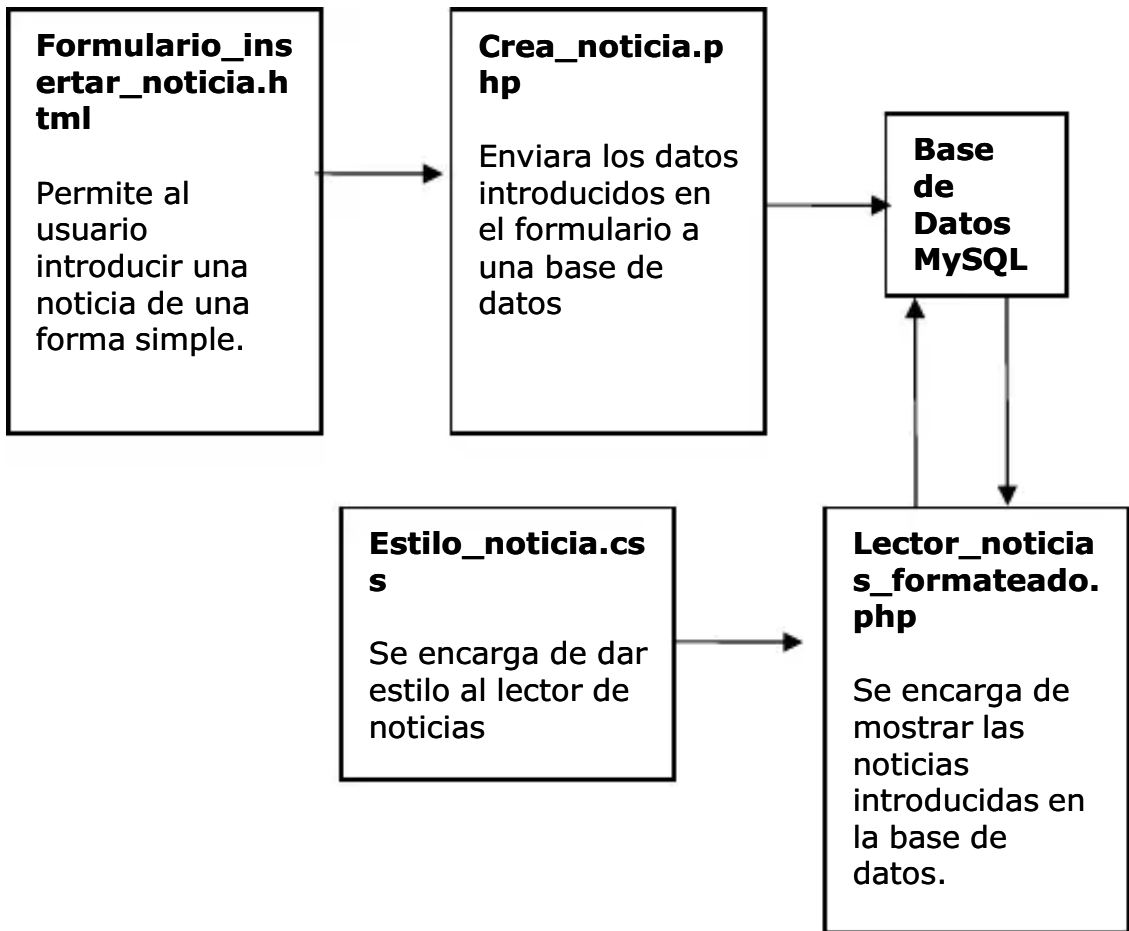


Gestor de noticias dinámico con PHP y MySQL

por Aarón Jiménez



Gestor de noticias en PHP

Vamos a crear una aplicación Web donde nosotros podamos introducir una noticia, esa noticia se almacene en una base de datos y por otro lado (por el lado del usuario cliente) haya otra página que lea las noticias que van apareciendo en la base de datos.

Una vez entendida la idea nos ponemos manos a la obra.

Empezamos creando la tabla en nuestra base de datos:

Nuestra base de datos se llama LIBRERÍA.

Y la tabla se llama NOTICIAS; esta tabla contendrá una serie de campos donde se almacenaran el titulo, el texto de la noticia, la fecha, la activación o no de la noticia, y el tipo de noticia.

