

AEV1 – Modelo Vista Controlador

Objetivos

- Demostrar que las bases de PHP y POO están ya consolidadas.
- Demostrar el conocimiento adquirido durante el tema 3 sobre el MVC.

Recursos generales

- Todas las presentaciones vistas hasta ahora.
- Documentación oficial de PHP.
- Códigos de ejemplo vistos en clase y en actividades previas.
- **No se puede utilizar ninguna IA, si se detecta que se ha utilizado para generar el código se marcará la evaluable como insuficiente.**

Actividad

1. Se debe clonar el repositorio de la invitación de Github Classroom y seguir los pasos del README. Se partirá del código del repositorio y habrá que realizar unas modificaciones para añadir funcionalidad nueva y para modificar parte de la funcionalidad existente.
2. Se debe implementar las especificaciones de autoloading de PSR-4 y la instalación de las dependencias se hará mediante Composer.
3. Se deben realizar como mínimo 5 subidas al repositorio de GIT demostrando un progreso adecuado de la aplicación y en esas subidas, debe añadirse un comentario con la descripción de las modificaciones realizadas.

Tareas a realizar

Ejercicio 1.

1. Entrar en la invitación de Github Classroom, seleccionar tu nombre y clonar el repositorio en un nuevo proyecto de PHPStorm.
2. Seguir el README de la carpeta raíz para poner en marcha el entorno Docker.
3. Inicializar composer en el proyecto:
 - Instalar el fichero `composer.json`.
 - Utilizar el namespace propuesto y usarlo en el código.
4. Hacer funcionar el código desde el navegador introduciendo la URL: `http://localhost:8010`
 - Realizar los cambios básicos necesarios para que funcione y documentarlos al final de este README.
 - Pistas:
 - Configuración BD
 - Namespaces
 - Enrutamiento
 - Enlaces en las vistas

Ejercicio 2.

1. Las siguientes rutas deben estar implementadas:
 - `/`
 - En esta vista mostraremos la plantilla de base o inicio con como mínimo un mensaje de bienvenida.
 - Se debe añadir unos botones o links que permitan acceder a la ruta de tareas, crear nuevas tareas y a la ruta de detalle sin ID y nos devolverá a la pantalla de error.
 - `/tareas` (lista de tareas)
 - Cargará una lista dentro de una tabla con todas las tareas que están en la BB.DD.
 - Se mostrarán los datos relevantes de cada tarea (título, descripción, fechas).
 - Tendremos un botón o link para volver al inicio o home.
 - Pinchando sobre el título nos llevará a mostrar los datos de la tarea con la id.
 - Añadir un link para cada tarea de Ver detalle que el link se construirá a partir de todas las letras del título de la tarea tipo `/detalle/nombre/mas/de/una/palabra/en/el/nombre`
 - `/crear` (formulario para añadir nuevas tareas)
 - Mostrará un formulario para añadir nuevas tareas.
 - Tendremos un botón o link para volver al inicio o home.

- **/detalle/id**

- Se mostrará una composición con los datos de la tarea, en este caso puede ser una tabla.
- Tendremos un botón o link para volver al inicio o home.
- Tendremos un botón o link para volver a la pantalla de selección de tareas (atrás).

- **/detalle/nombre/mas/de/una/palabra/en/el/nombre**

- Será lo mismo que en el punto anterior, pero se ha de recibir los datos a partir de los parámetros recibidos en la ruta y que están separados por la /
- En este caso esos datos corresponden únicamente al nombre de la tarea, por lo que habrá que pasarlos todos para que se procesen.
- Si el nombre de la tarea posee más de una palabra, cada palabra se pasará como un parámetro:
/detalle/palabra1/palabra2/palabra3/palabra4

2. Cualquier otra ruta recibida deberá procesarse:

- Cualquier ruta recibida debe procesarse y mostrar una página de error o NORUTA que indique que la ruta no está disponible y permitirá volver a la pantalla inicial.
- Crea una vista 404.html para gestionar los errores de rutas no definidas.

Ejercicio 3.

- Modificar la vista de la tabla de tareas para que las filas vencidas aparezcan con fondo **rojo** (cuando la fecha de vencimiento ya ha pasado).
- Recuerda que el procesado de la información hay que hacerlo en el **modelo**.

Ejercicio 4.

- Añadir la funcionalidad **Eliminar tareas** que permita borrar varias tareas a la vez:
 - Se podrán seleccionar varias de las tareas en la vista de lista de tareas.
 - Se podrá indicar que se quiere borrar todas a la vez.
 - Eliminar todas en BD.
 - Mostrar la lista resultante.

Entregables

1. Actualizar el README.md con:
 - Explicación de los cambios realizados en cada apartado.
 - Problemas encontrados y cómo se resolvieron.
 - Autorúbrica cumplimentada.
 2. Un .zip de todo el proyecto sin la carpeta /vendor (debe coincidir con el repositorio y funcionar).
 - Se pueden entregar dos versiones: una funcional y otra con el desarrollo a medias.
 3. Es posible que se pida a los alumnos que expliquen al profesor cómo han resuelto alguno de los apartados de forma verbal.
-

Rúbrica de evaluación

Existen cuatro niveles en los que se cataloga el resultado de la actividad:

1. Insuficiente (<5)

- El código entregado no funciona.
- No se ha implementado de forma completa ninguna de las modificaciones pedidas.
- Los ficheros añadidos no siguen el estándar PSR-4.

2. Suficiente (5 - 5,9)

- Se ha hecho funcionar el código entregado.
- Se ha creado la vista inicial con los links de la pantalla inicial y funcionan adecuadamente.
- Se ha implementado el gestor de dependencias Composer.
- El código está comentado.
- Se ha realizado el Ejercicio 2 a falta de la ruta dinámica de tareas.

3. Notable (6 - 8,9)

- Se cumplen los puntos del Suficiente y además...
- Se ha realizado el Ejercicio 3.
- Se han usado controladores nuevos si corresponde en todas las modificaciones.
- El código incluye siempre control de errores (por ejemplo, al usar las variables globales de PHP \$_GET, \$_SERVER, etc.).
- El código está preparado para PHPDoc.

4. Excelente (9 - 10)

- Se cumplen los puntos del Notable y además...
- Se ha realizado el Ejercicio 4.

- Se ha conseguido el link dinámico en la vista de tareas.
 - Se han implementado queries a BBDD con seguridad y control de errores.
 - La base de datos se ha implementado con una clase singleton.
-

Respuestas y explicaciones de las tareas

Completa el README del proyecto.

Vídeo del funcionamiento (opcional esta primera vez)

Grábate utilizando la herramienta de ScreenPal y pega el link en el README.

Autorúbrica (rellenar)

Rellenar en el README.