TIENDA SOCIAL

PRACTICA / EXAMEN

PHP

TIENDA SOCIAL

Tienda social es una practica en php simulando una tienda online utilizando el modelo, vista y controlador, en la cual un usuario puede registrarse para poder acceder a la sección de productos, los cuales tienen dos botones, uno para poder ver todas las características de los productos y otro para añadirlo al carrito.

Ademas también posee un apartado para que el administrador pueda añadir, modificar o eliminar productos.

LISTA DE REQUISITOS:

Inicio / login :

Que haya una pagina iniciar sesión con los campos de nombre de usuario y contraseña, un botón para loguearse y otro botón para registrarse por si el usuario no tiene cuenta.

Registro:

La pagina de registro debe de tener los campos nombre de usuario, el cual debe ser unico, una contraseña, nombre, apellidos, correo electrónico, lugar de residencia y un selector para que el usuario escoja entre hombre y mujer.

Ademas tendrá un botón para poder validar el registro y otro botón para poder volver atrás por si el usuario le hubiera dado al botón de registro sin querer.

Pagina de productos:

En esta pagina deben mostrarse todos los productos de la tienda.
Cada producto debe mostrar una foto del mismo, el nombre, el precio y un campo donde poder introducir la cantidad deseada, ademas de dos botones, uno para ver el producto en otra pagina con su descripción correspondiente y otro botón para poder añadirlo al carrito.

Si se añade un producto al carrito, se mostrara el icono de un carrito con la cantidad de items añadidos. Si el producto que se añade no se ha seleccionado una cantidad determinada, por defecto añadirá una sola unidad de ese producto, si ese producto ya se ha

añadido al carrito, se sumara uno a la cantidad, si al producto se le pone una cantidad establecida, se sumara esa cantidad a la ya puesta en el carrito.

Cuando se pulse sobre el icono de carrito, este debe de mostrar todos los productos que se hayan introducido, si se vuele a pulsar sobre el icono se cerrara de nuevo.

En la lista de productos del carrito debe mostrarse.

- Nombre del producto
- Cantidad añadida
- Precio base del producto
- Precio total por cantidad (cantidad del producto X precio)
- Un botón para actualidad la cantidad de ese producto
- Un botón para eliminar el producto del carrito.

El carrito también debe de mostrar el importe total de la suma de todos los productos por la cantidad de cada producto.

Si se elimina o se actualiza un producto, el precio total también se actualizara.

En la parte inferior del carrito deben de haber dos botones

- **Finalizar compra**
  Cuando se da a este botón, el carrito de la compra se vaciara y pasara la compra a la base de datos.

- **Vaciar carrito**
  Cuando se hace click sobre este botón se vacía por completo el carrito

Pagina del producto:

Tiene que mostrar los campos básicos de un producto.

Como titulo tendrá el nombre del producto, debajo de él, la imagen del producto y a un lado de la imagen, estarán la descripción, el precio y la categoría a la que pertenece el producto.

Ademas, tendrá un botón para volver de nuevo a la tienda de productos.

Pagina productos administrador:

Esta pagina contara con un botón para añadir productos.
Debajo habrá una lista con todos los productos añadidos y ordenados por la identificación en modo descendente
En la lista de productos , cada producto contendrá los siguientes apartados.

- **Id del producto**
- **Nombre**
- **Precio**
- **Stock**
- **Imagen**

Ademas de los botones para realizar unas funciones

- **Botón eliminar**
  Elimina el producto de la lista de productos

- **Botón Modificar**
  Te llevara a la pagina de añadir producto con los campos rellenos del producto seleccionado.

- **Botón información**
  Te llevara a la pagina de información del producto tal y como la ve un usuario en la tienda, para previsualizar como lo vera el usuario.

También debe de haber un botón para cerrar la sesión y salir de la tienda.

Pagina Añadir Producto:

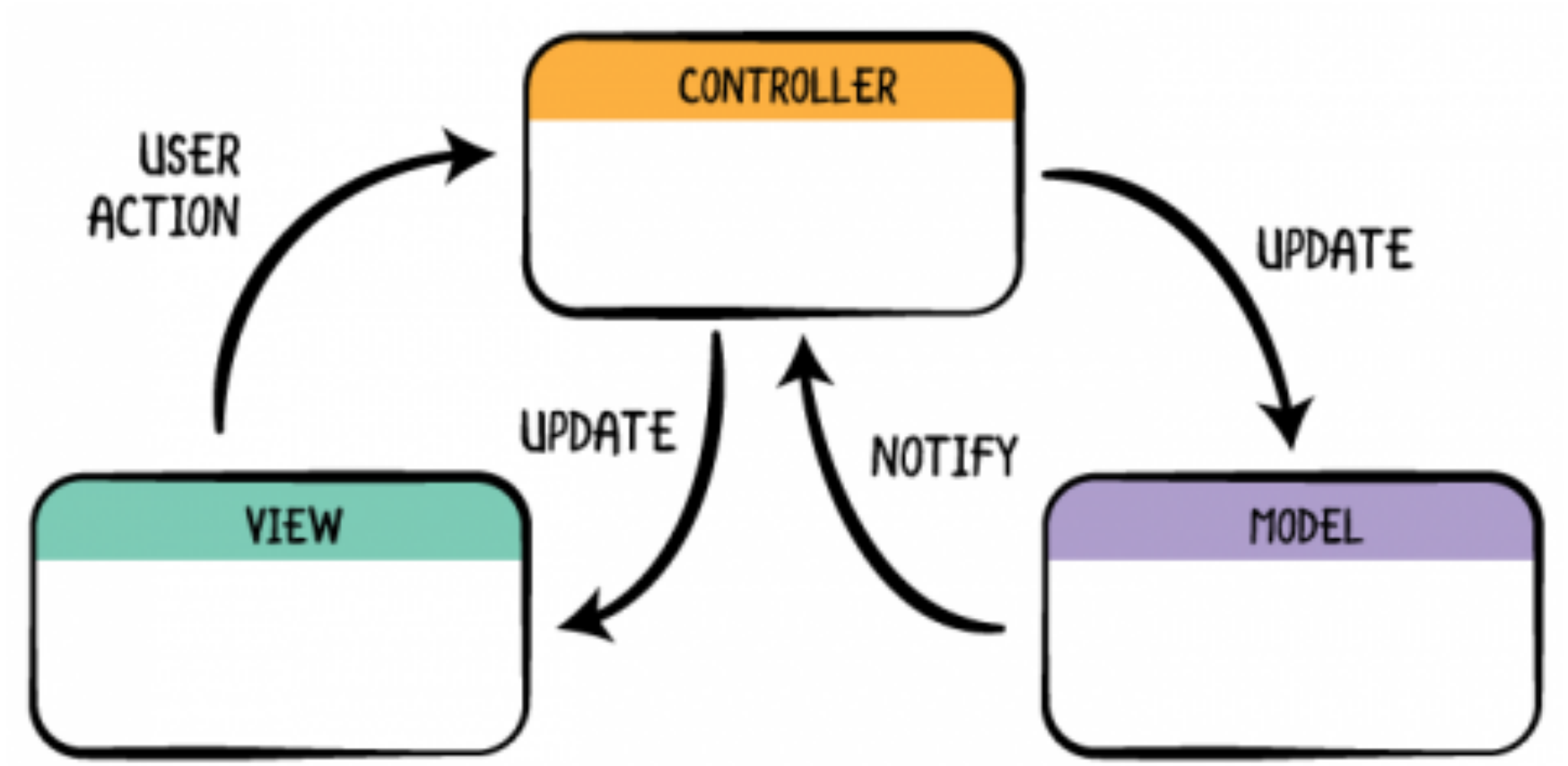
Sera un formulario con los campos de :

- **Nombre producto**
- **Descripción**
- **Stock**
- **Precio**
- **Categoría producto**
- **Opción para subir imagen del producto.**

Si no se sube ninguna imagen, pondrá una por defecto en la que indique no que hay imagen del producto.

También debe de haber un botón para guardar el producto y otro botón de volver, que te debe retornar a la pagina de list a de productos del administrador.

QUÈ ÉS EL PATRÓ MVC I COMFUNCIONA?



Es un sistema de archivos que nos permite tener organizados y de una manera estructurada toda la gestión de un proyecto.

El **MVC** se compone de tres elementos, el modelo, la vista y el controlador.

El modelo: es el encargado de gestionar todo lo relacionado con los datos del proyecto, este procesara los datos suministrador por el controlador para realizar las consultas, actualizaciones, borrar o añadir los datos en la base de datos o gestión de ficheros para devolver al controlador el proceso realizado.

ej. realizar una consulta en la base de datos sobre los productos de una tienda online.

El controlador: es el que se encarga de recoger, devolver y procesar la información que va desde la vista al modelo y viceversa. Este tratará la información y la manejara acorde a las necesidades de la vista.

ej. Recoger el nombre de usuario de un loggin para que lo envíe a la base de datos y comprobar que exista.

