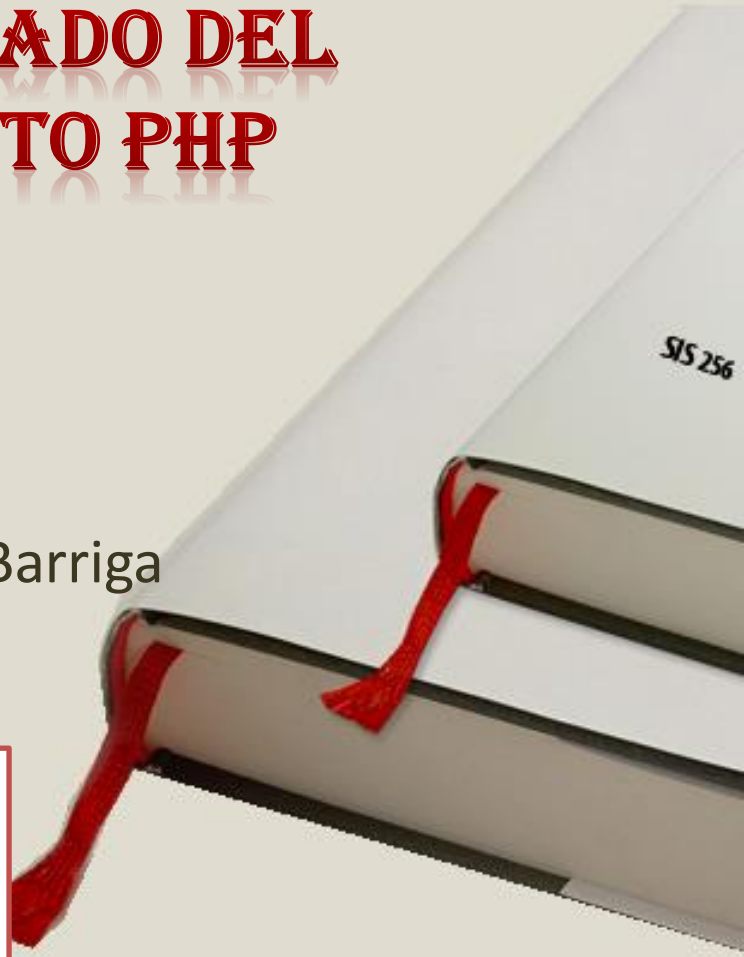
TEMA 3
PROGRAMACION DE LADO DEL SERVIDOR UTILIZANTO PHP

Ing. Carlos David Montellano Barriga



Hypertext Pre-processor

Originalmente se conocía como Personal Home Page

CONTENIDO

- Procesamiento del Lado del Servidor
- ¿Que es PHP?
- Ide’s para PHP
- Sintaxis del Lenguaje PHP
- Manejo de Cadenas
- Funciones y Librerías
- Envió de datos mediante GET,POST
- Programación Orientada a Objetos con PHP
- Sesiones y Cookies
- Acceso a Base de Datos
- Manejo de Archivos
- Seguridad
- Envió de Correos
- Gráficos en PHP
- Frameworks en php

PROCESAMIENTO DEL LADO DEL SERVIDOR

Tecleamos una dirección **URL** en el navegador www.usfx.bo

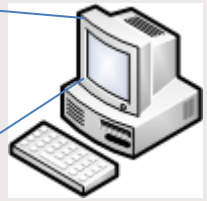
Cliente envía una petición al Servidor Web utilizando **http**

HTTP

Servidor deriva solicitud

Servidor procesa Solicitud y genera dinámicamente archivo en html

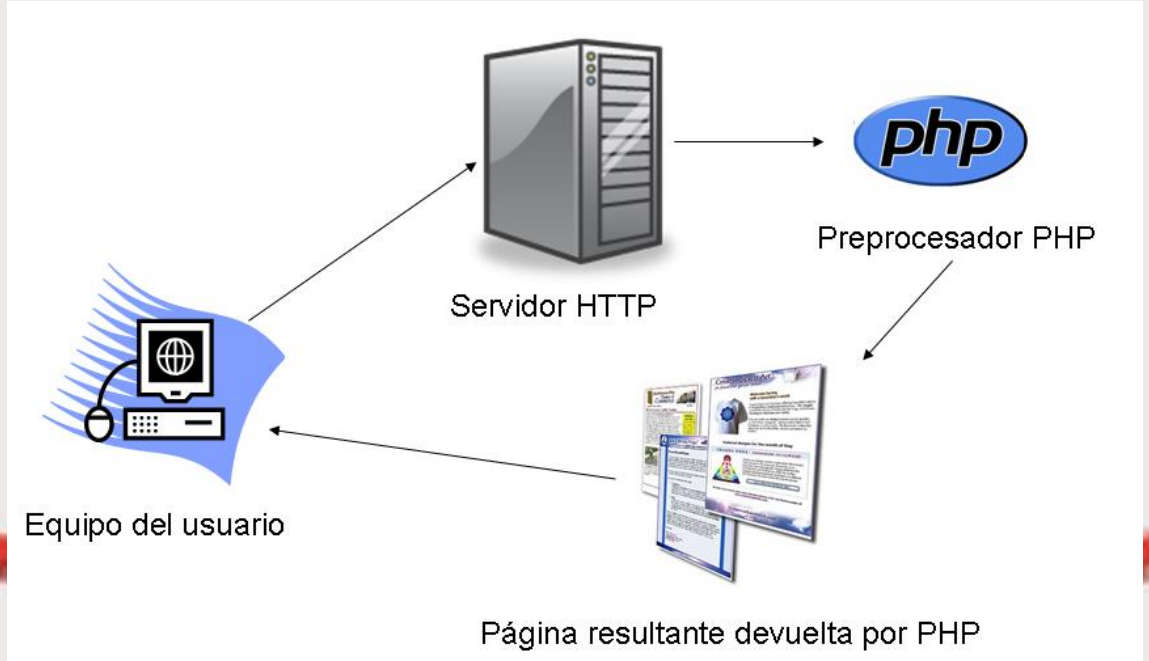
Servidor devuelve en lenguaje **html** solicitada utilizando **http**



El navegador se encarga de interpretar el código **html** y mostrar el resultado

¿QUE ES PHP?

- Lenguaje de programación interpretado
- Desarrollado para funcionar en el Web y puede ser incrustado dentro de código HTML.
- Ingresa Código PHP como su entrada y creando páginas Web como salida.



HOLA MUNDO

hola.php

```
<html>
<head>
   <title>hola mundo</title>
</head>
<body>

<?php
echo "hola mundo!";
?>

</body>
</html>
```

Ventajas

- Lenguaje Multiplataforma
- Capacidad de conexión con mayoría manejadores de base de datos. Destaca su conectividad con MySQL.
- Capacidad de expandir su potencial (llamados ext's o extensiones).
- Fácil de Aprender, existe amplia documentación
  - funciones del sistema están explicadas en su sitio oficial www.php.org
- Rápida Ejecución
- Servidores PHP son estables, fáciles de mantener y mas baratos.
- PHP tiene una infinidad de librerías y frameworks y todas son gratuitas.
- PHP es más sencillo de aprender.

Características de PHP

Ventajas

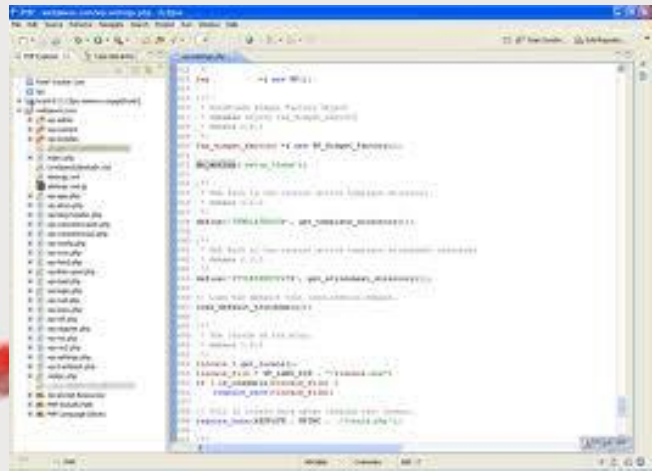
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.

