

# PHP MySQLi

Antonio Boronat Pérez

IES Joan Coromines (Benicarló)



## Funciones de PHP con MySQL.

Una vez que conocemos las características básicas de MySQL y las sentencias SQL que se pueden ejecutar, vamos a ver como se integra MySQL con PHP.

Las definiciones de bases de datos y tablas las crearemos desde la consola de MySQL, preferiblemente a través de un script SQL con las sentencias de creación e inicialización de tablas, o mejor mediante algún asistente web como PHPMyAdmin. La manipulación de datos en cambio se realizará desde PHP. Por lo tanto, las operaciones que nos interesan son:

- Leer datos.
- Añadir datos.
- Modificar datos.
- Eliminar datos.

PHP accede a MySQL a través de API, hay tres, pero PHP7 sólo soporta [MySQLi](#). Dispone de una extensión o biblioteca de funciones con múltiples funciones, que nos permiten acceder a la base de datos.

## Conectar con la base de datos.

Antes de empezar a consultar información tenemos que conectar con el sistema gestor de bases de datos y declarar la base de datos que vamos a usar. Esto se deberá hacer en cualquier aplicación que posteriormente quiera leer, añadir, modificar o eliminar datos.

### **mysqli\_connect**

Esta función establece una conexión con la base de datos devolviendo un objeto de la clase *mysqli* que usaremos para nuestra sesión, si se produce un error devuelve el valor *false*:

```
$conexion_bd = @mysqli_connect(Servidor, Usuario, Contraseña, [Base de datos], [Puerto]);
```

Los parámetros que recibe son:

- *Servidor*: nombre o dirección IP del servidor donde está alojada la base de datos. Si está instalado en la misma máquina que el servidor web, lo normal es que sea *localhost*. Se

---

1 @ es opcional, suprime los mensajes de error por si queremos tratarlos de otra manera.

puede elegir un número de puerto diferente del que se usa por defecto indicándolo en la misma función.

- *Usuario*: nombre de usuario con privilegios para acceder a la base de datos.
- *Contraseña*: contraseña del usuario.
- *Base de datos* indicaría, en caso de ponerla, la base de datos dentro del SGDB con la que vamos a trabajar. Si no se establece en la conexión se seleccionará como se muestra más adelante.
- *Puerto*, el puerto en que atiende el servidor si no es el definido por defecto.

La función `mysqli_connect()` está sobrecargada de forma que podemos elegir los parámetros que le pasamos en la llamada, de hecho si la ejecutamos sin ningún parámetro establecerá la conexión según una configuración por defecto de PHP, que se establece en el fichero `/etc/php/7.0/apache2/php.ini`.

## **mysqli\_close**

Esta función cierra una conexión con la base de datos, devolviendo un valor booleano que expresa si el cierre se ha producido correctamente o no.

```
$cierre_bd = @mysqli_close($conexion_bd);
```

```
if(!$cierre_bd)
```

```
    echo "Error en el cierre de la conexión a la BD.";
```

## **mysqli\_select\_db**

Con `mysqli_select_db` podemos seleccionar la base de datos si no la indicamos en la operación de conexión o bien la podemos cambiar para usar otra. Esta función devuelve un valor booleano que indica si la operación fue correcta.

```
$seleccion_bd = mysqli_select_db($conexion_bd, Base de datos);
```

```
if(!$seleccion_bd)
```

```
    echo "Error al seleccionar la BD.";
```

## Funciones estado de la conexión a BD

Estas funciones permiten acceder a los valores de una conexión a una base de datos:

Para obtener información sobre la conexión usaremos:

```
$info_con = mysqli_get_host_info($conexión_bd);
```

Mientras que para recuperar las características del servidor MySQL tenemos:

```
$info_serv = mysqli_get_server_info($conexión_bd);
```

Las funciones siguientes nos ayudarán en la gestión de errores producidos en la conexión con la base de datos:

La primera nos dará número de error producido o 0 si no lo hay.

```
$num_error = mysqli_connect_errno();
```

La siguiente función nos devuelve una cadena con el error producido o cadena vacía.

```
$msg_error = mysqli_connect_error();
```

