

Desarrollo Web en Entorno Servidor

Programación en PHP

José A. Lara

Bases de datos, MySQL y PHP

- 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.

Bases de datos, MySQL y PHP

Introducción al concepto de bases de datos

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.

Bases de datos, MySQL y PHP

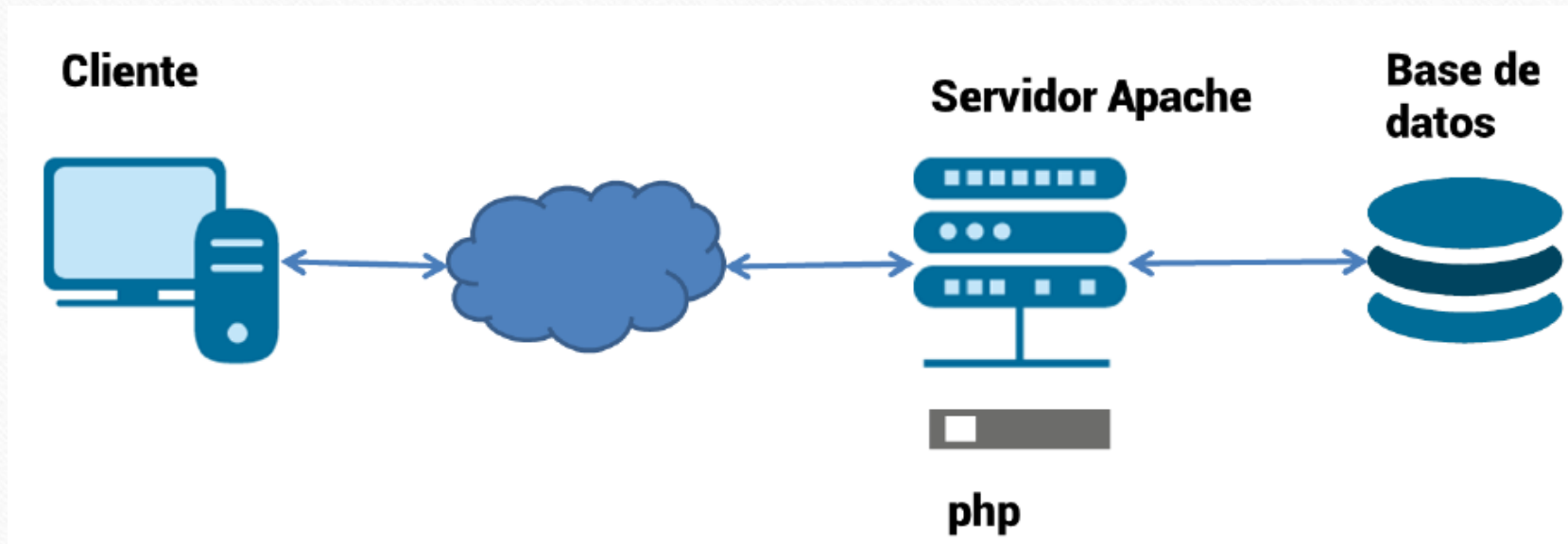
MySQL (MariaDB)

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.

Bases de datos, MySQL y PHP

MySQL (MariaDB)

- ¿Cómo se incorpora un servidor de base de datos a nuestro modelo de arquitectura cliente-servidor?