PHP DESVENTAJAS)

- Php es un leguaje interpretado (el servidor interpreta el código cada vez que lo va a utilizar) lo cual afecta a su rendimiento.
- Php depende de que su comunidad reaccione de alguna u otra manera ante los reportes de bugs.
- **Velocidad de desarrollo**:. PHP es rápido si se usa algún framework.
- La legibilidad del código puede verse afectada.

IDES PARA PHP

- IDE's(Integrated Development Environment) = Entornos de Desarrollo Integrados
- PDT, plugin Eclipse: GPL - (Sun).
- NetBeans: (instalar plugin para php) libre, para linux y windows.
- Zend Studio: Comercial - (Zend).
- Komodo IDE: Komodo Edit, libre y gratuito, el IDE es licencia comercial - (Mozilla).
- NuSphere PhpED: Comercial, para linux y windows.
- Quanta: GPL y gratuito, para GNU/linux con QT.
- Bluefish: GPL y gratuito, para GNU/linux con GTK.
- phpDesigner: Comercial y Freeware, para linux y windows.
- Rapid PHP: Comercial, para windows.
- Dream weber: Adobe



APLICACIONES DESARROLLADAS CON PHP

- Dokuwiki
- Drupal
- Facebook
- Joomla
- MediaWiki
- Moodle
- Phorum
- phpMyAdmin
- PHP-Nuke
- phpPgAdmin
- PhpWiki
- PmWiki
- Zikula (anteriormente llamado PostNuke)
- Smarty
- SPIP
- SugarCRM
- vBulletin
- WordPress
- Xaraya
- Xoops
- Joomla
- MODx
- SMF

SINTAXIS PHP

primer.php

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>

<body>

Parte de HTML normal.
<BR><BR>

<?php
  echo "Parte de PHP<br>";

  for($i=0;$i<10;$i++)
  {
    echo "Linea ".$i."<br>";
  }
?>

</body>
</html>
```

Etiqueta para codigo PHP

Variables

- Todas las variables comienzan con el símbolo del dólar $
- No es necesario definir una variable antes de usarla.
- Tampoco tienen tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres.

variables.php

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<?php
  $a = 1;
  $b = 3.34;
  $c = "Hola Mundo";
  echo $a,"<br>",$b,"<br>",$c;
?>
</body>
</html>
```

COMENTARIOS

- // Comentar texto
- /* Comentar texto de muchas lineas */
- # Comentar texto

Operadores Aritméticos

Operador	Nombre	Ejemplo	Descripción
+	Suma	5 + 6	Suma dos números
-	Resta	7 - 9	Resta dos números
*	Multiplicación	6 * 3	Multiplica dos números
/	División	4 / 8	Divide dos números
%	Módulo	7 % 2	Devuelve el resto de dividir ambos números, en este ejemplo el resultado es 1
++	Suma 1	$a++	Suma 1 al contenido de una variable.
--	Resta 1	$a--	Resta 1 al contenido de una variable.

Operadores Relacionales

Operador	Nombre	Ejemplo	Devuelve cierto cuando:
==	Igual	$a == $b	$a es igual $b
!=	Distinto	$a != $b	$a es distinto $b
<	Menor que	$a < $b	$a es menor que $b
>	Mayor que	$a > $b	$a es mayor que $b
<=	Menor o igual	$a <= $b	$a es menor o igual que $b
>=	Mayor o igual	$a >= $b	$a es mayor o igual que $b

Operadores Lógicos

Operador	Nombre	Ejemplo	Devuelve cierto cuando:
&&	Y	(7>2) && (2<4)	Devuelve verdadero cuando ambas condiciones son verdaderas.
and	Y	(7>2) and (2<4)	Devuelve verdadero cuando ambas condiciones son verdaderas.
||	O	(7>2) || (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera.
or	O	(7>2) or (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera.
!	No	! (7>2)	Niega el valor de la expresión.

Operadores de Asignación

Operador	Descripción
=	Asigna
+=	Adiciona y asigna
- =	Resta y asigna
* =	Multiplica y asigna
/ =	Divide y asigna
%=	Obtiene el resto y asigna
. =	Concatena y asigna

CONDICIONALES

if (<condicion 1>)
{ <declaracion 1>; }
Else
{ <declaracion 2>; }

mayor.php

```
<html>
    <head>
      <title>Ejemplo de PHP</title>
    </head>
    <body>
    <?php
      $a = 8;
      $b = 3;
      if ($a < $b)
      {
        echo "a es menor que b";
      }
      else
      {
        echo "a no es menor que b";
      }
    ?>
    </body>
</html>
```

switch (<expresion>)
{
case <literal o tipo>: < declaraciones >;
[break;]
case <literal o tipo>: < declaraciones >;
[break;]
default: < declaraciones >;
}

casos.php

```
<html>
    <head>
      <title>Ejemplo de PHP</title>
    </head>
    <body>
    <?php
      $posicion = "arriba";
     switch($posicion) {
        case "arriba":   // Bloque 1
          echo "La variable contiene";
          echo " el valor arriba";
          break;
        case "abajo":   // Bloque 2
          echo "La variable contiene";
          echo " el valor abajo";
          break;
        default:   // Bloque 3
          echo "La variable contiene otro valor";
          echo " distinto de arriba y abajo";
      }
    ?>
    </body>
</html>
```

BUCLES

for (<inicializacion>;<condicion>;<incremento>)
{
  <declaraciones>;
}

while (<condicion>)
{
  < declaraciones >;
}

numeros.php

```
<html>
    <head>
      <title>Ejemplo de PHP</title>
    </head>
    <body>
    Inicio<BR>
    <?php
      for($i=0 ; $i<10 ; $i++)
      {
        echo "El valor de i es ", $i,"<br>";
      }
    ?>
    Final<BR>
    </body>
</html>
```

mientras.php

```
<html>
    <head>
      <title>Ejemplo de PHP</title>
    </head>
    <body>
    Inicio<BR>
    <?php
      $i=0;
      while ($i<10)
      {
        echo "El valor de i es ", $i,"<br>";
        $i++;
      }
    ?>
    Final<BR>
    </body>
</html>
```

ARREGLOS

- $nombrematriz **= array();**
- Es posible asignar sus valores

```
$vocales = array ('a','e','i','o','u');
```

- Luego para acceder

```
$vocales[2]='i';
echo $vocales[1];
echo $vocales[3];
```

ARREGLOS EN PHP – ARRAYS()

- Arrays numéricos – el índice es numérico, de 0 en adelante
- Ejemplo de un array en cadena

$ciencias = array(“Física”,”Química”,”Biología”);

- Ejemplo de un array por índice

```
$ciencias[0] = “Física”;
$ciencias[1] = ”Química”;
$ciencias[2] = ”Biología”;
```

ARREGLOS EN PHP – ARRAYS()

- Ejemplo de un array uno tras otro

```
$ciencias[ ] = “Física”;
$ciencias[ ] = ”Química”;
$ciencias[ ] = ”Biología”;
```

- Para consultar un valor

```
print  $ciencias[0];
```

ARREGLO MULTIDIMENSIONAL

A | B | C | D | E | F
G | H | I | J | K | L

$array[0][0] = A
$array[0] = array que contiene los valores A | B | C | D | E | F
$array[1] = array que contiene los valores G | H | I | J | K | L
$array[0][1] = B,
$array[0][2] = C,
$array[0][3] = D

rellenar.php

```
$columnas = 7;
$filas = 3;
for($i=0; $i < $columnas; $i++)
{
        for($j=0; $j < $filas; $j++)
        {
                $matriz[$i][$j] = $i * $j;
        }
 }
```

```
foreach (<array> as [<valor> |<key> => <valor>])
{
    <declaraciones>;
    [break];
    [continue];
}
```

vocales.php

```
<?
    /* Creo un array */
    $las_vocales = array ('a','e','i','o','u');

    /* Recorro el array utilizando foreach */
    foreach ($las_vocales as $elem) {
    echo $elem.'<br>';
    }

    print '<br>';

    /* Es equivalente a realizar */
    for ($i=0;$i<count($las_vocales);$i++) {
    echo $las_vocales[$i].'<br>';
    }

    ?>
```

vocales2.php

```
<?
    /* Creo un array */
    $las_vocales = array ('a','e','i','o','u');

    /* Recorro el array utilizando foreach */
    foreach ($las_vocales as $indice=> $elem)
    {
    echo $elem.' en el indice: '.$indice.'<br>';
    }

    ?>
```

