

GUIÓN DE LA ACTIVIDAD DE EVALUACIÓN 1 (AEV1):

Título

Crear un Modelo Vista Controlador.

Descripción

Esta actividad evaluable consiste en crear una aplicación que cumpla los criterios vistos en el bloque 1. Debemos crear un proyecto que cumpla con el patrón MVC, que tenga un sistema de enrutamiento mediante el autoloader de composer y los namespaces. También usaremos TWIG para implementar las plantillas visuales.

Programa

1. Todo el bloque 1
2. La actividad tiene una temporalización de unas 8 horas aproximadamente.

Recursos generales

Presentaciones y videos de los temas 2 y 3:

- Introducción a MVC
- Introducción a TWIG

Material de apoyo:

- The Twig book.
- Código demo – Templating con TWIG e interfaces
- Y de más links del tema en Florida Oberta.

Actividades

Para ello se pide:

1. Buscar una plantilla gratuita de HTML en internet con la que debemos implementar TWIG y la herencia de este. Podéis usar las plantillas, por ejemplo, que están en la siguiente web gratuita <https://all-free-download.com/free-website-templates/> o en esta otra <https://templatemo.com/> lo único que no deben ser spa.
2. Crea un proyecto con la estructura típica que se ha visto en el tema 1 y 2 de MVC y TWIG.
3. Instala Composer (el gestor de dependencias que vamos a usar todo el curso) e implantaremos las especificaciones de autoloader [PSR-4](#).
4. Es importante que durante el desarrollo del proyecto se use un repositorio GIT, en este caso usaremos GitHub donde se os asignará a cada uno un repositorio donde deberéis hacer la subida de toda la actividad evaluable.
5. La estructura de carpetas estará compuesta por:
 - **Config** → Obtén la información de conexión de BB.DD. desde un fichero JSON que se conectará a la BB.DD. de MySQL que se entrega con la documentación de la actividad. (empresa.sql). Además, se incorporará otro fichero JSON con el enrutado de la aplicación. Es decir, que rutas se corresponden con qué controladores. Crea las clases necesarias para la gestión de esta información.
 - **Public** → Donde deberá estar toda la lógica del HTML publica, CSS, JS, etc... de la plantilla descargada. El archivo index.php deberá contener la información necesaria para acceder al autoloader.
 - **Src** → Contendrá la lógica de todo el proyecto mediante las siguientes subcarpetas:
 - **Controllers** → Estará contenida toda la lógica de los controladores que sea necesaria para hacer funcionar la aplicación con todas las rutas permitidas.
 - **Models** → Se adjunta un fichero SQL para exportar la BB.DD., empresa.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. Importante recordar que cada tabla deberá estar contenida en una clase diferente.

