

Desarrollo Web en entorno servidor

Unidad 4: Modificación de la aplicación de pedidos.

Prueba práctica

Tabla de contenido

•	Ejercicio 1.	2
•	Ejercicio 2.	2
•	Ejercicio 3.	3
•	Ejercicio 4.	4
•	Ejercicio 5.	5
•	Ejercicio 6.	5

La aplicación desarrollada hasta el momento carece de funcionalidades básicas para tener un control sobre los pedidos realizados, enviados o pendientes de enviar. A su vez, tampoco permite trabajar con las categorías y productos definidas inicialmente.

• Ejercicio 1.

En la base de datos `pedidos` crea una nueva tabla denominada `productospendientes`. Esta tabla tiene cuatro claves: CodPend, CodPed, CodProd, UdPend

- CodPend es la clave principal.
- CodPed es una clave foránea, relacionada N a 1 con la clave CodPed de la tabla `pedidos`. Guarda el código de cada uno de los pedidos en caso de que falte stock de algún producto.
- CodProd es una clave foránea, relacionada N a 1, con la clave CodProd de la tabla `productos`. Guarda el producto del que falta stock para cada pedido.
- UdPend es una clave de tipo entero que guarda la información de cuantas unidades faltan de cada producto en cada pedido.

Crea el documento stock.sql con las instrucciones necesarias para crear esta tabla con las relaciones indicadas.

• Ejercicio 2.

Cuando un restaurante tramita un pedido se ha de comprobar si hay suficientes productos en el stock o no. En caso de no haber suficiente, se ha de avisar al usuario de los productos que faltan y se le dará la opción de continuar con el pedido o no.

Modifica la página procesar_pedido.php para que antes de insertar el pedido en la base de datos (línea 18¹) se compruebe si existen suficientes productos en stock.

Para ello, define la función falta_stock(\$carrito) que tiene por argumento el carrito del usuario. La función ha de devolver un array que tenga por claves los códigos de los productos que no están en stock y por valores el número de unidades que faltan. Si todos los productos están en stock, devolverá False.

Si hay productos fuera de stock, en procesar_pedido.php se mostrará una tabla con el número de unidades pedidas de cada producto, el número de unidades que falta de cada uno de ellos, el nombre, la descripción del producto y la categoría a la que pertenece.

Para ello, define la función cargar_sin_stock(\$productos_sin_stock) que recibe como argumento el array obtenido en la función anterior y devuelve un cursor (el resultado de ejecutar una consulta) con los datos necesarios para rellenar la tabla.

Una vez mostrada la tabla anterior, se han de mostrar un botón con el texto “Tramitar pedido”.

Si se pulsa dicho botón, comenzará la inserción del pedido en la base de datos.

¹ \$resul = insertar_pedido(\$_SESSION['carrito'], \$_SESSION['usuario']['CodRes']);

Modifica la función `insertar_pedido($carrito, $codRes)` para que tenga un tercer argumento opcional, cuyo valor por defecto es 0. Este valor se corresponde con el valor de la clave ``Enviado`` de la tabla ``pedidos``.

Si el pedido contiene productos sin stock, entonces se ha de llamar a esta función desde `procesar_pedido.php` con el tercer argumento igual a 1, con lo que en la tabla ``pedidos``, la clave ``Enviado`` valga 1.

La inserción sigue con la anotación en la tabla ``productospendientes`` del número de unidades que faltan de los productos, del código del envío, el código de cada producto, y el código del envío correspondiente.

Para ello, define la función `insertar_sin_stock($envio_pendiente)` que recibe un array con los datos a insertar en la tabla ``productospendientes``.

El proceso termina como las líneas 19 a 34 de `procesar_pedido.php`.

```
if ($resul === FALSE) {
    echo "No se ha podido realizar el pedido<br>";
} else {
    $correo = $_SESSION['usuario']['Correo'];
    echo "Pedido realizado con éxito. Se enviará un correo de confirmación a:
$correo ";
    /* Este bloque comprueba que el correo se ha enviado correctamente con la
extensión PHPMailer
    $conf = enviar_correos($_SESSION['carrito'], $resul,
$correo);
    if($conf!==TRUE){
        echo "Error al enviar: $conf <br>";
    };*/
    //vaciar carrito
    $_SESSION['carrito'] = [];
    if (isset($_COOKIE['carrito'])) {
        setcookie('carrito', '', time() - 3600 * 24);
    }
}
?>
```

• Ejercicio 3.

