

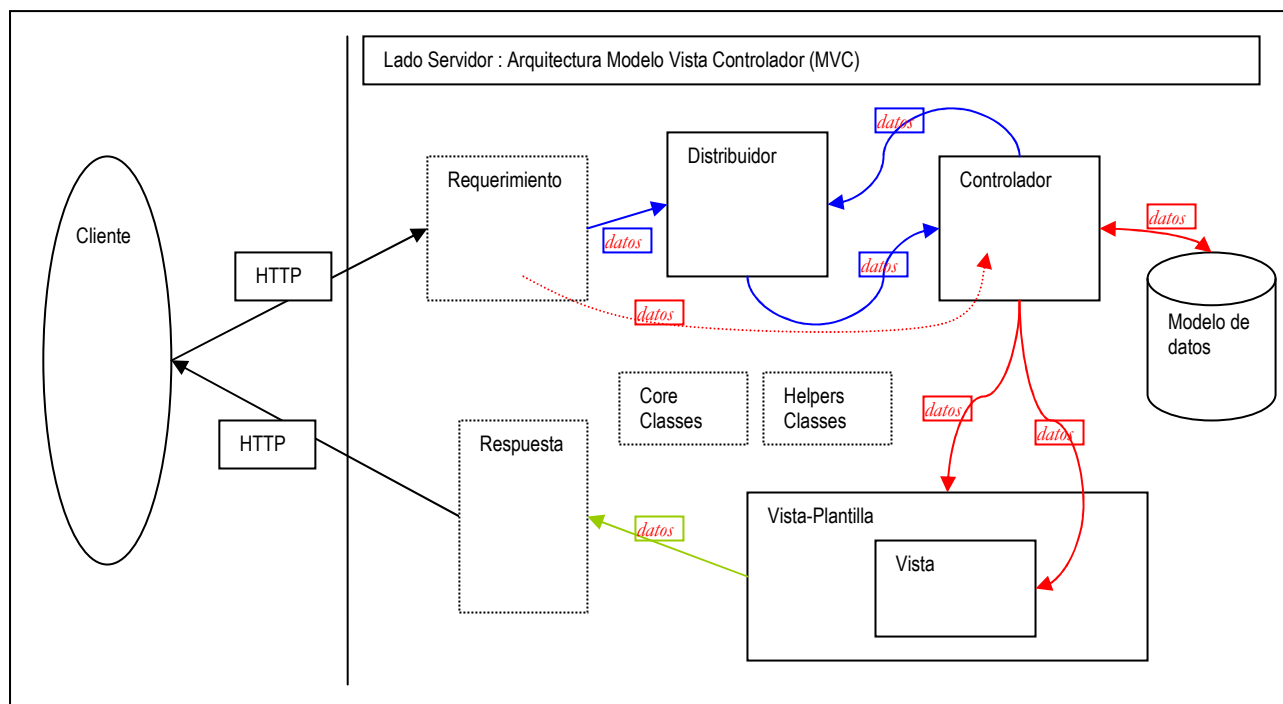
Modelo Vista Controlador

1.	Organización del software de una aplicación Web.....	2
2.	Referencias.....	3

1. Organización del software de una aplicación Web

El modelo MVC es un patrón muy utilizado para la organización del software que conforma una aplicación Web. Hay muchas interpretaciones del mismo. La que plantea fuelphp.com es bastante ortodoxa.

La adaptación del MVC que utilizamos en la aplicación **esmvcphp** podemos representarla gráficamente de la siguiente manera:



Una interpretación del modelo podría ser la siguiente:

El primer bloque (**Requerimiento**) recoge la petición del cliente y la adapta a un estilo de presentación.

El segundo bloque (**distribuidor=dispatcher=enrutador=router**) estudia la petición del cliente dentro del contexto o estado de la aplicación para decidir a que bloque de control (**controlador**) se la envía para que haga el tratamiento.

El tercer bloque (**controlador=controller**) atiende a la respuesta siguiendo las reglas de negocio, y si es necesario inyecta o recupera datos en la base de datos (sistema de permanencia) a través del bloque de modelo (**modelo de datos**). La flecha discontinua que une el elemento recoge el requerimiento con el controlador representa la situación en que una vez establecido el controlador que debe resolver la petición, este controlador recogerá el resto de datos necesarios para su funcionamiento desde el elemento de requerimiento, datos que procesará para llevarlos, si es necesario, al elemento de *modelo de datos* y hacia las *vista*.

El cuarto bloque (**modelo de datos=model**) almacena los datos recibidos desde el bloque controlador en bases de datos o recupera datos de bases de datos, y cuando tiene todos los datos preparados se los envía al bloque (**controlador**).

El controlador prepara la respuesta envía los datos a la vista (**respuesta=view**) que será la encargada de preparar los datos para transmitirlos al cliente dentro de un lenguaje de marcas: html, xml, pdf, ajax, ...).

Dentro de las vistas, hay que hacer una llamada sobre el concepto de plantilla o template en inglés. Si se trabaja con plantillas, la plantilla es como una vista principal o el documento final (sea en html, xml, pdf, etc) perfectamente conformado, con “huecos” en los que a través de php se introducirán datos o el contenido de alguna vista.

Además de los elementos principales asociados al modelo (controlador, modelo y vista) existen otros elementos que también son importantes pues en su mayoría son usados como soporte para los anteriores y para el funcionamiento del modelo. Estos elementos aparecen entre puntos discontinuos y son el *core o soporte para el sistema* y los *helpers que ayudarán al resto de módulos*. Por ejemplo, el elemento respuesta se encarga de enviar el paquete HTTP, conformado con su cabecera (*header*) y su cuerpo (*body*).

Del conjunto de datos recibidos en el elemento de requerimiento se pueden distinguir dos tipos, los datos para decidir cómo controlar o atender la petición (color azul) y los datos aportados para el tratamiento por el controlador (color rojo). Los datos en color verde representan los datos procesados más la información adicional para su presentación y tratamiento en el cliente.

Se recomienda hacer una lectura rápida de las referencias siguientes para hacer una aproximación a las distintas implementaciones del modelo.

2. Referencias

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

<http://www.neleste.com/modelo-vista-controlador/>

<http://book.cakephp.org/1.3/es/view/890/Entendiendo-Modelo-Vista-Controlador>

<http://fuelphp.com/docs/general/mvc.html>

<http://sphpf.coldstarstudios.com/> Creado por Marcos Sigueros, alumno de DAI que finalizó los estudios en el Instituto en el curso 2011/12 (no está actualizado, es más lo tiene abandonado según conversación del 16/11/2012)

http://codeigniter.com/user_guide/overview/mvc.html

<http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>

<http://www.yiiframework.com/doc/guide/1.1/en/basics.best-practices>