- **Core** → En esta carpeta incluiremos todas las clases necesarias para el autoloading, interfaces y demás clases que harán funcionar el MVC.
 - **Interfaces** → Deberá contener todos los interfaces que sean necesarios para definir los métodos mínimos que garanticen que la aplicación pueda ser escalada adecuadamente como un MVC.
 - **Templates** → Donde tendremos contenidas todas las plantillas TWIG de nuestra aplicación. A partir de la plantilla HTML descargada deberá extenderse las diferentes plantillas TWIG para que contenga la información requerida en las diferentes páginas del ejercicio.
6. La aplicación solo contendrá como máxima las siguientes rutas:
 - 6.1. Usaremos siempre TWIG para toda la lógica de la vista.
 - 6.2. Todas las vistas han de tener un link o botón, según sea más adecuado por la plantilla descargada, para poder volver a la ruta anterior.
 - 6.3. **/ o raíz** → En esta vista mostraremos la pantalla de inicio como mínimo una imagen, que deberá cambiar cada vez que recargemos la página. Deberá haber como mínimo 3 imágenes diferentes para alternarse y deberán estar contenidas con la plantilla HTML descargada previamente. Además, añadiremos según la estructura de la plantilla unos botones, links u otra acción que nos permita ir a las otras rutas que tenemos que se piden en el ejercicio.
 - 6.4. **/listaProductos** → Cargará la lista de todos los productos. Modifica la plantilla usada en la ruta raíz y extiende de ella de una forma adecuada. La lista de los productos estará contenida en una tabla que deberá quedar bien con la plantilla que hayas descargado.
 - 6.5. **/listaClientes** → Cargará la lista de todos los clientes.
 - 6.6. **/detalleCliente** → Cargará los datos de un cliente. Para ello, deberemos pasar el id del cliente como parámetro de ruta. Los datos del cliente deberán mostrarse en una tabla con una estructura diferente a la tabla de lista.
 - 6.7. **/departamentos** → Cargará una tabla con todos los departamentos y desde cada departamento podremos ir a la ruta **/empleados** que pertenezcan a ese departamento.
 - 6.8. **/empleados** → Cargará una tabla con todos los empleados de la empresa si no se ha recibido parámetro correspondiente al departamento. Si recibe el id del departamento mostrará únicamente los empleados de ese departamento.
 - 6.9. **/pedidos** → Mostrará un desplegable con el nombre de todos los clientes y una vez seleccionado el cliente correspondiente podremos pulsar un botón que nos debe mostrar en la misma página todos los pedidos de ese cliente. Pudiendo volver a seleccionar otro cliente en el desplegable. En cada pedido tendremos un link sobre el número de pedido que nos permita ir a otra plantilla **/detallePedido** con los detalles del pedido.
 - 6.10. **Cualquier otra ruta** deberá mostrar una pantalla de error muy similar a la **/** donde además de contener todo lo que tiene que mostrar esa ruta debe mostrar un mensaje de ruta no disponible.
 7. **Formato TWIG** → Hay que formatear:
 - 7.1. Las fechas al formato dd/mm/aaaa usando **los filtros TWIG**.
 - 7.2. Los textos deben estar todos en minúsculas, salvo la primera letra de títulos y nombres que estará en mayúsculas, para hacer esto debemos **usar filtros TWIG, no lógica de PHP**.
 - 7.3. Todos los valores económicos de dinero deberán estar filtrados con **el símbolo de € y deberán mostrar siempre dos decimales**.
 8. Realizar memoria con el formato que está en Florida Oberta, rellenando todos los puntos que indica.
 9. Realizar video explicando cómo funciona la aplicación desarrollada de entre 5 y 15 minutos. El video mostrará como se ejecuta la aplicación, debéis salir con la cámara activa en alguna esquina del video durante todo el video. Al igual que la explicación debéis darla vosotros, intentando evitar leer un documento.
 10. Entregar Autorubrica con vuestra propia evaluación, usando la rúbrica entregada.

Recursos adicionales

- Usar script creación BB.DD. [empresa.sql](#) en formato SQL que se ha entregado junto con los recursos de la actividad.
- Hoja de rubrica para rellenar la autorubrica y conocer los diferentes niveles de evaluación de la actividad.

Entregar:

- Fichero ZIP con la estructura completa del proyecto, incluyendo todos sus archivos. No es necesario incluir la carpeta vendor. **Este archivo a de ser exactamente igual a la versión subida en el repositorio de Git.**

- Incluir ruta con el link de OneDrive donde habéis subido el video y aseguráros que esta compartido conmigo y puedo acceder.
- Incluir archivo en formato Word o similar incluyendo la portada.

Rúbrica:

Existen cuatro niveles de corrección para esta actividad:

- Mínimos (hasta 5 puntos)
 - Crear estructura de carpetas según el patrón MVC correctamente como se ha visto en los últimos ejemplos.
 - Realizar memoria según documentación subida en Florida Oberta.
 - Realizar video explicativo de la actividad.
 - El sistema de autoloader debe cumplir el protocolo PSR-4 y estar implementado mediante el gestor de dependencias Composer.
 - La aplicación debe conectar con la BB.DD.
 - Que la ruta raíz, "/", se muestre al iniciar el proyecto desde la carpeta public, ejecutando los requisitos
- Suficiencia
 - Cumplir todos los requisitos del punto mínimo.
 - Añadir la implementación de las rutas: **"/listaProductos"**, **"/listaClientes"** y **"/detalleCliente"**.
- Notoriedad
 - Cumplir todos los requisitos del punto suficiencia.
 - Añadir la implementación de las rutas: **"/departamentos"** y **"/empleados"**.
 - Todas las plantillas implementadas hasta ese momento deben poder volver a la pantalla anterior.
- Excelencia
 - Cumplir todos los requisitos del punto notoriedad.
 - Añadir la implementación de las rutas: **"/pedidos"** y **"/detallePedido"**.

Tabla de corrección de rúbrica y Auto rúbrica

| Nivel de Rúbrica | Superado | No superado | Observaciones |
|------------------|----------|-------------|---------------|
| Mínimos | | | |
| Suficiencia | | | |
| Notoriedad | | | |
| Excelencia | | | |