PHP: Programación orientada a objeto

La mayoría de los lenguajes de programación modernos son orientados a objetos o en su defecto se aproximan mucho a éstos permitiendo algunas de sus características como es el caso de PHP. La programación OO principalmente hace uso de clases, objetos, relaciones, instancias, propiedades y métodos.

Objetos y clases

Cuando hablamos de software OO los objetos casi siempre son elementos físicos, como puede ser un cliente, proveedor, etc. o elementos conceptuales que existen en el entorno software, por ejemplo un objeto encargado del mantenimiento de archivos. El objetivo es representar a éstos elementos de la vida real y a los conceptuales como unidades de software.

La programación OO esta pensada para construir objetos que contienen atributos y operaciones de manera que cubran nuestras necesidades. Los atributos son variables que contienen información del estado de un objeto. Y las operaciones también conocidas como métodos, funciones y acciones realizan modificaciones del propio objeto o realizan alguna acción externa a éste.

Una de las principales ventajas de la programación OO es el concepto de encapsulación, conocido también como protección de datos, mediante el cual solo se pueden modificar los datos de un objeto accediendo a través de sus métodos u operaciones (interfaz del objeto). La funcionalidad de un objeto esta sujeta a los datos que este maneja, una ventaja de usar objetos es que podemos modificar la funcionalidad de éste, añadir mejoras o corregir errores sin necesidad de cambiar su interfaz. Ya que en caso contrario un proyecto estaría sujeto a un mayor número de fallos y los cambios serían más costosos.

En algunas áreas de la programación de aplicaciones Web el uso de la programación OO está desestimada, usándose una metodología estructurada basada en funciones, esto es debido a que determinados proyectos no son lo suficientemente extensos como para aplicarles una metodología OO.

En la programación OO los objetos son únicos y son instancias a una clase determinada. En principio se define la clase con los atributos y métodos correspondientes y luego se crea el objeto que esta basado en una determinada clase (esto se conoce como instancia).

Cómo crear clases, atributos y operaciones en PHP

Hasta ahora hemos hablado de las clases de una forma conceptual, a continuación veremos como se crean, para crear una clase en PHP usaremos la palabra reservada class.

La estructura mínima de una clase es la siguiente:

```
class NombreClase {

}
```

Para que una clase sea útil, necesita atributos y operaciones. Podemos crear atributos como si de variables se trataran, con la palabra reservada var

```
class NombreClase {
var $atributo1;
var $atributo2;
}
```

Podemos crear métodos declarando funciones dentro de la definición de la clase, el siguiente código crea una clase llamada NombreClase con dos operaciones que no hacen nada. A metodo1 no le pasamos ningún parámetro y a metodo2 le pasamos dos parámetros.

```
class NombreClase {
function metodo1() {
}
function metodo2($param1, $param2) {
}
}
```

Qué es el constructor de una clase

Las clases soportan un tipo de función especial que se conoce como constructor. El constructor es llamado cuando se crea el objeto. Normalmente utiliza para inicializar tareas como: asignación de valores a determinados atributos, crear nuevos objetos necesarios para el correcto funcionamiento del objeto, etc.

El constructor se declara de la misma forma que los métodos, lo único que debemos tener en cuenta es que debe tener el mismo nombre que la clase. A continuación veremos como se declara el constructor de una clase:

```
class NombreClase {

function NombreClase($param) {
echo "Constructor llamado con el parámetro $param";
}
}
```

Cómo usar objetos, instanciar una clase

Después de haber declarado una clase, ya podemos usarla creando un objeto que es una instancia a esa clase como ya se ha mencionado anteriormente. Es muy importante que esto quede claro, los objetos son instancias a una clase, por lo tanto cada objeto es único.

