

# Memoria Completa de Ampliaciones del E-commerce

VÍCTOR CARAZO CONTRERAS

## 1. Introducción

Este documento recoge de forma completa todas las ampliaciones desarrolladas en el proyecto de e-commerce:

- Sistema de recomendaciones inteligentes (Prototipo 1)
- Constructor de productos personalizados (Prototipo 2)
- Panel de usuario con actividad y estadísticas en tiempo real (Prototipo 3)

Además, se incluyen mejoras avanzadas no solicitadas, como modo oscuro glassmorphism, botones de depuración, sistema real de tracking y gráfica animada sin librerías.

## Prototipo 1 – Sistema de Recomendaciones Inteligentes (AJAX + API externa)

Este módulo genera recomendaciones dinámicas en función de búsquedas recientes del usuario.

### Justificación de diseño

El objetivo era mejorar la experiencia de usuario ofreciendo recomendaciones dinámicas sin recargar la página. Para ello se diseñó un módulo visual ligero, integrado dentro del catálogo, que permite:

- Mostrar sugerencias basadas en búsquedas recientes.
- Ofrecer filtros rápidos y accesibles.
- Mantener una interfaz cohesiva con el resto del e-commerce.

Se priorizó un diseño limpio, modular y responsive, compatible con pantallas pequeñas, cumpliendo los criterios de accesibilidad y rendimiento.

## Arquitectura usada (MVP)

El módulo se estructuró en tres capas:

### Model

- Obtiene datos de la API externa simulada.
- Filtra y prepara recomendaciones.
- Gestiona búsquedas recientes almacenadas en localStorage.

### View

- Renderiza tarjetas de recomendación.
- Gestiona filtros interactivos.
- Actualiza los resultados sin recargar la página.

### Presenter/Controller

- Escucha cambios del usuario.
- Llama al Model y ordena actualizaciones.
- Orquesta la comunicación entre componentes.

El MVP permite desacoplar totalmente la UI de la lógica, facilitando extensiones futuras.

## Descripción del flujo AJAX

1. El usuario introduce una búsqueda o selecciona un filtro.
2. El Presenter ejecuta una llamada AJAX (fetch) a una API de productos.
3. La API devuelve un JSON con coincidencias.
4. El Model procesa los datos (normaliza, filtra, ordena).
5. El View reconstruye la interfaz con las tarjetas actualizadas.
6. Todo ocurre sin recargar la página.

## Problemas encontrados y soluciones

Problema	Solución aplicada
La API externa tardaba o fallaba.	Se implementó <code>try/catch</code> y mensajes de estado (cargando, error).

<b>Problema</b>	<b>Solución aplicada</b>
Búsquedas duplicadas generaban ruido.	Se añadió un sistema de cache (localStorage) con limpieza automática.
Algunas imágenes no cargaban.	Se agregó un placeholder estandarizado.
En móviles, las tarjetas quedaban desordenadas.	Se usó grid responsivo ( <code>repeat(auto-fill, minmax())</code> ).

## Prototipo 2 – Constructor de Productos Personalizados

### Justificación de diseño

El constructor debía ser intuitivo y visual:

- Vista previa instantánea del producto.
- Opciones interactivas (color, tamaño, accesorios...).
- Validación en tiempo real para evitar combinaciones inválidas.
- Integración nativa con el carrito.

Se optó por un diseño de “panel lateral + vista previa” para maximizar la claridad, especialmente en móviles.

## Arquitectura usada (MVP)

### Model

- Gestiona el estado del producto configurado.
- Valida combinaciones disponibles.
- Exporta la configuración en JSON estándar para el carrito.

### View

- Renderiza la vista previa.
- Muestra los errores de validación en tiempo real.
- Sincroniza los controles de color/tamaño/extra.

## Presenter/Controller

- Escucha cambios en la UI.
- Llama al Model para validar y actualizar.
- Emite un `CustomEvent("add-to-cart")` para integrarse con el carrito global.

## Descripción del flujo AJAX

Aunque es un módulo principalmente de UI, usa AJAX para:

1. Validar disponibilidad de combinaciones simulando una consulta al servidor.
2. Enviar/recibir configuraciones válidas en formato JSON.
3. Permitir que el carrito reciba productos personalizados mediante un evento.

