

Memoria proyecto

Venta de camisetas



Trabajo realizado por:

Marlon Campoverde Méndez

David Garrido Rojas

David Llanes Martín

Miguel de Areba Oliva

José María de la Fuente

Marc Segalés Parés

Índice

Objetivos del proyecto	3
Tipos de usuario	3
Listado de scripts para las vistas	4
Scripts adicionales	5
Estructura de la Base de Datos	6

Objetivos del proyecto

Shopirt pretende innovar el mundo de las camisetas ofreciendo nuevos diseños de manera sostenible y ecológica ya que la gran mayoría de los materiales utilizados para la creación de cada producto son lo más sostenibles posibles.

Con este proyecto se pretende crear un amplio escaparate de camisetas para así poder abarcar gente de todo tipo y edad. También contamos con la posibilidad de crear tus propios diseños para aquellas personas que no se sientan identificadas con ningún diseño.

Por otro lado, contamos con un sistema de valoraciones por parte de los clientes que se activará si una persona ha comprado una camiseta para que pueda valorarla.

La página contará con un sistema de búsqueda por temas para que cada uno pueda ver las camisetas según sus gustos.

El objetivo final de este proyecto es crear una tienda de camisetas que pueda llegar a todo tipo de personas sin importar el lugar ni la edad, moda para todos. Al mismo tiempo de reducir la huella medioambiental lo máximo posible.

Tipos de usuario

Administrador: será el encargado de la administración de la página por lo que tendrá todos los privilegios. Además se encargará de:

- Subir productos nuevos.
- Modificar los productos ya existentes.
- Moderar las reseñas, eliminando o amonestando si incumplen con las normas.

Usuario registrado: podrá tramitar pedido además de dejar los productos comprados. También podrá hacer uso de la lista de favoritos y se quedarán los productos guardados en el perfil del usuario.

Usuario sin registrar: No podrá comprar ningún producto. La única acción que podrá realizar es ver productos.

Listado de scripts para las vistas

Para la implementación de este proyecto haremos uso de dos patrones de diseño software por una lado el patrón “Modelo vista controlador” donde el modelo es quien gestiona la información, el controlador se encarga de responder a eventos y comunicarlos al modelo, por último la vista quien gracias al modelo muestra la información necesaria al usuario. Por otro lado, tenemos al patrón “Singleton” el cual nos permite garantizar que una clase tenga una única instancia.

El proyecto lo vamos a dividir en cuatro carpetas:

- ❖ **Modelo:** se comunica con la base de datos y devuelve la información al controlador
- ❖ **Vista:** se encarga de generar código html para mostrarlo al usuario.
- ❖ **Controlador:** se comunica con el modelo y devuelve datos a la vista.
- ❖ **Comun:** archivos que comparten similitudes y permiten crear nuevos sin tener que escribir el mismo código en todos los nuevos.

Los scripts que se incluirán para las vistas son:

- **index.php:** esta será la vista principal de la web desde aquí se podrá navegar por las demás secciones de la página web. Los usuarios podrán elegir las diferentes secciones que tiene la página.
- **ventas.php:** aquí se podrá visualizar en forma de matriz cada una de las camisetas que se venden así como el precio de venta y para los usuarios registrados una estrella que les permite agregar a la lista de favoritos.
- **producto.php:** desde la página de ventas se accede a esta, donde se podrá ver información más detallada del producto seleccionado como una breve descripción, valoraciones de los demás usuarios, imágenes desde distintas perspectivas y el precio. Se podrá realizar cuatro acciones: seleccionar la talla, añadir a la lista de favoritos y a la de compra y dejar una valoración. Todas estas acciones son accesibles para los usuarios registrados.
- **perfilUsuario.php:** en esta página se encuentra toda la información del usuario así como la opción de editar esta información. Desde aquí también se le permite ver la información de los productos que ha comprado.
- **perfilAdmin.php:** se separa con la de usuario ya que desde aquí el administrador podrá realizar todas las acciones para las que está capacitado.
- **registro:** para todos los usuarios que no se hayan registrado podrá aportar la información necesaria para crear una cuenta en Shopirt. Los datos que se solicitarán serán: nombre completo, nickname, contraseña y repetir contraseña.
- **devoluciones.php:** desde aquí los usuarios registrados podrán procesar una devolución de un producto. Se le pedirá el motivo de la devolución para poder procesarla.
- **carritoDeLaCompra:** desde aquí se listará todos las camisetas que el usuario ha seleccionado para comprar. Podrá ver el precio de cada artículo, el número de camisetas en caso de que sean iguales, el valor total a pagar. En esta página hay dos opciones, por un lado podemos eliminar productos quitarlos de la lista y por otro lado “continuar” donde se confirmará el pedido. Se vacía una vez de confirma.

