



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Tutorial – Introducción a PHP

Introducción

- Con HTML y CSS es posible crear páginas web estáticas.
- Con JavaScript se pueden diseñar páginas web dinámicas de cliente.
- Con PHP se pueden diseñar páginas web dinámicas de servidor:
 - El script incluido en la página se ejecutará en el servidor web y la salida producida será enviada al cliente para que se visualice en el browser.
 - La salida puede ser un código HTML o una combinación de HTML y JavaScript

Características del lenguaje de programación PHP


- Es un lenguaje utilizado para programar scripts que se ejecutan en el servidor web, permitiendo así diseñar páginas web dinámicas de servidor.
- Brinda soporte para muchos servidores web, lo que permite que se ejecute bajo Apache, IIS, etc.
- A partir de la versión 5, tiene un gran soporte para el desarrollo de aplicaciones orientadas a objetos.
- Incluye extensiones para conectarse a varios motores de bases de datos.
- Incorpora soporte para varios protocolos de Internet, como ser POP3, FTP, TELNET, etc.
 - Esto, combinado con el soporte de XML, forma la base para el desarrollo de web services.
- Brinda soporte para implementar canales seguros para el intercambio de datos utilizando SSL (Secure Sockets Layer).
- Es libre, por tanto hay libertad para utilizarlo, para modificar su código fuente y para modificar las aplicaciones desarrolladas con él.
- Es de fácil aprendizaje y su sintaxis se basa en C. Incluso, las aplicaciones LAMP (Linux, Apache, MySQL y PHP) y/o WAMP (Windows, Apache, MySQL y PHP) facilitan mucho la instalación y desarrollo de los scripts.

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

El primer script

<pre>1 <!-- Codigo_6.php --> 2 <!DOCTYPE html> 3 <html lang="es"> 4 <head> 5 <title>Primer script PHP</title> 6 </head> 7 <body> 8 <header> 9 <?php 10 /*Este es un comentario*/ 11 echo "<h1>Comenzamos la Unidad N° 4!!!</h1>"; 12 ?> 13 </header> 14 <footer> 15 <p>Archivo: Codigo_6.php</p> 16 </footer> 17 </body> 18 </html></pre>	<p>Los archivos se guardan con extensión ".php"</p> <p>La inclusión del código PHP se demarca con "<?php" para el inicio y ">" para el fin</p> <p>Los comentarios se indican con /* y */</p> <p>Las sentencias finalizan con ";"</p> <p>Es posible incluir etiquetas HTML dentro de las sentencias de impresión</p>
---	---

***Variables***

- El identificador debe comenzar con el símbolo "\$", pueden contener caracteres alfanuméricos y no hay restricción en cuanto a la longitud del mismo.
- Son case sensitive.
- No se declaran y son dinámicas
 - Se determina su tipo cuando son asignadas por primera vez y este puede cambiar durante la ejecución del script, de acuerdo al contexto en donde se utilice.

Tipos de datos

- PHP incluye los siguientes tipos de datos:
 - Integer
 - Float (o double o real)
 - Boolean: el valor puede ser TRUE o FALSE



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

- String:
 - No hay restricción en cuanto a la longitud de las cadenas
 - Pueden indicarse con comillas simples o dobles. Las simples no se interpretan.
- Array: el índice puede ser numérico o una cadena (arreglos enumerados o asociativos)
- Object: se incluyó a partir de PHP 3 y luego fueron mejorados en PHP 5

Operadores

- Operador de asignación:
 - PHP usa el símbolo “=” para asignar valores a las variables.
 - Se puede combinar con los operadores aritméticos o con el de concatenación, a fin de simplificar la sintaxis de estas operaciones:

Sentencia	Sintaxis reducida
<variable> = <variable> + <valor>	<variable> += <valor>
<variable> = <variable> . <cadena>	<variable> .= <cadena>

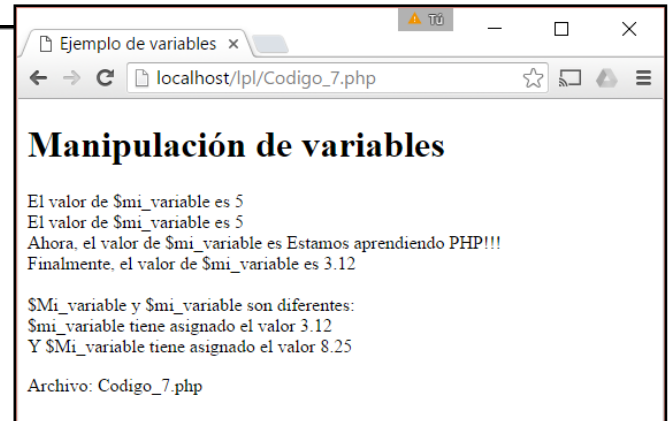
- Operador de concatenación: Para concatenar, se utiliza el “.”

Ejemplo 1: Uso de variables

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

```
1 <!-- Codigo_7.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo de variables</title>
6 </head>
7 <body>
8 <header>
9 <h1>Manipulaci&ocute;n de variables</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 $mi_variable = 5;
15 echo 'El valor de $mi_variable es '.$mi_variable;
16 echo "<br>El valor de \$mi_variable es ".$mi_variable;
17 $mi_variable = "Estamos aprendiendo PHP!!!";
18 echo "<br>Ahora, el valor de \$mi_variable es ".$mi_variable;
19 $mi_variable = 3.12;
20 echo "<br>Finalmente, el valor de \$mi_variable es ".$mi_variable;
21 $Mi_variable = 8.25;
22 echo "<br><br>\$Mi_variable y \$mi_variable son diferentes:";
23 echo "<br>\$mi_variable tiene asignado el valor ".$mi_variable;
24 echo "<br>Y \$Mi_variable tiene asignado el valor ".$Mi_variable;
25 ?>
26 </article>
27 </section>
28 <footer>
29 <p>Archivo: Codigo_7.php</p>
30 </footer>
31 </body>
32 </html>
```

**Ejemplo 2: Uso de variables**

```
1 <!-- Codigo_8.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Asignaci&ocute;n</title>
6 </head>
7 <body>
8 <header>
9 <h1>Operadores de asignaci&ocute;n</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 $mi_variable = 5;
15 echo "Primer valor de \$mi_variable: ".$mi_variable;
16 $mi_variable += 10; //Equivale a $mi_variable = $mi_variable + 10
17 echo "<br>Segundo valor de \$mi_variable: ".$mi_variable;
18 $mi_variable -= 3; //Equivale a $mi_variable = $mi_variable - 3
19 echo "<br>Tercer valor de \$mi_variable: ".$mi_variable;
20 $otra_variable = "Hola ";
21 $mi_otra_variable = "Mundo!!!";
22 echo "<br><br>Primer valor de \$otra_variable: ".$otra_variable;
23 $otra_variable .= $mi_otra_variable;
24 echo "<br>Segundo valor de \$otra_variable: ".$otra_variable;
25 ?>
26 </article>
27 </section>
28 <footer>
29 <p>Archivo: Codigo_8.php</p>
30 </footer>
31 </body>
32 </html>
```



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

- Operadores aritméticos

Operador	Operación
+	Suma de números (enteros o reales)
-	Resta de números (enteros o reales)
*	Multipliación de números (enteros o reales)
/	División de números (enteros o reales)
%	Resto de la división de dos números

- Operadores de incremento y decremento

Operador	Operación
++\$var	Incrementa en 1 y luego retorna el valor de \$var
\$var++	Retorna el valor de \$var y luego lo incrementa en 1
--\$var	Decrementa en 1 y luego retorna el valor de \$var
\$var--	Retorna el valor de \$var y luego lo decrementa en 1

- Operadores de comparación

Operador	Operación
==	Devuelve True si el valor de una variable es igual al de otra
===	Devuelve True si el valor de una variable es igual al de otra y ambas son del mismo tipo
!=	Devuelve True si el valor de una variable es distinto al de otra
!==	Devuelve True si el valor de una variable no es idéntico al de la otra (igual valor e igual tipo)
< / <=	Devuelve True si el valor de la primer variable es menor / menor o igual al de la segunda
> / >=	Devuelve True si el valor de la primer variable es mayor / mayor o igual al de la segunda

- Operadores lógicos

Operador	Operación
&&	Devuelve True si todos los operandos son verdaderos.
	Devuelve True si alguno de los operandos es verdadero.
!	Operador de negación. Devuelve True si el operando es falso y viceversa.
AND	Devuelve True si todos los operandos son verdaderos. Tiene mayor precedencia que &&.
OR	Devuelve True si alguno de los operandos es verdadero. Tiene mayor precedencia que .
XOR	Devuelve True si alguno de los operandos es verdadero, y False cuando ambos son verdaderos o falsos simultáneamente.

