

Patrón Arquitectónico MVC

{modelo-vista-controlador}

Arquitectura de software

Es la forma en la que se organizan los componentes de un sistema, interactúan y se relacionan entre sí y con el contexto, aplicando normas y principios de diseño y calidad, que fortalezcan y fomenten la usabilidad a la vez que dejan preparado el sistema, para su propia evolución.

Arquitectura de software

- Estilo Arquitectónico
 - Describir la estructura general de un sistema, independientemente de otros estilos
 - Definir los componentes del sistema, su relación e interactividad
 - Ejemplos : flujo de datos, llamada y retorno, etc.
- El Patrón Arquitectónico es el nivel en el cual la arquitectura de software:
 - Define la estructura básica de un sistema, pudiendo estar relacionado con otros patrones
 - Representa una plantilla de construcción que provee un conjunto de subsistemas aportando las normas para su organización
 - Ejemplos : Capas, MVC, Tuberías y Filtros, Pizarra, etc.
- El Patrón de Diseño la finalidad es la de precisar en detalle los subsistemas y componentes de la aplicación
 - Ejemplos : Proxy, Command, Factory, etc..

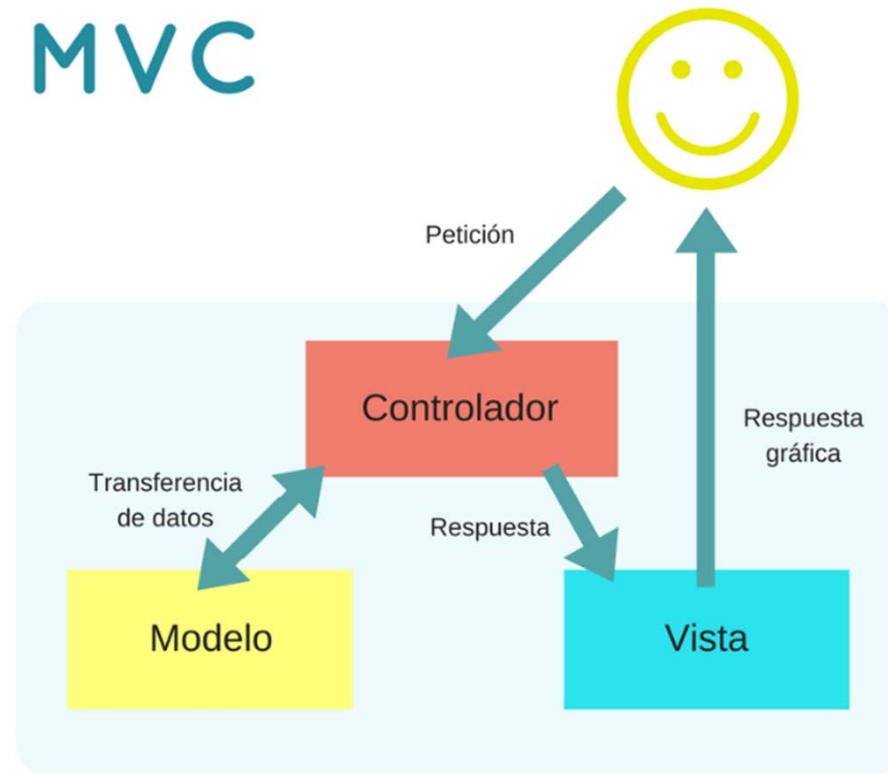
Patrón MVC

Es un **patrón de arquitectura de software** encargado **de separar la lógica de negocio de la interfaz del usuario** y es el más utilizado en aplicaciones Web, ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla, a la vez que **permite** *“no mezclar lenguajes de programación en el mismo código”*.

Patrón MVC

- **Modelo:** representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como “intermediario” con la base de datos.
- **Vista:** es la encargada de mostrar la información al usuario de forma gráfica y “humanamente legible”.
- **Controlador:** es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario.