SALIDA EN PANTALLA

Hasta ahora hemos usado la instrucción echo para realizar salida a pantalla, esta instrucción es bastante limitada ya que no nos permite formatear la salida. En esta página veremos la instrucción printf que nos da mucha más potencia

salidas.php

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<?php
  $var="texto";
  $num=3;
  printf("Puede fácimente intercalar <b>%s</b> con números <b>%d</b> <br>",$var,$num);

  printf("<TABLE BORDER=1 CELLPADDING=20>");
  for ($i=0;$i<10;$i++)
  {
    printf("<tr><td>%10.d</td></tr>",$i);
  }
  printf("</table>");
?>
</body>
</html>
```

MANEJO DE CADENAS

- **strlen**(cadena). Nos devuelve el número de carácteres de una cadena.
- **explode**(separador,cadena). Divide una cadena en varias usando un carácter separador.
- **substr**(cadena, inicio, longitud). Devuelve una subcadena de otra, empezando por inicio y de longitud longitud.
- **chop**(cadena). Elimina los saltos de línea y los espacios finales de una cadena.
- **strpos**(cadena1, cadena2). Busca la cadena2 dentro de cadena1 indicándonos la posición en la que se encuentra.
- **str_replace**(cadena1, cadena2, texto). Reemplaza la cadena1 por la cadena2 en el texto.

cadenas.php

```
<html>
    <head>
      <title>Ejemplo de PHP</title>
    </head>
    <body>
    <?php
      echo strlen("12345"),"<br>";

      $palabras=explode(" ","Esto es una prueba");
      for($i=0;$palabras[$i];$i++)
        echo $palabras[$i],"<br>";

      $resultado=sprintf("8x5 = %d <br>",8*5);
      echo $resultado,"<br>";

      echo substr("Devuelve una subcadena de otra",9,3),"<br><br>";

      if (chop("Cadena \n\n ") == "Cadena")
        echo "Iguales<br><br>";

      echo strpos("Busca la palabra dentro de la frase", "palabra"),"<br><br>";

      echo str_replace("verde","rojo","Un pez de color verde, como verde es la hierba."),"<br>";

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

FUNCIONES

```
function Nomre(parametro1, parametro2...)
    {
      instrucción1;
      instrucción2;
      …
      return valor_de_retorno;
    }
```

media.php

```
<html>
      <head>
        <title>Ejemplo de PHP</title>
      </head>
      <body>
      <?php

        function media_aritmetica($a, $b)
        {
          $media=($a+$b)/2;
          return $media;
        }
           echo media_aritmetica(4,6),"<br>";
           echo media_aritmetica(3242,524543),"<br>";
         ?>
      </body>
</html>
```

LIBRERIAS

Permiten agrupar varias funciones y variables en un mismo fichero

Podemos incluir librerías en distintas páginas y disponer de esas funciones fácilmente

libreria.php

```
<?php
  function CabeceraPagina()
  {
?>
  <FONT SIZE="+1">Esta cabecera estará en todas
sus páginas.</FONT><BR>
  <hr>
<?
  }

  function PiePagina()
  {
?>
  <hr>
  <FONT SIZE="-1">Este es el pie de
página.</FONT><BR>
  Autor: Carlos Montellano
<?
  }
?>
```

pagina.php

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<?php include("libreria.php") ?>
<?php CabeceraPagina(); ?>

Página 1
<BR><BR><BR><BR><BR>

Contenido blalbl blalb alb<BR><BR>
más cosas...<BR><BR>

fin<BR><BR>

<?php PiePagina(); ?>
</body>
</html>
```

ENVÍO Y RECEPCIÓN DE DATOS

formulario.html

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de formularios</H1>
Introduzca su nombre:
<FORM action= "recibir.php" method="GET">
<INPUT type="text" name="nombre"><BR>
<INPUT type="submit" value="Enviar">
</FORM>
</body>
</html>
```

recibir.php

```
<html>
<head>
  <title>Recibir PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de
formularios</H1>
El nombre que ha introducido es: <?php echo
$_GET['nombre'] ?>
<br>
</body>
</html>
```

PROGRAMACION ORIENTADA A OBJETOS

Estructura de una clase

class <nombre_clase> [<extends clase_base>]

{

[var | <modificadores*>] [<variables miembro de la clase>];

[<modificadores*>] **function** <funcion_nombre>([<parametros>])

{

< declaraciones >;

}

}

*Modificadores <public | private | protected> están implementados en PHP5.

- Acceso a propiedades y métodos dentro de la misma clase:

$this->nombrepropiedad

$this->nombre metodo

persona.php

```
<?php
class persona
{
  var $nombre;
  function set_nombre($nuevo_nombre) {
  $this->nombre = $nuevo_nombre;
   }
   function get_nombre() {
  return $this->nombre;
  }
 }?>
```

- **Declarar y usar clases**

$variable = new nombre_clase ();

$variable -> funcion_nombre ();

nombre_clase : : funcion_nombre(); **(Llamada estática).**

usarpersona.php

```
<?php include("persona.php");
$juan = new persona();
$ricardo = new persona;
$ juan ->set_nombre("Juan Perez");
$ ricardo ->set_nombre("Ricardo Martinez");
echo "Nombre completo de Juan: " . $juan->get_nombre();
echo "Nombre completo de Ricardo: " . $ricardo->get_nombre();
?>
```

CONSTRUCTORES

- Método que se ejecuta al instanciar la clase, para iniciar propiedades

```
<?php
class persona {
var $nombre;
function __construct($nombrepersona) {
$this->nombre = $nombrepersona;
}function set_nombre($nuevonombre) {
$this->nombre = $nuevonombre;
}
…
```

```
<?php include("persona.php"); ?>
</head>
<body>
<?php
$carlos = new persona("carlos
montellano ");
echo "Nombre Completo: " . $carlos-
>get_nombre();
?>
</body>
</html>
```

MODIFICADORES DE ACCESO".

- public
  - modificador por defecto. Significa que si no lo colocas se asume que es Publico.
- private
  - solo la misma clase puede acceder a la propiedad.
- *protected*
  - solo la misma clase y las clases derivadas de esta clase pueden acceder a la propiedad

```
<?php
class person {
var $name;
public $height;
protected $social_insurance;
private $pinn_number;
function __construct($persons_name) {
$this->name = $persons_name;
}
private function get
return $this->$pinn_number;
}
?>
```

HERENCIA

- *// 'extends' es la palabra clave que permite la herencia*

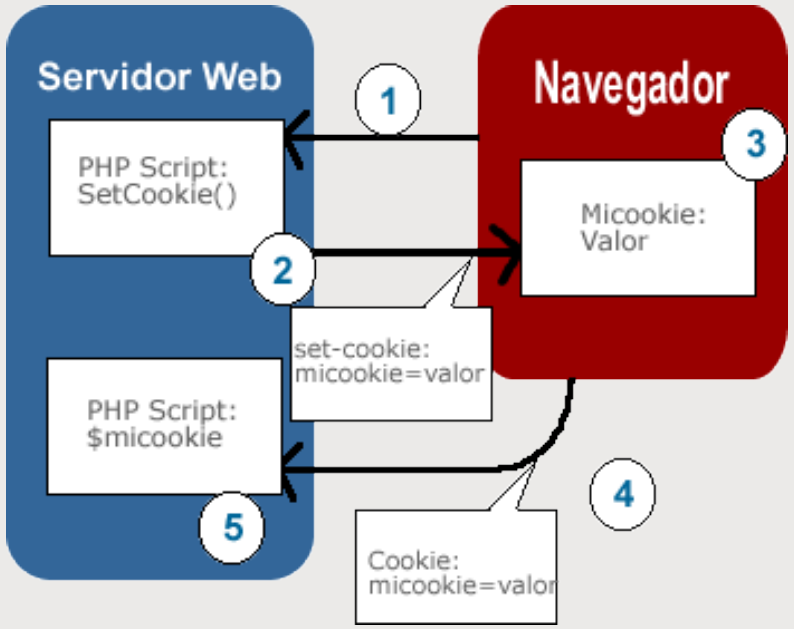
```
class empleado extends persona {
function __construct($nombre_empleado) {
$this->set_name($nombre_empoleado);
}}
```

Cookies y Sesiones

- HTTP sin estado
  - Conexiones por medio de HTTP no mantienen **un estado**.
  - No se pueden mantener variables en la conexión.
  - Se han inventado mecanismos para tener una especie de estado,
    - “cookies”, que permiten guardar un estado del lado del cliente
    - las sesiones, que permiten guardar un estado del lado del servidor.

