

UD-04a: Acceso a BD con PHP

Desarrollo Web en Entorno Servidor

Curso 2020/2021

Acceso a BD desde PHP

- Una de los usos más frecuentes de PHP es acceder y gestionar la información almacenada en una BD.
- Nos permite mostrar información extraída de la BD en una web, o enviar sentencias al gestor de la BD para modifique sus registros.
- PHP soporta más de 15 sistemas gestores de bases de datos: *SQLite, Oracle, SQL Server, PostgreSQL, IBM DB2, MySQL/MariaDB, etc.*
- Mediante una extensión de php, llamada **PDO**, podemos acceder de forma común a todos estos sistemas gestores. Permite seguir utilizando una misma sintaxis aunque cambiemos el SGBD.
- Como alternativa a PDO, las **extensiones nativas** específicas para cada SGDB, suelen ofrecer más potencia y, en algunos casos, también mayor velocidad.
- De los distintos SGBD existentes, utilizaremos MySQL/MariaDB, por ser de código abierto y ser el gestor de BD más usado con el lenguaje PHP.

Acceso a BD desde PHP

Existen 2 APIs principales en PHP cuando hablamos de conexión con **MySQL**:

- Objetos de Datos de PHP (**PDO**)
- Extensión **mysqli** de PHP (mysql mejorada)

Utilizar **mysqli** tiene numerosas mejoras sobre la extensión **mysql** como:

- *Interfaz orientada a objetos, Soporte para Declaraciones Preparadas, Soporte para Múltiples Declaraciones, Soporte para Transacciones, Mejoradas las opciones de depuración y Soporte para servidor empujado*

Con cualquiera de ambas extensiones, podrás realizar acciones sobre las BD como:

- ✓ Establecer conexiones.
- ✓ Ejecutar sentencias SQL.
- ✓ Obtener los registros afectados o devueltos por una sentencia SQL.
- ✓ Emplear transacciones.
- ✓ Gestionar los errores que se produzcan durante la conexión o en el establecimiento de la misma.