Creamos nuestra nueva base de datos LIBRERIA. Entramos en LIBRERÍA en la pestaña SQL y escribimos lo siguiente:

```
CREATE TABLE noticias(  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT ,  
titulo TEXT,  
texto TEXT,  
fecha DATE,  
activar BIT,  
tipo_noticia VARCHAR(12) ,  
imagen VARCHAR (30),  
PRIMARY KEY (id)  
)
```

Este código SQL nos creará la siguiente tabla:

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
id	int(10)		UNSIGNED	No		auto_increment
titulo	text	latin1_swedish_ci		Sí	NULL	
texto	text	latin1_swedish_ci		Sí	NULL	
fecha	date			Sí	NULL	
activar	bit(1)			Sí	NULL	
tipo_noticia	varchar(12)	latin1_swedish_ci		Sí	NULL	
imagen	varchar(30)	latin1_swedish_ci		No		

Es en esta tabla donde insertaremos todos los datos de la noticia.

titulo TEXT,

texto TEXT,

Aquí almacenamos el texto y el titulo de la noticia.

fecha DATE,

La fecha de la noticia que debe ser del tipo DATE

activar BIT,

Este campo lo utilizaremos para escoger las noticias que el usuario quiera publicar. Como es del tipo BIT(1) solo puede coger dos valores. Un BIT es un representador booleano; o está en '1' o en '0'.

tipo_noticia VARCHAR(12) ,

imagen VARCHAR (30),

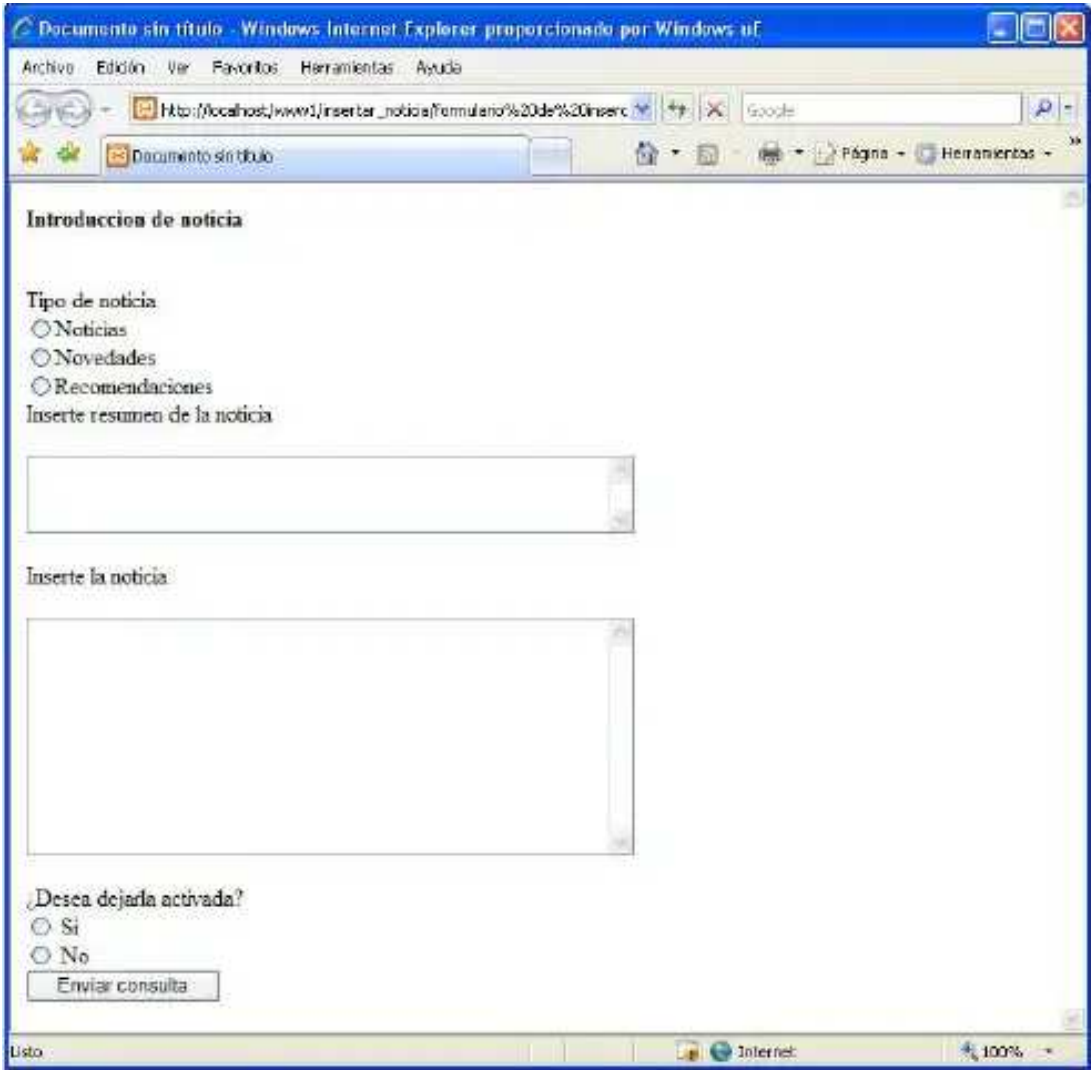
El tipo de noticia es para guardar una palabra que en el formulario veremos que pondremos tres opciones a elegir (noticias, novedades, recomendaciones). Y el campo imagen lo utilizaremos para guardar una palabra que será el nombre de la imagen.

