

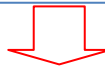
1. Utilizar el campo de tipo BLOB para subir archivos a una BD. Un campo BLOB permite almacenar un fichero. En este caso, partiremos de una nueva base de datos, obras_arte2, con una única tabla y la siguiente estructura:

```
mysql> desc obras;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| deno  | varchar(15) | NO   | PRI | NULL    |       |
| archivo | longblob | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.05 sec)
```

Reutilizaremos el código de la práctica guiada 2 e insertaremos una tupla en esta tabla, teniendo en cuenta el nuevo campo de tipo longblob:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>

<!--FORMULARIO -->
<br><br><br>
<table border=1 align="center" bgcolor="#D0F5A9">
<form action="tratamiento.php" method="post" enctype="multipart/form-data"> <!--enctype necesario para trabajar con archivos -->
  <tr><td>Denominación:</td><td><input type="text" name="texto_id"></td></tr>
  <tr><td>Imagen: </td><td><input type="file" name="fichero" size="20"></td></tr>
  <tr><td colspan="2"><input type="submit" name="Subir archivo"></td></tr>
</form>
</table>
</body>
</html>
```

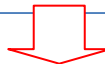


```
<?php

/* $_FILES es una variable superglobal que almacenará datos de la imagen recibida:
   name, almacena el nombre del fichero
   type, el tipo o extensión (se podrá limitar el tipo de archivos a subir)
   size, el tamaño (se podrá limitar el tamaño de los archivos a subir)
   tmp_name, el nombre del directorio temporal utilizado de forma previa al alojamiento definitivo del fichero
   que en este ejemplo será está en $dir
   error, si hay algún error*/
$nomf=$_FILES['fichero']['name'];
$t=$_FILES['fichero']['size'];
var_dump($t);
echo "Fichero subido: ".$nomf;
echo "<br>";
$dir=$_SERVER['DOCUMENT_ROOT'].'/uploads/'; //será necesario crear el directorio uploads en la raíz del servidor web
var_dump($dir);
if ($t<1000000)
  move_uploaded_file($_FILES['fichero']['tmp_name'],$dir.$nomf);
else
  echo "Demasiado grande";

include ("accesodb.php"); //acceder a la BD para insertar un registro

?>
```



```
<?php

$db_host="localhost";
$db_usuario="root";
$db_clave="";
$db_nombre="obras_arte2";

$conexion=mysqli_connect($db_host,$db_usuario,$db_clave);
if (mysqli_connect_errno())
{
    echo "Fallo en la conexi&#243n <br>";
    exit();
}
else
    echo "conexi&#243n establecida <br>";

