

Variable: \$ (no puede empezar por número y no puede contener símbolos extraños).
Las sentencias terminan en punto y coma;
Los strings van entre comillas. Entre comillas dobles también se sustituyen las variables por sus valores, en las simples no.

echo \$nombre,\$edad ; (Esto no lo admitiría print). Print es una función que devuelve y echo una expresión. Por lo que a print le lleva más tiempo que a echo, consume más recursos para mostrar información en pantalla. Por lo que normalmente se usa echo y no print.

Crear una función: function dameDatos(){
 echo "Este es el mensaje del interior de la función
";
}

Llamar a esa función: dameDatos();

include ("NombreDelArchivo.php"); // Es como si el código de ese archivo externo estuviera en nuestro documento. Se consigue modular, escalar el código. Funciones que se reutilizarán en otros archivos se pondrán en un archivo destinado a tal fin, ser reutilizado con la instrucción include.

require: se diferencia de include en que el include permite continuar con el código aunque no exista o haya cualquier problema con el archivo que se quiere incluir. El require no permitirá mostrar nada si hay un error en esa instrucción.

Ámbito: Local / Global /Super Global

Local: sólo dentro de la función.

Global: Visible y accesible desde cualquier lugar del código

Super global: Declarada como Array. Visible y accesible desde fuera del script PHP.

No funciona :

```
$nombre = "Juan";  
function dameNombre(){  
    $nombre = "El nombre es ".$nombre;  
    dameNombre();  
    echo $nombre;
```

Si funciona :

```
$nombre = "Juan";  
function dameNombre(){  
    global $nombre; //Esta instrucción de variable global siempre dentro de la función, fuera no  
    //funciona. No se utiliza tampoco mucho por el paso de parámetros.  
    $nombre = "El nombre es ".$nombre;  
    dameNombre();  
    echo $nombre;
```

Se podría sustituir por algo así:

```
$nombre= "Juan";  
function dameNombre($parametro){  
    $parametro= " El nombre es ".$parametro;  
    echo $parametro;  
}  
dameNombre($nombre);
```

Variables estáticas:

```
function DosEnDos(){
    static $duplicador=0;
    $duplicador=$duplicador+2;
    echo $duplicador."<br>";
}

DosEnDos(); //2
DosEnDos(); //4
DosEnDos(); //6
```

Nos permite una variable estática que el valor de la variable no se inicialice cada vez que llamamos a la función y además que se conserve y se pueda ir incrementando su valor. Resumiendo una variable estática conserva su valor al salir de su ámbito.

strcmp: compara dos strings y tiene en cuenta mayúsculas y minúsculas. Devuelve un 1 (verdadero) si no coinciden y un cero (falso) si coinciden.

strcasecmp: ignora las mayúsculas y minúsculas.

Operadores de comparación:

SINTAXIS	DEFINICIÓN	RESULTADO
\$var1==\$var2	Igual que	Verdadero si las dos variables son iguales
\$var1===\$var2	Idéntica que	Verdadero si las dos variables son iguales y del mismo tipo
\$var1!=\$var2	Diferente que	Verdadero si las dos variables no son iguales
\$var1<>\$var2	Diferente que	Verdadero si las dos variables no son iguales y además no son del mismo tipo
\$var1<\$var2	Menor que	Verdadero si \$var1 es menor que \$var2
\$var1>\$var2	Mayor que	Verdadero si \$var1 es mayor que \$var2
\$var1<=\$var2	Menor o igual que	Verdadero si \$var1 es menor o igual que \$var2
\$var1>=\$var2	Mayor o igual que	Verdadero si \$var1 es mayor que \$var2

Enviando formulario:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
<style>
    h1{
        text-align:center;
    }

    table{
        background-color:#FFC;
        padding:5px;
        border:#666 5px solid;
```

```

}

</style>
</head>

<body>
<h1>USANDO OPERADORES COMPARACIÓN</h1>

<form action="validacion.php" method="post" name="datos_usuario" id="datos_usuario">
<table width="15%" align="center">
<tr>
<td>Nombre:</td>
<td><label for="nombre_usuario"></label>
<input type="text" name="nombre_usuario" id="nombre_usuario"></td>
</tr>
<tr>
<td>Edad:</td>
<td><label for="edad_usuario"></label>
<input type="text" name="edad_usuario" id="edad_usuario"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" name="enviando" id="enviando" value="Enviar"></td>
</tr>
</table>
</form>

</body>
</html>

```

validacion.php:

```

<style>

.no_validado{
    font-size:18px;
    color:#F00;
    font-weight:bold;
}

.validado{
    font-size:18px;
    color:#0C3;
    font-weight:bold;
}

```

```

        }
</style>

<?php
    if(isset($_POST[“enviando”])){
        $usuario=$_POST[“nombre_usuario”];
        $edad=$_POST[“edad_usuario”];

        if($usuario==”Juan” && $edad>=18) echo “<p class='validado’>Puedes entrar</p>”;
        else echo “<p class='no_validado’>.”.”No puedes entrar”..”</p>”;
    }
?>
```

CONSTANTES:

`define(“NOMBRE_CONSTANTE”, valor); // Se puede añadir un tercer argumento, true para que funcione tanto si el nombre de la constante se pone en mayúsculas como en minúsculas.`

Ej: `define (“AUTOR”, “Juan”);
echo AUTOR;`

Esto no es posible: `echo “El autor es: AUTOR”;// A diferencia de lo que pasaba con las variables.
Sería así: echo “El autor es: “.AUTOR;`

El nombre de las constantes ha de ir en mayúsculas (convenio).

El nombre de las constantes no debe llevar el símbolo \$.

El ámbito de las constantes es global por defecto.

Las constantes no se pueden redefinir

Las constantes solo pueden almacenar valores escalares (no puede contener un array, por ejemplo).

Operadores Matemáticos:

OPERADOR	NOMBRE
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo
++ +=	Incremento
-- -=	Decremento

Condicionales:

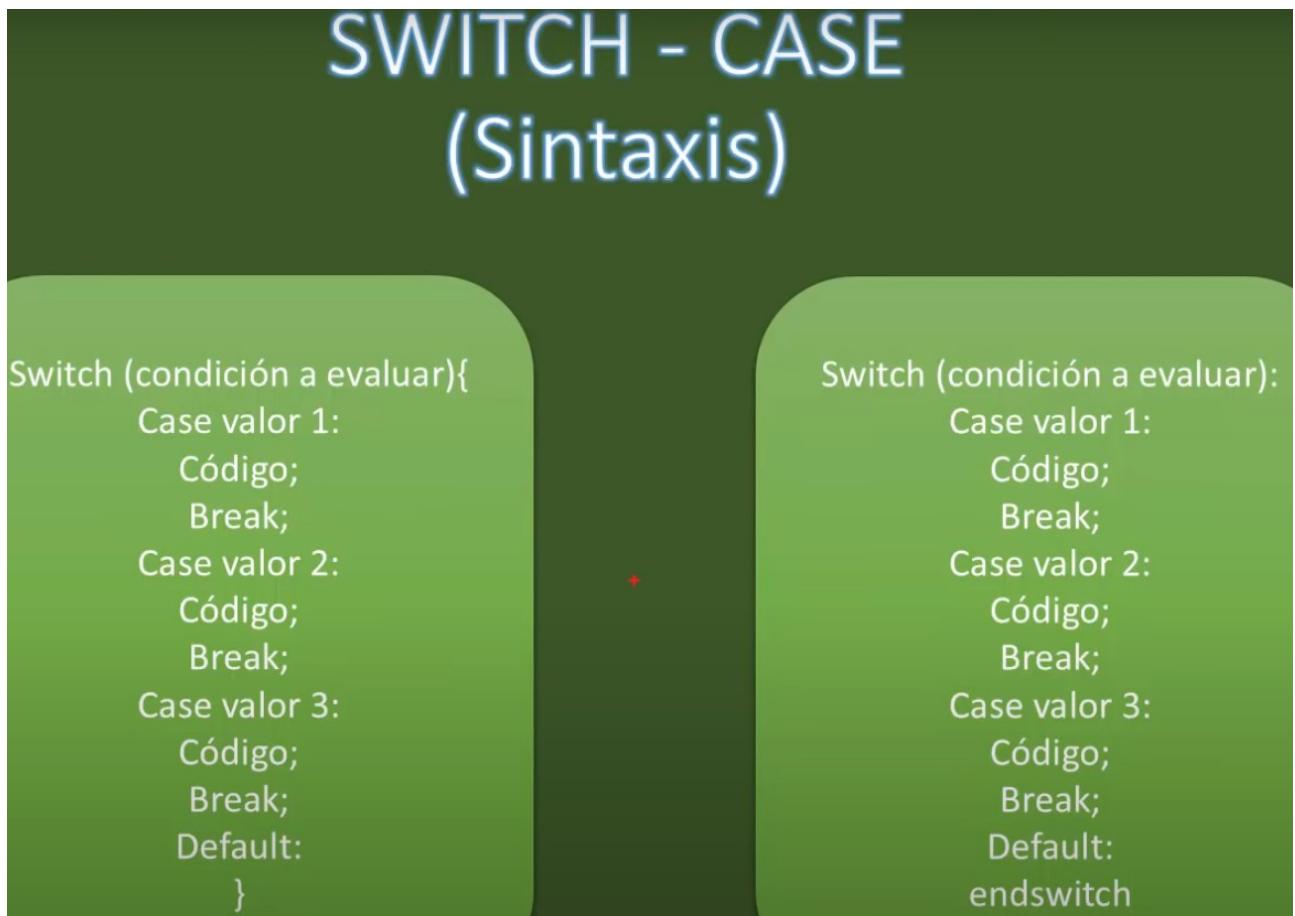
Ejemplo:

```
if(isset($_POST["enviando"])){
    $edad=$_POST["edad_usuario"];
    $resultado= $edad<18 ?true:false; //Operador ternario
    if($resultado) echo "No puedes acceder.<br/>";
    else if ($edad<110) echo "Puedes acceder.<br/>";

    if($resultado && $edad<0) echo "No existen edades negativas.";
    else if($resultado && $edad>=0) echo "Eres menor de edad.";
    else if(!$resultado && $edad<=40) echo "Eres joven.";
    else if(!$resultado && $edad<=65) echo "Eres maduro.";
    else if ($edad>65 && $edad<110)echo "Cúídate.";
    else echo "Has puesto una edad irreal.";

}
```

Switch:



En el endswitch falta un ; Sería: endswitch;

Paso de parámetros:

PARÁMETROS POR VALOR Y POR REFERENCIA



```
Function ejemplo ($param){  
    $param++;  
}
```



Paso de parámetro por valor

```
Function ejemplo (&$param){  
    $param++;  
}
```



Paso de parámetro por referencia

Paso por valor:

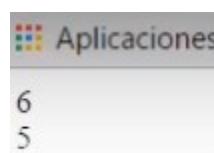
PHP MySQL. Funciones II. Parámetros por valor y por referencia. video 21

```
<?php
```

```
function incrementa($valor1){  
  
    $valor1++;  
  
    return $valor1;  
  
}  
  
$numero=5;  
  
echo incrementa($numero) . "<br>";  
  
echo $numero;  
??>  
  
</body>  
</html>
```

6
5 +

Resultado paso por valor:



Paso por referencia:

```
<?php
```

```
function incrementa(&$valor1){
```

```
    $valor1++;
```

6

```
    return $valor1;
```

```
}
```

```
$numero=5;
```

```
echo incrementa($numero) . "<br>";
```

```
echo $numero;
```

```
?>
```

5

referencia

Aplicaciones

6
6

Resultado paso por referencia:

PROGRAMACIÓN ORIENTADA A OBJETOS:

```
<?php

    class Coche{
        var $ruedas;
        var $color;
        var $motor;

        function Coche(){ //Método constructor;
            $this->ruedas=4;
            $this->color="";
            $this->motor=1600;
        }

        function arrancar(){

        }

        function girar(){

        }

        function frenar(){

        }

        function arrancar(){
            echo "Estoy arrancando";
        }

        function girar(){
            echo "Estoy girando";
        }

        function frenar(){
            echo "Estoy frenando";
        }
    }

    $renault=new Coche(); //Estado inicial al objeto o instancia
    $mazda=new Coche();
    $seat=new Coche();
```

Importante: ese constructor, llamado igual que la clase es un método obsoleto (deprecated), se usa como constructor el método __construct(), con doble guión bajo.

```
$mazda->girar(); //Salida: Estoy girando.  
echo $mazda->ruedas; //Salida: 4.
```

```
function establece_color($color_coche) {  
    $this->color=$color_coche;  
    echo "El color de este coche es: " . $this->color;  
}  
}  
}
```

```
$renault->establece_color("Rojo"); //Salida: El color de este coche es: Rojo.
```

```
function establece_color($color_coche, $nombre_coche) {  
    $this->color=$color_coche;  
    echo "El color de " . $nombre_coche . " es: " . $this->color . "<br>";  
}  
}  
  
$renault=new Coche(); //Estado inicial al objeto o instancia  
$mazda=new Coche();  
$seat=new Coche();  
$renault->establece_color("Rojo","Renault");  
$seat->establece_color("Azul","Seat");
```

```
<?php  
include("vehiculos.php");  
  
$mazda=new Coche();  
$pegaso=new Camion();  
  
echo "El Mazda tiene " . $mazda->ruedas . " ruedas <br>";  
echo "El Pegaso tiene " . $pegaso->ruedas . " ruedas <br>";  
  
?>  
</body>  
</html>
```

```

function arrancar () {
    parent::arrancar ();
    echo "Camión arrancado";
}

}

```



Get y Set:

```

private $ruedas;
var $color;
var $motor;

function Coche(){ //Método constructor;
    $this->ruedas=4;
    $this->color="";
    $this->motor=1600;
}

function get_ruedas(){
    return $this->ruedas;
}

function set_color($color_coche,$nombre_coche){
    $this->color=$color_coche;
    echo "El color de " . $nombre_coche . " es: " . $this->color . "<br>";
}

```

```

private static $ayuda=0;

static function descuento_gobierno() {
    if(date("m-d-y")>"05-01-15") {
        self::$ayuda=4500;
    }
}

function precio_final() {
    $valor_final=$this->precio_base-self::$ayuda;
    return $valor_final;
}// fin precio final

```

```

<?php

include("Concesionario.php");
//Compra_vehiculo::$ayuda=10000;
Compra_vehiculo::descuento_gobierno();
$compra_Antonio=new Compra_vehiculo("compacto");
$compra_Antonio->climatizador();
$compra_Antonio->tapiceria_cuero("blanco");
echo $compra_Antonio->precio_final() . "<br>";
$compra_Ana=new Compra_vehiculo("compacto");
$compra_Ana->climatizador();
$compra_Ana->tapiceria_cuero("rojo");
echo $compra_Ana->precio_final();

```

ARRAYS:

A) Arrays indexados:

Son equivalentes:

- 1) \$semana [] = "Lunes";
 \$semana [] = "Martes";
 \$semana [] = "Miércoles";
- 2) \$semana [0] = "Lunes";
 \$semana [1] = "Martes";
 \$semana [2] = "Miércoles";

Otra forma:

```
$semana=array("Lunes","Martes","Miércoles","Jueves");
```

B) Arrays Asociativos:

```
$datos=array("Nombre"=>"Juan","Apellidos"=>"Vidal de Sa","Edad"=>43);
```

Nota: En los arrays asociativos se utiliza el => para asignar valores, en cambio, para acceder a propiedades de objetos usamos algo parecido, pero no igual, lo siguiente: ->. En los arrays es un igual y en los objetos es un guión.

Función predefinida en PHP **is_array()**:

```
if (is_array($datos)) echo "Es un Array";  
else echo "No es un Array"; // Devuelve: "Es un Array".
```

Recorrer los elementos de un Array:

```
foreach ($datos as $clave=>$valor){ //para recorrer un array asociativo.
```

```
    echo "A $clave le corresponde $valor <br>";  
}
```

Para recorrer un array indexado, 3 formas:

```
$semana=array("Lunes","Martes","Miércoles","Jueves");
```

```
1)  
for ($i=0;$i<4;$i++){  
    echo $semana[$i]."<br/>";  
}
```

2) Cuando no sabemos el número de elementos de un array:
 for (\$i=0;\$i<count(\$semana);\$i++){