Las imágenes no las subiremos a la base de datos, si no que las subiremos al directorio donde guardemos los el formulario.html, el crea_noticia.php y el lector_noticia_formateada.php.

El lenguaje de MySQL no es casesensitive.

Este manual no pretende ser una guía de código MySQL, pero con este tipo de sentencias básicas podemos crear una base de datos como la que vemos en la imagen.

El siguiente paso sería crear un formulario que nos permitiese insertar un titulo para la noticia, insertar un texto, escoger el tipo de noticia que será y si la queremos activar o no. El resultado final debería quedar parecido a esto:



El código para crear este formulario es:

```
<title>Formulario inserción noticia</title>
<style type="text/css">
.box {font-family: Arial, Helvetica, sans-serif;font-size: 12px;border: 1px solid
#000000;}
</style>
</head> <body>
<form action="crea_noticia.php" method="post" enctype="multipart/form-data"
name="uploadform">
  <strong>Introducción de noticia</strong> <br>
  <br />
  <br />
  Tipo de noticia<br>
  <input name="tiponoticia" value="Noticias" type="radio">
  Noticias<br>
  <input name="tiponoticia" value="Novedades" type="radio">
  Novedades<br>
  <input name="tiponoticia" value="Recomendaciones" type="radio">
  Recomendaciones<br>
  Inserte título de la noticia
  <p>
    <textarea cols="50" rows="3" name="resumen_noticia"> </textarea>
  </p>
  Inserte texto de la noticia
  <p>
    <textarea cols="50" rows="10" name="texto_noticia"> </textarea>
  </p>
  <p> ¿Desea dejarla activada? <br />
    <input name="activar" value="1" type="radio"> Si <br>
    <input name="activar" value="0" type="radio"> No <br>
  </p>
  <p>
    Enviar imagen: <input name="foto" type="file" />
    <input name="submit" type="submit" value="Enviar formulario">
  </p>
</form>
</body>
```

El formulario será la vía para que un usuario sin muchos conocimientos de informática pueda acceder a la pagina "**formulario de inserción de noticia.html**" escribir el texto de una noticia, el titulo, cargar la foto y enviarla.

Ese envío será hacia una pagina.php que procese los datos del formulario y los introduzca en la base de datos.

El formulario envia los datos a crea_noticia.php

El código es el siguiente:

crea_noticia.php

```
<?
include("config.php");
if(is_uploaded_file($_FILES['foto']['tmp_name']))
{
    $nombrearchivo = $_FILES['foto']['name'];
    $directorio = 'imagenes/';
    $urlimagen= $directorio;
    if (move_uploaded_file($_FILES['foto']['tmp_name'],
$directorio.$nombrearchivo))
    {
        chmod($urlimagen, 0777);
        $fecha= date("Y-m-d");
    }
    else
    {
        echo "La imagen no ha podido moverse de directorio";
    }
}
else
{
    echo "La imagen no ha podido subirse al servidor";
}

$tiponoticia = $_POST['tiponoticia'];
$texto_noticia = $_POST['texto_noticia'];
$activar = $_POST['activar'];
$titulo_noticia = $_POST['resumen_noticia'];
$fecha= date("Y-m-d");

$link=mysql_connect($server,$dbuser,$dbpass);
$query="INSERT INTO noticias (titulo, texto, FECHA, ACTIVAR, tipo_noticia, imagen)
VALUES ('$titulo_noticia','$texto_noticia','$fecha',b'$activar','$tiponoticia',
'$nombrearchivo')";
mysql_db_query($database,$query,$link);
header("location:lector_noticias_formateado.php");
?>
```

include("config.php"); contiene los datos del servidor, BD, usuario y contraseña
if(is_uploaded_file(\$_FILES['foto']['tmp_name'])) Si la imagen ha sido subida correctamente...

la funcion *is_uploaded_file* permite comprobar si la imagen seleccionada en el formulario y enviada se ha alojado en el servidor.