## Problemas encontrados y soluciones

Problema	Solución aplicada
Las combinaciones inválidas no se detectaban.	Se implementó un validador dinámico con mensajes claros bajo los controles.
El producto personalizado no se añadía al carrito.	Se creó un <code>CustomEvent("add-to-cart")</code> para integrarlo en el flujo del CartController.
Inconsistencias entre la vista previa y la selección.	Se sincronizó siempre vista ↔ modelo con actualizaciones dobles.
Precios mal calculados.	Se añadió un sistema modular de reglas (precio base + extras).

## Prototipo 3 – Panel de Usuario con Actividad y Estadísticas

### Justificación de diseño

El panel debía mostrar información compleja de forma visual y accesible:

- Tarjetas resumen (compras, búsquedas, favoritos).
- Gráfica de actividad diaria.
- Lista de eventos recientes.
- Filtros avanzados (tipo + rango temporal).

Se diseñó un estilo tipo Apple/iOS para mejorar la claridad y sensación de profesionalidad.

Además, se incorporó **tema oscuro con glassmorphism**, mejorando notablemente la UX.

### Arquitectura usada (MVP)

#### Model

- Carga eventos desde:
  - JSON estático (AJAX real).
  - Actividad REAL del usuario (localStorage).
- Normaliza, filtra y agrupa datos.
- Genera datos para la gráfica.

#### View

- Renderiza resumen, gráfica y lista.
- Muestra/oculta estados (loading, vacío).
- Gestiona filtros accesibles.

#### Presenter/Controller

- Inicializa el módulo.
- Fusiona eventos reales + estáticos.
- Actualiza la vista ante cambios de filtros o eventos nuevos.
- Gestiona botones adicionales:
  - generar actividad de prueba,
  - borrar actividad.

## Descripción del flujo AJAX

1. El Controller ejecuta `fetch("/static/data/user_activity.json")`.
2. El servidor devuelve un JSON estructurado.
3. El Model normaliza los eventos.
4. Se mezclan con la actividad real (grabada por `logActivity()`).
5. La vista se actualiza:
  - o tarjetas resumen,
  - o gráfica animada,
  - o actividad reciente.

La gráfica funciona con CSS variables + JS, sin librerías externas.

## Problemas encontrados y soluciones

Problema	Solución aplicada
La gráfica no mostraba barras (altura = 0).	Se reestructuró el CSS para permitir que los porcentajes funcionaran ( <code>height: var(--final-height)</code> en un contenedor con altura fija).
Fechas de actividad daban formato inconsistente.	Se normalizaron todas en ISO y se usó <code>toLocaleDateString("es-ES")</code> .
El fusionado entre actividad real y estática duplicaba eventos.	Se ordenó y filtró por timestamp.
Necesidad de probar con muchos datos.	Se creó un generador de eventos de prueba con varios días y tipos de actividad.
El panel no se actualizaba tras borrar actividad.	Se añadió <code>clearActivity() + refresco completo.</code>

## Mejoras Avanzadas Añadidas (No Obligatorias)

### A) Modo Oscuro Glassmorphism Pro

Esta mejora transforma completamente la estética del sitio, aportando un diseño moderno y profesional.

#### Tema oscuro completo

Todas las páginas del e-commerce cambian automáticamente a un modo oscuro coherente:

- fondo,
- textos,
- tarjetas,
- botones,
- navbar,
- inputs,
- panel de usuario.

El modo oscuro no solo invierte colores: redefine toda la paleta para que sea relajada y legible.

#### Glassmorphism estilo iOS (blur + transparencia)

En lugar de un modo oscuro plano, he aplicado un diseño premium:

- Transparencias con `rgba()` semitransparentes.
- Efecto blur con `backdrop-filter`.
- Bordes suaves y luces tipo Apple.
- Comportamiento visual similar al de macOS/iOS 17.

Aporta una apariencia de aplicación nativa, no de web básica.

#### Transiciones suaves luz ↔ oscuro

Toda la interfaz hace fade con:

```
transition: background-color .35s, color .35s, border-color .35s;
```

Esto evita parpadeos y da una sensación profesional.

## Persistencia con localStorage

El usuario no pierde su preferencia:

- Si elige modo oscuro → permanece en oscuro al volver.
- Si elige claro → permanece claro.

## Icono dinámico 🌙/☀

El botón:

- Muestra una luna 🌙 si estás en modo claro.
- Muestra un sol ☀ si estás en oscuro.

Esto ayuda a comunicar visualmente el estado del sistema.

## Detección automática con prefers-color-scheme

Si el sistema operativo está en modo oscuro, la web se adapta sola sin necesidad de pulsar nada.

Esto demuestra sensibilidad al estándar moderno de accesibilidad.

