

## **UNIDAD DIDÁCTICA 7:**

### **BBDD, MySQL y PHP.**

**Módulo profesional:**  
**Desarrollo Web en Entorno Servidor**

## Índice

RESUMEN INTRODUCTORIO .....	3
INTRODUCCIÓN .....	3
CASO INTRODUCTORIO .....	3
1. INTRODUCCIÓN AL CONCEPTO DE BASE DE DATOS .....	4
1.1. Base de datos Mysql.....	4
1.2. Servidor Mysql.....	6
1.3. Consideraciones sobre seguridad .....	8
1.4. Bases de datos en la nube .....	8
2. BASES DE DATOS Y PHP.....	9
2.1. Paso 1. Creando una base de datos y una tabla con PHPMYADMIN 10	
2.2. Paso 2. Crear conexión .....	11
2.3. Mysql frente a mysql .....	14
2.4. Paso 3. Lectura de datos.....	14
RESUMEN FINAL .....	16

## RESUMEN INTRODUCTORIO

En esta unidad nos introduciremos en el uso de las bases de datos para el almacenamiento, lectura y modificación permanente de información.

Igual que ocurre en otros lenguajes, resulta imprescindible el uso de las bases de datos para mantener información y poderla utilizar, en este caso de forma individual, pero más delante de forma compartida entre varios usuarios.

## INTRODUCCIÓN

Las bases de datos son el mecanismo para poder organizar y mantener de una forma estructurada y permanente información dentro de una aplicación.

Las aplicaciones Web son aplicaciones que necesitan de una forma intensiva el uso de dichas bases de datos para la realización de cualquier aplicación por muy pequeña que sea.

Al igual que ocurre en otros lenguajes de programación, para poder hacer uso de una base de datos, necesitaremos seguir los pasos de preparación, conexión y definitivamente de uso de las mismas. Pero una vez aprendidos dichos pasos, veremos que es muy fácil el uso de las bases de datos, ya que dentro de la programación web, y en concreto php, todo se resumirá al uso de arrays asociativos para moverse por la información.

## CASO INTRODUCTORIO

Queremos realizar un informe de situación de una pequeña empresa a partir de la información almacenada en la base de datos de la misma.

# 1.INTRODUCCIÓN AL CONCEPTO DE BASE DE DATOS

Igual que ha ocurrido con el resto de conceptos y dentro del desarrollo en servidor, en nuestro caso con PHP, la creación de una aplicación depende de parámetros y tecnologías que se integran, se complementan o utilizan el PHP:

- PHP utiliza HTTP para la recepción y devolución de la información, por ello introducimos cómo funciona esta arquitectura. El modelo cliente servidor, la interpretación por parte del servidor Apache y por último los métodos POST y GET, son conceptos que hemos estudiado sobre la arquitectura HTTP.
- PHP se ejecuta junto y por encima del lenguaje HTML. PHP puede escribir HTML como se nutre de los formularios de HTML para recibir información y actuar de acuerdo a ella.

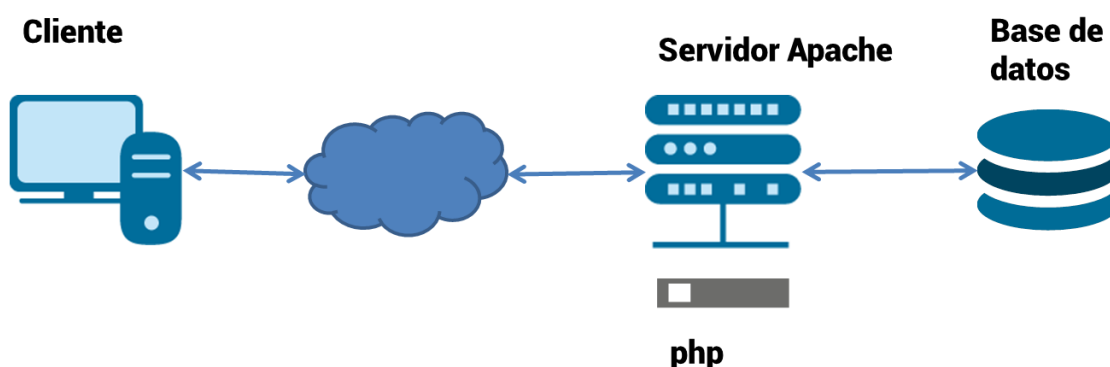
Le toca el turno de las bases de datos. No es el objetivo de este módulo estudiar en profundidad los conceptos sobre bases de datos, o sobre su tecnología, estos conceptos ya se consideran en 1º en el módulo específico sobre BBDD. Sin embargo tendremos una pequeña introducción sobre base de datos y porqué utilizamos estas junto a PHP.

