

Propuesta de Proyecto: MindSync

"Your Second Brain: AI-Powered Knowledge Management"

Integrantes:

- Anyelo Arthur Sánchez - 24001323
 - Fernando Alegría - 24000141
-

1. Objetivo y Alcance

Planteamiento del Problema: La Paradoja de la Memoria Digital

En la actualidad, vivimos inmersos en una paradoja de "infoxicación". Nuestra capacidad para capturar información es infinita (capturas de pantalla, notas de voz, fragmentos de texto), pero nuestra capacidad para procesarla y recuperarla es limitada. El resultado es un "cementerio digital": gigabytes de datos valiosos que se vuelven irrelevantes porque carecen de contexto, estructura y accesibilidad. Capturamos para recordar, pero el caos del almacenamiento nos condena a olvidar.

Nuestra Solución: MindSync

Concebimos **MindSync** no como una simple aplicación de notas, sino como un "**Segundo Cerebro**" impulsado por Inteligencia Artificial. Nuestra solución busca reducir la entropía digital mediante un ecosistema dual (Móvil/Web).

- **Móvil (Android):** Actúa como un *sensor de captura* inmediato, intuitivo y ubicuo.
- **Web (Next.js):** Funciona como el *centro de procesamiento lógico* y visualización.

Utilizando la **API de Gemini** como motor de inferencia, MindSync transforma datos no estructurados (imágenes, audio) en conocimiento estructurado, permitiendo al usuario dialogar con su propia información.

Audiencia Objetivo (Target)

La aplicación está diseñada para "**trabajadores del conocimiento**": investigadores académicos, desarrolladores de software y estudiantes universitarios que requieren sintetizar y organizar grandes volúmenes de información visual y auditiva en tiempos reducidos.

2. Características Clave e Infraestructura

Estrategia de Autenticación y Datos (Firebase)

Para garantizar la seguridad y la sincronización en tiempo real, utilizaremos el ecosistema de Firebase:

- **Autenticación:** Implementaremos **Firebase Authentication** unificado. Esto permitirá que el usuario acceda a su "cerebro digital" desde el móvil o la web manteniendo una sesión segura y persistente.
- **Almacenamiento de Archivos (Storage):** **Firebase Storage** servirá como repositorio de "raw data" (los archivos multimedia originales: imágenes y audios).
- **Base de Datos (Firestore):** Utilizaremos **Cloud Firestore** (NoSQL) para almacenar los metadatos semánticos generados por la IA. La estructura será documental, guardando referencias al archivo original junto con el JSON Enriquecido por Gemini (títulos, resúmenes, tags).

Las 5 Funcionalidades Distintivas y Plan de IA

A continuación, detallamos las funcionalidades operativas, destacando la integración de la **API de Gemini** en cada una y nuestras **Rutas de Flexibilidad** (alternativas técnicas para asegurar la entrega).

1. Ingesta Multimodal y Procesamiento Contextual (Móvil)

- **Descripción:** El usuario captura una imagen (ej. una pizarra con fórmulas) o graba una nota de voz.
- **Integración IA:** Gemini procesa el archivo, transcribe el contenido (audio a texto / OCR), genera un título autodescriptivo y asigna etiquetas (tags) automáticas para su categorización.
- **Valor:** Elimina la fricción cognitiva de tener que escribir manualmente qué es cada archivo.

2. Motor de Búsqueda Semántica (Web/Móvil)

- **Descripción:** Permite realizar consultas en lenguaje natural (ej: "*¿Dónde guardé los apuntes sobre arquitectura hexagonal?*").

- **Integración IA:** El sistema no busca por nombre de archivo, sino por el *significado* y el contexto del contenido analizado previamente por la IA.
- **Ruta de Flexibilidad:** Si la implementación de *Vector Search* resulta compleja para los plazos, implementaremos una **Búsqueda por Tags Indexados**, donde el buscador filtrará estrictamente por las palabras clave generadas por Gemini.

3. Extracción de Accionables / "Actionable Insights" (Móvil)

- **Descripción:** El sistema detecta intenciones temporales o tareas dentro de las imágenes (fechas de exámenes, citas en flyers) y sugiere crear recordatorios.
- **Integración IA:** Gemini analiza la imagen para extraer entidades de tipo Fecha, Hora y Evento.
- **Ruta de Flexibilidad:** Si la integración con Google Calendar presenta barreras de permisos (OAuth), el sistema generará **Notificaciones Push Locales** o enviará un correo electrónico estructurado (formato .ics).

4. Sistema de "Active Recall" / Modo Estudio (Web)

- **Descripción:** Basado en principios pedagógicos, la aplicación selecciona un grupo de notas y genera un cuestionario de repaso.
- **Integración IA:** Gemini recibe el contexto de las notas seleccionadas y genera preguntas de opción múltiple (JSON) para evaluar al usuario.
- **Valor:** Transforma el almacenamiento pasivo en aprendizaje activo.

5. Visualización de Conocimiento / Dashboard (Web)

- **Descripción:** Un panel analítico que agrupa visualmente las notas por categorías inferidas (ej. Salud, Finanzas, Proyectos).
 - **Integración IA:** Gemini clasifica cada nota en una "Categoría Maestra" al momento de la creación.
 - **Ruta de Flexibilidad:** Nuestra ambición es crear un **Grafo de Nodos** interactivo. Si la complejidad de UI es elevada, optaremos por un diseño tipo **Tablero Kanban Inteligente**, que cumple la misma función organizativa.
-

3. Stack Tecnológico y Justificación

- **Front-end Web (Next.js + Tailwind CSS):**
 - *Justificación:* Elegimos Next.js por su capacidad de renderizado híbrido y optimización SEO. La web será el "lienzo" principal para la visualización de datos complejos (Dashboard y Quiz), aprovechando el espacio de pantalla que el móvil no ofrece. Tailwind asegura un desarrollo UI rápido y consistente.
- **Desarrollo Móvil Nativo (Kotlin - Android):**
 - *Justificación:* La captura de información debe ser inmediata. Kotlin nos permite un acceso profundo y eficiente al hardware (cámara, micrófono) y el uso de "Intents" para recibir contenido compartido desde otras apps.
- **Backend y Orquestación (Firebase Cloud Functions + Node.js):**
 - *Justificación:* Utilizaremos un backend *serverless* para intermediar las peticiones a la API de Gemini. Esto es crucial para la seguridad (no exponer API Keys en el cliente) y para sanear los datos antes de persistirlos en Firestore.