mysqli_select_db ($conexion,$db_nombre) or die ("No se encontr&#243 la BD");
$deno=$_POST['texto_id'];
$nombre=$dir.$nomf; //ruta del fichero cuyo contenido se insertará en la BD
$f=fopen($dir.$nomf,"r"); //el fichero tiene que ser abierto en modo lectura
var_dump($dir);
$contenido=fread($f,$t);//se lee su contenido
var_dump($contenido);
$contenido=addslashes($contenido);//para un escapado correcto de los datos
fclose($f);
$insertar="insert into obras (deno,archivo) value ('$deno','$contenido')"; //otra forma de realizar una consulta
$resultado=mysqli_query($conexion,$insertar);
if (mysqli_affected_rows($conexion)>0)
    echo "Se ha insertado correctamente la insercion";
else
    echo "Error; no se ha realizado la insercion";
mysqli_close($conexion);
?>
```

Ejecución: Se insertan los datos en el formulario, se sube el fichero a /uploads y después, se inserta la tupla en la tabla "obras".

Denominación:	tatus
Imagen:	Seleccionar archivo tatus.jpg
<input type="button" value="Enviar"/>	

Desde PHPMYADMIN

	deno	archivo
<input type="checkbox"/>	tatus	[BLOB - 275.6KB]

Para recuperar la información de la base de datos, realizaremos una consulta y visualizaremos la imagen del fichero:

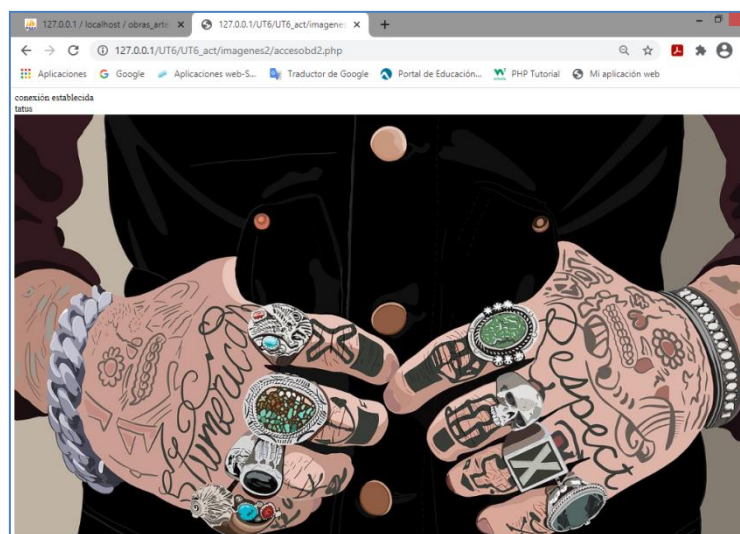
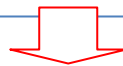
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>

<?php

$db_host="localhost";
$db_usuario="root";
$db_clave="";
$db_nombre="obras_arte2";

$conexion=mysqli_connect($db_host,$db_usuario,$db_clave);
if (mysqli_connect_errno())
{
    echo "Fallo en la conexi&#243n <br>";
    exit();
}
else
    echo "conexi&#243n establecida <br>";

mysqli_select_db ($conexion,$db_nombre) or die ("No se encontr&#243 la BD");
$consulta="select * from obras where deno='tatus'";
$resultados=mysqli_query($conexion,$consulta);
$fila=mysqli_fetch_array($resultados);
$deno=$fila[0];
echo $deno;
$fichero=$fila[1];
mysqli_close($conexion);
//es necesario utilizar la función base64_encode ya que los datos del fichero están
//almacenados utilizando la codificación base64
echo "<img src='data:image/jpg; base64,\". base64_encode($fichero).\"'>";
?>
</body>
</html>
```



(fuente:<https://pixabay.com/es/illustrations/tatuaje-anillos-manos-respeto-2803482/>)

2. Inserta las portadas de los libros utilizando un campo BLOB, siguiendo la actividad anterior.
3. Registro y acceso a una BBDD. En este ejercicio, se utilizará una BD, denominada usuarios, con una única tabla con la siguiente estructura:

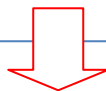
Field	Type	Null	Key	Default	Extra
id_usu	int(11)	NO	PRI	NULL	auto_increment
nombre	varchar(15)	NO		NULL	
clave	varchar(60)	NO		NULL	
rol	varchar(15)	NO		NULL	

Básicamente se trata de registrar usuarios en la BD, dando la opción de encriptar su clave. Para ello utilizaremos las funciones hash de PHP, donde el algoritmo de encriptación es *blogfish* (hay otros algoritmos de encriptación obsoletos como MD5, SHA-1,...debido a la facilidad de crakeo derivada de la información proporcionada por las tablas Rainbow (puedes consultar la información oficial de php para ampliar conocimientos: <https://www.php.net/manual/es/faq.passwords.php#faq.passwords.fasthash>). La función `password_hash` genera "la sal" de forma automática (también está en desuso la función `crypt()` la cual requiere de un proceso de generación de la sal, programado; en la documentación de php, por ello, se recomienda utilizar la librería nativa de php).