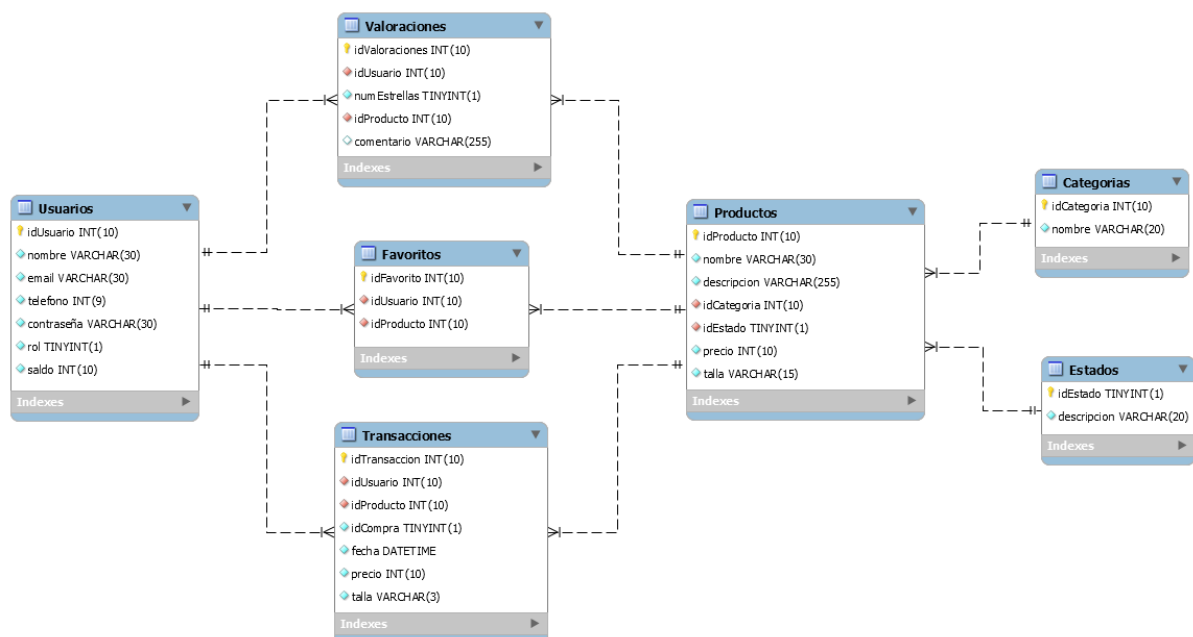
- **pedido.php:** página donde se aporta la información final para concluir con la compra como, dirección de envío y forma de pago. Además de comunicar a la base de datos que se descuenta las existencias correspondientes y crear la información necesaria para identificar el usuario y el producto.
- **listaFavoritos:** se listará todos los productos que le hayan gustado al usuario y le haya dado a la estrella. Se podrá realizar dos acciones, añadir al carrito o eliminar de la lista de favoritos.
- **login.php:** gestiona el inicio de sesión donde se comprueba los datos y en caso de error mostrar un error.
- **subirProducto.php:** accesible para los administradores, podrán subir una nueva camiseta aportando todos los datos necesarios.

Scripts adicionales

En este apartado vamos a definir los scripts que se encargan de procesar la información para mostrarla a la vista o comunicarse con las bases de datos.

- **procesarLogin.php:** con los datos aportados a través de un formulario conecta con la base de datos y comprueba si la información aportada es correcta, en caso afirmativo muestra un mensaje de bienvenida. En caso contrario comunica el error al usuario. La información se pasa a login ya que es quien se encarga de mostrar de forma visual a los usuarios.
- **procesarLogout.php:** desconecta al usuario y se le transfiere al inicio (index)
- **procesarRegistro.php:** al igual que procesar login realiza la misma función pero con los registros.
- **procesarSubirProducto.php:** la información aportada por la vista es transmitida a la bd.
- **procesarPedido.php:** transmite la información de la vista a las base de datos.
- **procesarEditarPerfil.php:** se encarga de reemplazar la información aportada por el usuario comunicándose con la bd para hacer esta modificación.
- **clases:** para abstraer el código se crearán clases como: Producto.php, Usuario.php, Valoracion.php, Transacciones.php, Categoria.php, Estado.php
- **bd.php:** gestiona la conexión con la base de datos.
- **cabecera.php:** dado que la información que contiene la cabecera es común para todas las vistas se escribe una vez y se añade en las que la requieran.
- **pie.php:** idem que la cabecera.

Estructura de la Base de Datos



Nuestra base de datos cuenta con 8 tablas:

- Tabla Valoraciones, para almacenar las valoraciones que hacen los usuarios sobre los productos página de la web, con los siguientes atributos:
 - idValoraciones, identificador que será clave primaria.
 - idUsuario, para poder relacionar una valoración con el usuario que la realiza.
 - numEstrellas, valoramos los productos de 0 a 5.
 - idProducto, para identificar el producto que se ha valorado.
 - Comentario, para complementar la puntuación del usuario a su valoración de un producto.
- Tabla Usuarios, para almacenar cada usuario y su información, con los siguientes atributos:
 - idUsuario, identificador que será clave primaria.
 - Nombre, nombre del usuario.
 - Email, correo electrónico del usuario.
 - Teléfono, número de teléfono del usuario.
 - Rol, tipo de usuario de la web, de inicio contamos solo con dos valores, 1 para usuarios y 2 para administradores.
 - Saldo, el dinero disponible del usuario en la web.
- Tabla Historial de compras, para llevar un registro de todos los productos que ha adquirido un usuario, con los siguientes atributos:
 - idHistorial, identificador que será clave primaria.
 - idUsuario, para identificar el usuario.
 - idProducto, para identificar el producto que ha sido comprado.