Los únicos usuarios que pueden controlar el stock faltante y definir nuevos productos y categorías son los responsables de la Sección de Pedidos de la empresa. Para ello, crea un nuevo usuario en la tabla ``restaurantes`` con el rol 2 que no tenga acceso a las opciones de administración.

Para estos usuarios, crea las páginas `modificar_categorias.php` y `modificar_productos.php` accesibles mediante sendos enlaces desde la página `zona_pedidos.php`. Esta última es accesible mediante su enlace correspondiente en la cabecera, que tiene por nombre “Gestión de productos y pedidos”.

La página `modificar_categorias.php` contiene un formulario que permite modificar el nombre y descripción de las categorías. Además, contiene enlaces por cada una de las categorías a los productos de dicha categoría. Haciendo clic en cualquier de ellos se ha de redirigir a `modificar_productos.php` y cargar los productos de dicha categoría. En caso de que se acceda directamente se mostraran todos los productos disponibles.

En la página `modificar_productos.php` se muestra una tabla con toda la información de los productos de la tabla ``productos`` y los botones de modificación correspondientes, al estilo de la página `datosusu.php`.

Ambas páginas también han de tener, a mayores, un formulario específico para añadir nuevas categorías y productos. El formulario para añadir nuevos productos solo será visible si hemos accedido a `modificar_productos.php` a través de `modificar_categorias.php`.

Las funciones a crear para las modificaciones y altas son:

```
actualizar_producto($datos)
actualizar_categoria($datos)
alta_producto(($datos)
alta_categoria(($datos)
```

Todas ellas reciben un array con los datos a modificar o insertar y devuelven TRUE O FALSE si todo ha ido correctamente. Además, todas ellas hacen uso de instrucciones SQL preparadas.

• Ejercicio 4.

Los usuarios con rol 2 también han de poder dar de bajas productos y categorías. Crea la página `"bajas_prod_cat.php"` que permita borrar categorías y productos. Elige la forma que prefieras para que estos usuarios puedan escoger de entre las categorías y productos disponibles y proceder a su borrado.

Para llevar a cabo estas bajas se han de definir dos funciones: `eliminar_productos($codProd)` y `eliminar_categorias($codCat)`. La primera recibe el código del producto a eliminar y la segunda, el código de la categoría a eliminar. Ambas devuelven TRUE si la eliminación ha sido correcta y FALSE en otro caso.

Para ejecutar las instrucciones de eliminación correspondientes usa sentencias SQL preparadas.

`eliminar_categorias($codCat)` ha de eliminar, en primer lugar, los productos asociados a la categoría `$codCat` de la tabla ``productos`` para después, eliminar los datos de `$codCat` de la tabla ``categorias``. Usar una transacción para proceder a este borrado.

Los productos que estén en las tablas ``pedidosproductos`` y ``productospendientes`` no pueden borrarse de las tablas ``productos`` y ``categorias``.

Para poder gestionar estos productos, incorpora una nueva clave a la tabla ``productos`` con el nombre `"descatalogado"`, de tipo entero y valor por defecto 0. Haz lo mismo en la tabla ``categorias``. Para ello crea el documento `descatalogado.sql`

- **Ejercicio 5.**

Modifica el ejercicio 3 para permitir cambiar el valor de la clave `descatalogado`. Si el número es 0 entonces es que no está descatalogado. Si el número es distinto de 0 entonces es que está descatalogado.

En la página de preferencias.php añade la opción de “Ver descatalogados”. Esta opción creará la cookie “descatalogado” y permitirá a los usuarios de rol 2 ver u ocultar estos productos en sus páginas.

Modifica las funciones de la bd.php asociadas a categorias.php y productos.php para que estás páginas no muestren productos ni categorías descatalogados para el resto de usuarios en ningún caso.

- **Ejercicio 6.**

Añadir en zona_pedidos.php un enlace a la página envios_pendientes.php.

En esta página ha de haber un botón que permita a los usuarios con rol 2 generar un documento en formato txt que contenga los datos de los productos sin stock, a saber: Código del restaurante, correo del restaurante, código de pedido, fecha del pedido, código del producto, nombre del producto y unidades pendientes de enviar.

Esta misma información ha de ser visible en esa misma página través de un formulario que permita modificar solo el número de unidades pendientes de enviar.

Si el número de unidades pendientes de enviar se actualiza a 0, entonces el producto asociado al pedido correspondiente se ha de borrar de la tabla `productospendientes`

Si un pedido que ha sido codificado como `Enviado` igual a 1 ya no tiene productos pendientes, entonces se actualiza a `Enviado` igual a 0.

Crea las funciones necesarias en bd.php para llevar a cabo ambas modificaciones.