

Documentación Actividad Tema 4

Alicia Ortega Castro

DWES

ÍNDICE

1. Descripción breve del proyecto.....	3
2. Requisitos funcionales.....	3
3. Datos que se van a almacenar.....	3
4. Esquema Entidad-Relación.....	4
5. Paso a Tablas.....	4
6. Desplazamiento entre pantallas.....	6
6.1 Desplazamiento de Cliente.....	6
6.2 Desplazamiento de Administración.....	7
7. Detalles de cada fichero.....	8
7.1 Login.php.....	8
7.2 Sesiones.php.....	8
7.3 Cabecera.php.....	8
7.4 Categorías.php.....	8
7.5 Productos.php.....	9
7.6 Carrito.php.....	11
7.7 Procesar_pedido.php.....	12
7.8 Pedidos.php.....	13
7.9 Pedido_producto.php.....	14
7.10 Clientes.php.....	15
7.11 Logout.php.....	16
7.12 Configuración.xml y configuración.xsd.....	16
7.13. Bd.php.....	16
8. Funciones de bd.php.....	16
1. `leer_config(\$nombre, \$esquema)`	17
2. `conexionBD()`	17
3. `comprobar_usuario(\$nombre, \$clave)`	17
4. `cargar_categorias()`	18
5. `cargar_categoria(\$codCat)`	18
6. `cargar_productos_categoria(\$codCat)`	19
7. `cargar_productos(\$arrayCodigosProductos)`	19
8. `insertar_pedido(\$carrito, \$codCliente, \$totalArticulos)`	19
9. `cargar_pedidos_cliente(\$codCliente)`	20
10. `cargar_pedidos()`	20
11. `pedido_recibido(\$codPedido)`	21
12. `pedido_enviado(\$codPedido)`	21
13. `cargar_productos_pedido(\$codCliente, \$codPedido)`	21
14. `cargar_datos_pedido(\$codCliente, \$codPedido)`	22
15. `cargar_datos_cliente(\$codCliente)`	22
16. `insertar_categoria(\$nombre, \$descripcion)`	23
17. `insertar_producto(\$nombre, \$descripcion, \$stock, \$precio, \$codCategoria)`	23
18. `cambiar_stock(\$codProducto, \$unidades)`	24
19. `cargar_usuarios()`	24

20. `insertar_usuario(\$nombre, \$contrasena, \$direccion, \$rol)`	24
21. `eliminar_usuario(\$codCliente)`	25
22. `consultarStockProducto(\$codProducto)`	25
9. Consideraciones extra para evaluar la aplicación.....	26

1.Descripción breve del proyecto

Va a realizarse una aplicación web cuyo propósito es hacer la parte de gestión del almacén que provee a varias tiendas de la misma empresa de componentes informáticos y videojuegos.

2.Requisitos funcionales

El perfil de administrador pertenece al administrativo del almacén y es el que tiene la capacidad de ver, añadir o eliminar clientes, así como añadir categorías de productos o productos y cambiar el stock de los mismos.

Los clientes podrán ver las categorías, los productos que hay en cada categoría, el stock y añadir al carrito los productos deseados, eliminarlos y vaciar la cesta o realizar el pedido.

Los clientes y el administrador podrán ver el estado del pedido que cambiará entre 'pendiente', 'enviado' y 'recibido'. El administrador podrá cambiar el estado a "enviado" cuando se lo notifiquen los operarios del almacén; y el cliente podrá confirmar si lo ha recibido: lo cual cambiará el estatus del pedido a "recibido".

Ambos podrán ver el resumen de cada pedido, donde se incluye los datos de los productos pedidos, los datos del pedido en sí mismo como la fecha, y el número de artículos totales.

No es necesario realizar pago ya que es una distribución de productos para tiendas de la misma empresa, pero sí es importante el correcto recuento de stock.

Tampoco se permitirá que personas ajenas a la cadena de tiendas puedan registrarse un perfil de cliente ya que es el almacén quien sabe a quienes tiene que distribuir.