En php creamos un objeto usando la palabra reservada new. También debemos indicar a que clase va a instanciar el objeto que creemos y pasarle los parámetros (si los requiere) al constructor, en el siguiente código vamos a ver un ejemplo de esto, para ello tomamos como referencia la clase anterior NombreClase:

```
$a = new NombreClase("Primero");
$b = new NombreClase("Segundo");
$c = new NombreClase();
```

Cómo usar los atributos de una clase

Una clase, tiene un puntero especial al que podemos referenciar como $this. Si nuestra clase tiene un atributo llamado $atributo, podemos hacer referencia a este desde nuestra clase (métodos) de la siguiente forma $this->atributo, a continuación podemos ver el código de ejemplo del acceso a un atributo de clase desde un método de la propia clase.

```
class NombreClase {
var $atributo;
function metodo($param) {
$this->atributo = $param;
echo $this->atributo;
}
}
```

Algunos lenguajes permiten limitar el acceso desde fuera a los atributos de una clase declarándolos como privados o protegidos ("private", "protected"). PHP no soporta esta característica y todos atributos y métodos pueden ser vistos desde fuera de la clase, lo que quiere decir que siempre son públicos.

Podemos realizar la misma acción que anteriormente desde fuera de la clase, usando esta sintaxis.

```
NombreClase {
var $atributo;
}
$a = new NombreClase();
$a->atributo = "valor";
echo $a->atributo;
```

Cómo usar los métodos de una clase

Los métodos de una clase se pueden llamar de la misma forma que se llaman a los atributos, supongamos que tenemos la siguiente clase:

```
class nombreClase {
function metodoa() {
return "Has llamado al método A";
}
function metodob($parametro1) {
return "Método B llamado con parámetro: ".$paramtro1;
}
}
```

Creamos un objeto de la case nombreClase con el nombre $obj de la siguiente forma:

```
$obj = new nombreClase();
```

Los métodos de un objeto, son llamados de la misma forma que se llaman a funciones normales, pasándoles como es el caso de metodob el parámetro que necesiten. Como los métodos son funciones que pertenecen a un objeto, deberemos indicar en la llamada el objeto que los incluye, la forma de llamar a un método es idéntica a como llamamos a los atributos de un objeto.

```
$obj->metodoa();
$obj->metodob("parámetro que pasamos");
```

Si nuestras operaciones, devuelven algún valor, lo capturamos asignándoselo a una variable, podemos ver el siguiente ejemplo

```
$txt_retorno = $obj->metodob("Hola");
```

$txt_retorno contendrá la cadena de texto:
"Metodo B llamado con parámetro: Hola"

Qué es la Herencia en PHP y como implementarla

Como su nombre indica el concepto de herencia se aplica cuando creamos una clase, que va a heredar los métodos y atributos de una ya definida, entonces la clase que hemos creado es una subclase. Para que una clase sea subclase de otra ya creada deberemos usar la palabra reservada extends en el siguiente código podremos ver como creamos una clase llamada SubClaseA que heredará los métodos y atributos de una clase definida con anterioridad llamada ClaseA.

```
class SubClaseA extends ClaseA {
var $atributo2;
function operacion2() {
}
}
```

Tenemos la clase ClaseA que es definida de la siguiente forma:

```
Class ClaseA {
var $atributo1;
function operacion1(){
}
}
```

Si creamos un objeto de la clase SubClaseA este heredará todos los métodos de la clase ClaseA, por lo tanto el siguiente código es válido:

```
$x = new SubClaseA();
$x->operacion1();
$x->atributo1 = 100;
$x->operacion2();
$x->atributo2 = 200;
```

Como podemos observar aunque declaremos un objeto de la clase SubClaseA, al ser una clase extendida de ClaseA podemos hacer uso de todos los métodos y atributos definidos en ClaseA como si estuvieran contenidos en SubClaseA.