La imagen subida se guarda en *\$_FILES['foto']* foto es el nombre que le damos en el `<input name= 'foto' ['tmp_name']` tmp_name es el directorio temporal donde se almacena esta imagen. Este directorio se va borrando cada cierto tiempo, por eso es importante mover la imagen a otro directorio para no perderla. También aprenderemos a hacerlo en este mismo código.

{ ... haremos

\$nombrearchivo = \$_FILES['foto']['name']; en **\$_FILES['foto']['name']**

guardamos el nombre del archivo (solo el nombre, no el archivo en si). Por lo tanto guardaremos ese nombre en una variable para hacer mas facil su llamada y

insertarla posteriormente en la base de datos.

\$directorio = 'imagenes/'; Creamos también otra variable con el directorio donde guardaremos la imagen. Es una dirección relativa. Trabajar siempre con direcciones relativas, porque podéis tener problemas de permisos a la hora de acceder al directorio.

if (move_uploaded_file(\$_FILES['foto']['tmp_name'], \$directorio.\$archivo)) si la función *move_uploaded_file* devuelve "1", es decir se ejecuta correctamente... La función *move_uploaded_file* sirve para mover un archivo de un directorio a otro. Necesita directorio origen, directorio objetivo.

```

    {
        // haremos
        chmod($directorio, 0777); dar permisos de lectura al directorio
        $fecha= date("Y-m-d");
    }
    else
    {
        echo "La imagen no ha podido moverse de directorio";
    }
}
else
{
    echo "La imagen no ha podido subirse al servidor";
}

```

\$tiponoticia = \$_POST['tiponoticia'];

\$texto_noticia = \$_POST['texto_noticia'];

\$activar = \$_POST['activar'];

Atención a esta variable que vale 1 o 0 según el usuario envía activar o no activar.

\$titulo_noticia = \$_POST['resumen_noticia'];

\$fecha= date("Y-m-d");

Guardamos los datos de cada apartado del formulario en diferentes variables

\$link=mysql_connect(\$server,\$dbuser,\$dbpass);

\$query="INSERT INTO noticias (titulo, texto, FECHA, ACTIVAR, tipo_noticia, imagen) VALUES el campo **ACTIVAR** de la base de datos es del tipo BIT(1), por lo tanto solo acepta 1 valor en binario (BIT en 1 o BIT en 0). Por esta razón hemos de transformar el valor decimal, que pasamos por el formulario, a valor binario. Como se hace?... En la línea de abajo vemos como al pasar la variable *\$activar* le ponemos delante la letra b y se la enviamos así: *b'\$activar'*. La letra b convierte el valor de la variable a valor decimal. Teníamos 1 en decimal y ahora pasamos a tener 1 en valor binario.

(' \$titulo_noticia', '\$texto_noticia', '\$fecha', b'\$activar', '\$tiponoticia', '\$archivo'))';

mysql_db_query(\$database,\$query,\$link);

Estos apartados ya los hemos dado en anteriores tutoriales.

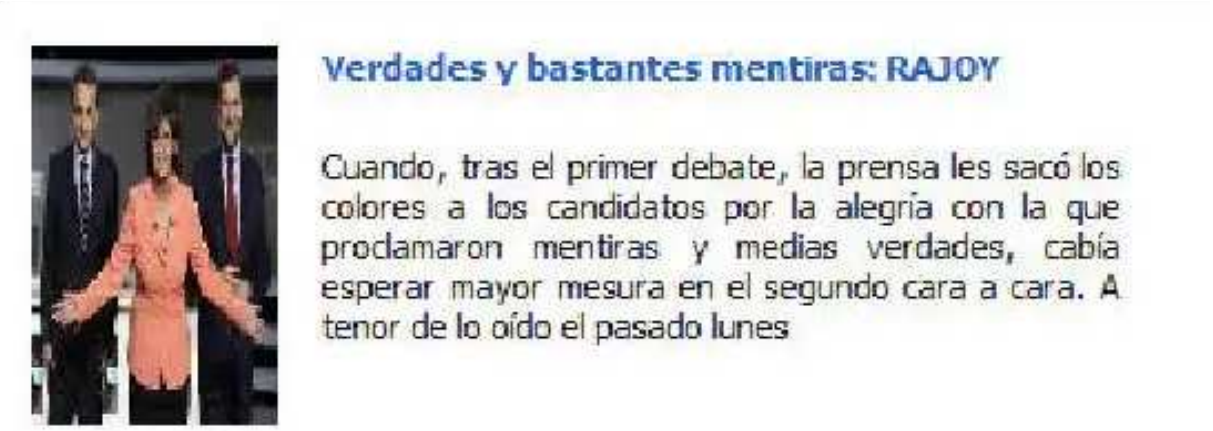
header("location:lector_noticias_formateado.php"); Redireccionamos la pagina a lector_noticias_formateado.php

