

UNIVERSIDAD TECNOLÓGICA DE TIJUANA



*Universidad Tecnológica de Tijuana*

Ingeniería en Desarrollo y Gestión de Software

Tsu en Tecnologías de la información - Área Desarrollo de Software  
Multiplataforma

3-E

Trabajo: Conectar PHP y MYSQL

Nombre del Alumno: Ariel Torres Iñiguez

Nombre de la Materia: Aplicaciones web

Nombre del Profesor: Ray Brunett Parra Galaviz

Fecha de entrega: 18/10/2024

## Introducción

Para conectar PHP a una base de datos MySQL, se pueden utilizar dos métodos principales: MySQLi o PDO. Ambos ofrecen ventajas según el tipo de proyecto y el nivel de seguridad que se desea implementar.

## Método Usando MySQLi

1- Crear conexión con MySQLi:

```
<?php
```

```
$servername = "localhost";
```

```
$database = "databasename";
```

```
$username = "username";
```

```
$password = "password";
```

```
$conn = mysqli_connect($servername, $username, $password, $database);
```

```
// Verificar la conexión
```

```
if (!$conn) {
```

```
    die("Connection failed: " . mysqli_connect_error());
```

```
}
```

```
echo "Connected successfully";
```

```
mysqli_close($conn);
```

```
?>
```

- `mysqli_connect()` establece la conexión a la base de datos.
- Si la conexión falla, se muestra el mensaje de error con `mysqli_connect_error()`.
- `mysqli_close()` cierra la conexión al finalizar el script.

2- Consulta a la base de datos: Después de conectarte, se pueden realizar consultas utilizando `mysqli_query()`:

```
$query = "SELECT * FROM your_tablename";
```

```
$result = mysqli_query($conn, $query);
```

```
if ($result) {
```

```
    while ($row = mysqli_fetch_assoc($result)) {
```

```
        echo "Nombre: " . $row['your_field'] . "<br>";
```

```
    }
```

```
} else {
```

```
    echo "Error en la consulta: " . mysqli_error($conn);
```

```
}
```

## Método Usando PDO

1- Crear conexión con PDO:

```
<?php
```

```
$host = 'localhost';
```

```
$dbname = 'databasename';
```

```
$username = 'username';
```

```
$password = 'password';
```

```
try {
```

```
    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
```

```
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
    echo "Connected successfully";
```

```
} catch (PDOException $e) {
```

```
    die("Connection failed: " . $e->getMessage());
```

```
}
```

```
?>
```

- PDO permite conectarse no solo a MySQL, sino también a otras bases de datos.
- Se utiliza un bloque try...catch para manejar excepciones y errores de conexión.
- PDO es seguro contra inyecciones SQL y permite usar sentencias preparadas.

2- Consulta a la base de datos con PDO:

```
$query = "SELECT * FROM your_tablename";
```

```
$stmt = $conn->query($query);
```

```
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
```

```
    echo "Nombre: " . $row['your_field'] . "<br>";
```

```
}
```

- MySQLi es ideal si solo necesitas trabajar con bases de datos MySQL y prefieres trabajar en un estilo procedural u orientado a objetos.
- PDO es más flexible si necesitas conectarte a diferentes tipos de bases de datos y manejar excepciones de manera más robusta.

**Referencias:**

Gustavo, B. (2023, Mayo 6). Cómo conectar PHP con Bases de datos MySQL. Hostinger Tutoriales. <https://www.hostinger.mx/tutoriales/conectar-php-mysql>

SiteGround. (s.f.). Conectarse a una base de datos MySQL. SiteGround.

<https://es.siteground.com/tutoriales/php-mysql/conectarse-base-datos/>

Hostinet. (2024, Abril 18). Cómo Conectar PHP con Bases de Datos MySQL. Hostinet.

<https://www.hostinet.com/formacion/bases-de-datos/conectar-php-bases-datos-mysql/>