Debemos tener en cuenta que la herencia solo trabaja en una dirección, la subclase o clase hija hereda las características de su clase padre o superclase, pero la clase padre no posee las características de la hija. Para el caso anterior ClaseA no tendría atributo2 ni metodo2();

Ejercicio # 1

```
<?php
class cuadrado {
// Estos son ATRIBUTOS de los objetos
        var $num=10;
// Este es el METODO para calcular
        function calcularCuadrado() {
        return ($this->num * $this->num);
        }
}
// Creamos el Objeto
$objeto = new cuadrado();
//Asignamos un atributo
//$objeto->num = 3;
// Invocamos un método
echo $objeto->calcularCuadrado();
?>
```

Ejercicio # 2

```
<?php
class imagen {
// Estos son ATRIBUTOS de los objetos
        var $src;
        var $border;
 // Esta función es el CONSTRUCTOR
        function imagen($src,$border) {
        $this->src=$src;
        $this->border=$border;
        }
// Esta función es un METODO
        function Imprimir() {
        echo " <img src=";
        echo $this->src;
        echo " border=";
        echo $this->border;
        echo " >";
        }
}
// Creamos el Objeto
$logo = new Imagen("imagen/msn2.jpg",8);
// Invocamos el método
$logo->Imprimir();
?>
```

Ejercicio # 3

```
<?
class Aritmetica {
// Estos son ATRIBUTOS de los objetos
        var $a;
        var $b;
// Este es el CONSTRUCTOR
        function Aritmetica($a, $b) {
        $this->a = $a;
   $this->b = $b;
        }
// Este es el METODO para sumar
        function sumar() {
        return ($this->a + $this->b);
        }
// Este es el METODO para restar
        function restar() {
        return ($this->a - $this->b);
        }
// Este es el METODO para multiplicar
        function multiplicar() {
        return ($this->a * $this->b);
        }
// Este es el METODO para dividir
        function dividir() {
        return ($this->a / $this->b);
        }
}
// Creamos el Objeto
$objeto = new Aritmetica(5, 8);
// Invocamos un método
echo $objeto->sumar();
?>
```

Ejercicio # 4

```
<?
// Superclase
class cuadrado {
// Estos son ATRIBUTOS de los objetos
        var $num;

// Este es el METODO para calcular el cuadrado
        function calcularCuadrado() {
        return ($this->num * $this->num);
        }
}

//Subclase
class cubo extends cuadrado{
// Este es el METODO para calcular el Cubo
        function calcularCubo() {
        return ($this->calcularCuadrado() * $this->num);
        }
}
// Creamos el Objeto
$objeto = new cubo();
//Asignamos un atributo
$objeto->num = 3;
// Invocamos un método
echo $objeto->calcularCubo();
?>
```

Ejercicio # 5: Conexiones

Primero creamos el archivo donde almacenaremos los datos de nuestro host, database, username y password de mysql, lo llamaremos **conexion.php**

```
<?php
class conexion{
var $hostname = "localhost"; //IP del servidor
var $database; //Nombre de la base de datos
var $username = "root";//usuario mysql
var $password = "root";//password de usuario mysql
function conectar($base){
$this-> database=$base;
 $link = @mysql_connect($this->hostname, $this->username, $this->password);
 if(!$link){
 echo "conexion fallida";
 exit;
 }else{
 mysql_select_db($this->database,$link);
 }
return $link;
}
}
```

Si la conexion se utilizara en algún otro archivo php, debe ser incluido en ese archivo el cual contiene la clase e instanciar la misma, un ejemplo de esto seria:

```
<?php
include_once("conexion.php");
$conexion=new conexion();
$conn=$conexion->conectar("prueba");
$sql="select * from tabla";
$consulta=mysql_query($sql,$conn);
?>
```