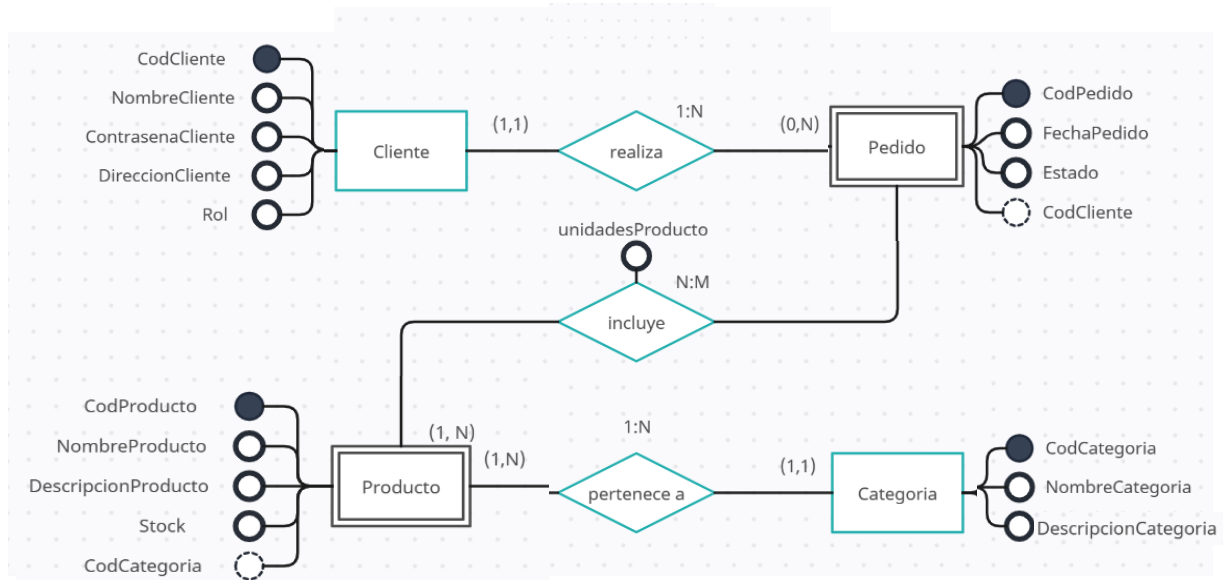
3.Datos que se van a almacenar

Se necesita guardar el código, nombre, dirección y contraseña del cliente. Además del rol: los roles podrán ser de cliente o administrador.

De los pedidos se especifica quien realizó el pedido, el código del pedido, la fecha en la que se realizó, su estado (pendiente, enviado o recibido), y las unidades de producto.

También se almacenarán de las categorías su código, nombre y descripción; y de los productos, su código, nombre, categoría a la que pertenece, descripción y stock.

4. Esquema Entidad-Relación



Se puede observar que “Pedido” y “Producto”, son entidades débiles de “Cliente” y de “Categoría” respectivamente, por lo que heredan la clave primaria de la entidad fuerte de la que dependen.

5. Paso a Tablas

Al haber dos entidades débiles y al tener en cuenta las cardinalidades, las tablas quedarían así:

Cliente
<u>CodCliente</u>
NombreCliente
ContraseñaCliente
DirecciónCliente
Rol

Pedido
<u>CodPedido</u>
CodCliente
Estado
FechaPedido
Total

Producto
<u>CodProducto</u>
NombreProducto
DescripciónProducto
Stock
CodCategoría

Precio

Pedido-Producto
<u>CodPedidoProducto</u>
<u>CodPedido</u>
<u>CodProducto</u>
unidadesProducto

Categoria
<u>CodCategoria</u>
NombreCategoria
DescripcionCategoria

Aparece la tabla Pedido-Producto por la relación de *muchos a muchos* que hay entre “Pedido” y “Producto”. Se crea una clave primaria para la nueva tabla y como Foreign Keys tiene las claves primarias de las dos tablas relacionadas, además del atributo que había en la relación.

Se ha creado la base de datos “tiendainformatica” a partir de las tablas mostradas arriba.

Los atributos que empiezan por “Cod” son el código de la tabla correspondiente, tienen autoincremento y son la Primary Key si a continuación tienen el mismo nombre de la tabla, en otro caso son Foreign Key, en cualquier caso son enteros con longitud de 10 (salvo en el caso de “CodCliente” que es smallint con longitud de 4). Los atributos que empiezan por “Nombre” son Unique.

Los atributos “Descripcion” son longtext con longitud de 500, la contraseña de Cliente puede tener una longitud de 255, el motivo se verá más adelante.

A través de bases de datos se creó un administrador y un cliente así como diferentes categorías y productos para hacer pruebas, por lo que en los siguientes elementos que sean creados, y en algunos que han permanecido tras las pruebas, se podrá observar que los números son discontinuos. Esto se debe a que tras borrar los elementos creados he decidido no reiniciar el autoincremento.

