Juan Andrés Zapata Giraldo / Juan Pablo Salazar Ceballos

Locality  CTMA SENA

Manual técnico de locality

ÍNDICE DEL MANUAL

[MANUAL TÉCNICO DEL APLICATIVO WEB LOCALITY 2](#_Toc221148941)

[1. REQUERIMIENTOS TÉCNICOS 3](#_Toc736328924)

[1.1 REQUERIMIENTOS SOFTWARE: 3](#_Toc989309443)

[1.2 REQUERIMIENTOS HARDWARE: 3](#_Toc1292961546)

[2. HERRAMIENTAS PARA EL DESARROLLO: 3](#_Toc12983148)

[2.1 Visual Studio Code: 3](#_Toc728165588)

[2.2 XAMPP: 3](#_Toc1023166735)

[2.3 phpMyAdmin: 3](#_Toc901007685)

[2.4 PHP: 4](#_Toc2056560657)

[2.5 Composer: 4](#_Toc1856064473)

[2.6 CSS: 5](#_Toc496265852)

[2.7 JavaScript: 5](#_Toc369970995)

[3. CONEXIÓN DEL APLICATIVO: 6](#_Toc942503686)

[4. CASOS DE USO: 7](#_Toc894214718)

[5. MODELO ENTIDAD RELACIÓN: 7](#_Toc1775395689)

[6. MODULOS O FUNCIONES: 8](#_Toc1008327913)

# **MANUAL TÉCNICO DEL APLICATIVO WEB LOCALITY**

Este manual describe los pasos necesarios para que cualquier persona pueda  
usar correctamente el aplicativo web creado. Es importante tener en cuenta que en el presente manual se hablará de las conexiones de forma local, requisitos, funciones del aplicativo y código de este.

1. **REQUERIMIENTOS TÉCNICOS**
   1. **REQUERIMIENTOS SOFTWARE:**

* Editor de texto
* Base de datos phpMyAdmin
* SO Windows 7 o Superior
  1. **REQUERIMIENTOS HARDWARE:**
* 2gb de memoria RAM
* Quad-Core 1.6Ghz / Ryzen 3 3100
* Red LAN o WAN

1. **HERRAMIENTAS PARA EL DESARROLLO:**
   1. **Visual Studio Code:**

Este es el IDE en el cual se desarrolló el aplicativo web, este nos ofrece diferentes herramientas las cuales simplifican el trabajo de desarrollo, optimizan el tiempo y ayudan de forma considerable a producir el aplicativo.

* 1. **XAMPP:**

Es un paquete de software libre. El cual será útil para gestionar la base de datos del aplicativo, gracias a este se creará un servidor web local para crear la base de datos de la aplicación y ejecutar de manera correcta Locality

* 1. **phpMyAdmin:**

La base de datos de Locality esta creada en phpMyAdmin la cual es una herramienta de software libre escrita en PHP que proporciona una interfaz gráfica de usuario basada en web para administrar y gestionar bases de datos MySQL. Es una herramienta muy popular y ampliamente utilizada para la administración de bases de datos MySQL en esta se almacenará toda la información de clientes, productos, cuentas, ventas, entre otras cosas. Es la parte más importante del aplicativo ya que desde aquí se despliega toda la información que se ve reflejada en el aplicativo.

* 1. **PHP:**

PHP es un lenguaje de programación de código abierto diseñado específicamente para el desarrollo de aplicaciones web dinámicas y sitios web interactivos. PHP es un acrónimo recursivo que significa "PHP: Hypertext Preprocessor". Fue creado en 1994 por Rasmus Lerdorf y ha sido continuamente desarrollado y mejorado por una comunidad de programadores de todo el mundo. PHP se ejecuta en el lado del servidor y se utiliza para generar contenido dinámico y personalizado en función de la interacción del usuario y la información almacenada en una base de datos. Es compatible con una amplia variedad de servidores web y sistemas operativos, lo que lo hace muy popular y ampliamente utilizado en la industria web.

* 1. **Composer:**

Composer es una herramienta de gestión de paquetes de software para PHP que se utiliza para instalar, actualizar y gestionar las dependencias de las bibliotecas de terceros en proyectos de PHP. Con Composer, los desarrolladores pueden definir y especificar las bibliotecas y paquetes que su proyecto necesita y Composer se encargará de descargar e instalar automáticamente todas las dependencias requeridas en la versión correcta. Composer también permite a los desarrolladores cargar y compartir sus propios paquetes de software con la comunidad PHP. En resumen, Composer es una herramienta esencial para los desarrolladores PHP que simplifica y automatiza el proceso de gestión de dependencias y bibliotecas de software en sus proyectos.

* 1. **CSS:**

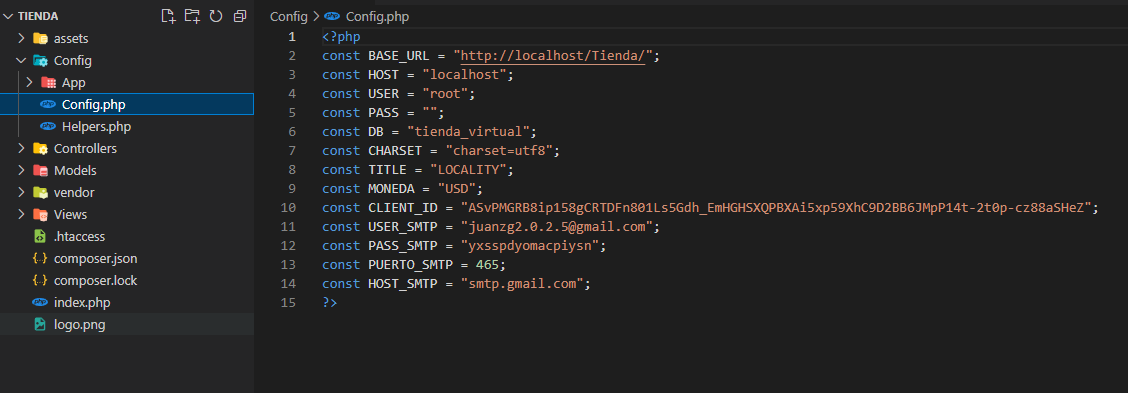
CSS (Cascading Style Sheets) es un lenguaje de diseño gráfico utilizado para definir el estilo visual y la presentación de una página web escrita en HTML o XML. CSS separa el contenido de la presentación y el diseño, lo que permite a los desarrolladores web controlar el estilo visual de una página web sin tener que alterar el contenido. Con CSS, los desarrolladores pueden definir estilos como el color, el tamaño y la posición de los elementos HTML, la tipografía, los efectos visuales, la disposición de la página y otros aspectos visuales. CSS se ha convertido en una herramienta esencial para los desarrolladores web y es ampliamente utilizado en la industria para crear páginas web con un diseño visual atractivo y coherente.