Ciclo: `<?php while (list($nombre,$apellido) = mysql_fetch_array($consulta)){?>`

Mostrar: `<?php print $nombre ?>` Cerrar ciclo: `<?php }?>`

Ejercicio nº 6: Combos desplegables enlazados:

```
-- Estructura de tabla para la tabla `ciudad`
CREATE TABLE `ciudad` (
  `id_ciudad` int(3) NOT NULL,
  `ciudad` varchar(20) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- Volcar la base de datos para la tabla `ciudad`
INSERT INTO `ciudad` VALUES (1, 'caracas');
INSERT INTO `ciudad` VALUES (2, 'maracay');
```

```
-- Estructura de tabla para la tabla `parroquia`
CREATE TABLE `parroquia` (
  `id_parroquia` int(3) NOT NULL,
  `id_ciudad` int(3) NOT NULL,
  `parroquia` varchar(20) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- Volcar la base de datos para la tabla `parroquia`
INSERT INTO `parroquia` VALUES (1, 1, 'caricuao');
INSERT INTO `parroquia` VALUES (2, 1, '23 de enero');
INSERT INTO `parroquia` VALUES (3, 2, 'las delicias');
INSERT INTO `parroquia` VALUES (4, 2, 'centro');
```

```
<?php
// datos de conexion a la BD.
include_once("conexion.php");
$conexion=new conexion();
$conn=$conexion->conectar("prueba");
 // Obtener el $id_padre..
   $id_ciudad=$_POST['id_ciudad'];
// Inicio Formulario .. PHP_SELF enviamos a si mismo (a este script).
   echo "<form action=\"".$_SERVER['PHP_SELF']."\" method=\"POST\">\n\n";
// Formar Select "Padre".
   echo "<select name=\"id_ciudad\" onChange=\"this.form.submit()\">\n";
   echo "<option value=\"\"> Seleccione un Item </option>\n";
   $SQLconsulta_ciudad="SELECT * FROM ciudad";
   $consulta_padre = mysql_query($SQLconsulta_ciudad,$conn) or die(mysql_error());
   while   ($registro_ciudad=mysql_fetch_assoc($consulta_padre)){
    // Se mira si el ID del registro es el mismo q el $id_ciudad q recibimos si hemos cambiado el select parroquia.
    // Se selecciona en consecuencia (selected) la opción elegida.
     if ($id_ciudad == $registro_ciudad['id_ciudad']){
        echo "<option value=\"".$registro_ciudad['id_ciudad']."\" selected>".$registro_ciudad['ciudad']."</option>\n";
     } else {
        echo "<option value=\"".$registro_ciudad['id_ciudad']."\">".$registro_ciudad['ciudad']."</option>\n";
     }
   }
   echo "</select>\n\n";
mysql_free_result($consulta_padre); // Liberar memoria usada por consulta.
// Formar Select "Parroquia"
   echo "<select name=\"id_parroquia\">\n";
// Si $id_ciudad no tiene valor (caso de que no se ha seleccionado ningua opcion del select hijo
   // se muestra el mensaje de "seleccine un item" (del select padre).
   if (!empty($id_ciudad)){
$SQLconsulta_parroquia="SELECT * FROM parroquia WHERE id_ciudad='$id_ciudad'";
      $consulta_parroquia = mysql_query($SQLconsulta_parroquia,$conn) or die(mysql_error());
      // se mira el total de registros de la consulta .. si es 0 se muestra mensaje en el select ..
      if (mysql_num_rows($consulta_parroquia) != 0){
         While   ($registro_parroquia=mysql_fetch_assoc($consulta_parroquia)){
           echo "<option
value=\"".$registro_parroquia['id_parroquia']."\">".$registro_parroquia['parroquia']."</option>\n";
         }
       } else {
          echo "<option value=\"\"> No hay registros para este Item </option>";
       }
   } else {
       echo "<option value=\"\"> <-- Seleccione un Item  </option>";
   }mysql_free_result($consulta_parroquia); // Liberar memoria usada por consulta.
   ?>
```

