

UT6. Utilización de técnicas de acceso a datos

DAW - Desarrollo Web Entorno Servidor
Fernando Galindo

1. Introducción
2. MySQL y herramientas
3. Utilización y acceso a BD (MySQLi y PDO)
4. Usuarios y permisos en MySQL

1. Introducción

Uno de los usos más extendidos de los servidores Web es generar una interface para acceder y gestionar información almacenada en una base de datos.

PHP nos permite trabajar con BD para extraer y mostrar información o actualizar registros almacenados en BD.

PHP trabaja de forma directa con distintos motores de BD: MySQL, SQLite3, SQLSRV, PostgreSQL, IBM DB2, MongoDB ...

Así como las clases de abstracción DBA, PDO y ODBC lo que nos permite trabajar con muchas BD.

1. Introducción

Clases de abstracción:

- DBA, nos permite trabajar con BD de tipo Berkeley DB
- PDO, *PHP Data Objects*. Define una interfaz sencilla que permite trabajar con diferentes BD como si fueran objetos.
Independientemente de la BD utilizada, los métodos para trabajar con la BD siguen siendo los mismos lo hace que cambiar de motor de BD sea sencillo.
- ODBC, es un estándar de acceso a BD en el que se usa el API de dicha BD para acceder a ellos

1. Introducción

Trabajaremos con PDO para el manejo de las BDs

Utilizaremos el motor MySQL(**MariaDB**) como BD y utilizaremos los métodos propios de MySQL(PDO) para trabajar con ella

Estos métodos al ser específicos para la BD, ofrecen un mejor rendimiento

Podemos ver las BD soportadas por PDO con la siguiente función

```
<?php
```

```
var_dump(PDO::getAvailableDrivers());
```

```
?>
```

1. Introducción

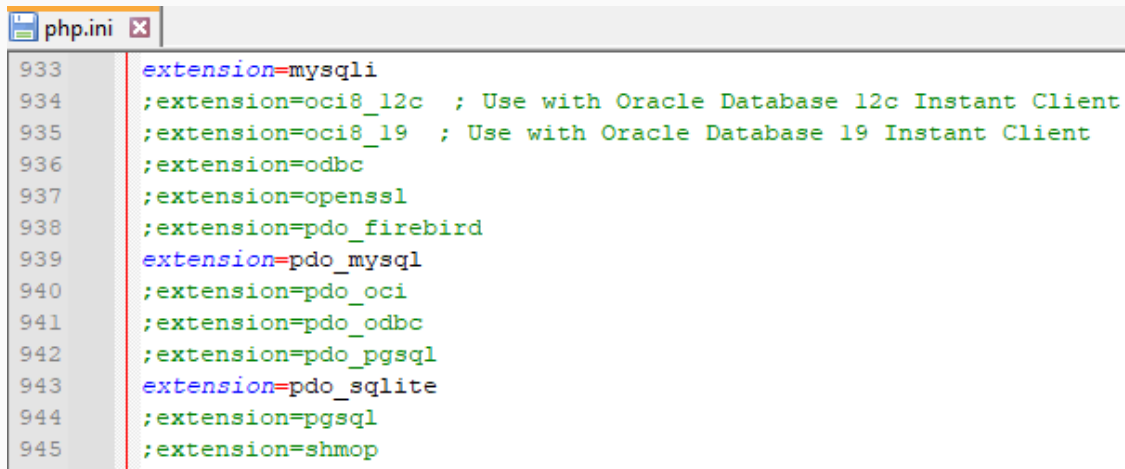
Dentro de PDO sólo MySQL y SQLite están habilitados por defecto

El resto tendríamos que habilitarlas desde el fichero php.ini indicando las extensiones de cada gestor de BD

array (size=2)

0 => string 'mysql' (length=5)

1 => string 'sqlite' (length=6)

A screenshot of a text editor window titled 'php.ini'. The window shows a list of database extensions being configured. The extensions listed are: mysql, oci8_12c, oci8_19, odbc, openssl, pdo_firebird, pdo_mysql, pdo_oci, pdo_odbc, pdo_pgsql, pdo_sqlite, pgsql, and shmop. Each extension is preceded by a semicolon and the word 'extension='.

```
933 extension=mysql
934 ;extension=oci8_12c ; Use with Oracle Database 12c Instant Client
935 ;extension=oci8_19 ; Use with Oracle Database 19 Instant Client
936 ;extension=odbc
937 ;extension=openssl
938 ;extension=pdo_firebird
939 extension=pdo_mysql
940 ;extension=pdo_oci
941 ;extension=pdo_odbc
942 ;extension=pdo_pgsql
943 extension=pdo_sqlite
944 ;extension=pgsql
945 ;extension=shmop
```