- Operador de supresión de error: PHP usa el operador "@" para suprimir errores en tiempo de ejecución. Así, si se produce un error, este no se mostrará y el script continuará su ejecución.

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejemplo 3: Uso del operador de supresión de error

```
1 <!-- Codigo_111.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Operador de supresi&ocute;n de error</title>
6 </head>
7 <body>
8 <header>
9 <h1>Operador de supresi&ocute;n de error</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 @$a = 5/0;
15 if (!$a)
16 { echo "Error. Divisi&ocute;n por cero<br>"; }
17 $b = 3/0;
18 ?>
19 </article>
20 </section>
21 <footer>
22 <p>Archivo: Codigo_111.php</p>
23 </footer>
24 </body>
25 </html>
```

**Ejercicio a**

Programa una página web dinámica de servidor que inicialice 2 variables numéricas num1 y num2 y calcule y muestre al usuario los resultados de la suma, el producto y el promedio de ambos.

Funciones generales para variables

- PHP incorpora 3 funciones especiales que permiten determinar el estado de una variable:

- isset: determina si una variable ha sido definida o no.

boolean isset (mixed \$variable)

- empty: determina si una variable definida está vacía. Se considera que una variable está vacía si tiene alguno de los siguientes valores: "", 0, "0", NULL, FALSE o array().

boolean empty (mixed \$variable)

- unset: permite desasignar variables (borrarlas de memoria).

void unset (mixed \$var1[, mixed \$var2, ...])



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Conversión de tipos

- Si bien las variables de PHP son dinámicas, es posible “forzar” que una variable sea de un determinado tipo, a través de operadores cast.
- Los operadores para conversión explícita son:

Tipo a convertir	Operador cast
Lógico	(bool), (boolean)
Entero	(int), (integer)
Real	(float), (double), (real)
Arreglo	(array)
Objeto	(object)

Funciones para manejo de tipos

- Con la función *settype*, es posible cambiar explícitamente el tipo a una variable:

boolean settype (mixed \$var, string nuevo_tipo)

- Con *gettype*, se puede determinar el tipo actual:

string gettype (mixed \$var)

- De la misma forma, se puede comprobar el tipo de una variable mediante las funciones: *is_array*, *is_bool*, *is_int*, *is_double*, *is_string*, *is_object*, etc.
 - Estas funciones reciben como parámetro la variable a examinar y devuelven un valor lógico que indica si la variable es o no de un tipo determinado.

Funciones para tratamiento de cadenas

- Algunas de las funciones que incorpora PHP para el tratamiento de cadenas son:

Función	Descripción
Trim	Elimina los caracteres en blanco dentro de una cadena
Str_pad	Rellena la cadena con un carácter hasta completar su longitud
Strtolower	Convierte los caracteres de una cadena a minúsculas
Strtoupper	Convierte los caracteres de una cadena a mayúsculas
Ord	Devuelve el valor ASCII de un carácter.
Str_replace	Reemplaza las apariciones de una cadena por otra en la original
Strlen	Devuelve la longitud (cantidad de caracteres) de la cadena
Strpos	Devuelve la posición de un carácter buscado en una cadena
Substr	Devuelve la subcadena de una cadena desde una posición y longitud
Substr_count	Cuenta el número de repeticiones de una subcadena en una cadena

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Función	Descripción
Strcmp	Compara dos cadenas
Strrev	Invierte una cadena
Ucwords	Convierte a mayúsculas el primer carácter de cada palabra de una cadena

Algunas funciones para tratamiento de fechas

- Algunas de las funciones que incorpora PHP para el manejo de fechas son:

Función	Descripción
Checkdate	Valida una fecha. Recibe el mes, el día y el año y devuelve un booleano.
Time	Devuelve la marca de tiempo Unix actual. La variante <i>Microtime</i> la devuelve con microsegundos.
Mktime	Devuelve la marca de tiempo Unix de una fecha. Los parámetros son: hora, min., seg., mes, día y año.
Getdate	Devuelve un arreglo con la información de fecha y hora de una marca de tiempo (opcional).
Date	Formatea una marca de tiempo (parámetro opcional) a un formato particular, indicado mediante una cadena.

- Algunos parámetros de la función date:

Carácter	Descripción
d	Día del mes con 2 dígitos.
D	Representación textual del día con 3 letras.
j	Día del mes.
l	Representación textual completa del día.
z	Día del año
W	Semana del año
n	Mes del año
t	Número de días del mes
F	Representación textual completa del mes

Carácter	Descripción
m	Mes del año con 2 dígitos
M	Representación textual del mes con 3 letras
Y	Año con 4 dígitos
y	Año con 2 dígitos
h	Hora en formato 12 hs.
H	Hora en formato 24 hs.
i	Minutos
s	Segundos
A	AM o PM

Constantes definidas por el usuario

- Las constantes se definen de la siguiente manera:

define(nombre_constante, valor_constante)

LABORATORIO DE PROGRAMACION Y LENGUAJES

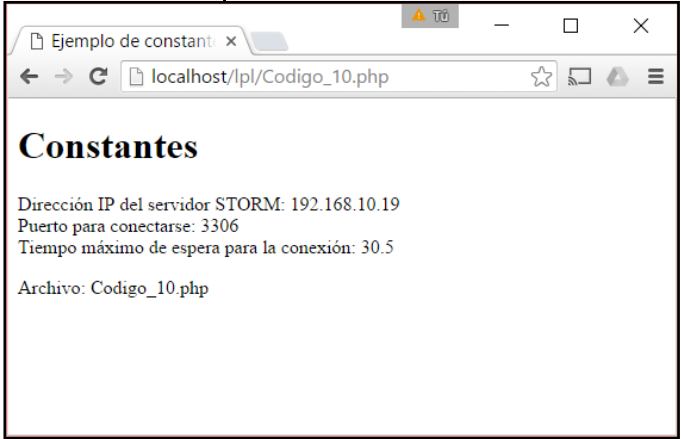
Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejemplo 4: Uso de constantes

```

1 <!-- Codigo_10.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo de constantes</title>
6 </head>
7 <body>
8 <header>
9 <h1>Constantes</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 define('server_storm','192.168.10.19');
15 define('puerto_conexion',3306);
16 define('tiempo_max',30.5);
17 define('verdadero',TRUE);
18 echo "Dirección IP del servidor STORM: ".server_storm."<br>";
19 echo "Puerto para conectarse: ".puerto_conexion."<br>";
20 echo "Tiempo máximo de espera para la conexión: ".tiempo_max."<br>";
21 ?>
22 </article>
23 </section>
24 <footer>
25 <p>Archivo: Codigo_10.php</p>
26 </footer>
27 </body>
28 </html>

```



Constantes de PHP

- PHP incorpora una serie de constantes predefinidas. Las más importantes son:

Constante	Contiene
PHP_VERSION	Versión del intérprete PHP instalada en el servidor web.
PHP_OS	Nombre del sistema operativo instalado en el servidor web.
__FILE__	Nombre del script en ejecución. Almacena la ruta completa, incluyendo carpetas.
__LINE__	Cantidad de líneas del script.

Ejercicio b

Programa una página web dinámica de servidor que inicialice una variable de tipo string con una frase de, al menos, 3 palabras. Luego, determine y muestre al usuario lo siguiente:

- La longitud de la frase (cantidad de caracteres)
- Toda la frase en mayúsculas
- La cantidad de veces que se repite la letra "a"
- La frase invertida



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Estructuras de control

- Las estructuras de control incorporadas a PHP son los bloques, las sentencias de selección y sentencias de iteración.

Bloques

- PHP permite agrupar sentencias de manera tal de formar bloques. La sintaxis para definirlos es:

```
{ Sentencia_1;
  Sentencia_2;
  .....
  Sentencia_n; }
```

Sentencias de selección

- PHP presenta 3 variantes para las sentencias de selección simple:

Sintaxis 1	Sintaxis 2	Sintaxis 3
if (Expresion) { Sentencias; }	if (Expresion) { Sentencias; } else { Sentencias; }	if (Expresion_1) { Sentencias; } elseif (Expresion_2) { Sentencias; } elseif (Expresion_3) { Sentencias; } else { Sentencias; }

- PHP incluye una sentencia de selección múltiple.
- La variable a evaluar puede ser de tipo entero, real o una cadena.
- La sentencia se comporta de la misma forma que en C++ (se debe usar “break” para salir de la opción).
- Incorpora un default, que se ejecutará cuando el valor de la variable no se encuentre en la lista de opciones.