Ejercicio # 7: Gráfico de Línea
Colocar librería gráfica jpgraph

```
<?php
include_once("conexion.php");
$conexion=new conexion();
$conn=$conexion->conectar("prueba");

include("jpgraph/jpgraph.php");
include("jpgraph/jpgraph_line.php");

$sql="select * from grafico";
$consulta=@mysql_query($sql,$conn);

$xdata= array();
$ydata= array();

while($row = mysql_fetch_array($consulta,
MYSQL_ASSOC)){
array_push ($xdata,$row["mes"]);
array_push ($ydata,$row["monto"]);
}

$graph = new Graph(450,350,"auto");
$graph->SetScale("textlin");
$graph->xaxis->SetTickLabels($xdata);

$lineplot1 = new LinePlot($ydata);
$lineplot1->SetColor("red");
$lineplot1->mark-
>SetType(MARK_FILLEDCIRCLE);
$lineplot1->mark->SetFillColor("red");
$lineplot1->mark->SetWidth(2);

$graph->img->SetMargin(55,20,20,55);
$graph->title->Set("Montos                Indice
de Morosidad");
$graph->xaxis->title->Set("Meses");
$graph->ygrid-
>SetFill(true,'#EFEFEF@0.5','#F9BB64@0.5');
$graph->SetShadow();

$graph->Add($lineplot1);
$graph->Stroke();
mysql_close($conexion);
?>
```

Ejercicio # 8: Gráfico de Torta
Colocar librería gráfica jpgraph

```
<?php
include_once("conexion.php");
$conexion=new conexion();
$conn=$conexion->conectar("prueba");

include ("jpgraph/jpgraph.php");
include ("jpgraph/jpgraph_pie.php");
include ("jpgraph/jpgraph_pie3d.php");

$sql="select * from grafico";
$consulta=@mysql_query($sql,$conn);

$xdata= array();
$ydata= array();

while($row = mysql_fetch_array($consulta,
MYSQL_ASSOC)){
array_push ($xdata,$row["mes"]);
array_push ($ydata,$row["monto"]);
}
$graph = new PieGraph(470,350,"auto");
$graph->SetShadow();
$graph->img->SetMargin(5,5,5,5);
$graph->title->Set("Indice de Morosidad");
$graph->title->SetColor("darkblue");
$graph->legend->Pos(0.05,0.2);
$p1 = new PiePlot3d($ydata);
$p1->SetTheme("sand");
$p1->SetCenter(0.4);
$p1->SetSize(80);
$p1->SetAngle(60);
$p1->SetStartAngle(60);
$p1->value->SetColor("navy");
$p1->SetEdge("navy");

$p1->SetLegends(array("Ene","Feb","Mar","Abr"));

$graph->Add($p1);
$graph->Stroke();
?>
```

Ejercicio # 9: Gráfico de barra: **Colocar librería gráfica jpgraph**	**Ejercicio # 10: Imprimir archivos PDF** **Colocar librería de reportes en pdf**

<?php
include_once("mysql.php");
$conexion=new conexion();
$conn=$conexion->conectar("prueba");
include("jpgraph/jpgraph.php");
include ("jpgraph/jpgraph_bar.php");
$sql="select * from grafico";
$consulta=@mysql_query($sql,$conn);
$xdata= array();
$ydata= array();
 	
while($row = mysql_fetch_array($consulta,
MYSQL_ASSOC)){
array_push ($xdata,$row["mes"]);
array_push ($ydata,$row["monto"]);
}
 	
