3308";
7  $conn = mysqli_connect($host, $usuario, $clave, $db, $puertos) or die("Error al
  conectarse a la base de datos");
8  ?>
```

**Listado 26.1.2.** conn.php

```
1  <?php
2  include "conn.php";
3  $nombre = $_POST["nombre"];
4  $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
5
6  $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
7
8  if (mysqli_query($conn, $sql)) {
9    print "Se insertó la imagen correctamente";
10 } else {
11   print "Error al insertar el registro";
12 }
13 ?>
```

**Listado 26.1.3.** subirArchivo.php

## 26.2. Desplegar una imagen almacenada en MySQL desde PHP

Una vez almacenada la imagen en la base de datos, en formato binario, debemos de poder desplegarla, en este caso con una etiqueta utilizando las funciones de PHP para manejar cadenas en formato Base64.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6    <?php
7      include "conn.php";
8      $sql = "SELECT * FROM imagenes";
9      $r = mysqli_query($conn, $sql);
10     ?>
11 </head>
12 <body>
13   <form action="subirArchivo.php" method="post" enctype="multipart/form-data">
14     <input type="text" required name="nombre" placeholder="Nombre del archivo en la
BD"/>
15     <input type="file" required name="imagen"/>
16     <input type="submit" value="Subir archivo"/>
17   </form>
18   <table border='1'>
19     <thead>
20       <tr>
21         <th>id</th>
22         <th>Nombre</th>
23         <th>Imagen</th>
24         <th>Borrar</th>
25         <th>Modificar</th>
26       </tr>
27     </thead>
28     <tbody>
29       <?php
30         while ($data = mysqli_fetch_assoc($r)) {
31           print "<tr>";
32           print "<td>".$data['id']. "</td>";
33           print "<td>".$data['nombre']. "</td>";
34           print "<td><img width='200'
src='data:image/jpeg;base64,'" .base64_encode($data['imagen']). "'/></td>";
35           print "<td><a href='index.php?m=b&id=".$data['id']."'>Borrar</td>";
36           print "<td><a href='index.php?m=m&id=".$data['id']."'>Modificar</td>";
37           print "</tr>";
38         }
39       ?>
```

```
40 </tbody>
41 </table>
42 </body>
43 </html>
```

**Listado 26.2.1.** index.php



*www.pacoarce.com*

## 26.3. Crear la estructura de las zonas condicionales en el archivo PHP

En estos ejemplos utilizaremos un solo archivo para realizar las altas, bajas y cambios de la tabla, así como las consultas respectivas, por lo que necesitaremos "zonas condicionales" que aparecerán y desaparecerán dependiendo del comando que manejemos.?

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6    <?php
7      include "conn.php";
8      //modos
9      //A-Alta
10     //B-Baja
11     //C-Cambio
12     //D-Eliminar
13     //S-Select
14     if (isset($_GET["m"])) {
15       $m = $_GET["m"];
16     } else {
17       $m = "S";
18     }
19     //Baja definitiva
20     if ($m=="D") {
21       $id = $_GET["id"];
22       $sql = "DELETE FROM imagenes WHERE id=".$id;
23       $m = "S";
24     }
25     //Select o mostrar
26     if ($m=="S") {
27       $sql = "SELECT * FROM imagenes";
28       $r = mysqli_query($conn, $sql);
29     }
30     //cambio o baja
31     if ($m=="C" || $m=="B") {
32       $id = $_GET["id"];
33       $sql = "SELECT * FROM imagenes WHERE id=".$id;
34       $r = mysqli_query($conn, $sql);
35       $data = mysqli_fetch_assoc($r);
36       //
37     }
38
39
40
41
```

```
42
43  ?>
44 </head>
45 <body>
46  <?php if($m=="A" || $m=="C"){ ?>
47  <form action="subirArchivo.php" method="post" enctype="multipart/form-data">
48  <input type="text" required name="nombre" placeholder="Nombre del archivo en la
BD"/>
49  <input type="file" required name="imagen"/>
50  <input type="submit" value="Subir archivo"/>
51 </form>
52 <?php } ?>
53
54 <?php if($m=="S"){ ?>
55 <table border='1'>
56   <thead>
57     <tr>
58       <th>id</th>
59       <th>Nombre</th>
60       <th>Imagen</th>
61       <th>Borrar</th>
62       <th>Modificar</th>
63     </tr>
64   </thead>
65   <tbody>
66     <?php
67     while ($data = mysqli_fetch_assoc($r)) {
68       print "<tr>";
69       print "<td>".$data['id']. "</td>";
70       print "<td>".$data['nombre']. "</td>";
71       print "<td><img width='200'
src='data:image/jpg;base64,'" .base64_encode($data['imagen']). "'/></td>";
72       print "<td><a href='index.php?m=B&id=".$data['id']."'>Borrar</td>";
73       print "<td><a href='index.php?m=C&id=".$data['id']."'>Modificar</td>";
74       print "</tr>";
75     }
76     ?>
77   </tbody>
78 </table>
79 <?php } ?>
80 </body>
81 </html>
```

## Listado 26.3.1. index.php

```
1 <?php
2 include "conn.php";
3 $nombre = $_POST["nombre"];
4 $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
5
6 $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
7
```



```
8  if (mysqli_query($conn, $sql)) {  
9    print "Se insertó la imagen correctamente";  
10 } else {  
11   print "Error al insertar el registro";  
12 }  
13 ?>
```

**Listado 26.3.2.** subirArchivo.php



*www.pacoarce.com*

## 26.4. Crear una pantalla de advertencia antes de borrar el registro

Un punto muy importante es crear una pantalla de advertencia antes de borrar un registro de una tabla, ya que MySQL no tiene un proceso de "**rollback**" de sus procesos. Esta pantalla también la manejaremos con zonas condicionales.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6    <?php
7      include "conn.php";
8      /*****
9      VARIABLES
10     *****/
11     $msg = array();
12     //modos
13     //A-Alta
14     //B-Baja
15     //C-Cambio
16     //D-Eliminar
17     //S-Select
18     if (isset($_GET["m"])) {
19       $m = $_GET["m"];
20     } else {
21       $m = "S";
22     }
23     //Baja definitiva
24     if ($m=="D") {
25       $id = $_GET["id"];
26       $sql = "DELETE FROM imagenes WHERE id=".$id;
27       $m = "S";
28     }
29     //Select o mostrar
30     if ($m=="S") {
31       $sql = "SELECT * FROM imagenes";
32       $r = mysqli_query($conn, $sql);
33     }
34     //cambio o baja
35     if ($m=="C" || $m=="B") {
36       $id = $_GET["id"];
37       $sql = "SELECT * FROM imagenes WHERE id=".$id;
38       $r = mysqli_query($conn, $sql);
39       //$data = mysqli_fetch_assoc($r);
40       //
41     }
```

```
42     ?>
43 </head>
44 <body>
45     <?php if($m=="S") { ?>
46         <label for="alta"></label>
47         <input type="button" name="alta" value="Subir una imagen" id="alta"/>
48     <?php } ?>
49
50     <?php if($m=="A" || $m=="C" || $m=="B") {
51         if (count($msg)>0) {
52             print "<div>";
53             foreach ($msg as $key => $valor) {
54                 print "<strong>* ".$valor."</strong>";
55             }
56             print "</div>";
57         }
58     } ?>
59
60     <?php if($m=="A" || $m=="C"){ ?>
61         <form action="subirArchivo.php" method="post" enctype="multipart/form-data">
62             <input type="text" required name="nombre" placeholder="Nombre del archivo en la
BD"/>
63             <input type="file" required name="imagen"/>
64             <input type="submit" value="Subir archivo"/>
65         </form>
66     <?php } ?>
67
68     <?php if($m=="S" || $m=="B"){ ?>
69     <table border='1'>
70         <thead>
71             <tr>
72                 <th>id</th>
73                 <th>Nombre</th>
74                 <th>Imagen</th>
75                 <?php if($m=="S"){ ?>
76                     <th>Borrar</th>
77                     <th>Modificar</th>
78                 <?php } ?>
79             </tr>
80         </thead>
81         <tbody>
82             <?php
83             while ($data = mysqli_fetch_assoc($r)) {
84                 print "<tr>";
85                 print "<td>".$data['id'].</td>";
86                 print "<td>".$data['nombre'].</td>";
87                 print "<td><img width='200'
src='data:image/jpeg;base64, ".base64_encode($data['imagen'])."'></td>";
88                 if($m=="S"){
89                     print "<td><a href='index.php?m=B&id=".$data['id']."'>Borrar</td>";
90                     print "<td><a href='index.php?m=C&id=".$data['id']."'>Modificar</td>";
91                 }
```

```
92     print "</tr>";
93     }
94     ?>
95     </tbody>
96     </table>
97     <?php
98     if($m=="B"){
99         print "<label for='si'>¿Desea borrar esta imagen?</label>";
100        print "<input type='button' id='si' value='Si' />";
101        print "<input type='button' id='no' value='No' />";
102        print "<p>Una vez borrado el registro NO se podrá recuperar</p>";
103    }
104    } ?>
105 </body>
106 </html>
```