Sintaxis
Switch (\$variable) { case valor_1: sentencias; break; case valor_2: sentencias; break; ... case valor_N: sentencias; break; default: sentencias; }

Sentencias de iteración

- PHP incluye 4 sentencias de iteración: while, do-while, for y foreach.
- La sintaxis de las tres primeras es:

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

While	Do-While	For
<pre>while (condición) { Sentencia_1; Sentencia_2; Sentencia_N; }</pre>	<pre>do { Sentencia_1; Sentencia_2; Sentencia_N; } while (condición)</pre>	<pre>for (inicialización; condición; incremento) { Sentencia_1; Sentencia_2; Sentencia_N; }</pre>

- La evaluación de la condición se realiza:
 - While y For: antes de la ejecución de cada ciclo. Puede no ejecutarse nunca.
 - Do-while: Luego de la ejecución de cada ciclo. Entonces, se ejecuta al menos una vez
- La sentencia For tiene las siguientes partes:
 - Inicialización: asignación que establece el valor inicial de la variable de control.
 - Condición: expresión que comprueba la variable de control.
 - Incremento: define como cambia el valor de la variable de control.
- La sentencia Foreach se utiliza para recorrer objetos o arreglos.

Ejemplo 5: Sentencias iterativas

```

1 <!-- Codigo_12.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Sentencia IF</title>
6 </head>
7 <body>
8 <header>
9 <h1>Sentencia IF</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 $a = 5;
15 $b = 8;
16 if ($a > $b)
17 { echo "a es mayor que b <br>"; }
18 else
19 { echo "a es menor o igual que b <br>"; }
20 //Otro IF
21 if ($b)
22 { echo "¿Se ejecutará este THEN???"; }
23 >
24 </article>
25 </section>
26 <footer>
27 <p>Archivo: Codigo_12.php</p>
28 </footer>
29 </body>
30 </html>
```

```

1 <!-- Codigo_16.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Sentencia WHILE</title>
6 </head>
7 <body>
8 <header>
9 <h1>Sentencia WHILE</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 $i = 6;
15 while ($i)
16 { echo $i--;
17   echo "<br>";
18 }
19 >
20 </article>
21 </section>
22 <footer>
23 <p>Archivo: Codigo_16.php</p>
24 </footer>
25 </body>
26 </html>
```

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejercicio c

Modifique el script del ejercicio 'a' de manera tal que, si num1 es mayor a num2, entonces calcule e informe $\text{num1}^{\text{num2}}$. Si num1 es menor o igual a num2, entonces debe calcular el cociente y el resto de la división de num2 por num1. La potencia debe calcularse utilizando las multiplicaciones sucesivas y el cociente y el resto deben calcularse utilizando las restas sucesivas.

Ejercicio d

Desarrollar un script que imprima en una página web las tablas de multiplicar del 1 al 10, de acuerdo al formato que se muestra en la siguiente figura:

Tabla de multiplicar del: 5

*	1	2	3	4	5	6	7	8	9	10
5	5	10	15	20	25	30	35	40	45	50

Ejercicio e

En el manual de PHP, investigar sobre las funciones **rand** y **srand**. Luego, probar los siguientes scripts y analizar los resultados obtenidos:

```

1  <?php
2  srand((double)microtime() * 1000000);
3  $a = rand(0, 3);
4  $b = rand(0, 3);
5  ($a==$b)?($mensaje = "$a = $b"):( $mensaje = "$a != $b");
6  echo $mensaje;
7  ?>

```

```

1  <?php
2  srand((double)microtime() * 1000000);
3  $d = $c = $b = $a = rand(0, 10);
4  echo "\$a = ".$a;
5  echo "<br>\$b = ".$b;
6  echo "<br>\$c = ".$c;
7  echo "<br>\$d = ".$d;
8  ?>

```

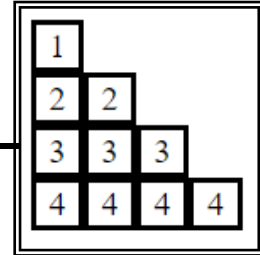
LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejercicio f

Desarrolle un script que, a partir de un número entero "n", con un valor entre 2 y 15 generado aleatoriamente, imprima en la página web una pirámide cuya cúspide tiene el valor 1 y cuya base tiene el valor "n" repetido n veces.

Por ejemplo, para n = 4, debe generar la siguiente pirámide →→→→→



Arreglos

- PHP incluye la estructura de datos arreglo y la implementa con dos variantes:
 - Como arreglo enumerado: el índice es numérico y, por defecto, comienza en 0.
 - Como arreglo asociativo: el índice es una cadena de caracteres (como si fuera una estructura de registro).
- Los arreglos de PHP tienen la particularidad de que los elementos no necesariamente deben ser todos del mismo tipo de datos.
- Además, en base a éstos, se pueden formar arreglos multidimensionales.

Ejemplos

- Arreglo enumerado:

\$totales	Valor	14	20	15	12
	Indice	0	1	2	3

- Arreglo asociativo:

\$claustro	Valor	125	44	22	12
	Indice	alumnos	docentes	graduados	no_docente

- Arreglo enumerado con elementos de diferente tipo:

\$de_todo	Valor	14	"vector"	5.14	[2,8,5,9]
	Indice	0	1	2	3

Ej: \$de_todo[3][1] = 8 (arreglo multidimensional)

LABORATORIO DE PROGRAMACION Y LENGUAJES

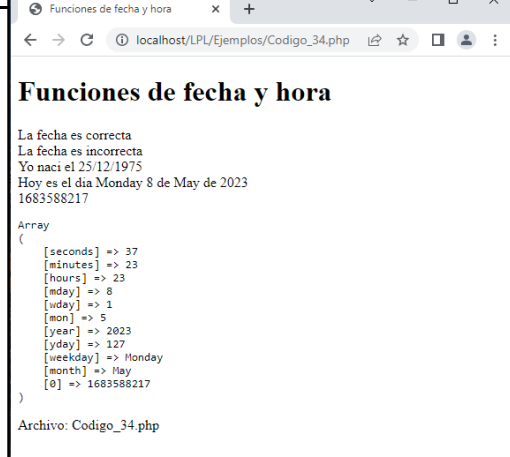
Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejemplo 6: Uso de arreglos y funciones de fecha y hora

```

1 <!-- Código_34.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Funciones de fecha y hora</title>
6 </head>
7 <body>
8 <header>
9 <h1>Funciones de fecha y hora</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 if (checkdate(12,25,2023))
15 { echo "La fecha es correcta<br>"; }
16 else
17 { echo "La fecha es incorrecta<br>"; }
18 if (checkdate(2,29,2023))
19 { echo "La fecha es correcta<br>"; }
20 else
21 { echo "La fecha es incorrecta<br>"; }
22 $fecha_nac = date("d/m/Y",mktime(0,0,0,12,25,1975));
23 echo "Yo nací el ".$fecha_nac."<br>";
24 $fecha = getdate();
25 echo "Hoy es el día ".$fecha['weekday']." ".$fecha['mday']." de ".$fecha['month']." de ".$fecha['year'];
26 echo "<br>";
27 echo time();
28 echo "<pre>";
29 print_r($fecha);
30 echo "</pre>";
31 ?>
32 </article>
33 </section>
34 <footer>
35 <p>Archivo: Código_34.php</p>
36 </footer>
37 </body>
38 </html>

```



Creación de arreglos

- Los arreglos de PHP no se declaran, y se crean e inicializan con el constructor `array`.
- A los elementos se accede con el nombre del arreglo y el índice entre corchetes.

Agregar elementos a los arreglos

- Para agregar elementos a un arreglo, se debe considerar:
 - Si el arreglo es enumerado:
 - Si la cantidad de elementos es conocida: `$vector[$i] = valor;`
 - Si la cantidad de elementos es desconocida: `$vector[] = valor;`

```

1 <!-- Código_27.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Creación de arreglos</title>
6 </head>
7 <body>
8 <header>
9 <h1>Arreglos</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 //Creo un arreglo vacío
15 $vec_enum = array();
16 //Creo el arreglo enumerado (1,5,8)
17 $vec_enum2 = array(1,5,8);
18 //Creo el arreglo enumerado ('AEP','EZE','CRD','COR')
19 $vec_enum3 = array('AEP','EZE','CRD','COR','MDZ');
20 //Creo el arreglo asociativo $trafico
21 $trafico = array('AEP' => 21, 'EZE' => 22.4,
22                 'CRD' => 2.5, 'COR' => 4.9,
23                 'MDZ' => 2.9, 'OTROS' => 46.3);
24 //Creo un arreglo multidimensional
25 $trafico2 = array('AEP' => array(8,9,4),
26                  'CRD' => array(2,0.5),
27                  'COR' => array(1,0.2,2.7,1),
28                  'MDZ' => array(1.4,1.5));
29 ?>
30 </article>
31 </section>
32 <footer>
33 <p>Archivo: Código_27.php</p>
34 </footer>
35 </body>
36 </html>

```



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

- Si el arreglo es asociativo:
 - Nombre del nuevo índice: `$vector['índice'] = valor;`
- PHP también incluye la función `array_push`, la que permite incorporar nuevos elementos:
`int array_push (array arreglo, mixed elementos)`
- `Array_push` devuelve el nuevo número de elementos

Recorrido e impresión de arreglos

- Para los arreglos enumerados puede utilizarse un ciclo *for*.
 - Con la función *count* se obtiene la cantidad de elementos.
- También puede usarse la sentencia *foreach*:

Sintaxis 1	Sintaxis 2
<pre>foreach (\$vector as \$valores) { echo \$valores; }</pre>	<pre>foreach(\$vec as \$ind => \$val) { echo "Campo: ".\$ind; echo "Valor: ".\$val; }</pre>

- Con la función *print_r*, se puede hacer una impresión del arreglo.