1. Introducción

Este pequeño ejemplo vemos cómo nos conectamos a una BD MySQL usando el PDO genérico y el PDO para MySQL

```
// PDO
```

```
$pdo = new PDO("mysql:host=localhost;dbname=database",  
'usuario', 'contraseña');
```

```
// MySQL (PDO)
```

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

2. MySQL y herramientas

MySQL es un gestor de BD relacional que permite utilizar diferentes motores

InnoDB es el motor por defecto que soporta integridad referencial como manipulación transaccional



| ENGINE | SUPPORT | COMMENT | TRANSACTIONS | XA | SAVEPOINTS |
|--------------------|---------|---|--------------|-----|------------|
| CSV | YES | Stores tables as CSV files | NO | NO | NO |
| MRG_MyISAM | YES | Collection of identical MyISAM tables | NO | NO | NO |
| MEMORY | YES | Hash based, stored in memory, useful for temporary... | NO | NO | NO |
| Aria | YES | Crash-safe tables with MyISAM heritage. Used for i... | NO | NO | NO |
| MyISAM | YES | Non-transactional engine with good performance and... | NO | NO | NO |
| SEQUENCE | YES | Generated tables filled with sequential values | YES | NO | YES |
| InnoDB | DEFAULT | Supports transactions, row-level locking, foreign ... | YES | YES | YES |
| PERFORMANCE_SCHEMA | YES | Performance Schema | NO | NO | NO |

2. MySQL y herramientas

Disponemos de diferentes herramientas para poder trabajar con MySQL

- mysqlq (mysqladmin y mysqlshow), terminal de línea de comandos para ejecutar sentencias desde la BD
- MySQL Workbench, es una interfaz genérica nativa que nos permite gestionar tanto el servidor como las BDs

<https://www.mysql.com/products/workbench/>

- phpMyAdmin, es una aplicación Web desarrollada en PHP, que permite la administración de las BD y su gestión

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

2. MySQL y herramientas

mysql -> Es la herramienta nativa y que está disponible, con la que podemos crear usuarios, bases de datos y comprobar el estado del SGBD

```
C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 15
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

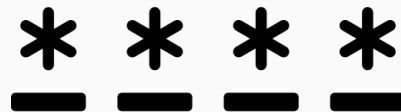
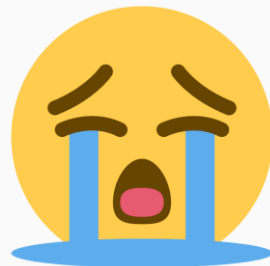
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Fijémonos que nos conectamos con el usuario root y sin contraseña