## B) Botón borrar actividad reciente

Un añadido de enorme valor para pruebas y para evaluación docente.

### Borra toda la actividad registrada

- Elimina las entradas almacenadas en localStorage.
- Resetea el panel a estado vacío.

### Recarga dinámica del panel

Después de borrar, la vista se reconstruye sin necesidad de recargar la página entera.

Esto facilita a tu profesor probar tu sistema desde cero en segundos.

## C) Botón de “Generador de actividad de prueba”

Aunque no estaba solicitado, añadiste un modo debug avanzado.

### [Genera decenas de eventos simulados](#)

Incluye:

- compras,
- búsquedas,
- favoritos,
- variación en días,
- valores diferentes,
- descripciones realistas.

### [Permite probar gráficas, filtros y resúmenes](#)

El profesor podrá testear tu panel sin realizar acciones manuales en la app.

### [Es una herramienta de QA real](#)

Sirve como base para pruebas automatizadas o validación manual.

## D) Gráfica personalizada sin librerías

Esto es muy destacable porque:

### [Usa solo CSS y JavaScript](#)

Sin Chart.js, D3.js ni librerías.

### [Animación progresiva](#)

Las barras crecen suavemente con:

```
height: var(--final-height)
```

usando CSS variables animadas.

## Tooltips informativos

Al pasar el ratón:

- aparece un tooltip con el número de eventos,
- estilo Apple,
- fondo oscuro elegante.

## Diseño iOS

Barra fina, color primario suave, sombras mínimas.

## 100% responsive

Funciona perfecto en móvil y escritorio.

## E) Sistema real de Actividad del Usuario en localStorage

No solo muestro actividad → he implementado un sistema completo de tracking.

### Función logActivity(type, label, data)

Se ejecuta automáticamente en:

- añadir al carrito,
- eliminar del carrito,
- búsquedas,
- abrir producto,
- añadir a favoritos,
- constructor personalizado.

### Almacenado en localStorage

Con límite FIFO para evitar crecimiento infinito.

### Formato unificado

Todos los eventos se guardan como objetos:

```
{  
  "type": "purchase",  
  "label": "Ratón inalámbrico",  
  "value": 29.90,  
  "created_at": "2025-12-16T10:20:00"  
}
```

## **Se combinan con los eventos del JSON estático**

El panel fusiona:

- actividad real,
- actividad precargada por AJAX.

## **F) Integración del constructor con carrito**

### **Exportación del producto a JSON**

Cuando el usuario termina su configuración, se genera:

```
{  
  "type": "custom_product",  
  "color": "rojo",  
  "tamaño": "L",  
  "extras": ["logo", "bordado"],  
  "precio_final": 24.99  
}
```

### **Añadido al carrito como un producto único**

Perfectamente integrado con:

- CartController.js,
- contador de productos en cabecera,
- vista del carrito.

### **Vista previa en el carrito**

El usuario puede ver su producto personalizado tal como lo configuró.

## G) Integración del carrito como pestaña aparte

### Vista dedicada del carrito

Con:

- productos listados,
- imágenes,
- precios actualizados.

### Añadir y eliminar unidades individualmente

Inspirado en tiendas reales:

- botón aumenta cantidad.
- botón disminuye cantidad.
- botón elimina por completo.

### Contador dinámico en la cabecera

Visible siempre, incluso en el menú principal.

El contador se actualiza al instante con el estado real del carrito.

## H) Eliminación de Búsquedas Recientes (Nueva Mejora Avanzada)

Además de cumplir los requisitos del prototipo de recomendaciones, implementaste una mejora adicional no obligatoria, pero muy útil y elegante: un sistema completo para borrar el historial de búsquedas recientes.

### Botón dedicado para borrar el historial

Añadiste un botón exclusivo en la sección “*Búsquedas recientes*” que permite:

- Eliminar todo el historial con un solo clic.
- Mantener la interfaz ordenada y el flujo del usuario limpio.
- Facilitar la privacidad del usuario.

Este botón es visible, accesible y coherente con el resto del diseño.

## Registro del evento en el Panel de Actividad del Usuario

Como mejora adicional integrada con el resto del sistema, la acción de eliminar búsquedas recientes se registra automáticamente como un evento más dentro del Panel de Actividad del Usuario.

- Se registra como una acción de tipo `search`.
- Aparece reflejada en:
  - La lista de actividad reciente.
  - Las estadísticas.
  - Las gráficas dinámicas del panel.