```
        echo $semana[$i]."<br/>";  
    }  
3)
```

```
foreach ($semana as $valor){  
    echo $valor."</br>";
```

}

Añadir un elemento a un array indexado: \$semana[]="Viernes";

Añadir un elemento a un array asociativo: \$datos["País"]="España";

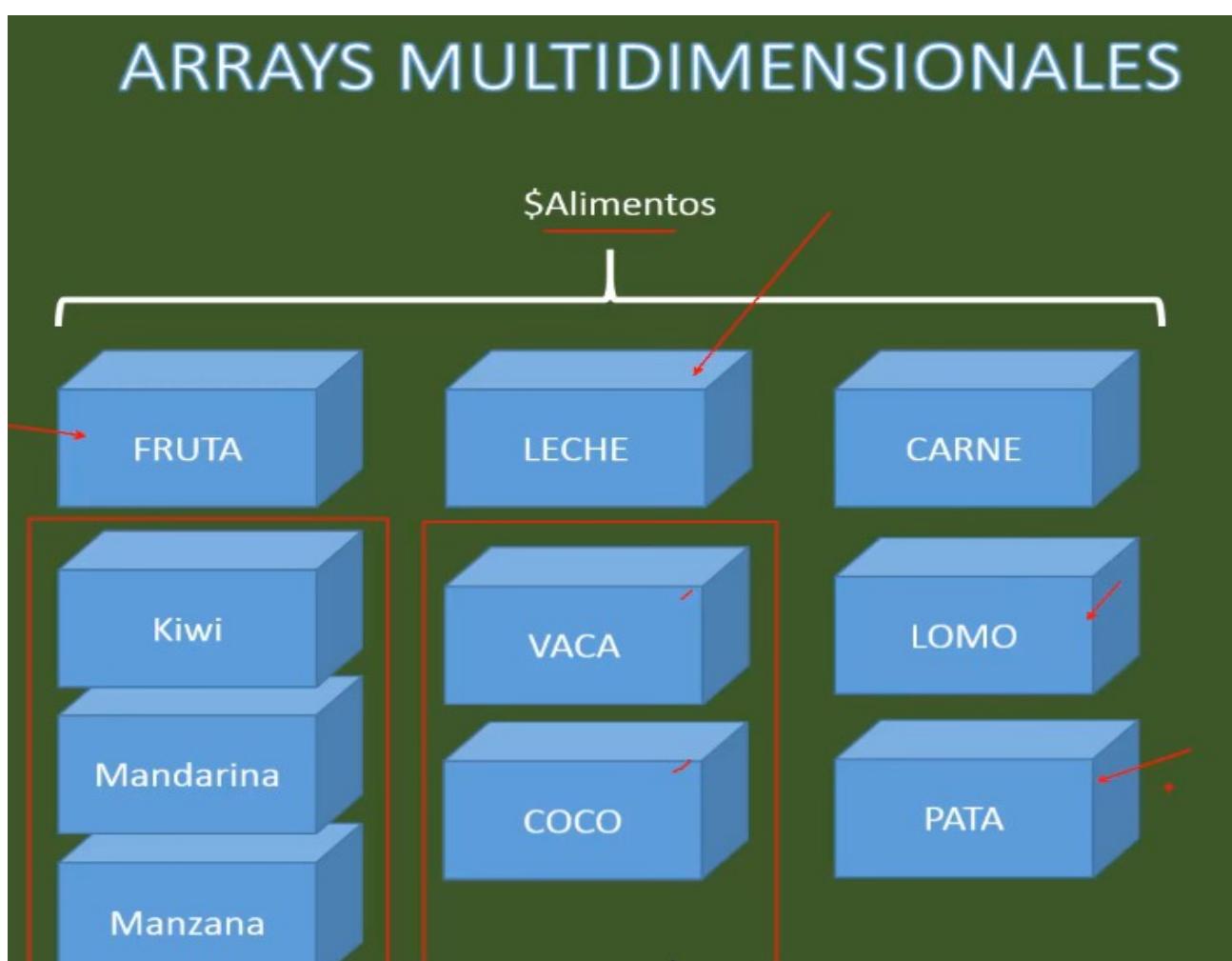
Ordenar un array:

```
$numeros=array(4,3,9,14,17,21,8,6);
sort ($numeros);
for ($i=0;$i<count($numeros);$i++){
    echo $numeros[$i]."<br/>";
}
```

Resultado:

```
3
4
6
8
9
14
17
21
```

ARRAY MULTIDIMENSIONALES (ARRAYS DENTRO DE UN ARRAY):



```

$alimentos=array("fruta"=>array("tropical"=>"kiwi",
                                  "cítrico"=>"mandarina",
                                  "otros"=>"manzana"),
                  "leche"=>array("animal"=>"vaca",
                                  "vegetal"=>"coco"),
                  "carne"=>array("vacuno"=>"lomo",
                                  "porcino"=>"pata"));

echo $alimentos["carne"]["vacuno"]; //imprime en pantalla lomo

```

//Mostrarlo en pantalla:

```

foreach ($alimentos as $clave=>$valor){
    echo "El contenido de $clave es: <br/>";
    foreach($valor as $clave2=>$valor2){
        echo "$clave2=$valor2 <br/>";
    }
    echo "<br>";
}

```

Él:

```

$alimentos=array("fruta"=>array("tropical"=>"kiwi",
                                  "cítrico"=>"mandarina",
                                  "otros"=>"manzana"),

                  "leche"=>array("animal"=>"vaca",
                                  "vegetal"=>"coco"),

                  "carne"=>array("vacuno"=>"lomo",
                                  "porcino"=>"pata"));

//echo $alimentos["carne"]["vacuno"];

```

```

foreach($alimentos as $clave_alim=>$alim) {

    echo "$clave_alim:<br>";

    while(list($clave,$valor)=each($alim)) {

        echo "$clave=$valor<br>";

    }
    echo "<br>";
}

```

Uso de la función var_dump():

```
var_dump($alimentos);
```

Salida:

```
C:\xampp\htdocs\html\Pildoras\2.php:52:  
array (size=3)  
  'fruta' =>  
    array (size=3)  
      'tropical' => string 'kiwi' (length=4)  
      'cítrico' => string 'mandarina' (length=9)  
      'otros' => string 'manzana' (length=7)  
  'leche' =>  
    array (size=2)  
      'animal' => string 'vaca' (length=4)  
      'vegetal' => string 'coco' (length=4)  
  'carne' =>  
    array (size=2)  
      'vacuno' => string 'lomo' (length=4)  
      'porcino' => string 'pata' (length=4)
```

MYSQL:

Conectarse al gestor de BBDD en local a través de consola:

La password por defecto de root es root si no se ha cambiado con anterioridad

```
c:\xampp\mysql\bin>mysql.exe -h localhost -u root -p  
Enter password: ****  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 83  
Server version: 10.1.22-MariaDB mariadb.org binary distribution  
  
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> create database probando;  
Query OK, 1 row affected (0.00 sec)  
  
MariaDB [(none)]> drop database probando;  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> show databases  
>;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'databse' at line 1  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| bombones |  
| rhues |  
| empleados |  
| empresa |  
| information_schema |  
| mysql |  
| new_schema |  
| new_schema2 |  
| performance_schema |  
| phonyadmin |  
| pinacoteca |  
| proyecto |  
| tenistas |  
| test |  
| viviendas |  
+-----+  
15 rows in set (0.00 sec)  
  
MariaDB [(none)]>
```

```
:01> mysql> CREATE DATABASE USUARIOS;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> USE USUARIOS;  
Database changed  
mysql> CREATE TABLE DATOSUSUARIOS (NOMBRE VARCHAR(30), CLAVE VARCHAR(10));  
Query OK, 0 rows affected (0.05 sec)  
  
mysql> _
```

mysql> DESCRIBE DATOSUSUARIOS;						
Field	Type	Null	Key	Default	Extra	
NOMBRE	varchar(30)	YES		NULL		
CLAVE	varchar(10)	YES		NULL		
2 rows in set (0.01 sec)						

Borrar un campo (columna) de una tabla:

```
1 | ALTER TABLE DATOSPERSONALES DROP EDAD;
```

Agregar un campo a una tabla:

```
1 | ALTER TABLE DATOSPERSONALES ADD COLUMN EDAD INT(2);
```

Agregar registros en una tabla:

```
INSERT INTO DATOSPERSONALES (NIF, NOMBRE, APELLIDO, EDAD) VALUES ("51972854W", "MARÍA", "GOMEZ", 27);
```

Consulta de selección:

```
SELECT NOMBRE, APELLIDO FROM DATOSPERSONALES
```

```
SELECT * FROM DATOSPERSONALES
```



Ejemplo importante vídeo 37:

```
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>
<body>
<?php

$db_host="localhost";
$db_usuario="root";
$db_contra="root";
$db_nombre="pinacoteca";

$conexion=mysqli_connect($db_host, $db_usuario,$db_contra/*,$db_nombre*/);

if (mysqli_connect_errno()){ // se ejecuta esa función cuando no hay éxito en la conexión a la BD.

    echo "Fallo al conectar con la base de datos";
    exit(); // la usamos para que no ejecute el resto del código.
}

mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");

mysqli_set_charset($conexion,"utf8"); //para que muestre bien las tildes, etc. (caracteres latinos);

$consulta="select * from pintor"; /* where npintor='Picasso' */;

$resultados=mysqli_query($conexion,$consulta);
$fila=mysqli_fetch_row($resultados);

foreach($resultados as $fila){
    foreach ($fila as $valor2){
        echo $valor2. " ";
    }
    echo "<br/>";
}

/*
while ($fila=mysqli_fetch_row($resultados)==true){
    echo $fila [0]." ";
    echo $fila [1]." ";
    echo $fila [2]." ";
    echo $fila [3]." ";
    echo "<br>";
}
*/

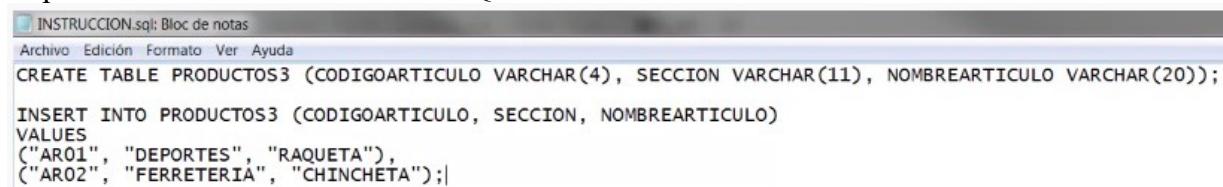
```

```
//cerrar conexión:  
  
mysqli_close($conexion);  
  
?>  
  
</body>  
</html>
```

Con formato y consulta más completa:

```
<style>  
    table, th, tr, td{  
        width: 50%;  
        border: 2px solid #FF0000;  
        margin:auto;  
    }  
  
    td {  
        color:blue;  
    }  
</style>  
</head>  
<body>  
<?php  
  
$db_host="localhost";  
$db_usuario="root";  
$db_contra="root";  
$db_nombre="pinacoteca";  
  
$conexion=mysqli_connect($db_host, $db_usuario,$db_contra/*,$db_nombre*/);  
  
if (mysqli_connect_errno()){ // se ejecuta esa función cuando no hay éxito en la conexión a la BD.  
    echo "Fallo al conectar con la base de datos";  
    exit(); // la usamos para que no ejecute el resto del código.  
}  
  
mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");  
  
mysqli_set_charset($conexion,"utf8"); //para que muestre bien las tildes, etc. (caracteres latinos);  
  
$consulta="select * from pintor where (PaisPintor='España' AND NPintorMaestro is null) order by CiudadPintor desc limit 3"; /* where npintor='Picasso'*/  
  
$resultados=mysqli_query($conexion,$consulta);  
$fila=mysqli_fetch_row($resultados);  
echo "<table>";  
echo "<tr><th>NPintor</th><th>CiudadPintor</th><th>PaisPintor</th><th>Fnac</th><th>Ffall</th><th>NombreE</th><th>NPintorMaestro</th></tr>";  
foreach($resultados as $fila){  
    echo "<tr>";  
    foreach ($fila as $valor2){  
        echo "<td>$valor2</td>";  
    }  
    echo "</tr>";  
}  
echo "</table>";
```

Importar datos a través de archivo SQL:



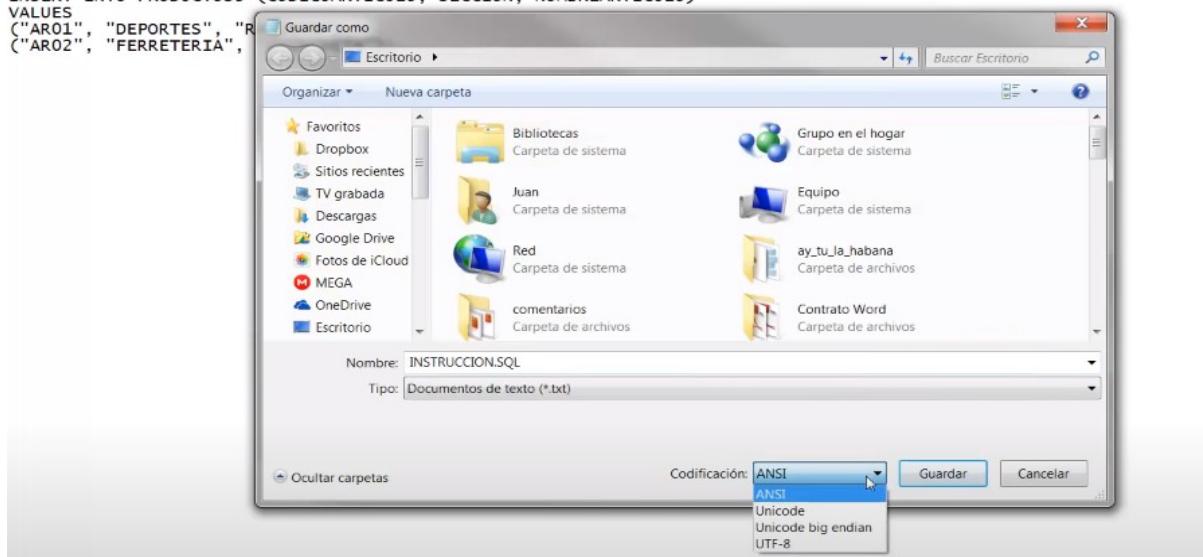
```
INSTRUCCION.sql: Bloc de notas  
Archivo Edición Formato Ver Ayuda  
CREATE TABLE PRODUCTOS3 (CODIGOARTICULO VARCHAR(4), SECCION VARCHAR(11), NOMBREARTICULO VARCHAR(20));  
INSERT INTO PRODUCTOS3 (CODIGOARTICULO, SECCION, NOMBREARTICULO)  
VALUES  
("AR01", "DEPORTES", "RAQUETA"),  
("AR02", "FERRETERIA", "CHINCHETA");|
```

Importante hay que guardar lo con la codificación que estamos usando, en este caso UTF8:

```

CREATE TABLE PRODUCTOS3 (CODIGOARTICULO VARCHAR(4), SECCION VARCHAR(11), NOMBREARTICULO VARCHAR(20));
INSERT INTO PRODUCTOS3 (CODIGOARTICULO, SECCION, NOMBREARTICULO)
VALUES
("AR01", "DEPORTES", "R"),
("AR02", "FERRETERIA", "S")

```



Importar una hoja de cálculo:

Importando en la tabla "pintor"

Archivo a importar:
 El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.
 Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**
 Buscar en su ordenador: No se ha seleccionado ningún archivo. (Máximo: 2,048KB)
 También puede arrastrar un archivo en cualquier página.
 Conjunto de caracteres del archivo:

Importación parcial:
 Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.)
 Omitir esta cantidad de consultas (en SQL) desde la primera:

Otras opciones:
 Habilita la revisión de las claves foráneas

Formato:

Opciones específicas al formato:

- La primer línea del archivo contiene los nombres de columna de la tabla (si no está activado la primera línea será parte de los datos)
- No importar filas vacías
- Importar porcentajes como decimales propios (por ejemplo: 12.00% como .12)
- Importar monedas (por ejemplo: \$5.00 como 5.00)

Hay que marcar la casilla “La primera línea del archivo...” cuando en la hoja de cálculo la primera fila es la de los nombres de las columnas.

Creando un formulario (3 ficheros: conexión, formulario y el que procesa el formulario):

1. Conexión:

```

Pildoras > 4conexion.php > ...
1  <?php
2
3  $db_host="localhost";
4  $db_usuario="root";
5  $db_contra="root";
6  $db_nombre="pinacoteca";
7  ?>

```

2. Formulario:

```
Pildoras > 40formulario_busqueda.php > html
1   <html>
2     <head>
3       <meta charset="utf-8">
4       <title>Vídeo 40</title>
5     </head>
6     <body>
7       <form action="40pagina_busqueda.php" method="get">
8         <label>Buscar: <input type="text" name="buscar"></label>
9         <input type="submit" name="enviando" value="Dale!">
10      </form>
11    </body>
12 </html>
```

3 archivo php:

```
?php

$buscar=$_GET["buscar"];
require("40conexion.php");
$conexion=mysqli_connect($db_host, $db_usuario,$db_contra/*,$db_nombre*/);

if (mysqli_connect_errno()){ // se ejecuta esa función cuando no hay éxito en la conexión a la BD.

    echo "Falló al conectar con la base de datos";
    exit();
}

mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");

mysqli_set_charset($conexion,"utf8");


$consulta="select * from cuadro where Npintor like '%$buscar%'";

$resultados=mysqli_query($conexion,$consulta);

$fila=mysqli_fetch_array($resultados);

echo "<table>";
echo "<tr><th>Ncuadro</th><th>Tecnica</th><th>Npintor</th><th>Npinacoteca</th></tr>";
foreach($resultados as $clave=> $fila){
    echo "<tr>";
    echo "<td>".$fila['Ncuadro']."</td>";
    echo "<td>".$fila['Tecnica']."</td>";
    echo "<td>".$fila['NPintor']."</td>";
    echo "<td>".$fila['NPinacoteca']."</td>";

    echo "</tr>";
}

echo "</table>";


mysqli_close($conexion);
?>
```

Formulario que llama a si mismo:

```
<?php

function ejecuta_consulta($labusqueda){

if (strlen($labusqueda)>=3){

require("40conexion.php");
$conexion=mysqli_connect($db_host, $db_usuario,$db_contra);

if (mysqli_connect_errno()){
echo "Fallo al conectar con la base de datos";
exit();
}

mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");

mysqli_set_charset($conexion,"utf8");

$consulta="select * from cuadro where Npintor like '%$labusqueda%'";
$resultados=mysqli_query($conexion,$consulta);

$fila=mysqli_fetch_array($resultados);

echo "<table>";
echo "<tr><th>Ncuadro</th><th>Tecnica</th><th>Npintor</th><th>Npinacoteca</th></tr>";
foreach($resultados as $clave=> $fila){
echo "<tr>";
echo "<td>".$fila['Ncuadro']."</td>";
echo "<td>".$fila['Tecnica']."'</td>";
echo "<td>".$fila['Npintor']."'</td>";
echo "<td>".$fila['NPinacoteca']."'</td>";

echo "</tr>";
}
echo "</table>";
mysqli_close($conexion);

} else {
echo "Ponte las pilas, escribe al menos pasa 3 caracteres para poder hacer la búsqueda";
}
}

?>
</head>
```

```
</body>
</html>

<?php
$buscar=$_GET[ 'buscar' ];
$mipagina=$_SERVER["PHP_SELF"];
//if($buscar!=NULL){ también valdría
if(isset($buscar)){
ejecuta_consulta($buscar);
} else{
echo "<form action='".$mipagina' method='get'>
<label>Buscar:<input type='text' name='buscar'></label>
<input type='submit' name='enviando' value='Dale!'>
</form>";
}
?>
```

Insertar valores en una BD (vídeo 43):

```
<?php
function inserta_valores($codigo, $cuadro,$pintor,$pinacoteca){

require("40conexion.php");
$conexion=mysqli_connect($db_host, $db_usuario,$db_contra);

if (mysqli_connect_errno()){
    echo "Fallo al conectar con la base de datos";
    exit();
}
mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");

mysqli_set_charset($conexion,"utf8");

$consulta="insert into cuadro (CodCuadro,Ncuadro, NPintor,NPinacoteca)
values ($codigo, '$cuadro', '$pintor', '$pinacoteca')";

if (mysqli_query($conexion,$consulta)) echo "Registro añadido adecuadamente";
else{
    echo "Ha habido un error<br/>";
    echo mysqli_error($conexion);
}
mysqli_close($conexion);
}
?>
</head>
<body>
<?php
$mipagina=$_SERVER["PHP_SELF"];

if(isset($_GET['enviando'])){
$pintor=$_GET['pintor'];
$pinacoteca=$_GET['pinacoteca'];
$codigo=$_GET['codigo'];

inserta_valores($codigo, $cuadro,$pintor,$pinacoteca);
}else if(isset($_GET['borrar']) || !isset($_GET['enviando'])){

echo "<form action='$mipagina' method='get'>
<label>Codigo:<input type='text' name='codigo'></label><br/>
<label>Cuadro:<input type='text' name='cuadro'></label><br/>
<label>Pintor:<input type='text' name='pintor'></label><br/>
<label>Pinacoteca:<input type='text' name='pinacoteca'></label><br/>
<input type='submit' name='enviando' value='Dale!'>
<input type='submit' name='borrar' value='Borrar'>
</form>";
}

?>
```

Eliminar valores:

```
<html>
<head>
<style>
table, th, tr, td{
```

```

        border: 2px solid #FF0000;
        margin: auto;
    }

    td {
        color: blue;
    }

</style>
<meta charset="utf-8">
<title>Vídeo 44</title>

<?php
function borra_valor($codigo){

require("40conexion.php");
$conexion=mysqli_connect($db_host, $db_usuario,$db_contra);

if (mysqli_connect_errno()){
    echo "Fallo al conectar con la base de datos";
    exit();
}
mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");

mysqli_set_charset($conexion,"utf8");
$consulta2="select CodCuadro, Ncuadro, NPintor, NPinacoteca from cuadro where CodCuadro=$codigo";
$resultados2=mysqli_query($conexion,$consulta2);
$fila2=mysqli_fetch_row($resultados2);
foreach($fila2 as $valor2){
    $codigoEnBD= $fila2[0];
    $cuadro= $fila2[1];
    $pintor= $fila2[2];
    $pinacoteca= $fila2[3];
}

/*
Se podría hacer también con la función mysqli_fetch_array, sería el siguiente código:
$ fila2=mysqli_fetch_array($resultados2);
foreach($resultados2 as $clave=> $fila){
    $codigo= $fila2['CodCuadro'];
    $cuadro= $fila2['Ncuadro'];
    $pintor= $fila2['NPintor'];
    $pinacoteca= $fila2['NPinacoteca'];
}
*/
$consulta="delete from cuadro where CodCuadro=$codigo";
mysqli_query($conexion,$consulta);

if (mysqli_affected_rows($conexion)>0){

```

```
/*if (!is_null($cuadro)){ // Para que falle cuando queremos eliminar un registro  
con un código de cuadro que no existe podemos poner  
// dentro del is_null cualquier variable que definimos en el foreach, porque ni  
ninguna se podría asignar porque la consulta nos daría cero filas.  
// en cambio si en el is_null ponemos el las variables consulta o resultado, éstas,  
aunque no dan ningún valor, tampoco son falsas, ni nulas, son arrays vacíos.  
//Por lo que no nos valdrían, así como tampoco nos valdría como condición mysql  
query($conexion,$consulta) como habíamos hecho en otros ejercicios, por  
//una cuestión similar, la condición no es falsa aunque no nos de ningún valor,  
afecta a cero filas, pero eso no es equivalente a que la consulta sea errónea/falsa  
. //La forma correcta de hacerlo sería con mysqli_affected_rows($conexion);*/  
  
echo "Registro eliminado adecuadamente<br/>";  
echo "El registro siguiente: <br/>";  
echo " Código: $codigoEnBD<br/>";  
echo " Cuadro: $cuadro<br/>";  
echo " Pintor: $pintor<br/>";  
echo " Pinacoteca: $pinacoteca<br/>";  
echo mysqli_affected_rows($conexion)." registros eliminados";
```

```
?>  
} else{  
    echo "Ha habido un error<br/>";  
    echo mysqli_error($conexion);  
}  
mysqli_close($conexion);  
}  
?  
</head>  
<body>  
<?php  
$mipagina=$_SERVER["PHP_SELF"];  
  
if(isset($_GET['enviando'])){  
    $codigo=$_GET['codigo'];  
    borra_valor($codigo);  
} else if(isset($_GET['borrar']) || !isset($_GET['enviando'])){  
  
    echo "<form action='".$mipagina' method='get'>  
        <label>Codigo:<input type='text' name='codigo'></label><br/>  
        <input type='submit' name='enviando' value='Dale!'>  
        <input type='submit' name='borrar' value='Borrar'>  
    </form>";  
}  
?  
</body>  
</html>
```

Actualizar valores: 2 ficheros

```
<html>
<head>
<style>
    table, th, tr, td{
        border: 2px solid #FF0000;
        margin:auto;
    }

    td {
        color:blue;
    }
</style>
<meta charset="utf-8">
<title>Vídeo 44</title>

<?php

function muestra_valor($codigo){

    require("40conexion.php");
    $conexion=mysqli_connect($db_host, $db_usuario,$db_contra);

    if (mysqli_connect_errno()){
        echo "Fallo al conectar con la base de datos";
        exit();
    }
    mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");

    mysqli_set_charset($conexion,"utf8");
    $consulta2="select CodCuadro, Ncuadro, NPintor, NPinacoteca from cuadro where CodCuadro=$codigo";
    $resultados2=mysqli_query($conexion,$consulta2);
    $fila2=mysqli_fetch_row($resultados2);

    echo "<form action='45actualizar2.php' method='get'>
        <label>Codigo:<input type='text' readonly='readonly' name='codigo2' value=$fila2[0]></label><br/>
        <label>Cuadro:<input type='text' name='cuadro' value='$fila2[1]'></label><br/>
    >
        <label>Pintor:<input type='text' name='pintor' value='$fila2[2]'></label><br/>
        <label>Pinacoteca:<input type='text' name='pinacoteca' value='$fila2[3]'></label><br/>
        <input style='visibility: hidden' type='text' name='antiguo0' value='$fila2[0]'><br/>
    </form>"
```

```

        <input style='visibility: hidden' type='text' name='antiguo1' value='$fila2[1]
'><br/>
        <input style='visibility: hidden' type='text' name='antiguo2' value='$fila2[2]
'><br/>
        <input style='visibility: hidden' type='text' name='antiguo3' value='$fila2[3]
'><br/>
        <input type='submit' name='enviando2' value='Actualizar'>
        <input type='submit' name='borrar2' value='Borrar'>

    </form>";

```



```

mysqli_close($conexion);
}

?>
</head>
<body>
<?php
$mipagina=$_SERVER["PHP_SELF"];

if(isset($_GET['enviando'])){
$codigo=$_GET['codigo'];
muestra_valor($codigo);
}else if(isset($_GET['borrar']) || !isset($_GET['enviando'])){

echo "<form action='$mipagina' method='get'>
        <label>Codigo:<input type='text' name='codigo'></label><br/>
        <input type='submit' name='enviando' value='Muestra datos'>
        <input type='submit' name='borrar' value='Borrar'>
    </form>";
}
?>
</body>
</html>

```

2º fichero:

```

<?php
if (!empty($_GET['borrar2'])) {

$antiguo= array($_GET['antiguo0'],$_GET['antiguo1'], $_GET['antiguo2'],
$_GET['antiguo3']);
echo "<form action='45actualizar2.php' method='get'>
        <label>Codigo:<input type='text' readonly='readonly' name='codigo2' value='".
$_GET['codigo2']."'></label><br/>

```

```

<label>Cuadro:<input type='text' name='cuadro' value=''></label><br/>
<label>Pintor:<input type='text' name='pintor' value=''></label><br/>
<label>Pinacoteca:<input type='text' name='pinacoteca' value=''></label><br/>
<input style='visibility: hidden' type='text' name='antiguo0' value=$antiguo[0]><br/>
<input style='visibility: hidden' type='text' name='antiguo1' value=$antiguo[1]><br/>
<input style='visibility: hidden' type='text' name='antiguo2' value=$antiguo[2]><br/>
<input style='visibility: hidden' type='text' name='antiguo3' value=$antiguo[3]><br/>
<input type='submit' name='enviando2' value='Actualizar'>
<input type='button' name='borrar2' value='Borrar'>

</form>";

```

```

}
else{
    $antiguo= array($_GET['antiguo0'],$_GET['antiguo1'], $_GET['antiguo2'],
$_GET['antiguo3'] );
    echo "Registro anterior: <br/> Código: $antiguo[0] <br/> Cuadro:$antiguo[1] <br/>
Pintor: $antiguo[2] <br/>Pinacoteca: $antiguo[3] <br/><br/> ";
    $array=array($_GET['codigo2'],$_GET['cuadro'], $_GET['pintor'],
$_GET['pinacoteca'] );
    require("40conexion.php");
    $conexion=mysqli_connect($db_host, $db_usuario,$db_contra);
    if (mysqli_connect_errno()){
        echo "Fallo al conectar con la base de datos";
        exit();
    }
    mysqli_select_db($conexion,$db_nombre) or die("No se encuentra la BBDD");

    mysqli_set_charset($conexion,"utf8");
$consulta="update cuadro set CodCuadro=$array[0], Ncuadro='$array[1]', NPintor='$array[2]', NPinacoteca='$array[3]' where CodCuadro=$array[0]";
if (mysqli_query($conexion,$consulta)){
    echo "Registro actualizado adecuadamente<br/>";
    echo "Registro actual: <br/> Código: $array[0] <br/> Cuadro:$array[1] <br/> Pinto
r: $array[2] <br/>Pinacoteca: $array[3] ";

}
else{
    echo "Ha habido un error<br/>";
    echo mysqli_error($conexion);
}

mysqli_close($conexion);
?>
```

Inyección Sql:

```
SELECT USUARIO, PASSWORD, DIRECCION, TELEFONO FROM DATOSPERSONALES WHERE USUARIO='SBUSQUEDA' OR 'Z'='Z';
```



Como z=z mostrará los datos de todos los usuarios y no sólo de un usuario.

Otro ejemplo:

Usuario: Paco
Contraseña: ' or '1'='1

```
SELECT * FROM USUARIOS WHERE USUARIO = 'Paco' AND CONTRA= " or '1'='1'
```

- **¿Cómo evitar la inyección SQL?**
 - Uso de la función mysqli_real_escape_string()
 - Uso de la función mysqli_addslashes()
 - Consultas preparadas

DELETE * FROM USUARIOS....