```
MariaDB [(none)]> create database dwes_ejemplo;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use dwes_ejemplo;
Database changed
MariaDB [dwes_ejemplo]>
```



2. MySQL y herramientas

MySQL Workbench

Xampp

File Edit View Query Database Server Tools Scripting Help

Navigation Administration - Server Status

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Connection Name: Xampp

Host: DESKTOP
Socket: C:/xampp/mysql/mysqlsock
Port: 3306
Version: 10.4.28-MariaDB (mariadb.org binary distribution)
Compiled For: Win64 (x64)
Configuration File: C:\xampp\mysql\bin\my.ini
Running Since: Sat Nov 11 10:38:39 2023 (0:00)

Refresh

Available Server Features

| | | | |
|------------------------------|---------------------------|-------------------------|---------------------------|
| Performance Schema: | <input type="radio"/> Off | Windows Authentication: | <input type="radio"/> Off |
| Thread Pool: | <input type="radio"/> n/a | Password Validation: | <input type="radio"/> n/a |
| Memcached Plugin: | <input type="radio"/> n/a | Audit Log: | <input type="radio"/> n/a |
| Semisync Replication Plugin: | <input type="radio"/> n/a | Firewall: | <input type="radio"/> n/a |
| SSL Availability: | <input type="radio"/> Off | Firewall Trace: | <input type="radio"/> n/a |

Server Directories

| | |
|-------------------------|---|
| Base Directory: | <input checked="" type="radio"/> On |
| Data Directory: | <input checked="" type="radio"/> On |
| Disk Space in Data Dir: | 272.21 GB of 465.12 GB available |
| Plugins Directory: | <input checked="" type="radio"/> On |
| Tmp Directory: | <input checked="" type="radio"/> On |
| Error Log: | <input checked="" type="radio"/> On .\mysql_error.log |
| General Log: | <input type="radio"/> Off |
| Slow Query Log: | <input type="radio"/> Off |

Replica

: this server is not a replica in a replication setup

Authentication

| | |
|------------------------------|---------------------------|
| SHA256 Password Private Key: | <input type="radio"/> n/a |
| SHA256 Password Public Key: | <input type="radio"/> n/a |

Output



Server Status: Running
CPU Load: 16%
Connections: 5
Traffic: 4.91 KB/s
Key Efficiency: 0.0%
Selects per Second: 0
InnoDB Buffer Usage: 41.8%
InnoDB Reads per Second: 0
InnoDB Writes per Second: 0

Administration Schemas

Information

Connection Details

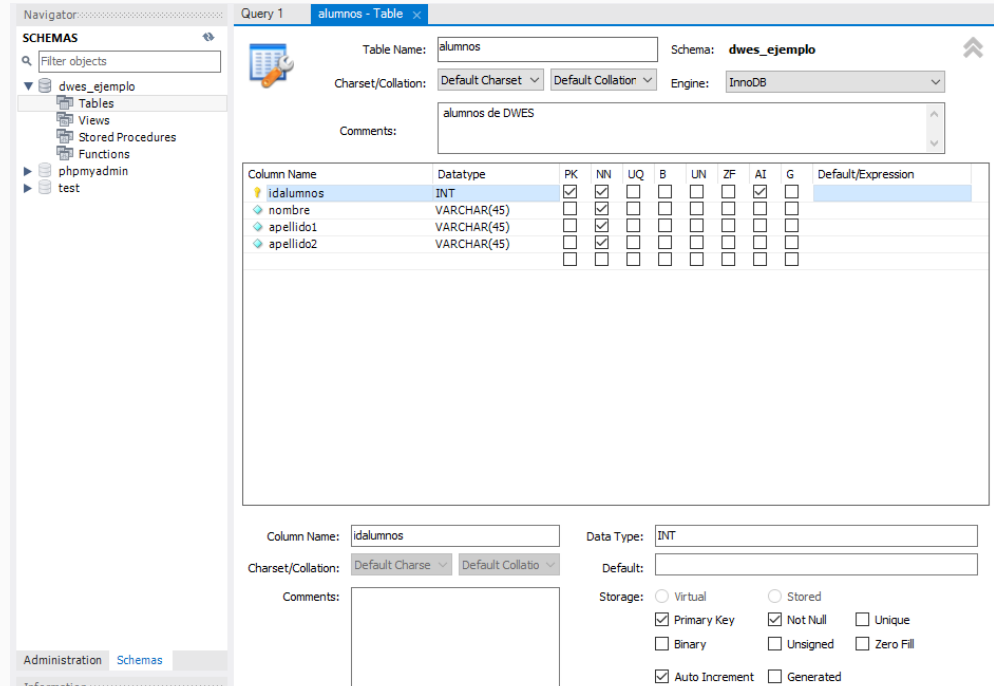
Name: Xampp
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: SSL not used



2. MySQL y herramientas

MySQL Workbench es una herramienta gráfica más amigable en la que podemos gestionar la BD, cuadros de mandos con el estado de la misma y podemos arrancar y parar la misma

Podemos crear tablas sobre el esquema creado en modo comando, junto con sus columnas y propiedades



2. MySQL y herramientas

Obtenemos los comandos utilizados para crear y operar con nuestra BD

Resulta útil poder ir guardando estos comandos, para poder tener un script a la hora de poder crear un script y poder exportar nuestra BD, o bien crear una tabla igual con los mismos campos

Review the SQL Script to be Applied on the Database