Para el proceso de registro, se utiliza el siguiente código (*registro.php*):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>

<!--FORMULARIO -->
<br><br><br>
<table border=1 align="center" bgcolor="#D0F5A9">
<form action="tratamiento.php" method="post">
  <tr><td>Nombre usuario:</td><td><input type="text" name="usuario"></td></tr>
  <tr><td>Password:</td><td><input type="text" name="password"></td></tr>
  <tr><td>
    Elegir:<br>
    <input type="radio" name="c" value=1>Cifrar password<br>
    <input type="radio" name="c" value=2>No cifrar password<br>
  </td></tr>
  <tr><td colspan="2"><input type="submit" name="Registrar usuario"></td></tr>
</form>
</table>
</body>
</html>
```



Este es el tratamiento realizado (*tratamiento.php*):

```
<?php

$db_host="localhost:3308";
$db_usuario="root";
$db_clave="";
$db_nombre="usuarios";

$conexion=mysqli_connect($db_host,$db_usuario,$db_clave);
if (mysqli_connect_errno())
{
    echo "Fallo en la conexión#243n <br>";
    exit();
}
else
    echo "conexión#243n establecida <br>";

mysqli_select_db ($conexion,$db_nombre) or die ("No se encontró#243n la BD");
$elegir=htmlentities(addslashes($_POST['c']));
$nombre=htmlentities(addslashes($_POST['usuario']));
$password=htmlentities(addslashes($_POST['password']));
if ($elegir=='1')
    $password=password_hash($password,PASSWORD_DEFAULT);
else
    $password=$password;
$insertar="insert into usu (nombre,clave, rol) values ('$nombre','$password','usuario')"; //el campo clave será un varchar(60)
$resultado=mysqli_query($conexion,$insertar);
if (mysqli_affected_rows($conexion)>0)
    echo "Se ha insertado correctamente la inserción";
else
    echo "Error; no se ha realizado la inserción";
mysqli_close($conexion);
?>
```

Tras el registro de los usuarios, la BD, en concreto, la tabla "usu", quedaría así:

+ Opciones			id_usu	nombre	clave	rol
<input type="checkbox"/>	Editar	<input type="checkbox"/>	1	esther	12345	usuario
<input type="checkbox"/>	Editar	<input type="checkbox"/>	14	javier	\$2y\$10\$XFX2Cc/pW1IFiY8xFNmniO97LJBvoQcgX7izm1s4zah...	usuario
<input type="checkbox"/>	Editar	<input type="checkbox"/>	15	victor	\$2y\$10\$liM/KBB8eCK9JP2QyCapJucv4EQzwmRK2alU3rRfC/...	usuario

Una vez realizado el registro, se ejecuta el acceso (*acceso.php*):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>

<!--FORMULARIO -->
<br><br>
<table border=1 align="center" bgcolor="#D0F5A9">
<form action="autorizar.php" method="post">
    <tr><td>Nombre usuario:</td><td><input type="text" name="usuario"></td></tr>
    <tr><td>Password:</td><td><input type="text" name="password"></td></tr>
    <tr><td colspan="2"><input type="submit" name="Registrar usuario"></td></tr>
</form>
</table>
</body>
</html>
```

Nombre usuario: esther

Password: 12345

Enviar

Acceso que será denegado

Nombre usuario: javier

Password: 12345

Enviar

Acceso que será autorizado

Y el tratamiento, está recogido en el siguiente código (*autorizar.php*):

```
<?php

$db_host="localhost:3308";
$db_usuario="root";
$db_clave="";
$db_nombre="usuarios";

$conexion=mysqli_connect($db_host,$db_usuario,$db_clave);
if (mysqli_connect_errno())
{
    echo "Fallo en la conexi&#243n <br>";
    exit();
}
else
    echo "conexi&#243n establecida <br>";

mysqli_select_db ($conexion,$db_nombre) or die ("No se encontr&#243 la BD");
$nombre=htmlspecialchars(addslashes($_POST['usuario']));
$pass=htmlspecialchars(addslashes($_POST['password']));
$consultar="select * from usu where nombre='$nombre'"; //el campo clave será un varchar(60)
$resultado=mysqli_query($conexion,$consultar);
$comprobar=0; //controla los usuarios con clave no encriptada
$comprobar1=0; //controla los usuarios con clave encriptada
if (mysqli_affected_rows($conexion)>0)
    $comprobar=1;
    while ($fila=mysqli_fetch_array($resultado))
        if (password_verify($pass, $fila['clave']))
            $comprobar1=1;
if ($comprobar==1 || $comprobar1==1)
    echo "Acceso autorizado".$fila['nombre'];
else
    echo "Acceso denegado. No está registrado";

mysqli_close($conexion);
?>
```

conexión establecida
Acceso denegado. No está registrado

conexión establecida
Acceso autorizado

- Realiza un ejercicio similar al anterior. Como modificación te puedes plantear un menú previo con registro y con acceso.
- Utilizando la POO, programar una conexión y realización de una consulta a una BD. Será para ello preciso, crear nuestras propias clases.