Ejemplo 7: recorrido de arreglos

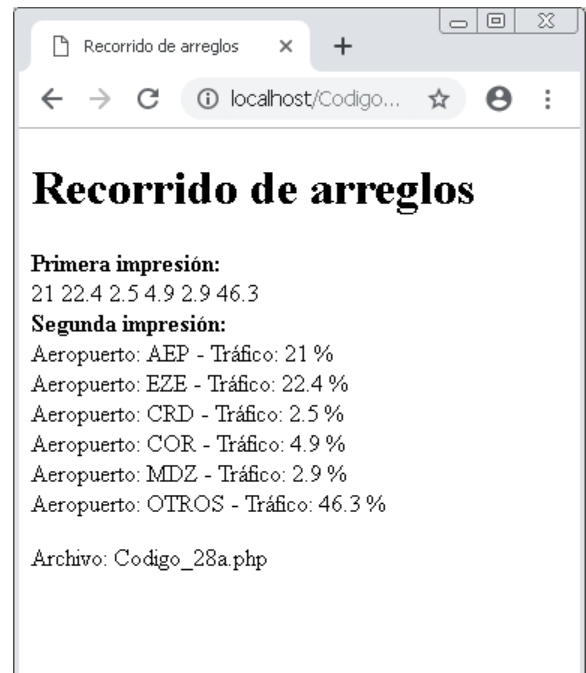
LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

```

1 <!-- Codigo_28.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Recorrido de arreglos</title>
6 </head>
7 <body>
8 <header>
9 <h1>Recorrido de arreglos</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 $trafico = array('AEP' => 21, 'EZE' => 22.4,
15                 'CRD' => 2.5, 'COR' => 4.9);
16 $trafico['MDZ'] = 2.9;
17 $trafico['OTROS'] = 46.3;
18 //Primer recorrido con FOREACH
19 echo "<strong>Primera impresión:</strong><br>";
20 foreach($trafico as $elemento)
21 { echo $elemento.' ' ; }
22 echo "<br>";
23 //Segundo recorrido con FOREACH
24 echo "<strong>Segunda impresión:</strong><br>";
25 reset($trafico); //Vuelvo el cursor al inicio del vector
26 foreach($trafico as $clave => $valor)
27 { echo "Aeropuerto: ".$clave." - ";
28   echo "Tráfico: ".$valor." % <br>"; }
29 ?>
30 </article>
31 </section>
32 <footer>
33 <p>Archivo: Codigo_28.php</p>
34 </footer>
35 </body>
36 </html>

```



Algunas funciones para manipular arreglos

Función	Descripción
Array_shift	Extrae el primer elemento del arreglo.
Array_pop	Extrae el último elemento del arreglo.
Array_slice	Extrae una parte del arreglo. Recibe como parámetro el índice y la cantidad de elementos a extraer.
In_array	Determina si un elemento está en un arreglo.
Sort	Ordena un arreglo en forma ascendente. Variantes: rsort (descendente), ksort (ordena por clave), asort (ordena manteniendo la clave).
Reset / End	Pone el cursor del arreglo en su primer / último elemento.
Array_intersect	Determina los elementos que tienen en común dos o mas arreglos y los devuelve en otro.
Array_merge	Combina los elementos de dos o mas arreglos y los devuelve en otro.
Array_diff	Determina los elementos de un arreglo que no están en otros y los devuelve en otro arreglo.
Shuffle	Desordena y mezcla los elementos de un arreglo
Implode	Crea una cadena a partir de los elementos de un arreglo. Los elementos son concatenados a través de un carácter o cadena.
Explode	Crea un arreglo a partir de una cadena. Para determinar los elementos del arreglo, se debe especificar el carácter separador de la cadena.



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejercicio g

Modifique el script del ejercicio 'b', agregándole la generación de un arreglo que contenga todas las potencias de num1, desde 0 hasta num2. Luego, imprimir el arreglo.

Ejercicio h

Programa una página web que genere un arreglo de diez números enteros aleatorios y luego determine y muestre al usuario:

- El arreglo generado.
- La cantidad y suma de todos los elementos.
- El primer y último elemento del arreglo.
- El menor y el mayor elemento del arreglo.
- Si el número 5 está en el arreglo.
- El promedio de los valores que se encuentran en el arreglo.
- El arreglo ordenado de menor a mayor.

Arreglos superglobales de PHP

- PHP incluye varios arreglos asociativos predefinidos que están siempre disponibles durante la ejecución de un script:

Arreglo	Descripción
<code>\$GLOBALS</code>	Contiene referencias a todas las variables que definidas en el script
<code>\$_SERVER</code>	Es creado por el servidor web. Almacena información de las cabeceras HTTP, rutas y ubicaciones del script, como ser: <code>PHP_SELF</code> , <code>SERVER_ADDR</code> , <code>SERVER_NAME</code> , <code>SERVER_PROTOCOL</code> , <code>REQUEST_METHOD</code> , <code>REQUEST_TIME</code> , <code>QUERY_STRING</code> , <code>REMOTE_ADDR</code> , <code>REMOTE_PORT</code> , <code>HTTPS</code>
<code>\$_GET</code>	Contiene todas las variables enviadas desde un formulario HTML al script usando HTTP GET
<code>\$_POST</code>	Contiene todas las variables enviadas desde un formulario HTML al script utilizando el método HTTP POST
<code>\$_FILES</code>	Contiene información de los archivos enviados al servidor web a través del método HTTP POST (operación de upload de archivos)
<code>\$_COOKIES</code>	Contiene todas las variables enviadas al script actual usando cookies HTTP
<code>\$_SESSION</code>	Contiene todas las variables de sesión definidas en el script. Esto permite acceder a variables definidas en otros scripts que se ejecutan en la sesión actual

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejemplo 8: Constantes y acceso al arreglo superglobal \$_SERVER

```
1 <!-- Codigo_11.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Constantes y variables de PHP</title>
6 </head>
7 <body>
8 <header>
9 <h1>Constantes y variables de PHP</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 echo "Contenido de __FILE__ : ".__FILE__;
15 echo "<br>";
16 echo "Contenido de \$_SERVER['PHP_SELF'] : ".\$_SERVER['PHP_SELF'];
17 echo "<br>";
18 echo "Contenido de \$_SERVER['SERVER_ADDR'] : ".\$_SERVER['SERVER_ADDR'];
19 echo "<br>";
20 echo "Contenido de \$_SERVER['SERVER_NAME'] : ".\$_SERVER['SERVER_NAME'];
21 ?>
22 </article>
23 </section>
24 <footer>
25 <p>Archivo: Codigo_11.php</p>
26 </footer>
27 </body>
28 </html>
```

**Objetos**

- Aunque PHP no es un lenguaje puro de objetos, soporta el paradigma.
- La primera versión en soportar el paradigma fue la 3, aunque solo se podían definir clases y crear objetos. Tanto las propiedades como los métodos eran todos públicos.
- A partir de PHP 5, se aprovecha la mayoría de las ventajas del paradigma, entre ellas la posibilidad de definir niveles de acceso para propiedades y métodos.

Definición de clases

- La definición de una clase en PHP se realiza de acuerdo a la siguiente sintaxis:



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

```
class Nombre_Clase

{ [public | protected | private] $propiedad_1 [= valor];

.....

[public | protected | private] $propiedad_n [= valor];

[public | protected | private] function metodo_1([$parametro])

{ ..... }

[public | protected | private] function metodo_N([$parametro])

{ ..... }

}
```

- Las propiedades y métodos de la clase se definen con alguno de los siguientes niveles de acceso:
 - Public: se permite el acceso desde cualquier ámbito, incluso desde otra clase.
 - Private: sólo se permite el acceso desde la clase a la que pertenece.
 - Protected: sólo se permite el acceso desde la clase a la que pertenece y desde las clases hijas (subclases).
- Dentro de los métodos de la clase, y para referenciar al objeto que se está usando en ese momento, se utiliza una pseudo variable *\$this*.
- La definición de la clase debe guardarse en un archivo con extensión “.class.php”

Definición de métodos

- La definición de métodos se realiza como la definición de funciones de PHP, las que pueden o no tener parámetros y devolver o no un único valor.
- La sintaxis para definir métodos es:

```
[public | protected | private] function nom_metodo ([$par1, $par2, ..., $parN])

{      Sentencia_1;

...

Sentencia_N;

[return $valor_a_retornar;]      }
```

- Los parámetros no se declaran de un tipo determinado y pueden tener valores por defecto.