## 1.1. Base de datos Mysql

Una de las bases de datos más utilizadas dentro del desarrollo Web es MySQL.

	<p><b>Mysql</b> <a href="https://dev.mysql.com/doc/refman/5.7/en/introduction.html">https://dev.mysql.com/doc/refman/5.7/en/introduction.html</a></p> <p>Mysql es un servidor de gestión de bases de datos relacional. Históricamente, Mysql y PostgreSQL se han repartido el monopolio del uso como gestor de bases de datos dentro de los desarrollos de aplicaciones web debido a su licencia opensource.</p> <p>Para más información sobre la última versión y los diferentes tipos de licencias, es conveniente visitar el enlace al fabricante facilitado.</p>
---	--

¿Cómo se incorpora un servidor de base de datos a nuestro modelo de arquitectura cliente-servidor? Veámoslo a través de una imagen:



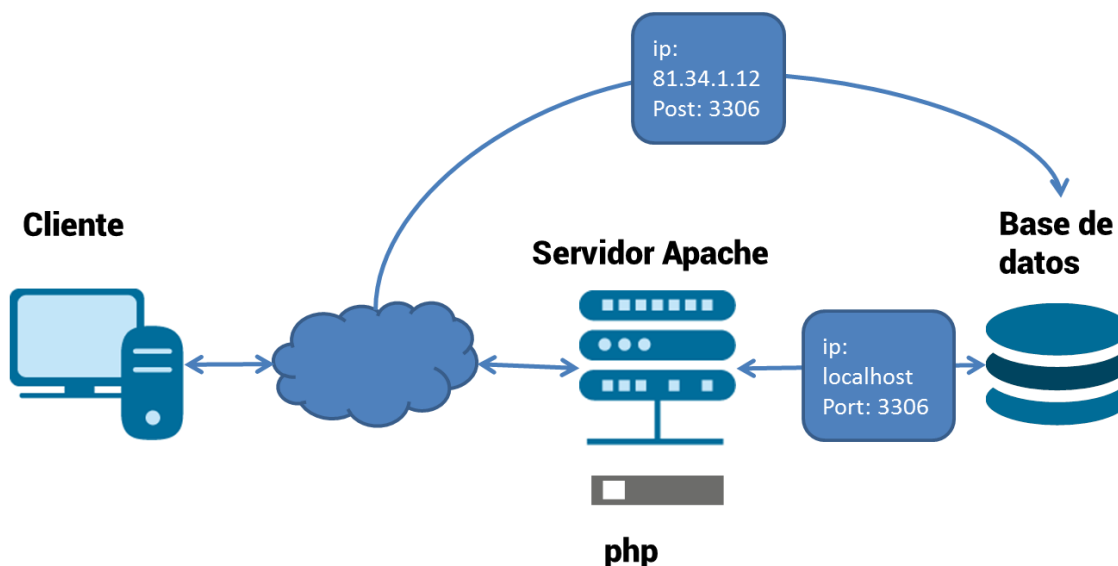
**Imagen 1: Arquitectura cliente-servidor**

Fuente de la imagen: propia

Podemos ver que aparece un nuevo elemento dentro de nuestro modelo, la base de datos. Al servidor de base de datos se accederá a través de nuestro ya conocido servidor Apache que será a su vez quien va a interpretar el código php.