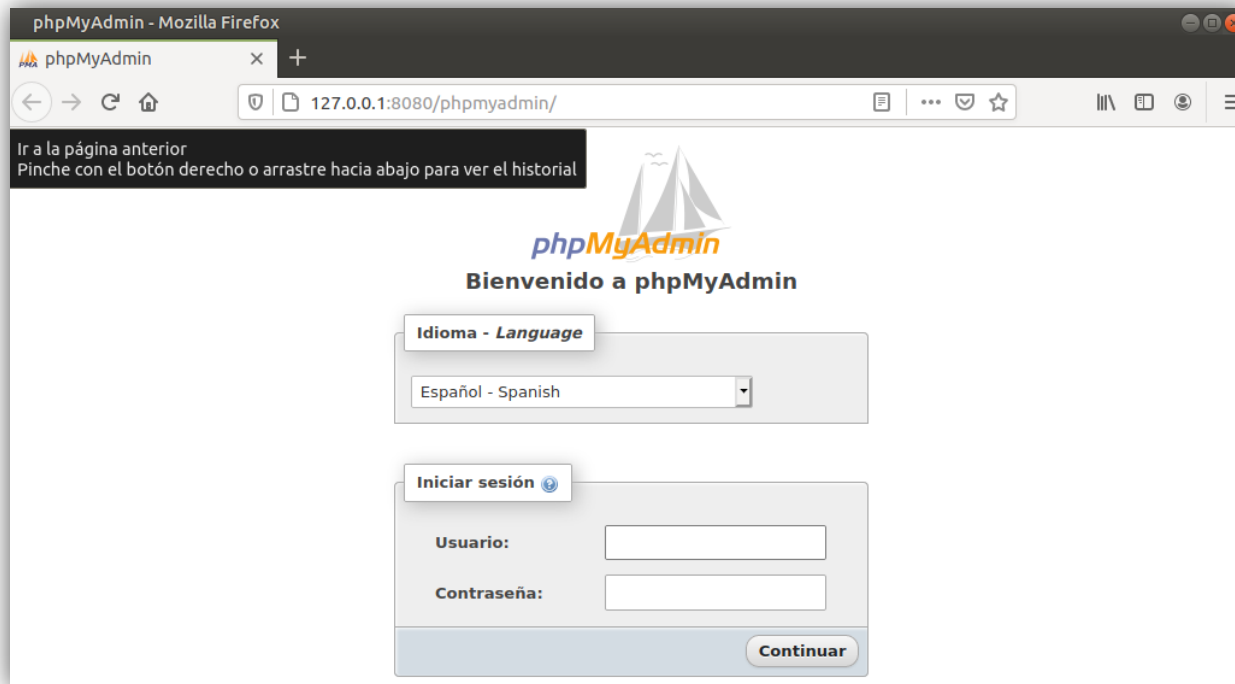
MySQL/MariaDB

Motores de almacenamiento

- MySQL/MariaDB incorpora múltiples motores de almacenamiento, cada uno con características propias: unos son más veloces, otros aportan mayor seguridad o mejores capacidades de búsqueda.
- Cuando crees una base de datos, podemos elegir el motor en función de las características propias de la aplicación. Por defecto, el motor que se utiliza se llama InnoDB es un poco más lento, pero soporta integridad referencial y tablas transaccionales. Sin embargo, también podemos usar el motor MyISAM, que es muy rápido, pero a cambio no contempla estas dos características.
- MySQL se usa en múltiples aplicaciones web ligado al lenguaje PHP y al servidor web *Apache*.
- Para ampliar información, se puede consultar el manual en línea de MySQL: <https://dev.mysql.com/doc/refman/5.0/es/>

MySQL/MariaDB

- Cuando instalamos nuestro LAMP también se instaló **phpMyAdmin**, que será la interfaz de gestión de la BD.
- Accederemos a esta herramienta en nuestro entorno mediante un navegador web, haciendo referencia en la URL a la IP del servidor seguido de */phpmyadmin*



Utilización de MySQL/MariaDB en PHP

Para operar con MySQL desde PHP hay que realizar una serie de pasos:

- **Obtener una conexión** con el SGBD pasando una cadena de conexión con los datos como el nombre de usuario, contraseña, base de datos, opciones, etc

```
// usando constructores y métodos de la poo
$conexion = new mysqli('localhost','usuario','contraseña','base_datos');

// utilizando llamadas a funciones
$conexion = mysqli_connect('localhost','usuario','contraseña','base_datos');

// en ambos casos, la variable $conexion es de tipo objeto.
```

- Con la conexión obtenida en el paso anterior, podremos **lanzar sentencias SQL**, tanto **SELECT** para obtener datos, como **INSERT, UPDATE, DELETE**, etc.
- Usaremos sentencias de tipo **SELECT** para generar información en la web y el resto de operaciones SQL para la información recogida a partir de formularios.

MySQLi

Establecimiento de conexiones

Por tanto, establecer una conexión con el servidor significa **crear una instancia de la clase `mysqli`**. El constructor de la clase puede recibir seis parámetros, todos opcionales, aunque lo más habitual es usar los **cuatro** primeros:

- ❖ El nombre o dirección IP del servidor MySQL al que te quieres conectar.
- ❖ Un nombre de usuario con permisos para establecer la conexión.
- ❖ La contraseña del usuario.
- ❖ El nombre de la base de datos a la que conectarse.
- ❖ El número del puerto en que se ejecuta el servidor MySQL.
- ❖ El socket o la tubería con nombre (named pipe) a usar.

```
$conexion = new mysqli('localhost', 'usuario', 'contraseña', 'bd');
```

MySQLi

También podemos crear la conexión con el método **connect**.

```
$conexion = new mysqli();  
$conexion->connect('localhost', 'usuario', 'contraseña', 'base_datos');
```

Para verificar que la conexión se ha establecido correctamente y comprobar si se produce error, se pueden usar las siguientes propiedades de la clase mysqli:

- **connect_errno** (o la función mysqli_connect_errno) devuelve el número de error o null si no se produce ningún error.
- **connect_error** (o la función mysqli_connect_error) devuelve el mensaje de error o null si no se produce ningún error.

```
if ($conexion->connect_errno) {  
    echo "Error conectando con MySQL: " . $conexion->connect_error;  
    exit();    // sale del script  
}
```


MySQLi

Operador de control de errores (@)

- En PHP, ocasionalmente, podemos anteponer a cualquier expresión el operador de control de errores @ para que ignore cualquier posible error que pueda producirse al ejecutarla.
- Simplemente lo que se consigue es que PHP no muestre el típico mensaje de error con el motivo del error, el archivo y la línea de código específica donde se ha provocado.

<http://es.php.net/manual/es/language.operators.errorcontrol.php>

Base de datos de ejemplo: *bdservidor*

The screenshot shows the phpMyAdmin web interface in a browser. The address bar indicates the URL is `127.0.0.1:8080/phpmyadmin/db_s`. The interface is in Spanish. On the left, a sidebar shows the database structure with 'bdservidor' selected, containing a 'usuarios' table. The main area displays the 'Estructura de tabla' (Table Structure) for 'usuarios'. Below this, a 'Vista de relaciones' (Relationships View) is shown, displaying a table with 6 rows of user data. A dropdown menu is open over the table, showing options like 'Editar' (Edit), 'Copiar' (Copy), and 'Borrar' (Delete) for each row.

phpMyAdmin

Reciente Favoritas

Servidor: localhost:3306 » Base de datos: bdservidor » Tabla: usuarios

Examinar Estructura SQL Buscar Más

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
1	id	int(5)			No	Ninguna
2	nombre	varchar(50) utf8_general_ci			No	Ninguna

Selecc. todo Examinar Cambiar Índice Texto completo

Imprimir Mover columnas

Agregar 1 columna Continuar

+ Opciones

				id	nombre
<input type="checkbox"/>	Editar	Copiar	Borrar	1	Pepe
<input type="checkbox"/>	Editar	Copiar	Borrar	2	Rosa
<input type="checkbox"/>	Editar	Copiar	Borrar	3	Juan
<input type="checkbox"/>	Editar	Copiar	Borrar	4	Pedro
<input type="checkbox"/>	Editar	Copiar	Borrar	5	Nuria
<input type="checkbox"/>	Editar	Copiar	Borrar	6	Manuel

Ejemplo usando extensión MySQLi

```
<?php
$conexion = @new mysqli('localhost', 'root', 'root', 'bdservidor');

if ($conexion->connect_errno){
    // die es equivalente a exit
    die('Lo siento, hubo un problema con la base de datos');
}else{
    $id = $_GET['id'];
    $sentencia_sql = "SELECT * FROM usuarios WHERE ID = $id";
    $resultado = $conexion->query($sentencia_sql);

    // num_rows: Obtiene el número de filas de resultado
    if ($resultado->num_rows){

        // fetch_assoc: Obtiene una fila como un array asociativo
        while ($fila = $resultado->fetch_assoc()){
            echo $fila['id'].' - '. $fila['nombre'].'<br>';
        }
    } else {
        echo 'No hay datos';
    }
}

?>
```

MySQLi

Si una vez establecida la conexión queremos cambiar la BD, podemos usar el método **select_db** (o *mysqli_select_db*) para indicar el nombre de la nueva.

```
$conexion->select_db('otra_bd');
```

Una vez finalizadas las tareas con la BD, usamos el método **close** (o *mysqli_close*) para cerrar la conexión con la BD y liberar los recursos que utiliza.

```
$conexion->close();
```

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

Utilización de MySQL en PHP

Consultas de las que NO devuelven datos

- La forma más rápida de ejecutar una consulta utilizando mysqli es el método **query**, equivalente a la función **mysqli_query**.
- Si se ejecuta una consulta que no devuelve datos (como una sentencia SQL de tipo **UPDATE**, **INSERT** o **DELETE**), la llamada devuelve true si se ejecuta correctamente o false en caso contrario.
- El número de registros afectados se puede obtener con la propiedad **affected_rows** (o con la función **mysqli_affected_rows**).