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Métodos especiales de una clase

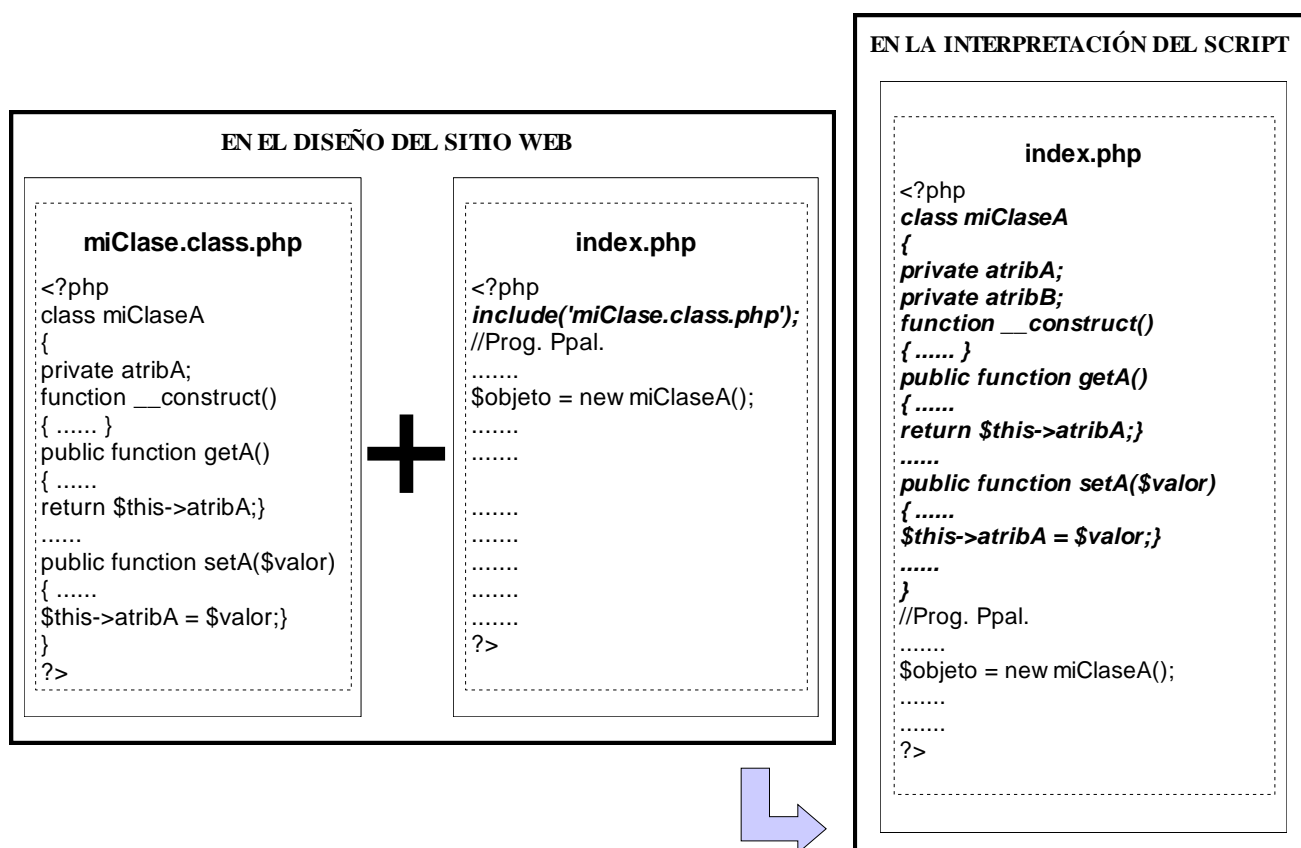
- `__construct`: se ejecuta cuando se crea el objeto. Se utiliza para inicializar las propiedades.
- `__destruct`: se ejecuta cuando se destruye el objeto (al finalizar el script). Puede invocarse explícitamente usando la función `unset($objeto)`.
- `__set`: se ejecuta cuando se trata de asignar un valor a una propiedad que no pertenece a la clase.
- `__get`: se ejecuta cuando se intenta obtener el valor de una propiedad que no está definida en la clase.
- `__call`: se ejecuta cuando se llama a un método que no está definido en la clase.

Utilización de las clases en un script PHP dentro de una página web

- Para utilizar una clase en un script PHP, se deben incluir en el script alguna de las siguientes funciones, enviando como parámetro el nombre del archivo que contiene la definición de la clase (archivo con extensión `.class.php`):
 - `include`
 - `include_once`
 - `require`
 - `require_once`
- La diferencia entre las funciones `include/include_once` y `require/require_once` es que, en caso de que no se encuentre la librería (o archivo que contiene la definición de la clase), `include` produce una advertencia y `require` finaliza la ejecución del script.
- Además, las funciones `include_once` y `require_once`, chequean previamente si la librería fue cargada previamente o no en la ejecución del script. Si la librería (o la definición de la clase) se incorpora 2 ó más veces en el script que la invoca, se produce un error en ejecución por sobrecarga de definiciones de clases.

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor



Instanciación de clases – Creación de objetos

- La creación de objetos se realiza a través del operador *new*, seguido del nombre de la clase y los atributos (si son necesarios).

\$objeto = new clase_objeto()

- Una de las ventajas del paradigma orientado a objetos es la sobrecarga de métodos. Así, se podrían definir varios constructores para una misma clase.
 - Como PHP no soporta la sobrecarga de métodos, una opción es inicializar los parámetros dentro del método constructor, utilizando valores por defecto.

Ejemplo 9: Definición de clases y creación de objetos

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

```

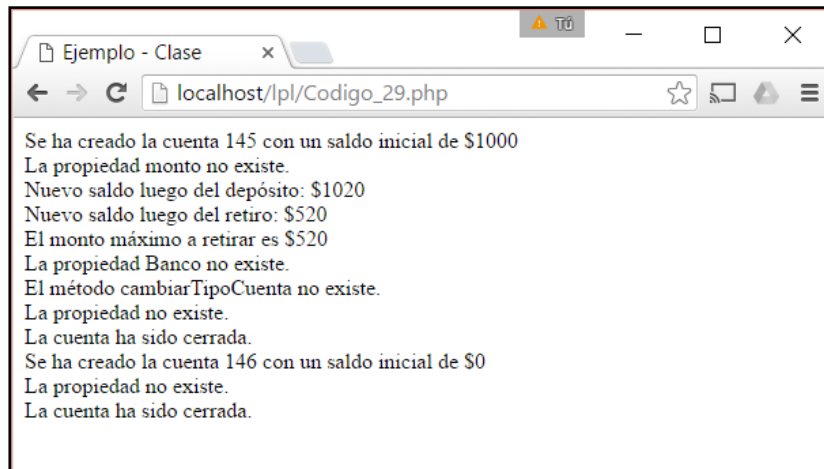
1  <?php
2  class Cuenta
3  {   private $numero_cuenta;
4      private $monto;
5
6      function __construct($num_cuenta, $inicial = 0.0)
7      {   $this->numero_cuenta = $num_cuenta;
8          $this->monto = $inicial;
9      }
10
11     function __destruct()   {   echo "La cuenta ".$this->$numero_cuenta." ha sido cerrada.<br>";   }
12
13     function __get($propiedad) {   echo "La propiedad ".$propiedad." no existe.<br>";   }
14
15     function __set($propiedad, $valor) {   echo "La propiedad ".$propiedad." no existe.<br>";   }
16
17     function __call($metodo, $atributos) {   echo "El método ".$metodo." no existe.<br>";   }
18
19     public function getNumeroCuenta()   {   return $this->numero_cuenta;   }
20
21     public function getMonto()           {   return $this->monto;   }
22
23     public function deposito($monto_deposito)   {   $this->monto += $monto_deposito;   }
24
25     public function retiro($monto_retiro)
26     {   if ($monto_retiro > $this->getMonto())
27         {   echo "El monto máximo a retirar es ".$this->getMonto()."<br>";   }
28         else
29         {   $this->monto -= $monto_retiro;   }
30     }
31 }
32 ?>