El código, podría ser el siguiente:

Partimos de un archivo *index.php* desde el cual se instancia un objeto asociado a una consulta:

```
<?php
echo "pasa";
require ("bd_class.php");
$objeto=new Consultas();
$alumnos=$objeto->GetDatos(); //$alumnos será un array
echo "<table border=1>";
echo "<tr><th>ID ALUMNO</th><th>NOMBRE</th><th>EDAD</th><th>CURSO</th></tr>";
foreach ($alumnos as $al)
{
    echo "<tr><td>".$al['id_al']."</td>";
    echo "<td>".$al['nombre']."</td>";
    echo "<td>".$al['edad']."</td>";
    echo "<td>".$al['codigo']."</td></tr>"; //en la BD desde la que probé, no tenía datos
}
echo "</table>";
?>
```

En el siguiente archivo, se detalla la codificación de las clases definidas, para este pequeño ejemplo de POO para acceder a una BD:

```
<?php
//ejercicio previo al MVC
require("datos_config.php");
class Conexiones {

    protected $bd_conect; //es un objeto, el mismo que hemos manejado en ejercicios anteriores para crear una conexión utilizando
                           //la extensión mysqli con POO
    public function __construct()
    {
        $this->bd_conect=new mysqli(BD_H,BD_U,BD_PW,BD_NOM);

        if ($this->bd_conect->connect_errno)
            echo "No se ha podido establecer la conexión";
        else
            echo "Conexión establecida";

        $this->bd_conect->set_charset(BD_C); //para establecer el código UTF-8
    }

}

class Consultas extends Conexiones {

    public function __construct() {

        parent::__construct();

    }

    public function GetDatos(){

        $resultado=$this->bd_conect->query('select * from alumno');
        $datos=$resultado->fetch_all(MYSQLI_ASSOC);
        return $datos;

    }

    //la visualización que se realiza en index, debería ser un nuevo método Visualizar().
}
?>
```

```
<?php
const BD_H="localhost:3308"; //o la función define (BD_H,"localhost:3308");
const BD_U="root";
const BD_PW="";
const BD_NOM="ciclo";
const BD_C="UTF8";
?>
```

6. Crea una conexión con alguna de tus BBDD y diseña las clases correspondientes.
7. En la UT 5, vimos la posibilidad de comprobar el acceso a una aplicación web, utilizando LDAP, un servidor de otra entidad, Google, por ejemplo, y es precisamente el objetivo de esta actividad, pero utilizando una BD creada en el servidor MySQL. Será necesario almacenar los usuarios con acceso a las aplicaciones web. Los usuarios serán almacenados en una BD en el proceso de registro que ofrezca la aplicación web.

Para ello una aplicación web, ofrecerá la opción de registro y de acceso. Si se accede a acceso, se informará de la necesidad de registrarse previamente. Para el *registro*, se solicitará, usuario, contraseña y correo electrónico. Para el acceso, el usuario y la contraseña.

En el proceso de registro de un usuario, en la BD usuarios, se almacenará, nombre del usuario, contraseña, email, rol (con un valor por defecto, "usuario") un fichero con los nombres de las aplicaciones web a las que el usuario tiene acceso y una foto (opcional en el momento del registro); desde la aplicación web, podrá modificar su perfil, más adelante, subiendo una foto.

En el proceso de acceso:

- si el usuario tiene rol "admin" y clave "12345", se mostrará un menú (CRUD, Create, Read, Update, Delete) sobre una aplicación web concreta (este usuario inicialmente ya existe en la BD usuarios, no requiere de registro previo; el campo rol será "administrador" y el fichero que almacena los nombres de las aplicaciones web, se irá actualizando a medida que crezcan las aplicaciones web).
- si el usuario tiene rol "usuario", sólo podrá consultar datos de la BD que maneja la aplicación web a la cual ha accedido.