Utilización de MySQL en PHP

Ejemplo:

<?php

```
$conexion = new mysqli('localhost', 'root', 'root', 'bdproductos');  
$error = $conexion->connect_errno;  
if ($error == null) {  
    $consulta = $conexion->query('DELETE FROM stock WHERE unidades=0');  
    if ($consulta) {  
        echo "<p>$conexion->affected_rows registros borrados.</p>";  
    }  
    $conexion->close();  
}
```

?>

Utilización de MySQL en PHP

Ejecución de consultas que SÍ devuelven datos

fetch_array

Los datos obtenidos del servidor se devuelven en forma de objeto de la clase `mysqli_result`. Para trabajar con estos datos, existen varias posibilidades:

- **fetch_array** (o función **mysqli_fetch_array**). Obtiene un registro completo del conjunto de resultados y lo almacena en un array. Por defecto el array contiene tanto claves numéricas como asociativas. Por ejemplo, para acceder al primer campo devuelto, podemos utilizar como el índice 0 o su nombre, indistintamente.

```
$consulta = $conexion->query('SELECT producto, unidades FROM stock WHERE id=5');  
$stock = $consulta->fetch_array(); // Obtenemos el primer registro  
$prod = $stock['producto']; // también $stock[0];  
$unid = $stock['unidades']; // también $stock[1];  
echo "<p>Producto $prod: $unid unidades.</p>";
```

(En este ejemplo sólo se muestra el primer registro obtenido. No hacemos bucle)

Podemos añadir opcionalmente un parámetro a la función, que puede tomar los siguientes valores:

- **MYSQLI_NUM**. Devuelve un array con claves numéricas.
- **MYSQLI_ASSOC**. Devuelve un array asociativo.
- **MYSQLI_BOTH**. Comportamiento por defecto. Devuelve un array con claves tanto numéricas como asociativas.

fetch_object

```
@$conexion = new mysqli('localhost',root,'root','bdpokemon');  
// comprobar aquí errores conexión BD  
$consulta = $conexion->query("SELECT id, nombre FROM empleado WHERE tipo='SS'");  
$resultado = $consulta->fetch_object();  
while ($resultado != null)  
{  
    echo $resultado->id . " - " . $resultado->nombre . "<br/>";  
    $resultado = $consulta->fetch_object();  
}
```

Como puede verse, `fetch_object` va devolviendo los registros en forma de objeto.
Para acceder a los campos de dichos registros, usamos la flecha (`->`)

fetch_assoc

```
@$conexion = new mysqli('localhost',root,'root','bdpokemon');  
  
//Comprobar errores conexión BD  
  
$consulta = $conexion->query("SELECT id, nombre FROM empleado WHERE tipo='SS'");  
$resultado = $consulta->fetch_assoc();  
while ($resultado != null)  
{  
    echo $resultado['id'] . " - " . $resultado['nombre'] . "<br/>";  
    $resultado = $consulta->fetch_assoc();  
}
```

Como puede verse, fetch_assoc va devolviendo los registros en forma de array asociativo. Para acceder a los campos de dichos registros, usamos corchetes y el nombre del campo

fetch_row

```
@$conexion = new mysqli('localhost',root,'root','bdpokemon');  
  
//Comprobar errores conexión BD  
  
$consulta = $conexion->query("SELECT id, nombre FROM empleado WHERE tipo='SS'");  
$resultado = $consulta->fetch_row();  
while ($resultado != null)  
{  
    echo $resultado[0] . " - " . $resultado[1] . "<br/>";  
    $resultado = $consulta->fetch_row();  
}
```

Como vemos, `fetch_row` va devolviendo los registros en forma de array numérico. Para acceder a los campos de los registros, usamos corchetes y el índice numérico

Por tanto, podríamos decir que `fetch_array` es una combinación de `fetch_assoc` + `fetch_row`



EJERCICIO

Crea una página llamada **ud04ejer01.php** y título “UD 4 Ejercicio 1”. Luego, en el servidor, crearemos con *phpMyAdmin* una base de datos llamada **bdlibros**. Para ello, abrimos phpMyAdmin y clicamos en **Nueva**.