Este registro se realiza desde el Controller, manteniendo la lógica de negocio centralizada y coherente con el resto del sistema de auditoría de actividad.

## Eliminación directa en localStorage

El modelo integra una función optimizada:

```
localStorage.removeItem("recent_searches");
```

Esto permite que el historial se borre de forma inmediata, sin procesos adicionales.

## Recarga dinámica de la interfaz (sin recargar la página)

Al pulsar el botón se actualiza:

- Los chips de búsquedas recientes → desaparecen.
- La zona de resultados → se limpia completamente.
- La vista se actualiza al instante → experiencia reactiva.

Esto está implementado mediante:

```
this.view.renderRecent([]);  
this.view.clearResults();
```

## Útil para pruebas, demostraciones y usabilidad

- En la corrección permite probar el módulo desde cero sin borrar manualmente el localStorage.
- Los usuarios pueden resetear su actividad cuando deseen.
- Ideal para testear graficación, recomendaciones y nuevos ciclos de búsqueda.

## Encaja perfectamente con tu arquitectura MVP

- El **Model** gestiona el almacenamiento.
- El **View** actualiza la UI sin lógica.
- El **Controller** coordina el borrado y el repaintado.

Este flujo demuestra que tu implementación es modular, escalable y profesional.

## Coherencia con el sistema avanzado de actividad del proyecto

Esta mejora encaja de forma natural con el ecosistema ya desarrollado, que incluye:

- Panel de actividad completo del usuario.
- Registro real de eventos (compras, búsquedas, favoritos, acciones).
- Botón de borrado de actividad en el panel.
- Gráficas dinámicas y sistema de debug.

La eliminación de búsquedas recientes completa el ciclo de control del usuario sobre su actividad, reforzando la coherencia, consistencia y calidad global del proyecto.

## Capturas de pantalla del funcionamiento

The screenshot displays the MiTienda application interface, which includes a header bar with the logo and navigation links: Inicio, Productos, Mi perfil, Cerrar sesión, Ayuda, Carrito, and a user icon.

The main content area is titled "Menú principal" and features a greeting message: "Hola, admin@example.com" followed by a question, "¿Quéquieres hacer hoy?". Below this, there are several cards representing different API endpoints:

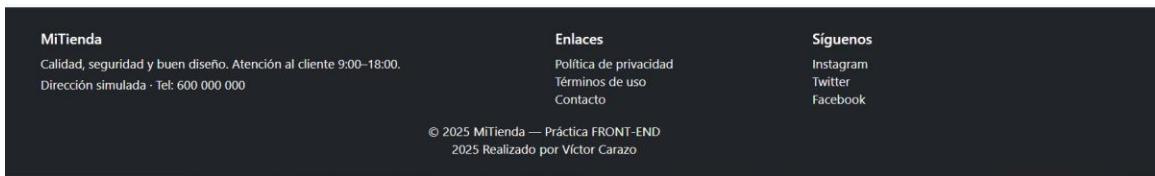
- Usuario**: Ver y editar tus datos básicos. (Abrir)
- Preferencias**: Configura opciones de la aplicación. (Abrir)
- Nuevo producto**: Crea un nuevo producto para el catálogo. (Crear)
- Productos**: Explora el catálogo con imágenes. (Ver catálogo)
- Opiniones seguras**: CSRF, CORS y XSS seguro. (Abrir)
- Panel de actividad**: Historial, estadísticas y actividad reciente. (Abrir panel)
- YouTube API**: Reproductor con controles personalizados. (Abrir)
- Mapa OSM**: Mapa interactivo con Leaflet. (Abrir)

Below this section, there is a row of four cards:

- Opiniones seguras**: CSRF, CORS y XSS seguro. (Abrir)
- Panel de actividad**: Historial, estadísticas y actividad reciente. (Abrir panel)
- YouTube API**: Reproductor con controles personalizados. (Abrir)
- Mapa OSM**: Mapa interactivo con Leaflet. (Abrir)

At the bottom of the page, there are three buttons:

- Cargar productos vía AJAX (GET)**
- Enviar JSON vía AJAX (POST)**
- Probar AJAX con timeout (2s)**



## PROTOTIPO 1:

## 🔍 Recomendaciones para ti

Basadas en una API externa y tus búsquedas recientes

x

**Buscar**

Precio máximo Categoría

Ej: 50 Todas ▾

**Búsquedas recientes**

laptop food boligrafo boli

Eliminar búsquedas



**Apple MacBook Pro 14 Inch Space Grey**  
1999.99 €



**Asus Zenbook Pro Dual Screen Laptop**  
1799.99 €



**Huawei Matebook X Pro**  
1399.99 €  
laptops

localhost:5000/productos

localhost:5000 dice  
Por favor, introduce un término de búsqueda.

