

Curso Symphony 4 - Clase 1

MVC: En líneas generales, MVC es un diseño de arquitectura de software o también llamado patrón de diseño de software, utilizado para separar el código por sus distintas responsabilidades funcionales. Estas responsabilidades se ven separadas por capas que se encargan de hacer un tarea muy concreta, lo que ofrece principalmente beneficios como: facilidad de mantenimiento, reutilización de código y la separación de conceptos.

Su base es la separación del código en tres capas diferentes, limitadas por su responsabilidad, las capas son: Modelos, Vistas y Controladores, también conocido como Model, Views & Controllers.

Modelos: En esta capa se trabaja con los datos, por lo tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos de cualquier aplicación o sistema de información se encontrará habitualmente en la base de datos, por lo que en esta capa tendremos todas las consultas para acceder a la misma.

En dicha capa se suele trabajar con librerías como PDO, en el caso de php, o algún ORM como Doctrine, estas librerías nos permiten trabajar con abstracción de base de datos y persistencia en objetos. Se utiliza un dialecto de acceso a los datos basados en clases y objetos.

Vistas: Esta capa, como su nombre lo indica, contiene el código que va a reproducir la visualización de las interfaces de usuario, habitualmente el código de la aplicación en HTML.

En la vista se trabaja con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas solicitaron los datos a los modelos y luego ellas generarán la salida, tal como nuestra aplicación lo requiera.

Controladores: Esta capa contiene el código que responde las acciones que se solicitan en la aplicación, como mostrar un elemento, realizar una compra, búsquedas de información, etc.

Es la capa que nos sirve de enlace entre las vistas y los modelos, además solicita las “necesidades” que puedan requerirse para implementar las necesidades de nuestra aplicación. Su responsabilidad servir de enlace entre los modelos y las vistas aplicando la lógica de negocio que la aplicación necesite.

Symfony y configuración del proyecto:

Symfony es un entorno de trabajo estandarizado (framework PHP) que se utiliza para el desarrollo de aplicaciones web.

Una de las principales ventajas de Symfony es que posee una licencia MIT, o lo que es lo mismo que una licencia de software libre permisivo, que se puede utilizar dentro del software del propietario y al no poseer Copyright permite su modificación.

Para poder trabajar con esta herramienta es necesario disponer de un servidor web, una consola de comandos del sistema operativo y cualquier versión de PHP5 o PHP7 si se quiere utilizar las últimas versiones.

Archivos de configuración:

Las aplicaciones Symfony se configuran con los archivos almacenados en el config/ directorio, que tiene esta estructura predeterminada:



El routes.yaml archivo define la configuración de enrutamiento ; el services.yaml archivo configura los servicios del contenedor de servicios ; el bundles.php archivo habilita / deshabilita paquetes en su aplicación.

Trabjará principalmente en el "config/packages/directorio". Este directorio almacena la configuración de cada paquete instalado en su aplicación. Los paquetes (también llamados "complementos / módulos" en otros proyectos) agregan funciones listas para usar a sus proyectos.

Se aconseja leer la siguiente documentación:

<https://symfony.com/doc/4.4/configuration.html>

Dependencias para nuestro proyecto en Symfony

Como ya sabemos symfony es un framework de PHP, por lo tanto necesitaremos un cliente PHP para levantar nuestro servidor y poder correr nuestra aplicación en symfony.

1 - XAMPP es un entorno el cual contiene varios módulos funcionales para el desarrollo de aplicaciones en PHP y otros lenguajes, trabajaremos con XAMPP.

<https://www.apachefriends.org/es/download.html> (Link de descarga)

(Descargar la versión 7.2.34 / PHP 7.2.34)



2 - Composer es un manejador de dependencias de PHP. Es vital tener este módulo en nuestra pc o ambiente de trabajo.

<https://getcomposer.org/> (Link de descarga)

3- Necesitaremos, en caso de que trabajen en Windows, una consola de desarrollo.

Cmder, entre otras, es una gran elección.

<https://cmder.net/> (*Link de descarga*)

4- Además necesitamos tener instalado GIT en nuestra pc.

<https://git-scm.com/> (*Link de descarga*)

5- Algún editor de texto que prefieran. En mi caso recomiendo el Visual Studio Code.

<https://code.visualstudio.com/> (*Link de descarga*)