
Aplicación UPOLOTE

Realizado por Daniel Gomez Reina, Yasmina Torres Elena, Jesus Manuel Tortolero y Mario Gordon

Descripción

Es una aplicación web que gestiona los datos de una tienda llamada UPOLOTE que se dedica a la venta de productos como lotes, la cual tiene más funcionalidades.

La tienda se encarga de registrar productos que permanecen en nuestra base de datos. Puedes acceder a ella como cliente o repartidor, al igual que darte de baja en el caso de cliente. También tramita pedidos, en los que intervienen un cliente y un repartidor.

La aplicación registra los datos de los productos, clientes, repartidores y pedidos, y puede eliminar los datos de un cliente o un pedido.

Los formularios se hacen con carga dinámica, que se realizan desde el fichero index.js. Desde el menú podemos acceder a distintos formularios y listados.

Funcionalidades de actualización de datos y localización de Ajax y formatos de envío/recepción de datos


- Alta producto:
 - ☐ Al pulsar el botón aceptar se desencadena un proceso de validación e inserción de datos.
 - ☐ Valida que los datos están rellenos y escritos correctamente.
 - ☐ Inserta los datos del producto en la base de datos mediante AJAX.
 - ❖ Utiliza el método (\$.post), y realiza el envío de datos al servidor mediante objeto JSON, respuesta JSON.
 - ❖ Escribe en “./altaProducto/altaProducto.php” de forma que inserta un registro en la tabla de productos a través de los datos recibidos. Devolverá un objeto JSON con información sobre la operación de la base de datos.
-

- ☐ Muestra el mensaje de "Alta de producto OK." en el diálogo de mensajes y se cierra si no hay errores.
- Alta cliente
 - ☐ Al pulsar el botón aceptar se desencadena un proceso de validación e inserción de datos.
 - ☐ Valida que los datos están rellenos y escritos correctamente.
 - ☐ Inserta los datos del cliente en la base de datos mediante AJAX.
 - ❖ Utiliza el método (\$.get), y realiza el envío de datos al servidor mediante objeto JSON, respuesta JSON, petición asíncrona/síncrona.
 - ❖ Escribe en "./altaCliente/altaCliente.php" de forma que inserta un registro en la tabla de clientes a través de los datos recibidos. Devolverá un objeto JSON con información sobre la operación de la base de datos.
 - ☐ Muestra el mensaje de "Alta de cliente OK." en el diálogo de mensajes y se cierra si no hay errores.
- Baja cliente
 - ☐ Al pulsar el botón aceptar se desencadena un proceso de validación y eliminación de datos.
 - ☐ Valida que los datos están rellenos y escritos correctamente.
 - ☐ Escribe en "./bajaCliente/bajaCliente.php" de forma que elimina un registro en la tabla de clientes a través de los datos recibidos. Devolverá un objeto JSON con información sobre la operación de la base de datos. Y también escribe en "./bajaCliente/getClientesBaja.php" para rellenar el select.
 - ☐ Elimina los datos del cliente en la base de datos mediante AJAX con JQuery.
 - ☐ Muestra el mensaje de "Baja de cliente OK." en el diálogo de mensajes y se cierra si no hay errores.
- Alta repartidor
 - ☐ Al pulsar el botón aceptar se desencadena un proceso de validación e inserción de datos.
 - ☐ Valida que los datos están rellenos y escritos correctamente.
 - ☐ Inserta los datos del repartidor en la base de datos mediante AJAX.
 - ❖ Utiliza Ajax sin JQuery, método POST. Respuesta JSON.
 - ❖ Escribe en "./altaRepartidor/altaRepartidor.php" de forma que inserta un registro en la tabla de repartidores a través de los datos recibidos. Devolverá un objeto JSON con información sobre la operación de la base de datos.
 - ☐ Muestra el mensaje de "Alta de repartidor OK." en el diálogo de mensajes y se cierra si no hay errores.
- Alta pedido

- ☐ Al pulsar el botón aceptar se desencadena un proceso de validación e inserción de datos.
- ☐ Valida que los datos están rellenos y escritos correctamente.
- ☐ Inserta los datos del pedido en la base de datos mediante AJAX.
 - ❖ Utiliza Ajax sin JQuery, método GET. También usa el localStorage para recuperar datos para usar en selects.
 - ❖ Escribe en `"/altaPedido/altaPedido.php"` de forma que inserta un registro en la tabla de pedidos a través de los datos recibidos. Devolverá un objeto JSON con información sobre la operación de la base de datos. También escribe en `"/altaPedido/getClientes.php"`, `"/altaPedido/getProductos.php"` y `"/altaPedido/getRepartidores.php"` para obtener cada uno.
- ☐ Muestra el mensaje de "Alta de pedido OK." en el diálogo de mensajes y se cierra si no hay errores.
- Cancelar pedido
 - ☐ Al pulsar el botón aceptar se desencadena un proceso de validación y eliminación de datos.
 - ☐ Valida que los datos están rellenos y escritos correctamente.
 - ☐ Escribe en `"/bajaPedido/bajaPedido.php"` de forma que elimina un registro en la tabla de pedidos a través de los datos recibidos. Devolverá un objeto JSON con información sobre la operación de la base de datos. Y también escribe en `"/bajaPedido/getPedidoBaja.php"` para rellenar el select.
 - ☐ Elimina los datos del pedido en la base de datos mediante AJAX con JQuery.
 - ☐ Muestra el mensaje de "Cancelación pedido OK." en el diálogo de mensajes y se cierra si no hay errores.

Funciones para generación de listados

- Mostrar productos
 - ☐ Obtiene un listado de productos.
 - ☐ El listado se muestra en una tabla HTML.
- Mostrar clientes
 - ☐ Obtiene un listado de clientes.
 - ☐ El listado se muestra en una tabla HTML.
- Mostrar repartidores
 - ☐ Obtiene un listado de repartidores.
 - ☐ El listado se muestra en una tabla HTML.
- Mostrar pedidos por fecha

- 
- ☐ Obtiene un listado de pedidos entre dos fechas introducidas por el usuario en el formulario.
 - ☐ Serán fechas válidas.
 - ☐ Al pulsar el botón “Aceptar” realiza una llamada a Ajax que realiza la consulta en la base de datos para obtener los datos de pedidos entre las fechas seleccionadas.
 - ☐ El listado se muestra en una tabla HTML .
 - ☐ La llamada Ajax para obtener los datos del listado tiene las siguientes características:
 - ❖ Usa Ajax, envío de datos al servidor mediante objeto JSON, respuesta JSONP.
 - ❖ Escribe en “./listadoPedidos/listadoPedidos.php” de forma que recibe de entrada dos fechas, realiza la consulta de los pedidos entre dichas fechas y devuelve los datos recuperados en formato JSON empaquetado como JSONP.