

****Informe Semanal****

****1. Estructura del proyecto****

Se ha observado que la estructura actual del repositorio no está completamente alineada con las convenciones de proyectos React + Firebase. Existen dos carpetas principales (`202509-sigma` y `src`) en la raíz, lo que genera ambigüedad en la organización del código. El código fuente debería concentrarse en una única carpeta `src` dentro del proyecto principal. También se detectan carpetas `.idea` generadas por IDEs, que no deberían incluirse en el repositorio.

****2. Código fuente y tipado****

El proyecto utiliza TypeScript (`.tsx`, `.ts`), lo cual es positivo, pero se han encontrado componentes y módulos con estructura básica o incompleta. Algunos archivos carecen de tipado explícito en props y estados, lo cual puede derivar en errores en tiempo de compilación. También se observan componentes con responsabilidades mezcladas, como manejo de lógica y renderizado en el mismo archivo.

****3. Configuración de Firebase****

El archivo `firebase.ts` dentro de la carpeta `lib` parece incompleto o sin la configuración necesaria para la autenticación, Firestore y Analytics. Es recomendable separar claramente las instancias exportadas (`auth`, `db`, `analytics`) y mantener las credenciales seguras mediante variables de entorno reales, no en archivos públicos.

****4. Estilos y organización visual****

Se detecta una mezcla de estilos en CSS tradicional (`main.css`, `App.css`) sin un sistema claro de clases o uso de utilidades modernas como TailwindCSS. Esto puede generar inconsistencias visuales entre componentes. Es recomendable adoptar un enfoque unificado de estilos para garantizar coherencia visual y mantenimiento sencillo.

****5. Buenas prácticas de desarrollo****

Faltan convenciones de commits descriptivos y una estructura modular clara. Algunos archivos no siguen un patrón de nombrado coherente. También se observa la falta de comentarios explicativos en el código, lo cual dificulta la colaboración en equipo. Es recomendable mantener comentarios detallados y dividir la lógica en módulos reutilizables.

****6. Conclusiones****

El proyecto muestra avances iniciales pero requiere una revisión de arquitectura, tipado y estructura general para alinearse con las mejores prácticas de desarrollo web moderno. Se recomienda una refactorización estructural, documentación más clara y una mejor gestión de dependencias y configuración.

****Firmado:****

Jose Manuel García Segura

Equipo Sigma — Nexeus Big Data Labs