## 1.2. Servidor Mysql

Por lo tanto a la vista del anterior diagrama vemos que como cualquier otro servidor, la base de datos sobre MySQL se gestionará y será accesible mediante unos parámetros de conexión que deberemos conocer:



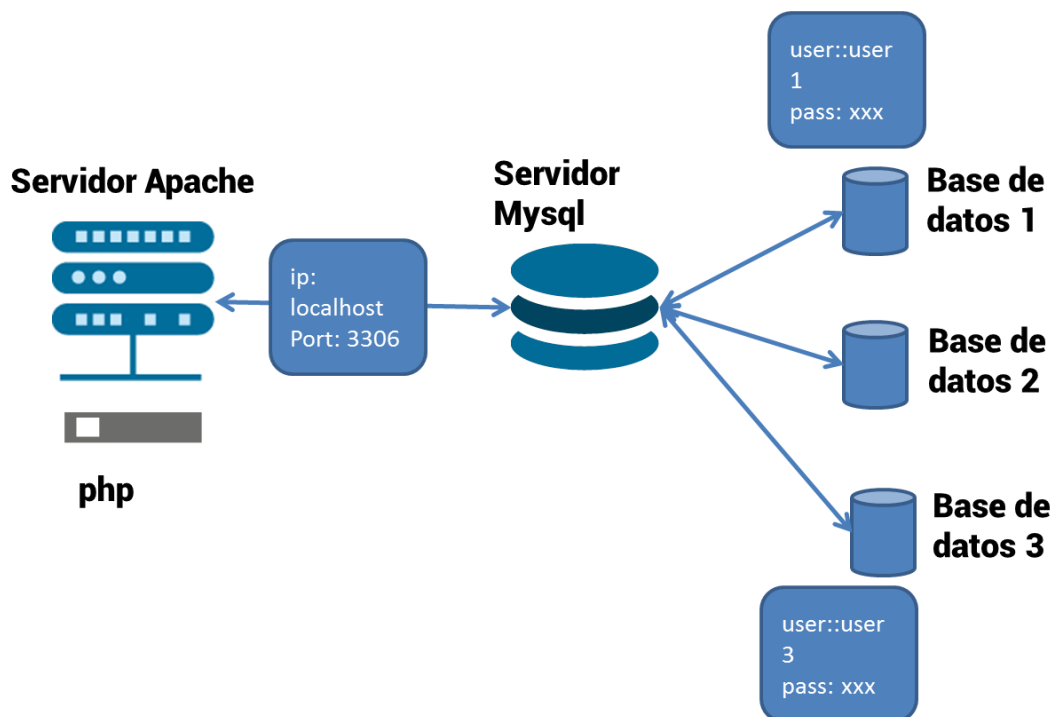
**Imagen 2: Parámetros de conexión**

Fuente de la imagen: propia

Como se puede observar, el servidor mysql va a tener una serie de parámetros básicos de conexión al servidor:

- **ip** o dirección de internet, dependerá desde dónde estemos "atacando" a nuestra base de datos, esta dirección será local o pública. Como podemos observar en la imagen, el proceso habitual es comunicarnos con nuestra base de datos a través del servidor Apache, y por lo tanto la dirección será "localhost".
- **Puerto**, el puerto por defecto de una base de datos mysql es el 3306, aunque es configurable.

Una vez que tenemos claros estos parámetros básicos para la conexión contra el **servidor** de base de datos debemos considerar que la interacción de nuestra aplicación se va a realizar contra una base de datos. Por ese motivo hemos querido destacar la palabra servidor:



**Imagen 3: Gestión de bases de datos**

Fuente de la imagen: propia

Como observamos en esta imagen, al aparecer las bases de datos que gestiona nuestro servidor MySQL, ya tendríamos todos los parámetros para una conexión contra una base de datos:

- **ip**
- **puerto**
- **base de datos**
- **user**
- **pass**



**COMPRUEBA LO QUE SABES:**

Una vez estudiado qué es un servidor MySQL, ¿serías capaz utilizar el servidor PostgreSQL (<https://www.postgresql.org/>)? Coméntalo en el foro.

