

Hola Mundo en XAMPP

Patrón MVC

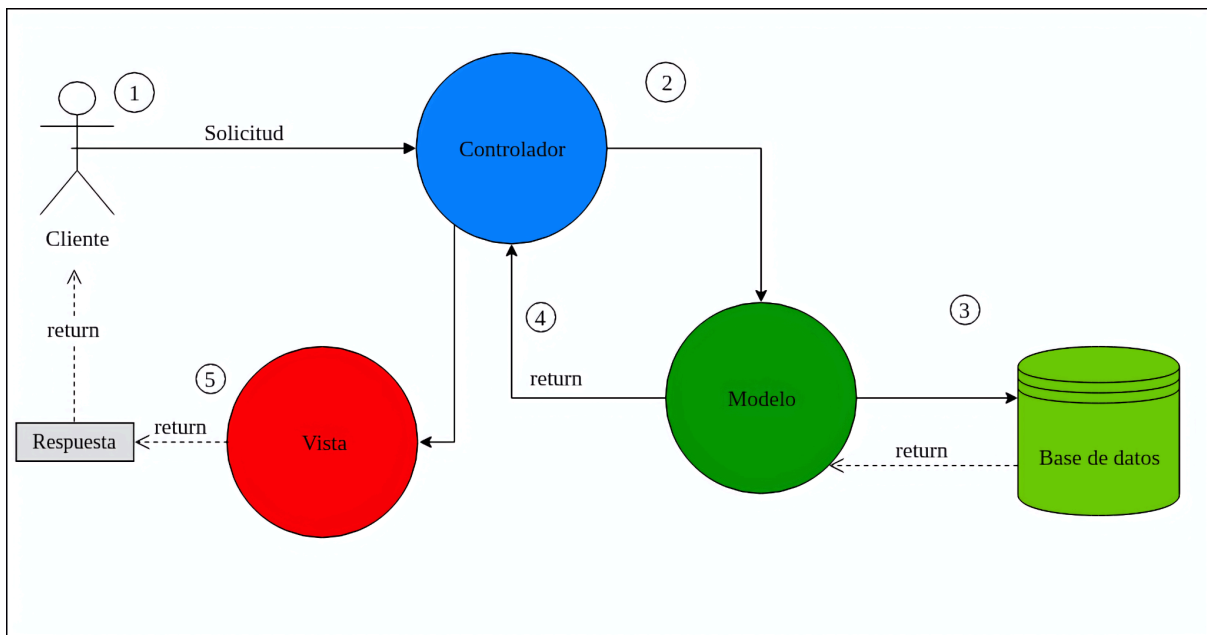
MVC: **Model, View & Controller**. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad.

- **Modelo:** Es la capa donde se trabaja con los datos. El modelo contiene las **entidades** que representan el dominio, la **lógica de negocio**, y los **mecanismos de persistencia** de nuestro sistema. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes selects, updates, inserts, etc. También es habitual utilizar otras librerías como PDO o algún ORM, que nos permiten trabajar con abstracción de bases de datos y persistencia en objetos. Por ello, en vez de usar directamente sentencias SQL se utiliza un dialecto de acceso a datos basado en clases y objetos. Si nuestra aplicación forma parte de un sistema distribuido, es decir, consume servicios prestados por otros sistemas, en el modelo encontraremos las clases de transferencia de datos (DTO, Data Transfer Objects) que nos permitirán intercambiar información con ellos.
- **Vista:** Las vistas contienen el código que produce la visualización de las interfaces de usuario. La vista trabaja con los datos que le suministra el controlador, sin embargo, no se realiza un acceso directo a éstos. Suele decirse que la vista es una representación del estado del modelo en un momento concreto y en el contexto de una acción determinada.
- **Controlador:** En el controlador se encuentran los componentes capaces de procesar las interacciones del usuario. Las acciones e información procedentes del usuario serán recogidas exclusivamente por los controladores. El controlador es el responsable de seleccionar las vistas apropiadas en cada momento, luego tomará información procedente del modelo y la adaptará a formatos o estructuras de datos que la vista sea capaz de manejar.

Distinción entre la lógica de negocio y la lógica de aplicación.