## Listado 26.4.1. index.php

---

```
1  <?php
2  include "conn.php";
3  $nombre = $_POST["nombre"];
4  $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
5
6  $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
7
8  if (mysqli_query($conn, $sql)) {
9      print "Se insertó la imagen correctamente";
10 } else {
11     print "Error al insertar el registro";
12 }
13 ?>
```

## Listado 26.4.2. subirArchivo.php

*www.pacoarce.com*

## 26.5. Borrar definitivamente el registro de la base de datos

Una vez que el usuario ratifica su selección de borrar el registro, procedemos a hacerlo con la sentencia DELETE.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6    <?php
7    include "conn.php";
8    /*****
9    VARIABLES
10   *****/
11   $msg = array();
12   //modos
13   //A-Alta
14   //B-Baja
15   //C-Cambio
16   //D-Eliminar
17   //S-Select
18   if (isset($_GET["m"])) {
19     $m = $_GET["m"];
20   } else {
21     $m = "S";
22   }
23   //Baja definitiva
24   if ($m=="D") {
25     $id = $_GET["id"];
26     $sql = "DELETE FROM imagenes WHERE id=".$id;
27     if (mysqli_query($conn, $sql)) {
28       array_push($msg,"Registro borrado correctamente");
29     } else {
30       array_push($msg,"Error al borrar el registro");
31     }
32     $m = "S";
33   }
34   //Select o mostrar
35   if ($m=="S") {
36     $sql = "SELECT * FROM imagenes";
37     $r = mysqli_query($conn, $sql);
38   }
39   //cambio o baja
40   if ($m=="C" || $m=="B") {
41     $id = $_GET["id"];
42     $sql = "SELECT * FROM imagenes WHERE id=".$id;
43     $r = mysqli_query($conn, $sql);
44   }
```

```
45 }
46 ?>
47 <script>
48     window.onload = function(){
49         <?php if($m=="S"){ ?>
50             document.getElementById("alta").onclick = function(){
51                 window.open("index.php?m=A", "_self");
52             }
53         <?php } ?>
54
55         <?php if($m=="B") { ?>
56             document.getElementById("si").onclick = function(){
57                 var id = <?php print $id; ?>;
58                 window.open("index.php?m=D&id="+id, "_self");
59             }
60             document.getElementById("no").onclick = function(){
61                 var id = <?php print $id; ?>;
62                 window.open("index.php", "_self");
63             }
64
65         <?php } ?>
66     }
67 </script>
68 </head>
69 <body>
70     <?php if($m=="S") { ?>
71         <label for="alta"></label>
72         <input type="button" name="alta" value="Subir una imagen" id="alta"/>
73     <?php } ?>
74
75     <?php if($m=="A" || $m=="C" || $m=="B") {
76         if (count($msg)>0) {
77             print "<div>";
78             foreach ($msg as $key => $valor) {
79                 print "<strong>* ".$valor."</strong>";
80             }
81             print "</div>";
82         }
83     } ?>
84
85     <?php if($m=="A" || $m=="C"){ ?>
86         <form action="subirArchivo.php" method="post" enctype="multipart/form-data">
87             <input type="text" required name="nombre" placeholder="Nombre del archivo en la
88 BD"/>
89             <input type="file" required name="imagen"/>
90             <input type="submit" value="Subir archivo"/>
91         </form>
92     <?php } ?>
93
94     <?php if($m=="S" || $m=="B"){ ?>
95         <table border='1'>
```

```
96     <thead>
97     <tr>
98         <th>id</th>
99         <th>Nombre</th>
100        <th>Imagen</th>
101        <?php if($m=="S"){ ?>
102            <th>Borrar</th>
103            <th>Modificar</th>
104        <?php } ?>
105    </tr>
106 </thead>
107 <tbody>
108 <?php
109 while ($data = mysqli_fetch_assoc($r)) {
110     print "<tr>";
111     print "<td>".$data['id']. "</td>";
112     print "<td>".$data['nombre']. "</td>";
113     print "<td><img width='200'
src='data:image/jpeg;base64, ".base64_encode($data['imagen']). "' /></td>";
114     if($m=="S"){
115         print "<td><a href='index.php?m=B&id=".$data['id']."'>Borrar</td>";
116         print "<td><a href='index.php?m=C&id=".$data['id']."'>Modificar</td>";
117     }
118     print "</tr>";
119 }
120 ?>
121 </tbody>
122 </table>
123 <?php
124 if($m=="B"){
125     print "<label for='si'>¿Desea borrar esta imagen?</label>";
126     print "<input type='button' id='si' value='Si' />";
127     print "<input type='button' id='no' value='No' />";
128     print "<p>Una vez borrado el registro NO se podrá recuperar</p>";
129 }
130 } ?>
131 </body>
132 </html>
```

## Listado 26.5.1. index.php

```
1 <?php
2 include "conn.php";
3 $nombre = $_POST["nombre"];
4 $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
5
6 $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
7
8 if (mysqli_query($conn, $sql)) {
9     print "Se insertó la imagen correctamente";
10 } else {
11     print "Error al insertar el registro";
```