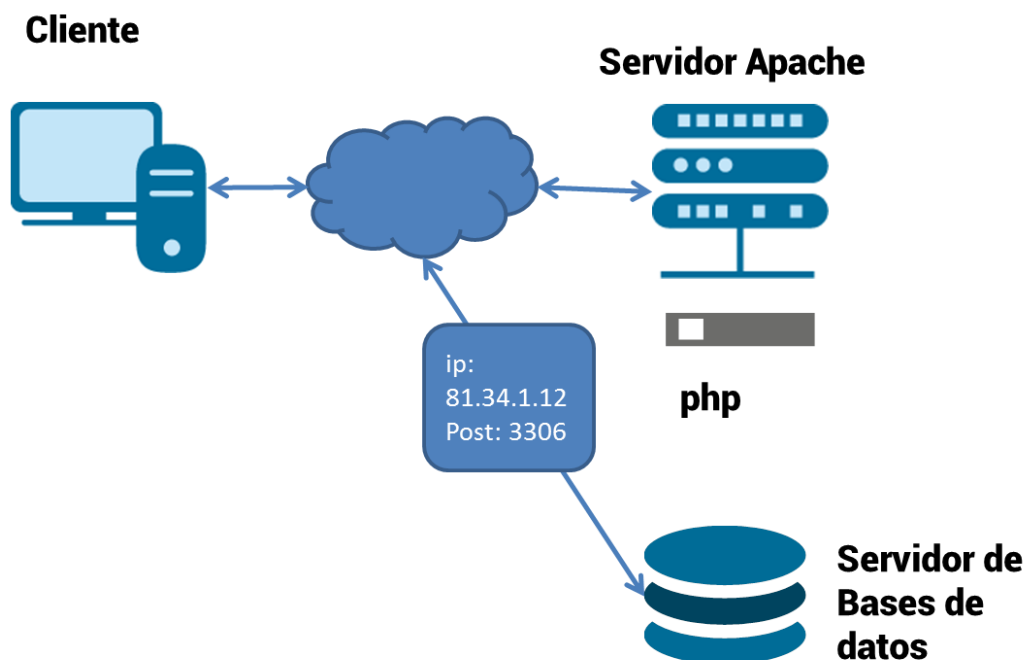
### 1.3. Consideraciones sobre seguridad

Tampoco queremos entrar en el detalle sobre configuración y seguridad sobre bases de datos y en concreto sobre MySQL que ya se estudió en módulos anteriores. Sin embargo sí que queremos incidir en algunos aspectos:

- El servidor de base de datos no debería ser accesible de forma directa desde el exterior o de forma pública.
- El usuario root, no debería poderse utilizar de forma pública
- Todos los usuarios deben tener contraseña
- Crear usuarios para cada una de las bases de datos o aplicaciones que se vayan a utilizar con diferentes niveles de seguridad.

### 1.4. Bases de datos en la nube

Actualmente con sistemas como AWS, el concepto de hosting y servidor ha evolucionado enormemente. Simplemente mostrar otra alternativa que es ahora mismo muy factible, escalable y mantenible:



**Imagen 3: Gestión de bases de datos**



Fuente de la imagen: propia

Como vemos el sistema en la nube es transparente para el desarrollador, ya que va a necesitar de los mismos parámetros que con una base de datos en local. Las mejoras que introduce este sistema son:

- Si nuestro sistema está diseñado suficientemente independiente de la tecnología de base de datos, podemos escalar o incluso cambiar las bases de datos de una forma inmediata
- Los sistemas de backup o de seguridad se mejoran, ya que los servicios en la nube de sistemas de bases de datos están adaptados y configurados al mantenimiento de bases de datos

## 2. BASES DE DATOS Y PHP

El uso de bases de datos con PHP es muy sencillo, y más aún si ya se ha realizado programación orientada a objetos y con estructuras de datos permanentes y bases de datos relacionales como MySQL.

Los pasos serán prácticamente los mismos en cualquier lenguaje de programación orientado a objetos:

- Diseñar e implementar la estructura dentro de nuestra base de datos.
- Crear la conexión.
- Implementar las funciones necesarias para poder realizar Lectura, Escritura, Actualización y Borrado (CRUD), con una base de datos.

