

# Reporte de Auditoría Técnica

## 1. Lista Categorizada de Hallazgos

### Resumen Cuantitativo:

- **Críticos:** 2
- **Mejoras:** 3
- **Sugerencias:** 2

### Detalle de Hallazgos:

Tipo	Línea aprox.	Problema detectado
● Crítico	27	<b>Manejo inseguro de respuesta:</b> response.json() fallará si el servidor devuelve HTML (error 500/502), rompiendo la aplicación.
● Crítico	38	<b>Error de sintaxis:</b> Etiqueta </script> duplicada al final del archivo.
● Mejora	15	<b>Falta de Timeout:</b> La petición puede quedar colgada indefinidamente si la red falla, bloqueando recursos.
● Mejora	23	<b>Errores genéricos:</b> Se lanza un error solo con el código de estado, ignorando el mensaje explicativo que envía el servidor.
● Mejora	30	<b>Falta de validación:</b> Se asume que <code>datos</code> siempre es una tabla válida sin verificar la estructura antes de usarla.
● Sugerencia	10	<b>Hardcoding:</b> La URL de la API está escrita directamente en el código.
● Sugerencia	General	<b>Responsabilidad única:</b> La función <code>obtenerProductos</code> mezcla lógica de red con lógica de presentación (UI).

# Justificación de Mejoras Seleccionadas

## 1. Implementación de Timeouts (AbortController)

- **Decisión:**  Implementada.
- **Justificación:** En el mundo real, las redes son inestables. Sin un timeout, una petición a un servidor caído podría dejar al navegador del usuario "pensando" indefinidamente, lo que lleva a una mala experiencia de usuario (UX). He configurado un límite de 5 segundos para liberar los recursos si el servidor no responde rápido.

## 2. Propagación de detalles del error (Leer body en errores 4xx/5xx)

- **Decisión:**  Implementada.
- **Justificación:** El código original solo decía "Error 400". Esto es inútil para debuggear. Al leer el cuerpo de la respuesta (donde el backend suele enviar mensajes como "Falta header X" o "Token inválido"), facilitamos enormemente la corrección de problemas tanto para el desarrollador como para el usuario final.