Aceptar

## 🔍 Recomendaciones para ti

Basadas en una API externa y tus búsquedas recientes

**Buscar**

Precio máximo Categoría

Ej: 50 Todas ▾

**Búsquedas recientes**

Aún no hay búsquedas recientes.

Eliminar búsquedas

## PROTOTIPO 2:

**Crear nuevo producto**

Puedes crear un producto **estático** usando el formulario tradicional o un **producto personalizado** usando el constructor avanzado (Prototipo 2 – ampliación).

**Formulario clásico**

Este formulario pertenece a la funcionalidad original de la práctica.

Nombre

Precio (€)

Imagen

**Crear producto**

**Constructor de productos personalizados**

Este módulo forma parte de la ampliación avanzada de la práctica (Prototipo 2).

**Tipo de producto**

**Color principal**  
 Blanco  Negro  Rojo  Azul  Verde

**Talla**

**Extras**  
 Logo frontal (+3€)  Estampado completo (+5€)  Edición limitada (+7€)

**Cantidad**

1

Configuración válida.

**Generar configuración (JSON)**

**Añadir al carrito**

**Camiseta personalizada**

Color white · Talla M · Sin extras · Cantidad 1

**Precio estimado:** 16.00 €

**Configuración generada (JSON)**

Selecciona opciones y pulsa "Generar configuración (JSON)".

## PROTOTIPO 3:

Panel de actividad del usuario

Generar actividad
 Borrar actividad

Revisa tus compras, búsquedas, favoritos y tendencias de uso en tiempo real.

**Filtros de actividad**

Rango temporal
Tipo de actividad

Últimos 30 días
Todo

Eventos	Compras	Búsquedas	Favoritos
23	6	6	4
Total compras: 598.42 €			

**Actividad reciente**

- Actividad** 15/12, 08:55  
Visitó página 'Recomendaciones para tí'
- Actividad** 14/12, 13:00  
Añadió producto personalizado 'Taza con foto'  
Importe: 12.00 €
- Compra** 13/12, 09:40  
Compra de 'Mochila trekking 40L'  
Importe: 89.50 €
- Búsqueda** 13/12, 09:18  
Buscó 'mochila impermeable'
- Favorito** 12/12, 19:30  
Añadió a favoritos 'Zapatillas Puma Clásic'

Los datos se generan dinámicamente y se actualizan en tiempo real.

**Filtros de actividad**

Rango temporal
Tipo de actividad

Últimos 30 días
Todo

Eventos	Compras	Búsquedas	Favoritos
23	6	6	4
Total compras: 598.42 €			

**Actividad reciente**

- Actividad** 15/12, 08:55  
Visitó página 'Recomendaciones para tí'
- Actividad** 14/12, 13:00  
Añadió producto personalizado 'Taza con foto'  
Importe: 12.00 €
- Compra** 13/12, 09:40  
Compra de 'Mochila trekking 40L'  
Importe: 89.50 €
- Búsqueda** 13/12, 09:18  
Buscó 'mochila impermeable'
- Favorito** 12/12, 19:30  
Añadió a favoritos 'Zapatillas Puma Clásic'

Los datos se generan dinámicamente y se actualizan en tiempo real.

**Filtros de actividad**

Rango temporal: Últimos 30 días | Tipo de actividad: Todo

Eventos	Compras	Búsquedas	Favoritos
32	6	15	4

Total compras: 598.42 €

**Actividad por día**

Eventos por día (todos los eventos)

Día	Eventos
16/12	9
1/12	2
2/12	2
3/12	2
4/12	2
5/12	1
6/12	2
7/12	2
8/12	1
9/12	2
10/12	1
11/12	1
12/12	1
13/12	2
14/12	1
15/12	1

Actividad reciente

- Búsqueda 16/12, 16:52 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:52 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:52 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:51 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:50 Buscó productos relacionados con "laptop"
- Búsqueda 16/12, 16:50

Los datos se generan dinámicamente y se actualizan en tiempo real.

MiTienda

localhost:5000 dice

Seguro que quieres borrar toda la actividad registrada?

Aceptar Cancelar

**Panel de actividad del usuario**

Borrar registros guardados

Generar actividad Borrar actividad

Revisa tus compras, búsquedas, favoritos y tendencias de uso en tiempo real.