Después, escribimos **bdlibros** y clicamos en **Crear**.

Crear base de datos

bdlibros utf8_general_ci

Crear



Ahora, vamos a crear dentro de esa BD una tabla. Para ello, clicamos el nombre de la BD a la izquierda. Después, damos el nombre **libros** a la tabla , la cual tendrá 4 columnas.

Crear tabla

Nombre: libros

Número de columnas: 4

Continuar

Después, definimos los nombres y tipos de las columnas:

id (INT y CLAVE PRIMARIA)

titulo (VARCHAR, 100)

autor (VARCHAR, 100)

paginas (INT)

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice
id	INT		Ninguno			<input type="checkbox"/>	PRIMARY
Seleccionar desde las columnas centrales							
titulo	VARCHAR	100	Ninguno			<input type="checkbox"/>	---
Seleccionar desde las columnas centrales							
autor	VARCHAR	100	Ninguno			<input type="checkbox"/>	---
Seleccionar desde las columnas centrales							
paginas	INT		Ninguno			<input type="checkbox"/>	---
Seleccionar desde las columnas centrales							

Luego, más abajo, pulsamos el botón de **Guardar**

Se nos quedará algo así:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	id	int		No	Ninguna			Cambiar Eliminar Primaria
<input type="checkbox"/>	2	titulo	varchar(100)	latin1_swedish_ci	No	Ninguna			Cambiar Eliminar Primaria
<input type="checkbox"/>	3	autor	varchar(100)	latin1_swedish_ci	No	Ninguna			Cambiar Eliminar Primaria
<input type="checkbox"/>	4	paginas	int(100)		No	Ninguna			Cambiar Eliminar Primaria

Ahora, desde el archivo *php* (no desde phpMyAdmin) realiza las siguientes acciones:

- Inserta estos libros a la BD y muéstralos todos en forma de tabla, sin especificar orden.

1	Jarry Choped y la Hierba Filosofal	JK Bowling	301
2	El Senyor de los Palillos	JRR TolQuien	550
3	Jarry Choped y la Sabana Secreta	JK Bowling	302
4	Los Polares de la Tierra	Ken Follonet	400
5	Jarry Choped y el peluquero de Azkaban	JK Bowling	303
6	Los Juegos de Enjambre	Suzanne Collonins	450
7	Jarry Choped y el lapiz de fuego	JK Bowling	304
8	El Bolido da Vinci	Dan Black	500

En este ejercicio, si hubiera un error en la conexión con la BD debe mostrarse (usando `connect_errno` y `connect_error`).

Prueba a cambiar algo en tu línea de conexión para ver si muestras el error.

- Muestra todos los libros ordenados por número de páginas descendente.
- Muestra sólo título y páginas de los libros de *JK Bowling*.
- Elimina los libros de JK Bowling (di cuántos se han eliminado con **affected_rows**) y muestra todos los libros.
- Cambia el título del libro cuya *id* es 8 para que sea *El Morbido da Vinci* y muestra todos los libros.

Por lo tanto, en tu página aparecerán 5 tablas. Antes de cada tabla, haremos un echo explicando la acción. Te debe quedar algo así:



Introducimos 8 libros en la tabla y mostramos...