$months = $gDateLocale->GetShortMonth();
$graph = new Graph(470,350);
$graph->SetScale("textlin");
$graph->SetMarginColor('white');
$graph->SetMargin(55,20,20,55);
$graph->SetBox();
$graph->SetFrame(false);
$graph->ygrid-
>SetFill(true,'#DDDDDD@0.5','#BBBBBB@0.5');
$graph->ygrid->SetLineStyle('dashed');
$graph->ygrid->SetColor('gray');
$graph->xgrid->Show();
$graph->xgrid->SetLineStyle('dashed');
$graph->xgrid->SetColor('gray');
$graph->img->SetMargin(55,20,20,55);
$graph->title->Set("Indice de Morosidad");
$graph->xaxis->title->Set("Meses");
$graph->yaxis->title->Set("Indice");
$bplot = new BarPlot($ydata);
$bplot->SetWidth(0.6);
$fcol='#440000';
$tcol='#FF9090';
$bplot->SetLegend("Mora");
$bplot-
>SetFillGradient($fcol,$tcol,GRAD_LEFT_REFLECTION);
$bplot->SetWeight(0);
$graph->Add($bplot);
$graph->Stroke();
?>
	<?php
	include('class.ezpdf.php');
	$pdf =& new Cezpdf('a4');
	$pdf->selectFont('fonts/Courier.afm');
	$pdf->ezText("titulo\n",20);
	$pdf->ezText("prueba\n\n",12);
	$pdf->ezText("Fecha: ".date("d/m/Y"),10);
	$pdf->ezText("Hora: ".date("H:i:s"),10);
	$pdf->ezStream();
	?>

Ejercicio # 11: Imprimir archivo pdf
Colocar librería de reportes en pdf

```
<?php
error_reporting(7); //esto debe ir en la primera linea
include ('class.ezpdf.php');

include_once("conexion.php");
$conexion=new conexion();
$conn=$conexion->conectar("prueba");

$xclave=$_GET['clave'];

$sql="select * from cliente";
$rs=mysql_query($sql,$conn);
$row=mysql_fetch_assoc($rs);

//titulos para la tabla
$titulo[0]='Codigo';
$titulo[1]='Nombre del Cliente';
$titulo[2]='Telefono';

do{
$registro[$titulo[0]]=$row['codigo'];
$registro[$titulo[1]]=$row['nombre'];
$registro[$titulo[2]]=$row['telefono'];

$tabla[]=$registro;
}while($row=mysql_fetch_assoc($rs));

//tamaño de la hoja carta y letra helvetica
$pdf=new Cezpdf('letter','landspace');
$pdf->selectFont('./fonts/Courier-Bold.afm');

//para colocar la imagen
$imagen=ImageCreatefromjpeg('imagen/msn2.jpg');
$pdf->addImage($imagen,10,$pdf->y-30,50,0);

//para bajar 15 pixeles y colocar el titulo del reporte y luego 35 pixeles después del titulo

$pdf->ezsetdy(-10);
$pdf->ezText("Distribuidora Vimportet",12,array('justification'=>'center'));
$pdf->ezsetdy(-5);

$pdf->ezsetdy(-10);
$pdf->ezText("Listado de Clientes",12,array('justification'=>'center'));
$pdf->ezsetdy(-10);

$pdf->selectFont('./fonts/Helvetica.afm');
$pdf->ezTable($tabla); //$tabla es la tabla que se lleno en el do while
$pdf->ezStream();
?>
```

Ejercicio # 12: Foro por temas

```
CREATE TABLE `mensaje` (
  `id` int(11) NOT NULL auto_increment,
  `nombre` varchar(30) NOT NULL,
  `tema` text NOT NULL,
  `fecha` date NOT NULL,
  `hora` varchar(10) NOT NULL,
  PRIMARY KEY  (`id`)
)
```

```
CREATE TABLE `respuestas` (
  `id` int(11) NOT NULL auto_increment,
  `nombre` varchar(30) NOT NULL,
  `tema` text NOT NULL,
  `fecha` date NOT NULL,
  `hora` varchar(10) NOT NULL,
  `id_mensaje` int(11) NOT NULL,
  PRIMARY KEY  (`id`)
)
```