COOKIES

Servidor cuando regresa un objeto HTTP al cliente puede enviar también un **paquete de información de estado** que el cliente va a almacenar de manera persistente.



Incluido con ese estado se encuentra un rango de URLs para los cuales ese estado es válido.

Cualquier solicitud HTTP futura hecha por el cliente que caiga dentro de tal rango va a incluir una transmisión del valor actual del objeto desde el cliente al servidor.

COOKIES EN PHP

- Función setcookie define una cookie para ser enviada con la información de encabezado.
- Debe ser enviada antes de cualquier otra información de encabezado.
- Sintaxis:
  - int **setcookie** (string name, string value, int expire, string path, string domain, int secure)

contadorcookie.php

```
<?php
if (isset( $_COOKIE['contador'] ) ) {
        $valor=$_COOKIE['contador']+1;
        echo "es la visita $valor";
        setcookie("contador",$valor,0); /* expira en 1 hora */
        }
else
        {
   echo "es la visita 1";
   setcookie("contador",1,0); /* expira en 1 hora */
        }
?>
```

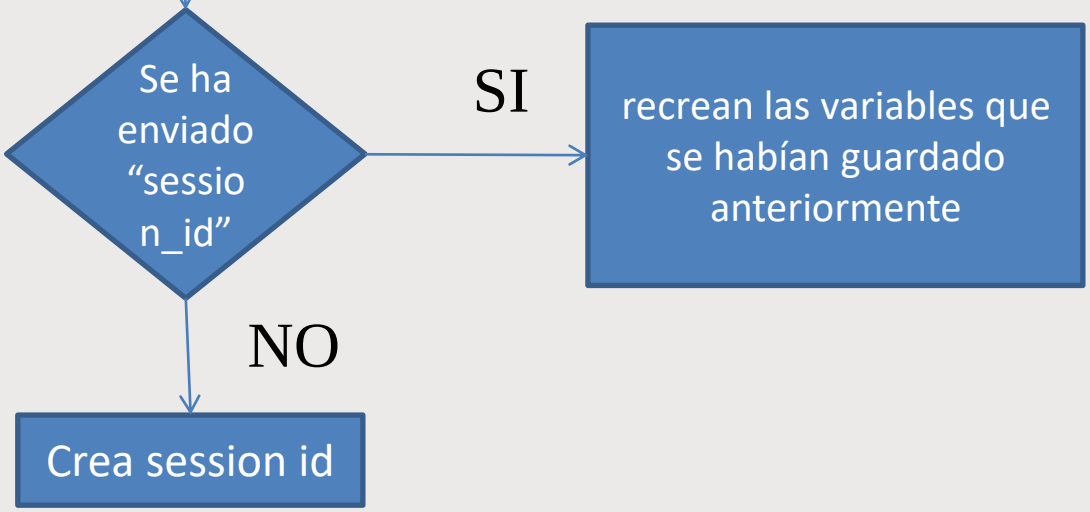
SESIONES EN PHP

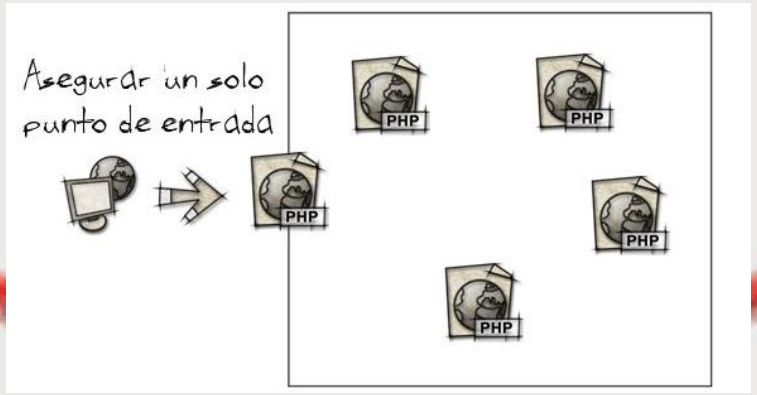
- PHP tiene apoyo para conservar estado con sesiones.
- Sesiones mantienen variables en el lado del servidor.
- A cada visitante que accede a la página se le asigna un identificador único, llamado "session id" (identificador de sesión).
- Éste se almacena en una cookie por parte del usuario o se propaga en la URL (método GET).
- Permite registrar un número arbitrario de variables que se conservarán en las siguientes solicitudes.

SESIONES EN PHP

pagina.php

<?php session_start();?>





SESIONES EN PHP

sesion.php

```
<?php session_start();
    if (isset($_SESSION['contador'])) {
      $_SESSION['contador']++;
    }
    else {
      $_SESSION['contador'] = 0;
    }
    ?>
```

SESIONES EN PHP

- borrar una variable con $_SESSION:

borrarsesion.php

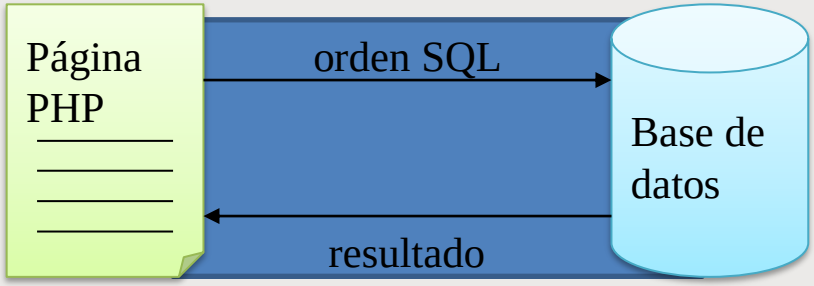
```
<?php
    session_start();
    unset($_SESSION['contador'
]);
    session_destroy();
    ?>
```

ACCESO A BASES DE DATOS

- PHP permite el acceso a las siguientes bases de datos:
  - dbase
  - dbm
  - db++
  - FrontBase
  - filePro
  - Informix
  - InterBase
  - Ingres II
  - SQL Server
  - mSQL
  - Mysql
  - Oracle
  - Ovrimos SQL Server
  - PostgreSQL
  - SESAM
  - Sybase

ACCESO A BASES DE DATOS

- PHP también tiene interfaces abstractas:
  - DBA (DataBase Abstraction)
  - DBX
  - ODBC



MYSQL

- MySQL es un sistema de bases de datos desarrollado originalmente por MySQL AB. Actualmente la control a ORACLE
- El sistema de bases de datos se da bajo licencia GPL que es una licencia de software libre o se vende bajo una licencia comercial.