**Libro recomendado, *Hacking with PHP*** (Hudson, 2015)

En el capítulo 9 encontramos la referencia a ejemplos y documentación sobre el manejo de bases de datos con PHP

<http://www.hackingwithphp.com/9/0/0/databases>



**Libro recomendado, PHP 7 Programming CookBook**

(Bierer, 2016)

En el capítulo 5 encontraremos una fantástica referencia la interacción entre PHP 7 y las Bases de datos



## 2.1. Paso 1. Creando una base de datos y una tabla con PHPMYADMIN

Como hemos dicho anteriormente, no es el objetivo centrarnos en el uso de phpmyadmin, una herramienta fantástica para poder interactuar con nuestra base de datos en mysql. Pero sí recordaremos o realizaremos acciones básicas con esta herramienta para poder crear nuestra base de datos y nuestra primera tabla.

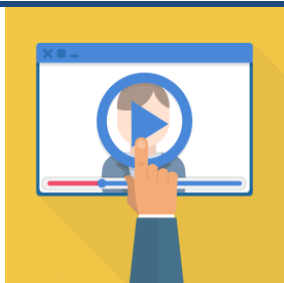


**phpmyadmin**

<https://www.phpmyadmin.net/>

phpmyadmin es una herramienta libre programada con PHP y que nos sirve para gestionar bases de datos en mysql por ejemplo

Veamos la explicación completa a través de un vídeo:



En el vídeo que encontrarás una introducción a los formularios



<https://youtu.be/lisvmzREY4E>



Tras iniciarnos en la creación las bases de datos con mysql y phpmyadmin:

- 1) Abriremos un navegador y escribiremos la dirección a phpmyadmin
- 2) Crearemos una nueva base de datos con el nombre Alumnos
- 3) Crearemos una nueva tabla denominada alumnos con los siguientes campos:
  - a. Nombre, tipo varchar 64
  - b. Apellidos, tipo varchar 128
  - c. Edad, int
- 4) Introduciremos al menos 4 nuevos alumnos

## 2.2. Paso 2. Crear conexión

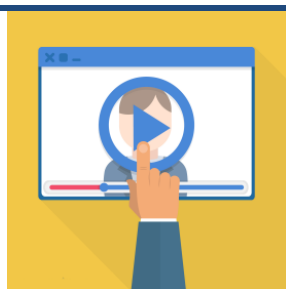
El primer paso en nuestro proceso es el de la conexión con la base de datos, y para ello necesitamos:

- Utilizar las librería mysqli (mysql mejorada)
- Enviar la información básica para poder conectarnos:
  - host, o máquina donde queremos conectarnos
  - db, o base de datos
  - user, usuario con permisos
  - password, la contraseña
- Veamos la explicación completa a través de un vídeo:



### RECUERDA... UD5, OBJETOS EN PHP

La palabra reservada para la creación de un objeto es *new*, sea a partir de una clase creada por nosotros o proporcionada por PHP.



En el vídeo que encontrarás cómo generar la conexión a una base de datos

 [https://youtu.be/aU\\_weN3azZ0](https://youtu.be/aU_weN3azZ0)

Y el código utilizado es:



En el siguiente enlace tienes el código utilizado:



<https://github.com/pacogomezarnal/codigoEjemploCursoPHP/tree/master/db>

Como podemos observar tanto en el vídeo como en el código utilizado, la base de la conexión a la base de datos consiste en utilizar las librerías de mysql:

#### Conexión a base de datos

```
<?php
//PLANTILLA

$mysqli = new mysqli("localhost", "usuario", "contraseña",
"basedatos");
if ($mysqli->connect_errno) {
    echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ")
" . $mysqli->connect_error;
}
echo $mysqli->host_info . "\n";

//BASE DE DATOS LOCAL

$mysqli = new mysqli("127.0.0.1", "usuario", "contraseña",
"basedatos", 3306);
if ($mysqli->connect_errno) {
    echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ")
" . $mysqli->connect_error;
}

echo $mysqli->host_info . "\n";
?>
```

Como podemos observar en el anterior código:

- Utilizamos librerías dentro de php que nos permiten la conexión a la base de datos
- El constructor de mysqli permite introducir todos los parámetros necesarios para la conexión a la base de datos, `mysqli("localhost", "usuario", "contraseña", "basedatos")`
- Mediante el método `connect_errno` podemos comprobar si la conexión se ha realizado con éxito.