```
12 }  
13 ?>
```

**Listado 26.5.2.** subirArchivo.php



*www.pacoarce.com*



## 26.6. Integrar el proceso de Alta en el archivo PHP

En la primera parte hicimos el proceso de alta a la base de datos en un archivo aparte. Ahora lo integraremos para luego modificarlo para el proceso de "Cambio" o modificación de la información en la tabla de imágenes.?

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6    <?php
7      include "conn.php";
8      /*****
9      VARIABLES
10     *****/
11     $msg = array();
12     //modos
13     //A-Alta
14     //B-Baja
15     //C-Cambio
16     //D-Eliminar
17     //S-Select
18     if (isset($_POST['nombre'])) {
19       $nombre = $_POST["nombre"];
20       $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
21       //
22       $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
23       //
24       if (mysqli_query($conn, $sql)) {
25         array_push($msg,"Se insertó la imagen correctamente");
26       } else {
27         array_push($msg,"Error al insertar el registro");
28       }
29     }
30     if (isset($_GET["m"])) {
31       $m = $_GET["m"];
32     } else {
33       $m = "S";
34     }
35     //Baja definitiva
36     if ($m=="D") {
37       $id = $_GET["id"];
38       $sql = "DELETE FROM imagenes WHERE id=".$id;
39       if (mysqli_query($conn, $sql)) {
40         array_push($msg,"Registro borrado correctamente");
41       } else {
42         array_push($msg,"Error al borrar el registro");
43       }
44     }
45   }
46   </?php>
47 </head>
48 </html>
```

```
44     $m = "S";
45 }
46 //Select o mostrar
47 if ($m=="S") {
48     $sql = "SELECT * FROM imagenes";
49     $r = mysqli_query($conn, $sql);
50 }
51 //cambio o baja
52 if ($m=="C" || $m=="B") {
53     $id = $_GET["id"];
54     $sql = "SELECT * FROM imagenes WHERE id=".$id;
55     $r = mysqli_query($conn, $sql);
56     //
57 }
58 ?>
59 <script>
60     window.onload = function(){
61         <?php if($m=="S"){ ?>
62             document.getElementById("alta").onclick = function(){
63                 window.open("index.php?m=A", "_self");
64             }
65         <?php } ?>
66
67         <?php if($m=="B") { ?>
68             document.getElementById("si").onclick = function(){
69                 var id = <?php print $id; ?>;
70                 window.open("index.php?m=D&id="+id, "_self");
71             }
72             document.getElementById("no").onclick = function(){
73                 var id = <?php print $id; ?>;
74                 window.open("index.php", "_self");
75             }
76
77         <?php } ?>
78     }
79 }
80 </script>
81 </head>
82 <body>
83     <?php if($m=="S") { ?>
84         <label for="alta"></label>
85         <input type="button" name="alta" value="Subir una imagen" id="alta"/>
86     <?php } ?>
87
88     <?php if($m=="A" || $m=="C" || $m=="B") {
89         if (count($msg)>0) {
90             print "<div>";
91             foreach ($msg as $key => $valor) {
92                 print "<strong>* ".$valor."</strong>";
93             }
94             print "</div>";
95         }
96     }
```

```
96     } ?>
97
98     <?php if($m=="A" || $m=="C"){
99         if($m=="C") $data=mysqli_fetch_assoc($r);
100     ?>
101     <form action="index.php" method="post" enctype="multipart/form-data">
102         <input type="text" required name="nombre" placeholder="Nombre del archivo en la
103 BD" value="<?php if($m=='C') print $data['nombre']; ?>" />
104         <input type="file" <?php if($m!='C') print 'required'; ?> name="imagen"/>
105         <input type="submit" value="Subir archivo"/>
106     </form>
107     <?php } ?>
108
109 <?php if($m=="S" || $m=="B"){ ?>
110 <table border='1'>
111     <thead>
112     <tr>
113         <th>id</th>
114         <th>Nombre</th>
115         <th>Imagen</th>
116         <?php if($m=="S"){ ?>
117             <th>Borrar</th>
118             <th>Modificar</th>
119         <?php } ?>
120     </tr>
121 </thead>
122 <tbody>
123 <?php
124 while ($data = mysqli_fetch_assoc($r)) {
125     print "<tr>";
126     print "<td>".$data['id']. "</td>";
127     print "<td>".$data['nombre']. "</td>";
128     print "<td><img width='200'
129 src='data:image/jpeg;base64, ".base64_encode($data['imagen']). "' /></td>";
130     if($m=="S"){
131         print "<td><a href='index.php?m=B&id=".$data['id']."'>Borrar</td>";
132         print "<td><a href='index.php?m=C&id=".$data['id']."'>Modificar</td>";
133     }
134     print "</tr>";
135 }
136 ?>
137 </tbody>
138 </table>
139 <?php
140 if($m=="B"){
141     print "<label for='si'>¿Desea borrar esta imagen?</label>";
142     print "<input type='button' id='si' value='Si' />";
143     print "<input type='button' id='no' value='No' />";
144     print "<p>Una vez borrado el registro NO se podrá recuperar</p>";
145 }
146 } ?>
147 </body>
```

146 </html>

## Listado 26.6.1. index.php

---

```
1  <?php
2  include "conn.php";
3  $nombre = $_POST["nombre"];
4  $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
5
6  $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
7
8  if (mysqli_query($conn, $sql)) {
9      print "Se insertó la imagen correctamente";
10 } else {
11     print "Error al insertar el registro";
12 }
13 ?>
```

## Listado 26.6.2. subirArchivo.php

*www.pacoarce.com*

## 26.7. Cambiar la imagen y su información en la base de datos

Uno de los procesos más elaborados es el proceso de modificación de la información de la tabla. En esta sección realizaremos la primera parte del proceso.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6    <?php
7    include "conn.php";
8    /*****
9    VARIABLES
10   *****/
11   $msg = array();
12   //modos
13   //A-Alta
14   //B-Baja
15   //C-Cambio
16   //D-Eliminar
17   //S-Select
18   if (isset($_POST['nombre'])) {
19     $nombre = $_POST["nombre"];
20     $id = (isset($_POST["id"]))?$_POST["id"] : "";
21     if ($id=="") {
22       # Alta
23       $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
24       //
25       $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
26       //
27       if (mysqli_query($conn, $sql)) {
28         array_push($msg,"Se insertó la imagen correctamente");
29       } else {
30         array_push($msg,"Error al insertar el registro");
31       }
32     } else {
33       # Cambio
34       if (isset($_FILES["imagen"]["tmp_name"]) && $_FILES["imagen"]["tmp_name"]!="") {
35         $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
36       } else {
37         $imagen = "";
38       }
39       if ($nombre=="") {
40         array_push($msg,"El nombre no puede estar vacío");
41       } else {
42         $sql = "UPDATE imagenes SET ";
43         $sql.= "nombre='". $nombre. "'";
44         if($imagen!=""){
```

```
45     $sql .= ", imagen='". $imagen. "' ";
46 }
47 $sql.= "WHERE id=". $id;
48 //
49 if (mysqli_query($conn,$sql)) {
50     array_push($msg,"Se modificó el registro correctamente");
51 } else {
52     array_push($msg,"Error al modificar el registro");
53 }
54 }
55 }
56 }
57 if (isset($_GET["m"])) {
58     $m = $_GET["m"];
59 } else {
60     $m = "S";
61 }
62 //Baja definitiva
63 if ($m=="D") {
64     $id = $_GET["id"];
65     $sql = "DELETE FROM imagenes WHERE id=". $id;
66     if (mysqli_query($conn, $sql)) {
67         array_push($msg,"Registro borrado correctamente");
68     } else {
69         array_push($msg,"Error al borrar el registro");
70     }
71     $m = "S";
72 }
73 //Select o mostrar
74 if ($m=="S") {
75     $sql = "SELECT * FROM imagenes";
76     $r = mysqli_query($conn, $sql);
77 }
78 //cambio o baja
79 if ($m=="C" || $m=="B") {
80     $id = $_GET["id"];
81     $sql = "SELECT * FROM imagenes WHERE id=". $id;
82     $r = mysqli_query($conn, $sql);
83     //
84 }
85 ?>
86 <script>
87     window.onload = function(){
88         <?php if($m=="S"){ ?>
89             document.getElementById("alta").onclick = function(){
90                 window.open("index.php?m=A","_self");
91             }
92         <?php } ?>
93
94         <?php if($m=="B") { ?>
95             document.getElementById("si").onclick = function(){
96                 var id = <?php print $id; ?>;
```

```
97     window.open("index.php?m=D&id="+id,"_self");
98 }
99 document.getElementById("no").onclick = function(){
100     var id = <?php print $id; ?>;
101     window.open("index.php","_self");
102 }
103
104 <?php } ?>
105
106 }
107 </script>
108 </head>
109 <body>
110 <?php if($m=="S") { ?>
111     <label for="alta"></label>
112     <input type="button" name="alta" value="Subir una imagen" id="alta"/>
113 <?php } ?>
114
115 <?php if($m=="A" || $m=="C" || $m=="B") {
116     if (count($msg)>0) {
117         print "<div>";
118         foreach ($msg as $key => $valor) {
119             print "<strong>* ".$valor."</strong>";
120         }
121         print "</div>";
122     }
123 } ?>
124
125 <?php if($m=="A" || $m=="C"){
126     if($m=="C") $data=mysqli_fetch_assoc($r);
127 } ?>
128 <form action="index.php" method="post" enctype="multipart/form-data">
129     <input type="text" required name="nombre" placeholder="Nombre del archivo en la
BD" value="<?php if($m=='C') print $data['nombre']; ?>" />
130     <input type="file" <?php if($m!='C') print 'required'; ?> name="imagen"/>
131     <input type="hidden" name="id" id="id" value="<?php print $id; ?>">
132     <input type="submit" value="Subir archivo"/>
133 </form>
134 <?php } ?>
135
136 <?php if($m=="S" || $m=="B"){ ?>
137 <table border='1'>
138     <thead>
139     <tr>
140         <th>id</th>
141         <th>Nombre</th>
142         <th>Imagen</th>
143         <?php if($m=="S"){ ?>
144             <th>Borrar</th>
145             <th>Modificar</th>
146         <?php } ?>
147     </tr>
```

