Tienda. Objetivos

Índice

1. Introducción

En este proyecto se ha realizado una aplicación 'ObJEtivos', desde la definición de la estructura de datos hasta la aplicación cliente, pasando por la aplicación backend como requisito para la superación del curso Foap2023.

La realización de este proyecto también supone una oportunidad en mi caso para desarrollar una aplicación de uso propio para el inventario y venta de material de fotografía de segunda mano.

2. Descripción

En este Proyecto de final de curso se propone la implementación de una aplicación para una tienda.

Desde el punto de vista técnico, para la construcción de la base de datos se ha optado por el uso de MariaDB; para la concepción de la aplicación backEnd o de servidor se ha usado PHP y la aplicación frontEnd se ha construido usando el HTML, CSS, y JavaScript, aunque en algunos programas se generan las vistas HTML, en forma de tablas, que enviaba al cliente desde el servidor.

Para codificar esta aplicación se ha optado por un modelo Vista-Modelo-Controlador desarrollado en OOP, que permite separar mejor las diferentes responsabilidades de cada parte de la aplicación (la parte vista presenta la información mientras que la parte servidor gestiona la información y la procesa).

Se han usado las librerías Domppdf para generación de PDF y PHPmailer para envío de mensajes en tiempo real.

3. Objetivos y pasos a seguir

Los objetivos que se persiguen en este proyecto y que deben completarse necesariamente al final del proyecto son:

- Analizar e identificar las entidades y las necesidades que la aplicación debe resolver.
- Diseñar y realizar una aplicación backend con la lógica necesaria para responder al uso para los que esta aplicación está pensada.
- Diseñar y desplegar una aplicación frontend para consumir los servicios ofrecidos por la aplicación backend.
- Aplicar en lo posible los conocimientos de diseño de interfaces web adquiridos en este curso.
- Integración de la aplicación con servicios de mensajería de terceros para el envío de notificaciones en tiempo real. (Integración de la aplicación con APIs de terceros).
- Aplicación de pruebas unitarias y funcionales para asegurar el funcionamiento deseado de las partes de las aplicaciones.

El producto final de esta aplicación es un conjunto de ficheros HTML, Php, CSS y JavaScript que deben ser alojados en un servidor Apache.

4. Arquitectura de la aplicación

4.1. La base de datos

La base de datos tiene tres tablas para datos fijos de la aplicación, que son las tiendas, los vendedores y las tablas de códigos postales de los municipios. Una pequeña aplicación "webAdmin" se ha desarrollado para mantener estos datos por parte de un administrador de la aplicación web. A través de esta aplicación se mantiene también la tabla de productos, que almacena los datos de los productos a la venta y una pequeña tabla de categorías de productos.

El resto de las tablas es como sigue:

- La tabla de usuarios, que almacena los datos de los usuarios que se registran para el uso de la aplicación.
- La tabla de pedidos, que almacena los datos de cabecera de los pedidos realizados por los usuarios.
- La tabla de líneas de pedido, que almacena los detalles de los pedidos.
- La tabla de carrito, que almacena el carrito temporal creado para el usuario mientras realiza la compra.
- La tabla de líneas de carrito, que almacena el contenido del carrito.
- La tabla de existencias, que almacena los datos de stock de los productos.

Se incluye un diagrama completo de la base de datos con las tablas y sus atributos y las relaciones entre tablas.

4.2. Aplicación back-end

En lo concerniente al patrón de arquitectura seguido para construir la aplicación backend, se ha tomado como referencia el modelo de arquitectura Vista-Modelo-Controlador sobre Php.

Se ha seguido este patrón con algunas licencias para dar cabida a otras necesidades de la aplicación.

4.3. Aplicación front-end

La parte frontend de la aplicación se ha desarrollado en HTML, CSS y JavaScript. Por cuestiones de disponibilidad de tiempo se ha utilizado en algunas pantallas Bootstrap (librería OpenSource de componentes CSS y JS).

4.4. Despliegue

El despliegue se ha hecho por transferencia de archivos en red vía linea de comando y SPC (Secure Copy Protocol). La base de datos se ha creado en la aplicación PHPMyAdmin del servidor mediante un fichero ejecutable 'sql' .