* 1. **JavaScript:**

JavaScript es un lenguaje de programación interpretado de alto nivel que se utiliza principalmente en el desarrollo web para crear aplicaciones web interactivas y dinámicas. Fue creado por Brendan Eich en 1995 y es un lenguaje de programación basado en objetos que se ejecuta en el lado del cliente (navegador web). JavaScript se utiliza para agregar interactividad a las páginas web, como la validación de formularios, la animación de elementos HTML, la creación de efectos visuales, la carga dinámica de contenido y la interacción con el usuario. Con la ayuda de herramientas como Node.js, también se puede usar JavaScript en el lado del servidor. JavaScript se ha convertido en uno de los lenguajes de programación más populares y ampliamente utilizados en la industria web.

1. **CONEXIÓN DEL APLICATIVO:**

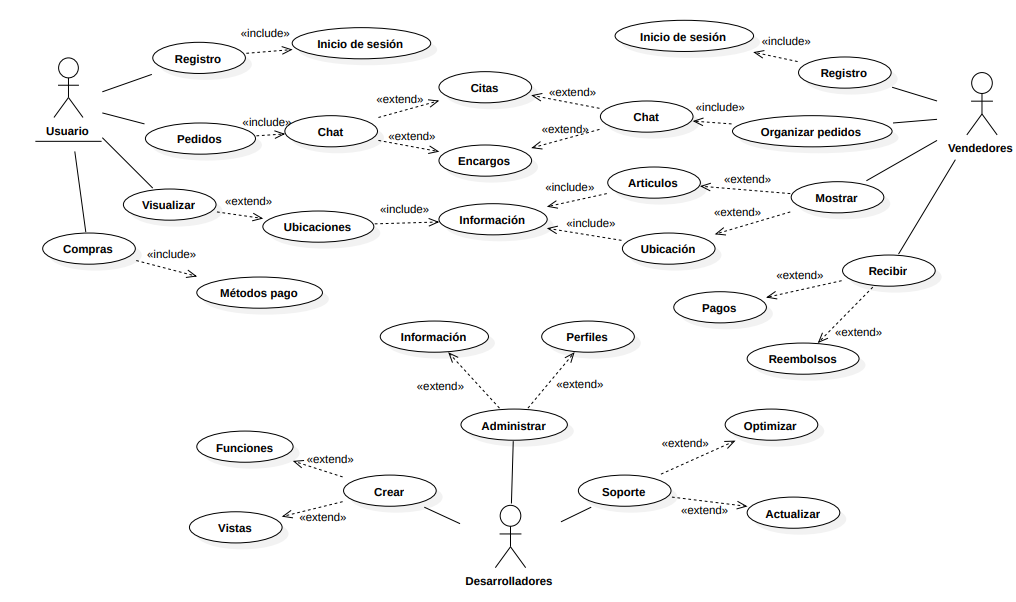
Para la correcta conexión del aplicativo se necesita tener el administrador de bases de datos XAMPP activo, se debe abrir phpMyAdmin en el cual debemos revisar el puerto en el que se ejecutar para hacer una correcta conexión, allí importaremos la base de datos del aplicativo, importación que viene incluida con el repositorio del aplicativo, este es el archivo que se debe importar:

Luego de importar el archivo, se verifica la conexión entre el aplicativo y la base de datos desde el IDE en la carpeta de Config y en el archivo Config.php, así:

Con esto realizado ya se podrá observar Locality de manera correcta. No debe presentar ningún problema, en caso tal estar seguro de que el puerto de XAMPP y phpMyAdmin correspondan con el mismo puerto que se encuentra en el archivo config.php dentro de la carpeta Config.

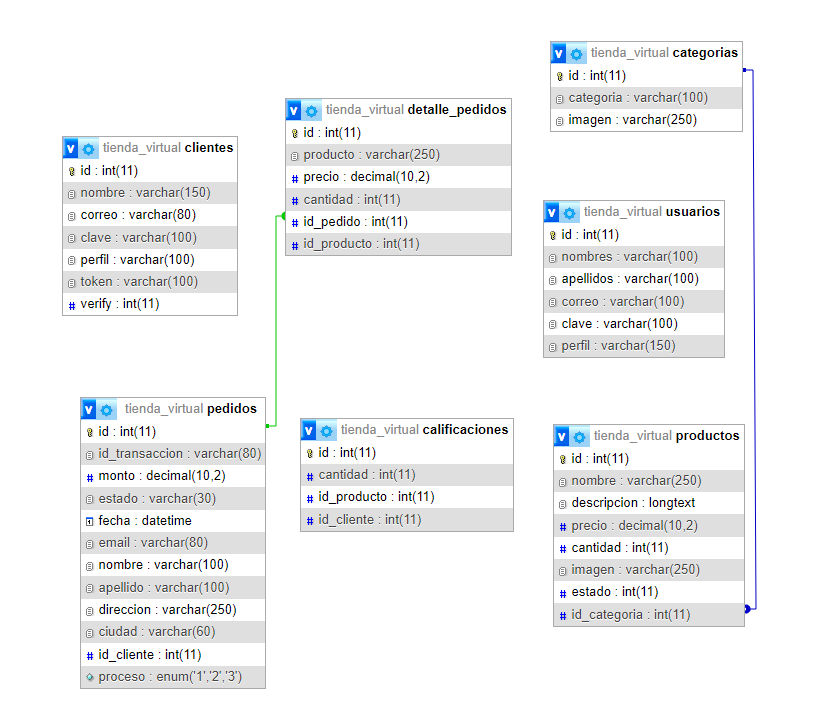
1. **CASOS DE USO:**

En este se decidió que en los casos de uso tendrían actividades o funciones bastante similares tanto los clientes como los vendedores, esto se tomó desde diferentes actividades que estos podrían realizar, a pesar de que el vendedor tiene un mayor control que el cliente, ambos compartirán algunas funciones

**** Desde el punto de vista del desarrollador lo más importante es implementar mayores funciones para vendedores ya que estos deben tener la posibilidad de publicar y ofrecer sus artículos o servicios en este aplicativo.

Se toma en cuenta en este gráfico la importancia del acompañamiento constante por parte de los desarrolladores para crear, administrar y realizarle su respectivo mantenimiento a Locality, al ser un aplicativo donde se realizan constantes actualizaciones siempre existirá la posibilidad de que ocurra algún fallo durante la transferencia de datos o cosas por el estilo.

1. **MODELO ENTIDAD RELACIÓN:**

Aquí se presenta el modelo entidad relación de Locality:

El modelo entidad relación es una técnica para diseñar bases de datos relacionales que utiliza entidades para representar objetos del mundo real y relaciones para describir cómo se conectan estos objetos entre sí. Se utiliza para visualizar y planificar la estructura de una base de datos y ayuda a los diseñadores a entender cómo se organizan y relacionan los datos.

1. **MODULOS O FUNCIONES:**

**6.1 CÓMO FUNCIONA EL CÓDIGO:**

El código esta creado bajo el sistema modelo, vista y controlador, ya que es la forma más práctica para trabajar en un proyecto como este y es simple de entender cómo funciona el código con tal solo verlo, con esto se espera que el desarrollador pueda comprender fácilmente la estructura del aplicativo.