```
148     </thead>
149     <tbody>
150         <?php
151         while ($data = mysqli_fetch_assoc($r)) {
152             print "<tr>";
153             print "<td>".$data['id']. "</td>";
154             print "<td>".$data['nombre']. "</td>";
155             print "<td><img width='200'
src='data:image/jpeg;base64, ".base64_encode($data['imagen']). "' /></td>";
156             if($m=="S"){
157                 print "<td><a href='index.php?m=B&id=".$data['id']."'>Borrar</td>";
158                 print "<td><a href='index.php?m=C&id=".$data['id']."'>Modificar</td>";
159             }
160             print "</tr>";
161         }
162     ?>
163 </tbody>
164 </table>
165 <?php
166 if($m=="B"){
167     print "<label for='si'>¿Desea borrar esta imagen?</label>";
168     print "<input type='button' id='si' value='Si' />";
169     print "<input type='button' id='no' value='No' />";
170     print "<p>Una vez borrado el registro NO se podrá recuperar</p>";
171 }
172 } ?>
173 </body>
174 </html>
```

**Listado 26.7.1.** index.php

*www.pacoarce.com*



## 26.8. Modificar el aspecto de la aplicación de alta, bajas y cambios con CSS

Como un punto final modificaremos el aspecto de la aplicación de altas, bajas y cambios (o CRUD en inglés) por medio de los estilos en cascada.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Subir una imagen a la base de datos</title>
5    <meta charset="utf-8">
6    <style>
7      body{
8        background-color: #bbd4d8;
9        width: 440px;
10       margin: 0 auto;
11      }
12      th{
13        background-color: #728987;
14      }
15      tr:nth-child(even) {background-color: #486273;}
16      tr:nth-child(odd) {background-color: #ccc;}
17      table{ width:100%; }
18    </style>
19    <?php
20    include "conn.php";
21    /*****
22    VARIABLES
23    *****/
24    $msg = array();
25    //modos
26    //A-Alta
27    //B-Baja
28    //C-Cambio
29    //D-Eliminar
30    //S-Select
31    if (isset($_POST['nombre'])) {
32      $nombre = $_POST["nombre"];
33      $id = (isset($_POST["id"]))?$_POST["id"] : "";
34      if ($id=="") {
35        # Alta
36        $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
37        //
38        $sql = "INSERT INTO imagenes(nombre,imagen) VALUES('$nombre','$imagen')";
39        //
40        if (mysqli_query($conn, $sql)) {
41          array_push($msg,"Se insertó la imagen correctamente");
42        } else {
43          array_push($msg,"Error al insertar el registro");
```

```
44     }
45 } else {
46     # Cambio
47     if (isset($_FILES["imagen"]["tmp_name"]) && $_FILES["imagen"]["tmp_name"]!="") {
48         $imagen = addslashes(file_get_contents($_FILES["imagen"]["tmp_name"]));
49     } else {
50         $imagen = "";
51     }
52     if ($nombre=="") {
53         array_push($msg,"El nombre no puede estar vacío");
54     } else {
55         $sql = "UPDATE imagenes SET ";
56         $sql.= "nombre='". $nombre. "'";
57         if($imagen!=""){
58             $sql .= ", imagen='". $imagen. "' ";
59         }
60         $sql.= "WHERE id='". $id;
61         //
62         if (mysqli_query($conn,$sql)) {
63             array_push($msg,"Se modificó el registro correctamente");
64         } else {
65             array_push($msg,"Error al modificar el registro");
66         }
67     }
68 }
69 }
70 if (isset($_GET["m"])) {
71     $m = $_GET["m"];
72 } else {
73     $m = "S";
74 }
75 //Baja definitiva
76 if ($m=="D") {
77     $id = $_GET["id"];
78     $sql = "DELETE FROM imagenes WHERE id='". $id;
79     if (mysqli_query($conn, $sql)) {
80         array_push($msg,"Registro borrado correctamente");
81     } else {
82         array_push($msg,"Error al borrar el registro");
83     }
84     $m = "S";
85 }
86 //Select o mostrar
87 if ($m=="S") {
88     $sql = "SELECT * FROM imagenes";
89     $r = mysqli_query($conn, $sql);
90 }
91 //cambio o baja
92 if ($m=="C" || $m=="B") {
93     $id = $_GET["id"];
94     $sql = "SELECT * FROM imagenes WHERE id='". $id;
95     $r = mysqli_query($conn, $sql);
```

```
96  //
97  }
98  ?>
99  <script>
100  window.onload = function(){
101      <?php if($m=="S"){ ?>
102          document.getElementById("alta").onclick = function(){
103              window.open("index.php?m=A", "_self");
104          }
105      <?php } ?>
106
107      <?php if($m=="B") { ?>
108          document.getElementById("si").onclick = function(){
109              var id = <?php print $id; ?>;
110              window.open("index.php?m=D&id="+id, "_self");
111          }
112          document.getElementById("no").onclick = function(){
113              var id = <?php print $id; ?>;
114              window.open("index.php", "_self");
115          }
116
117      <?php } ?>
118  }
119  </script>
120 </head>
121 <body>
122     <?php if($m=="S") { ?>
123         <center>
124             <label for="alta"></label>
125             <input type="button" name="alta" value="Subir una imagen" id="alta"/>
126         </center>
127     <?php } ?>
128
129     <?php if($m=="A" || $m=="C" || $m=="B") {
130         if (count($msg)>0) {
131             print "<div>";
132             foreach ($msg as $key => $valor) {
133                 print "<strong>* ".$valor."</strong>";
134             }
135             print "</div>";
136         }
137     } ?>
138
139     <?php if($m=="A" || $m=="C"){
140         if($m=="C") $data=mysqli_fetch_assoc($r);
141     } ?>
142
143     <form action="index.php" method="post" enctype="multipart/form-data">
144         <input type="text" required name="nombre" placeholder="Nombre del archivo en la
145         BD" value="<?php if($m=='C') print $data['nombre']; ?>" />
146         <input type="file" <?php if($m!='C') print 'required'; ?> name="imagen"/>
```

```
146     <input type="hidden" name="id" id="id" value="<?php print (isset($id))? $id: '' ;
147     <input type="submit" value="Subir archivo"/>
148 </form>
149 <?php } ?>
150
151 <?php if($m=="S" || $m=="B"){ ?>
152 <table border='1'>
153     <thead>
154         <tr>
155             <th>id</th>
156             <th>Nombre</th>
157             <th>Imagen</th>
158             <?php if($m=="S"){ ?>
159                 <th>Borrar</th>
160                 <th>Modificar</th>
161             <?php } ?>
162         </tr>
163     </thead>
164     <tbody>
165         <?php
166         while ($data = mysqli_fetch_assoc($r)) {
167             print "<tr>";
168             print "<td>". $data['id']. "</td>";
169             print "<td>". $data['nombre']. "</td>";
170             print "<td><img width='200'
src='data:image/jpeg;base64, ".base64_encode($data['imagen']). "' /></td>";
171             if($m=="S"){
172                 print "<td><a href='index.php?m=B&id=". $data['id']. "'>Borrar</td>";
173                 print "<td><a href='index.php?m=C&id=". $data['id']. "'>Modificar</td>";
174             }
175             print "</tr>";
176         }
177     ?>
178 </tbody>
179 </table>
180 <?php
181     if($m=="B"){
182         print "<label for='si'>¿Desea borrar esta imagen?</label>";
183         print "<input type='button' id='si' value='Si' />";
184         print "<input type='button' id='no' value='No' />";
185         print "<p>Una vez borrado el registro NO se podrá recuperar</p>";
186     }
187 } ?>
188 </body>
189 </html>
```

**Listado 26.8.1.** index.php

## 26.9. Bajar a un archivo en formato Base64 con PHP

Un paso muy necesario es bajar la información binaria almacenada en formato Base64 a un archivo de imágenes físico, pero esta vez en el lado del servidor.

```
1  <?php
2  include "conn.php";
3
4  $sql = "SELECT * FROM imagenes";
5
6  if ($r = mysqli_query($conn, $sql)) {
7      while($d = mysqli_fetch_assoc($r)){
8          $imagen = $d["imagen"];
9          $nombre = utf8_encode($d["nombre"]);
10         $archivo = $nombre.".jpg";
11         file_put_contents($archivo, $imagen);
12         print "<img src='". $archivo.'" />";
13         print "<p>". $nombre."</p>";
14     }
15
16 }
17
18
19
20 ?>
```

**Listado 26.9.1.** bajalimagen.php

*www.pacoarce.com*

# Todo PHP7

## De novato a experto

### Capítulo 27: Crear archivos PDF con la librería HTML2PDF

**Objetivo:** El alumno podrá crear archivos PDF desde la estructura HTML con la librería gratuita HTML2PDF.

#### Introducción:

Por medio de la librería gratuita HTML2PDF podemos convertir de una forma más directa páginas creadas en PHP a PDF. Los temas que trataremos en esta sección son:

- Instalar la librería HTML2PDF y crear una salida
- Algunos parámetros de la librería HTML2PDF
- Leer un archivo HTML externo
- Crear un botón y pasar parámetros
- Las etiquetas page y bookmark para crear un índice automático

## 27.1. Instalar la librería HTML2PDF y crear una salida

Similar a **npm** para Node.js o **bundler** para Ruby.

