

TEMA 4.2

Acceso a datos con PHP

PHP y MySQL

- El lenguaje PHP está en constante evolución.
- Sus utilidades para acceder a bases de datos han ido evolucionando y mejorando con las sucesivas versiones del lenguaje.
- En concreto, para acceso a MySQL, php ha ofrecido tres opciones:
 1. **Funciones no OO** (no Orientadas a Objetos), con el prefijo **mysql_**, **deprecadas**. Ya no se utilizan, son previas a la versión 5 de PHP.
 2. La **opción OO** de las funciones anteriores, llamada **mysqli**. Con esta opción ya se tiene un objeto de tipo "conexión a base de datos". Es posible que nos encontremos código que la utilice. Nosotros no la usaremos en este módulo, pues actualmente se utiliza PDO.
 3. La **opción OO** llamada **PDO** (Portable Data Objects). Todo el acceso a bases de datos lo haremos mediante esta opción.

Ejemplo de mysqli orientado a objetos

- Si alguna vez os encontráis código con mysqli, el manual es el siguiente:

<https://www.php.net/manual/es/book.mysqli.php>

- Un ejemplo de conexión a MySQL con mysqli en el localhost es:

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

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

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

Ejemplo de conexión con mysqli: Manual de PHP en www.php.net

Ejemplo de PDO

- En este tema aprenderemos a utilizar **PDO** para conectarnos a MySQL
- El manual de PDO es el siguiente:

<https://www.php.net/manual/es/book.pdo.php>

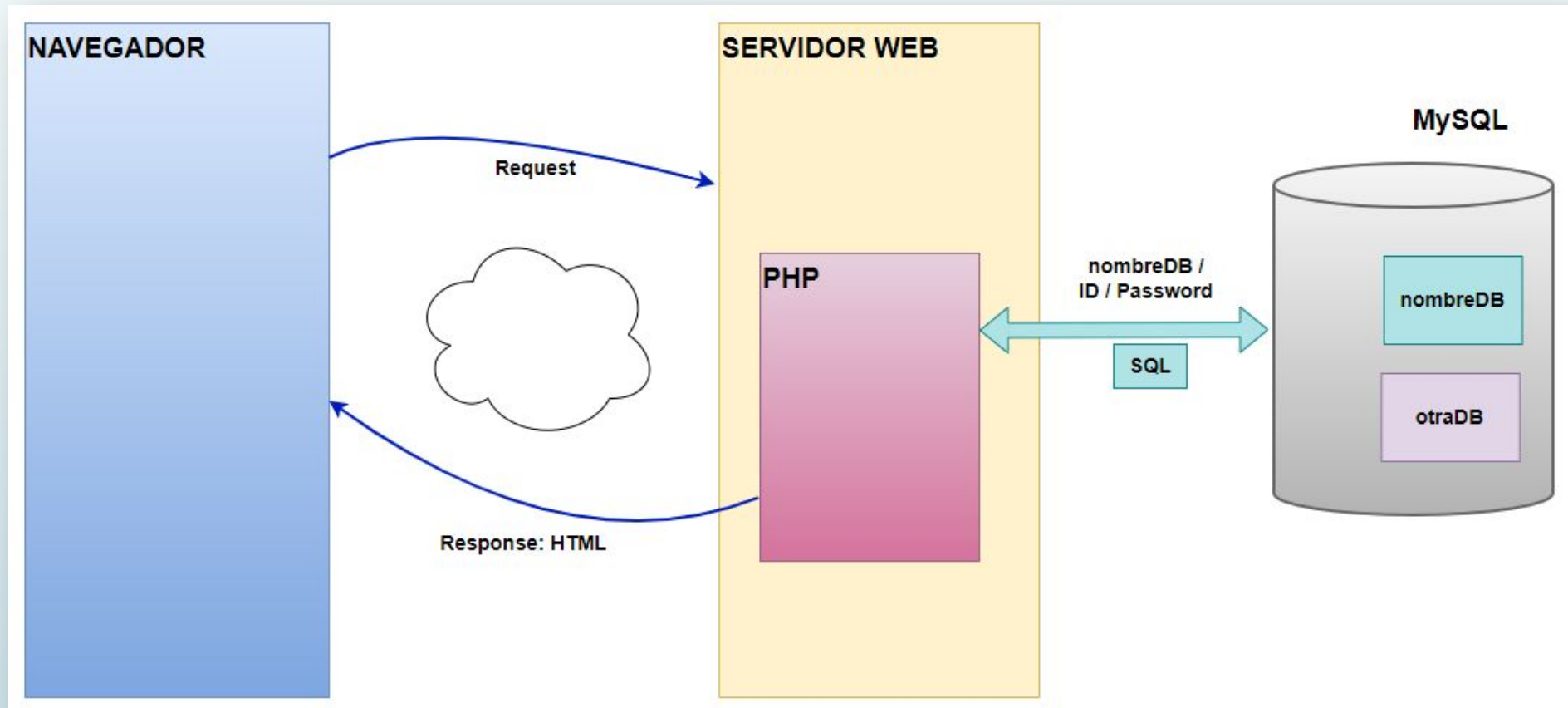
- Para conectarnos a MySQL mediante PDO, creamos un objeto de la clase PDO.
- Este objeto es una conexión, y su constructor pide como parámetros: El dominio del servidor de base de datos, el nombre de la base de datos, el usuario y la contraseña.
- Un ejemplo de conexión a MySQL con PDO en el localhost sería:

```
<?php
$mbd = new PDO('mysql:host=localhost;dbname=prueba', $usuario, $contraseña);
?>
```