```
<?php
```

```
require ("datos_conexion.php");

$conexion=mysqli_connect($db_host, $db_usuario, $db_contra);

$usuario = mysqli_real_escape_string($conexion, $_GET["usu"]);
$contra = mysqli_real_escape_string($conexion, $_GET["con"]);

if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion, "utf8");

//Consulta="SELECT * FROM PRODUCTOS WHERE NOMBREARTICULO LIKE '%$busqueda%'";
$consulta="DELETE FROM USUARIOS WHERE USUARIO = '$usuario' AND CONTRA= '$contra';

echo "$consulta<br><br>";

if(mysqli_query($conexion, $consulta)){
    echo "Baja procesada";
}

mysqli_close($conexion);

?>
```

Cambiaremos el final de la captura anterior para que no nos diga baja procesada cuando no se elimina ningún usuario usando mysqli_affected_rows();:

```
mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");

mysqli_set_charset($conexion, "utf8");

//$consulta="SELECT * FROM PRODUCTOS WHERE NOMBREARTICULO LIKE '%$busqueda%'";
$consulta="DELETE FROM USUARIOS WHERE USUARIO = '$usuario' AND CONTRA= '$contraseña';

echo "$consulta<br><br>";

mysqli_query($conexion, $consulta);

if(mysqli_affected_rows($conexion)>0) {
    echo "Baja procesada";
}

} else{
    echo "No se ha encontrado usuario";
}

/*if(mysqli_query($conexion, $consulta)) {
    echo "Baja procesada";
}*/

mysqli_close($conexion);
```

La otra función no evita que se sustituya el carácter (' , &, etc.) por ascii, eses es el principal problema de la función mysqli_addslashes().

CONSULTAS PREPARADAS:

CONSULTAS PREPARADAS

• VENTAJAS



- Evitan inyección SQL

- En consultas de tipo INSERT son más rápidas y eficientes.

• INCONVENIENTES

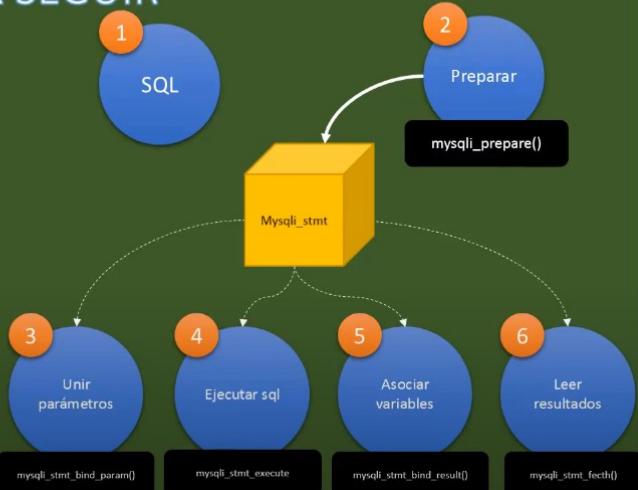


- Hay que escribir más código.

- Hay que conocer nuevas funciones.

CONSULTAS PREPARADAS PASOS A SEGUIR

- 1 • Creamos la sentencia SQL sustituyendo los valores de criterio por el símbolo ?
• \$sql="SELECT * FROM PRODUCTOS WHERE PAÍSDEORIGEN = ?";
- 2 • Preparamos la consulta con la función mysqli_prepare(). Esta función requiere dos parámetros: el objeto conexión y la sentencia sql.
• \$resultado=mysqli_prepare(\$conexion, \$sql); La función devuelve objeto de tipo mysqli_stmt.
- 3 • Unimos los parámetros a la sentencia sql. De esto se encarga la función mysqli_stmt_bind_param(). Devuelve true o false.
• Esta función requiere tres parámetros: el objeto mysqli_stmt (devuelto por mysqli_prepare), el tipo de dato que se utilizará como criterio en sql, variable con criterio.
- 4 • Ejecutar la consulta con la función mysqli_stmt_execute(). Esta función devuelve true o false.
• Necesita como parámetro el objeto mysqli_stmt.
- 5 • Asociar variables al resultado de la consulta. Esto lo conseguimos con la función mysqli_stmt_bind_result(). Devuelve true o false.
• Necesita como parámetros el objeto mysqli_stmt y tantas variables como columnas en consulta sql.
- 6 • Lectura de valores. Para ello utilizamos la función mysqli_stmt_fetch().
• Pide como parámetro el objeto mysqli_stmt



```

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");

mysqli_set_charset($conexion, "utf8");

$sql="SELECT CÓDIGOARTÍCULO, SECCIÓN, PRECIO, PAÍSDEORIGEN FROM PRODUCTOS WHERE PAÍSDEORIGEN= ?";

$resultado=mysqli_prepare($conexion, $sql);

$ok=mysqli_stmt_bind_param($resultado, "s", $pais);

$ok=mysqli_stmt_execute($resultado);

if($ok==false){

    echo "Error al ejecutar la consulta";

} else{

    $ok=mysqli_stmt_bind_result($resultado, $codigo, $seccion, $precio, $pais);

    echo "Artículos encontrados: <br><br>";

    while(mysqli_stmt_fetch($resultado)) {

        echo $codigo . " " . $seccion . " " . $precio . " " . $pais . "<br>";

    }

    mysqli_stmt_close($resultado);
}

```

Consulta preparada de inserción:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
</head>

<body>

<?php

$c_art = $_GET["c_art"];
$secc = $_GET["secc"];
$n_art = $_GET["n_art"];
$pre = $_GET["pre"];
$fec = $_GET["fec"];
$imp = $_GET["imp"];
$p_ori = $_GET["p_ori"];

require ("datos_conexion.php");

$conexion=mysqli_connect($db_host, $db_usuario, $db_contra);

if(mysqli_connect_errno()){

    echo "Falló al conectar con la BBDD";
    exit();
}

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");

mysqli_set_charset($conexion, "utf8");

$sql="INSERT INTO PRODUCTOS (CÓDIGOARTÍCULO, SECCIÓN, NOMBREARTÍCULO, PRECIO, FECHA,
IMPORTADO, PAÍSDEORIGEN) VALUES (?,?,?,?,?,?)";


```

```

$resultado=mysqli_prepare($conexion, $sql);

$ok=mysqli_stmt_bind_param($resultado, "ssssss", $c_art, $secc, $n_art, $pre, $fec, $imp, $p_ori);

$ok=mysqli_stmt_execute($resultado);

if($ok==false){

    echo "Error al ejecutar la consulta";
}
else{
    +
    // $ok=mysqli_stmt_bind_result($resultado, $codigo, $seccion, $precio, $pais);
    echo "Agregado nuevo registro";
    /*while(mysqli_stmt_fetch($resultado)){
        echo $codigo . " " . $seccion . " " . $precio . " " . $pais . "<br>";
    }*/
    mysqli_stmt_close($resultado);
}

>
```

Conectarse con BD MySQLi y POO:

APIS PHP

FORMAS DE CONECTAR CON BBDD



MYSQL

MYSQLi

PDO

Ejemplo:

```
$conexion= new mysqli("localhost", "root", "", "pruebas");

if($conexion->connect_errno){

    echo "Falló la conexión " . $conexion->connect_errno;

}

//mysqli_set_charset($conexion, "utf8");

$conexion->set_charset("utf8");

$sql="SELECT * FROM PRODUCTOS";

//$resultados=mysqli_query($conexion, $sql); [REMOVED LINE]

$resultados=$conexion->query($sql);
```

```
$resultados=$conexion->query($sql);

if($conexion->errno){

    die($conexion->error);

}

//while($fila=mysqli_fetch_array($resultados, MYSQL_ASSOC)) {
while($fila=$resultado->fetch_assoc()) {

    echo "<table><tr><td>";

    echo $fila['CÓDIGOARTÍCULO'] . "</td><td> ";
    echo $fila['NOMBREARTÍCULO'] . "</td><td> ";
    echo $fila['SECCIÓN'] . "</td><td> ";
    echo $fila['IMPORTADO'] . "</td><td> ";
    echo $fila['PRECIO'] . "</td><td> ";
    echo $fila['PAÍSDEORIGEN'] . "</td><td></tr></table>";

    echo"<br>";
    echo"<br>";

}
```

```

    //mysqli_close($conexion);
    $conexion->close(); //|
```

o también de esta forma:

```

$conexion->set_charset("utf8");

$sql="SELECT * FROM PRODUCTOS";

// $resultados=mysqli_query($conexion, $sql);

$resultados=$conexion->query($sql);

if($conexion->errno){

die($conexion->error);

}

//while($fila=mysqli_fetch_array($resultados, MYSQL_ASSOC)) {
while($fila=$resultados->fetch_array()){

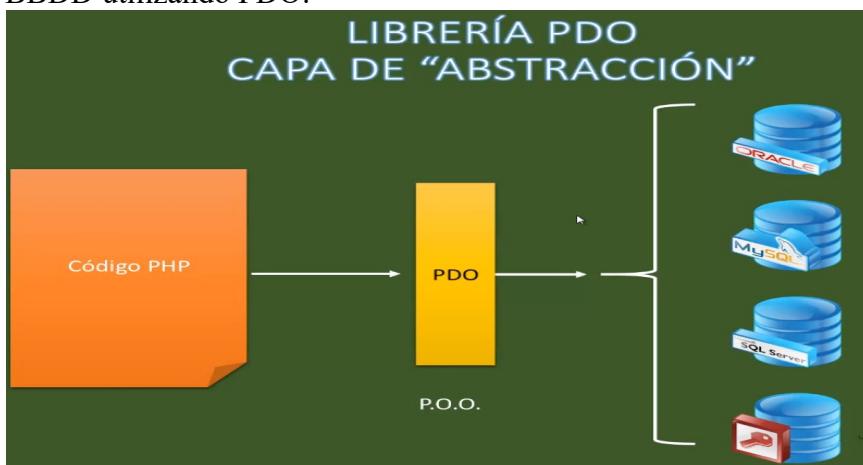
echo "<table><tr><td>";

echo $fila[0]. "</td><td> ";
echo $fila[1]. "</td><td> ";
echo $fila[2]. "</td><td> ";
echo $fila[3]. "</td><td> ";
echo $fila[4]. "</td><td> ";
echo $fila[5]. "</td><td></tr></table>";

echo"<br>";
echo"<br>";
```

}

Conexión a una BBDD utilizando PDO:



```

<?php
try{
    $base=new PDO('mysql:host=localhost; dbname=pruebassss', 'root', '');
    echo 'Conexión OK';
} catch (Exception $e) {
    die('Error: ' . $e->getMessage());
}
finally{
    $base=null;
}
?>
```

El `finally` se ejecuta siempre y con la instrucción `$base=null` lo que hacemos es liberar la memoria, es parecido a cerrar la conexión.

Consultas preparadas con PDO:



```

</head>
<body>
<?php
try{
    $base=new PDO('mysql:host=localhost; dbname=pruebas', 'root', '');
    $base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $base->exec("SET CHARACTER SET utf8");
    $sql="SELECT NOMBREARTÍCULO, SECCIÓN, PRECIO, PAÍSDEORIGEN FROM PRODUCTOS WHERE NOMBREARTÍCULO = ?";
    $resultado=$base->prepare($sql);
    $resultado->execute(array("Destornillador"));
    while($registro=$resultado->fetch(PDO::FETCH_ASSOC)){
        echo "Nombre Artículo: " . $registro['NOMBREARTÍCULO'] . " Sección: " . $registro['SECCIÓN'] .
        " Precio: " . $registro['PRECIO'] . " País de Origen: " . $registro['PAÍSDEORIGEN'] . "<br>";
    }
    $resultado->closeCursor();
}catch(Exception $e){
}
    
```

Con valor que viene de un formulario

```

</head>
<body>
<?php
$busqueda= $_GET["buscar"];
try{
    $base=new PDO('mysql:host=localhost; dbname=pruebas', 'root', '');
    $base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $base->exec("SET CHARACTER SET utf8");
    $sql="SELECT NOMBREARTÍCULO, SECCIÓN, PRECIO, PAÍSDEORIGEN FROM PRODUCTOS WHERE NOMBREARTÍCULO = ?";
    $resultado=$base->prepare($sql);
    $resultado->execute(array($busqueda));
    while($registro=$resultado->fetch(PDO::FETCH_ASSOC)){
        echo "Nombre Artículo: " . $registro['NOMBREARTÍCULO'] . " Sección: " . $registro['SECCIÓN'] .
        " Precio: " . $registro['PRECIO'] . " País de Origen: " . $registro['PAÍSDEORIGEN'] . "<br>";
    }
    $resultado->closeCursor();
}
    
```

Uso de marcadores en consultas preparadas:

- Cómo utilizar marcadores en una consulta preparada.
 - Uso de arrays asociativos en consultas preparadas.

Diferencias con el ejemplo anterior:

```
$sql="SELECT NOMBREARTÍCULO, SECCIÓN, PRECIO, PAÍSDEORIGEN FROM PRODUCTOS WHERE NOMBREARTÍCULO = :n_art";
+
$resultado=$base->prepare($sql);
+
$resultado->execute(array(":n_art"=>$busqueda));
```

Nota:

```
$base=new PDO('mysql:host=localhost; dbname=pruebas', 'root', '');
+
$base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
+
$base->exec("SET CHARACTER SET utf8");
```

```
public bool PDO::setAttribute ( int $attribute , mixed $value )
```

setAttribute establece un atributo. Por lo que la línea subrayada indica que coja las excepciones como atributos. Por lo que será capaz de crear esos objetos excepciones para analizar, mostrar, etc. ATTR_ERRMODE: Reporte de errores, entre ellos la opción ERREMODE_EXCEPTION. ERREMODE_EXCEPTION: Lanza excepciones.

con } catch (Exception \$e){ die(\$e->getMessage());

Podemos capturar el error porque se ha creado el objeto la excepción, sin la linea de settAttribute no se crearán objetos Exception.

- PDO::ATTR_ERRMODE: Reporte de errores.
- PDO::ERRMODE_SILENT: Establece los códigos de error.
- PDO::ERRMODE_WARNING: Eleva E_WARNING.
- PDO::ERRMODE_EXCEPTION: Lanza exceptions.

◦ PDO::ERRMODE_EXCEPTION

Además de establecer el código de error, PDO lanzará una PDOException y establecerá sus propiedades para reflejar el error y la información del mismo. Esta configuración también es útil durante la depuración, ya que, de hecho, señalará el lugar del error del script, apuntando a áreas potencialmente problemáticas del código (recuerde: las transacciones son automáticamente revertidas si la excepción causa la finalización del script).

El modo Exception también es útil porque se puede estructurar el manejo de errores con más claridad que con el estilo tradicional de advertencias de PHP, y con menos código/anidación que con la ejecución en modo silencioso, comprobando así explícitamente el valor devuelto de cada llamada a la base de datos.

Véase la referencia de [Excepciones](#) para más información sobre excepciones en PHP.

PDO utiliza las cadenas de código de error SQLSTATE del estándar SQL-92; cada controlador de PDO es responsable de la correspondencia de su códigos nativos con los códigos SQLSTATE apropiados. El método [PDO::errorCode\(\)](#) devuelve un único código SQLSTATE. Si fuera necesaria más información específica sobre el error, PDO también ofrece el método [PDO::errorInfo\(\)](#), que devuelve un array que contiene el código SQLSTATE, el código de error específico del controlador, y la cadena de error específica.

Sinopsis de la Clase

```
Exception {  
    /* Propiedades */  
    protected string $message ;  
    protected int $code ;  
    protected string $file ;  
    protected int $line ;  
  
    /* Métodos */  
    public __construct ([ string $message = "" [, int $code = 0 [, Exception $previous = NULL ]]] )  
    final public string getMessage ( void )  
    final public Exception getPrevious ( void )  
    final public mixed getCode ( void )  
    final public string getFile ( void )  
    final public int getLine ( void )  
    final public array getTrace ( void )  
    final public string getTraceAsString ( void )  
    public string __toString ( void )  
    final private void __clone ( void )  
}
```

La clase Exception tiene el método getMessage, pero también tiene varios métodos más.

Tabla de contenidos

- [Exception::__construct](#) — Constructor de la excepción
- [Exception::getMessage](#) — Obtiene el mensaje de Excepción
- [Exception::getPrevious](#) — Devuelve la excepción anterior
- [Exception::getCode](#) — Obtiene el código de una excepción
- [Exception::getFile](#) — Obtiene el fichero en el que se creó la excepción
- [Exception::getLine](#) — Obtiene la línea en el que se creó la excepción
- [Exception::getTrace](#) — Obtiene la traza de la pila
- [Exception::getTraceAsString](#) — Obtiene la traza de la pila como una cadena de caracteres
- [Exception::__toString](#) — Representación de la excepción en formato cadena
- [Exception::__clone](#) — Clona la excepción

Insertar/Eliminar registros de una BD con PDO: Ya no lo hacemos porque sería similar, lo que no nos hace falta es el bucle porque no vamos a mostrar por pantalla el resultado de una búsqueda.

Conexión a BD utilizando Clases POO:

Hasta ahora hemos visto al forma de programar procedimental bien usando la extensión mysqli, bien usando la librería PDO. Ahora usaremos la forma de programar Orientada a Objetos (POO) tanto con la extensión mysqli, como con la librería PDO.