- Composer es un manejador de dependencias. La instalación siempre es local para cualquier proyecto.
- Para instalar la librería necesitamos Composer.
- Podemos utilizar los estilos en cascada o CSS para maquetar y añadir colores.
- La librería HTML2PDF nos permitirá maquetar con HTML, la cual la hace muy sencilla de manejar.

---

```
1  <?php
2  require __DIR__.' /vendor/autoload.php';
3
4  use Spipu\Html2Pdf\Html2Pdf;
5
6  $html2pdf = new Html2Pdf();
7  $html2pdf->writeHTML('<h1>Hola mundo</h1>todo bajo control...');
8  $html2pdf->output();
9
10 ?>
```

**Listado 27.1.1.** holamundo.php

---

```
1  {
2  "require": {
3    "spipu/html2pdf": "^5.1"
4  }
5  }
6
```

**Listado 27.1.2.** composer.json

*www.pacoarce.com*

## 27.2. Algunos parámetros de la librería HTML2PDF

Algunos parámetros de configuración son:

\$orientation P P (portrait) o L (landscape)

\$format A4 Por omisión es A4

\$lang fr

\$unicode true

\$encoding UTF-8

\$margins array(5, 5, 5, 8) (left, top, right, bottom) en mm

\$pdfafalse

---

```
1  <?php
2  require __DIR__.'./vendor/autoload.php';
3
4  use Spipu\Html2Pdf\Html2Pdf;
5
6  $html2pdf = new Html2Pdf('P', 'LETTER', 'es', 'UTF-8');
7  $html2pdf->writeHTML('<h1>Hola mundo</h1>Algunas cosas en Español');
8  $html2pdf->output("salida.pdf");
9
10 ?>
```

**Listado 27.2.1.** holamundo.php

*www.pacoarce.com*



## 27.3. Leer un archivo HTML externo

Para poder leer un archivo externo con las etiquetas HTML necesitamos:

- Recuperar la salida con la función `ob_get_clean()`
- Utilizar la función `ob_start()`
- Llamar al archivo con una instrucción `"require_once"`.

---

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <title>Hola mundo</title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8    <h1>Hola mundo</h1>
9    <h2>Bienvenido a la librería HTML2PDF</h2>
10 </body>
11 </html>
```

**Listado 27.3.1.** salida.php

*www.pacoarce.com*

## 27.4. Crear un botón y pasar parámetros

Podemos añadir un botón y pasar parámetros por medio de un formulario en HTML, leyendo los datos desde el arreglo POST y validándolo por medio de isset().

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Hola mundo</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8 <h1>Hola mundo</h1>
9 <h2>Bienvenido a la librería HTML2PDF</h2>
10 <h2>Autor: <?php print $autor; ?></h2>
11 </body>
12 </html>
```

**Listado 27.4.1.** salida.php

*www.pacoarce.com*

## 27.5. Las etiquetas page y bookmark para crear un índice automático

Para poder manejar el formato de una página tenemos las siguientes etiquetas:

Con estas etiquetas no debe de utilizar las etiquetas body ni html

page\_footer

page\_header

page

```
1  <style>
2    table.page_header {width: 100%; border: none; background-color: #DDDDFF;
3    border-bottom: solid 1mm #AAAADD; padding: 2mm }
4    table.page_footer {width: 100%; border: none; background-color: #DDDDFF;
5    border-top: solid 1mm #AAAADD; padding: 2mm}
6    h1 {color: #000033}
7    h2 {color: #000055}
8    h3 {color: #000077}
9    div.nivel{
10     padding-left: 5mm;
11   }
12 </style>
13 <page backtop="14mm" backbottom="14mm" backleft="10mm" backright="10mm"
14 style="font-size: 12pt">
15 <page_header>
16   <table class="page_header">
17     <tr>
18       <td style="width: 100%; text-align: left">
19         Titulo del libro
20       </td>
21     </tr>
22   </table>
23 </page_header>
24 <page_footer>
25   <table class="page_footer">
26     <tr>
27       <td style="width: 100%; text-align: right">
28         página 443/461
29       </td>
30     </tr>
31   </table>
32 </page_footer>
33 <bookmark title="Titulo" level="0" ></bookmark>
34 </page>
35 <page pageset="old">
36   <bookmark title="Capítulo 1" level="0" ></bookmark><h1>Capítulo 1</h1>
37   <div class="nivel">
38     Contenido capítulo 1
39   </div>
```

```
37 </page>
38 <page pageset="old">
39 <bookmark title="Capítulo 2" level="0" ></bookmark><h1>Capítulo 2</h1>
40 <div class="nivel">
41 Introducción al Capítulo 2
42 <bookmark title="Introducción a PHP" level="1" ></bookmark><h2>Capítulo 2.1</h2>
43 <div class="nivel">
44 Contenido Capítulo 2.1
45 </div>
46 <bookmark title="Capítulo 2.2" level="1" ></bookmark><h2>Capítulo 2.2</h2>
47 <div class="nivel">
48 Contenido Capítulo 2.2
49 </div>
50 <bookmark title="Capítulo 2.3" level="1" ></bookmark><h2>Capítulo 2.3</h2>
51 <div class="nivel">
52 Contenido Capítulo 2.3
53 </div>
54 </div>
55 </page>
56 <page pageset="old">
57 <bookmark title="Capítulo 3" level="0" ></bookmark><h1>Capítulo 3</h1>
58 <div class="nivel">
59 Introducción Capítulo 3
60 <bookmark title="Capítulo 3.1" level="1" ></bookmark><h2>Capítulo 3.1</h2>
61 <div class="nivel">
62 Contenido Capítulo 3.1
63 </div>
64 <bookmark title="Capítulo 3.2" level="1" ></bookmark><h2>Capítulo 3.2</h2>
65 <div class="nivel">
66 Introducción Capítulo 3.2
67 <bookmark title="Capítulo 3.2.1" level="2" ></bookmark><h3>Capítulo 3.2.1</h3>
68 <div class="nivel">
69 Contenido Capítulo 3.2.1
70 </div>
71 <bookmark title="Capítulo 3.2.2" level="2" ></bookmark><h3>Capítulo 3.2.2</h3>
72 <div class="nivel">
73 Contenido Capítulo 3.2.2
74 </div>
75 </div>
76 </div>
77 </page>
```

**Listado 27.5.1.** libroContenido.php

# Todo PHP7

## De novato a experto

### Capítulo 28: Manejo de errores

**Objetivo:** El alumno aprenderá a manejar los errores generados por PHP, crear mensajes de advertencia, noticias o errores y lanzar y atrapar excepciones.

#### Introducción:

El manejo de los errores, tanto generados por PHP o por el programador, nos será de enorme ayuda para darle a nuestros usuarios una mejor experiencia en el uso del sistema o aplicación. En esta sección estudiaremos los siguientes puntos:

- Manejo de errores en PHP7
- Las funciones die() y exit()
- Manejador de errores personalizado
- Mandar un mensaje en tiempo de ejecución: trigger\_error()
- Error Log
- Excepciones
- Try, throw y catch

## 28.1. Manejo de errores en PHP7

- Por omisión **PHP** envía errores a pantalla con archivo, número de línea y mensaje.
- En algunas instalaciones, por ejemplo en **MAMP**, los errores pueden estar deshabilitados.
- Podemos habilitarlos con:

```
ini_set('display_errors', 1);  
error_reporting(E_ALL);
```



*www.pacoarce.com*

## 28.2. Las funciones die() y exit()

Por medio de las funciones **die()** o **exit()** podemos terminar un programa enviando un mensaje de error.

```
1 <?php
2 if(file_exists("datos.txt")){
3 $archivo = fopen("datos.txt","r");
4 } else {
5 die("No existe el archivo 'datos.txt', no podemos continuar.");
6 }
7 print "Este mensaje no se despliega si hay un error.";
8
9 ?>
```

**Listado 28.2.1.** dieExit.php

*www.pacoarce.com*

## 28.3. Manejador de errores personalizado

Podemos hacer un manejador de errores personalizado con la instrucción con al menos dos se los siguientes parámetros:

- **error\_level**: (requerido). Es un número que especifica el número de error según la tabla que se verá más adelante.
- **error\_message**: (requerido). Mensaje de error
- **error\_file**: Nombre del archivo donde se produjo el error.
- **error\_line**: Número de línea donde se produjo el error.
- **error\_context**: Opcional. Contiene un arreglo con el valor de las variables en su contexto.