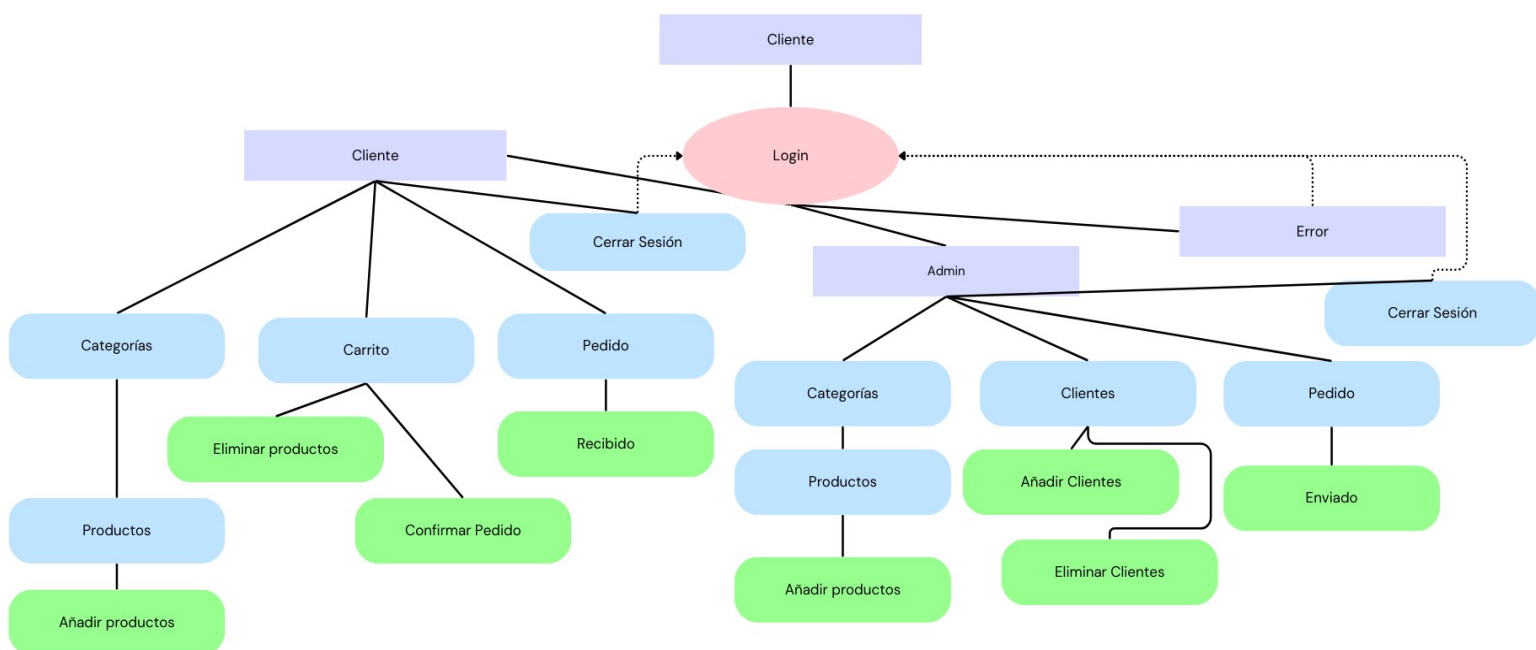
Las categorías que han permanecido creadas son “videojuegos”, “ratones”, “teclados” y “portátiles” y a cada categoría se le han añadido 2 productos con cantidades de stock variadas.

Hay varios usuarios que han permanecido. Uno de tipo administrador es Admin0 con contraseña “0” y uno de tipo cliente es Tienda 1 con contraseña “1”.

6.Desplazamiento entre pantallas.

Habr  una pantalla de login, donde adem s de filtrar el acceso ya que no hay usuarios invitados, permitir  mostrar diferentes funcionalidades e informaci n a los clientes normales y a los “super clientes” es decir, los administradores que se caracterizan por tener rol igual a 0.

Si los clientes o el administrador cierran sesi n vuelven a la pantalla de login. Si hay un error en el inicio de sesi n se muestra el mensaje pero vuelven a la pantalla de login.



6.1 Desplazamiento de Cliente

Este ser a el recorrido de un cliente est ndar (cuando rol es igual a 1):

Tras el login exitoso cualquier pantalla que vea incluirá una cabecera con las opciones de navegación de **“Categorías-Ver carrito-Pedidos-Cerrar sesión”**.

La pantalla que sigue al login es la de categorías. Cuando seleccione una categoría podrá ver los productos y cuando esté en la pantalla de productos podrá añadir elementos al carrito si hay stock suficiente.

En la pantalla de carrito podrá ver los elementos añadidos y podrá eliminarlos si quiere o realizar el pedido, para lo que se le pedirá que vuelva a introducir su contraseña. Si la contraseña es correcta se muestra un resumen de su pedido y después podrá volver al carrito, que ahora se mostrará vacío.

En la pantalla de ver pedidos podrá comprobar los pedidos realizados, su estado y si lo ha recibido podrá marcarlo como tal para cambiar su estado a finalizado. También podrá pulsar para ver el resumen de cada pedido.

En la pantalla de cerrar sesión, tras confirmar la acción el cliente podrá volver a la pantalla de login.

6.2 Desplazamiento de Administración

Un cliente de administración tendrá un desplazamiento y funcionalidades similares pero su menú de navegación será **“Categorías-Clientes-Pedidos-Cerrar sesión”**.

La sección de Categorías funciona similar a los clientes, a partir de ahí puede ver los productos, con la diferencia que en categorías puede añadir nuevas categorías y en productos de esa categoría podrá añadir nuevos. Además de cambiar el stock de cada producto. No tendrá carrito.

En Clientes puede ver los clientes que hay registrados, y puede añadir nuevos o eliminar los que haya. No podrá auto-eliminarse.

En Pedidos puede ver los pedidos que se han realizado y su estado. Puede cambiar el estado de *pendiente* a *enviado* y podrá ver el resumen de cada pedido.

Cerrar sesión funciona igual que un cliente estándar.

7. Detalles de cada fichero

7.1 Login.php

Mediante formulario se pide el nombre y contraseña de usuario. Si falla puede volver a intentarlo tantas veces como quiera. Si los campos introducidos coinciden con lo guardado en base de datos se crea la sesión de usuario donde se guardan todos los valores del cliente que hay en la tabla Cliente.

7.2 Sesiones.php

En este fichero se guardan las funciones que comprueban que la sesión que se crea tras el login exitoso exista, que la sesión pertenezca a un usuario cliente o que pertenezca a un usuario administrador. De no ser el caso redirigen a login.php.

7.3 Cabecera.php

La sesión de usuario creada tras el login se guarda en una variable “\$cliente” y se muestra el nombre por pantalla. Además, según el ‘Rol’ se mostrarán los enlaces de desplazamiento entre pantallas correspondiente o a cliente o a administrador.