```
1 CREATE TABLE `dwes_ejemplo`.`alumnos` (  
2   `idalumnos` INT NOT NULL AUTO_INCREMENT,  
3   `nombre` VARCHAR(45) NOT NULL,  
4   `apellido1` VARCHAR(45) NOT NULL,  
5   `apellido2` VARCHAR(45) NOT NULL,  
6   PRIMARY KEY (`idalumnos`))  
7   COMMENT = 'alumnos de DWES';  
8
```

2. MySQL y herramientas

phpMyAdmin, es el modo Web y que usaremos por defecto

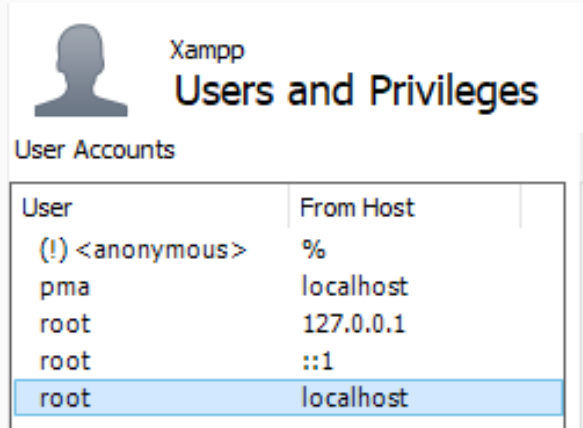
The screenshot shows the phpMyAdmin web interface in a browser. The address bar indicates the URL: `localhost/phpmyadmin/index.php?route=/table/change&db=dwes_ejemplo&table=alumnos`. The interface includes a sidebar on the left with a tree view of databases and tables, including 'dwes_ejemplo' and 'alumnos'. The main area displays the table structure for 'alumnos' with columns: 'idalumnos' (int(11)), 'nombre' (varchar(45)), 'apellido1' (varchar(45)), and 'apellido2' (varchar(45)). The 'Valor' column shows the current values: 'Fernando' for 'nombre', 'Galindo' for 'apellido1', and 'Merino' for 'apellido2'. A 'Continuar' button is visible at the bottom right.

| Columna | Tipo | Función | Nulo | Valor |
|-----------|-------------|---------|------|----------|
| idalumnos | int(11) | | | |
| nombre | varchar(45) | | | Fernando |
| apellido1 | varchar(45) | | | Galindo |
| apellido2 | varchar(45) | | | Merino |

2. MySQL y herramientas

Establezcamos una contraseña a nuestro servidor de BD


Fijémonos, que el usuario root aparece varias veces dependiendo de cómo nos estemos conectando usando IPv4, IPv6 o FQDN



Xampp
Users and Privileges

User Accounts

| User | From Host |
|-----------------|-----------|
| (!) <anonymous> | % |
| pma | localhost |
| root | 127.0.0.1 |
| root | ::1 |
| root | localhost |



¡Acceso denegado!

127.0.0.1/phpmyadmin/

Bienvenido a phpMyAdmin

Error

MySQL ha dicho: ⓘ

No se estableció la conexión: los parámetros están incorrectos.

❗ mysqli::real_connect(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: NO)

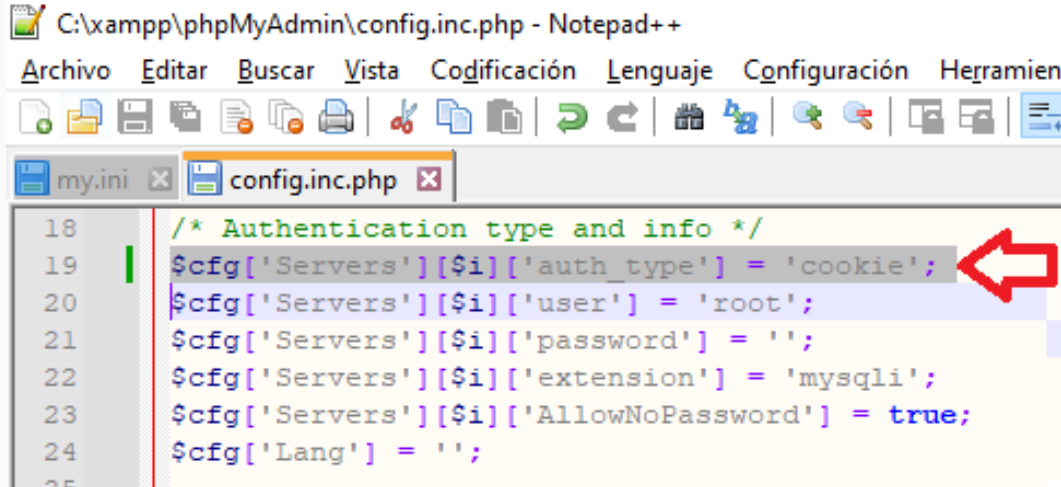
❗ phpMyAdmin intentó conectarse con el servidor MySQL, y el servidor rechazó esta conexión. Deberá revisar el host, nombre de usuario y contraseña en config.inc.php y asegurarse que corresponden con la información provista por el administrador del servidor MySQL.

Reintentar conexión

2. MySQL y herramientas

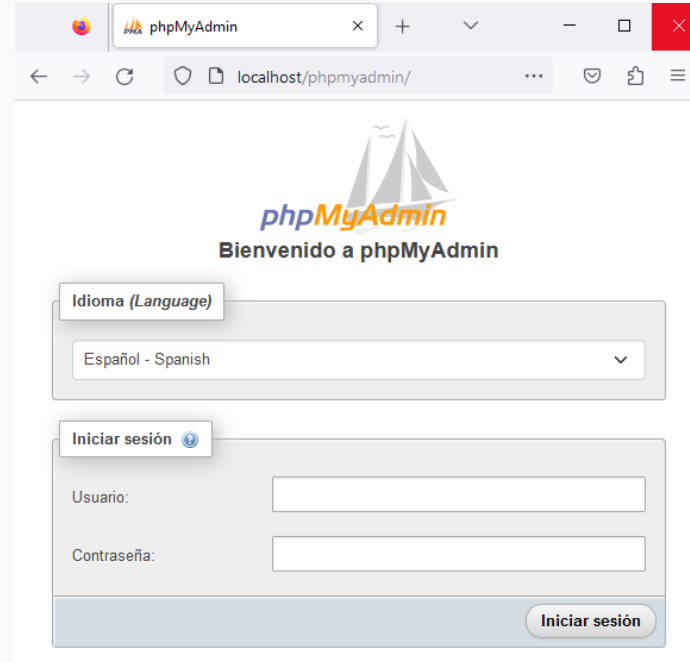
Buscamos el fichero de configuración de phpmyadmin (config.inc.php)

Modificamos el auth_type a cookie