Definir la función personalizada:

```
set_error_handler("customError");
```

### Manejador de errores personalizado

2	E_WARNING	Error no fatal, el programa puede continuar.
8	E_NOTICE	Run-time notices. error no fatal.
256	E_USER_ERROR	Fatal user-generated error. Es similar a E_ERROR.
512	E_USER_WARNING	Non-fatal user-generated warning. Similar a E_WARNING.

Imagen 28.3.1. Constantes de errores en PHP

[www.pacoarce.com](http://www.pacoarce.com)



## Manejador de errores personalizado

1024	E_USER_NOTICE	User-generated notice. Es similar a E_NOTICE.
4096	E_RECOVERABLE_ERROR	Catchable fatal error. Es similar a E_ERROR.
8191	E_ALL	Todos los errores y mensajes.

**Imagen 28.3.2.** Constantes de errores en PHP

```

1  <?php
2  function manejadorErrores($nivel, $mensaje, $archivo, $linea, $contexto)
3  {
4  print "Error nivel : ".$nivel."<br>";
5  print "Mensaje : ".$mensaje."<br>";
6  print "En el archivo: ".$archivo."<br>";
7  print "En la linea : ".$linea."<br>";
8  //print "Contexto: ";
9  //var_dump($contexto);
10 }
11 print $temporal = "temporal";
12 set_error_handler("manejadorErrores");
13
14
15 print $noexiste;
16
17
18 ?>

```

**Listado 28.3.1.** rutinaError.php

## 28.4. Mandar un mensaje en tiempo de ejecución: `trigger_error()`

Por medio de la función *trigger\_error()* podemos enviar un mensaje de error, advertencia o aviso de nivel de usuario

---

```
1 <?php
2 $num = 0;
3 if ($num<1) {
4 trigger_error("No podemos dividir entre cero", E_USER_NOTICE);
5 }
6 ?>
```

**Listado 28.4.1.** triggerError.php

*www.pacoarce.com*

## 28.5. Error Log

Por medio de la función `error_log()` podemos enviar un mensaje de error a otro archivo diferente de o a un correo electrónico.

Sintaxis:

**`error_log`** ( string \$mensaje

[, int \$tipo = 0

[, string \$destino

, string \$encabezados ]])

0 El mensaje es enviado al archivo por omisión. Opción predeterminada.

1 El mensaje es enviado por correo.

2 Ya no es una opción.

3 El mensaje es añadido al final del archivo destino.

4 El mensaje es enviado directamente al gestor de registro de la SAPI.

---

```
1  <?php
2  function manejadorErrores($nivel, $mensaje, $archivo, $linea, $contexto)
3  {
4      print "Error nivel : ".$nivel."<br>";
5      print "Mensaje : ".$mensaje."<br>";
6      print "En el archivo: ".$archivo."<br>";
7      print "En la linea : ".$linea."<br>";
8      //print "Contexto: ";
9      //var_dump($contexto);
10     error_log("Error: [$nivel] $mensaje",1,"mipaginawe@micorreo.com","From:
11     administrador@micirreo.com");
12 }
13 print $temporal = "temporal";
14 set_error_handler("manejadorErrores");
15
16 trigger_error("Este es un mensaje de error", E_USER_WARNING);
17
18
19 ?>
```

**Listado 28.5.1.** errorLog.php

## 28.6. Excepciones

Una excepción se ejecuta cuando ocurre un error. Desde PHP5 contamos con una clase *Exception*.

---

```
1  <?php
2  function suma($n1,$n2){
3  $suma = $n1+$n2;
4  if ($suma>10) {
5  throw new Exception("Las sumas no pueden ser mayores a 10");
6  }
7  print $suma;
8  }
9
10
11 suma(14,5);
12 ?>
```

**Listado 28.6.1.** excepciones.php

[www.pacoarce.com](http://www.pacoarce.com)

## 28.7. Try, throw y catch

Una excepción debe contar con una estructura de tipo try-throw-catch.

**try** - La función que nos podría generar una excepción (o error) debe ir dentro de un bloque "try". Si no se lanza la excepción el código sigue en forma normal. Si el error aparece (excepción) se lanza el bloque **throw**.

**throw** - Es cuando se lanza la excepción. Cada "**throw**" deberá tener su "**catch**" (atrapar).

**catch** - Un bloque "**catch**" está relacionada con una excepción y crear un objeto que contiene la información del objeto.

---

```
1  <?php
2
3  function suma($n1, $n2)
4  {
5      $suma = $n1+$n2;
6      if ($suma>10) {
7          throw new Exception("La suma debe ser menor a 10");
8      }
9      return true;
10 }
11
12 try {
13     suma(10,5);
14     //
15     print "Si ves esta cadena, es que la suma es menor a 10";
16 } catch (Exception $e) {
17     print "Mensaje: ".$e->getMessage();
18 }
19
20
21 ?>
```

**Listado 28.7.1.** tryThrowCatch.php

[www.pacoarce.com](http://www.pacoarce.com)

## Indice

Capítulo 1: Introducción a PHP .....	p. 2
1.1. Breve historia de PHP .....	p. 3
1.2. ¿Cómo funciona PHP? .....	p. 4
1.3. ¿Qué necesitamos para usar PHP? .....	p. 5
1.4. Instalar el Apache + PHP + MySQL: XAMPP .....	p. 6
1.5. Instalar MAMP en una computadora con Windows .....	p. 7
1.6. Instalar MAMP en una Mac .....	p. 9
1.7. Nuevas características de PHP7 .....	p. 11
1.8. Nuestra primera página: Hola mundo .....	p. 12
1.9. Mezclar HTML y PHP .....	p. 13
1.10. Comentarios y otras cosas .....	p. 14
Cuestionario .....	p. 15
Capítulo 2: Tipos de datos y variables .....	p. 17
2.1. Variables .....	p. 18
2.2. Tipos de datos: cadenas .....	p. 19
2.3. Concatenación de cadenas .....	p. 21
2.4. Tipos de datos: números enteros .....	p. 22
2.5. Tipos de datos: punto flotante .....	p. 23
2.6. Tipo de dato: Booleano .....	p. 25
2.7. Variables de variables y constantes .....	p. 26
2.8. PHP7: Secuencias de escape para caracteres Unicode .....	p. 28
2.9. Los tipos de datos compuestos y especiales .....	p. 29
2.10. Forzar el tipo de dato .....	p. 30
2.11. Eliminar una variable con unset() .....	p. 32
Cuestionario .....	p. 33
Capítulo 3: Operadores en PHP .....	p. 36
3.1. Introducción a los operadores y sus conceptos en PHP .....	p. 37
3.2. Operadores aritméticos .....	p. 38
3.3. Los operadores de asignación y los operadores abreviados o atajos .....	p. 39
3.4. Operadores de bit a bit .....	p. 40
3.5. Operadores de comparación .....	p. 41
3.6. Operadores Lógicos .....	p. 42
3.7. Otros operadores en PHP .....	p. 43
3.8. La precedencia de los operadores .....	p. 44
3.9. PHP7: El operador de fusión de null .....	p. 45