La vista: es cara visible de un proyecto, es donde el usuario puede interactuar con el manejo de la información.

ej. Tienda online donde el usuario puede registrarse y realizar compras

- **Tienda**

  - **index.php**
    Pagina de inicio que sirve de puente para ir de una pagina a otra por los controladores

  - **Controller (carpeta)**
    Carpeta que contiene cada uno de los controllers relacionados con las vistas

    - **Cart**
      Se encarga de gestionar el carrito de la compra cuando le daños a añadir un producto

    - **tienda.controller.php**

      - **Index()**
        Te lleva a la pagina principal de la tienda.

      - **Formproduct()**
        Formulario para la creación de un producto

      - **Eliminar()**
        elimina un producto de la tienda en base a la id del producto obtenida

      - **PaginaProducto()**
        te lleva a la vista descriptiva del producto recogiendo el id del producto

      - **Guardar()**
        Cuando se añade un producto a la tienda, este lo guarda en la base de datos

      - **Carrito()**
        Añade un producto al carrito recogiendo la id, el nombre, la cantidad y su precio a través de unas variables de sesión.

      - **actualizarProductoCarrito()**
        actualiza el carrito cuando se cambia la cantidad o se elimina del carrito

      - **borrarCarrito()**
        Limpia el carrito y lo deja completamente vacío

    - **loginregister.controller.php**

      - **Index()**
        te lleva a la pagina de registro de la tienda

      - **getReristerUser()**
        Recoge todos los datos del formulario pasados por post y los introduce en un objetos de tipo usuario donde después lo envía al modelo que lo inserta en la base de datos

- **common.controller.php**

  - **Index()**
    Te lleva a la pagina para poder loguearte

  - checkUser()
    Recoge los datos de usuario y contraseña pasados por post y los envía al modelos para comprobar que esos valores sean correctos o no con getUserPass
    Si los datos son correctos te le llevara a la tienda, sino de nuevo a la pagina de login

  - **CerrarSesion()**
    Destruye la variable de sesión y te lleva a la pagina de inicio

- **Model (carpeta)**
  En modelo

  - **products.php**
    Es una clase de producto el cual tiene toda una serie atributos de manera privada y de la cuales para poder hacer uso de ellas se hacen con sus Setters y getters correspondiente.
    También tiene los siguientes métodos para trabajar con la base de datos.
    Es necesario y obligatorio establecer una conexión con la base de datos con la clase de conexión.

    - **insertProduct($product)**
      Inserta en la base de datos un producto pasado como parámetro el objeto del mismo

    - **Mostrar()**
      Recoge todos los productos de la base de datos y los pasa a un objeto para que el controlador los maneje

    - **Obtener($id)**
      Devuelve como objeto un producto de la base de datos pasando por parámetro la id del mismo

    - **Eliminar($id)**
      Eliminar de la base de datos un producto pasando por parámetro la id del producto

    - **Actualizar($product)**
      actualiza en la base de datos un producto en concreto

- **conexión.php**
  contiene toda la información para conectar con la base de datos del servidor.

  - connectdb()
    Establece la conexión con la base de datos con el sistema de PDO, en la que se debe de poner el servidor, nombre de usuario, contraseña y el nombre de la base de datos con la que se quiera conectar.

- **users.php**

  - **getUserPass()**
    comprueba el usuario y la contraseña y los valida con la base de datos, si los datos son correctos valida la variable de sesión y redirecciona al usuario al inicio de la tienda.

  - **setterUserDates()**
    Inserta los valores pasados por parámetro con la base de datos cuando un usuario se loguea y todo ha sido correcto.

  - **insertRegisterUser()**
    Cuando un usuario se registra desde el controlador envia los datos al modelo y desde la misma función e inserta los datos en la base de datos.

- **Vista (carpeta)**

  - footer.php
  - formProduct.php
  - Header.php
  - login.php
  - paginaproducto.php
  - register.php
  - tienda.php
  - tienda2.php
  - **Css (carpeta)**
    - Reset.css
    - Style.css
  - **Js (carpeta)**
    - javascript.js
  - **Image (carpeta)**
    - …

PER QUÈ EL PROGRAMA REALITZAT COMPLEIX EL PATRÓ MVC?

Porque están separadas todas las secciones respetando el modelo vista controlador, en el que por un lado tenemos las diferentes vistas con las que interactuará el usuario, sus respectivos controladores que hacen la función de intermediarios y finalmente los modelos de sus respectivas vistas y controladores que manejan los valores desde la base de datos y las devuelve de nuevo al controlador para devolverlo a la vista.