Nombre	
Módulo	
Curso	
Fecha	

Práctica 2. PHP

Introducción.

En esta práctica correspondiente a la segunda parte del trimestre del módulo de Desarrollo Web en Entorno Servidor se trabajará en un proyecto web con el lenguaje PHP.

La idea principal es consolidar los conocimientos adquiridos en la primera parte del trimestre especialmente en la creación de un proyecto que contenga el patrón de diseño Modelo-Vista-Controlador (en adelante MVC).

Sin embargo y para compensar el mayor peso de este proyecto sobre la calificación del trimestre, el nivel de detalle aumenta, ampliando el número de requisitos del proyecto.

La temática se basa en la realización de una tienda online para la venta de productos y/o servicios. Para esta tienda no se requieren (como sí pasaba en el anterior proyecto) un número fijo de CRUDS o de tablas en la base de datos, pero se debe realizar su construcción con el mayor rigor posible.

Un elemento importante del proyecto es la implementación de un carrito en la tienda que almacene los productos y/o servicios seleccionados por el cliente. Una herramienta básica es la utilización de las variables **$_SESSION** para mantener dicho carrito activo para el cliente, aunque este pueda cerrar sesión, teniéndolo así disponible una vez vuelva a loguearse. Una vez finalizado el pedido o encargo de productos y/o servicios, se procede a la inserción en la base de datos dejando el carrito vacío.

Objetivos.

Los objetivos de esta práctica son los siguientes:

- Conocimiento del funcionamiento de PHP y su sintaxis.
- Realización de buenas prácticas en PHP.
- Reconocimiento e implementación correcta del patrón de diseño MVC.
- Reconocimiento e implementación correcta de los modelos.
- Reconocimiento e implementación correcta de los controladores.
- Reconocimiento e implementación correcta de las vistas.
- Reconocimiento e implementación correcta de motores de plantillas.
- Reconocimiento e implementación correcta de rutas.
- Reconocimiento e implementación correcta de las variables de sesión.
- Reconocimiento e implementación correcta de distintos roles de usuarios.
- Reconocimiento e implementación correcta de las cookies en PHP.

Requisitos.

Requisitos del proyecto:

1. El proyecto debe contener estilos **CSS** en todos los ficheros (Bootstrap y/o estilos propios).
2. El proyecto debe tener una página web de caída (*index.twig*).
3. El proyecto debe estar correctamente ordenado con una adecuada estructura de carpetas y ficheros.
4. El proyecto debe contener un fichero ***.htaccess*** que contenga la implementación de rutas y las direcciones de las páginas de errores.
5. El proyecto debe trabajar con el motor de plantillas ***Twig***.
6. El proyecto debe contener distintos *tags*, *filters* y *functions* en las vistas de *Twig*.
7. El proyecto debe tener las vistas en sus correspondientes carpetas por tipo y funcionalidad: *auth*, *users*, *pedidos*, *carrito*, *productos*, etc.

   Se deben tener en su interior las vistas con los nombres adecuados y vistos en clase siempre y cuando estén implementados:

   a. ***index.twig***, para mostrar todos los elementos principales.
   b. ***create.twig***, para crear un elemento nuevo.
   c. ***edit.twig***, para actualizar los valores de un elemento.
   d. ***show.twig***, (opcional) para ver los detalles de un elemento.

8. El proyecto debe contener varios roles de usuarios, como mínimo los siguientes:
    a. **Administrador**: entra a la parte de administración de la web donde tiene permisos para realizar las funciones correspondientes a los CRUD (siempre y cuando tengan sentido).
    b. **Cliente**: entra a la parte de compra de productos y/o servicios (según temática elegida). Sus funciones son limitadas.
9. El proyecto debe tener al menos una plantilla para la parte pública y otra para la parte privada (recomendable una tercera para distinguir entre ambos usuarios y evitar errores).
10. El proyecto puede contener un módulo de registro de usuarios (se guardan con rol de cliente).
11. El proyecto debe implementar en los modelos la interfaz Model que contiene las siguientes funciones:
    a. ***findAll()***, función que devuelve todos los elementos (utilizado para el *index*)
    b. ***findById(),*** función que devuelve un elemento (utilizado para el *edit* y *show*)
    c. ***save(),*** función para insertar un elemento en la base de datos.
    d. ***update()***, función para actualizar datos sobre un elemento en la base de datos.
    e. ***delete()***, función para eliminar un elemento en la base de datos.
12. El proyecto debe contener modelos que tengan *constructor*, *getters* y *setters* obligatoriamente. Además, se pueden implementar métodos nuevos si son necesarios y tienen coherencia con el modelo en el que se desarrollan.
13. El proyecto debe tener un **controlador frontal**, concretamente el fichero *index.php* el cual contiene la lectura e interpretación de las rutas para realizar la llamada al controlador correspondiente.
14. El proyecto debe implementar en los controladores la interfaz Controller que contiene las siguientes funciones:
    a. ***index()***, función que muestra la vista de *index.twig* correspondiente.
    b. ***create()***, función que muestra la vista de *create.twig* correspondiente.
    c. ***edit()***, función que muestra la vista de *edit.twig* correspondiente.
    d. ***update()***, función que tiene como objetivo actualizar un registro en la base de datos.
    e. ***delete()***, función que tiene como objetivo eliminar un registro en la base de datos.
15. El proyecto debe contener una carpeta **config** que contenga los elementos de configuración de la página web. Se destacan dos ficheros vistos en clase:
    a. Fichero **Database.php**: contiene la configuración de la base de datos (opcional pero recomendable que su función conectar sea *static*).

b. Fichero **Parameters.php**: contiene constantes utilizadas durante el proyecto y accesibles desde cualquier fichero php.

16. El proyecto debe contener la carpeta **assets** con toda la aportación de imágenes, css propio, js propio, etc.

Entrega

La entrega del proyecto se realizará mediante los siguientes elementos:

1. Código del proyecto en formato .zip con el siguiente formato:
   "*Apellido1_Apellido2_Nombre.zip*". Ejemplo: "*Fontan_Llamas_Sergio.zip*".
   Nota: La penalización por no cumplir este apartado es de 1 punto.

2. Vídeo del contenido del proyecto donde muestra el cumplimiento o no de los requisitos del proyecto.
   Se recomienda realizar una explicación de su contenido para una mejor comprensión y calificación.
   El formato de entrega es libre siempre y cuando el profesor pueda visualizarlo (se recomienda .mp4).
   En cuanto al nombre del fichero deberá ser igual al del proyecto. Ejemplo:
   "*Fontan_Llamas_Sergio.mp4*".
   Nota: La penalización por no cumplir este apartado es de 3 puntos.

3. Enlace del proyecto alojado en un repositorio de GitHub, GitLab o similar.
   Nota: La penalización por no cumplir este apartado es de 1 punto.