Alumno: Bravo Flores Selegna Odracir

Matricula: 0322103684

Docente: Ray Brunett Parra Galaviz

Grupo: 3E

Asignatura: Aplicaciones Web

Tarea: Conectar PHP y MYSQL

16/10/2024

Para conectar **PHP** con **MySQL**, tienes dos opciones principales: usar la extensión **MySQLi** o **PDO (PHP Data Objects)**. Ambas permiten interactuar con bases de datos, pero tienen algunas diferencias importantes en cuanto a flexibilidad y seguridad.

Conexión con MySQLi:

1. **Crear conexión**: Con mysqli, puedes establecer la conexión pasando los parámetros del servidor, usuario, contraseña y base de datos:

```
php
Copiar código
<?php
$servidor = "localhost";
$usuario = "usuario";
$password = "contraseña";
$base_datos = "nombre_de_base_de_datos";

$conn = new mysqli($servidor, $usuario, $password, $base_datos);

if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
echo "Conexión exitosa";
?>

```

2. **Consulta**: Luego de la conexión, puedes realizar consultas usando mysqli_query para operaciones como SELECT, INSERT, UPDATE, o DELETE. Es recomendable usar declaraciones preparadas para evitar inyecciones SQL.

Conexión con PDO:

1. **Ventajas**: **PDO** es más flexible ya que soporta múltiples tipos de bases de datos (MySQL, PostgreSQL, SQLite, etc.) y ofrece seguridad mejorada con consultas preparadas.

2. **Ejemplo de conexión**:

```
php
Copiar código
<?php
$dsn = 'mysql:host=localhost;dbname=nombre_de_base_de_datos';
$usuario = 'usuario';
$password = 'contraseña';
```

```
9.
10. try {
11.     $conn = new PDO($dsn, $usuario, $password);
12.     $conn->setAttribute(PDO::ATTR_ERRMODE,
    PDO::ERRMODE_EXCEPTION);
13.     echo "Conexión exitosa con PDO";
14. } catch (PDOException $e) {
15.     echo "Error de conexión: " . $e->getMessage();
16. }
17. ?>
18.
```

19. **Consultas preparadas**: Para evitar vulnerabilidades de seguridad, como la inyección SQL, usa consultas preparadas al realizar operaciones:

```
20. php
21. Copiar código
22. $stmt = $conn->prepare("INSERT INTO tabla (columna) VALUES (:valor)");
23. $stmt->bindParam(':valor', $valor);
24. $stmt->execute();
25.
```

Mejores prácticas:

- Siempre valida y sanitiza los datos antes de insertarlos en la base de datos para evitar ataques.
- Usa privilegios mínimos de usuario en MySQL, asignando solo los necesarios.
- Cierra la conexión al finalizar el script.

Con estas conexiones podrás manipular datos en tu base de datos MySQL desde PHP de manera eficiente y segura. Si quieres profundizar, puedes revisar más ejemplos sobre el uso de **MySQLi** o **PDO** para adaptar la conexión según tus necesidades.

REFERENCIAS:

- Gustavo, b. (mayo 06, 2023). Cómo conectar PHP con Bases de datos MySQL. Recuperado el 18 de octubre de 2024 de https://www.hostinger.es/tutoriales/conectar-php-mysql
- Cyberstream. (noviembre 24, 2023). Conexión entre PHP y MySQL en un servidor web: Guía completa y detallada. Recuperado el 18 de octubre de 2024 de https://www.byronvargas.com/web/como-conectar-php-con-mysql-en-un-servidor-web/