Archivo de configuración con los datos de conexión (config.php):

```
<?php  
  
define ('DB_HOST', 'localhost');  
define ('DB_USUARIO', 'root');  
define ('DB_CONTRA', '');  
define ('DB_NOMBRE', 'pruebas');  
define ('DB_CHARSET', 'utf8');  
  
?>
```

Archivo de conexión a la base de datos (Conexion.php):

```

<?php

require ("config.php");
class Conexion{
    protected $conexion_db;

    public function Conexion(){
        $this->conexion_db=new mysqli(DB_HOST, DB_USUARIO, DB_CONTRA, DB_NOMBRE);
        if($this->conexion_db->connect_errno){
            echo "Falló al conectar a MySql: " . $this->conexion_db->connect_error;
            return;
        }
        $this->conexion_db->set_charset(DB_CHARSET);
    }
}

?>

```

Archivo de consulta (un select en este caso) DevuelveProductos.php:

```

<?php
require "Conexion.php";
class DevuelveProductos extends Conexion{

    public function DevuelveProductos(){
        parent::__construct();
    }

    public function get_productos(){

        $resultado=$this->conexion_db->query('SELECT * FROM PRODUCTOS');
        $productos=$resultado->fetch_all(MYSQLI_ASSOC);
        return $productos;
    }
}

```

Creamos el html para mostrar el array que retornamos en el archivo anterior (index.php):

```

<?php
require "DevuelveProductos.php";
$productos=new DevuelveProductos();
$array_productos=$productos->get_productos();

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
</head>
<body>
    <?php
        foreach($array_productos as $elemento){

            echo "<table><tr><td>";
            echo $elemento['CÓDIGOARTÍCULO'] . "</td><td>";
            echo $elemento['NOMBREARTÍCULO'] . "</td><td>";
            echo $elemento['SECCIÓN'] . "</td><td>";
            echo $elemento['PRECIO'] . "</td><td>";
            echo $elemento['FECCHA'] . "</td><td>";
            echo $elemento['IMPORTADO'] . "</td><td>";
            echo $elemento['PAÍSDEORIGEN'] . "</td><td></tr></table>";

            echo "<br>";
            echo "<br>";

        }
    </?php>

```

Conexión a BD utilizando Clases POO y PDO:

El archivo config valdría el mismo, pero el de conexión ya no, sería el siguiente. En este ejemplo omitimos el archivo config, establecemos los parámetros de esta conexión en este mismo archivo): Archivo Conexion.php:

```
<?php

    require ("config.php");

    class Conexion{

        protected $conexion_db;

        public function Conexion(){

            try{

                $this->conexion_db=new PDO('mysql:host=localhost; dbname=pruebas','root','');
                $this->conexion_db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
                $this->conexion_db->exec("SET CHARACTER SET utf8");
                return $this->conexion_db;

            }catch(Exception $e){

                echo "la linea de error es: " . $e->getLine();
            }
        }

        //---

        /*$this->conexion_db=new mysqli(DB_HOST, DB_USUARIO, DB_CONTRA, DB_NOMBRE);
        if($this->conexion_db->connect_errno){
            echo "Falló al conectar a MySql: " . $this->conexion_db->connect_error;
        }
    }
}
```

archivo DevuelveProductos.php:

```
class DevuelveProductos extends Conexion{

    public function DevuelveProductos(){

        parent::__construct();

    }

    public function get_productos($dato){

        $sql="SELECT * FROM PRODUCTOS WHERE PAÍSDEORIGEN='".$dato."'";
        $sentencia=$this->conexion_db->prepare($sql);
        $sentencia->execute(array());
        $resultado=$sentencia->fetchAll(PDO::FETCH_ASSOC);
        $sentencia->closeCursor();
        return $resultado;
    }

    $this->conexion_db=null;

    /*$resultado=$this->conexion_db->query('SELECT * FROM PRODUCTOS WHERE PAÍSDEORIGEN="'.$dato.'');
    $productos=$resultado->fetch_all(MYSQLI_ASSOC);
    return $resultado; */

}

?>
```

El archivo index.php nos vale uno parecido:

```
<?php
    require "DevuelveProductos.php";
    $pais=$_GET["buscar"];
    $productos=new DevuelveProductos();
    $array_productos=$productos->get_productos($pais);

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
</head>
<body>
<?php
    foreach($array_productos as $elemento){
        echo "<table><tr><td>";
        echo $elemento['CÓDIGOARTÍCULO'] . "</td><td>";
        echo $elemento['NOMBREARTÍCULO'] . "</td><td>";
        echo $elemento['SECCIÓN'] . "</td><td>";
        echo $elemento['PRECIO'] . "</td><td>";
        echo $elemento['FECHA'] . "</td><td>";
        echo $elemento['IMPORTADO'] . "</td><td>";
        echo $elemento['PAÍSDEORIGEN'] . "</td><td></tr></table>";
        echo "<br>";
        echo "<br>";
    }
}
```

El formulario es el siguiente:

```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Documento sin título</title>
    </head>
    <body>
        <form action="index.php" method="get">
            <label>Introduce país: <input type="text" name="buscar"></label>
            <input type="submit" name="enviando" value="¡Pále!">
        </form>
    </body>
</html>
```

SISTEMA DE LOGIN COMPLETO (SE TRABAJA CON SESIONES):

Creamos la tabla en la base de datos:

```
1 CREATE TABLE USUARIOS_PASS (ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, USUARIOS VARCHAR (20) NOT NULL, PASSWORD VARCHAR(20) NOT NULL)
```

Página de login (formulario):

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>
<body>
<h1> INTRODUCE TUS DATOS</h1>
<form action="comprueba_login.php" method="post">
<table>
<tr>
<td class="izq">
Login:</td><td class="der"><input type="text" name="login"></td></tr>
<tr><td class="izq">Password:</td><td class="der"><input type="password" name="password"></td></tr>
<tr><td colspan="2"><input type="submit" name="enviar" value="LOGIN"></td></tr></table>
</body>
</html>
```

El css del formulario es el siguiente:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>

<style>
    h1{text-align:center;
}
    table{
        width:25%;
        background-color:#FFC;
        border: 2px dotted #F00;
        margin:auto;
    }
    .izq{text-align:right;
}
    .der{
        text-align:left;
}
    td{
        text-align:center;
        padding:10px;
}
}

</style>
</head>
<body>
<h1> INTRODUCE TUS DATOS</h1>
```

quedando así:

INTRODUCE TUS DATOS

Login:

Password:

Página comprueba_login.php:

notas: htmlentities convierte cualquier símbolo en html (comillas, guión bajo, etc.)
addslashes escapa cualquier carácter del tipo que nombramos anteriormente: comilla, guión bajo, punto y coma, barra invertida, etc.; para que no se tenga en cuenta y evitar la inyección sql.
bindValue (es similar a bindParam ya usada en consultas preparadas cuando usamos parámetros (?) en vez de marcadores): para establecer que lo que esta guardado en \$login se corresponde a :login, por ejemplo. En este caso no necesitamos mostrar por pantalla el resultado de la consulta, por eso utilizamos la función rowCount() que lo que nos cuenta es el número de filas que se obtuvieron como resultado de esa consulta. En este caso de login el resultado de esa función es uno cuando el login es correcto y cero cuando no se metieron datos que están en la tabla de la base de datos USUARIOS_PASS.

Importante: Las sesiones se crean para que no nos llegue con saber la ruta (url) de una página para acceder a ella. Por ejemplo, si el login es correcto nos redirige a una página de usuarios registrados, pero si conocemos la ruta de esa página podríamos acceder a ella sin tener que pasar por el formulario de login. Entonces se tiene que crear una sesión y que sea imprescindible que la sesión esté abierta para poder acceder a la página de usuarios registrados.

session_start: inicia una nueva sesión o reanuda la existente. El identificador de sesión se pasa mediante una petición GET o POST, o mediante una cookie.

De momento:

```
<body>
<?php
```

```

try {

    $base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql="SELECT * FROM USUARIOS_PASS WHERE USUARIOS= :login AND PASSWORD= :password";

    $resultado=$base->prepare($sql);

    $login=htmlentities(addslashes($_POST["login"]));

    $password=htmlentities(addslashes($_POST["password"]));

    $resultado->bindValue(":login", $login);

    $resultado->bindValue(":password", $password);

    $resultado->execute();

    $numero_registro=$resultado->rowCount();

    if($numero_registro!=0){

        //echo "<h2>Adelante!!</h2>";

        session_start();

        $_SESSION["usuario"]=$_POST["login"];
        header("Location:usuarios_registrados1.php");
    }else{

        header("location:login.php");
    }

} catch(Exception $e){

    die("Error: ".$e->getMessage());
}
?>

```

usuarios_registrados1.php (página a donde dirigimos cuando el login es correcto, cuando el login era incorrecto no lo dejábamos salir de la página del formulario login.php). Aquí session_start() reanuda la sesión que iniciamos en la página de login:

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>

<body>

<?php

    session_start();

    if(!isset($_SESSION["usuario"])){
        header("Location:login.php");
    }

?>

<h1>Bienvenidos Usuarios</h1>]
<?php

    echo "Hola: " . $_SESSION["usuario"] . "<br><br>";

?>

<p>Esto es información solo para usuarios registrados</p>
</body>
</html>

```

Para cerrar sesión (cierre.php): En todas las páginas para usuarios registrados se incluye un enlace que conecta con esta página para cerrar sesión.

```

ay errores de sintaxis.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin titulo</title>
</head>

<body>
<?php

    session_start();
    |
    session_destroy();

    header("location:login.php");

?>
</body>
</html>

```

Haciendo lo mismo con `$_SERVER["PHP_SELF"]`:

```

</style>
</head>

<body>

<h1> INTRODUCE TUS DATOS</h1>

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    |
<table>
<tr>
<td class="izq">
Login:</td><td class="der"><input type="text" name="login"></td></tr>
<tr><td class="izq">Password:</td><td class="der"><input type="password" name="password">
<tr><td colspan="2"><input type="submit" name="enviar" value="LOGIN"></td></tr></table>
</form>

```

COOKIES: podemos hacer lo mismo que con las sesiones, pero de otra forma:
`setcookie` es la que establece una cookie, tiene 2 parámetros (nombre, valor).

`cookie.php`:

```

ay errores de sintaxis.
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin titulo</title>
</head>

<body>
<?php
    |
    setcookie("prueba", "Esta es la información de nuestra primera cookie");

?>

</body>
</html>

```

leer la cookie (`leer_cookie.php`):

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>

<body>
<?php

if(isset($_COOKIE["prueba"])){
    echo $_COOKIE["prueba"];
} else{
    echo "la cookie no se ha creado";
}

?>
</body>
</html>

```

resultado: “Esta es la información de nuestra primera cookie”.

Parámetros adicionales:

Ejemplo, para que la cookie duré medio minuto y que tenga vigencia en una ruta determinada:

```

<?php

setcookie("prueba", "Esta es la información de nuestra primera cookie",time()+30,
"/Curso%20PHP/sesiones%20s%20cookies/zona_contenidos/");

?>

```

Eliminar una cookie, se hace poniéndole un tiempo inferior al actual, se usa setcookie de la misma manera:

```

<body>
<?php

setcookie("prueba", "Esta es la información de nuestra primera cookie",time()-1,
"/Curso%20PHP/sesiones%20s%20cookies/zona_contenidos/");

?>
</body>
</html>

```

Ejemplo 1 Cookie. Guardar en una cookie la preferencia de idioma de un usuario:

primer archivo (pag1.php):

```

<body>
<?php
if(isset($_COOKIE["idioma_seleccionado"])){
    if($_COOKIE["idioma_seleccionado"]=="es"){
        header("Location:spanish.php");
    }else if($_COOKIE["idioma_seleccionado"]=="in"){
        header("Location: english.php");
    }
}

?>
<table width="25%" border="0" align="center">
<tr>
<td colspan="2" align="center"><h1>Elige idioma</h1></td>
</tr>
<tr>
<td align="center"><a href="creaCookie.php?idioma=es"></a></td>
<td align="center"><a href="creaCookie.php?idioma=in"></a></td>
</tr>
</table>
<p>&ampnbsp</p>
<p>&ampnbsp</p>
</body>

```

segundo archivo (creaCookie.php):

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin titulo</title>
</head>

<body>
<?php

    setcookie("idioma_seleccionado", $_GET["idioma"], time() + 86400);

    header("Location:ver_cookie.php");

?>
</body>
</html>

```

tercer archivo (ver_cookie.php):

```

<body>
<?php

    if(!$_COOKIE["idioma_seleccionado"]){
        header("Location:pag1.php");
    }else if($_COOKIE["idioma_seleccionado"]=="es"){

        header("Location:spanish.php");

    }else if($_COOKIE["idioma_seleccionado"]=="in"){

        header("Location: english.php");
    }

?>
</body>
</html>

```

cuarto archivo (destruye_cookie) una para cada idioma (esto es mejorable):

```

<body>
<?php

    setcookie("idioma_seleccionado", "es", time() -1);

?>
</body>

```

Segundo ejemplo de cookie. Guardar en una cookie los datos de login para no volver a loguearse en ese equipo, si el usuario marca esa opción:

primer archivo (formulario.html):

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
<style>
    h1{text-align:center;}
    table{
        width: 25%;
        background-color:#FFC;
        border: 2px dotted #FOO;
        margin:auto;
    }
    .izq{text-align:right;}
    .der{
        text-align:left;
    }
    td{
        text-align:center;
        padding:10px;
    }
</style>
</head>
<body>
<h1> INTRODUCE TUS DATOS</h1>

<form action="php echo $_SERVER['PHP_SELF']; ?" method="post">
<table>
<tr><td class="der"><input type="text" name="login"></td></tr>
<tr><td class="izq">Password:</td><td class="der"><input type="password" name="password"></td></tr>
<tr><td class="izq">Recordar:</td><td class="der"><input type="checkbox" name="recordar" id="recordar">
    <label for="recordar"></label></td></tr>
<tr><td colspan="2"><input type="submit" name="enviar" value="LOGIN"></td></tr></table>
</form>
</body>
</html>
```

segundo archivo (login.php) cambios con respecto al sistema de sesiones anterior:

```
</style>
</head>
<body>
<?
if ($autenticado=false);

if (isset($_POST["enviar"])){
try{
$base=new PDO("mysql:host=localhost; dbname=pruebas" , "root", "");
$base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$sql="SELECT * FROM USUARIOS_PASS WHERE USUARIOS= :login AND PASSWORD= :password";
$resultado=$base->prepare($sql);

$login=htmlentities(addslashes($_POST["login"]));
$password=htmlentities(addslashes($_POST["password"]));

$resultado->bindValue(":login", $login);
$resultado->bindValue(":password", $password);

$resultado->execute();
$numero_registro=$resultado->rowCount();
```

```
if ($numero_registro!=0){
    $autenticado=true;
    if (isset($_POST["recordar"])){
        setcookie("nombre_usuario", $_POST["login"], time()+86400);
    }
} else{
    //header("location:login.php");
    echo "Error. Usuario o contraseña incorrectos";
}

}catch(Exception $e){
```

```

die("Error: " . $e->getMessage());

}

}

?>
<?php I

if($autenticado==false){
    if(!isset($_COOKIE["nombre_usuario"])){
        include("formulario.html");
    }
}

?>
<h2>CONTENIDO DE LA WEB</h2>
<table width="800" border="0">
    <tr>
        <td></td>
        <td></td>
    </tr>
    <tr>
        <td></td>
        <td></td>
    </tr>
</table>
<?php

if($autenticado==true || isset($_COOKIE["nombre_usuario"])){
    include("zona_registrados.html");
}

```

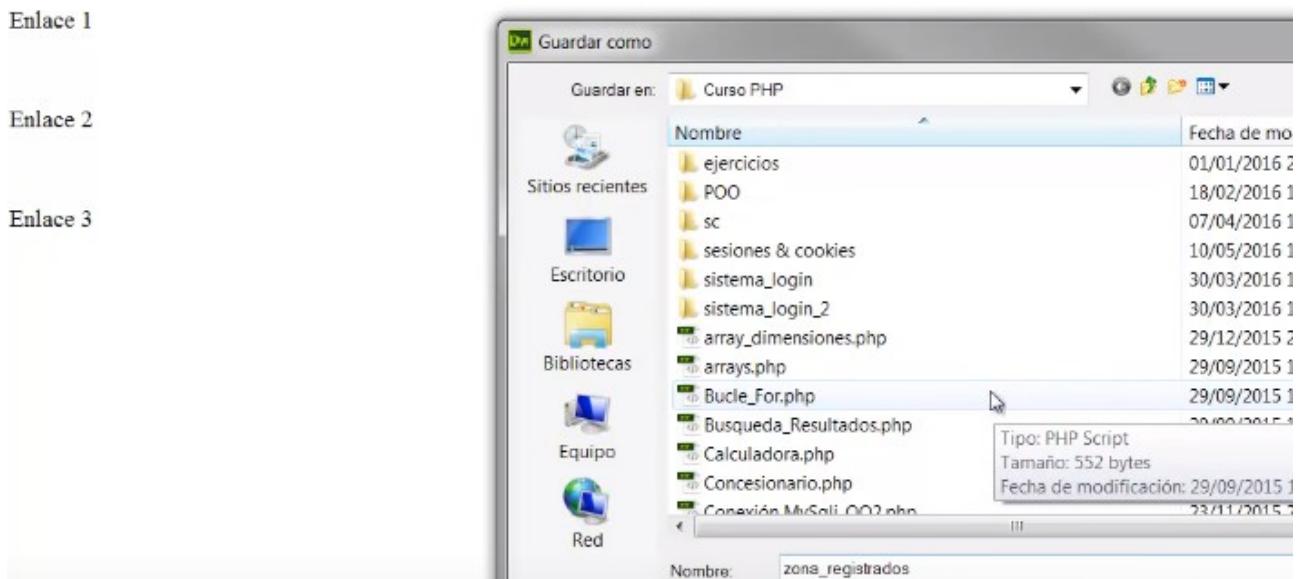
tercer archivo (zona_registrados.html): contenido que sólo pueden ver los usuarios que hayan hecho login con éxito.