ID	TITULO	AUTOR	PAGINAS
1	Jarry Choped y la Hierba Filosofal	JK Bowling	301
2	El Senyor de los Palillos	JRR TolQuien	550
3	Jarry Choped y la Sabana Secreta	JK Bowling	302
4	Los Polares de la Tierra	Ken Follonet	400
5	Jarry Choped y el peluquero de Azkaban	JK Bowling	303
6	Los Juegos de Enjambre	Suzanne Collonins	450
7	Jarry Choped y el lapiz de fuego	JK Bowling	304
8	El Bolido da Vinci	Dan Black	500

Mostramos todos ordenados por número de páginas descendente:

ID	TITULO	AUTOR	PAGINAS
2	El Senyor de los Palillos	JRR TolQuien	550
8	El Bolido da Vinci	Dan Black	500
6	Los Juegos de Enjambre	Suzanne Collonins	450
4	Los Polares de la Tierra	Ken Follonet	400
7	Jarry Choped y el lapiz de fuego	JK Bowling	304
5	Jarry Choped y el peluquero de Azkaban	JK Bowling	303
3	Jarry Choped y la Sabana Secreta	JK Bowling	302
1	Jarry Choped y la Hierba Filosofal	JK Bowling	301



Mostramos sólo título y páginas de los libros de JK Bowling:

TITULO	PAGINAS
Jarry Choped y la Hierba Filosofal	301
Jarry Choped y la Sabana Secreta	302
Jarry Choped y el peluquero de Azkaban	303
Jarry Choped y el lapiz de fuego	304

Este 4 no se ha escrito tal cual. Se ha hecho un *echo* de affected_rows

Eliminamos los libros de JK Bowling (4 filas eliminadas), y mostramos todo:

ID	TITULO	AUTOR	PAGINAS
2	El Senyor de los Palillos	JRR TolQuien	550
4	Los Polares de la Tierra	Ken Follonet	400
6	Los Juegos de Enjambre	Suzanne Collonins	450
8	El Bolido da Vinci	Dan Black	500

Modificamos el título del libro 8 y mostramos todo:

ID	TITULO	AUTOR	PAGINAS
2	El Senyor de los Palillos	JRR TolQuien	550
4	Los Polares de la Tierra	Ken Follonet	400
6	Los Juegos de Enjambre	Suzanne Collonins	450
8	El Morbido da Vinci	Dan Black	500



EJERCICIO

Crea una página llamada ***ud04ejer02.php*** y título “UD 4 Ejercicio 2”. Luego, usa *phpMyAdmin* para vaciar la tabla *libros*.

Ahora, en tu página php, crea un formulario como el siguiente:

Id:
Título:
Autor:
Páginas:

ID	TITULO	AUTOR	PAGINAS
----	--------	-------	---------

Aquí abajo se debe mostrar SIEMPRE la tabla de libros, con un select de todos los libros, pero inicialmente no hay ninguno.

- ✓ Todos los campos son obligatorios.
- ✓ Si se pulsa el botón con algún campo sin rellenar, debe salir un aviso.
- ✓ Los campos que **sí** estuvieran rellenos, se conservan .

Id: <-- ¡Debes introducir un id!
Título: <-- ¡Debes introducir un título!
Autor: <-- ¡Debes introducir un autor!
Páginas: <-- ¡Debes introducir el número de páginas!

ID	TITULO	AUTOR	PAGINAS
----	--------	-------	---------

- ✓ Cuando se pulse el botón '*Introducir libro*' con todos los campos rellenos, se insertará el libro en la base de datos (*damos por hecho que el usuario va a poner un int en id, un texto en titulo, etc., por lo que no hace falta comprobar eso en este ejercicio*)

Así se debe quedar la página tras insertar el primer libro.
Los campos del formulario quedarán vacíos, preparados para seguir insertando.

Id:
Título:
Autor:
Páginas:

ID	TITULO	AUTOR	PAGINAS
1	Jarry Choped y la Hierba Filosofal	JK Bowling	301

No es mala idea, mientras se hacen estos ejercicios, tener abierto phpMyAdmin para ir viendo qué pasa en la BD. A lo mejor, por ejemplo, en nuestra página no nos sale lo que queremos pero sí se ha hecho el cambio en la BD.