7.4 Categorías.php

Se muestran las categorías que haya guardadas en la base de datos, cada nombre de categoría es un enlace a productos.php donde al enlace se le añade con el formato típico de GET el código de la categoría.

Si se detecta que es un administrador se le mostrará el formulario para añadir una nueva categoría. Se procesa en el mismo archivo y la nueva categoría se mostrará en el mismo formato que las anteriores.

Así es como vería la pantalla un usuario administrador:

Lista de categorías

- [Portátiles](#)
- [Ratones](#)
- [Teclados](#)
- [Videojuegos](#)

Añadir categoría

Nombre de la categoría Descripción de la categoría

7.5 Productos.php

Primero carga el nombre y la descripción de la categoría. Obtiene el código de la categoría de la url mediante GET.

Si no hay productos de la categoría se muestra por mensaje esta circunstancia, si hay los carga y los muestra.

Si no hay stock se muestra que está agotado, si hay stock el cliente podrá pulsar el botón del formulario para añadir al carrito. Hay un botón por producto por lo que no podrá añadir unidades de diferentes productos a la vez pero podrá pulsar tantas veces como quiera. El stock de producto mostrado se va actualizando para que no pueda añadir más unidades de las que existen, aunque no se cambia el stock de la base de datos hasta que se realice el pedido. Para ello uso los datos guardados en `$_SESSION['carrito']`.

Cuando se añade un producto al carrito se muestra un mensaje de confirmación.

De procesar este formulario se encarga el mismo fichero.

Así vería un cliente esta pantalla:

Usuario: Tienda 1

[Categorías](#) [Ver Carrito](#) [Ver Pedidos](#) [Cerrar sesión](#)

Portátiles

¿Tienes un presupuesto único para un aparato que te sirva para estudiar o para el trabajo pero que a la vez te permita dedicarle un montón de horas a tu hobby favorito? Pues tenemos la solución perfecta con nuestra selección de portátiles gamers. Tiene leds configurables para que tengas ambientes diferentes según estés en la biblioteca y quieras discreción, o en tu cuarto y prefieras iluminación de discoteca.

Nombre	Descripción	Stock en Almacén	Precio de Venta	Añadir al Carrito
Portátil pesado	El fuerte de este portátil es sin duda su batería. Con la adición de tamaño y peso consigue una autonomía extra de 2 horas con respecto a la competencia. Tal vez pierde un poco de portabilidad, pero merece la pena si necesitas estar fuera de casa sin preocuparte en exceso por la autonomía.	4	556.90	<input type="text" value="0"/> <input type="button" value="Añadir al carrito"/>
Portátil helicóptero	Destaca por su excelente sistema de refrigeración con el que podrás exprimir al extremo las capacidades de este ordenador. Conviene sujetarlo para que no eche a volar.	Producto	721.98	<input type="text" value="0"/> <input type="button" value="Añadir al carrito"/>

Si el usuario es administrador no podrá hacer la parte relacionada con el carrito, pero podrá cambiar el stock de los productos existentes o añadir nuevos productos. De procesar estos formularios se encarga el propio fichero.

Así ve esta pantalla un administrador:

Usuario: Admin0

[Categorías](#) [Ver Clientes](#) [Ver Pedidos](#) [Cerrar sesión](#)

Videojuegos

Nos hemos especializado en claves de los mejores videojuegos para ordenador. Consigue tus videojuegos para ordenador favoritos al mejor precio y con la parte coleccionista de las claves físicas. Todas las claves que tenemos son para Steam, la plataforma más popular para los videojuegos en PC.

Nombre	Descripción	Stock en Almacén	Precio de Venta
Baldur's Gate 3	Baldur's Gate 3 es el juego que mejor recibimiento por la parte de la comunidad ha tenido en su año de lanzamiento, 2023. Sin lugar a dudas un imprescindible en tu biblioteca	84 Indique la cantidad real de stock <input type="text" value="84"/> <button>Actualizar Stock</button>	59.99
Spirit's Farer	Este juego indie ha conquistado nuestro corazón con su preciosa historia, arte acogedor y jugabilidad sencilla. Si estás pasando por un duelo te recomendamos a que esperes antes de jugarlo porque te derretirá el corazón	2 Indique la cantidad real de stock <input type="text" value="2"/> <button>Actualizar Stock</button>	15.96

Añadir producto

Nombre del producto	<input type="text"/>	Descripción del producto	<input type="text"/>	Stock del producto	<input type="text"/>
	Precio del producto	<input type="text"/>	<button>Añadir producto</button>		

7.6 Carrito.php

En carrito.php se comprueba que el usuario sea un cliente mediante las funciones de sesiones.php.

Carga los productos que haya en la sesión de carrito y mediante formulario tiene los botones para eliminar las unidades que quiera del producto seleccionado o vaciar por completo el carrito. El resultado de los formularios se procesa en el mismo fichero.