Esta es la zona de usuarios registrados

Enlace 1

Enlace 2

Enlace 3



Se puede poner todo lo que quieras y se mostraría después de la tabla que muestra las fotos.

Una pequeña modificación sobre el archivo (login.php) es que muestre un saludo personalizado antes de la tabla con las fotos, ese trozo de código modificado sería así:

```

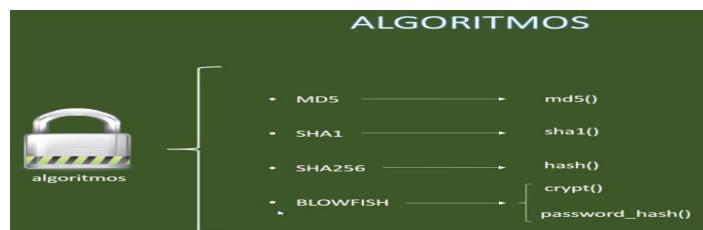
<?php
    if(Sautenticado==false){
        if(!isset($_COOKIE["nombre_usuario"])){
            include("formulario.html");
        }
    }

    if(isset($_COOKIE["nombre_usuario"])){
        echo "Hola " . $_COOKIE["nombre_usuario"] . "!";
    }else if(Sautenticado==true){
        echo "Hola " . $_POST["login"] . "!";
    }
?>

<h2>CONTENIDO DE LA WEB</h2>
<table width="800" border="0">
<tr>
    <td></td>
    <td></td>
</tr>
<tr>
    <td></td>
    <td></td>

```

Encriptando Password (algoritmo Blowfish con password_hash()):



Ejemplo: Primer archivo (Formulario_Insertar_Usuarios.php):

```

<title>Documento sin título</title>

<style>
table{
    width:300px;
    margin:auto;
    background-color:#FFC;
    border:2px solid #F00;
    padding:5px;
}

td{
    text-align:center;
}
h1{text-align:center}
</style>

</head>

<body>
<h1>REGISTRATE</h1>

<form action="pagina_insertar_usuarios.php" method="post">
<table><tr>
<td> Usuario </td><td><input type="text" name="usu" id="usu"></td></tr>
<tr>
<td> Contraseña </td><td><input type="text" name="contra" id="contra"></td></tr>
<tr><td colspan="2"> <input type="submit" name="enviando" value="Dale!">
</td></tr></table>
</form>

</body>

```

2º Archivo (pagina_insertar_usuarios.php):

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin titulo</title>
</head>

<body>

<?php

$usuario= $_POST["usu"];
$contrasenia= $_POST["contra"];

$pass_cifrado=password_hash($contrasenia, PASSWORD_DEFAULT);

try{

    $base=new PDO('mysql:host=localhost; dbname=pruebas', 'root', '');
    $base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $base->exec("SET CHARACTER SET utf8");

    $sql="INSERT INTO USUARIOS (USUARIO, PASSWORD) VALUES (:usu, :contra)";

    $resultado=$base->prepare($sql);

    $resultado->execute(array(":usu"=>$usuario, ":contra"=>$pass_cifrado));

    echo "Registro insertado";

    $resultado->closeCursor();

} catch(Exception $e){

    echo "Linea del error: " . $e->getLine();

} finally{

    $base=null;

}

?>
</body>
```

Añadirle un coste distinto al que viene por defecto:

```
$pass_cifrado = password_hash($contrasenia, PASSWORD_DEFAULT, array("cost"=>12));
```

Función para comparar la contraseña introducida por el usuario con la contraseña que hay almacenada en la BD:

password_verify(contraseña, hash). Los parámetros son la contraseña introducida por el usuario y la contraseña almacenada en la BD.

Ejemplo de uso (2 archivos):

Primer archivo (login.php):

```
</style>
</head>
<body>
<h1> INTRODUCE TUS DATOS</h1>
<form action="comprueba_login.php" method="post">
<table>
<tr style="">
<td class="izq">
Login:</td><td class="der"><input type="text" name="login"></td></tr>
<tr><td class="izq">Password:</td><td class="der"><input type="password" name="password"></td></tr>
<tr><td colspan="2"><input type="submit" name="enviar" value="LOGIN"></td></tr>
</table>
</form>
</body>
</html>
```

2º archivo (comprueba_login.php):

```
<?php

try{

    $login=htmlentities(addslashes($_POST["login"]));

    $password=htmlentities(addslashes($_POST["password"]));
    $contador=0;

    $base=new PDO("mysql:host=localhost; dbname=pruebas" , "root", "");

    $base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql="SELECT * FROM USUARIOS_PASS WHERE USUARIOS= :login";

    $resultado=$base->prepare($sql);

    $resultado->execute(array(":login"=>$login));

    while($registro=$resultado->fetch(PDO::FETCH_ASSOC)) {

        //echo "Usuario: " . $registro['USUARIOS'] . " Contraseña: " . $registro['PASSWORD'] . "<br>";

        if(password_verify($password,$registro['PASSWORD'])){

            $contador++;

        }

    }

    if($contador>0){

        echo "Usuario registrado";
    }else{
        echo "Usuario no registrado";
    }

    $resultado->closeCursor();

}

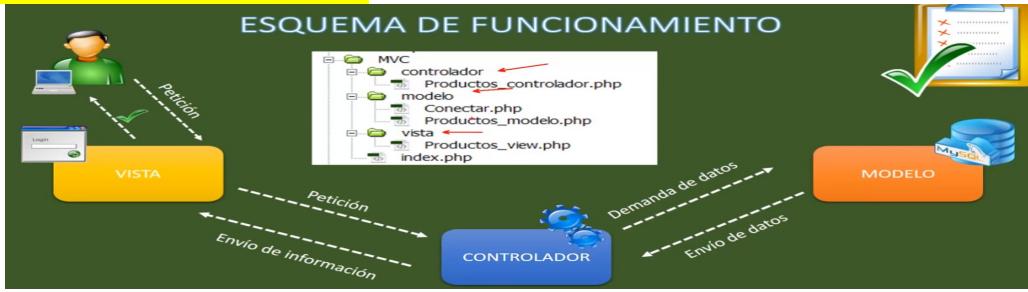
}catch(Exception $e){

    die("Error: " . $e->getMessage());
}

?>
</body>
```

CRUD:

MODELO VISTA CONTROLADOR:



Conectrar.php:

```
<?php
class Conectrar{

    public static function conexion(){

        try{
            $conexion=new PDO('mysql:host=localhost; dbname=pruebas', 'root', '');
            $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $conexion->exec("SET CHARACTER SET UTF8");

        }catch(Exception $e){
            die("Error" . $e->getMessage());
            echo "Linea del error" . $e->getLine();

        }
        return $conexion;
    }

}
?>
```

Productos_modelo.php (que nos proporcione todos los productos, por ejemplo):

```
<?php
class Productos_modelo{

    private $db;
    private $productos;

    public function __construct(){

        require_once("modelo/Conectrar.php");
        $this->db=Conectrar::conexion();
        $this->productos=array();

    }

    public function get_productos(){
        $consulta=$this->db->query("SELECT * FROM PRODUCTOS");
        while($filas=$consulta->fetch(PDO::FETCH_ASSOC)){
            $this->productos[]=$filas;

        }
        return $this->productos;
    }

}
```

Productos_view.php:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin titulo</title>
<style>
    td{
        border:1px dotted #FF0000;
    }
</style>
</head>
<body>
<table>
<tr><td>NOMBRE DEL ARTÍCULO</td>

<?php
    foreach($matrizProductos as $registro){
        echo "<tr><td>" . $registro["NOMBREARTICULO"] . "</td></tr>";
    }
?
</table>
</body>
</html>
```

Productos_controlador.php: (las rutas difieren de lo que debería de ser porque este archivo lo llamamos desde el index.php, es como si estuviese allí).

```
<?php
    require_once ("modelo/Productos_modelo.php");

    $producto=new Productos_modelo();
    $matrizProductos=$producto->get_productos();

    require_once ("vista/Productos_view.php");
?

```

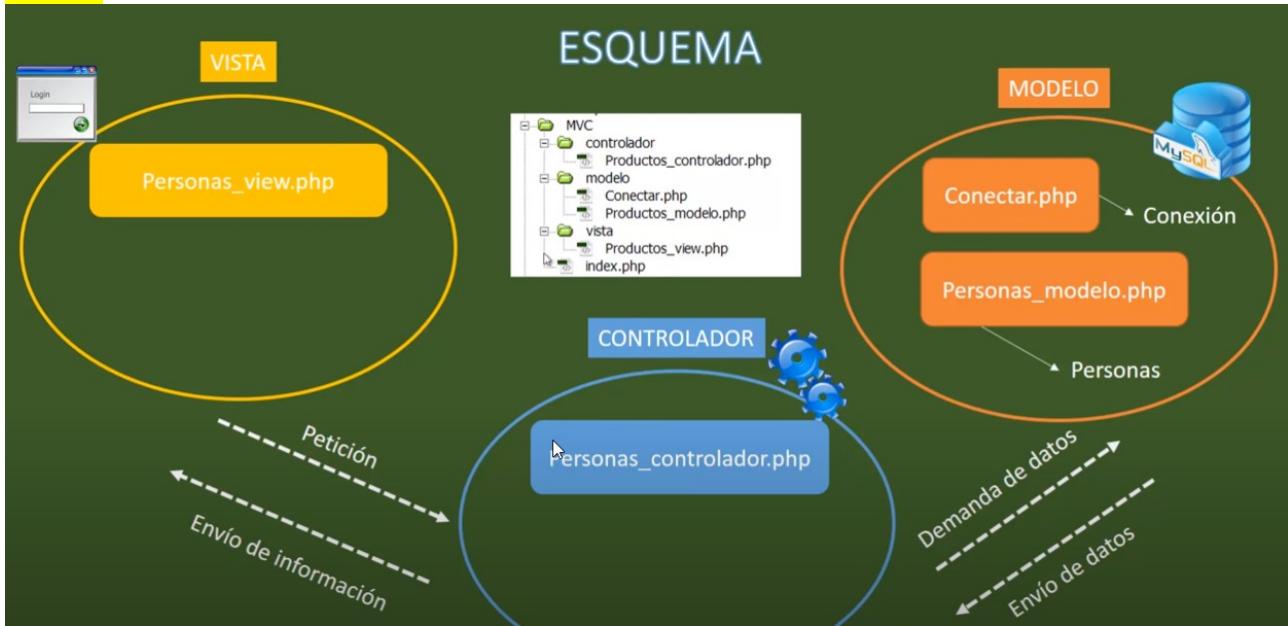
index.php:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin titulo</title>

<style>
    h1{text-align:center;
)
</style>
</head>

<body>
<h1>Modelo Vista Controlador</h1>
<br>
<?php
    require_once ("controlador/Productos_controlador.php");
?
</body>
</html>
```

CRUD:



index.php (se incluye también el enlace a la hoja de estilos):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
<link rel="stylesheet" type="text/css" href="hoja.css">
</head>

<body>
<h1>Modelo Vista Controlador</h1>
<br>
<?php

    require_once("controlador/Personas_controlador.php");
    I

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

Personas_controlador.php:

```
<?php
    //Llamada al modelo

    require_once("/modelo/Personas_modelo.php");

    $persona=new Personas_model();

    $matrizPersonas=$persona->get_personas();

    //Llamada a la vista

    require_once("/vista/Personas_view.php");

?>
```

Personas modelo.php:

```
<?php

class Personas_model{
    private $db;
    private $personas;

    public function __construct(){
        require_once("/modelo/Conectar.php");
        $this->db=Conectar::conexion();
        $this->personas=array();
    }

    public function get_personas(){
        require("paginacion.php");
        $consulta=$this->db->query("SELECT * FROM datos_usuarios LIMIT Sempezar_desde, Stamagno_pagina");
        //while ($filas=$consulta->fetch_assoc()){
        while($filas=$consulta->fetch(PDO::FETCH_ASSOC)){
            $this->personas[]=$filas;
        }
        return $this->personas;
    }

}

?>
```

hoja.css (la colocamos en la misma carpeta que el index.php porque todo parte del index.php):

```
@charset "utf-8";
/* CSS Document */
h1{text-align:center;
}

.subtitle{
    font-size:12px;
}

body{
    background-color:#FPC;
}

table td{
    text-align:center;
    border:1px #000099 dotted;
}

table .sin{
    border:0;
}

table .bot{
    padding:0 5px;
    display:inline;
    border:0;
}

table .primera_fila{
    font-size:1.5em;
    text-decoration:underline;
    background-color:#FC3;
}

.centrado{
    text-align:center;
}
```

paginacion.php (dentro de la carpeta modelo):

```
<?php

    require_once ("Conectar.php");
    $base=Conectar::conexion();
    $stamagno_paginas=3;

    if(isset($_GET["pagina"])){
        if ($_GET["pagina"]==1){

            header("Location:index.php");

        }else{
            $pagina=$_GET["pagina"];
        }
    }else{
        $pagina=1;
    }

    $empezar_desde=($pagina-1)*$stamagno_paginas;
    $sql_total="SELECT * FROM DATOS_USUARIOS";
    $resultado=$base->prepare($sql_total);
    $resultado->execute(array());
    $num_filas=$resultado->rowCount();
    $total_paginas=ceil($num_filas/$stamagno_paginas);
```

Personas_view.php:

```
<body>

<?php

    require ("modelo/paginacion.php");

?>

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    <table width="50%" border="0" align="center">
        <tr >
            <td class="primera_fila">Id</td>
            <td class="primera_fila">Nombre</td>
            <td class="primera_fila">Apellido</td>
            <td class="primera_fila">Dirección</td>
            <td class="sin">&ampnbsp</td>
            <td class="sin">&ampnbsp</td>
            <td class="sin">&ampnbsp</td>
        </tr>
    </table>
    <?php
        foreach($matrixPersonas as $persona):?>
        <tr>
            <td><?php echo $persona["Id"]?> </td>
            <td><?php echo $persona["Nombre"]?></td>
            <td><?php echo $persona["Apellido"]?></td>
            <td><?php echo $persona["Direccion"]?></td>
```

```

<td class="bot"><a href="borrar.php?Id=<?php echo $persona["Id"]?>"><input type='button' name='del' id='del' value='Borrar'></a></td>
<td class="bot"><a href="editar.php?Id=<?php echo $persona["Id"]?> & nom=<?php echo $persona["Nombre"]?> & ape=<?php echo $persona["Apellido"]?> & dir=<?php echo $persona["Direccion"]?>"><input type='button' name='up' id='up' value='Actualizar'></a></td>
</tr>

<?php
endforeach;

?>

<tr>
<td></td>
<td><input type='text' name='Nom' size='10' class='centrado'></td>
<td><input type='text' name='Ape' size='10' class='centrado'></td>
<td><input type='text' name='Dir' size='10' class='centrado'></td>
<td class='bot'><input type='submit' name='cr' id='cr' value='Insertar'></td></tr>
<tr><td><?php

//-----PAGINACIÓN-----

for($i=1; $i<=$total_paginas; $i++) {
    echo "<a href='?pagina=" . $i . "'>" . $i . "</a>  ";
}