Ejemplo de conexión con PDO: Manual de PHP en www.php.net

PHP y MySQL

En este tema aprenderemos a utilizar PDO para conectarnos a MySQL



Ejecutando queries en PHP

- En nuestro programa php, lo primero que hay que hacer para poder manejar una BD en MySQL es conectarse.
- Para la conexión, es necesario que tengamos un usuario en la BD y los permisos necesarios.
- Aunque MySQL ofrece menús interactivos para realizar la gestión de permisos, es interesante utilizar sentencias SQL:
 1. Necesitamos que haya creada una BD y saber su nombre:
CREATE DATABASE nombreDB
 2. Necesitarmemos que el usuario exista y tenga todos los permisos:
GRANT ALL ON nombreDB TO 'marta'@'localhost' IDENTIFIED BY 'marta'
GRANT ALL ON nombreDB TO 'marta'@'127.0.0.1' IDENTIFIED BY 'marta'
- Para más información sobre cómo realizar una configuración inicial de MySQL, ver las transparencias:

TEMA 4.1.- Configuración inicial de MySQL

Conexión a la BD: new PDO

- Para realizar la conexión a la BD desde PHP, utilizaremos un objeto PDO, que será nuestra conexión.

- El comando para conectarse es el siguiente:

```
$pdo = new PDO('mysql:host=localhost;port=3306;dbname=martabd', 'marta', 'marta');
```

Nota: Si estás utilizando docker, el host será “**host=db**” en vez de “host:localhost”.

- Siendo:

- **Host:** El dominio del servidor.

- **Port:** El puerto del servidor de BD:

- En el caso de Linux y Windows es el **3306**

- En el caso de Mac es el **8889**

- **Dname:** El nombre de la base de datos

- **User y password:** Los dos últimos parámetros son **usuario** y **contraseña** de la BD. En el caso de docker serán **'root', 'root'**

- En la variable **\$pdo**, tenemos una referencia a la la BD a la que nos hemos conectado. Para que la conexión sea posible, tendrá que existir en mysql

Mi primer PHP con PDO: new PDO, query, fetch

En el siguiente programa PHP, hacemos los siguientes pasos:

1. Conexión a BD: Instanciamos la conexión en un objeto **\$pdo**, será la referencia a la misma.
2. Ejecución de una query a esa BD: Con la función **query** de la clase PDO.
3. **\$resultado**: Resultado de la query en un objeto de la clase PDOStatement. Pueden ser varias filas.
4. **fetch**: Obtención de la siguiente fila de \$resultado. Empezará por la primera.
5. **\$fila**: Es un array asociativo, siendo la clave el nombre del campo y el valor el contenido de dicho campo.

```
primeraConexion.php
1  <?php
2  echo "<pre>\n";
3  // Me conecto a la BD martabd
4  $pdo=new PDO("mysql:host=localhost;port=3306;dbname=martabd",
5  |    'root','root');
6  // Realizo una query a la BD y obtengo el resultado en $resultado
7  $resultado = $pdo->query("SELECT * FROM usuarios");
8  // En un bucle, voy recuperando cada fila con fetch
9  while ($fila=$resultado->fetch(PDO::FETCH_ASSOC)) {
10 |     print_r($fila);
11 | }
12 echo "</pre><br/>";
```

```
localhost/marta/primeraConexion.php

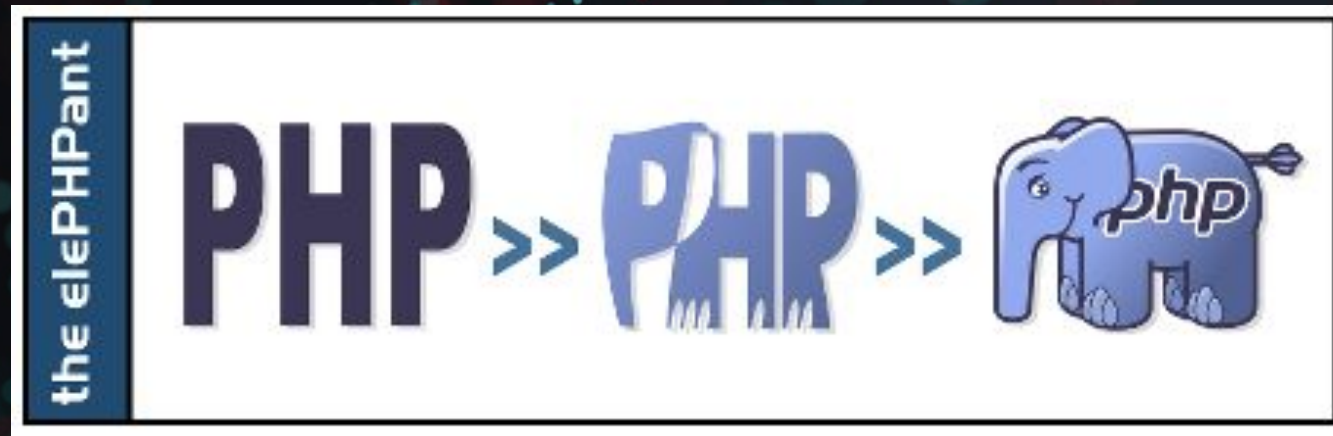
Array
(
    [usuario_id] => 1
    [nombre] => Marta Olmedilla
    [email] => marta0@europa.es
    [password] => 1a52e17fa899cf40fb04cfc42e6352f1
)
Array
(
    [usuario_id] => 2
    [nombre] => alumno
    [email] => alumno@europa.es
    [password] => alumno
)
```




Ejercicio 1:

Acceso a mi tabla con PHP

1. Crea una tabla con 4 campos en tu BD. Inserta 3 registros.
2. Crea un programa php que se conecte a tu BD y haga una select para recuperar todos los registros de la tabla que has creado.
3. Haz que tu programa php haga un print_r de cada uno de los registros recuperados al hacer la query
4. Comprueba que funciona
5. Modifica el programa anterior para que muestre los resultados en forma de tabla HTML.



<https://www.php.net/docs.php>

<https://www.php.net/manual/es/book.pdo.php>