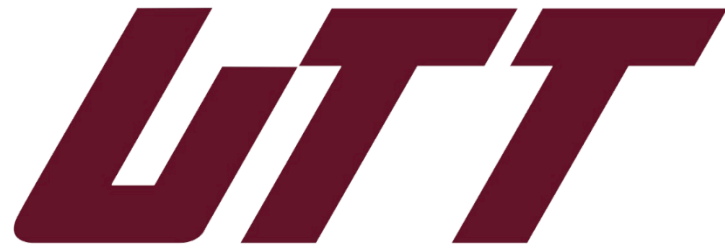


UNIVERSIDAD TECNOLÓGICA DE TIJUANA



*Universidad Tecnológica de Tijuana*

TSU EN TECNOLOGÍAS DE LA INFORMACIÓN

ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA

**MATERIA:**

Desarrollo de Aplicaciones WEB

**Conectar PHP y MYSQL**

**ESTUDIANTE:**

Medina Becerra Alan Oswaldo

**PROFESOR:**

Ray Brunett Parra Galaviz

**GRUPO:**

3E

**FECHA:**

18/10/2024

## ¿Cómo conectar PHP con MYSQL?

Existen dos métodos diferentes para conectar desde PHP a una base de datos MySQL: **MySQLi** y **PDO**.

**MySQLi** es una extensión de MySQL que agrega ciertas funciones a la interfaz de conexión con una base de datos MySQL. Las siglas se corresponden con **MySQL Improved** (MySQL Mejorado). MySQLi soporta tanto el paradigma procedimental como el paradigma de la programación orientada a objetos.

En las primeras versiones de MySQL, se dividen las tareas en **procedimientos lineales**, lo que dificulta la modificación porque se tiene que editar el código desde la parte superior. Sin embargo, en la nueva versión mejorada de MySQLi, se ven los datos como un conjunto de **objetos**, lo que nos permite agregar o eliminar datos de una forma sumamente sencilla.

**PDO** son las siglas de **PHP Data Object** y está orientado exclusivamente al paradigma de la programación orientada a objetos. Una de las grandes ventajas es que admite diferentes tipos de bases de datos que se usan en PHP, como MySQL, MariaDB, PostgreSQL, MSSQL, Informix etc...

Uno de los puntos más importantes que soporta PDO son las **declaraciones preparadas**. Utilizando estas declaraciones, se aceleran las consultas de MySQL.

### -Conexión mediante mysqli

PHP incorpora una función interna que posibilita la conexión a un servidor MySQL. El método utilizado para conectar a la base de datos es **mysqli\_connect()**.

Para simplificar el código y hacerlo un poco mas "friendly", lo normal es crear unas variables que serán pasadas posteriormente como parámetros al método **mysqli\_connect()**. Los cuatro parámetros necesarios que tenemos que pasar a la función son: **\$servername** (dirección del servidor), **\$database** (base de datos), **\$username** (usuario) y **\$password** (contraseña).

Cuando se realiza la conexión a la base de datos, **si no es correcta**, se ejecuta la función **die()**. Lo que hace esta función es detener la ejecución por completo y mostrar un **mensaje de error** y/o advertencia seguido de un mensaje del error exacto que describe el problema.

Si la conexión se realiza de forma correcta, se mostrará un mensaje de "Conexión satisfactoria".

```
<? php
$servername = "servername"; // Nombre/IP del servidor
$dbname = "dbname"; // Nombre de la BBDD
$username = "username"; // Nombre del usuario
$password = "password"; // Contraseña del usuario
// Creamos la conexión
$con = mysqli_connect($servername, $username, $password, $dbname);
// Comprobamos la conexión
if (!$con) {
    die("La conexión ha fallado: " . mysqli_connect_error());
}
echo "Conexión satisfactoria";
mysqli_close($con);
?>
```

## -Conexión mediante PDO

Cuando creamos una conexión a una base de datos MySQL utilizando PDO, necesitamos crear un nuevo objeto PDO con un: (DSN – Data Source Name), nombre de usuario y contraseña.

Deberemos definir un DSN que es el tipo de base de datos, un \$host (nombre o ip del servidor), un \$dbname (nombre de la base de datos), un \$username (nombre de usuario) y un \$password (contraseña). Estas variables las declararemos antes de realizar la conexión.

En este ejemplo, utilizaremos un try/catch que básicamente significa que ejecutará lo que hay dentro del "try" y si hay un error ejecutará la parte del "catch". Si la conexión es correcta, se mostrará el mensaje "Conexión satisfactoria". Si la conexión resulta fallida, se mostrará un mensaje de error y se detendrá la ejecución.

```
<? php
try {
    $host = "hostname";
    $dbname = "databasename";
    $username = "username";
    $password = "password";
    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
    echo "Conexión satisfactoria";
} catch (PDOException $e) {
    die("No se pudo conectar con databasename :". $e->getMessage());
}
?>
```

## Bibliografía

Fernández, R. P. (S.F.). *raulprietofernandez.net*. Obtenido de raulprietofernandez.net:  
<https://www.raulprietofernandez.net/blog/programacion/como-conectar-php-con-una-base-de-datos-mysql#h3-2-diferentes-metodos-de-conexion>