?>

A continuación crearemos la última pagina que se encargará de extraer los datos de la base de datos y además darle un formato visual atractivo gracias a Cascading Style Sheet (CSS).

Yo recomiendo (y es lo más sensato) crear el la pagina web en HTML y CSS que contendrá la noticias sin cargar los datos de la base de datos.

Debemos crear algo parecido a esto.



Vamos ahora a crear este formato de noticia con HTML y los estilos CSS.

lector_de_noticias.html

```
<head>
<link href="Estilos/estilo_noticia.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="listing">
    <div id="img_cont_1">
        <div id="image_left">
            
        </div>
    </div>
    <h1>Verdades y bastantes mentiras: RAJOY</h1>
    <p>Cuando, tras el primer debate, la prensa les sacó los colores a los
candidatos por la alegría con la que proclamaron mentiras y medias verdades, cabía
esperar mayor medida en el segundo cara a cara. A tenor de lo oído el pasado
lunes.</p>
</div>
```

estilo_noticia.css

```
#listing {border:0px solid #CCCCCC;font-family:tahoma;
font-size:11px;font-size-adjust:none;font-stretch:normal;
font-style:normal;font-variant:normal;font-weight:normal;
height:147px;line-height:normal;margin:10px 0px 0px 5px;
padding:5px 5px 5px 0px;width:355px;
}
#listing h1 {color:#1A5DCD;font-size:100%;margin:0px;
padding:5px 0px 0px;text-align:justify;
}
#img_cont_1 {font-size:85%;line-height:50px;padding:5px 0px 0px;
}
#image_left {float:left;margin:5px 15px 0px 0px;
}
```

No voy a desglosar el código de HTML y CSS porque no es el objetivo de este manual. Entender como funciona CSS es sencillo, pero llegar a entender profundamente todo el potencial que nos podría proporcionar este lenguaje es difícil y requiere mucha práctica.

Una recomendación si me permitís: Si eres ya un iniciado en los estilos CSS te recomiendo encarecidamente que te descargues un plugin para firefox que se llama Firebug. Te servirá para ver los estilos CSS, el código HTML, etc. que utilizan las paginas por las que navegues. Así si encuentras algún menú, algún bloque de noticias que te gusta, etc. puedes mirar su código y copiarlo. Es una herramienta esencial para un diseñador.

Bueno, una vez escrito el código HTML que mostrara una noticia de forma extática formateada con los estilos, vamos a aprender como añadiéndole código PHP esta pagina será capaz de hacer peticiones a la base de datos y de ahí sacar las noticias. Convirtiéndose así en una pagina Web dinámica.

Antes de añadir el código PHP a la pagina HTML, crearemos primero el código PHP por separado y luego ya lo insertaremos dentro del HTML.

Nuestro código PHP deberá conectarse a la Base de datos LIBRERÍA, hacer una petición de consulta a la tabla NOTICIAS en lenguaje SQL, guardar la consulta que hemos hecho en una matriz donde estén todas las filas de la base de datos que le hemos pedido en la consulta y para acabar presentar todas las noticias que tengamos almacenadas en la matriz, haciendo un bucle.

lector_noticias.php

```
<?
include("config.php");

$link=mysql_connect($server,$dbuser,$dbpass) or die("Error en la conexión a
MySQL");
$query="SELECT * FROM `noticias` where `tipo_noticia`='Novedades' and
`ACTIVAR` = b'1'";
$resultado=mysql_db_query($database,$query,$link);
$row=mysql_fetch_row($resultado);
while($row=mysql_fetch_array($resultado))
{
    echo"imagenes/".$row["imagen"];
    print(htmlentities($row["titulo"])."<BR>");
    print(htmlentities($row["texto"])."<BR>");
}
?>
```

El código que veis arriba extrae el titulo, el texto y el nombre de la imagen de la noticia de la Base de Datos.