MYSQL

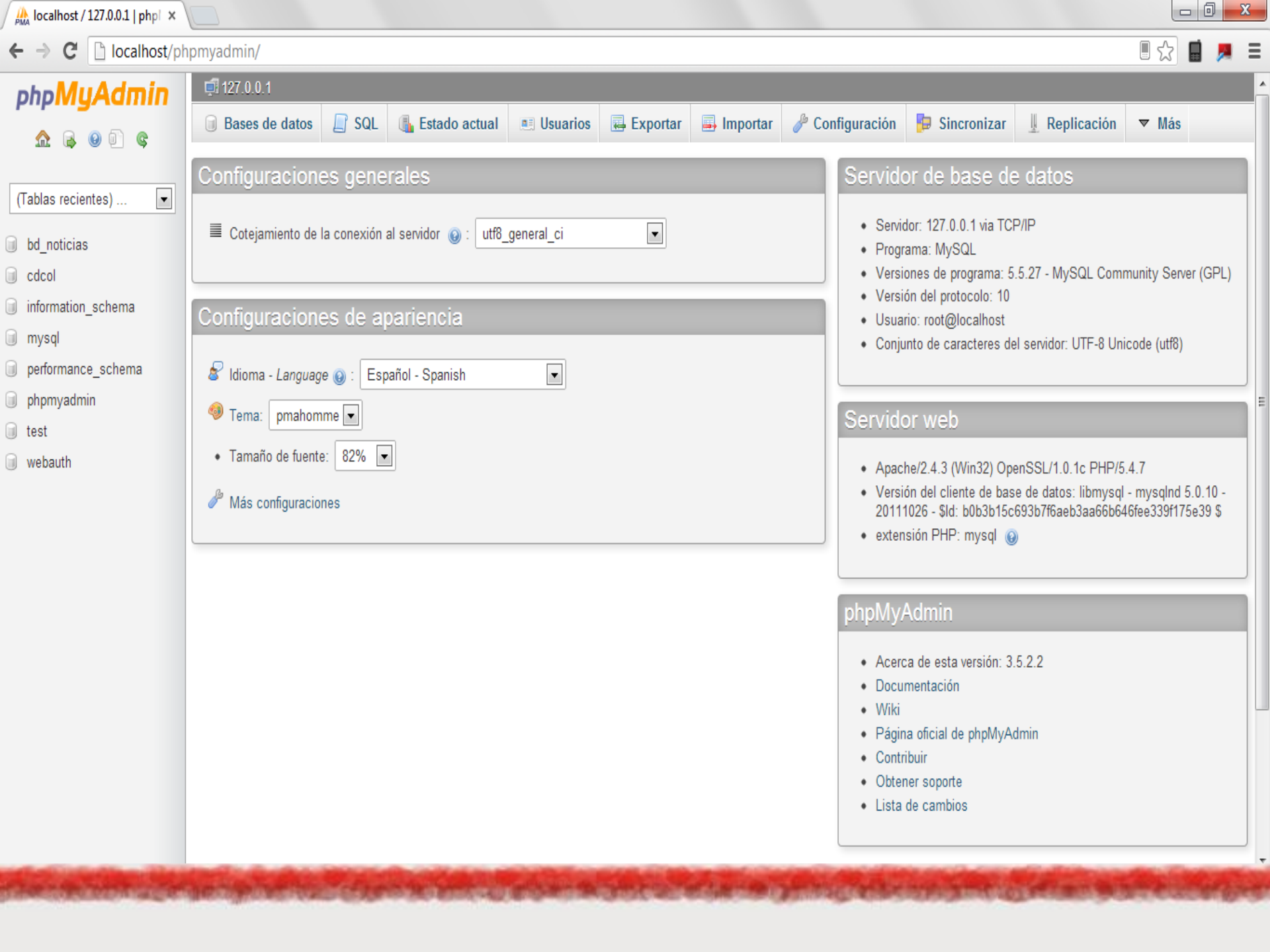
- **Características de MySQL**
  - Modelo relacional, multiusuario
- **Tipos de datos**
  - Numéricos
    - tinyint, smallint, mediumint, int, integer, bigint
    - decimal, float, numeric
  - Fecha y hora
    - date, time, datetime, year, timestamp
  - Cadena
    - char, varchar
    - tinytext, text, mediumtext, longtext
    - tinyblob, blob, mediumblob, longblob
    - enum, set
  - Debe elegirse adecuadamente el tipo y el tamaño de cada campo

MYSQL

- **Operadores**
  - Aritméticos
    - +, -, *, /
  - Comparación
    - =, !=, <=, <, >=, >, IS NULL, IS NOT NULL
  - Lógicos
    - not (!), and (&&), or (||), xor
- **Funciones**
  - Funciones de cadena
  - Funciones de comparación de cadenas
  - Funciones numéricas
  - Funciones de fecha y hora
  - Funciones de agregado

HERRAMIENTAS DE ADMINISTRACIÓN: PHPMYADMIN

- **phpMyAdmin** es una herramienta para la administración del servidor de bases de datos MySQL
- Dispone de una interfaz gráfica y es de libre distribución
- Permite realizar todo tipo de operaciones sobre bases de datos:
  - crear, borrar y modificar tablas
  - consultar, insertar, modificar y eliminar datos
  - definir usuarios y asignar permisos
  - realizar copias de seguridad
  - etc
- Está escrita en php y se ejecuta desde el navegador
- Puede administrar bases de datos locales y remotas


localhost / 127.0.0.1 | php
localhost/phpmyadmin/
phpMyAdmin
(Tablas recientes) ...
bd_noticias
cdcol
information_schema
mysql
performance_schema
phpmyadmin
test
webauth
127.0.0.1
Bases de datos
SQL
Estado actual
Usuarios
Exportar
Importar
Configuración
Sincronizar
Replicación
Más
Configuraciones generales
Cotejamiento de la conexión al servidor :
utf8_general_ci
Configuraciones de apariencia
Idioma - Language :
Español - Spanish
Tema:
pmahomme
Tamaño de fuente:
82%
Más configuraciones
Servidor de base de datos
Servidor: 127.0.0.1 via TCP/IP
Programa: MySQL
Versiones de programa: 5.5.27 - MySQL Community Server (GPL)
Versión del protocolo: 10
Usuario: root@localhost
Conjunto de caracteres del servidor: UTF-8 Unicode (utf8)
Servidor web
Apache/2.4.3 (Win32) OpenSSL/1.0.1c PHP/5.4.7
Versión del cliente de base de datos: libmysql - mysqlnd 5.0.10 - 20111026 - $Id: b0b3b15c693b7f6aeb3aa66b646fee339f175e39 $
extensión PHP: mysql
phpMyAdmin
Acerca de esta versión: 3.5.2.2
Documentación
Wiki
Página oficial de phpMyAdmin
Contribuir
Obtener soporte
Lista de cambios

FUNCIONES PHP PAR MYSQL

- Las funciones concretas de MySQL que realizan estas operaciones son:
  - Conectar con el servidor de bases de datos:
    - **mysqli_connect()**
  - Enviar la instrucción SQL a la base de datos:
    - **mysqli_query()**
  - Obtener y procesar los resultados:
    - **mysqli_num_rows() y mysqli_fetch_array()**
  - Cerrar la conexión con el servidor de bases de datos:
    - **mysqli_close()**