También hay un enlace para realizar el pedido que redirige a procesar_pedido.php

El cliente vería esta página así:

Usuario: Tienda 1

[Categorías](#) [Ver Carrito](#) [Ver Pedidos](#) [Cerrar sesión](#)

Carrito de la compra

Vaciar Carrito

Nombre	Descripción	Unidades	Eliminar
Teclado rocoso	Este teclado es el favorito de aquellas personas que fuera del gimnasio aún quieren entrenar. Hace falta una presión extra para realizar la pulsación. Cuando finalmente pulsas la tecla suena como si chocaras dos piedras.	8	<input type="text" value="1"/> Eliminar
Teclado tormenta	Su nombre viene de que con su potente retroiluminación y su ruido de pulsación mecánica es como ver los rayos y escuchar los truenos de una gran tormenta. Perfecto si no quieres ver la pantalla ni escuchar el audio de nada.	1	<input type="text" value="1"/> Eliminar
Baldur's Gate 3	Baldur's Gate 3 es el juego que mejor recibimiento por la parte de la comunidad ha tenido en su año de lanzamiento, 2023. Sin lugar a dudas un imprescindible en tu biblioteca	7	<input type="text" value="1"/> Eliminar
Spirit's Farer	Este juego indie ha conquistado nuestro corazón con su preciosa historia, arte acogedor y jugabilidad sencilla. Si estás pasando por un duelo te recomendamos a que esperes antes de jugarlo porque te derretirá el corazón	3	<input type="text" value="1"/> Eliminar

[Realizar pedido](#)

7.7 Procesar_pedido.php

Lo primero que vería el cliente sería una petición de introducir la contraseña para confirmar el pedido. Su nombre de usuario la obtengo de la variable \$cliente de la cabecera, la cual era una copia de los datos de la sesión de usuario.

Si se equivoca de contraseña puede repetir hasta que acierte. Una vez acierte la contraseña se insertan los datos en la base de datos y se vacía la sesión de carrito. El cliente verá un resumen de su pedido. Para desplazarse a partir de aquí no he puesto ningún enlace para dirigirlo, pero tiene la cabecera para ir a donde quiera, como a “ver pedidos”.

Usuario: Tienda 1

[Categorías](#) [Ver Carrito](#) [Ver Pedidos](#) [Cerrar sesión](#)

Contraseña errónea, vuelva a intentarlo para procesar el pedido

Por favor, para realizar el pedido introduzca su contraseña

Contraseña

Usuario: Tienda 1

[Categorías](#) [Ver Carrito](#) [Ver Pedidos](#) [Cerrar sesión](#)

Pedido realizado con éxito. Resumen del pedido:

Pedido nº 13

Artículos totales: 19

Nombre	Descripción	Precio Recomendado	Unidades
Teclado rocoso	Este teclado es el favorito de aquellas personas que fuera del gimnasio aún quieren entrenar. Hace falta una presión extra para realizar la pulsación. Cuando finalmente pulsas la tecla suena como si chocaras dos piedras.	78.99	8
Teclado tormenta	Su nombre viene de que con su potente retroiluminación y su ruido de pulsación mecánica es como ver los rayos y escuchar los truenos de una gran tormenta. Perfecto si no quieres ver la pantalla ni escuchar el audio de nada.	34.85	1
Baldur's Gate 3	Baldur's Gate 3 es el juego que mejor recibimiento por la parte de la comunidad ha tenido en su año de lanzamiento, 2023. Sin lugar a dudas un imprescindible en tu biblioteca	59.99	7
Spirit's Farer	Este juego indie ha conquistado nuestro corazón con su preciosa historia, arte acogedor y jugabilidad sencilla. Si estás pasando por un duelo te recomendamos a que esperes antes de jugarlo porque te derretirá el corazón	15.96	3

7.8 Pedidos.php

Tanto cliente como administrador pueden interactuar con este fichero. Si es cliente solo verá sus pedidos, si es administrador verá todos los pedidos.

Si no hay pedidos que mostrar se mostrará un mensaje anunciando esa circunstancia.

Ambos tipos de usuarios tienen un botón en cada fila de pedidos para poder ver un pedido en concreto en detalle. Ese POST lo procesa el fichero `pedido_producto.php`, y aunque el resultado que muestra es muy similar al que se obtiene en `procesar_pedido.php` decidí hacerlo en un fichero aparte para no hacerlo más difícil de leer.

Hay otro botón al lado para cambiar el estado del pedido. Este botón sólo aparece si se cumple la condición necesaria: si el pedido está en pendiente el administrador podrá cambiarlo a enviado, si está en enviado el cliente podrá cambiarlo a recibido.

Usuario: Admin0

[Categorías](#) [Ver Clientes](#) [Ver Pedidos](#) [Cerrar sesión](#)

Pedidos

Código del Pedido	Fecha del Pedido	Artículos Totales	Estado del Pedido	Acciones que puede realizar
13	2023-12-09 03:23:02.000000	19	pendiente	<div>Marcar como Enviado</div> <div>Ver Productos del Pedido</div>

Aquí se ve cómo el botón desaparece para este usuario administrador tras marcarlo como enviado:

Pedido marcado como enviado con éxito

Usuario: Admin0

[Categorías](#) [Ver Clientes](#) [Ver Pedidos](#) [Cerrar sesión](#)

Pedidos

Código del Pedido	Fecha del Pedido	Artículos Totales	Estado del Pedido	Acciones que puede realizar
13	2023-12-09 03:23:02.000000	19	enviado	<div>Ver Productos del Pedido</div>

7.9 Pedido_producto.php

En este fichero se obtienen datos de 4 tablas de la base de datos a la vez: Cliente, Pedido, Producto y Pedido_Producto; aunque los datos del cliente que realizó el pedido solo se muestran si el usuario que consulta es de tipo administrador.