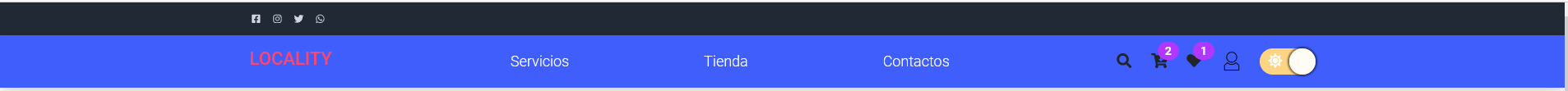
Para trabajar en el código es importante saber que el aplicativo ésta creado gracias a los lenguajes de PHP y JavaScript, los cuales ayudan a traer tanto información en la base de datos como crear funciones para las diferentes ventanas emergentes, verificaciones de pago, creación de listas, etc.

En el código se encontrarán los archivos PHP y JavaScript relacionados con su nombre, cada vista o archivo PHP tiene su propio archivo Java para crear las funciones necesarias para esa vista, así:

**6.2 MODULOS DEL APLICATIVO:**

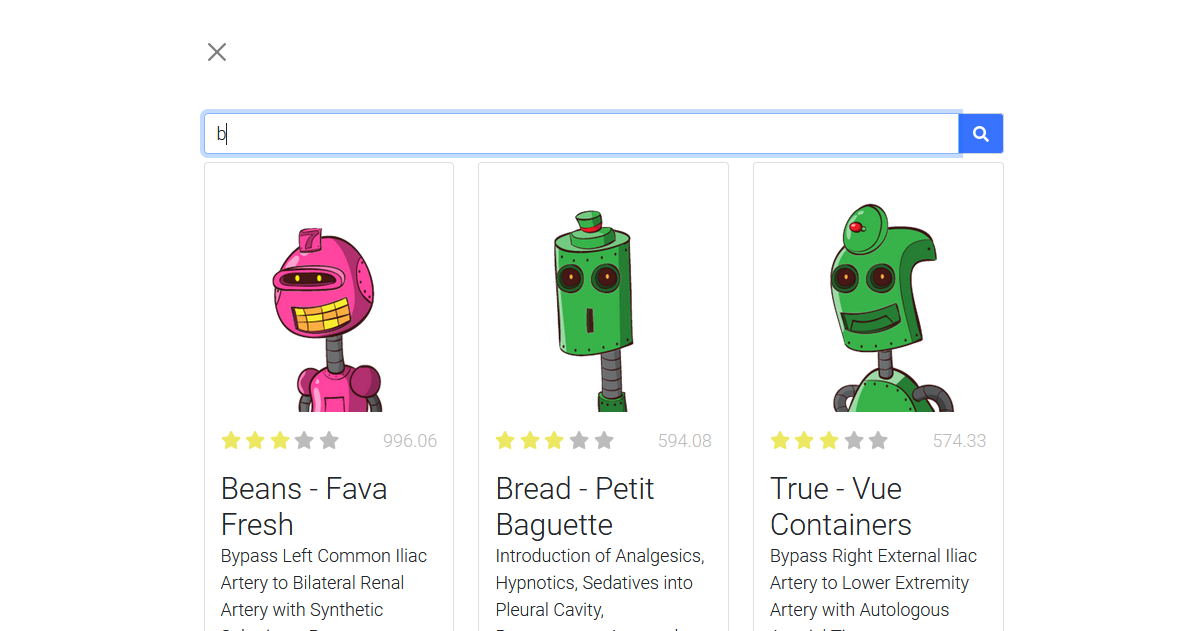
Aquí se muestra para que sirve cada uno de los módulos del aplicativo, cabe resaltar que los módulos “Servicios” y “Contactos” realmente no tienen ninguna función especial por lo que se omitirán estos, en “Servicios” se encuentra información sobre la empresa como su visión y misión, algunos anuncios que se publican allí e información sobre la tienda, en “Contactos” hay un formulario con dos inputs, uno para ingresar el correo y el otro es un mensaje que el usuario desee enviarle a la empresa. Dicho esto, a continuación, se mostrarán los módulos importantes del aplicativo:

**6.2.1 MÓDULO BARRA DE NAVEGACIÓN:**

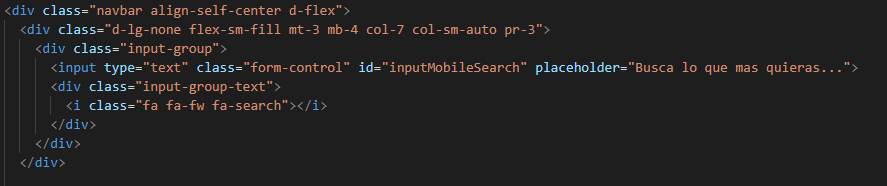


La barra de navegación es parte importante del aplicativo, pues en ella existe la posibilidad encontrar diversas funciones como una búsqueda integrada en la barra de navegación la cual ayudará a filtrar los productos para obtener con mayor facilidad lo que el usuario necesita, también está la lista de deseados para que el usuario visualice aquellos productos que añadió a dicha lista, ahí se encuentra el carrito de compras y el perfil, en ambas se puede observar los productos que el usuario desea ordenar.

Vista de barra de búsqueda:

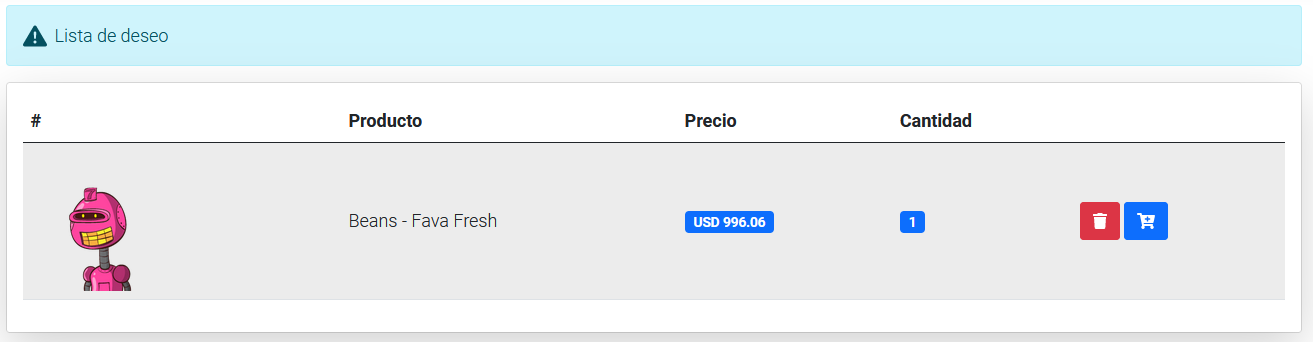


Código búsqueda:



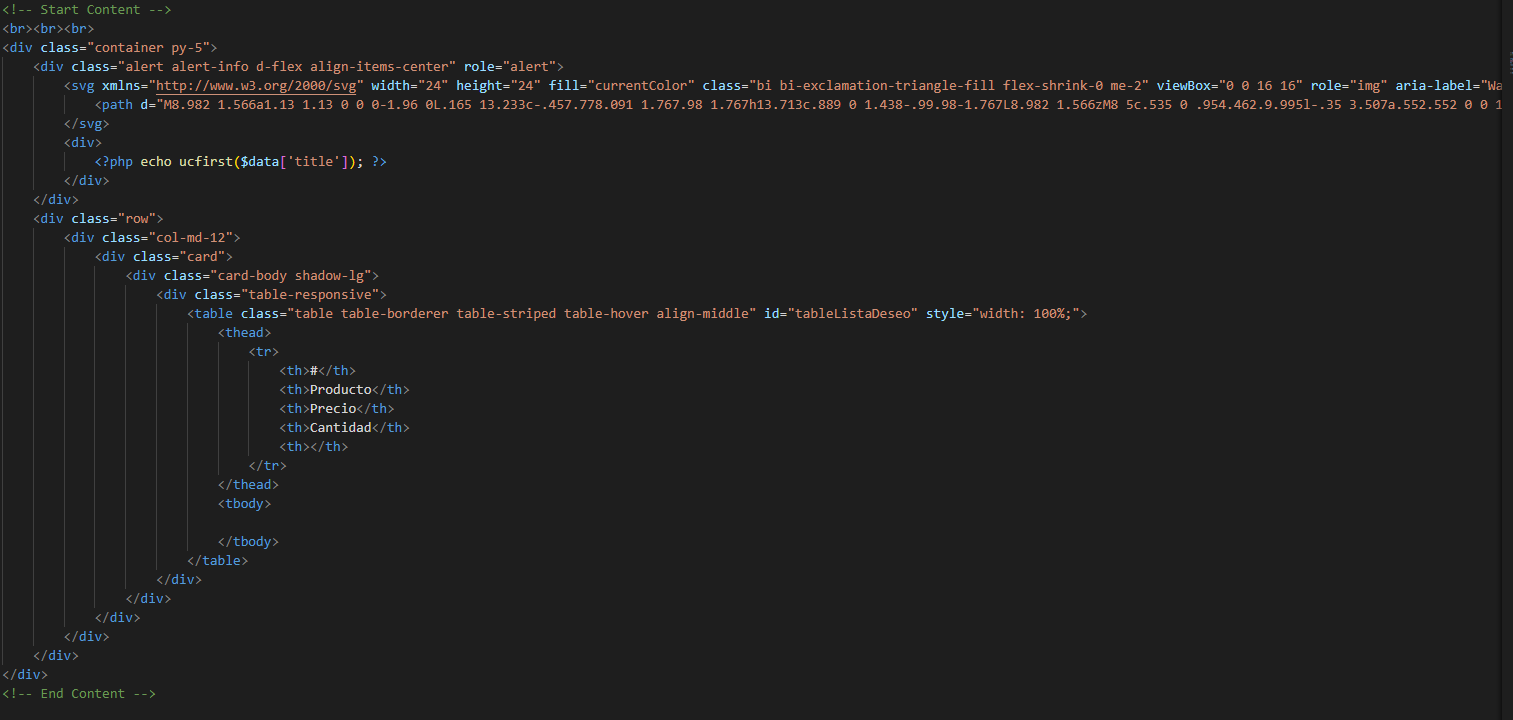
Aquí a medida que introducimos un nombre irá filtrando los productos

Vista lista de Deseados:

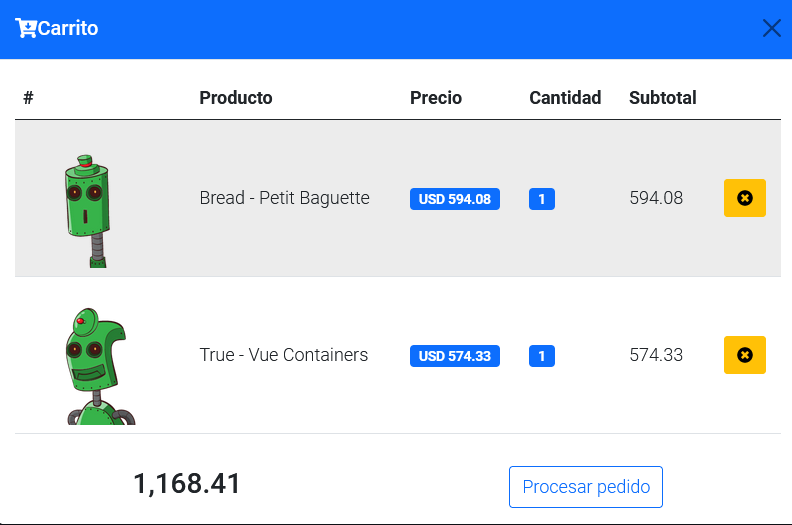


En la lista de Deseados el usuario podrá observar, añadir al carrito y eliminar de esta lista los productos que estén en ella.

Código lista de deseos:

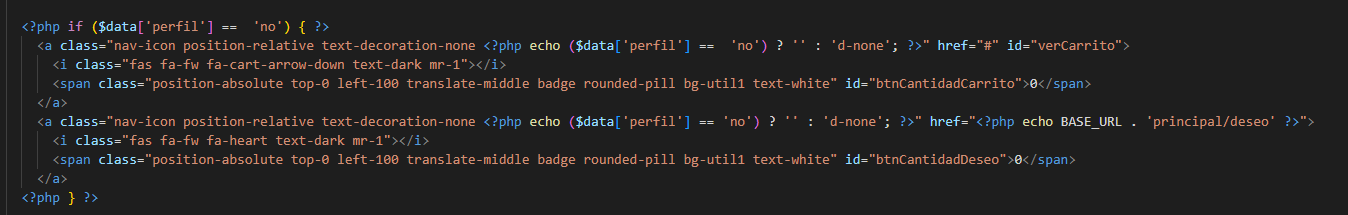


Vista de Carrito:

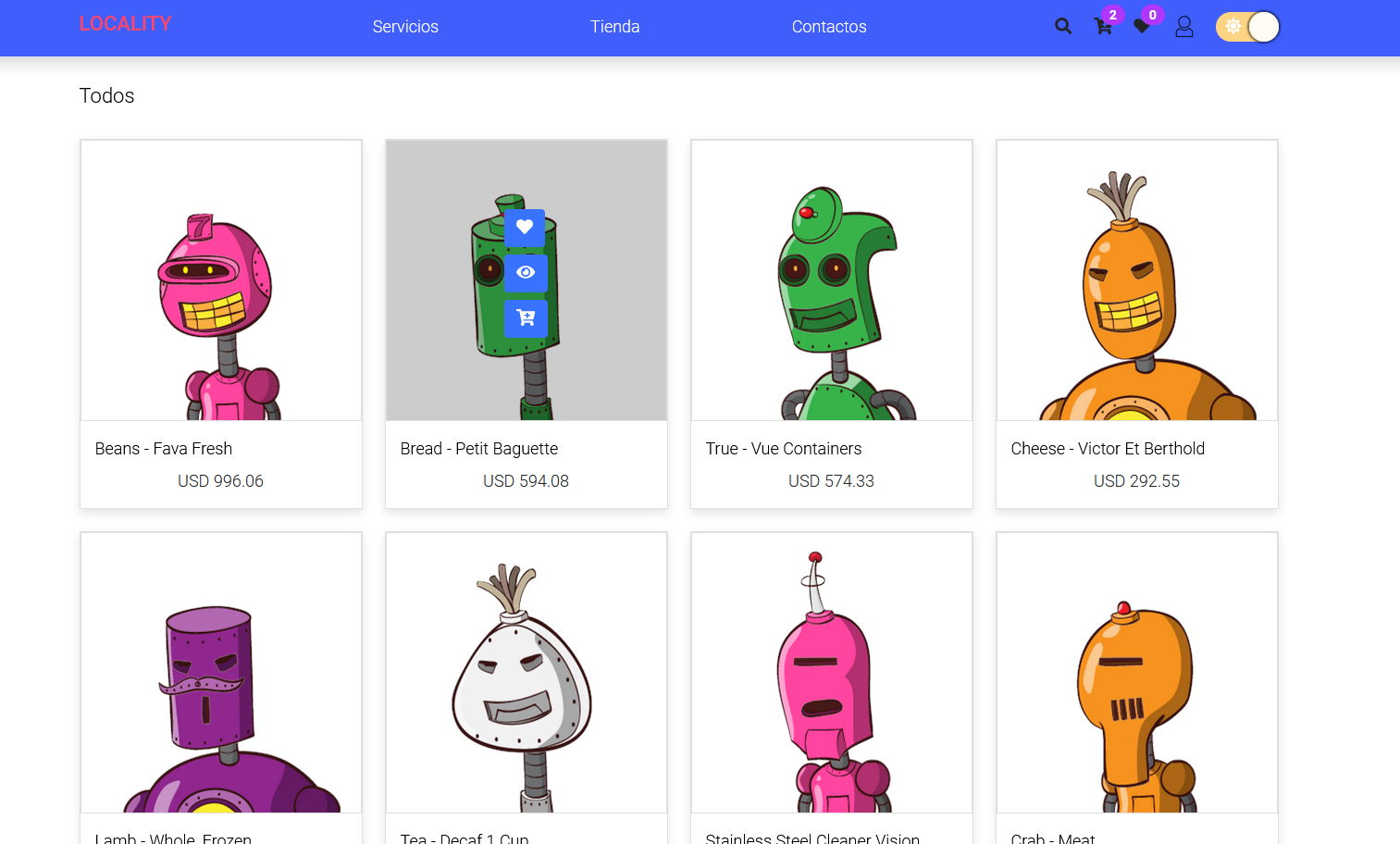


En la vista de carrito el usuario observará todos los productos que desea comprar, este puede retractarse eliminándolos con el botón amarillo que está al lado derecho hasta el final de cada producto y puede continuar con el pago de sus productos desde el botón procesar pedido.

Código carrito:



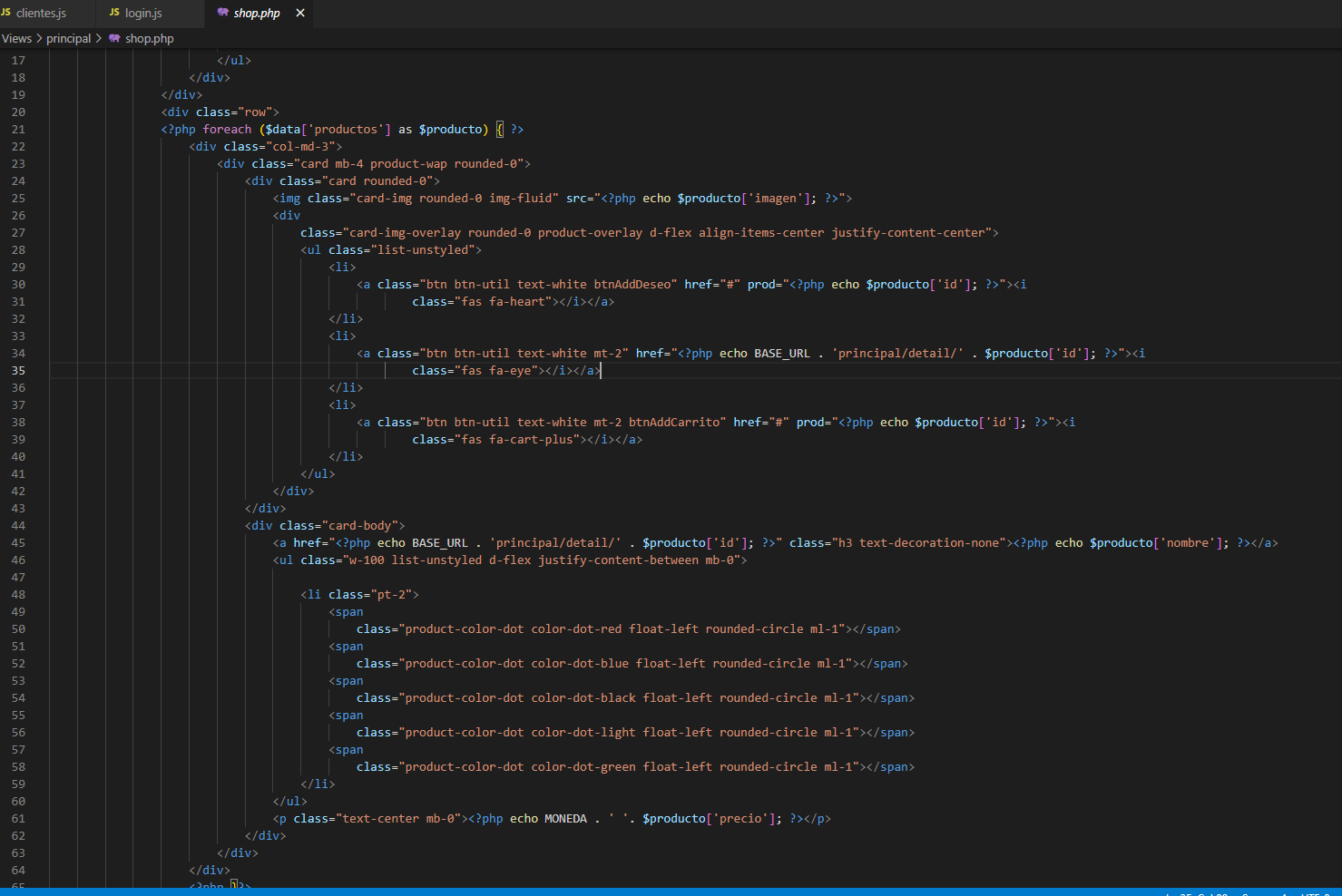
**6.2.2 MÓDULO DE TIENDA:**



En este módulo se encuentran todos los artículos disponibles para el consumidor final, aquí podemos observar los ítems que tenemos en la tienda y con los cuales podremos interactuar de la siguiente manera:

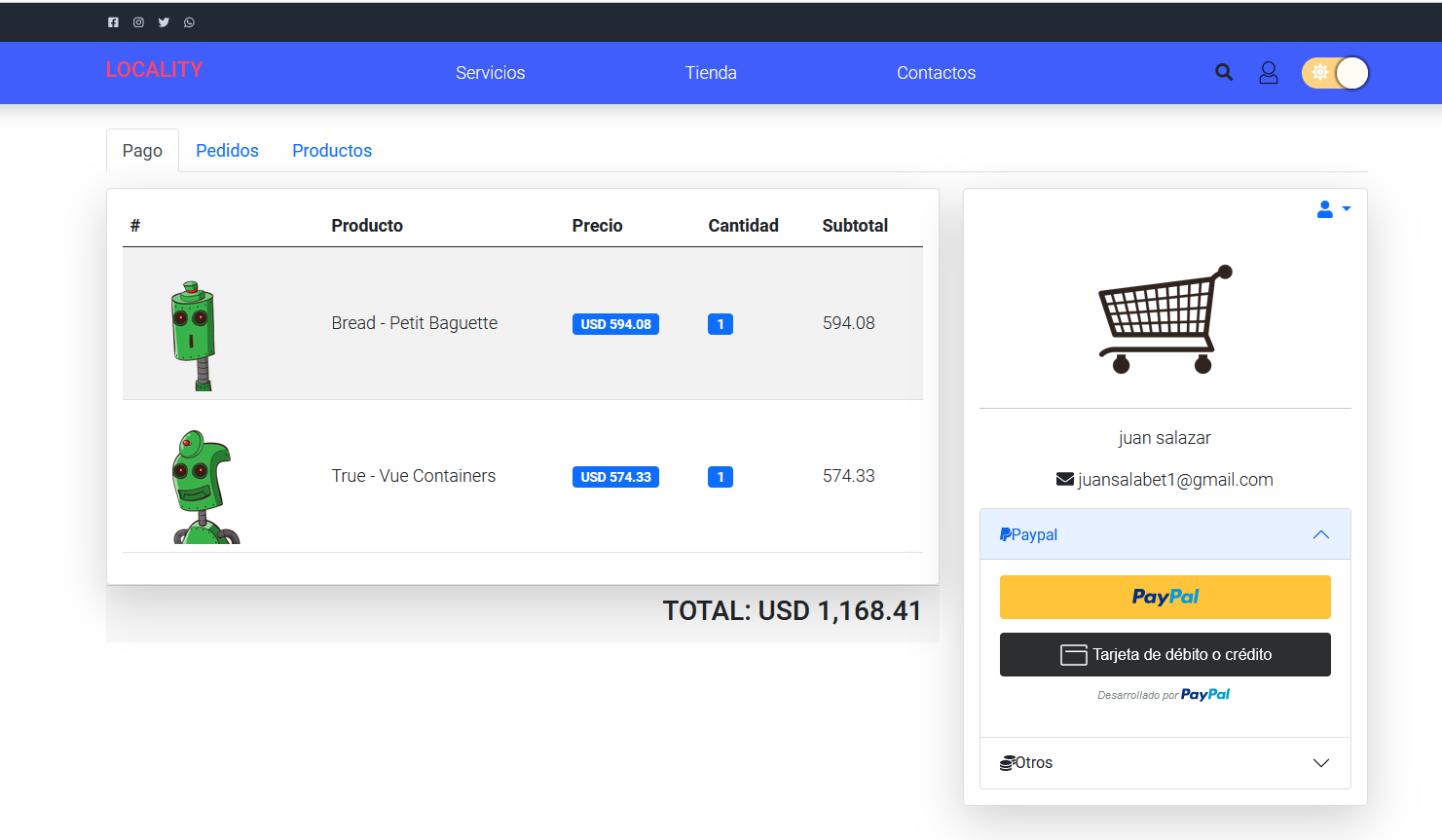


En esta se tienen tres opciones de arriba hacia abajo, la primera es para agregar a una lista de deseados, la segunda para tener más información sobre el producto y la tercera para agregar al carrito, la información que se muestra en la tarjeta como lo son el nombre, la imagen y el precio vienen de la base de datos y son traídas a la vista por medio de PHP así:



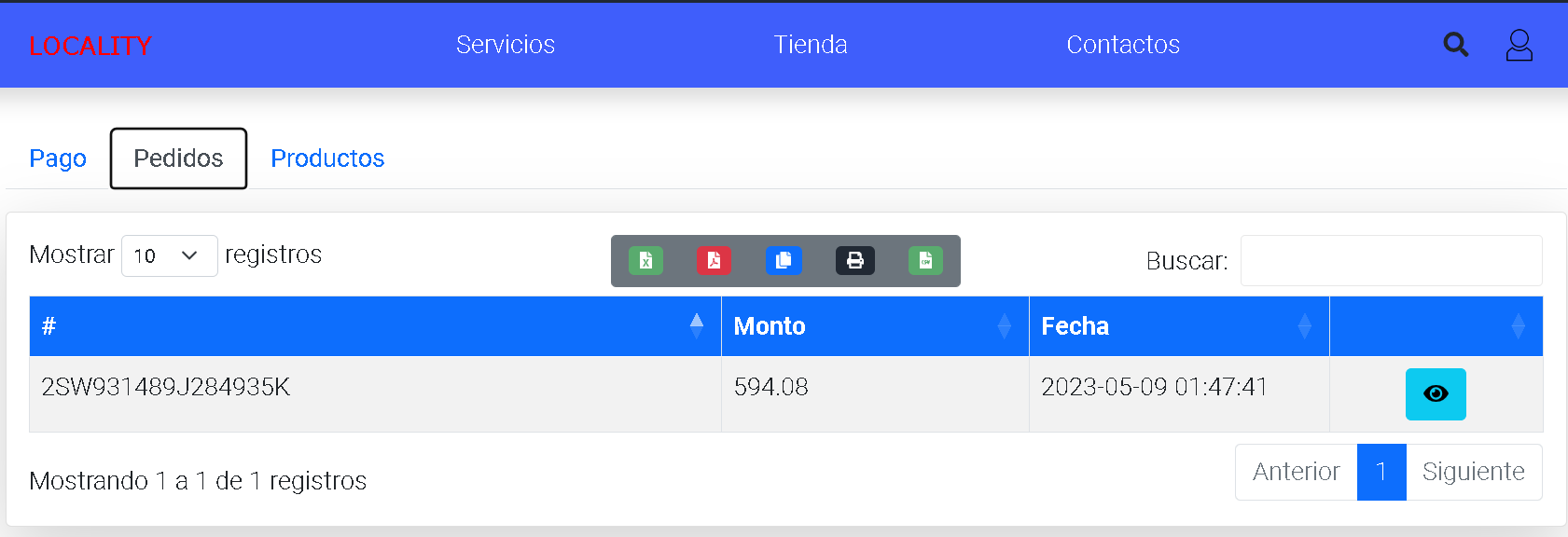
Nótese que hay un foreach para que por cada producto que exista en la base de datos se cree una tarjeta con la información del mismo y donde se le añaden las 3 opciones con las que se interactúa con el producto.