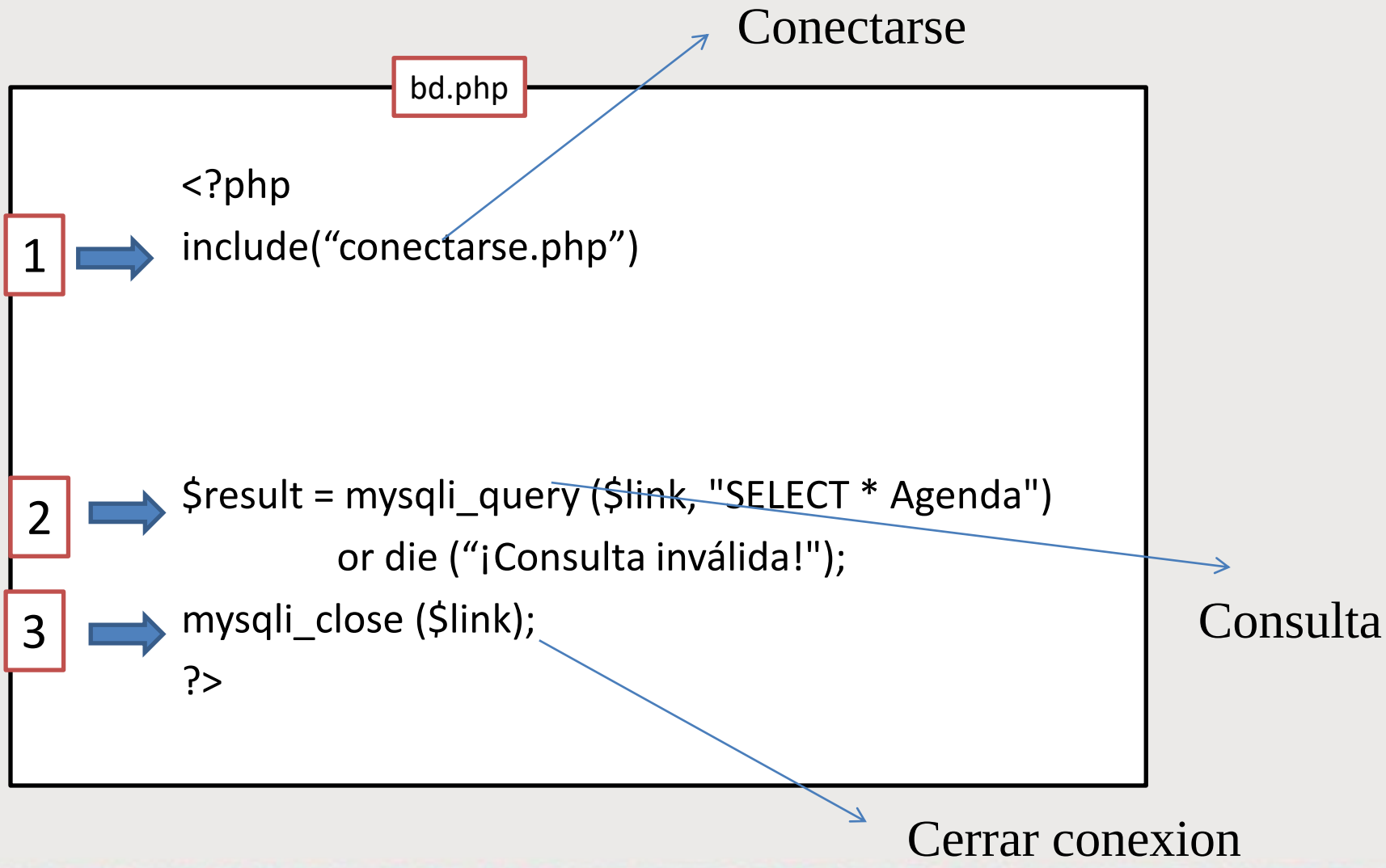
CONEXIÓN A LA BASE DE DATOS

servidor usuario password

conexion.php

```
<?php
$con =new mysqli("localhost", "root", "","bd_prueba");
if ($con->connect_error)
 die ("conexion fallada".$con->connect_error);

?>
```

FUNCIONAMIENTO ACCESO A DATOS



LISTAR DATOS

listar.php

```
<?php
 include('conexion.php');
 $sql = "select ID,Nombres,Direccion,Telefono,Email,Celular,IdProfesion from agenda ";
 $consulta = mysqli_query($link, $sql);
?>
<table width="100%" border="1">
  <tr>
    <td>Nombres</td>
    <td>Direccion</td>
    <td>Telefono</td>
    <td>Email</td>
    <td>Celular</td>
  </tr>
 <?php
while ($fila = mysqli_fetch_array($consulta)) {
?>
  <tr>
    <td><?php echo $fila['Nombres']?></td>
    <td><?php echo $fila['Direccion']?></td>
    <td><?php echo $fila['Telefono']?></td>
    <td><?php echo $fila['Email']?></td>
    <td><?php echo $fila['Celular']?></td>
  </tr>
  <?php
  }
  ?>
</table>
```

INSERTAR

finsertar.html

```
<form id="form1" name="form1" method="post"
action='javascript:insertarPersona()'>
  Nombres
    <input name="txtNombres" type="text" id="txtNombres"  />
    <br />
   Dirección   <input type="text" name="txtDireccion" id="txtDireccion"
/><br />
   Telefono <input type="text" name="txtTelefono" id="txtTelefono"
/><br />
   Email <input type="text" name="txtEmail" id="txtEmail"   /><br />
   Celular <input type="text" name="txtCelular" id="txtCelular" />
      <br />
  <input type="submit" name="button" id="button" value="Enviar" />
  <input type="reset" name="button2" id="button2" value="Limpiar"
/>
</form>
<?php
 ?>
```

insertar.php

```
<?php
   include('conexion.php');
                $sql = "insert into agenda (
                Nombres,Direccion,Telefono,Email,Celular)
values

('$_POST[txtNombres]','$_POST[txtDireccion]','$_POST[txtTe
lefono]','$_POST[txtEmail]','$_POST[txtCelular]')";
                //echo $sql;
                $consulta = mysql_query($sql,$link);
                if (isset($consulta))
                { echo "Elemento insertado con exito";
                }

?>
```

EDITAR

feditar.html

```
<?php
   include('conexion.php');
                    $sql = "select * from agenda WHERE ID=".$_GET['id'];
                    $consulta = mysql_query($sql,$link);
                    if (isset($consulta))
                    {
                      $fila = mysql_fetch_array($consulta);
                                    }
?>
<form id="form1" name="form1" method="post" action='editar.php' >
    Nombres
    <input type="text" name="txtNombres" id="txtNombres" value="<?php echo $fila['Nombres'] ?>" /><br />
    Dirección
    <input type="text" name="txtDireccion" id="txtDireccion"  value="<?php echo $fila['Direccion'] ?>" /><br />
  Telefono
  <input type="text" name="txtTelefono" id="txtTelefono" value="<?php echo $fila['Telefono'] ?>" /><br />
  Email
  <input type="text" name="txtEmail" id="txtEmail" value="<?php echo $fila['Email'] ?>"  /><br />
  Celular
  <input type="text" name="txtCelular" id="txtCelular" value="<?php echo $fila['Celular'] ?>" /><br />
  Profesion
  <input type="hidden" name="txtID" id="txtID" value="<?php echo $_GET['id']; ?>"  /><br />
  <input type="submit" name="button" id="button" value="Enviar" />
  <input type="reset" name="button2" id="button2" value="Limpiar" />
</form>
```

EDITAR...

editar.php

```
<?php
   include('conexion.php');
            $sql = "update agenda set
Nombres='$_POST[txtNombres]',Direccion='$_
POST[txtDireccion]',Telefono='$_POST[txtTelef
ono]',Email='$_POST[txtEmail]',Celular='$_POS
T[txtCelular]' where ID=".$_POST['txtID'] ;
            //echo $sql;
            $consulta = mysql_query($sql,$link);
            if (isset($consulta))
            { echo "Cambios realizados con
exito";
            }
?>
```

ELIMINAR

eliminar.php

```
<?php
include('conexion.php');
$id=$_GET['id'];
$sql='delete from agenda where ID='.$id;
$consulta=mysql_query($sql,$link);
 ?>
Elemento eliminado con exito
```

MANEJO DE ARCHIVOS

Funciones Útiles

- Copiar:
  - **copy($origen,$destino)**
- Renombrar:
  - **rename($antes,$despues)**
- Eliminar:
  - **unlink($archivo)**

DIRECTORIOS

directorio.php

```
<?php
$directorio = "./";
$descriptor = opendir($directorio);
while ($entrada = readdir($descriptor) ) {
if (is_dir($directorio.$entrada) ) {
echo "[Directorio] " . $entrada . "<br>“;
} elseif ( is_file ($directorio. $entrada) ) {
echo " [Fichero] " . $entrada . "<br>";
}
}
closedir($descriptor);
?>
```

APERTURA DE ARCHIVOS

- fopen -> abre un archivo y le asigna un identificador id
- $id=fopen($path,$modo);

file handler

  - Path -> ruta completa del archivo a abrir

Modo	Significado
r	Sólo lectura
r+	Lectura y escritura
w	Sólo escritura, si no existe el archivo lo crea, si existe lo trunca
w+	Lectura y escritura, si existe lo trunca, si no existe lo crea
a	Modo append sólo escritura si no existe lo crea
a+	Modo append lectura y escritura si no existe lo crea

FUNCIONES

- Lectura de archivos
  - variable=**fgets**(file_handler, longitud)
    - Lee una línea de texto hasta el fin de línea o bien hasta que se cumpla la longitud indicada.
  - variable=**fread**(file_handler, cantidad)
    - Lee la cantidad de bytes indicados ignorando saltos de línea.
- Escritura de archivos
  - fwrite(file_handler, variable, longitud);
    - Escribe la variable al file_handler.
    - "longitud" (opcional) se escribirán tantos bytes como la longitud indicada o como la longitud de la variable,devuelve la cantidad de bytes escritos en el archivo.