#### **Mysql mejorada**

<http://php.net/manual/es/book.mysqli.php>

Podemos ver todas los métodos y clases dentro de las librerías mysql mejorada



Utilizando la base de datos creada realizaremos los siguientes pasos para la conexión:

- 1) Generamos la estructura de ficheros dentro con una nueva carpeta y un php que denominaremos alumnos.html
- 2) Realizaremos una nueva conexión creando un nuevo objeto mysqli con los parámetros necesarios para su conexión (IP,nombre de la base de datos, usuario y pass)
- 3) Comprobaremos si se ha conectado con éxito contra la bbdd mediante `connect_errno`

## 2.3. Mysqli frente a mysql

Como podemos observar en el anterior código y ejemplo, directamente utilizamos mysqli frente a mysql (obsoleto pero mantenido por compatibilidad hacia atrás).

Mysql es una librería no orientada a objetos y mantenida por php para desarrollos más antiguos pero que no va a ser ni mejorada y mantenida.

Por lo tanto, para nuevos desarrollos siempre utilizaremos mysqli.

## 2.4. Paso 3. Lectura de datos

Para la devolución de datos almacenados en una determinada tabla, deberemos utilizar la función query, la cual a su vez devolverá un resultado que es a su vez otro objeto. Un ejemplo lo tenemos en el siguiente código:

### Lectura de datos

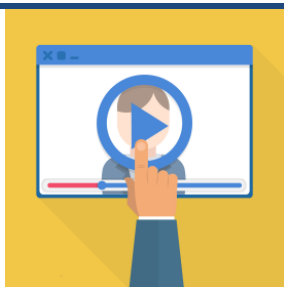
```
<?php
//Hacemos una consulta
$resultado = $mysqli->query("SELECT * FROM jugadores");
//Cuántas filas nos devuelve
echo "el numero de jugadores es: ".$resultado->num_rows."<br>";
for($i=0;$i<$resultado->num_rows;$i++){
    $fila=$resultado->fetch_assoc();
    echo "El jugador ".$fila['id']." se
llama: ".$fila['nombre']."<br>";
}
?>
```

Una vez lanzada la consulta y almacenada en una variable, según la documentación, vemos que el resultado es a su vez otro objeto, mysqli\_result, el cual contiene una propiedad super importante, num\_rows que me permitirá conocer el número de filas de la tabla, y también el método fetch\_assoc que me permitirá devolver una a una todas las filas que tiene la misma.

**Objeto `mysqli_result`**

<http://php.net/manual/es/class.mysqli-result.php>

Representa el conjunto de resultados obtenidos a partir de una consulta en la base de datos.



En el vídeo que encontrarás cómo leer datos desde la base de datos

 <https://youtu.be/L1-95frfJuY>



En el siguiente enlace tienes el código utilizado:



<https://github.com/pacogomezarnal/codigoEjemploCursoPHP/tree/master/db>

## RESUMEN FINAL

En esta unidad nos hemos adentrado en la interacción entre BBDD y lenguaje de servidor, en nuestro caso PHP. Las aplicaciones web y en concreto el código de servidor tiene su principal objetivo en interactuar contra una BDD de datos.

Junto con los conocimientos adquiridos en la unidad anterior, interacción con formularios, queda prácticamente cerrada las posibilidades que nos permite PHP para realizar un proceso completo de interacción con el usuario:

- El usuario introduce información a través de un formulario HTML
- El lenguaje de cliente, javascript, junto con el de servidor, PHP verifican que toda la información escrita es correcta
- PHP recibe la información verificada e interactúa con la BDD
- PHP realiza una respuesta al cliente