- Tabla Favoritos, para que el usuario almacene sus productos favoritos, con los siguientes atributos:
 - idFavorito, identificador que será clave primaria.
 - idUsuario, para identificar el usuario.
 - idProducto, para identificar el producto guardado en favoritos.

- Tabla Productos, para almacenar los productos de nuestra página web, con los siguientes atributos:
 - idProducto, identificador que será clave primaria.
 - Nombre, nombre del producto.
 - Descripción, descripción del producto.
 - idCategoria, para identificar ese producto dentro de una categoría.
 - idEstado, para identificar el estado de ese producto.
 - Precio, el precio del producto.
 - Talla, tallas disponibles.

- Tabla Estados, para determinar el estado en el que se encuentra un producto, con los siguientes atributos:
 - idEstado, identificador que será clave primaria.
 - Descripción, para conocer el estado de un producto, si está agotado, disponible para comprar, disponible para reservar...

- Tabla Categorías, para clasificar y catalogar los productos, con los siguientes atributos:
 - idCategoria, identificador que será clave primaria.
 - Nombre, nombre de cada categoría.

- Tabla transacciones, para gestionar una compra de un producto, con los siguientes atributos:
 - idTransaccion, identificador que será clave primaria.
 - idUsuario, para indicar el usuario que realiza la transacción.
 - idProducto, para identificar el producto.
 - idCompra, indica si la transacción es una compra o una devolución.
 - Fecha, para registrar la fecha en que se produce la transacción.
 - Precio, el precio de la transacción.
 - Talla, la talla del producto elegida por el usuario.

Funcionamiento:

Para hacer funcionar la página se parte desde el script index.php.

Para poder ver las distintas funcionalidades de los usuarios (user y admin) ya que no tenemos bases de datos hemos creado de forma interna a los usuarios:

- ➔ Nombre de usuario: *usuario1* y contraseña: *pass*
- ➔ Administrador: *admin* y contraseña: *adminpass*

Entrega 3:

Listado de scripts:

Mantenemos la idea principal de usar el patrón DAO así como una clase intermedia entre el DAO y las vistas, esta es las clases SA. La clase SA se encarga de comunicar a la clase DAO que realice las consultas pertinentes y transferir los datos a quien los demande.

Por otro lado, en casi todas las vistas se ha introducido un script más. Por ejemplo en el script *"tienda.php"* va acompañado del script *"mostrarTienda.php"*. Hacemos esto para que visualmente sea más fácil, *"tienda.php"* contiene la estructura básica de la página, la cabecera, el cuerpo con su contenido correspondiente que es desde donde se llama al script *"mostrarTienda.php"* y finalmente en este tenemos el pie de la página. Por otro lado, *"mostrarTienda.php"* es quien se encarga de traer los datos de cada producto para sacarlos por pantalla, se comunica con su script SA correspondiente. Una vez tenemos la información que queremos mostrar la colocamos con código HTML.

Listado de funcionalidades:

- Login/registrarse 100% funcional.
- Carrito
- catálogo
- página de producto, con interacción.
- admin: subir, eliminar producto, editar producto: solo vista
- realizar pedido
- Inicio: mostrará novedades (últimos productos añadidos) o más vendidos.
- perfil: haciendo click arriba a la derecha en el nombre del usuario direcciona a la página dedicada a visualizar los datos del usuario.
- Favoritos: se guardan los productos que al usuario le gustan con la posibilidad de eliminar o trasladar al carrito. Se guarda en la base de datos.

Por otro lado, hay varios archivos php que están vacíos para futuras implementaciones. También la página de inicio no está del todo terminada puesto que no sabemos si poner las "novedades" productos recién añadidos, los más vendidos, los mejores puntuados o una mezcla de todo.

Para futuras entregas la base de datos se construirá sola en el script *"includes/create_bd.php"* creará la base de datos, las tablas y le agregara datos ficticios para poder funcionar.

Por ello se ha creado el script *"inicio.php"* el cual contendrá todo lo que se quiera mostrar en la página de Inicio, la cual está en desarrollo.

Debido a esto *"index.php"* es quien llama a esta vista pero a su misma vez una llamada al script de crear las bases de datos. *Index.php* es la puerta de enlace de la aplicación desde donde se inicia, como en varios lugares se hace referencia al inicio se ha decidido separarlos para que solo se llame una vez al script que se encarga de cargar la base de datos.

Por último, en las tablas que están en la base de datos no es el diseño final, según vayamos avanzando en la construcción iremos agregando la tabla correspondiente.