FUNCIONES

- **Cierre de archivos**
  - fclose(file_handler)
    - Cierra un archivo abierto con fopen.
- **Fin de archivo**
  - boolean = feof(file_handler);
    - Devuelve verdadero si no quedan más bytes para leer en el archivo o si se produce algún tipo de error al leer.

EJEMPLO

escribir.php

```
<?php
$archivo = "miarchivo.txt";
$id = fopen($archivo, 'w+');
$cadena = "Aquí lo que queremos escribir".PHP_EOL;
fwrite($id, $cadena);
fwrite($id, "La segunda linea de lo que queremos escribir".PHP_EOL);
fwrite($id, "La tercera linea de lo que queremos escribir");
fclose($id);
?>
```

leer.php

```
<?php
header("Content-Type:text/html;charset=utf-8"); //enviar
la cabecera para utilizar utf
$descriptor = fopen ("miarchivo.txt","a+");
$linea_numero = 1;
while (!feof($descriptor)) {
$linea = fgets ($descriptor,4096) ;
echo "línea número: $numero_linea es: $linea","<BR>";
$linea_numero++;
}
fclose($descriptor);
?>
```

SUBIR ARCHIVOS AL SERVIDOR

- Muchas veces se enviar un archivo al servidor junto con la petición http
- El archivo debe ser enviado mediante el método POST
- Existe un tipo especial de input para este tipo file.
  - <input type="file">
- AL formulario se debe especificar el atributo para que tenga la capacidad de enviar archivos
  - **enctype**="multipart/form-data"

HTML

"subirF.php": Fichero en php encargado de subir el fichero al servidor.

"post": Enviaremos el fichero por la entrada estándar.

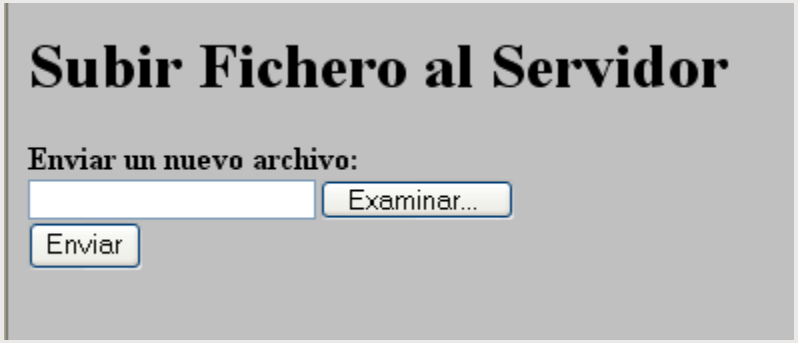
archivo.php

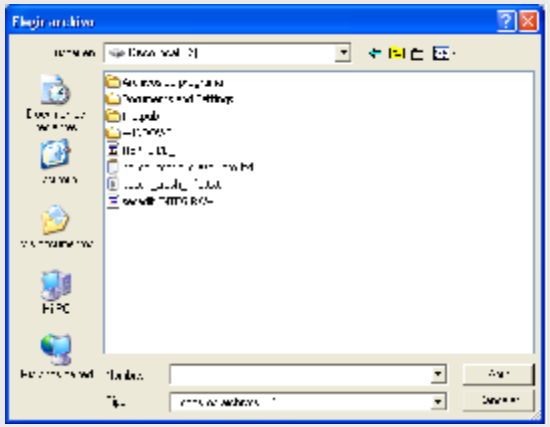
```
<form action="subirF.php" method="post"
      enctype="multipart/form-data">

      <input type="hidden" name="MAX_FILE_SIZE"
      value="100000">
      <b>Enviar un nuevo archivo: </b>
      <input type="file"" name="nfichero"
      <input type="submit" value="Enviar">
  </form>
```

"multipart/form-data":Permite subir datos y archivos en un mismo formulario.

"MAX_FILE_SIZE":Palabra reservada: Máxim tamaño del fichero(en bytes). Debería de comp el servidor.

Entrada de Fichero





¿COMO FUNCIONA?

- Una vez en el servidor, el archivo se almacena en un directorio temporal.
- La información del archivo es recibida en la variable $_FILES[] con el nombre del input
  - $_FILES[“nombre”][name]: Nombre original del archivo del cliente
  - $_FILES[“nombre”][tmp_name]: Nombre del archivo temporal en el servidor.
  - $_FILES[“nombre”][type]: Tipo de archivo
  - $_FILES[“nombre”][size]: Tamaño en bytes del archivo.
  - $_FILES[“nombre”][error]: Error asociado al archivo.
- En base a esta información se decide que hacer con el archivo recibido

RECIBIR ARCHIVO DEL USUARIO

subir.php

```
<?php
    header("Content-Type:text/html;charset=utf-8");
    //Información del Archivo
    $nombre_archivo = $_FILES['nArchivo']['name'];
    $tamano_archivo = $_FILES['nArchivo']['size'];
    $nombre_temporal= $_FILES['nArchivo']['tmp_name'];
    foreach ($_FILES["nArchivo"] as $indice => $valor)
    {
            echo "$indice: $valor<br>";
    }
    //compruebo si las características del archivo son las que deseo
    if ($tamano_archivo > 100000)
    {
         echo "El tamaño de archivo incorrecto.Tamaño maximo 100 Kb máximo. ";
    }
    else
    {
            if(copy($nombre_temporal,"images/".$nombre_archivo))
            echo "El archivo ha sido cargado correctamente.";
            else
            echo "Ocurrió algún error al subir el fichero. No pudo guardarse.";
    }
?>
```

PROPIEDADES DE CONFIGURACIÓN DE ARCHIVOS

- Fichero php.ini:
  - Permitir subir ficheros al servidor

    file_uploads = On
  - Seleccionamos un directorio temporal para los archivos

    upload_tmp_dir: "carp_temp/"
  - Tamaño máximo de los archivos.

    upload_max_filesize = 2M

ENVIAR CORREOS DESDE PHP

- Sólo existe una función para enviar correo electrónico desde PHP 5.
- La función mail () devolverá un valor *true si los datos son enviados correctamente.*

mail.php

```
<?php
$resultado = mail (" carlosmontellano@gmail.com" , "Ejemplo de Asunto", "Cuerpo del mensaje");
if ($resultado) {
echo "Correo enviado correctamente";
} else {
echo "El correo no ha podido enviarse" ;
}
?>
```

maildetalle.php

```
<?php
$correo = "luis@nccextremadura.org";
$asunto = "Ejemplo de Asunto";
$cuerpo = "Cuerpo del mensaje";
$cabecera = "From:
fernanda@\r\nbcc:pedro@nccextremadura.org\r\nContent-
      type:text/plain\r\nX-mailer: PHP/" . phpversion () ;
$resultado = mail($correo, $asunto, $cuerpo, $cabecera);
if ($resultado) {
echo "Correo enviado correctamente";
} else {
echo "El correo no ha podido enviarse";
}
```

PHPMAILER

- PHPMailer está basado en programación orientada a objetos.
- Contiene varios ficheros que deben guardarse en el directorio de PHP dedicado a librerías o en nuestro espacio de trabajo.