?></td></tr>

</table>
</form>
</body>
</html>

```

SUBIR IMÁGENES AL SERVIDOR

Formulario para subir imágenes (index.php):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
<style>
table{
    margin:auto;
    width:450px;
    border:2px dotted #FF0000;
}
</style>
</head>
<body>
<form action="datosImagen.php" method="post" enctype="multipart/form-data">
<table>
<tr>
<td>
<label for="Imagen">Imagen:</label></td>
<td><input type="file" name="Imagen" size="20"></td></tr>
<tr><td colspan="2" style="text-align:center"><input type="submit" value="Enviar Imagen"></td></tr></table>
</form>
</body>
</html>
```

Archivo que gestiona las imágenes (datosImagen.php):

Las mueve de la carpeta temporal a una carpeta del servidor y también sube el nombre de la imagen a un campo de la base de datos. Limita que los archivos que se quieran subir sean sólo imágenes y también el tamaño de esas imágenes. Es importante saber que `$_SERVER['DOCUMENT_ROOT']` equivale a la carpeta raíz del servidor, enxampp es `htdocs`, por ejemplo. A continuación hay que concatenar el resto de la ruta a la carpeta del servidor elegida para almacenar las imágenes

```
<?php
// Recibimos los datos de la imagen

$nombre_imagen=$_FILES['Imagen']['name'];
$tipo_imagen=$_FILES['Imagen']['type'];
$stamagno_imagen=$_FILES['Imagen']['size'];

//echo $tipo_imagen;

if($stamagno_imagen<=3000000){
    if($tipo_imagen=="image/jpeg" || $tipo_imagen=="image/jpg" || $tipo_imagen=="image/png" || $tipo_imagen=="image/gif"){
        // Ruta del la carpeta destino en servidor
        $carpeta_destino=$_SERVER['DOCUMENT_ROOT'].'/intranet/uploads/';
        //Movemos la imagen del directorio temporal al directorio escogido
        move_uploaded_file($_FILES['Imagen']['tmp_name'],$carpeta_destino.$nombre_imagen);
    }else{
        echo "Solo se pueden subir imágenes jpg/jpeg/png/gif";
    }
}else{
    echo "El tamaño es demasiado grande";
}

require("datosConexion.php");
$conexion=mysqli_connect($db_host, $db_usuario, $db_contra);

if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion, "utf8");

//sql="INSERT INTO PRODUCTOS (FOTO) VALUES ('$nombre_imagen')";
$sql="UPDATE PRODUCTOS SET FOTO='$nombre_imagen' WHERE CÓDIGOARTÍCULO='AR01'";
$resultado=mysqli_query($conexion, $sql);
```

Archivo para leer las imágenes de la base de datos (hay que recordar que hasta ahora sólo hemos almacenado el nombre de la imagen en un campo de la BD). Archivo leer_imagen_bbdd.php:

```
<?php  
require("datos_conexion.php");  
$conexion=mysqli_connect($db_host, $db_usuario, $db_contra);  
if(mysqli_connect_errno()){  
    echo "Fallo al conectar con la BBDD";  
    exit();  
}  
mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");  
mysqli_set_charset($conexion, "utf8");  
$consulta="SELECT FOTO FROM PRODUCTOS WHERE CÓDIGOARTÍCULO='AR01';  
$resultado=mysqli_query($conexion, $consulta);  
while($fila=mysqli_fetch_array($resultado)){  
    $ruta_img=$fila["FOTO"];  
  
}  
  
?>  
<div>  
      
</div>  
</body>  
</html>
```

IMÁGENES EN LA BD MYSQL/ CAMPOS BLOB:

Lo primero que tenemos que hacer es tener una columna en la BD con campos BLOB, es donde se guardarán las imágenes (un campo BLOB guarda también otro tipo de archivos que no sean imágenes).

El archivo index.php sería similar al anterior index.php.:

```
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>Documento sin título</title>  
<style>  
table{  
    margin:auto;  
    width:450px;  
    border:2px dotted #FF0000;  
}  
  
</style>  
</head>  
<body>  
<form action="datosArchivos.php" method="post" enctype="multipart/form-data">  
<table>  
<tr>  
<td>  
<label for="imagen">Imagen:</label></td>  
<td><input type="file" name="archivo" size="20"></td></tr>  
<tr><td colspan="2" style="text-align:center"><input type="submit" value="Enviar Archivo"></td></tr></table>  
</form>  
</body>  
</html>
```

Al archivo datosImagenes.php le llamaremos datosArchivos.php.

Sería también muy similar:

```

$nombre_archivo=$_FILES['archivo']['name'];
$tipo_archivo=$_FILES['archivo']['type'];
$stamagno_archivo=$_FILES['archivo']['size'];

//echo $tipo_imagen;

if($stamagno_archivo<=1000000){

    //if($tipo_imagen=="image/jpeg" || $tipo_imagen=="image/jpg" || $tipo_imagen=="image/png" || $tipo_imagen=="image/gif"){

        // Ruta del la carpeta destino en servidor
        $carpeta_destino=$_SERVER['DOCUMENT_ROOT'].'/intranet/uploads/';

        //Movemos la imagen del directorio temporal al directorio escogido
        move_uploaded_file($_FILES['archivo']['tmp_name'],$carpeta_destino.$nombre_archivo);

    //}else{
        //echo "Solo se pueden subir imágenes jpg/jpeg/png/gif";
    //}
}

}else{
    echo "El tamaño es demasiado grande";
}

```

```

require("datos_conexion.php");

$conexion=mysqli_connect($db_host, $db_usuario, $db_contra);

if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion, "utf8");

$archivo_objetivo=fopen($carpeta_destino . $nombre_archivo, "r");
$contenido=fread($archivo_objetivo, $stamagno_archivo);
$contenido=addslashes($contenido);
fclose($archivo_objetivo);

$sql="INSERT INTO ARCHIVOS (Id, Nombre, Tipo, Contenido) VALUES (0, '$nombre_archivo', '$tipo_archivo',
'$contenido')";

// $sql="UPDATE PRODUCTOS SET FOTO='$nombre_imagen' WHERE CÓDIGOARTÍCULO='AR01'";
$resultado=mysqli_query($conexion, $sql);

if(mysqli_affected_rows($conexion)>0){
    echo "Se ha insertado el registro con éxito";
} else {
    echo "No se ha podido insertar el registro";
}

```

**COMO LEER LA INFORMACIÓN (IMÁGENES, ETC) DE UN CAMPO BLOG ALMACENADA EN LA BD:
leer_archivo_bd.php:**

```

require("datos_conexion.php");

$conexion=mysqli_connect($db_host, $db_usuario, $db_contra);

if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion, "utf8");

$consulta="SELECT * FROM ARCHIVOS WHERE ID=4";
$resultado=mysqli_query($conexion, $consulta);

while($fila=mysqli_fetch_array($resultado)){
    $id=$fila["id"];
    $contenido=$fila["Contenido"];
    $tipo=$fila["Tipo"];
}

echo "Id: " . $id;

```

```

        echo "Id: " . $id . "<br>";
        echo "Tipo: " . $tipo . "<br>";
        echo "<img src='data:image/jpeg; base64," . base64_encode($contenido) . "'>";

?>
<div>

</div>
</body>
</html>

```

Enviar correo electrónico:

Formulario_envio.php:

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>

<style>

table{
    margin:auto;
    left:0;
}

</style>
</head>

<body>

<form name="formulario_contacto" method="post" action="Enviar_mail.php">
<table width="500px">
<tr>
<td>
<label for="nombre">Nombre: *</label>
</td>
<td>
<input type="text" name="nombre" maxlength="50" size="25">
</td>
</tr>
<tr>
<td valign="top">
<label for="apellido">Apellido: *</label>
</td>
<td>
<input type="text" name="apellido" maxlength="50" size="25">
</td>
</tr>
<tr>
<td>

```

```

<label for="email">Dirección de E-mail: *</label>
</td>
<td>
<input type="text" name="email" maxlength="80" size="35">
</td>
</tr>
<tr>
<td>
<label for="tfno">Número de teléfono:</label>
</td>
<td>
<input type="text" name="tfno" maxlength="25" size="15">
</td>
</tr>
<tr>
<td>Asunto:</td>
<td><label for="asunto"></label>
<input type="text" name="asunto" id="asunto"></td>
</tr>
<tr>
<td>
<label for="comments">Comentarios: *</label>
</td>
<td>
<textarea name="comentarios" maxlength="500" cols="30" rows="5"></textarea>
</td>
</tr>
<tr>
<td colspan="2" style="text-align:center">
<input type="submit" value="Enviar">
</td>
</tr>
</table>
</form>
</body>
</html>

```

Enviar_mail.php (hay que usar la funcion mail de php):

```

<?php
if($_POST["nombre"]=="") || $_POST["apellido"]=="") || $_POST["email"]=="") || $_POST["comentarios"]=="") {
    echo "ha habido un error. Revisa los campos";
    die();
}

$texto_mail=$_POST["comentarios"];
$destinatario=$_POST["email"];
$asunto=$_POST["asunto"];
$headers="MIME-Version: 1.0\r\n";
$headers.="Content-type: text/html; charset=iso-8859-1\r\n";
$headers.="From: Prueba Juan < cursos@pildorasinformaticas.es >\r\n";
$envio=mail($destinatario,$asunto,$texto_mail,$headers);

if($envio){
    echo "Mensaje enviado con éxito";
} else{

```

```

echo "Ha habido un error";
?>

```

CREACIÓN DE BLOG I

The screenshot shows the 'Estructura' (Structure) tab of MySQL Workbench. A new table named 'conetenido' is being created. The table has five columns: 'Id' (INT), 'Titulo' (VARCHAR(25)), 'Fecha' (DATE), 'Comentario' (TEXT), and 'Imagen' (VARCHAR(35)). The 'Imagen' column has a length of 35. The 'Cotejamiento' (Collation) for all columns is set to 'Ninguno'. The 'Motor de almacenamiento' (Storage Engine) is set to 'InnoDB'. The 'Cotejamiento' dropdown also shows 'InnoDB'.

Fecha será datetime en vez de date.

The screenshot shows the results of an 'ALTER TABLE' command. The message says 'Los cambios en la Tabla contenido se hicieron exitosamente' (The changes in the Table contenido were made successfully). The command executed was 'ALTER TABLE `contenido` CHANGE `Fecha` `Fecha` DATETIME NOT NULL;'. The table structure is displayed with the 'Fecha' column now defined as 'datetime'.

The screenshot shows a 'Nueva entrada' (New Entry) form. It has fields for 'Título:' (Title:) and 'Comentarios:' (Comments:). Below these is a note: 'Seleccione una imagen con tamaño inferior a 2 MB' (Select an image with a size less than 2 MB). There is a file input field labeled 'Examinar...' (Browse...) and a submit button labeled 'Enviar' (Send). At the bottom, there is a link 'Página de visualización del blog' (Blog Post View Page).

Código:

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Blog Píldoras</title>
<style>

h2{
    text-align:center;
}

table{
    width:50%;
    margin:auto;
    background-color:#FF9;
    border:solid 2px #FF9900;
    padding:5px;
}

td{
    padding:5px 0;
}

</style>
</head>

<body>
<h2>Nueva entrada</h2>
<form action="Insertar Contenido.php" method="post" enctype="multipart/form-data"
name="form1">
<table >
<tr>
<td>Título:
    <label for="campo_titulo"></label></td>
<td><input type="text" name="campo_titulo" id="campo_titulo"></td>

</tr>
<tr><td>Comentarios:
    <label for="area_comentarios"></label></td>
    <td><textarea name="area_comentarios" id="area_comentarios" rows="10"
cols="50"></textarea></td>
</tr>
    <input type="hidden" name="MAX_TAM" value="2097152">
<tr>
<td colspan="2" align="center">Seleccione una imagen con tamaño inferior a 2 MB</td></tr>
<tr>
<td colspan="2" align="left"><input type="file" name="imagen" id="imagen"></td>
</tr>
<tr>

```

```

<td colspan="2" align="center">
<input type="submit" name="btn_enviar" id="btn_enviar" value="Enviar"></td></tr>
<tr><td colspan="2" align="center"><a href="Mostrar Blog.php">Página de visualización del
blog</a></td></tr>
</table>
</form>
<p>&nbsp;</p>
</body>
</html>

```



Explicación de los mensajes de error

PHP devuelve un código de error apropiado junto con el array de ficheros. El código de error se puede encontrar en el segmento `error` del array de ficheros que PHP crea durante la subida de ficheros. En otras palabras, el error puede encontrarse en `$_FILES['fichero_usuario']['error']`.

- UPLOAD_ERR_OK**
Valor: 0; No hay error, fichero subido con éxito.
- UPLOAD_ERR_INI_SIZE**
Valor: 1; El fichero subido excede la directiva `upload_max_filesize` de `php.ini`.
- UPLOAD_ERR_FORM_SIZE**
Valor: 2; El fichero subido excede la directiva `MAX_FILE_SIZE` especificada en el formulario HTML.
- UPLOAD_ERR_PARTIAL**
Valor: 3; El fichero fue sólo parcialmente subido.
- UPLOAD_ERR_NO_FILE**
Valor: 4; No se subió ningún fichero.
- UPLOAD_ERR_NO_TMP_DIR**
Valor: 6; Falta la carpeta temporal. Introducido en PHP 5.0.3.
- UPLOAD_ERR_CANT_WRITE**
Valor: 7; No se pudo escribir el fichero en el disco. Introducido en PHP 5.1.0.

Insertar Contenido.php:

```

<meta charset="utf-8">
<title>Documento sin título</title>
</head>

<body>

<?php

$miConexion=mysqli_connect("localhost", "root", "", "bbddBlog");

/*Comprobar conexión*/
if(!$miConexion){

    echo "La conexión ha fallado: " . mysqli_error();
    exit();
}

```

```

if($_FILES['imagen']['error']){
    switch($_FILES['imagen']['error']){
        case 1: // Error exceso de tamaño de archivo en php.ini
            echo "El tamaño del archivo excede lo permitido por el servidor";
            break;
        case 2: //Error tamaño archivo maracado desde formulario
            echo "El tamaño del archivo excede 2 MB";
            break;
        case 3: //Corrupción de archivo
            echo "El envío de archivo se interrumpió";
            break;
        case 4: // No hay fichero
            echo "No se ha enviado ningún archivo";
            break;
    }
} else{
    echo "No hay error en la transferencia del archivo. <br/>";
    if(isset($_FILES['imagen']) && ($_FILES['imagen']['error'] == UPLOAD_ERR_OK)){
        $destino_de_ruta='imagenes/';
        move_uploaded_file($_FILES['imagen']['tmp_name'], $destino_de_ruta . $_FILES['imagen']['name']);
        echo "El archivo " . $_FILES['imagen']['name'] . " Se ha copiado en el directorio de imágenes";
    } else{
        echo "El archivo no se ha copiado en el directorio de imágenes";
    }
}

```

```

$eltitulo=$_POST['campo_titulo'];
$lafecha=date("Y-m-d H:i:s");
$elcomentario=$_POST['area_comentarios'];
$laImagen=$_FILES['imagen']['name'];

$miconsulta="INSERT INTO contenido (Titulo, fecha, Comentario, Imagen) VALUES ('" . $eltitulo . '", "' . $lafecha . '", "' . $elcomentario . '", "' . $laImagen . '")";

$resultado=mysqli_query($miconexion, $miconsulta);

/* Cerramos conexión*/
mysqli_close($miconexion);

echo "<br/> Se ha agregado el comentario con éxito.<br/><br/>";

```

```

?>

<a href="Formulario.php">Añadir nueva entrada</a>
<a href="Mostrar_Blog.php">
</body>
</html>

```

Nueva entrada

Título:	<input type="text" value="Ejemplo de entrada"/>
Este es un ejemplo de la entrada de hoy	
Comentarios:	
Seleccione una imagen con tamaño inferior a 2 MB	
<input type="button" value="Seleccionar archivo"/> atardecer.jpg	
<input type="button" value="Enviar"/>	
Página de visualización del blog	

No hay error en la transferencia del archivo.
 El archivo atardecer.jpg Se ha copiado en el directorio de imágenes
 Se ha agregado el comentario con éxito.