3.10. PHP7: Operador nave espacial .....	p. 46
Cuestionario .....	p. 47
Capítulo 4: Las sentencias condicionales .....	p. 51
4.1. Sentencias condicionales .....	p. 52
4.2. Operadores condicionales anidados .....	p. 54
4.3. Crear rangos con operadores condicionales .....	p. 55
4.4. Sintaxis alterna a las sentencias condicionales .....	p. 56
4.5. El operador condicional .....	p. 57
4.6. La estructura condicional switch .....	p. 58
Cuestionario .....	p. 60
Capítulo 5: Las estructuras cíclicas .....	p. 62
5.1. El ciclo While .....	p. 63
5.2. El ciclo do... while .....	p. 65
5.3. El ciclo for .....	p. 67
5.4. Los comandos break y continue .....	p. 69
Cuestionario .....	p. 70
Capítulo 6: Creación y manejo de arreglos .....	p. 72
6.1. Crear arreglos .....	p. 73
6.2. El ciclo foreach .....	p. 75
6.3. Arreglos asociativos .....	p. 76
6.4. Algunas funciones de los arreglos .....	p. 77
6.5. Ordenamiento de arreglos .....	p. 79
6.6. Ordenar los arreglos asociativos .....	p. 81
6.7. Desordenar un arreglo y selección aleatoria de elementos de un arreglo .....	p. 82
6.8. Funciones para ordenar arreglos en forma ascendente y descendente .....	p. 84
6.9. Funciones de navegación de elemento en un arreglo .....	p. 86
6.10. PHP7: Arreglos de constantes .....	p. 87
6.11. La función var_dump() para mostrar el contenido de una variable .....	p. 88
6.12. La función print_r() para visualizar una variable, arreglo u objeto .....	p. 89
6.13. Arreglos multidimensionales .....	p. 90
6.14. Arreglos globales o .....	p. 91
6.15. El comando List() para asignar valores de arreglos .....	p. 92
6.16. Funciones para arreglos asociativos .....	p. 93
Cuestionario .....	p. 95
Capítulo 7: Creación de funciones personalizadas .....	p. 101
7.1. Crear y llamar a una función .....	p. 102
7.2. El alcance de las variables .....	p. 103
7.3. Regresar un valor de una función .....	p. 106

7.4. Argumentos o parámetros por referencia o por valor .....	p. 108
7.5. PHP7: Valores predeterminados de argumentos .....	p. 109
7.6. PHP7: Declaraciones de tipo en las funciones .....	p. 110
7.7. PHP7: Declaraciones de tipo de devolución .....	p. 111
7.8. PHP7: Tipificación estricta .....	p. 112
7.9. PHP7: Listas de argumentos de longitud variable .....	p. 113
7.10. Crear generadores en PHP .....	p. 114
7.11. Utilizar yield para facilitar valores con clave desde un generador .....	p. 115
7.12. PHP7: El uso de la sentencia Return en los generadores .....	p. 116
7.13. PHP7: Delegación de generadores .....	p. 117
7.14. Las funciones anónimas en PHP .....	p. 119
7.15. Tomar valores externos desde una función anónima .....	p. 120
Cuestionario .....	p. 121
Capítulo 8: Validación de datos de los formularios .....	p. 125
8.1. Comunicación de un Fomulario con PHP .....	p. 126
8.2. Recibir variables simples .....	p. 127
8.3. Recibir variables múltiples .....	p. 128
8.4. Validación de cadenas .....	p. 130
8.5. Validación de arreglos con isset() .....	p. 133
8.6. Recibir campos ocultos de formas HTML .....	p. 136
8.7. Redireccionar a otras páginas con el comando header .....	p. 139
8.8. Uso de plantillas PHP .....	p. 141
Cuestionario .....	p. 143
Capítulo 9: Manejo de fechas .....	p. 146
9.1. Uso de fechas con time() y date() .....	p. 147
9.2. La función getdate() .....	p. 149
9.3. La función mktime() .....	p. 151
9.4. Verificar una fecha con checkdate() .....	p. 152
9.5. Cambiar la zona horaria: date_default_timezone_set .....	p. 153
9.6. Convertir una cadena a fecha con strtotime() .....	p. 154
Cuestionario .....	p. 155
Capítulo 10: Formateo de datos .....	p. 156
10.1. Formato de un número con printf() .....	p. 157
10.2. Salida a una cadena con sprintf() .....	p. 160
10.3. Formato de un número con number_format() .....	p. 161
10.4. Modificar la configuración local con setlocale() .....	p. 163
10.5. Serialización de objetos y arreglos .....	p. 164
10.6. Filtros para unserialize() .....	p. 165



Cuestionario .....	p. 167
Capítulo 11: Manejo de cadenas y funciones matemáticas .....	p. 169
11.1. Longitud de una cadena con strlen() .....	p. 170
11.2. Buscar una subcadena en una cadena: el comando strpos() .....	p. 171
11.3. Buscar una subcadena en una cadena: el comando strpos() .....	p. 172
11.4. Extraer una subcadena: el comando substr() .....	p. 174
11.5. Pasar una cadena a mayúsculas minúsculas: strtolower() y strtoupper() .....	p. 175
11.6. Reemplazar partes de una cadena: str_replace() .....	p. 176
11.7. Sanear una cadena con PHP, la función str_replace() .....	p. 177
11.8. Crear una cadena aleatoria con str_shuffle() .....	p. 179
11.9. Convertir caracteres especiales de texto a HTML .....	p. 180
11.10. Ajusta la longitud de una cadena: wordwrap() .....	p. 181
11.11. Dividir una cadena en diferentes cadenas: explode() .....	p. 182
11.12. Limpiar cadenas: trim(), ltrim() y rtrim() .....	p. 183
11.13. Cambiar la primera letra a mayúsculas .....	p. 184
11.50. Verificar si es un número con is_number .....	p. 185
11.51. Redondear valores .....	p. 186
11.52. Funciones trigonométricas .....	p. 188
11.53. La función range() .....	p. 189
11.54. Generar números aleatorios .....	p. 190
11.55. Exponentes y raíz cuadrada .....	p. 192
11.56. Cambio de bases .....	p. 193
11.57. Recuperar la parte entera y la parte decimal de una división .....	p. 194
11.58. Funciones de CSPRNG .....	p. 195
Cuestionario .....	p. 196
Capítulo 12: Manejo de archivos .....	p. 199
12.1. Verificar si existe un archivo: file_exists() .....	p. 200
12.2. Determinar si es un archivo o un directorio .....	p. 201
12.3. Propiedades de un archivo .....	p. 202
12.4. Leer el contenido de un directorio .....	p. 204
12.5. Constantes predefinidas .....	p. 206
12.6. Leer el contenido de una carpeta o directorio con scandir() .....	p. 207
12.7. Conocer la ruta de un archivo con dirname(), basename(), realpath() .....	p. 208
Cuestionario .....	p. 209
Capítulo 13: Archivos de datos en el servidor .....	p. 212
13.1. El comando touch() .....	p. 213
13.2. Borrar un archivo, el comando unlink() .....	p. 214
13.3. Modos de abrir un archivo .....	p. 215

13.4. Leer un archivo de datos .....	p. 218
13.5. Escribir en un archivo de datos .....	p. 220
13.6. Añadir datos a un archivo .....	p. 222
13.7. Proteger los archivos con flock() .....	p. 224
13.8. Copiar y renombrar archivos .....	p. 227
13.9. Leer un archivo CSV con fgetcsv() .....	p. 229
13.10. Crear un archivo con formato CSV .....	p. 230
13.11. Leer todo un archivo en una cadena: file_get_contents .....	p. 231
Cuestionario .....	p. 232
Capítulo 14: Crear, modificar y eliminar las Galletas o cookies .....	p. 235
14.1. Crear una cookie .....	p. 236
14.2. Leer una cookie .....	p. 238
14.3. Borrar una cookie .....	p. 239
14.4. Crear un contador de visitas con cookies .....	p. 240
14.5. Almacenar datos de configuración del usuario con cookies. ....	p. 241
14.6. Almacenar los datos del usuario en un login con cookies .....	p. 242
Cuestionario .....	p. 244
Capítulo 15: Conexión con MySQL .....	p. 246
15.1. Crear la base de datos y una tabla de prueba .....	p. 247
15.2. Conectar una base de datos MySQL con PHP .....	p. 249
15.3. Las sentencias include y require .....	p. 250
15.4. Insertar un registro en la base de datos .....	p. 251
15.5. Borrar un registro de una tabla .....	p. 252
15.6. Modificar un registro de la tabla .....	p. 253
15.7. Seleccionar archivos de una tabla, el comando SELECT .....	p. 254
15.8. Establecer el conjunto de caracteres para la base de datos .....	p. 255
15.9. Recuperar el último "id" insertado .....	p. 256
15.10. Leer un archivo CSV e insertarlo en una tabla MySQL .....	p. 257
15.11. Verificar que los datos no se dupliquen. ....	p. 258
15.12. Bajar la información de MySQL a CSV .....	p. 260
Cuestionario .....	p. 261
Capítulo 16: Manejo de imágenes con PHP .....	p. 264
16.1. Galería de fotos desde una carpeta .....	p. 265
16.2. Tabla de imágenes tomándola de una carpeta .....	p. 267
16.3. Advertencia para una imagen .....	p. 269
16.4. Borrar físicamente la imagen .....	p. 272
16.5. Parámetros para la optimización de una imagen .....	p. 276
16.6. Optimiza la imagen por un porcentaje .....	p. 280

