

**Kenia Sarahi Zamora Marroquín 1815310**  
**Briseidy Montserrat Basoria Treviño 1896734**

## Propuesta Api de Terceros

Para nuestra aplicación nos gustaría usar el Api de WooCommerce, es un api que nos proporcionará herramientas avanzadas de gestión de productos, inventario y ventas.

Las características que consideramos relevantes sobre ella se enlistan a continuación:

- Gestión avanzada de productos: Permite a los usuarios crear, actualizar, eliminar y obtener detalles completos de los productos de su tienda. Esto incluiría la capacidad de administrar atributos, variaciones, imágenes y categorías de productos de forma sencilla.
- Gestión de inventario: Ofrece funcionalidades avanzadas de gestión de inventario, como ajustes de inventario, seguimiento de existencias, notificaciones de existencias bajas y sincronización de inventario en tiempo real.
- Integraciones con servicios de envío y pago: Facilita la integración con servicios de envío y pago populares para automatizar el proceso de cumplimiento de pedidos y mejorar la experiencia del cliente.
- Seguridad y autenticación: Implementa medidas sólidas de seguridad y autenticación para proteger los datos confidenciales de los clientes y garantizar la integridad de la plataforma.
- Escalabilidad y rendimiento: Diseñada para ser altamente escalable y capaz de manejar un gran volumen de transacciones y operaciones simultáneas sin comprometer el rendimiento.

# Función Ajax

Nuestra propuesta es en la sección de la página principal, usaremos una función Ajax para poder cargar productos destacados, traídos por el servidor sin necesidad de recargarlo cada vez que actualizamos la página.

## Cómo lo implementaremos:

Crear un archivo PHP, por ejemplo, `productos_destacados.php`, que se encargue de manejar la solicitud AJAX y devolver los datos de los productos destacados en formato JSON. Este archivo podría realizar consultas a la base de datos para obtener los productos destacados.

En la sección de los productos destacados, escribir un script AJAX que haga una solicitud al archivo PHP creado en el paso anterior y maneje la respuesta para mostrar los productos en la página sin recargar.

Una vez que el script AJAX reciba la respuesta del servidor con los datos de los productos destacados, usamos JavaScript para mostrar los productos en la página. Esto podría ser mediante la manipulación del DOM para agregar elementos HTML que representen los productos.

Y a continuación ponemos un ejemplo de cómo podría ser aplicado en nuestro proyecto:

```
// Script AJAX para cargar los productos destacados
function cargarProductosDestacados() {
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'productos_destacados.php', true);
    xhr.onreadystatechange = function() {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            if (xhr.status === 200) {
                // Convertir la respuesta JSON en un objeto JavaScript
                var productos = JSON.parse(xhr.responseText);
                // Lógica para mostrar los productos en la página
                mostrarProductos(productos);
            } else {
                // Manejar errores de la solicitud AJAX
                console.error('Error al cargar los productos destacados');
            }
        }
    };
    xhr.send();
}

// Función para mostrar los productos en la página
function mostrarProductos(productos) {
```

```
// Aquí puedes usar JavaScript para manipular el DOM y mostrar los productos en la
// página
}

// Llamar a la función para cargar los productos cuando la página se cargue
window.onload = function() {
    cargarProductosDestacados();
};
```