Ejemplo de uso de las funciones:

```
<?php
/* Ejemplo de conexión a una base de datos MySQL
 * Modificando los parámetros de conexión podemos comprobar
 * la sobrecarga de la función de conexión, quitando el
 * BASE_DATOS y luego usando la función mysqli_select_db
 */

DEFINE ("SERVIDOR", "localhost");
DEFINE ("USER", "root");
DEFINE ("PASSWD", "root");
DEFINE ("BASE_DATOS", "projectes");

echo "Conexión DB:<br>";

$conexion_bd = @mysqli_connect(SERVIDOR, USER, PASSWD, BASE_DATOS);
if ($conexion_bd) {
    echo "Conexion correcta <br>";
    $info_con = mysqli_get_host_info($conexion_bd);
    echo "Información de conexión: " . $info_con . "<br>";
    $info_serv = mysqli_get_server_info($conexion_bd);
    echo "Información del servidor: " . $info_serv . "<br>";
    //Cierre de la conexión a la BD
    $cierre_bd = @mysqli_close($conexion_bd);
    if($cierre_bd){
        echo "Desconexión de la BD correcta <br>";
    }
}
```

```
        else {
            echo "Error al cerrar la conexión a la BD <br>";
        }
    }
    else {
        echo "Error de connexion con la BD<br>";
        $num_error = mysqli_connect_errno();
        echo "Número de error: " . $num_error . "<br>";
        $msg_error = mysqli_connect_error();
        echo "Mensaje error de conexión: " . $msg_error . "<br>";
    }
?>
```

## Acceso a datos

Las operaciones relativas al acceso a datos las ejecutaremos mediante la función:

```
$res = mysqli_query($conexión_db, sentencia sql);
```

el resultado devuelto es, para una consulta un objeto de tipo *mysqli\_result* o *false* si se produce un error y si la operación realizada es, por ejemplo un *insert*, el valor será booleano indicando el estado de finalización del proceso.

En caso de error podemos usar las funciones *mysqli\_errno* y *mysqli\_error* para documentar la situación producida.

## Consultas

Para conocer el número de líneas recuperadas en la consulta:

```
$num_col = mysqli_num_rows($res);
```

Las filas devueltas en la consulta se puede obtener usando varias funciones:

```
$res_array = mysqli_fetch_array($res);
```

esta función devuelve una línea de resultado y avanza el descriptor a la siguiente, el valor devuelto es un array en el que cada elemento corresponde a una columna de la fila actual. El índice del array puede ser tanto un entero o el nombre del campo de la tabla de la base de datos. Por ejemplo:

```
echo "El identificador es: ". $res_array[0]; o bien: echo "El identificador es: ". $res_array[id];
```

```
$res_array = mysqli_fetch_assoc($res);
```

como la función anterior, devuelve un array con la línea actual y avanza el descriptor. El array es asociativo y el índice del array es el nombre de la columna.

```
echo "El identificador es: ". $res_array[id];
```

```
$res_array = mysqli_fetch_row($res);
```

esta función es como las dos anteriores pero el índice de acceso al array es numérico. Por ejemplo:

```
echo "El identificador es: ". $res_array[0];
```

```
$res_obj = mysqli_fetch_object($res);
```

con esta función la línea es devuelta en un objeto en el que cada atributo es una columna y su nombre corresponde al de la columna de la tabla. Como las anteriores, cada vez que se ejecuta avanza el descriptor a la línea siguiente del resultado. Por ejemplo:

```
echo "El identificador es: ". $res_obj->id;
```

Las funciones anteriores devuelven el valor *NULL* cuando ya no quedan más líneas en el resultado de la consulta.

```
$res_array = mysqli_fetch_all($res[,MYSQLI_ASSOC o MYSQLI_NUM o MYSQLI_BOTH]);
```