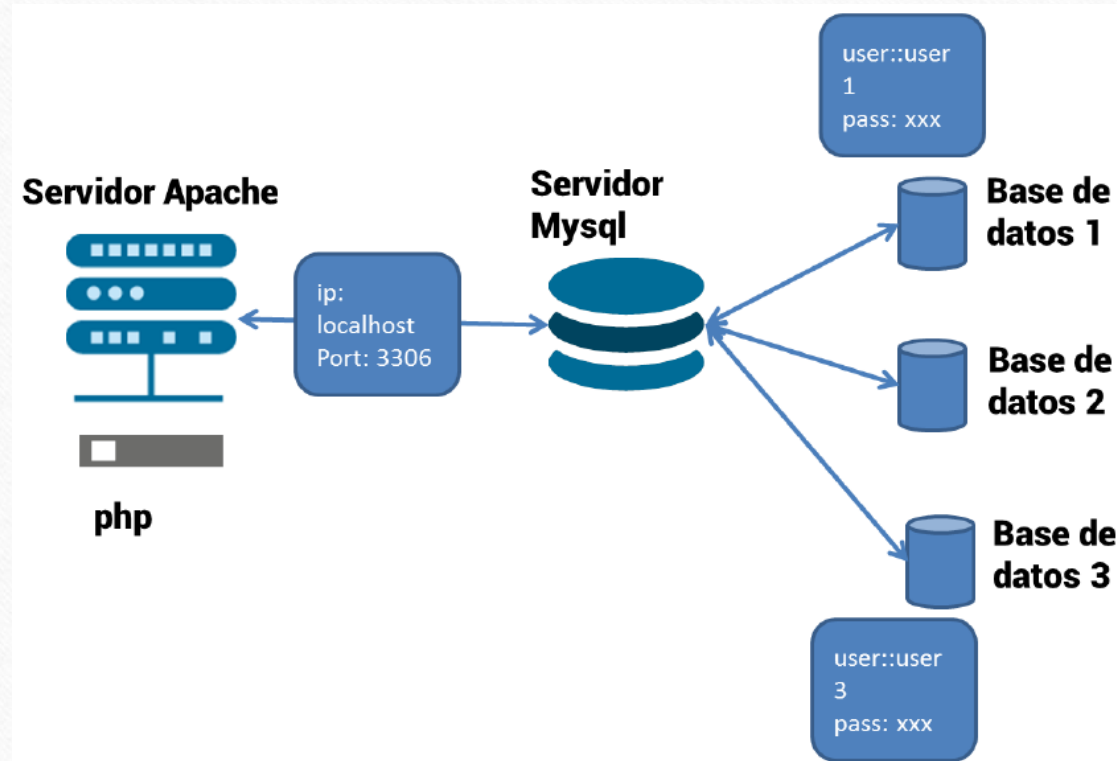
Bases de datos, MySQL y PHP

MySQL (MariaDB)

- La base de datos sobre MySQL se gestionará y será accesible mediante unos parámetros de conexión que deberemos conocer:
 - **ip o dirección de internet**, dependerá desde dónde estemos “atacando” a nuestra base de datos, esta dirección será local o pública. 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.

Bases de datos, MySQL y PHP

MySQL (MariaDB)



Bases de datos, MySQL y PHP

MySQL (MariaDB)

- Parámetros para una conexión contra una base de datos:
 - ip
 - puerto
 - base de datos
 - user
 - password

Bases de datos, MySQL y PHP

MySQL (MariaDB)

- Consideraciones sobre seguridad
 - 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.

Bases de datos, MySQL y PHP

Bases de datos en la nube

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 lo 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

Bases de datos, MySQL 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.

Bases de datos, MySQL y PHP

Crear una base de datos y una tabla en PHPMyAdmin

Tras iniciarnos en la creación las bases de datos con MySQL y PHPMyAdmin:

1. Abriremos un navegador y escribiremos la dirección de 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, tipo int
4. Introduciremos al menos 4 nuevos alumnos

Bases de datos, MySQL y PHP

Crear una conexión

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

- Utilizar la 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

Bases de datos, MySQL y PHP

Crear una conexión

```
<?php
//Crear el objeto de conexion
$mysqli = new mysqli("localhost", "usuario", "contraseña", "basedatos", 3306);
if ($mysqli->connect_errno) {
    //Ha habido error
    echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
}else{
    echo "<h3>Conexion realizada</h3><br><hr>";
}
echo $mysqli->host_info . "\n";
?>
```

← → ↻ ⓘ localhost/php/pruebadb.php

Conexion realizada

localhost via TCP/IP

← → ↻ ⓘ localhost/php/pruebadb.php

Warning: mysqli::__construct(): (HY000/1049): Unknown database 'eba' in C:\xampp\htdocs\php\pruebadb.php on line 3
Fallo al conectar a MySQL: (1049) Unknown database 'eba'
Warning: main(): Couldn't fetch mysqli in C:\xampp\htdocs\php\pruebadb.php on line 10

Bases de datos, MySQL y PHP

Ejercicio

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 a la base de datos mediante `connect_errno`

Bases de datos, MySQL y PHP

¿mysql o mysqli?

- 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.

Bases de datos, MySQL y PHP

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.

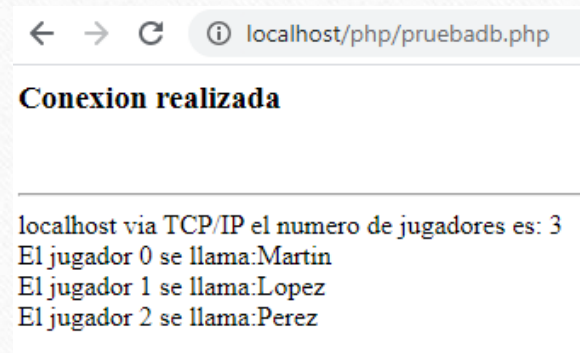
```
<?php
//Crear el objeto de conexion
$mysqli = new mysqli("localhost", "root", "", "prueba", 3306);
if ($mysqli->connect_errno) {
    //Ha habido error
    echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
}else{
    echo "<h3>Conexion realizada</h3><br><hr>";
}
echo $mysqli->host_info . "\n";

//Hacemos una consulta
$resultado = $mysqli->query("SELECT * FROM jugadores");
//Cuantas filas nos devuelve
echo "el numero de jugadores es: ".$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>";
}
?>
```

Bases de datos, MySQL y PHP

Lectura de datos

- Una vez lanzada la consulta y almacenada en una variable, 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.



A screenshot of a web browser window. The address bar shows 'localhost/php/pruebadb.php'. Below the address bar, the text 'Conexion realizada' is displayed. Further down, the output of a PHP script is shown: 'localhost via TCP/IP el numero de jugadores es: 3', 'El jugador 0 se llama:Martin', 'El jugador 1 se llama:Lopez', and 'El jugador 2 se llama:Perez'.

```
localhost/php/pruebadb.php

Conexion realizada

localhost via TCP/IP el numero de jugadores es: 3
El jugador 0 se llama:Martin
El jugador 1 se llama:Lopez
El jugador 2 se llama:Perez
```