- **lógica de negocio:** la lógica del negocio **se encuentra en los modelos**, es una serie de procesos que son la implementación de las reglas, acciones y restricciones que nos permiten gestionar las entidades del dominio.
- **lógica de aplicación:** la lógica de aplicación **se encuentra en los controladores**, es un conjunto de acciones que se realizan invocando métodos de los modelos y mandando datos a las vistas.

Diagrama del patrón MVC



En esta imagen podemos ver como funciona el patrón MVC en una serie de pasos:

1. El cliente (usuario) hace una petición al servidor.
2. El controlador procesa la petición que hace el usuario y solicita los datos al modelo.
3. El modelo se comunica con la base de datos y retorna la información.
4. El modelo retorna los datos al controlador.
5. El controlador retorna la vista al cliente con los datos solicitados.

Diagrama de colaboración del patrón MVC

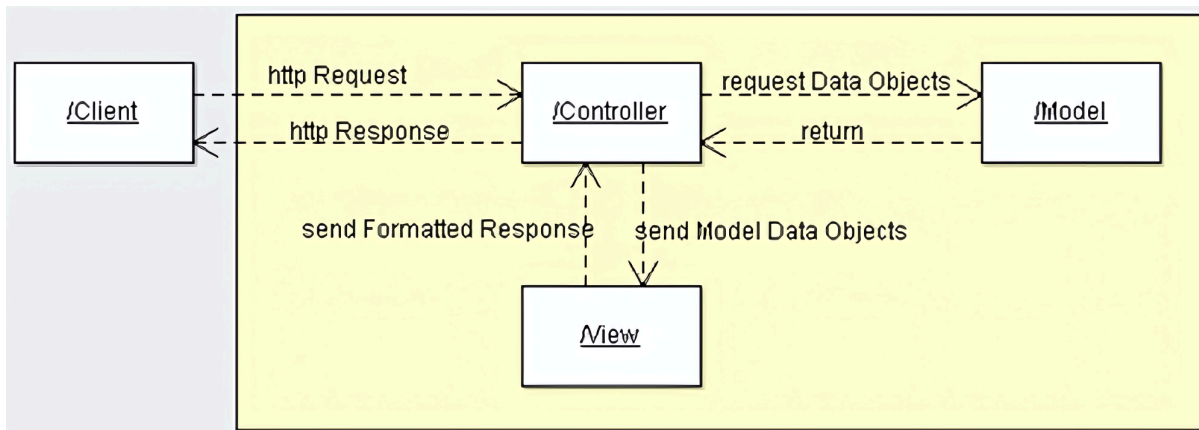
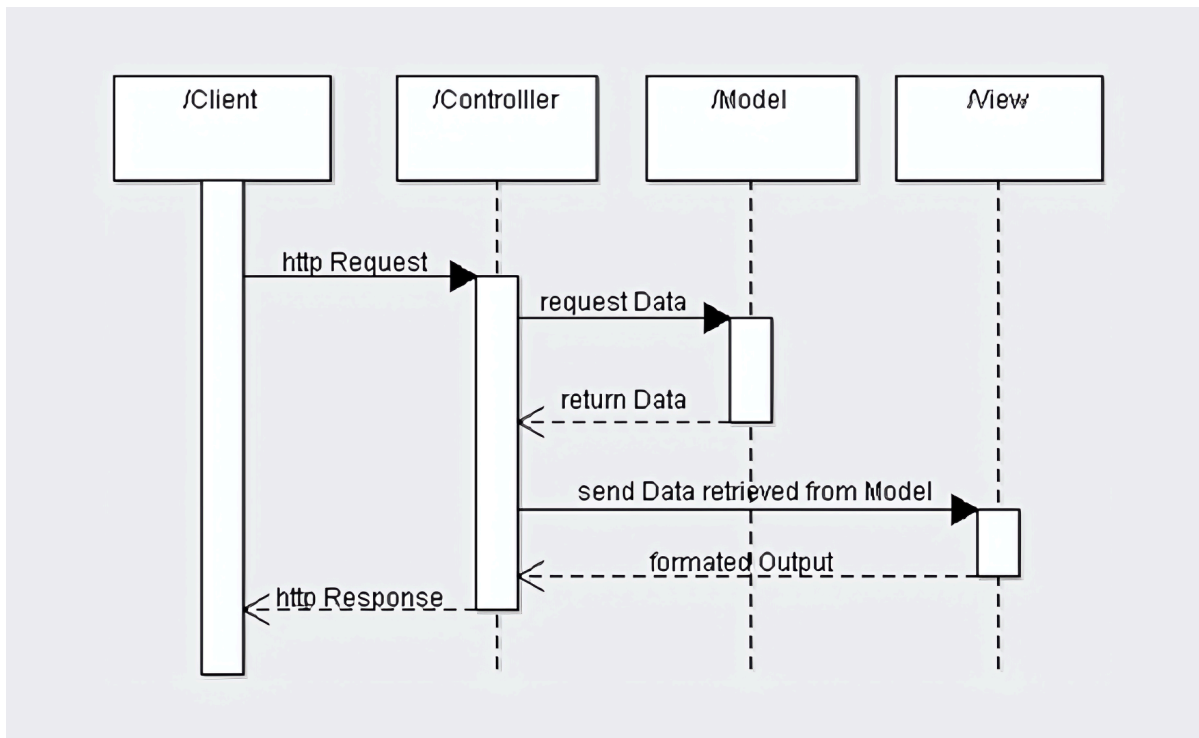