16.7. Optimizar una imagen por su ancho en forma proporcional .....	p. 282
16.8. El uso de filtros a una imagen .....	p. 284
16.9. Subir un archivo desde la computadora al servidor .....	p. 287
16.10. Validar el tipo de archivo subido .....	p. 288
16.11. Subir una imagen de PHP a la base de datos MySQL .....	p. 290
16.12. Guardar la imagen y desplegar una tabla .....	p. 292
16.13. Desplegar una columna binaria de MySQL desde PHP .....	p. 294
16.14. de BLOB a un archivo de imagen .....	p. 295
Capítulo 17: Ejemplos y ejercicios .....	p. 296
17.1. Utilizar un mismo archivo con estados en PHP .....	p. 297
17.2. Mandar un mail desde PHP .....	p. 299
17.3. Mandar un mail en formato HTML desde PHP .....	p. 301
17.4. Calcular las edades con PHP .....	p. 302
17.5. Leer y mostrar una tabla de la base de datos .....	p. 303
17.6. Realizar la paginación de una tabla con PHP .....	p. 304
17.7. Crear los botones HTML para la paginación con PHP .....	p. 306
17.8. Crear botones inicio-fin en la paginación con PHP .....	p. 309
17.9. Crear la imagen de un captcha con PHP .....	p. 312
17.10. Generar el texto aleatorio en el Captcha con PHP .....	p. 313
17.11. Llamar al captcha y verificarlo desde un formulario con PHP .....	p. 315
17.12. Conocer la IP, el browser y el sistema operativo del usuario .....	p. 316
Capítulo 18: Trabajar con XML y JSON desde PHP .....	p. 317
18.1. Conceptos generales para leer un documento XML desde PHP .....	p. 318
18.2. Crear el programa que lee un documento XML con PHP .....	p. 319
18.3. Escribir un documento XML desde PHP .....	p. 321
18.4. Leer un archivo JSON .....	p. 323
18.5. Escribir un archivo JSON .....	p. 324
Capítulo 19: Introducción a Google Chart con PHP y MySQL .....	p. 325
Capítulo 20: Crear un archivo PDF desde PHP .....	p. 326
20.1. Crear un PDF desde PHP con la librería FPDF .....	p. 327
20.2. Crear un encabezado y un pie de página con FPDF .....	p. 330
20.3. Leer textos externos y crear un archivo PDF con FPDF .....	p. 333
20.4. Manejo de columnas en texto con FPDF .....	p. 338
20.5. Crear una tabla dinámica con FPDF y PHP .....	p. 341
Capítulo 21: Manejar sesiones con PHP .....	p. 345
21.1. Crear páginas privadas con session_start() .....	p. 346
21.2. Validar y cerrar una sesión .....	p. 348
21.3. Limitar el tiempo de las sesiones con el comando time() .....	p. 350

21.4. PHP7: El arreglo en la función session_start()	p. 351
Capítulo 22: Las expresiones regulares en PHP	p. 352
22.1. Expresiones regulares y PHP	p. 353
22.2. Validar un correo electrónico con expresiones regulares	p. 354
22.3. PHP7: preg_replace_callback_array()	p. 355
Capítulo 23: Depurar nuestro código con Aserciones	p. 356
23.1. Aserciones	p. 357
23.2. assert_options()	p. 358
23.3. Ejemplos de assert PHP 5 y 7	p. 361
23.4. Ejemplos de assert PHP 7	p. 362
Capítulo 24: Crear un filtro para sanar y evaluar datos de entrada	p. 363
24.1. Introducción a los filtros: filter_list() y filter_id()	p. 364
24.2. La función filter_var para sanear y validar	p. 365
24.3. La función filter_input() para filtrar o validar variables externas	p. 366
24.4. La función filter_var_array() para sanear o filtrar varias variables en arreglo	p. 367
24.5. Uso de banderas en filtros	p. 369
24.6. El uso del filtro FILTER_CALLBACK	p. 370
Capítulo 25: Manejo de PHP, XML, MySQL y AJAX	p. 371
25.1. Introducción a AJAX	p. 372
25.2. La instrucción XMLHttpRequest	p. 373
25.3. Crear una solicitud o	p. 374
25.4. La instrucción XMLHttpRequest	p. 376
25.5. Recuperar la información desde AJAX: responseText y responseXML	p. 377
25.6. El manejo de eventos en AJAX	p. 379
25.7. Llamar a archivos PHP desde AJAX	p. 381
25.8. Ligar el AJAX con Bases de datos (PHP+MySQL)	p. 383
25.9. Leer un archivo XML con AJAX generado desde PHP de una base de datos My	p. 387
25.10. Leer un archivo XML con AJAX generado desde PHP de una base de datos M	p. 391
25.11. Sugerir la captura de un archivo por medio de AJAX y PHP	p. 393
25.12. Crear filtros para la lectura de una tabla en MySQL con PHP y AJAX	p. 395
25.13. Leer un archivo XML desde PHP y desplegarlo en HTML por medio de AJAX	p. 398
25.14. Concluir la lectura de un catálogo XML con PHP y AJAX	p. 400
25.15. Crear un	p. 401
25.16. Crear un	p. 404
25.17. Crear un lector de noticias RSS con AJAX y PHP, primera parte	p. 405
25.18. Crear un lector de noticias RSS con AJAX y PHP, segunda parte	p. 407
25.19. Crear una encuesta con PHP y MySQL	p. 409
Capítulo 26: Crear una aplicación de altas, bajas y cambios (crud) de una tabla con PHP	p. 411

26.1. Subir una imagen a una base de datos MySQL con PHP .....	p. 412
26.2. Desplegar una imagen almacenada en MySQL desde PHP .....	p. 413
26.3. Crear la estructura de las zonas condicionales en el archivo PHP .....	p. 415
26.4. Crear una pantalla de advertencia antes de borrar el registro .....	p. 418
26.5. Borrar definitivamente el registro de la base de datos .....	p. 421
26.6. Integrar el proceso de Alta en el archivo PHP .....	p. 425
26.7. Cambiar la imagen y su información en la base de datos .....	p. 429
26.8. Modificar el aspecto de la aplicación de alta, bajas y cambios con CSS .....	p. 433
26.9. Bajar a un archivo en formato Base64 con PHP .....	p. 437
Capítulo 27: Crear archivos PDF con la librería HTML2PDF .....	p. 438
27.1. Instalar la librería HTML2PDF y crear una salida .....	p. 439
27.2. Algunos parámetros de la librería HTML2PDF .....	p. 440
27.3. Leer un archivo HTML externo .....	p. 441
27.4. Crear un botón y pasar parámetros .....	p. 442
27.5. Las etiquetas page y bookmark para crear un índice automático .....	p. 443
Capítulo 28: Manejo de errores .....	p. 445
28.1. Manejo de errores en PHP7 .....	p. 446
28.2. Las funciones die() y exit() .....	p. 447
28.3. Manejador de errores personalizado .....	p. 448
28.4. Mandar un mensaje en tiempo de ejecución: trigger_error() .....	p. 450
28.5. Error Log .....	p. 451
28.6. Excepciones .....	p. 452
28.7. Try, throw y catch .....	p. 453
Índice .....	p. 454

[www.pacoarce.com](http://www.pacoarce.com)