```

```

1  <!-- Código_29.php -->
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5  <title>Ejemplo - Clase</title>
6  </head>
7  <body>
8  <section>
9  <article>
10 <?php
11 include_once("Codigo_29.class.php");
12 //Creo un objeto. Se envía un mensaje al método constructor
13 $MiCuenta = new Cuenta(145, 1000);
14 echo "Se ha creado la cuenta ".$MiCuenta->getNumeroCuenta()." con un saldo inicial de $".$MiCuenta->getMonto()."<br>";
15 //Trato de acceder a propiedades privadas
16 echo $MiCuenta->monto;
17 //Pruebo el método depósito
18 $MiCuenta->deposito(20);
19 echo "Nuevo saldo luego del depósito: $".$MiCuenta->getMonto()."<br>";
20 //Pruebo el método retiro
21 $MiCuenta->retiro(500);
22 echo "Nuevo saldo luego del retiro: $".$MiCuenta->getMonto()."<br>";
23 $MiCuenta->retiro($MiCuenta->getMonto() + 10);
24 //Pruebo los métodos __get, __set y __call
25 $MiCuenta->Banco = "Nuevo Banco Patagónico";
26 $MiCuenta->cambiarTipoCuenta();
27 //Pruebo el destructor
28 unset($MiCuenta);
29 //Pruebo el constructor con el parámetro por defecto
30 $OtraCuenta = new Cuenta(146);
31 echo "Se ha creado la cuenta ".$OtraCuenta->getNumeroCuenta()." con un saldo inicial de $".$OtraCuenta->getMonto()."<br>";
32 ?>
33 </article>
34 </section>
35 </body>
36 </html>

```

LABORATORIO DE PROGRAMACION Y LENGUAJES**Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor****Herencia - Subclases**

- PHP permite crear subclases a través del concepto de herencia.
- Para esto, admite solamente herencia simple, aunque mediante las interfaces, se puede “simular” herencia múltiple.
- Con la herencia, la subclase hereda todas las propiedades y métodos de su clase padre, siempre que no sean privados.
- Además, es posible redefinir alguno o todos los métodos heredados, siempre y cuando no estén definidos como finales.

Ejemplo 10: Definición y utilización de subclases

```
1 <?php
2 class Caja_ahorro extends Cuenta
3 {   private $tasa_interes = 0.5;
4
5     public function __construct($num_cuenta, $inicial = 0.0, $tasa = 0.5)
6     {   /*
7         El método constructor del padre no se llama implícitamente
8         cuando se genera un objeto de la subclase, entonces debe
9         invocarse explícitamente.
10        */
11        parent::__construct($num_cuenta, $inicial);
12        $this->tasa_interes = $tasa;
13        echo "<strong>Tasa de inter&eacute;s mensual: </strong>". $this->tasa_interes. "<br>";
14    }
15
16    public function __destruct()
17    {   echo "La caja de ahorros ha sido cerrada.<br>"; }
18
19    public function getTasaInteres()
20    {   return $this->tasa_interes;
21    }
22
23    public function setTasaInteres($nueva_tasa)
24    {   $this->tasa_interes = $nueva_tasa;
25    }
26 }
27 ?>
```

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

```

1 <?php
2 class Cuenta
3 {
4     protected $numero_cuenta;
5     protected $monto;
6
7     public function __construct($num_cuenta, $inicial = 0.0)
8     {
9         $this->numero_cuenta = $num_cuenta;
10        $this->monto = $inicial;
11    }
12
13    public function __destruct()
14    {
15        echo "La cuenta ha sido cerrada.<br>";
16    }
17
18    public function __get($propiedad)
19    {
20        echo "La propiedad ".$propiedad." no existe.<br>";
21    }
22
23    public function __set($propiedad, $valor)
24    {
25        echo "La propiedad ".$propiedad." no existe.<br>";
26    }
27
28    public function __call($metodo, $atributos)
29    {
30        echo "El método ".$metodo." no existe.<br>";
31    }
32
33    public function getNumeroCuenta() { return $this->numero_cuenta; }
34
35    public function getMonto() { return $this->monto; }
36
37    public function deposito($monto_deposito) { $this->monto += $monto_deposito; }
38
39    public function muestroDatosCuenta()
40    {
41        echo "<strong>Datos de la caja de ahorros:</strong><br>";
42        echo "<strong>Número de cuenta: </strong>".$this->numero_cuenta."<br>";
43        echo "<strong>Saldo inicial: </strong>".$this->monto."<br>";
44    }
45
46    public function retiro($monto_retiro)
47    {
48        if ($monto_retiro > $this->getMonto())
49        {
50            echo "Error. El monto máximo a retirar es ".$this->getMonto()."<br>";
51        }
52        else
53        {
54            $this->monto -= $monto_retiro;
55        }
56    }
57 }
58 ?>

```

Ejemplo - Subclase

localhost/lpl/Codigo_30.php

Tasa de interés mensual: 0.8%

Datos de la caja de ahorros:

Número de cuenta: 147

Saldo inicial: \$1000

La propiedad tasa_interes no existe.

Nuevo saldo luego del depósito: \$1200

Nuevo saldo luego del retiro: \$460

Error. El monto máximo a retirar es \$460

La propiedad Banco no existe.

El método cambiarTipoCuenta no existe.

Tasa de interés mensual: 0.5%

Datos de la caja de ahorros:

Número de cuenta: 148

Saldo inicial: \$0

La caja de ahorros ha sido cerrada.

La caja de ahorros ha sido cerrada.

```

1 <!-- Codigo_30.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Subclase</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <?php
11 function __autoload($clase)
12 {
13     $archivo = "Codigo_30/".$clase.".class.php";
14     require_once($archivo);
15 }
16 //Creo un objeto. Se envía un mensaje al método constructor
17 $MiCaja = new Caja_ahorro(147, 1000, 0.8);
18 $MiCaja->muestroDatosCuenta();
19 //Trato de acceder a propiedades privadas
20 echo $MiCaja->tasa_interes;
21 //Pruebo el método depósito, que se hereda
22 $MiCaja->deposito(200);
23 echo "Nuevo saldo luego del depósito: ".$MiCaja->getMonto()."<br>";
24 //Pruebo el método retiro
25 $MiCaja->retiro(740);
26 echo "Nuevo saldo luego del retiro: ".$MiCaja->getMonto()."<br>";
27 $MiCaja->retiro($MiCaja->getMonto() + 10);
28 //Pruebo los métodos __set y __call, que también son heredados
29 $MiCaja->Banco = "Nuevo Banco Patagónico";
30 $MiCaja->cambiarTipoCuenta("Cuenta corriente");
31 //Pruebo el constructor con los parámetros por defecto
32 $OtraCaja = new Caja_ahorro(148);
33 $OtraCaja->muestroDatosCuenta();
34 ?>
35 </article>
36 </section>
37 </body>
38 </html>

```




LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Propiedades y métodos estáticos

- PHP permite la definición de propiedades y métodos estáticos dentro de una clase.
- Esta categorización permite definir propiedades y métodos de clase, por lo cual el acceso a ellos no se realiza a través de los objetos.
- Además, en las clases hijas, no es posible redefinirlos.
- La sintaxis para definir propiedades y métodos estáticos es:
 - [public | protected | private] static *\$propiedad* [= *valor*];
 - [public | protected | private] static function *metodo*([*\$par*])
- La definición de un método como estático implica que dentro de él sólo puede accederse a las propiedades estáticas de la clase, a las variables locales al método y a los parámetros.
- El acceso a las propiedades estáticas dentro de los métodos de la clase se realiza utilizando *self*, seguido del operador de resolución (::). Si se quiere acceder desde los métodos de una subclase, se utiliza *parent* en lugar de *self*.
- Este operador también se utiliza para acceder a las propiedades y métodos estáticos de una clase en un script. En ese caso, no se utiliza *self*, sino el nombre de la clase.

Clases abstractas

- Las clases abstractas son aquellas que no pueden ser instanciadas, es decir, son plantillas para otras clases que si pueden ser instanciadas.
- Dentro de ellas, se definen propiedades y métodos, y éstos últimos a su vez pueden ser abstractos o no.
- Los métodos definidos como abstractos son declarados en la clase abstracta y definidos en las clases derivadas.
- Como la clase abstracta no se puede instanciar, sus propiedades y métodos deben tener un nivel de acceso público o protegido.