Patrón MVC



Ejemplo

<https://www.adaweb.es/modelo-vista-controlador-mvc-en-php-actualizado-2022/>



[PHP](#) [Mysql](#) [Javascript](#) [jQuery](#) [CSS](#) [WordPress](#) [Prestashop](#) [Servidor](#) [Sistemas Operativos](#)

Modelo vista controlador (MVC) en PHP
[Actualizado 2022]

> ¿Qué es MVC?

> Definiciones

> ¿Qué nos permite MVC?

> ¿Por qué deberías usar MVC?

> Ejemplo práctico

- > Base de datos
- > Estructura de ficheros
- > config/config.php
- > controller/note.php
- > model/db.php
- > model/note.php
- > view/template/footer.php
- > view/template/header.php
- >
- > view/confirm_delete_note.php
- > view/delete_note.php
- > view/edit_note.php
- > view/list_note.php
- > index.php

 Modelo vista controlador (MVC) en PHP
[Actualizado 2022]

📅 Creado: 10/02/2022 📅 Actualizado: 10/02/2022 📁 PHP

MVC es un patrón de arquitectura de software, que aporta cierto orden y escalabilidad en los desarrollos. En un primer momento puede resultar intimidante o complejo, pero si lo analizamos con calma, comprobaremos que tal dificultad no es real.

Como programadores deberemos conocer este paradigma y la forma correcta de aplicarlo. A continuación veremos una pequeña explicación y un ejemplo muy simple en PHP.

¿Qué es MVC?

MVC significa modelo (model) vista (view) controlador (controller). Esto es lo que significan cada uno de esos componentes.

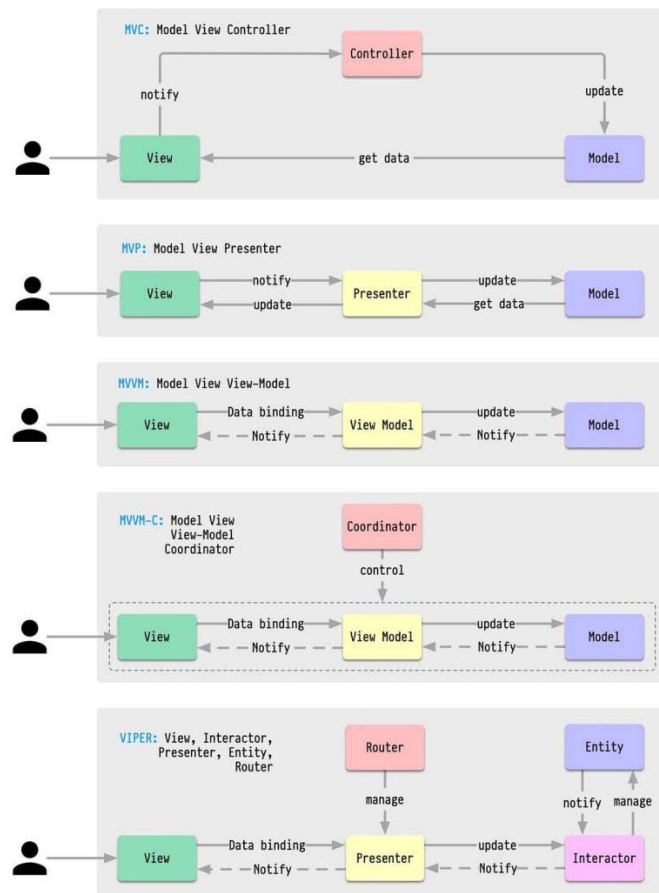
Definiciones

Modelo: se encarga de, por ejemplo, cargar datos y realizar operaciones en ellos.

Vista: el frontend o interfaz gráfica de usuario (GUI), es la capa que ve el usuario.

Controlador: se encarga de solicitar datos al modelo y enviarlos a la vista. Interconecta el frontend con el backend.

Tarea 😊



<https://dev.to/ayushsoni1010/mvc-mvp-mvvm-mvvm-c-and-viper-architecture-patterns-1l3g>