Archivo Foro:

```
<a href="addtema.php">Agregar Tema</a>
 <?php
include "conexion.php";
$sql="select id, nombre, tema, fecha from mensaje order by fecha,hora desc";
$consulta=mysql_query($sql,$conexion);
?>
Tabla con: tema, autor, respuestas y fecha

<?php while($fila=mysql_fetch_array($consulta)){ ?>

Tema= <a href="ver.php?id=<?php print $fila[0]; ?>"><?php print $fila[2]; ?>

Autor= <?php print $fila[1]; ?>

Respuesta= <?php $buscar="select id from respuestas where id_mensaje='$fila[0]'";
$con=mysql_query($buscar,$conex); $mostrar=mysql_num_rows($con); print $mostrar; ?>

Fecha= <?php print substr($fila[3],8,2).substr($fila[3],4,4).substr($fila[3],0,4); ?>

Cerrar el while= <?php } mysql_close($conex); ?>
```

Archivo addtema.php:

Interfaz: form+tabla 4x2, nombre+mensaje

```
<?php
if ($_POST[nombre]) {
include "conexion.php";
$fecha= date ( "Y-m-d" );
$hora= date ( "H:i:s" );
$sql="insert into mensaje values('','$_POST[nombre]','$_POST[tema]','$fecha','$hora')";
$consulta=mysql_query($sql,$conexion);
if (!mysql_error()) { ?>
<script>alert('Mensaje agregado correctamente');</script>
<script>location='foro.php'</script>
<?php } else { print mysql_error();?>
<script>alert('Estamos en mantenimiento');</script>
<?php
}mysql_close($conexion); }
?>
```

Archivo: tema.php

```
<?php
include "conexion.php";
$sql="select * from mensaje where id='$_GET[id]'";
$ver=mysql_fetch_array(mysql_query($sql,$conexion));
$respuesta="select * from respuestas where id_mensaje='$_GET[id]' order by id desc";
$consulta=mysql_query($respuesta,$conexion);
?>
```

Interfaz: tabla 1x2

```
Responder=<a href="responder.php?id=<?php print $_GET[id]; ?>">Responder</a>
Regresar al foro=<a href="foro.php">Regresar al foro</a>

Interfaz: tabla 1x3 para el tema
Tema= <?php print $ver[2]; ?>

Autor=<?php print $ver[1]; ?>

Fecha y hora= <?php print substr($ver[3],8,2).substr($ver[3],4,4).substr($ver[3],0,4); ?> - <?php print $ver[4]; ?>
```

Interfaz: tabla 4x2

```
Titulo: respuesta
<?php while($res=mysql_fetch_array($consulta)){ ?>

Responde= <?php print $res[1]; ?>

Respuesta= <?php print $res[2]; ?>

Fecha y hora= <?php print substr($ver[3],8,2).substr($ver[3],4,4).substr($ver[3],0,4); ?> - <?php print $ver[4]; ?>

Cierra el while = <?php } mysql_close($conex); ?>
```

Archivo: responder.php

```
<?php
include "conexion.php";
$sql="select * from mensaje where id='$_GET[id]'";
$ver=mysql_fetch_array(mysql_query($sql,$conexion));
?>
```

Interfaz: tabla 5x2

```
Tema= <?php print $ver[2]; ?>

Campo oculto= <input name="oculto" type="hidden" id="oculto" value="<?php print $_GET[id]; ?>" />

Después del form

<?php
if ($_POST[nombre]) {
include "conexion.php";
$fecha= date ( "Y-m-d" );
$hora= date ( "H-i-s" );
$sql="insert into respuestas values('','$_POST[nombre]','$_POST[tema]','$fecha','$hora','$_POST[oculto]')";
$consulta=mysql_query($sql,$conex);
```

```
if (!mysql_error()) { ?>
<script>alert('Mensaje agregado correctamente');</script>
<script>location='ver.php?id=<?php print $_GET[id]; ?>'</script>
<?php } else { print mysql_error();?>
<script>alert('Estamos en mantenimiento');</script>
<?php
}
mysql_close($conex); }
?>
```