```
C:\xampp\phpMyAdmin\config.inc.php - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas

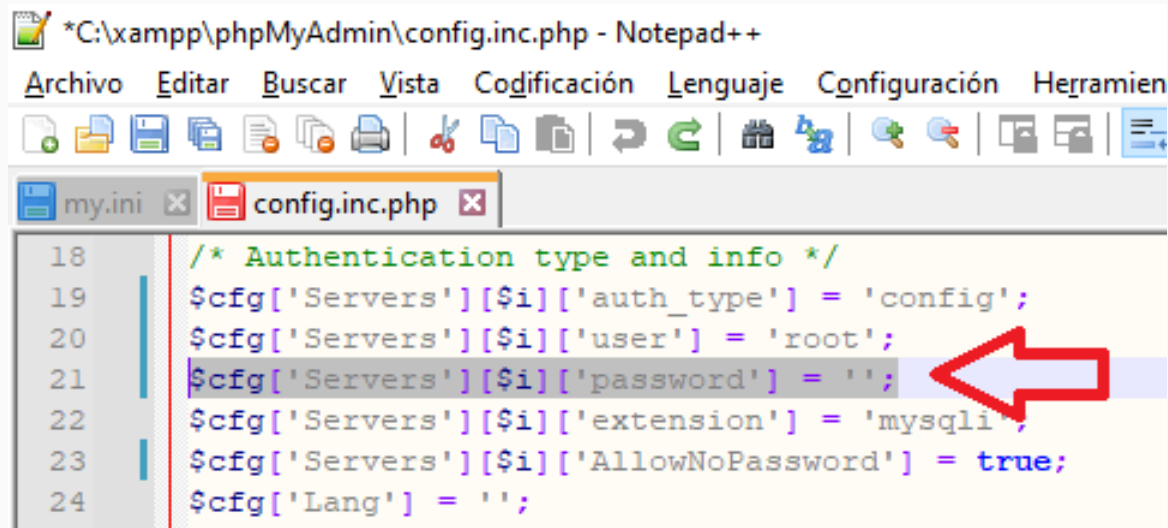
18  /* Authentication type and info */
19  $cfg['Servers'][$i]['auth_type'] = 'cookie';
20  $cfg['Servers'][$i]['user'] = 'root';
21  $cfg['Servers'][$i]['password'] = '';
22  $cfg['Servers'][$i]['extension'] = 'mysqli';
23  $cfg['Servers'][$i]['AllowNoPassword'] = true;
24  $cfg['Lang'] = '';
```



2. MySQL y herramientas

En nuestro nivel, no resulta conveniente estar escribiendo la contraseña todo el rato, mantendremos la contraseña de la BD, pero la recordaremos en la herramienta phpMyAdmin