5. Entorno de desarrollo

La aplicación se ha desarrollado en ordenadores personales de aula y propio, en entorno Windows 10 y sobre plataforma XAMPP, con servidor Apache local.

Para la edición de código se ha utilizado Visual Studio Code.

Para crear los modelos de BD se ha utilizado PHPMyAdmin y MYSQL Workbench 8.0.

El tiempo de desarrollo ha sido de cuatro semanas.

6. Diagrama de uso y relación de módulos y sus funciones

Con esta memoria se entrega un diagrama de la aplicación y sus usos.

*ver "Diagrama-proyecto-FOAP2023_ver2-.jpg "

7. Views. Recursos. Diseño de pantallas e includes

- Presentar pantalla en secciones: Header, barra de navegación.
- Uso de Carousel para visualización de grupos de productos y fotografías de cada producto.
- Uso de table para visualización de productos en pantalla de productos.
- Uso de table para visualización de pedidos en pantalla de pedidos.
- Uso de table para visualización de tiendas en pantalla de tiendas.
- Uso de iconos.

8. Clases

Se ha usado el modelo de extensión de clases desde la clase 'Connection' que es el modelo para acceso a la base de datos. Partiendo de esta clase como clase padre, se definen por extensión las clases que incluyen todos los métodos de acceso a datos para las entidades.

La clases ‘controlador’, gestionan las validaciones antes del acceso a datos y también la entrega de la respuesta.

No hay una correspondencia exacta en el uso de clases y tablas, quiero decir que por ejemplo el controlador de pedido suministra igualmente la cabecera de pedido (tabla ‘pedidos’)y las lineas de pedido (tabla ‘lineasPedido’).

Siempre que es posible, los datos de las entidades se guardan en los atributos de cada clase, de manera que, las llamadas entre funciones no precisan de argumentos.

Aunque pueda parecer que los métodos Getters y Setters de las clases puedan parecer redundantes , se usan para dar claridad al código.

9. Perfiles de usuario

Todos los usuarios que se registran en la aplicación y que pueden realizar las acciones habituales previstas son Usuarios sin privilegios. Los usuarios administradores trabajan sobre otra aplicación WebAdmin.

Todos los usuarios de la aplicación son propios, es decir que se ha registrado mediante el formulario de registro de usuarios en la aplicación.

10. Requisitos de instalación/implantación/uso

Como se ha comentado en apartados anteriores, el despliegue es una parte importante del desarrollo de la aplicación, puesto que se pretende crear una aplicación que sea fácilmente desplegable y escalable en caso de ser necesario.

Para ello para nombrar a los ficheros se ha seguido una rigurosa nomenclatura de ficheros y tables (uso de mayúsculas y minúsculas así cómo caracteres especiales) para cada tipo de fichero, de modo que no se produzcan errores si se instala sobre un sistema operativo diferente al entorno de desarrollo. También se ubican los ficheros en carpetas diferentes según su función.

Todas las direcciones de los ficheros (y las llamadas entre ello) son relativas al directorio raíz de la instalación. Mediante un fichero de configuración , y previo reconocimiento del sistema operativo (que se detecta via el comando OS.php) se configuran las rutas, así como los nombres y las palabras de paso en las bases de datos del sistema operativo donde se hospeda la aplicación.

11. Usabilidad y accesibilidad

La aplicación se ha diseñado para ser fácil de usar y comprender. Aunque la aplicación no se ha inspirado en Mobile First (dado el objetivo de la aplicación para venta de material de fotografía de segunda mano), la aplicación es responsive y adaptable a diferentes tamaños de pantalla.

En ella se usan de patrones de diseño de interfaces (como el uso de cards y modales) que hacen pensar que no habrá que realizar grandes cambios en dispositivos.

• Navegación.

En relación con el diseño de la navegación se ha hecho a traves de
con un menú horizontal común a todos los programas que se encuentra en la parte superior de la aplicación.
El menú ocupa todo el ancho de la pantalla y que contiene
enlaces directos. Los enlaces se activan o desactivan en función del programa que se está ejecutando, y si el usuario se ha identificado.

• Cards.