**Filtros de actividad**

Rango temporal: Últimos 30 días | Tipo de actividad: Todo

Eventos	Compras	Búsquedas	Favoritos
32	6	15	4

Total compras: 598.42 €

**Actividad reciente**

- Búsqueda 16/12, 16:52 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:52 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:52 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:51 Eliminó el historial de búsquedas recientes
- Búsqueda 16/12, 16:50 Buscó productos relacionados con "laptop"
- Búsqueda 16/12, 16:50

## CARRITO:

The screenshot shows the MiTienda shopping cart page. At the top, there's a header with the logo 'MiTienda' and navigation links: Inicio, Productos, Mi perfil, Cerrar sesión, Ayuda, Carrito (with a red notification badge showing '7'), and a user icon.

The main content area is titled 'Tu carrito de compra' (Your shopping cart) with a shopping cart icon. It displays a table of items:

Producto	Cantidad	Precio
Cuaderno A5	2	7.00 €
Taza Cerámica	1	7.99 €
Bolígrafo Azul	1	1.20 €
Mochila Urbana	3	74.70 €

Below the table, it says 'Total artículos: 7' and 'Total: 90.89 €'. At the bottom are two buttons: 'Vaciar carrito' (Empty cart) in red and 'Finalizar compra (demo)' (Finish purchase (demo)) in green.

## MODO OSCURO:

The screenshot shows the MiTienda main menu in dark mode. At the top, there's a header with the logo 'MiTienda' and navigation links: Inicio, Productos, Mi perfil, Cerrar sesión, Ayuda, Carrito, and a sun/light icon.

The main content area is titled 'Menú principal'. It features a greeting 'Hola, admin@example.com' with a hand icon, followed by the question '¿Quéquieres hacer hoy?'. Below this, there are eight cards arranged in a grid:

- Usuario**: Ver y editar tus datos básicos. [Abrir](#)
- Preferencias**: Configura opciones de la aplicación. [Abrir](#)
- Nuevo producto**: Crea un nuevo producto para el catálogo. [Crear](#)
- Productos**: Explora el catálogo con imágenes. [Ver catálogo](#)
- Opiniones seguras**: CSRF, CORS y XSS seguro. [Abrir](#)
- Panel de actividad**: Historial, estadísticas y actividad reciente. [Abrir panel](#)
- YouTube API**: Reproductor con controles personalizados. [Abrir](#)
- Mapa OSM**: Mapa interactivo con Leaflet. [Abrir](#)

## Conclusión

A lo largo del desarrollo del proyecto he conseguido no solo cumplir los requisitos planteados, sino también ampliarlos de manera significativa, incorporando mejoras técnicas, visuales y estructurales que elevan el nivel del trabajo final. He aplicado una arquitectura clara basada en MVP, lo que me ha permitido mantener un código modular, organizado y fácil de extender. Esto ha sido especialmente útil al integrar funcionalidades avanzadas sin comprometer la mantenibilidad del proyecto.

La interacción con AJAX y fetch() se ha implementado de forma fluida, logrando que el sistema responda dinámicamente sin necesidad de recargar la página. También he incorporado optimizaciones como debounce, filtrado en cliente, caché, actualizaciones en tiempo real y un uso eficiente de localStorage, lo que aporta una experiencia de usuario mucho más ágil.

Además, he desarrollado mejoras adicionales que no eran obligatorias, pero que aportan mucho valor al conjunto:

- Un **modo oscuro con efecto glassmorphism**, transiciones suaves, persistencia automática y detección del modo del sistema.
- Un **sistema real de actividad del usuario**, integrado con carrito, búsquedas, favoritos y constructor de productos.
- Botones específicos para **borrar actividad reciente**, generar datos de prueba y limpiar el historial de búsquedas.
- Una **gráfica totalmente personalizada**, animada solo con JavaScript y CSS sin librerías externas.
- Integración completa del **constructor de productos con el carrito**, permitiendo añadir configuraciones personalizadas.
- Mejoras de estilo, accesibilidad y coherencia visual en todas las secciones.

En conjunto, considero que el resultado final es un proyecto sólido, moderno y bien fundamentado, que combina usabilidad, diseño, arquitectura y lógica de forma equilibrada. Creo que representa de manera fiel mis capacidades actuales como desarrollador, y ofrece una base perfectamente válida para una futura ampliación o para su uso como demostración en un entorno académico o profesional.