```
$cfg['Servers'][$i]['password'] = 'tu_contraseña_secreta';
```



```
*C:\xampp\phpMyAdmin\config.inc.php - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas

my.ini x config.inc.php x
18  /* Authentication type and info */
19  $cfg['Servers'][$i]['auth_type'] = 'config';
20  $cfg['Servers'][$i]['user'] = 'root';
21  $cfg['Servers'][$i]['password'] = '';
22  $cfg['Servers'][$i]['extension'] = 'mysqli';
23  $cfg['Servers'][$i]['AllowNoPassword'] = true;
24  $cfg['Lang'] = '';
```


2. MySQL y herramientas

Autenticación mediante sesiones y PHP

```
$cfg['Servers'][$i]['auth_type'] = 'signon';
```

<https://docs.phpmyadmin.net/en/latest/setup.html#signon-authentication-mode>

```
/* Authentication type and info */  
$cfg['Servers'][$i]['auth_type'] = 'signon';  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';  
$cfg['Servers'][$i]['extension'] = 'mysqli';  
$cfg['Servers'][$i]['AllowNoPassword'] = true;  
$cfg['Lang'] = '';
```



3. Utilización y acceso a BD

Estableciendo nuestra primera conexión a la BD

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

Estamos trabajando con objetos, lo veremos en más detalle en la siguiente unidad de trabajo

De esta forma `$conexión` es una variable de tipo objeto

```
var_dump($conexion);
```

Podemos acceder a sus propiedades con

```
echo $conexion->server_info; //Devuelve 10.4.28-MariaDB
```

3. Utilización y acceso a BD

Antes de realizar ninguna operación debemos comprobar que la conexión se ha realizado correctamente

Podríamos utilizar el siguiente código:

```
if ($conexion->connect_error != null) {  
  
    echo "<p>Error $conexion->connect_error conectando a la  
base de datos: $conexion</p>";  
  
    exit();  
  
}
```

3. Utilización y acceso a BD

```
//Creamos una conexión con la BD
$conexion = new mysqli('localhost', 'usuario', 'contraseña',
'base_de_datos');
$error=$conexion->connect_error
//Comprobamos que la conexión se ha realizado correctamente
if ($error != null) {
    echo "<p>Error $conexion->connect_error conectando a la base de datos:
$conexion</p>";
    exit();
}
//Operaciones con la BD

//Cerramos la conexión
$conexion->close();
```

3. Utilización y acceso a BD

Ejecución de consulta método query

```
$resultado=$conexion->query("insert into alumnos (nombre, apellido1,apellido2)  
values ('Juan','Ejemplo','Ejemplo')");
```

```
//Resultado es un objeto que podemos comprobar si un booleano
```

```
if($resultado){  
    echo 'Consulta ejecutada correctamente<br>';  
}
```

Este funcionamiento sería
equivalente para los borrados
o update

(delete from...)

SELECT * FROM `alumnos`

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo

 | Número de filas:

25

▼

 Filtar filas:

Buscar en esta tabla

Opciones extra

← T →

▼ idalumnos nombre apellido1 apellido2

☐

Editar

Copiar

Borrar

1 Fernando Galindo Merino

☐

Editar

Copiar

Borrar

2 Raquel Bueno Guzmán

☐

Editar

Copiar

Borrar

3 Juan Ejemplo Ejemplo

3. Utilización y acceso a BD

Ejecución de consulta método `query`, y manejo de resultados (**arrays**)

```
$resultado=$conexion->query('select nombre,apellido1 from alumnos');  
//Resultado es un objeto que podemos comprobar si un booleano  
if($resultado) {  
    echo 'Consulta ejecutada correctamente<br>';  
    // fetch_array devuelve una fila en formato de array asociativo y numérico  
    $registros = $resultado->fetch_array(); // Obtenemos el primer registro  
    echo "El alumno es ".$registros['0']." ".$registros['1']."<br>";  
    echo "El alumno es ".$registros['nombre']." ".$registros['apellido1'];  
}
```

Consulta ejecutada correctamente
El alumno es Fernando Galindo
El alumno es Fernando Galindo

3. Utilización y acceso a BD

Podemos cambiar el comportamiento de `fetch_array`

Tendríamos un array numérico:

```
$registros = $resultado->fetch_array(MYSQLI_NUM);  
// fetch_row es equivalente a la anterior  
$registros = $resultado->fetch_row();
```

Tendríamos un array asociativo:

```
$registros = $resultado->fetch_array(MYSQLI_ASSOC);  
// fetch_assoc es equivalente a la anterior  
$registros = $resultado->fetch_assoc();
```

Tenemos los dos arrays, comportamiento por defecto