Empecemos a desglosar el código.

```
<?
include("config.php");
$link=mysql_connect($server,$dbuser,$dbpass) or die("Error en la conexión a
MySQL");
$query="SELECT * FROM `noticias` where `tipo_noticia`='Novedades' and
`ACTIVAR` = b'1'";
```

Selecciona de la tabla noticias las filas que cumplan: que el campo `tipo_noticia` sea igual a Novedades y `ACTIVAR` tenga un valor de BIT en 1

```
$resultado=mysql_db_query($database,$query,$link);
```

esta función ya la vimos en anteriores capitulos. Hacemos la consulta a la Base de datos y la almacenamos en una variable (\$resultado).

```
while($row=mysql_fetch_array($resultado))
```

Nos traemos todas las filas que coinciden con la consulta a la Base de Datos y las guardaremos en un array. Entiendo que es complejo entender esto. Lo que tenemos que tener claro que este tipo de sentencia nos permite almacenar las filas que coinciden con la consulta en

una matriz. Y el while o utilizamos para hacer un bucle mientras se cumpla una condición. While (condición que debe cumplirse).

```
{
    echo"imagenes/".$row["imagen"]; llamamos al contenido de la base de
    datos que esta almacenado en el campo "imagen".
    print(htmlentities($row["titulo"])."<BR>");
    print(htmlentities($row["texto"])."<BR>");
}
?>
```

Ahora llega el momento de introducir el pequeño código PHP dentro del código HTML.
Vemos el ejemplo.

le ctor_noticias_formateado.php

```
<head>
<link href="Estilos/estilo_noticia.css" rel="stylesheet" type="text/css" />
</head>
<body>
<?
include("config.php");
$link=mysql_connect($server,$dbuser,$dbpass) or die("Error en la conexión a
MySQL");
$query="SELECT * FROM `noticias` where `tipo_noticia`='Novedades' and
`ACTIVAR` = b'1'";
$resultado=mysql_db_query($database,$query,$link);
$row=mysql_fetch_row($resultado);
while($row=mysql_fetch_array($resultado))
{
?>
<div id="listing">
    <div id="img_cont_1">
        <div id="image_left">
"/>
        </div>
        </div>
        <h1><? print(htmlentities($row["titulo"]); ?></h1>
        <p>><? print(htmlentities($row["texto"]); ?></p>

    <? >?>
</div>
```

Seguiremos teniendo la hoja de estilos asociada a nuestra pagina lector_noticias_formateado.php

estilo_noticia.css

```
#listing {border:0px solid #CCCCCC;font-family:tahoma;
font-size:11px;font-size-adjust:none;font-stretch:normal;
font-style:normal;font-variant:normal;font-weight:normal;
height:147px;line-height:normal;margin:10px 0px 0px 5px;
padding:5px 5px 5px 0px;width:355px;
}
#listing h1 {color:#1A5DCD;font-size:100%;margin:0px;
padding:5px 0px 0px;text-align:justify;
}
#img_cont_1 {font-size:85%;line-height:50px;padding:5px 0px 0px;
}
#image_left {float:left;margin:5px 15px 0px 0px;
}
```

Puede parecer un follon a simple vista, pero no es así. Explicamos el código final de **lector_noticias_formateado.php**

```
<head>
<link href="Estilos/estilo_noticia.css" rel="stylesheet" type="text/css" />
</head>
```

Dentro de la etiqueta `body` comenzamos a insertar el código PHP

```
<?
include("config.php");
$link=mysql_connect($server,$dbuser,$dbpass) or die("Error en la conexión a
MySQL");
$query="SELECT * FROM `noticias` where `tipo_noticia`='Novedades' and
`ACTIVAR` = b'1'";
$resultado=mysql_db_query($database,$query,$link);
$row=mysql_fetch_row($resultado);
while($row=mysql_fetch_array($resultado))
{
    Toda esta parte del código es invisible para el navegador porque esta entre
    etiquetas <?> que engloban al lenguaje php y porque no hay ninguna función
    echo o print que imprimirían algo por pantalla.
}
?>
<div id="listing">
    <div id="img_cont_1">
        <div id="image_left">
            "/>
        </div>
    </div>
    <h1><? print(htmlentities($row["titulo"]); ?></h1>
    <p><? print(htmlentities($row["texto"]); ?></p>
}
?>
```

Hemos acabado con este capítulo y creo que el resultado es positivo, nos estamos embarcando en el aprendizaje PHP a través de ejemplos reales. Un saludo a todos.