[Añadir nueva entrada](#) [Ver blog](#)

+ Opciones					
	Id	Título	Fecha	Comentario	Imagen
<input type="checkbox"/>  Editar  Copiar  Borrar	1	Ejemplo de entrada	2016-11-07 10:13:08	Este es un ejemplo de la entrada de hoy	atardecer.jpg
<input type="checkbox"/> Marcar todos Para los elementos que están marcados:  Cambiar  Borrar  Exportar					
Número de filas: <input type="button" value="25"/>					

  proyecto_blog	
  imágenes	
 atardecer.jpg	28..
 cielo.jpg	90Ki
 Formulario.php	2Ki
 Insertar Contenido.php	3Ki
  SC	
muestra 8 más datos	

Página que muestra el blog: Mostrar Blog.php

```
y errores de sintaxis.
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>

<body>

<h2>Blog</h2>
<hr/>

<?php

$miConexion=mysqli_connect("localhost", "root", "", "bbddBlog");

/*Comprobación de conexión */

if(!$miConexion){

    echo "La conexión ha fallado: " . mysqli_error();

    exit();

}

$miConsulta="SELECT * FROM CONTENIDO ORDER BY FECHA DESC";

if($resultado=mysqli_query($miConexion, $miConsulta)){

    while($registro=mysqli_fetch_assoc($resultado)) {

        echo "<h3>" . $registro['Titulo'] . "</h3>";

        echo "<h4>" . $registro['Fecha'] . "</h4>";

        echo "<div style='width:400px'>" . $registro['Comentario'] . "</div><br/><br/>";

        if($registro['Imagen']!=""){

            echo "<img src='imagenes/" . $registro['Imagen'] . "' width='300px' />";

        }

        echo "<hr/>";

    }

}

?>
</body>
```

localhost/Curso%20PHP/proyecto_blog/Mostrar%20Blog.php

Aplicaciones Udemy - Online Cours Analytics - YouTube Píldoras Informáticas | Productos y servicios Most Visited Getting Started Google am

Blog

Segunda entrada +

2016-11-07 10:14:31

Este es el comentario de la segunda entrada del blog



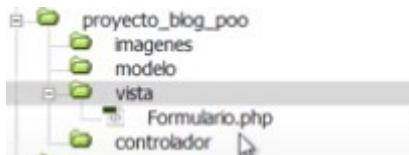
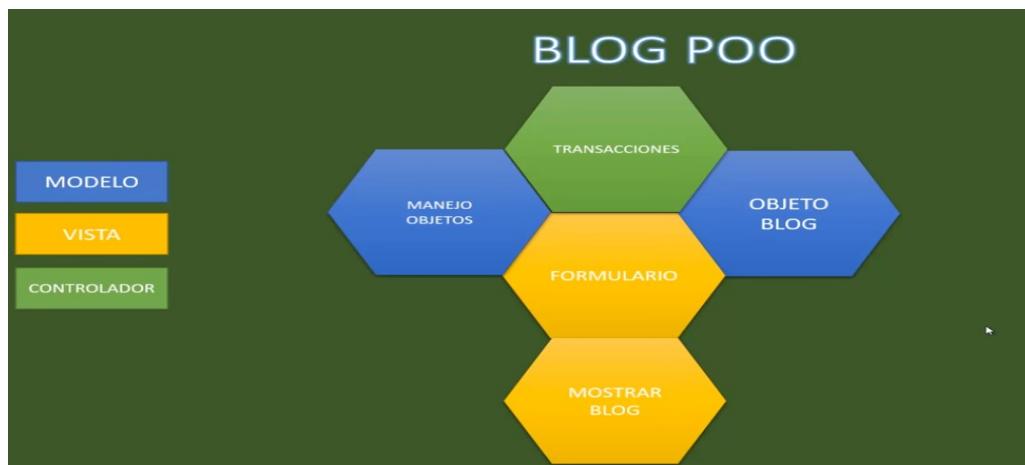
Ejemplo de entrada 

2016-11-07 10:13:08

Esto es un ejemplo de la entrada de hoy



BLOG CON PROGRAMACIÓN ORIENTADA A OBJETOS.



El archivo Formulario es similar, sólo cambia la siguiente línea (el archivo al que apunta):

```
<form action="../controlador/Transacciones.php" method="post" enctype="multipart/form-data" name="form1">
```

Archivo Objeto_Blog.php (dentro de la carpeta modelo):

```
<?php

class Objeto_Blog{

    private $id;
    private $Titulo;
    private $Fecha;
    private $Comentario;
    private $Imagen;

    public function getId(){

        return $this->id;

    }

    public function setId($id){

        $this->id=$id;

    }

    public function getTitulo(){

        return $this->Titulo;

    }

    public function setTitulo($Titulo){
```

Crearíamos el resto de métodos de acceso (get y set) de la misma manera para el total de las 5 propiedades.

Tanto manipular los objetos Blog (getters y setters), como la base de datos, para insertar la información como para extraer la información en la BD.

Manejo_Objetos.php (en la carpeta modelo):

// Para establecer la conexión con la BD:
// El argumento de la función setConexion -> (PDO \$conexion) viene de archivo transacciones(controlador) que todavía no hemos creado

```
<?php
```

```
include_once("Objeto_Blog.php");

class Manejo_Objetos{

    private $conexion;
    public function __construct($conexion) {
        $this->setConexion($conexion);
    }

    public function setConexion(PDO $conexion) {
        $this->conexion=$conexion;
    }

    public function getContenidoPorFecha() {
        $matriz=array();
        $contador=0;
        $resultado=$this->conexion->query("SELECT * FROM CONTENIDO ORDER BY FECHA");
        while($registro=$resultado->fetch(PDO::FETCH_ASSOC)) {
            $blog=new Objeto_Blog();
            $blog->setId($registro["Id"]);
            $blog->setTitulo($registro["Titulo"]);
            $blog->setFecha($registro["Fecha"]);
            $blog->setComentario($registro["Comentario"]);
            $blog->setImagen($registro["Imagen"]);
            $matriz[$contador]=$blog;
            $contador++;
        }
        return $matriz;
    }

    public function insertaContenido(Objeto_Blog $blog) {
        $sql="INSERT INTO CONTENIDO (Titulo, Fecha, Comentario, Imagen) VALUES ('" . $blog->getTitulo() . "','" . $blog->getFecha() . "','" . $blog->getComentario() . "','" . $blog->getImagen() . "')";
        $this->conexion->exec($sql);
    }
}
```

?>

Archivo Transacciones.php (dentro de la carpeta controlador):

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>

<body>

<?php

    include_once("../modelo/Objeto_Blog.php");
    include_once("../modelo/Manejo_Objetos.php");
    try{
        $miconexion=new PDO('mysql:host=localhost; dbname=bbddBlog', 'root', '');
        $miconexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        if($_FILES['imagen']['error']){
            switch($_FILES['imagen']['error']){
                case 1: // Error exceso de tamaño de archivo en php.ini
                    echo "El tamaño del archivo excede lo permitido por el servidor";
                    break;
                case 2: //Error tamaño archivo marcado desde formulario
                    echo "El tamaño del archivo excede 2 MB";
                    break;
                case 3: //Corrupción de archivo
                    echo "El envío de archivo se interrumpió";
                    break;
                case 4: // No hay fichero
                    echo "No se ha enviado ningún archivo";
                    break;
            }
        }else{
            echo "No hay error en la transferencia del archivo. <br/>";
            if((!isset($_FILES['imagen']) || ($_FILES['imagen']['error'] == UPLOAD_ERR_OK)) ||
                $destino_de_ruta='.../imágenes/';

            move_uploaded_file($_FILES['imagen']['tmp_name'], $destino_de_ruta . $_FILES['imagen']['name']);
            echo "El archivo " . $_FILES['imagen']['name'] . " Se ha copiado en el directorio de imágenes";
        }
    }

    $Manejo_Objetos=new Manejo_Objetos($miconexion);
    $blog=new Objeto_Blog();
    $blog->setTitulo(htmlentities(addslashes($_POST["campo_titulo"])), ENT_QUOTES));
    $blog->setFecha(Date("Y-m-d H:i:s"));
    $blog->setComentario(htmlentities(addslashes($_POST["area_comentarios"])), ENT_QUOTES));
    $blog->setImagen($_FILES["imagen"]["name"]);

    $Manejo_Objetos->insertaContenido($blog);

    echo "<br/> Entrada de blog agregada con éxito <br/>";


```

```

} catch (Exception $e) {

    die("Error: " . $e->getMessage());
}

```



Mostrar_Blog (en la carpeta vista):

```

<body>

<?php

    include("../modelo/Manejo_Objetos.php");
    try{
        $miconexion=new PDO('mysql:host=localhost; dbname=bbddBlog', 'root', '');
        $miconexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $Manejo_Objetos=new Manejo_Objetos($miconexion);
        $stabla_blog=$Manejo_Objetos->getContenidoPorFecha();
        if(empty($stabla_blog)){
            echo "No hay entradas de blog aún";
        }else{
            foreach($stabla_blog as $valor){
                echo "<h3>" . $valor->getTitulo() . "</h3>";
                echo "<h4>" . $valor->getFecha() . "</h4>";
                echo "<div style='width:400px'>";
                echo $valor->getComentario() . "</div>";
                if($valor->getImagen()!=""){
                    echo "<img src='../imagenes/'";
                    echo $valor->getImagen() . "' width='300px' height='200px' />";
                }
                echo "<hr/>";
            }
        }
    }catch(Exception $e){
        die("Error: " . $e->getMessage());
    }
?>
</body>

```

PERFILES DE USUARIO Y MENÚS.

The screenshot shows the MySQL Workbench interface with a database table and a file browser. The table has four rows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	Id	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
2	Usuario	varchar(30)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
3	Password	varchar(30)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
4	Perfil	varchar(15)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más

The file browser shows a folder named 'profiles usuario' containing the following files:

- datosConexion.php
- Formulario_inserta_perfiles.php
- Formulario_login_perfiles.php
- insertar_registros_perfiles.php
- página_datos_perfiles.php
- menu_administrador.html
- menu_usuarios.html

Formulario_inserta_perfiles.php:

```
<!doctype html>
<html>

    <head>

        <meta charset="utf-8">
        <title>Documento sin título</title>

    </head>

    <body>

        <form action="insertar_registros_perfiles.php" method="get">

            <p>
                <label>Usuario:</label>
                <input type="text" name="usu">
            </label>
            <br>
            <label>Contraseña:</label>
            <input type="text" name="con">
            </label>
            </p>
            <p>Perfil:<br>
                <label for="perfil"></label>
                <select name="perfil" id="perfil">
                    <option>administrador</option>
                    <option>usuario</option>
                </select>
            </p>
            <p>
                <input type="submit" name="enviando" value="Registro">
            </p>
        </form>
    </body>
```

Usuario:

Contraseña:

Perfil:

Formulario_login_perfiles.php:

```
<!doctype html>
<html>

    <head>
        <meta charset="utf-8">
        <title>Documento sin título</title>
    </head>
    <body>
        <form action="pagina_datos_perfiles.php" method="get">
            <p>
                <label>Usuario:<br>
                    <input type="text" name="usu">
                </label>
                <br>
                <label>Contraseña:<br>
                    <input type="text" name="con">
                </label>
            </p>
            <p>
                <input type="submit" name="enviando" value="Login">
            </p>
        </form>
    </body>
</html>
```

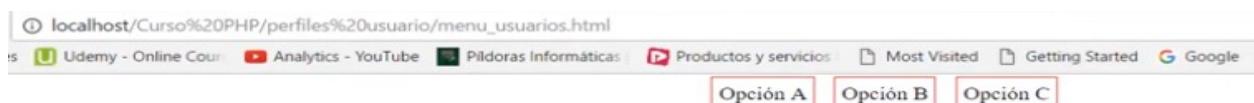
Usuario:

Contraseña:



menu_usuarios.html:

```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Documento sin título</title>
        <style>
            ul{
                list-style:none;
                width:400px;
                margin:auto;
            }
            li{
                display:inline;
                margin:0 5px;
                /*background-color:#F00; */
                padding:5px;
                border:solid 1px #FF0000;
            }
        </style>
    </head>
    <body>
        <ul>
            <li>Opción A</li>
            <li>Opción B</li>
            <li>Opción C</li>
        </ul>
    </body>
</html>
```



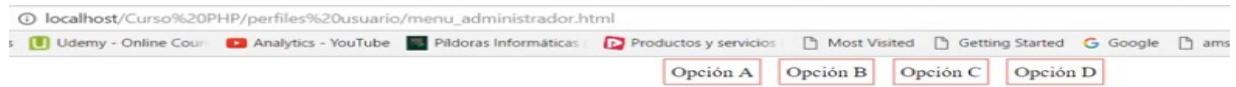
menu_administrador.html:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
<style>

    ul{
        list-style:none;
        width:400px;
        margin:auto;
    }

    li{
        display:inline;
        margin:0 5px;
        /*background-color:#FO0; */
        padding:5px;
        border:solid 1px #FF0000;
    }

</style>
</head>
<body>
<ul>
<li>Opción A</li>
<li>Opción B</li>
<li>Opción C</li>
<li>Opción D</li>
</ul>
</body>
</html>
```



pagina_datos_perfiles.php:

```
<!doctype html>
<html>

<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>
<body>
<?php

$usuario = $_GET["usu"];
$password = $_GET["con"];
require ("datos_conexion.php");
$cexion=mysqli_connect($db_host, $db_usuario, $db_contra);
if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}
mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion, "utf8");

$consulta="SELECT USUARIO, PASSWORD, PERFIL FROM PERFILEUSUARIOS WHERE USUARIO = ? AND PASSWORD= ?";
echo "<br><br>";
```

```

// $resultados=mysqli_query($conexion, $consulta);
    |
$resultados=mysqli_prepare($conexion,$consulta);
    |
$ok=mysqli_stmt_bind_param($resultados, 'ss', $usuario, $password);
    |
$ok=mysqli_stmt_execute($resultados);
    |
if($ok==false){
    echo "Error en la consulta";
} else{
    $ok=mysqli_stmt_bind_result($resultados,$usuario,$password, $perfil);
    |
}
    |

while(mysqli_stmt_fetch($resultados)){
    echo "Hola " . $usuario . "<br>";
    echo "Tu perfil es " . $perfil. "<br>";
}

```

Hasta ahora el archivo anterior lo único que nos dice al loguearnos es si somos usuario o administrador.

```

if($perfil=="administrador"){
    |
    include("menu_administrador.html");
    |
}else{
    |
    include("menu_usuarios.html");
    |
}

```

Ahora ya nos mostrará un menú o otro en función de nuestro perfil, con las líneas anteriores.

```

    |
    mysqli_stmt_close($resultados);
    mysqli_close($conexion);
    |
    ?>
    |
</body>
    |
</html>

```

insertar_registros_perfiles.php:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin titulo</title>
</head>
<body>
<?php

$usu = $_GET["usu"];
$con = $_GET["con"];
$perfil = $_GET["perfil"];

require ("datos_conexion.php");

$conexion=mysqli_connect($db_host, $db_usuario, $db_contra);

if(mysqli_connect_errno()){

echo "Fallo al conectar con la BBDD";

exit();
}

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");

mysqli_set_charset($conexion, "utf8");

$sql="INSERT INTO PERFILUSUARIOS (USUARIO, PASSWORD, PERFILE) VALUES (?,?,?)";

$resultado=mysqli_prepare($conexion, $sql);

$ok=mysqli_stmt_bind_param($resultado, "sss", $usu, $con, $perfil);

$ok=mysqli_stmt_execute($resultado);

if($ok==false){

echo "Error al ejecutar la consulta";

}else{

// $ok=mysqli_stmt_bind_result($resultado, $codigo, $seccion, $precio, $pais);

echo "Agregado nuevo registro";

/*while(mysqli_stmt_fetch($resultado)) {

echo $codigo . " " . $seccion . " " . $precio . " " . $pais . "<br>";

}

 */

}

mysqli_stmt_close($resultado);

?>

</body>
</html>

```

- DataTables
- Makeeditable databases
- Perfiles de usuario
- Login con token
- Chat PHP
- Envío correos electrónicos
- Restablecimiento de contraseñas con PHP
- Chats
- Tags
- Largo etc.

- Laravel
- Codeigniter
- Phalcon
- Symfony
- Largo etc.