Al final he puesto un enlace para dirigirlo de nuevo a pedidos.php.

Usuario: Admin0

[Categorías](#) [Ver Clientes](#) [Ver Pedidos](#) [Cerrar sesión](#)

Detalles del pedido n° 13

Fecha del pedido: 2023-12-09 03:23:02.000000

Estado del pedido: pendiente

Código del cliente: 9

Nombre del cliente: Tienda 1

Dirección del cliente: Granada Centro - San Antón

Código del Producto	Nombre	Descripción	Precio Recomendado	Unidades
5	Teclado rocoso	Este teclado es el favorito de aquellas personas que fuera del gimnasio aún quieren entrenar. Hace falta una presión extra para realizar la pulsación. Cuando finalmente pulsas la tecla suena como si chocaras dos piedras.	78.99	8
6	Teclado tormenta	Su nombre viene de que con su potente retroiluminación y su ruido de pulsación mecánica es como ver los rayos y escuchar los truenos de una gran tormenta. Perfecto si no quieres ver la pantalla ni escuchar el audio de nada.	34.85	1
7	Baldur's Gate 3	Baldur's Gate 3 es el juego que mejor recibimiento por la parte de la comunidad ha tenido en su año de lanzamiento, 2023. Sin lugar a dudas un imprescindible en tu biblioteca	59.99	7
8	Spirit's Farer	Este juego indie ha conquistado nuestro corazón con su preciosa historia, arte acogedor y jugabilidad sencilla. Si estás pasando por un duelo te recomendamos a que esperes antes de jugarlo porque te derretirá el corazón	15.96	3

Artículos totales del pedido: 19

[Volver a la lista de pedidos](#)

7.10 Clientes.php

Como este fichero solo debería ser visible para administradores uso las funciones de sesiones.php para asegurarme de ello.

En este fichero hay una tabla para mostrar todos los usuarios registrados y dos formularios que se procesan en este mismo fichero: uno para añadir clientes y otro para eliminar el cliente de la fila del botón. Podrá eliminar a otros administradores pero no se podrá eliminar a sí mismo.

La eliminación de clientes funciona tanto si el cliente ha realizado pedidos como si no.

Usuario: Admin0

[Categorías](#) [Ver Clientes](#) [Ver Pedidos](#) [Cerrar sesión](#)

Código del Usuario/a	Nombre	Dirección	Rol	Acciones de administración
8	Admin0	almacén	Administración	No aplica
9	Tienda 1	Granada Centro - San Antón	Cliente	<button>Eliminar Tienda 1</button>
10	Tienda 2	Granada Norte - Estación autobuses	Cliente	<button>Eliminar Tienda 2</button>
11	Tienda 3	Granada Armilla	Cliente	<button>Eliminar Tienda 3</button>
12	Admin 0	almacén	Administración	<button>Eliminar Admin 0</button>

Añadir Clientes

Nombre de Usuario Clave provisional asignada Ubicación
 Es cliente o administración Añadir Cliente

7.11 Logout.php

La última pantalla pertenece a este fichero. Borra la sesión e imprime un mensaje de despedida y un enlace hacia el login.

7.12 Configuracion.xml y configuracion.xsd

Su única finalidad es servir datos para las funciones de conexión a base de datos.

7.13. Bd.php

Este fichero es el verdadero corazón de la aplicación. Nutre de todas las funciones que interaccionan con la base de datos y que están repartidas por la mayoría de los demás ficheros. Para profundizar más en este fichero vaya al epígrafe punto 8 correspondiente a las funciones de bd.php.

8.Funciones de bd.php

A continuación mostraré de forma resumida las 21 funciones de este fichero. Para más información recomiendo consultar el fichero ya que hay líneas comentadas que explican lo necesario.

1. `leer_config(\$nombre, \$esquema)`

Descripción: Lee la configuración de los archivos de configuración.

Parámetros:

- `\$nombre` (string): Ruta del archivo de configuración.
- `\$esquema` (string): Ruta del archivo de esquema para validar la configuración.

Tipo de Retorno: array|false. Un array con la información de conexión o FALSE si hay un error.

Ejemplo:

```
$configuracion = leer_config("configuracion.xml", "configuracion.xsd");
```

2. `conexionBD()`

Descripción: Crea una conexión a la base de datos.

Parámetros: Ninguno.

Tipo de Retorno: PDO|false. Una instancia de PDO si la conexión es exitosa, FALSE si hay un error.

Ejemplo:

```
$conexion = conexionBD();
```

3. `comprobar_usuario(\$nombre, \$clave)`

Descripción: Comprueba si el usuario existe en la base de datos y verifica la contraseña.

Parámetros:

- `\$nombre` (string): Nombre del usuario.

- `\$clave` (string): Contraseña del usuario.

Tipo de Retorno: array|false. Un array con los datos del usuario si existe y la contraseña es correcta, FALSE si no existe o hay un error.

Ejemplo:

```
$usuario = comprobar_usuario("nombre_usuario", "contrasena");
```

4. `cargar_categorias()`

Descripción: Carga el nombre y el código de todas las categorías.

