

¿Qué es MVC?

MVC (Modelo-Vista-Controlador) es un patrón de diseño de software que separa la lógica de la aplicación en tres componentes conectados. Este patrón se usa ampliamente en el desarrollo web para crear aplicaciones escalables.

- **Modelo:** Representa los datos y la lógica de la aplicación (Backend)
- **Vista:** Se encarga de la presentación y formato de los datos al usuario final (Frontend)
- **Controlador:** Actúa como intermediario entre el modelo y la vista, procesando todas las solicitudes del usuario pudiendo responder solicitudes propiamente sin necesidad de invocar al modelo (Backend)

¿Cuáles son los componentes de MVC y cómo aplicarlo?

1. Modelo

- Gestiona los datos y la lógica de la aplicación.
- Se conecta con la base de datos.
- Es independiente de la presentación.

Ejemplo básico de un Modelo:

```
<?php
0 references | 0 implementations
class UserModel {
    3 references
    private $bd;

    0 references | 0 overrides
    public function __construct($bd) {
        $this->bd = $bd;
    }

    0 references | 0 overrides
    public function obtenerUser($ci) {
        $stmt = $this->bd->prepare("SELECT * FROM usuarios WHERE ci = ?");
        $stmt->execute([$ci]);
        return $stmt->fetch();
    }

    0 references | 0 overrides
    public function crearUser($nombre, $email, $password) {
        $stmt = $this->bd->prepare("INSERT INTO usuarios (nombre, email, password) VALUES (?, ?, ?)");
        return $stmt->execute([$nombre, $email, password_hash($password, PASSWORD_DEFAULT)]);
    }
}
```

2. Vista

- Se encarga de la presentación de los datos al usuario.
- Generalmente son archivos HTML con código PHP embebido.
- No debe contener ninguna parte de la lógica.

Ejemplo básico de una Vista:

```
<!DOCTYPE html>
<html>
<head>
  <title>Perfil de Usuario</title>
</head>
<body>
  <h1>Perfil de <?php echo htmlspecialchars($user['username']); ?></h1>
  <p>Email: <?php echo htmlspecialchars($user['email']); ?></p>
</body>
</html>
```

3. Controlador

- Recibe las solicitudes del usuario.
- Se comunica con el modelo para obtener o modificar datos.
- Selecciona la vista adecuada para presentar los datos.

Ejemplo básico de un Controlador:

<?php

0 references | 0 implementations

class UserController {

3 references

private \$userModel;

0 references | 0 overrides

public function __construct(\$bd) {

 \$this->userModel = new UserModel(\$bd);

}

0 references | 0 overrides

public function mostrarProfile(\$ci) {

 \$user = \$this->userModel->obtenerUser(\$ci);

 if (\$user) {

 include 'views/userprofile.php';

 } else {

 echo "Usuario no existe";

 }

}

0 references | 0 overrides

public function registrar() {

 if (\$_SERVER['REQUEST_METHOD'] === 'POST') {

 \$nombre = \$_POST['nombre'];

 \$email = \$_POST['email'];

 \$password = \$_POST['password'];

 if (\$this->userModel->crearUser(\$nombre, \$email, \$password)) {

 echo "Usuario registrado con éxito";

 } else {

 echo "Error al registrar el usuario";

 }

 } else {

 include 'views/registro.php';

 }

}