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejemplo 11: Definición y uso de clases abstractas

```

1 <?php
2 class Caja_ahorro extends Cuenta
3 {     private $tasa_interes = 0.5;
4
5     public function __construct($num_cuenta, $inicial = 0.0, $tasa = 0.5)
6     {     $this->numero_cuenta = $num_cuenta;
7         $this->monto = $inicial;
8         $this->tasa_interes = $tasa;
9     }
10
11    public function __destruct()
12    {     echo "La caja de ahorros ha sido cerrada.<br>";
13
14    public function getTasaInteres()
15    {     return $this->tasa_interes;
16
17    public function setTasaInteres($nueva_tasa)
18    {     $this->tasa_interes = $nueva_tasa;
19
20    public function muestroDatosCuenta()
21    {     echo "<b>Datos de la caja de ahorros:</b><br>";
22        echo "<b>Número de cuenta: </b>". $this->numero_cuenta. "<br>";
23        echo "<b>Saldo inicial: </b>$". $this->monto. "<br>";
24        echo "<b>Tasa de interés mensual: </b>". $this->tasa_interes. "%<br>"; }
25    }
26    ?>

```

```

1 <?php
2 abstract class Cuenta
3 {     protected $numero_cuenta;
4     protected $monto;
5
6     public function getNumeroCuenta()
7     {     return $this->numero_cuenta;
8
9     public function getMonto()
10    {     return $this->monto;
11
12    ?>

```

```

1 <!-- Codigo_31.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Clase abstracta</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <?php
11 function __autoload($clase)
12 {     $archivo = "Codigo_31.". $clase. ".class.php";
13     require_once ($archivo);
14 }
15
16 //Creo un objeto. Se envía un mensaje al método constructor
17 $MiCaja = new Caja_ahorro(148, 200);
18 $MiCaja->muestroDatosCuenta();
19 //Cambio la tasa de interés
20 $MiCaja->setTasaInteres(0.75);
21 //Muestro la nueva tasa de interés
22 echo "La nueva tasa de interés de la cuenta ". $MiCaja->getNumeroCuenta();
23 echo " es ". $MiCaja->getTasaInteres(). "%<br>";
24 ?>
25 </article>
26 </section>
27 </body>
28 </html>

```

Ejemplo - Clase abstr...

localhost/lpl/Codigo_31.php

Datos de la caja de ahorros:
 Número de cuenta: 148
 Saldo inicial: \$200
 Tasa de interés mensual: 0.5%
 La nueva tasa de interés de la cuenta 148 es 0.75%
 La caja de ahorros ha sido cerrada.

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Interfaces

- Las interfaces se utilizan principalmente para declarar un conjunto de métodos que deberán ser definidos en las clases que las implementen. En una interfaz no se definen propiedades.
- Así, los métodos declarados dentro de la interfaz contienen su nombre, argumentos y son públicos.
- Las clases pueden implementar mas de una interfaz. De esta forma, se puede simular la herencia múltiple.
- La sintaxis para implementar varias interfaces es:

class MiClase implements Int_1, Int_2, ..., Int_n

Ejemplo 12: Definición y uso de interfaces

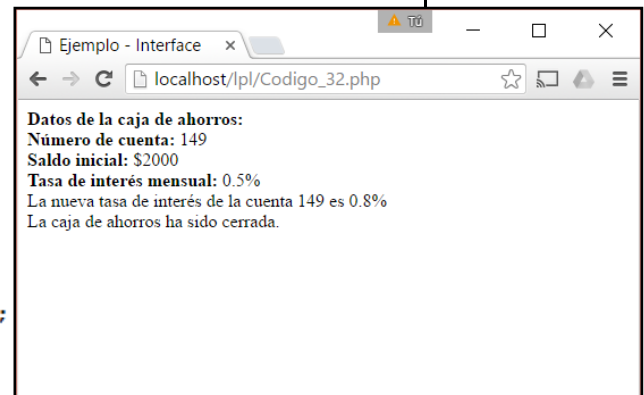
```
1  <?php
2  class Caja_ahorro implements Cuenta
3  //Todos los métodos de la interface deben ser implementados
4  {   private $numero_cuenta;
5      private $saldo;
6      private $tasa_interes = 0.5;
7
8      public function __construct($num_cuenta, $inicial = 0.0, $tasa = 0.5)
9      {   $this->numero_cuenta = $num_cuenta;
10         $this->saldo = $inicial;
11         $this->tasa_interes = $tasa;      }
12
13     public function __destruct()
14     {   echo "La caja de ahorros ha sido cerrada.<br>";      }
15
16     public function getNumeroCuenta()
17     {   return $this->numero_cuenta;      }
18
19     public function getSaldo()
20     {   return $this->saldo;      }
21
22     public function getTasaInteres()
23     {   return $this->tasa_interes;      }
24
25     public function setTasaInteres($nueva_tasa)
26     {   $this->tasa_interes = $nueva_tasa;      }
27
28     public function muestroDatosCuenta()
29     {   echo "<b>Datos de la caja de ahorros:</b><br>";
30         echo "<b>Número de cuenta: </b>". $this->numero_cuenta. "<br>";
31         echo "<b>Saldo inicial: </b>". $this->saldo. "<br>";
32         echo "<b>Tasa de interés mensual: </b>". $this->tasa_interes. "%<br>"; }
33     }
34  ?>
```

```
1  <?php
2  interface Cuenta
3  {
4      public function getNumeroCuenta();
5      public function getSaldo();
6  }
7  ?>
```

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

```
1 <!-- Codigo_32.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Interface</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <?php
11 function __autoload($clase)
12 {   $archivo = "Codigo_32.".$clase.".class.php";
13     require_once($archivo);
14 }
15
16 //Creo un objeto. Se envía un mensaje al método constructor
17 $MiCaja = new Caja_ahorro(149, 2000);
18 $MiCaja->muestroDatosCuenta();
19 //Cambio la tasa de interés
20 $MiCaja->setTasaInteres(0.8);
21 //Muestro la nueva tasa de interés
22 echo "La nueva tasa de intereseacute;s de la cuenta ".$MiCaja->getNumeroCuenta();
23 echo " es ".$MiCaja->getTasaInteres()."€<br>";
24 ?>
25 </article>
26 </section>
27 </body>
28 </html>
```



Operador Instanceof

- Es un operador que permite determinar si un objeto es una instancia de una clase:

```
1 <!-- Codigo_33.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head><title>Ejemplo - Operador Instanceof</title></head>
5 <body>
6 <section>
7 <article>
8 <?php
9 include_once("Codigo_30.Cuenta.class.php");
10 include_once("Codigo_30.Caja_ahorro.class.php");
11 $MiCaja = new Caja_ahorro(150, 2000);
12 if ($MiCaja instanceof Caja_ahorro)
13 {   echo 'El objeto $MiCaja es una instancia de la clase Caja_ahorro<br>'; }
14 else
15 {   echo 'El objeto $MiCaja no es una instancia de la clase Caja_ahorro<br>'; }
16 ?>
17 </article>
18 </section>
19 </body>
20 </html>
```



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Algunas funciones de clases / objetos

Función	Descripción
Get_class	Determina el nombre de la clase de un objeto
Get_class_methods	Crea un arreglo con los métodos de una clase
Get_class_vars	Crea un arreglo con las propiedades de una clase
Get_parent_class	Determina el nombre de la clase padre de un objeto o clase
Class_exists	Determina si una clase ha sido definida o no.
Interface_exists	Determina si una interface ha sido definida o no
Method_exists	Determina si un método ha sido definido en una clase o no.
Property_exists	Determina si un objeto o clase tiene una propiedad específica.
Is_subclass_of	Determina si un objeto tiene a una cierta clase (parámetro) como padre.

Ejercicio i

Utilizando el paradigma orientado a objetos, programe en PHP una calculadora de dos números complejos generados aleatoriamente. La calculadora, utilizando los números generados, deberá calcular y mostrar al usuario la siguiente información:

- Los dos números complejos generados, en forma binómica
- El módulo de cada uno de los números complejos generados
- El conjugado de cada uno de los números complejos generados
- El resultado de la suma y de la resta de los dos números complejos generados (ambos resultados deben mostrarse)

Sobre los números complejos

- Un número complejo z se define como un par ordenado de números reales 'a' y 'b':

$z = (a, b)$, donde 'a' representa la parte real y 'b' representa la parte imaginaria.

Ejemplo: $z = (2, -3)$

- Un número complejo z se puede expresar también de **forma binómica**:

$z = a + bi$, donde i representa a la unidad imaginaria (raíz cuadrada de -1)

Ejemplo: $z = (2, -3)$ se expresa de forma binómica como $z = 2 - 3i$

- Todo número complejo z tiene un **módulo**, que se calcula de la siguiente forma:

$$|z| = \sqrt{a^2 + b^2}$$



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Ejemplo: si $z = 2 - 3i$, entonces $|z| = \sqrt{4 + 9} = \sqrt{13}$

- Todo número complejo z tiene un **conjugado** \bar{z} , que se calcula de la siguiente forma:

$$\text{si } z = a + bi, \text{ entonces } \bar{z} = a - bi$$

Ejemplo: si $z = 2 - 3i$, entonces $\bar{z} = 2 + 3i$

- La **suma** de dos números complejos $z_1 = a_1 + b_1i$ y $z_2 = a_2 + b_2i$ se define de la siguiente forma:

$$z_1 + z_2 = (a_1 + a_2) + (b_1 + b_2)i$$

Ejemplo: si $z_1 = 2 - 3i$ y $z_2 = -3 + 5i$, entonces $z_1 + z_2 = -1 + 2i$



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Lista de ejercicios prácticos

Resolver los siguientes ejercicios utilizando hojas de estilos para la presentación de los elementos. Los script php, deben estar en una librería con el fin de reutilizarse, en caso de ser posible, en los distintos ejercicios.

