Diseñador de Capa de Logging

Muestra de Logs Reales
Se ejecutó una batería de pruebas simulando diferentes escenarios para validar la captura de logs en todos los niveles requeridos.

Captura de Evidencia: En la siguiente imagen se observan los logs generados en tiempo real desde la consola del navegador:

DEBUG (Gris): Muestra el inicio de la petición y la URL objetivo.

WARN (Naranja): Alerta automática disparada cuando la latencia supera los 2000ms.

ERROR (Rojo): Captura de excepciones críticas (ej. DNS no resuelto o errores 404).

Propuesta de Mejoras para Versión 2.0
Respondiendo a la pregunta: "¿Qué información adicional te gustaría capturar y cómo mejorarías el sistema?", se proponen las siguientes dos evoluciones:

A. Implementación de Correlation IDs (Trazabilidad Distribuida)
Actualmente, si un usuario reporta un error, es difícil saber qué ocurrió exactamente en el servidor.

Mejora: Generar un X-Correlation-ID (UUID único) en el cliente al inicio de cada petición.

Beneficio: Este ID se envía al Backend y se registra en todos los logs (Frontend, API, Base de Datos). Esto permite rastrear una transacción completa "punta a punta" filtrando por un solo ID.

B. Persistencia y Agregación en la Nube
Actualmente, los logs solo viven en la consola del navegador del usuario ("Console.log"), por lo que se pierden al cerrar la pestaña.

Mejora: Crear un mecanismo de "Batching" (lotes) que acumule logs en un array y los envíe cada 30 segundos a un servicio externo (como Datadog, Sentry o ELK Stack).

Beneficio: Permite al equipo de ingeniería visualizar dashboards en tiempo real y recibir alertas automáticas (ej. Slack/Email) cuando la tasa de errores sube repentinamente, sin esperar a que los usuarios se quejen.