```
$registros = $resultado->fetch_array(MYSQLI_BOTH);
```


3. Utilización y acceso a BD

Ejecución de consulta método `query`, y manejo de resultados (**objetos**)

```
$resultado=$conexion->query('select nombre,apellido1 from alumnos');  
//Resultado es un objeto que podemos comprobar si un booleano  
if($resultado) {  
    echo 'Consulta ejecutada correctamente<br>';  
    // fetch_object devuelve una fila en formato de objeto  
    $registros = $resultado->fetch_object(); // Obtenemos el primer registro  
    echo "El alumno es ".$registros->nombre." ".$registros->apellido1;  
}
```

Consulta ejecutada correctamente
El alumno es Fernando Galindo

3. Utilización y acceso a BD

Recuperar más resultados variable `num_rows`

En el caso de update y delete número de filas modificadas `affected_rows`

```
if($resultado){  
    echo 'Consulta ejecutada correctamente<br>';  
    echo "La consulta tiene $resultado->num_rows filas <br>";  
    $registros = $resultado->fetch_assoc(); // Obtenemos el primer  
registro  
    echo "El alumno es ".$registros['nombre']."  
".$registros['apellido1'];  
}
```

Consulta ejecutada correctamente
La consulta tiene 2 filas
El alumno es Fernando Galindo

3. Utilización y acceso a BD

Una posible forma de recorrer todas las filas es con un bucle while

```
$registros=$resultado->fetch_assoc();  
while($registros){  
    echo "El alumno es ".$registros['nombre']."  
    ".$registros['apellido1']."<br>";  
    $registros = $resultado->fetch_assoc(); // Obtenemos el siguiente  
registro  
}
```

```
El alumno es Fernando Galindo  
El alumno es Raquel Bueno
```

3. Utilización y acceso a BD

El servidor debe analizar la consulta antes de ejecutarse, determinadas consultas con operaciones de inserción que se realizan muchas veces pueden retrasar la ejecución

Para solucionarlo se trabaja con **consultas preparadas** que son almacenadas en el servidor y posteriormente se ejecutan

```
$conexion = new mysqli('localhost', 'usuario', 'contraseña', 'esquema');  
$consulta = $conexion->stmt_init();  
$consulta->prepare('insert into alumnos (nombre, apellido1,apellido2) values  
(\'Juan\',\'Ejemplo\',\'Ejemplo\')');  
$consulta->execute();  
$consulta->close();
```

3. Utilización y acceso a BD

Utilizando las consultas preparadas de la forma anterior no nos aporta ninguna ventaja, pero si utilizamos parámetros dentro de nuestra consulta

```
$consulta->prepare('insert into alumnos (nombre, apellido1,apellido2) values  
(?,?,?)');
```

Asociando los parámetros a variables de nuestro código PHP

```
$consulta->bind_param('sss', $nom, $app1,$app2); // Siempre son variables
```

Donde \$nom, \$app1 y \$app2 son variables de nuestro código

| Carácter | Tipo del parámetro |
|----------|--------------------|
| I | Número entero |
| D | Número real |
| S | Cadena de texto |

3. Utilización y acceso a BD

Aprovechando las consultas preparadas con un ejemplo completo

```
// Creamos una consulta preparada
$consulta = $conexion->stmt_init();
$consulta->prepare('insert into alumnos (nombre, apellido1,apellido2) values
(?,?,?)');
// Creamos unas variables con valores en PHP
$nom='Fernando'; $app1='Galindo'; $app2='Merino';

// Realizamos la asociación entre las variables y la consulta
$consulta->bind_param('sss', $nom, $app1,$app2);

// Realizamos la operación, primera inserción
$consulta->execute();

// Modificamos las variables
$nom='Raquel'; $app1='Bueno'; $app2='Martín';

// Realizamos la operación, segunda inserción
$consulta->execute();
```

3. Utilización y acceso a BD

Más formas de hacer la asociación (bind) de los parámetros

```
$consulta->prepare('insert into alumnos (nombre, apellido1,apellido2) values  
(:nom, :app1, :app2)');
```

```
$consulta->bind_param(':nom', $nom);
```

```
$consulta->bind_param(':app1', $app2);
```

```
$consulta->bind_param(':app2', $app2);
```

Donde `$nom`, `$app1` y `$app2` son variables de nuestro código

3. Utilización y acceso a BD

Más formas de hacer la asociación (bind) de los parámetros

```
$consulta->prepare('insert into alumnos (nombre, apellido1,apellido2) values  
(?,?,?)');
```