Diagrama de secuencia del patrón MVC



Control de Versiones

El control de versiones, también conocido como "control de código fuente", es la práctica de rastrear y gestionar los cambios en el código de software. Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo. A medida que los entornos de desarrollo se aceleran, los sistemas de control de versiones ayudan a los equipos de software a trabajar de forma más rápida e inteligente.

El software de control de versiones realiza un seguimiento de todas las modificaciones en el código en un tipo especial de base de datos. Si se comete un error, los desarrolladores pueden ir hacia atrás en el tiempo y comparar las versiones anteriores del código para ayudar a resolver el error, al tiempo que se minimizan las interrupciones para todos los miembros del equipo.

Herramientas de VCS

Los sistemas de control de versiones (VCS) han experimentado grandes mejoras en las últimas décadas y algunos son mejores que otros. Los VCS a veces se conocen como herramientas de SCM (Gestión del código fuente) o RCS (Sistema de control de revisiones). Una de las herramientas de VCS más populares hoy en día se llama Git. Git es un VCS distribuido, una categoría conocida como DVCS, que explicaremos con más detalle después. Al igual que muchos de los sistemas de VCS más populares disponibles hoy en día, Git es gratuito y de código abierto.

El repositorio de código

Su misión principal es la de sincronizar e integrar el código desarrollado por un equipo en un producto o componente software, pero también nos permite albergar toda la documentación relativa al desarrollo, o gestionar las distintas entregas y la promoción de código.

Información que provee un repositorio de código sobre la entrega del producto (time to market)

- Identificación de **equipos caóticos y/o desorganizados**, observando cómo se gestionan ramas y flujos de código.
- Equipos poco motivados suelen presentarse con información y **documentación obsoleta, comentarios poco explicativos, cambios muy frecuentes** donde se añade, se modifican y se deshacen las **mismas funcionalidades**.

- También es posible detectar problemas en la definición funcional, donde Product Owners y Usuarios ofrecen poca visión del producto y de las historias de usuario. En estos escenarios se propician los habituales **abandonos de ramas** que nunca se han integrado en las distintas entregas, o historias de usuario poco o mal definidas que implican numerosos cambios en momentos de aceptación.
- Respecto a la **calidad del producto**, en cualquier repositorio donde el número de ramas hotfix es elevado, nos indica que hay que revisar y analizar los requisitos de calidad, en cuanto a cobertura de pruebas, análisis estáticos y la automatización de pruebas funcionales.

El comentario de un commit

Un comentario aporta la información para entender el por qué (la motivación del commit) y para qué se ha realizado ese cambio.

BIBLIOGRAFÍA

Que es el modelo vista controlador (MVC) y como funciona | Educación para todos
<https://articulosvirtuales.com/articles/educacion/que-es-el-modelo-vista-controlador-mvc-y-como-funciona>

Model View Controller(MVC) in PHP | phponweb
<https://phponweb.wordpress.com/2011/04/28/model-view-controllermvc-in-php/>

Qué es MVC
<https://desarrolloweb.com/articulos/que-es-mvc.html>

¿Qué es el patrón MVC en programación y por qué es útil? | campusMVP.es
<https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>

Git - Como gestionar y cuidar nuestro código
<https://www.enmilocalfunciona.io/git-como-gestionar-y-cuidar-nuestro-codigo/>