**6.2.3 MÓDULO DE COMPRA:**



Aquí se mostrarán todos los articulos que el usuario desee comprar, formas de pago y otras opciones como visualizar el estado de los pedidos y mirar el código de los productos

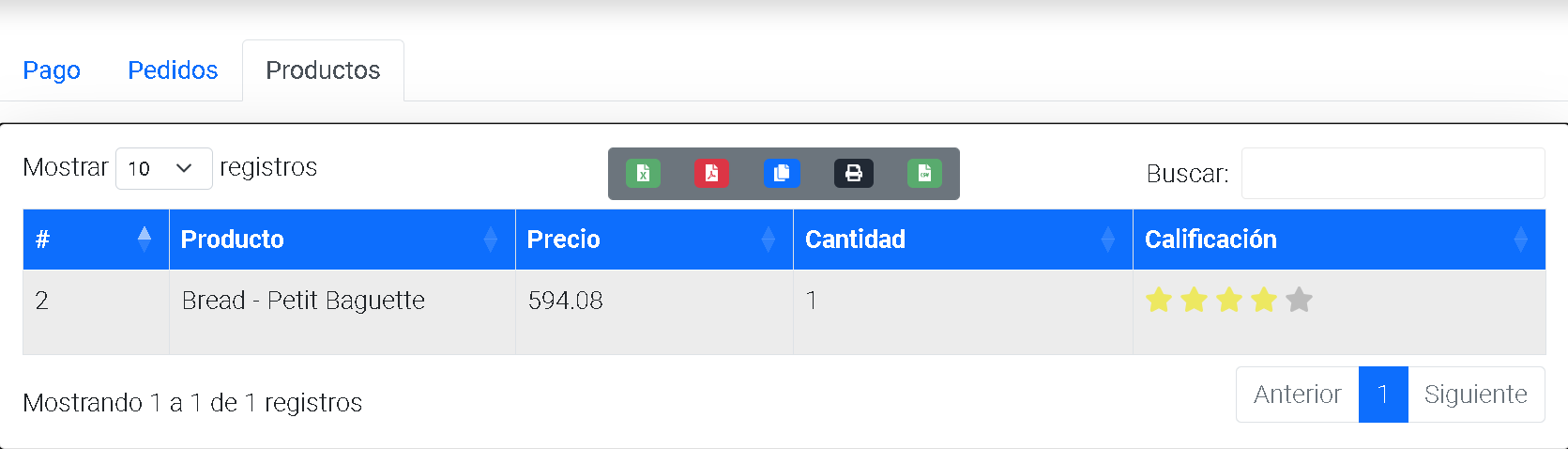
Vista Pedidos:



Se abre la pestaña de Pedidos para lograr observar esta pantalla en la cual el usuario podrá mirar el código de su pedido, el costo y la fecha en la cual realizo el mismo.

Toda esta información se guarda en la base de datos

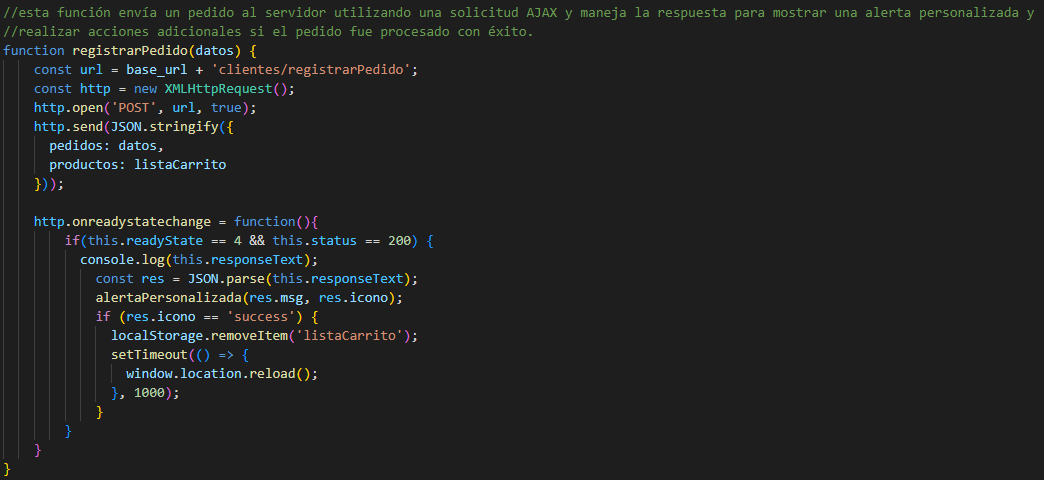
Vista Productos:



En la pestaña de Productos, se observa el nombre del producto, el precio, la cantidad ordenada y además se puede calificar dicho producto, esta información se almacena en la base de datos y se actualiza a medida que se hagan cambios en dicho producto.

**6.3 FUNCIONES A EXPLICAR:**

**6.3.1 FUNCIÓN REGISTRAR PEDIDO:**



La función `registrarPedido` se utiliza para enviar un pedido al servidor utilizando una solicitud AJAX.

Primero, se define la URL a la que se enviará el pedido a través de una variable `url` que concatena la URL base con la ruta específica del método que maneja el registro de pedidos (`clientes/registrarPedido`).

Luego, se crea una instancia del objeto `XMLHttpRequest()` y se utiliza para abrir una conexión `POST` a la URL especificada. Los datos del pedido se envían en el cuerpo de la solicitud como un objeto JSON que incluye la información del pedido (`datos`) y la lista de productos en el carrito (`listaCarrito`), convertidos en formato JSON mediante el método `JSON.stringify()`.

La función `onreadystatechange` se define para manejar la respuesta de la solicitud AJAX cuando cambia su estado. Si el estado es 4 (la solicitud ha sido procesada y la respuesta está lista) y el código de estado HTTP es 200 (OK), se realizan las siguientes acciones:

- Imprimir la respuesta de la solicitud en la consola utilizando el método `console.log ()`.

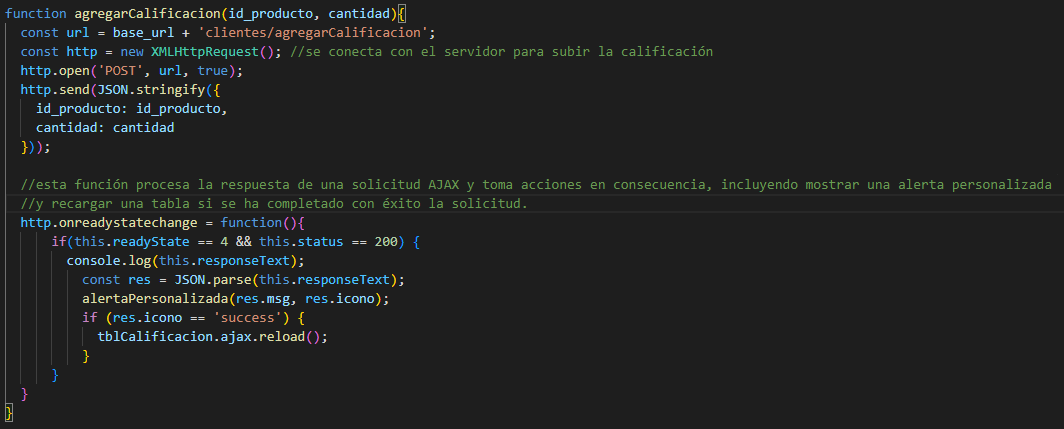
- Analizar la respuesta de la solicitud como un objeto JSON utilizando el método `JSON.parse() `.

- Llamar a una función `alertaPersonalizada()` con los valores `msg` e `icono` del objeto JSON como argumentos para mostrar un mensaje de alerta personalizado.