```
$consulta->bind_param(1, $nom);
```

```
$consulta->bind_param(2, $app2);
```

```
$consulta->bind_param(3, $app2);
```

Donde `$nom`, `$app1` y `$app2` son variables de nuestro código

3. Utilización y acceso a BD

Consultas preparadas con SELECT

```
$consulta = $conexion->stmt_init();  
$consulta->prepare('SELECT nombre, apellido1 FROM alumnos');  
// Ejecución de la consulta  
$consulta->execute();  
$consulta->bind_result($nom, $app1);  
while($consulta->fetch()) {  
    echo "Usuario $nom y apellido $app1 <br>";  
}
```

3. Utilización y acceso a BD

Las funciones vistas en este documento, tan sólo son las funciones básicas para poder trabajar con BD, la documentación oficial dispone de muchas más funciones y que nos van a permitir realizar muchas más operaciones

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

MySQLi

<https://www.php.net/manual/en/class.mysql.php>

<https://www.php.net/manual/en/class.mysql-result>

Consultas preparadas

<https://www.php.net/manual/es/class.mysql-stmt.php>

3. Utilización y acceso a BD

Objetos PDO, nos soporta diferentes BD por lo que tenemos funciones genéricas para trabajar con diferente gestores

```
$pdo = new PDO("mysql:host=localhost;dbname=esquema", 'usuario', 'contrasena');
```

En el caso de consultas INSERT, UPDATE y DELETE

```
$registros = $pdo->exec('DELETE FROM alumnos WHERE nombre='Fernando');  
  
print "<p>Se han borrado $registros registros.</p>";
```

En el caso de consultas SELECT que devuelven valores

```
$resultado = $pdo->query('SELECT nombre FROM alumnos');  
  
while ($fila=$resultado->fetch()){  
    echo "El nombre del usuario es $fila['nombre'] ";  
}
```

3. Utilización y acceso a BD

Transacciones, es el modo por defecto de tal forma que las operaciones de insert, update y delete se realizan automáticamente y es imposible revertirlas

Podemos deshabilitar el modo transaccional

```
$conexion->autocommit(false); // deshabilitamos el modo transaccional automático  
$conexion->query('DELETE FROM alumnos WHERE nombre="Juan"'); // Inicia una  
transacción  
$conexion->commit();
```

- commit -> Realizar una operación "commit" de la transacción actual, devolviendo true si se ha realizado correctamente o false en caso contrario.
- rollback -> Realizar una operación "rollback" de la transacción actual, devolviendo true si se ha realizado correctamente o false en caso contrario.

4. Usuarios y permisos en MySQL

Utilizar el usuario root en las conexiones no es lo común ya que tiene acceso a todas las BD y privilegios sobre todas las tablas

```
$conexion = new mysqli('localhost', 'root', 'contraseña', 'esquema_bd');
```

Nos permite cambiar de BD sin necesidad de crear una nueva conexión con la función `select_db`

```
$conexion->select_db('otraBD'); // Las consultas se realizan sobre otra BD
```

4. Usuarios y permisos en MySQL

Resulta adecuado definir un usuario por cada base de datos, de esta forma nuestro usuario sólo puede operar con dicha BD

Suponemos que tenemos una BD denominada **BANCO** y crearemos un usuario para dicha BD con una contraseña

```
CREATE USER 'admin_banco'@'localhost' IDENTIFIED BY 'mi_contraseña';
```

```
GRANT ALL PRIVILEGES ON banco.* TO 'admin_banco'@'localhost' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES; //No es obligatorio, se recargan las tablas con los permisos con los nuevos cambios
```

Se ha sido generoso con los permisos dados al usuario (ALL PRIVILEGES)

Recordemos que podemos limitar los permisos del usuario tanto por BD, como por tabla de dicho esquema *CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, SELECT, REFERENCES, RELOAD*

4. Usuarios y permisos en MySQL

Las conexiones estarían definidas de la siguiente forma

```
$conexion = new mysqli('localhost', 'admin_banco', 'mi_contrasenya', 'banco');
```

El resto del código sigue exactamente igual

Si intentamos cambiar de BD, no nos permite realizar la operación

```
$conexion = new mysqli('localhost', 'admin_banco', 'no_lo_digas', 'banco');  
// El código de esta parte se realiza sobre la BD banco  
// Intentamos cambiar de BD  
if ($conexion->select_db('dwes_ejemplo'))  
    echo "Cambio realizado correctamente";  
else  
    echo "No se ha realizado el cambio";
```

No se ha realizado el cambio