

Patrón arquitectónico MVC

(Modelo-Vista-Controlador)

En muchas ocasiones se mezcla el código propio de la lógica de la aplicación, con el código necesario para crear la interface web que se presenta a los usuarios.

Existen varios métodos que permiten separar la lógica de presentación (en nuestro caso, la que genera las etiquetas HTML) de la lógica de negocio, donde se implementa la lógica propia de cada aplicación.

El más extendido es el patrón de diseño Modelo - Vista - Controlador. Este patrón pretende dividir el código en tres partes, dedicando cada una, a una función definida y diferenciada de las otras:

- **Modelo:** Es el encargado de manejar los datos propios de la aplicación. Debe proveer mecanismos para obtener y modificar la información del mismo. Si la aplicación utiliza algún tipo de almacenamiento para su información (como un SGBD), tendrá que encargarse de almacenarla y recuperarla.
- **Vista:** Es la parte del modelo que se encarga de la interacción con el usuario. En esta parte se encuentra el código necesario para generar la interface de usuario (en nuestro caso en HTML), según la información obtenida del modelo.
- **Controlador:** En este módulo se decide qué se ha de hacer, en función de las acciones del usuario con su interface. Con esta información, interactúa con el modelo para indicarle las acciones a realizar y, según el resultado obtenido, envía a la vista las instrucciones necesarias para generar la nueva interface.

La arquitectura MVC separa la **lógica de negocio** (el modelo) y la **presentación** (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

Si por ejemplo, una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo, manteniendo el controlador y el modelo original.

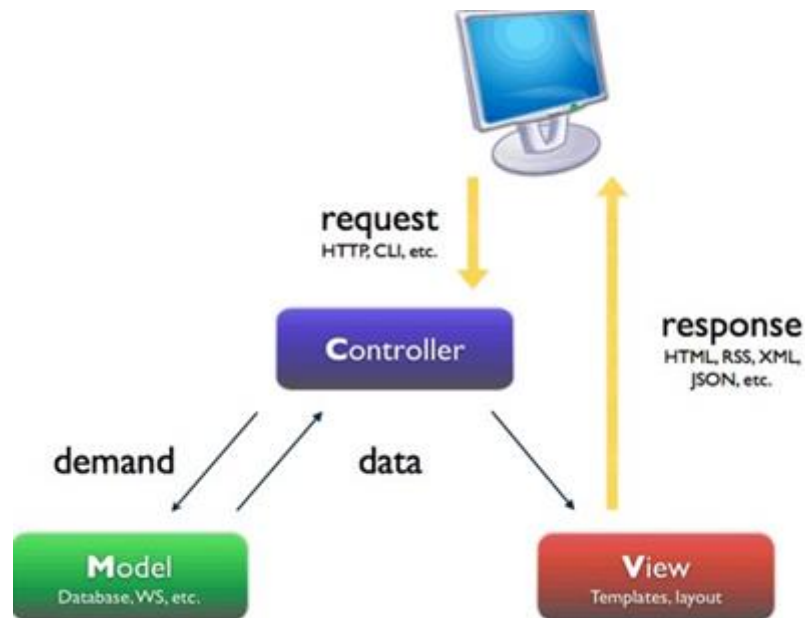
El controlador se encarga de **aislar al modelo y a la vista de los detalles del protocolo** utilizado para las peticiones (HTTP, consola de comandos, email, etc.).

El modelo se encarga de la **abstracción de la lógica relacionada con los datos**, haciendo que la vista y las acciones sean independientes

de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

Funcionamiento MVC

1. El usuario envía petición al controlador vía una URL
2. El controlador solicita al modelo los datos
3. El modelo devuelve los datos
4. El controlador selecciona una vista
5. Se devuelve la vista seleccionada al controlador
6. El controlador devuelve una vista (página php) que carga los datos del modelo seleccionado



Aunque se puede programar utilizando MVC por tu cuenta, es más habitual utilizar el patrón MVC en conjunción con un framework o marco de desarrollo. Existen numerosos frameworks disponibles en PHP, muchos de los cuales incluyen soporte para MVC.

Para poder entender las ventajas de utilizar el patrón MVC, se va a transformar una aplicación simple realizada con PHP en una aplicación que sigue la arquitectura MVC.