Parámetros: Ninguno.

Tipo de Retorno: PDOStatement|false. Un objeto PDOStatement si hay resultados, FALSE si hay un error.

Ejemplo:

```
$categorias = cargar_categorias();
```

5. `cargar_categoria(\$codCat)`

Descripción: Carga el nombre y la descripción de una categoría específica.

Parámetros:

- `\$codCat` (int): Código de la categoría.

Tipo de Retorno: array|false. Un array con la información de la categoría si existe, FALSE si no existe o hay un error.

Ejemplo:

```
$categoricalInfo = cargar_categoria(1);
```

6. `cargar_productos_categoria(\$codCat)`

Descripción: Carga los productos de una categoría específica.

Parámetros:

- `\$codCat` (int): Código de la categoría.

Tipo de Retorno: array|int|false. Un array con los productos si hay resultados, 0 si no hay productos, FALSE si hay un error.

Ejemplo:

```
$productos = cargar_productos_categoria(1);
```

7. `cargar_productos(\$arrayCodigosProductos)`

Descripción: Carga información de productos basada en un array de códigos de productos.

Parámetros:

- `\$arrayCodigosProductos` (array): Array de códigos de productos.

Tipo de Retorno: PDOStatement|false. Un objeto PDOStatement si hay resultados, FALSE si hay un error.

Ejemplo:

```
$productosCarrito = cargar_productos([1, 2, 3]);
```

8. `insertar_pedido(\$carrito, \$codCliente, \$totalArticulos)`

Descripción: Inserta un nuevo pedido en la base de datos.

Parámetros:

- `\$carrito` (array): Array asociativo con códigos de productos y unidades.

- `\$codCliente` (int): Código del cliente.

- ``$totalArticulos` (int)`: Número total de artículos en el carrito.

Tipo de Retorno: `int|false`. El código del pedido si la inserción es exitosa, `FALSE` si hay un error.

Ejemplo:

```
$pedidoCodigo = insertar_pedido(["1" => 2, "3" => 1], 1, 3);
```

9. ``cargar_pedidos_cliente($codCliente)``

Descripción: Carga los pedidos de un cliente.

Parámetros:

- ``$codCliente` (int)`: Código del cliente.

Tipo de Retorno: `array|false`. Un array con los pedidos si hay resultados, `FALSE` si no hay pedidos o hay un error.

Ejemplo:

```
$pedidosCliente = cargar_pedidos_cliente(1);
```

10. ``cargar_pedidos()``

Descripción: Carga todos los pedidos registrados.

Parámetros: Ninguno.

Tipo de Retorno: `array|false`. Un array con los pedidos si hay resultados, `FALSE` si no hay pedidos o hay un error.

Ejemplo:

```
$pedidos = cargar_pedidos();
```

11. `pedido_recibido(\$codPedido)`

Descripción: Actualiza el estado de un pedido a "Recibido".

Parámetros:

- `\$codPedido` (int): Código del pedido.

Tipo de Retorno: bool. TRUE si la actualización es exitosa, FALSE si hay un error.

Ejemplo:

```
$actualizado = pedido_recibido(1);
```

12. `pedido_enviado(\$codPedido)`

Descripción: Actualiza el estado de un pedido a "Enviado".

Parámetros:

- `\$codPedido` (int): Código del pedido.

Tipo de Retorno: bool. TRUE si la actualización es exitosa, FALSE si hay un error.

Ejemplo:

```
$actualizado = pedido_enviado(1);
```

13. `cargar_productos_pedido(\$codCliente, \$codPedido)`

Descripción: Carga información de los productos de un pedido.

Parámetros:

- `\$codCliente` (int): Código del cliente.

- `\$codPedido` (int): Código del pedido.

Tipo de Retorno: array|false. Un array con la información de los productos si hay resultados, FALSE si no hay productos o hay un error.

Ejemplo:

```
$productosPedido = cargar_productos_pedido(1, 1);
```

14. `cargar_datos_pedido(\$codCliente, \$codPedido)`

Descripción: Carga los datos de un pedido.

Parámetros:

- `\$codCliente` (int): Código del cliente.
- `\$codPedido` (int): Código del pedido.

Tipo de Retorno: array|false. Un array con la información del pedido si hay resultados, FALSE si no hay pedido o hay un error.

Ejemplo:

```
$datosPedido = cargar_datos_pedido(1, 1);
```

15. `cargar_datos_cliente(\$codCliente)`

Descripción: Carga los datos de un cliente.

Parámetros:

- `\$codCliente` (int): Código del cliente.

Tipo de Retorno: array|false. Un array con la información del cliente si hay resultados, FALSE si no hay cliente o hay un error.

Ejemplo:

```
$datosCliente = cargar_datos_cliente(1);
```

16. `insertar_categoria(\$nombre, \$descripcion)`

Descripción: Inserta una nueva categoría en la base de datos.

Parámetros:

- `\$nombre` (string): Nombre de la categoría.
- `\$descripcion` (string): Descripción de la categoría.

Tipo de Retorno: bool. TRUE si la inserción es exitosa, FALSE si hay un error.

Ejemplo:

```
$insertado = insertar_categoria("Nueva Categoría", "Descripción de la nueva categoría");
```

17. `insertar_producto(\$nombre, \$descripcion, \$stock, \$precio, \$codCategoria)`

Descripción: Inserta un nuevo producto en la base de datos.