mail.php

```
<?php
require_once("phpmailer/class.phpmailer.php");
$correo = new phpmailer();
//Datos personales del emisor
$correo->From = "luis@nccextremadura.org";
$correo->FromName = "Luis Miguel Cabezas";

$correo->Subject = "Mensaje de prueba";

$correo->Body = "Cuerpo del mensaje";
//Direcci—n de destino
$correo->AddAddress("zeev@php.zend.com","Zeev Suraski");

if (!$correo->Send()) {
                    echo "Correo enviado correctamente";
} else {
                    echo "El correo no ha podido enviarse";
}
?>
```

GRAFICOS EN PHP

- La librería gráfica GD está escrita en lenguaje C y permite crear y manipular gráficos fácilmente.
- Permite importar y exportar gráficos de distinto tipo (GIF, JPG y PNG)

imagensimple.php

```
<?php
$imagen =
imagecreatefrompng("zend.png");
header("Content-Type:
image/png");
imagepng($imagen);
?>
```

EJEMPLO SUBIR ARCHIVO Y REDUCIRLO

imagen.php

```
<?php
if (isset($_FILES["fichero"])) {
		foreach($_FILES["fichero"] as $indice=>$valor)
		{
			echo $indice,$valor,'<br>';
		}
		echo "Imagen Original:<br>";
		$fotografia = $_FILES["fichero"]["tmp_name"];
		copy($fotografia,$_FILES["fichero"]["name"]);
		$foto_copia =
$_FILES["fichero"]["name"]."&modo=original";
		$url =
"clase_imagen.php?fotografia=$foto_copia";
		echo "<img src=\"$url\">";
		echo "Imagen Miniatura:<br>";
		$foto_copia =
$_FILES["fichero"]["name"]."&modo=miniatura";
		$url =
"clase_imagen.php?fotografia=$foto_copia";
		echo "<img src=\"$url\">";
}
?>
```

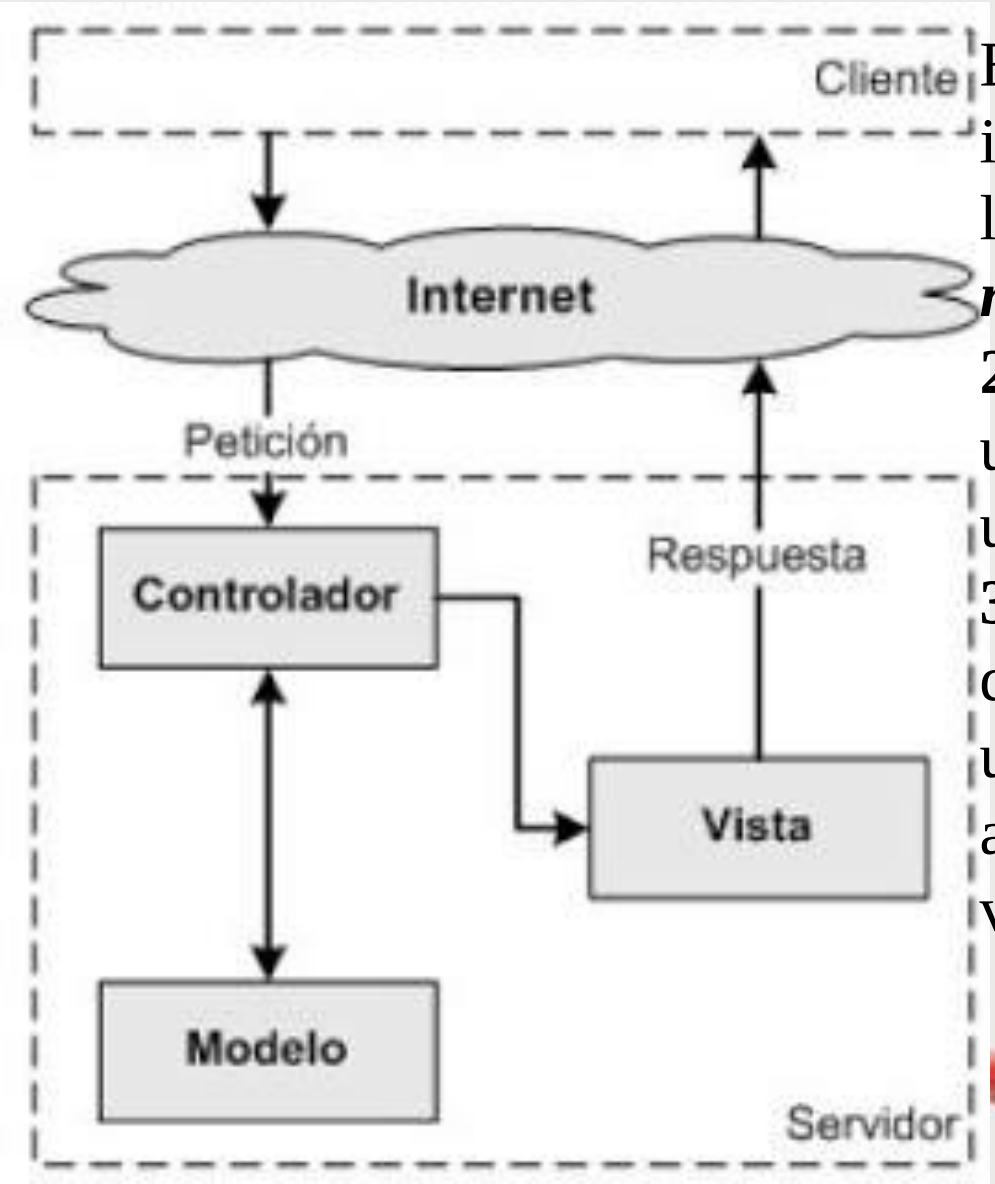
CAPTCHA

- Es un mecanismo de seguridad para evitar llenao automatico de formularios

form-catcha.php

```
<?php
session_start();
if(isset($_POST["palabra"])) {
if($_POST["palabra"] == $_SESSION["oculto"])
die("¡Felicidades! Ha leído correctamente la palabra.");
else
die("No has adivinado la palabra. Prueba otra vez");
} else {
?>
<form method="post" action="">
<img src="capu-captcha.php">
<input type="text" name="palabra">
<input type="submit" value="Comprobar">
</form>
<?php
}
?>
```

MODELO VISTA CONTROLADOR



El **modelo** representa la información con la que trabaja la aplicación, es decir, su ***lógica de negocio***.

2. La **vista** transforma el modelo en una página web que permite al usuario interactuar con ella.

3. El **controlador** se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Frameworks en PHP

- Los Frameworks ayudan en el desarrollo de software
- Proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez
- Se utiliza la Programación Orientada a Objetos (POO), permitiendo la reutilización de nuestro código

Algunos Frameworks

- Zend Framework
  - Es simple, no necesita instalación especial, requiere PHP 5 e incorpora el patrón MVC.
  - Se debe descargar y copiarlo hacia nuestro servidor local, veamos a través de un pequeño ejemplo cómo podemos crear un lector de RSS. Los ficheros que creemos podemos copiarlos dentro del directorio “library”.

```
<?php // Componente requerido require_once 'Zend/Feed.php';
 // Incluimos la dirección de nuestro fichero rss que deseamos importar
$feed = Zend_Feed::import('http://localhost/rss.php');
// Se recorrerán todos los item del fichero, mostrando el titulo y el enlace
foreach ($feed->items as $item)
   { echo "<p>" . $item->title() . "<br />";
    echo $item->link() . "</p>";
} ?>
```

Algunos Frameworks

- **Symfony**
  - Diseñado con el objetivo de optimizar la creación de las aplicaciones web, con el uso de sus características.
  - Posee una librería de clases que permiten reducir el tiempo de desarrollo.
  - Desarrollado en PHP5, se puede utilizar en plataformas *nix (Unix, Linux) y Windows.
  - Requiere de una instalación, configuración y líneas de comando, incorpora el patrón MVC, soporta AJAX, plantillas y un gran número de bases de datos.

OTROS FRAMEWORKS

- PHP Prado
- CakePHP
- Qcodo
- Kumbia
- PHP4ECore
- CodeIgniter
- Yii Framework
- Tomates Framework