- Si el valor de `icono` es 'success', se eliminará el carrito de compras almacenado en el `localStorage` y se recargará la página después de 1 segundo utilizando el método `setTimeout()`.

En resumen, esta función envía un pedido al servidor utilizando una solicitud AJAX y maneja la respuesta para mostrar una alerta personalizada y realizar acciones adicionales si el pedido fue procesado con éxito.

**6.3.2 FUNCIÓN AGREGAR CALIFICACIÓN:**



La función `agregarCalificacion` se utiliza para enviar una calificación al servidor para un producto específico a través de una solicitud AJAX utilizando el método HTTP POST.

Primero, se construye la URL de la solicitud a través de la concatenación de la URL base de la aplicación y la ruta específica del método que maneja el registro de calificaciones en el servidor.

Luego, se crea una instancia del objeto `XMLHttpRequest` que se utilizará para realizar la solicitud AJAX. Se llama al método `open` de la instancia de `XMLHttpRequest` y se establece la configuración de la solicitud como `POST`, la URL de la solicitud, y `true` para indicar que la solicitud se realizará de forma asíncrona.

Se envía la solicitud utilizando el método `send` de la instancia de `XMLHttpRequest` y se proporcionan los datos de la calificación en formato JSON como un objeto con las propiedades `id\_producto` y `cantidad`.

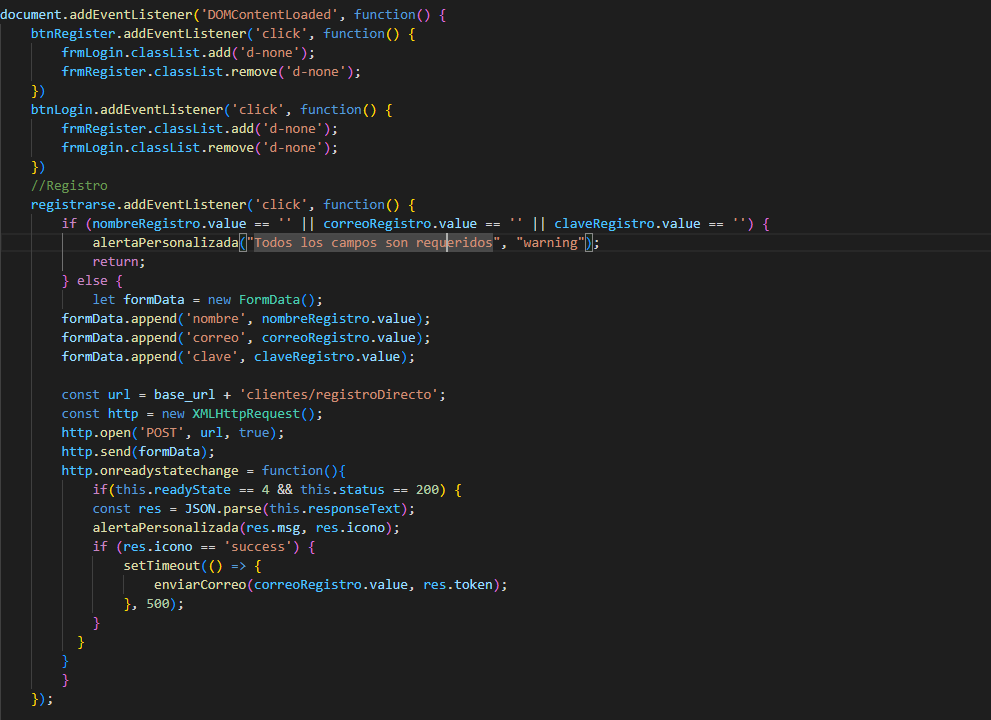
Se define una función para manejar la respuesta de la solicitud AJAX, utilizando el evento `onreadystatechange` de la instancia de `XMLHttpRequest`. En la función manejadora, se verifica si el estado de la solicitud es `4` y si el código de estado HTTP es `200`, lo que indica que la solicitud se ha procesado correctamente.

Si la solicitud se procesa correctamente, se analiza la respuesta de la solicitud como un objeto JSON utilizando el método `JSON.parse`. Luego se llama a la función `alertaPersonalizada` con los valores `msg` y `icono` del objeto JSON como argumentos, para mostrar una alerta personalizada en función de la respuesta del servidor.

Si el valor de `icono` en la respuesta del servidor es `'success'`, se recarga la tabla que muestra las calificaciones de los productos utilizando el método `ajax.reload() ` de la biblioteca DataTables.

En resumen, la función `agregarCalificacion` envía una calificación al servidor para un producto específico mediante una solicitud AJAX y maneja la respuesta para mostrar una alerta personalizada y actualizar la tabla de calificaciones si la solicitud se procesa correctamente.

**6.3.3 FUNCIÓN PRINCIPAL DE REGISTRO Y LOGIN:**



La función es un controlador de eventos para el evento 'DOMContentLoaded' del objeto document. Cuando se carga la página, el código ejecuta una serie de acciones en respuesta a los clics del usuario en los botones del formulario de registro e inicio de sesión.

En particular, el código hace lo siguiente:

1. Agrega un controlador de eventos al botón 'btnRegister' para que cuando se haga clic en él, se oculte el formulario de inicio de sesión y se muestre el formulario de registro.

2. Agrega un controlador de eventos al botón 'btnLogin' para que cuando se haga clic en él, se oculte el formulario de registro y se muestre el formulario de inicio de sesión.

3. Agrega un controlador de eventos al botón 'registrarse' para que cuando se haga clic en él, se verifique que los campos 'nombreRegistro', 'correoRegistro' y 'claveRegistro' no estén vacíos. Si se detecta un campo vacío, se muestra una alerta al usuario y se detiene la ejecución del código. Si no hay campos vacíos, se crea un objeto FormData que contiene los valores de los campos de registro. Luego se envía una solicitud AJAX a través de XMLHttpRequest para enviar los datos del formulario al servidor y registrar un nuevo usuario. Si la solicitud se completa con éxito, se muestra una alerta personalizada y se envía un correo electrónico de confirmación al usuario registrado.

En resumen, la función gestiona la lógica de registro de usuarios en un sitio web, mediante el uso de solicitudes AJAX y eventos del DOM.

**6.3.4 FUNCIÓN LOGIN:**



Esta función está asociada a un botón de inicio de sesión y se ejecuta cuando se hace clic en él. En primer lugar, comprueba si los campos de correo electrónico y contraseña están vacíos y si lo están, muestra una alerta personalizada con un mensaje de advertencia y devuelve el control.

Si los campos no están vacíos, crea un objeto FormData que se utilizará para enviar los datos de inicio de sesión al servidor. Luego, establece la URL a la que se enviará la solicitud AJAX y crea un nuevo objeto XMLHttpRequest.

La solicitud AJAX se envía al servidor y, si se completa correctamente, la función de devolución de llamada se ejecutará. La respuesta del servidor se convierte en un objeto JSON y se muestra una alerta personalizada con el mensaje y el icono correspondientes.

Si el icono es 'success', la función de tiempo de espera se ejecutará después de un segundo, lo que provocará que la página se recargue y se actualice con la sesión iniciada del usuario.