Parámetros:

- `\$nombre` (string): Nombre del producto.
- `\$descripcion` (string): Descripción del producto.
- `\$stock` (int): Cantidad en stock del producto.
- `\$precio` (float): Precio del producto.
- `\$codCategoria` (int): Código de la categoría del producto.

Tipo de Retorno: bool. TRUE si la inserción es exitosa, FALSE si hay un error.

Ejemplo:


```
$insertado = insertar_producto("Nuevo Producto", "Descripción del nuevo producto", 10, 19.99, 1);
```

18. `cambiar_stock(\$codProducto, \$unidades)`

Descripción: Actualiza el stock de un producto.

Parámetros:

- `\$codProducto` (int): Código del producto.
- `\$unidades` (int): Nueva cantidad en stock del producto.

Tipo de Retorno: bool. TRUE si la actualización es exitosa, FALSE si hay un error.

Ejemplo:

```
$actualizado = cambiar_stock(1, 5);
```

19. `cargar_usuarios()`

Descripción: Carga todos los usuarios registrados en la base de datos.

Parámetros: Ninguno.

Tipo de Retorno: PDOStatement|false. Un objeto PDOStatement si hay resultados, FALSE si hay un error.

Ejemplo:

```
$usuarios = cargar_usuarios();
```

20. `insertar_usuario(\$nombre, \$contrasena, \$direccion, \$rol)`

Descripción: Inserta un nuevo usuario en la base de datos.

Parámetros:

- ``$nombre`` (string): Nombre del usuario.
- ``$contrasena`` (string): Contraseña del usuario.
- ``$direccion`` (string): Dirección del usuario.
- ``$rol`` (int): Rol del usuario (0 para admin, 1 para cliente).

Tipo de Retorno: bool. TRUE si la inserción es exitosa, FALSE si hay un error.

Ejemplo:

```
$insertado = insertar_usuario("NuevoUsuario", "Contraseña123", "Calle 123", 1);
```

21. ``eliminar_usuario($codCliente)``

Descripción: Elimina un usuario de la base de datos.

Parámetros:

- ``$codCliente`` (int): Código del cliente a eliminar.

Tipo de Retorno: bool. TRUE si la eliminación es exitosa, FALSE si hay un error.

Ejemplo:

```
$eliminado = eliminar_usuario(1);
```

22. ``consultarStockProducto($codProducto)``

Descripción: consulta el stock de la base de datos correspondiente a un producto.

Parámetros:

- ``$codProducto`` (int): Código del producto a consultar.

Tipo de Retorno: bool. TRUE si la actualización ha ido bien, FALSE si hay un error.

Ejemplo:

```
$stockPortátilPotente=consultarStockProducto(1);
```

9. Consideraciones extra para evaluar la aplicación

Las contraseñas han sido encriptadas mediante hash, por lo que para probar la aplicación las contraseñas no se pueden saber mirando la base de datos.

Los administradores tienen de contraseña "0" y para los clientes (las tiendas) su contraseña es el mismo número que le acompaña en el nombre. Por ejemplo: Tienda 1 tiene la contraseña "1".

Se ha encriptado la contraseña y se controla el stock desde antes de hacer el pedido (ya que de otra forma la base de datos podría llegar a tener una cantidad de stock negativa), y el stock solo se cambia en base de datos cuando se confirma el pedido.

En varias funciones se hace una distinción entre que la consulta sql no funcione o que esté vacía. En estas funciones si está vacía no se devuelve false para poder hacer la distinción e imprimir el mensaje adecuado, por ejemplo: en cargar productos de categoría, si no hay productos se imprime el mensaje de que no existen productos para esa categoría.

Se hacen varias comprobaciones de stock: para añadir al carrito se comprueba teniendo en cuenta los datos de stock de la base de datos, y se mira si en el carrito hay unidades para mostrar una cantidad diferente disponible a ese usuario. A la hora de hacer el pedido se comprueba si sigue habiendo stock disponible y que sea mayor que la cantidad de unidades del carrito. Todo ello para evitar cantidades de stock negativas en la base de datos.

En cuanto a las funcionalidades añadidas, las más importantes son que se pueden crear usuarios, categorías y productos, vaciar cesta, se pide contraseña para confirmar pedido, se muestra el pedido, se puede ver el historial de pedidos, revisar cada pedido realizado y cambiar el estado del pedido; y se pueden eliminar usuarios.

Actualmente se puede usar la aplicación sin tener que introducir datos directamente en la base de datos. Sólo haría falta realizar por base de datos la eliminación de un producto o categoría.

En cuanto a bd.php no todas las consultas se realizan mediante prepare. Estas cuestiones, y la eliminación de categorías y productos o que los clientes puedan cambiar su contraseña son las limitaciones o cuestiones a mejorar que tiene mi aplicación.

También he observado que si se le da a retroceder desde la flecha a la izquierda del navegador puede dar problemas tras realizar alguna operación con formulario. Por este motivo algunos formularios que van a ser procesados en el mismo fichero tienen el action completo y otros usan #: estaba buscando cómo funcionaba mejor, sin embargo aún puede dar problemas por lo que recomiendo usar los enlaces de navegación propios de la aplicación.