1. Diseñar un algoritmo que muestre por separado los números pares, los números impares y los números primos, incluidos en el rango de 0 a 1000. Presentar en una página web el resultado obtenido.
2. Simular un calendario de un mes de 31 días, suponer que el día 1 es domingo. tener en cuenta que todos los domingos deben estar en colorado y los días sábados en verde. Presentar el calendario en una página web.
3. Mostrar una página web, la tabla de multiplicar (de 1 a 10) generada a partir de un número aleatorio. Por ejemplo: Número aleatorio: 5, se muestra la tabla del 5:

1	x	5	=	5
2	x	5	=	10
3	x	5	=	15
4	x	5	=	20
5	x	5	=	25
6	x	5	=	30
7	x	5	=	35
8	x	5	=	40
9	x	5	=	45
10	x	5	=	50

4. Una empresa que desarrolla aplicaciones móviles nos pide que desarrollemos una página web que permita definir el aumento salarial que va a recibir cada uno de sus trabajadores y el resultado presentarlo en una página web.

Para definir el porcentaje de aumento se tiene en cuenta la antigüedad en la empresa y el salario actual que percibe. Para realizar los cálculos se tienen los siguientes datos:

Antigüedad en años	Porcentaje
de 0 a 5	20
de 6 a 10	18
de 11 a 15	15
más de 15	12



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Salarios	Determinación del porcentaje de aumento.
De 100000 hasta 200000	Reciben el porcentaje completo.
De 200000,01 hasta 300000	Reciben el 50 % del porcentaje de aumento.
Mayor a 300000,01	Reciben el 10 % del porcentaje de aumento.

Listado de Empleados

Empleado	Salario	Antigüedad
J. Perez	150000	5
A. Fernandez	180000	8
L. Alonso	155000	6
M. Peralta	195000	7
A. Campana	215000	10
S. Ferrera	220000	18
I. Garcia	250000	25
D. Miranda	275000	4
G. Benitez	320000	6
A. Hernandez	230000	12

5. Diseñar un formulario donde el usuario pueda calcular la cantidad de bolsas que tendrá que comprar, para alimentar a su mascota durante un mes. Para esto tiene que:
- Seleccionar la raza de su mascota (Fox Terrier, Labrador, Caniche, Chiguagua, etc.).
 - Ingresar la cantidad de comida que debe comer por día (en gramos).
 - El tipo de bolsa disponible (1/2 Kilogramos, 1 Kilogramo, 3 Kilogramos) que desea comprar.

Calcular las bolsas necesarias para alimentar su mascota un mes

Mascota	<input type="text" value="Fox Terrier"/>
Cantidad Alimento	<input type="text" value="100"/> Gramos
Tipo de bolsa	<input type="text" value="1 Kilogramo"/>
Total de Bolsas	Debe comprar 3 bolsas de alimentos.
<input type="button" value="Calcular"/>	

6. Diseñar un formulario de suscripción a eventos deportivos, que le solicite al usuario sus datos personales, (Nombre, Apellidos, Sexo, Fecha de Nacimiento, Nro. Teléfono, E-mail) y los eventos



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

(tipo de evento, fecha de realización, lugar). Cuando el usuario haga envío de los datos, se mostrará un mensaje con la información ingresada y el resultado de la suscripción. En caso de que falten datos, solicitar al usuario que los complete, antes de realizar del envío del formulario.

Tener en cuenta:

- a. Para que se acepte la suscripción, el usuario tiene que ser mayor de edad.
- b. La suscripción tiene que tener al menos un evento.
- c. Tipos de evento: torneo de tenis, campeonato de ajedrez, competencias escolares, etc.

7. Un centro de llamadas le solicita un programa para que calcule el importe de las llamadas telefónicas que realizan los clientes. El costo de las llamadas se calcula con: el importe por minuto, correspondiente al lugar geográfico donde se llame, por la cantidad de minutos que dure la llamada. Si la llamada dura menos de 3 minutos entonces se cobrará el mínimo de \$ 45.50. Se pide diseñar un formulario para:

- a. Seleccionar el destino de la llamada.
- b. Mostrar el importe por minuto respecto del destino.
- c. Introducir la cantidad de minutos que duró la llamada.
- d. Mostrar en pantalla el total a pagar.

8. Un propietario de una pizzería, le solicita una página web donde los clientes puedan ingresar y hacer pedidos de pizza. El cliente quiere que junto con la descripción de la pizza se muestre una imagen y el precio. Los clientes de la pizzería, tienen que poder ingresar a la página, seleccionar las pizzas y poner la cantidad de cada tipo; con esta información calcular el total a pagar. Cuando el cliente acepte comprar, se le deben solicitar los datos de entrega. Si lo retira en el establecimiento, se le informará la hora que puede pasar a retirar el pedido. Si la entrega es en el domicilio, tendrá que completar un formulario con la dirección de entrega y un teléfono.

9. Le solicitan diseñar una página web que le permita a un usuario ingresar una solicitud de crédito personal. El usuario tiene que ingresar sus datos personales (Apellido, Nombre, DNI, fecha de nacimiento, domicilio), datos laborales (Empresa, domicilio de la empresa, salario que percibe, fecha de ingreso) y datos del crédito (importe que solicita).



LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

Cuando el usuario presione el botón enviar, se tiene que validar que el solicitante sea mayor de edad y la antigüedad en el empleo sea mayor a dos años. Se calcula, en base al monto solicitado:

la cantidad mínima y la cantidad máxima de cuotas posibles y para cada cuota el importe a pagar.

Tener en cuenta que:

- a. La cantidad máxima de cuotas que puede tener un préstamo es de sesenta (60) cuotas.
- b. El importe mínimo de una cuota es de \$ 550,00.
- c. La cuota que se calcule debe ser como máximo el 30 % del salario.

Presentar en pantalla el resultado de la consulta; en caso de que no se pueda aprobar la solicitud, informar al usuario la causa.

10. Desarrollar un sitio web, cuyo index.php despliegue información de tres secciones distintas. Cada sección incorporará un contador de visitas, para registrar la cantidad de veces que se accede a cada sección y, según el día de la semana, deberá mostrar una imagen diferente relacionada a la sección.
11. Desarrolle un script PHP que contabilice la cantidad de accesos de un usuario a la página. Para esto, diseñe una página web de inicio que contenga un formulario donde el usuario puede ingresar su nombre. Luego, diseñe una segunda página que contendrá el script antes mencionado, el cual recibirá el nombre del usuario y mostrará la cantidad de visitas que dicho usuario ha efectuado al sitio web.
12. Desarrolle una página web dinámica de servidor autoprosesada que permita a un usuario ingresar una serie de palabras (una a una, en distintos tiempos, como en el ejemplo de la diapositiva 33 de la teoría de la unidad 4), y las vaya procesando de manera tal que, a medida que se ingresa cada una, y cuando el usuario pulse un botón de envío al servidor web, se muestre la cantidad de palabras ingresadas, la lista completa de palabras ingresadas hasta ese momento junto a la cantidad de caracteres de cada palabra.

LABORATORIO DE PROGRAMACION Y LENGUAJES

Trabajo Practico Nro. 4. Tecnología de Desarrollo Web del Lado del Servidor

13. Diseñar una grilla que tenga diez columnas y diez filas, donde las filas pares correspondan a números pares y las filas impares correspondan a números impares. Además, cada fila par y cada fila impar debe tener un color. El link le mostrará al usuario un mensaje informando el número que seleccione. Funcionalidad: el usuario ingresa su nombre y tiene que desplegarse un mensaje: **“Hola Pepe elegiste el Número x”**, en la misma sección pero ocultando la grilla. Agregar un botón para desplegar nuevamente la grilla de número.

Nombre:

1	3	5	7	9	11	13	15	17	19
2	4	6	8	10	12	14	16	18	20
21	23	25	27	29	31	33	35	37	39
22	24	26	28	30	32	34	36	38	40
41	43	45	47	49	51	53	55	57	59
42	44	46	48	50	52	54	56	58	60
61	63	65	67	69	71	73	75	77	79
62	64	66	68	70	72	74	76	78	80
81	83	85	87	89	91	93	95	97	99
82	84	86	88	90	92	94	96	98	100