En la página principal de la aplicación, la información de las diferentes categorías de productos se
agrupa de forma en tarjetas o Cards que se muestran agrupadas en una fila. El número
de tarjetas por fila es de 4 y se permite movimiento entre categorías a través de un carrusel.
El uso del carrusel para mostrar imágenes permite, entre otras cosas, reducir la cantidad de imágenes cargadas al mismo tiempo . Al seleccionar una tarjeta se muestra en formato table los productos de la categoría.

• Modals
El uso de modales permite atraer toda la atención del usuario hacia un contenido determinado que se muestra superpuesto al contenido previo. Los modales se utilizan para mostrar información como alternativa a los cards y forzar que se haya visto, dado que el usuario tiene que cerrar el modal. También para la confirmación/cancelación de operaciones como las de borrado.

En lo concerniente a los formularios, también se han aplicado varios principios y patrones de diseño de interfaces.

• Relleno de campos. Este principio hace referencia a la gestión de los inputs cuando estos se muestran vacíos. En el diseño de la aplicación se ha previsto

el uso de valores existentes para alimentar los campos de un formulario cuando el usuario este actualizando o modificando la información, así como el uso de placeholders en aquellos casos en que se trata de añadir nueva información al sistema.

• Password y nombre de usuario.

En la página de registro se imponen unas condiciones para aceptar un nombre de usuario y la palabra de paso, siendo estas un número mínimo de 7 caracteres, con una mayúscula, una minúscula , un número y un carácter especial. El objetivo es que el usuario introduzca palabras más robustas y seguras frente a un ataque.
Para garantizar que la contraseña introducida por el usuario es correcta el sistema solicita la confirmación de la contraseña.

• Dirección de email.

El sistema pide en registro una dirección de correo. Esta dirección debe ser única en la base de datos.

• Accesibilidad.

Se han tenido en cuenta las indicaciones de la accesibilidad usando un elemento <nav> en barra de navegación para identificarla a los usuarios. También se han respetado los etiquetados de categorías de heading.

12. **Seguridad**

El objetivo de un esquema de seguridad es aislar una aplicación para reducir al máximo el riesgo de intrusión en el servidor de base de datos y proteger la información.

Las medidas tomadas son :

- El acceso a la parte privada que implica uso de datos se realiza mediante sesiones y el servidor conserva información del usuario durante la sesión .

- Todas las entradas se validan dos veces, una en el lado del cliente en JavaScript y otra en el lado del servidor.

- Todas las entradas son filtradas en los campos de entrada eliminando caracteres HTML.

- La password se encripta.

- Todos los accesos a base de datos se hacen mediante las sentencias "prepare" y "execute" con el uso de values '?'.

- Cuando un usuario se registra, la nueva cuenta se crea desactivada. Asi el usuario recibe en la cuenta de email que indica , un link de activación con un token. Mediante esta operación se verifica que la cuenta de email exista realmente. El mismo procedimiento se utiliza para el cambio de la contraseña.

13. **Pruebas de funcionalidad de la aplicación**

Se controla la gestión de errores de ejecución de comandos de BD con el uso de las instrucciones "Try" y "catch" para evitar errores fatales que interrumpa la ejecución del programa.

Se han realizado pruebas de uso en los navegadores Google Chrome y Mozilla Firefox.

14. **Instrucciones de implantación**

La instalación de la aplicación en el servidor es rápida y sencilla.

15. Proyección a futuro

La aplicación puede ser ampliada con varias funcionalidades.
La primera será la ampliación de la información asociada a cada producto. Asi como la incorporación de 6 imagenes mínimo (ahora hay una)

También será necesaria incorporar acceso a una pasarela de pago.

16. Ficheros del proyecto

A continuación, se indican los ficheros que se presentan junto con esta documentación y que contienen el código de las aplicaciones:

Foap2023ProyecteFinal.zip . Es una fichero zip que contiene el código de la aplicación y también un archivo de configuración de la base de datos que se subirá al Moodle de Sarti. Incluye documentación del proyecto, el diagrama de la aplicación y el diagrama de la estructura de datos .

Existe un *repositorio GIT público* de la aplicación:
https://github.com/BelucaNM/Foap2023ProyecteFinal.git.

La aplicación producto de este proyecto *se encuentra desplegada* en la dirección: http://147.83.7.203/objetivos/web/