Esta función devuelve todas las líneas en un array de dos dimensiones, el primer índice es entero y corresponde al número de línea, mientras que el segundo podrá ser entero (*MYSQLI\_NUM*) que es el valor por defecto en esta función, asociativo (*MYSQLI\_ASSOC*) donde la segunda dimensión se accede mediante el nombre de la columna y finalmente con *MYSQLI\_BOTH* se puede acceder mediante los dos valores entero o nombre de columna como en *mysqli\_fetch\_array*

```
<?php
include "funcion_conexion_bd.php";

DEFINE ("SERVIDOR", "localhost");
DEFINE ("USER", "root");
DEFINE ("PASSWD", "root");
DEFINE ("BASE_DATOS", "projectes");

$con_bd = conexion_bd(SERVIDOR, USER, PASSWD);
// Con esta no usamos mysqli_select_db
// $con_bd = conexion_bd(SERVIDOR, USER, PASSWD, BASE_DATOS);

$cambio_ok = mysqli_select_db($con_bd, BASE_DATOS); //Selecciona la BD
if($cambio_ok === TRUE){
    $sql = "SELECT * FROM projecte";
    if($res = mysqli_query($con_bd, $sql)) {
        $num_fil = mysqli_num_rows($res);
        echo "Núm. filas resultado: " . $num_fil . "<br>";
        $res_array = mysqli_fetch_array($res); // Acceso fila a fila
        while($res_array){
            print_r($res_array);
            echo "<br><br>";
            $res_array = mysqli_fetch_array($res);
        }
        mysqli_free_result($res);
    }
    else{
        echo "Error en la consulta: " . mysqli_error($con_bd) . "<br>";
    }
}
//Todas la filas y acceso a los campos como array asociativo
```

```
if($res = mysqli_query($con_bd, $sql)) {
    $res_array = mysqli_fetch_all($res, MYSQLI_ASSOC);

    for ($i=0; $i < $num_fil; $i++) {
        foreach($res_array[$i] as $clave => $valor){
            echo $clave . "-->" . $valor . "<br>";
        }
        echo "<br>";
    }
    mysqli_free_result($res);
}
else{
    echo "Error en la consulta: " . mysqli_error($con_bd) . "<br>";
}
}
else{
    echo "Error en el cambio de BD: " . mysqli_error($con_bd) . "<br>";
}
}
mysqli_close($con_bd);
?>
```

Con la siguiente función podemos desplazar el descriptor de acceso al resultado de la consulta a la fila que no interese, por ejemplo, si se ha llegado al final para volver al principio (desplazamiento 0) y reutilizar los datos. El valor devuelto es booleano en función de si se ha ejecutado correctamente.

`$desplazamiento_error = mysqli_data_seek ( $res , $desplazamiento );`

por ejemplo en el código anterior, no sería necesario ejecutar la consulta de nuevo con:

```
mysqli_data_seek($res, 0); // Pone el descriptor al principio
$res_array = mysqli_fetch_all($res, MYSQLI_ASSOC);

for ($i=0; $i < $num_fil; $i++) {
    foreach($res_array[$i] as $clave => $valor){
        echo $clave . "-->" . $valor . "<br>";
    }
    echo "<br>";
}
mysqli_free_result($res);
```

En caso de ejecutar operaciones que modifiquen el contenido de la base de datos, como *insert*, *update* o *delete*, podemos usar la función *mysqli\_affected\_rows* para comprobar el número de filas accedidas o -1 en caso de error.

`$num_filas = mysqli_affected_rows ( $con_bd );`



## Liberar el resultado de una consulta

Cuando ya hemos accedido a la información de una consulta es conveniente liberar el espacio que ocupa con:

```
mysqli_free_result($res);
```

esta función no devuelve ningún resultado y el parámetro es un objeto devuelto por *mysqli\_query*.

## Información columnas de la consulta

Podemos conocer el número de columnas presentes en el resultado de la consulta actual mediante la función:

```
$num_column = mysqli_num_fields ( $res );
```

La siguiente función nos indica el número de la columna en la que se encuentra el descriptor de acceso al resultado de la consulta que estamos tratando:

```
$column = mysqli_field_tell ( $res );
```

Y podemos posicionar el descriptor en la columna que nos interese (para acceder a la primera columna 0) con :

```
$pos_column_error = mysqli_field_seek ($res , $column );
```

Con la siguiente función obtenemos un objeto con toda la información de la estructura de la columna actual, tal como su nombre, tipo de dato, longitud máxima, tabla, base de datos, etc.

```
$obj_column = mysqli_fetch_field ( $res );
```

Los códigos numéricos devueltos en los atributos se pueden consultar en la web de PHP [mysqli fetch field](#) por ejemplo 253 es VARCHAR o 13 YEAR.

Modificando el ejemplo anterior, podemos ver cómo usar la funciones anteriores:

```
$sql = "SELECT * FROM projecte";
if($res = mysqli_query($con_bd, $sql)) {
    // Acceso a las columnas del resultado
    $num_column = mysqli_num_fields($res);
    echo "Número de campos: " . $num_column . "<br>";
    for($i = 0; $i < $num_column ; $i++){
        $obj_column = mysqli_fetch_field($res);
        echo "Columna núm: " . mysqli_field_tell($res) . " Tabla: " .
        $obj_column->table . " nombre campo: " . $obj_column->name . " tipo: " .
        $obj_column->type . "<br>";
    }
    mysqli_free_result($res);
}
else{
    echo "Error en la consulta: " . mysqli_error($con_bd) . "<br>";
}
```