

GUIÓN DE LA ACTIVIDAD PRACTICA 2.2 (AP2.2):

Título

Modelo Vista Controlador Routing con Composer y Patrón Singleton.

Descripción

Esta actividad práctica consiste en cumplir los mismos requisitos que la actividad AP2 pero añadiendo el autoloader por PSR-4 y usar el manager de dependencias para PHP Composer. Además, usaremos un patrón Singleton para conectarnos con la BB.DD. evitando de esa forma utilizar más de una instancia para conectarnos con la BB.DD.

Programa

1. Tema 1 y 2.
2. La actividad tiene una temporalización aproximada de 2 horas tras la revisión de toda la documentación y la realización de la AP2.

Recursos generales

Presentaciones y videos de Temas 1 y 2:

- Introducción a MVC
- Espacios de nombres PHP.
- Estructura típica de un proyecto MVC (composer)
- Singleton (Tema 1)

Material de apoyo:

- POO y MVC en PHP
- Y de más links del tema en Florida Oberta.

Actividades

Para ello se pide:

1. Crea un proyecto con la estructura típica que se ha visto en el tema 2 de MVC.
2. Instala el autoloader de Composer (el gestor de dependencias que vamos a usar todo el curso).
3. La conexión a la BB.DD. debe realizarse siguiendo el patrón Singleton.
4. La estructura de carpetas estará compuesta por:
 - **Public** → Donde vamos a alojar el frontController, que llamaremos index.php, del proyecto como único punto de entrada al mismo. Además, en esta carpeta podemos depositar todos archivos como imágenes, css, etc... auxiliares de nuestro proyecto.
 - **Src** → Contendrá la lógica de todo el proyecto mediante las siguientes subcarpetas:
 - **Controllers** → Estará contenida toda la lógica de los controladores que sea necesaria.
 - **Models** → Se adjunta un fichero SQL para exportar la BB.DD., todolist.sql. Con la que poder importarla a tu MySQL y generar las clases para obtener los datos necesarios. Toda la lógica del modelado de la BB.DD. debe estar en esta carpeta.
 - **Views** → No es necesario crear ningún motor de renderización. Genera clases capaces de mostrar por pantalla la información que se requiere, con una estructura de tabla será suficiente.
 - **Core** → Se puede añadir esta carpeta extra, si necesitas crear clases adicionales que no cumplan los criterios de las anteriores y que sean de utilidad.
 - **Config** → Obtén la información de conexión de BB.DD. desde un fichero JSON. Incorpora en el mismo (o en un fichero separado) el enrutado de la aplicación. Es decir, que rutas se corresponden con qué controladores. Crea las clases necesarias para la gestión de esta información.
5. La aplicación solo contendrá dos rutas:
 - 5.1. **/ o raíz** → Es donde se listará el total de los registros de la BB.DD., incluyendo título y fecha de vencimiento.

- 5.2. **/detalle** → Es donde se mostrará la totalidad de la información de una tarea.
6. **Recomendable** → Re-factoriza, si lo crees necesario, para cumplir los máximos principios SOLID que puedas.

Recursos adicionales

- Script creación BB.DD. **todolist** en formato SQL.

Entregar:

- No hay que entregar ningún archivo.