



Documentación Técnica Provisional: Módulo Coach Financiero (AI)

Proyecto: [Nombre de tu App]

Módulo: Asistente Inteligente (Gemini AI)

Fecha: 12 de Febrero, 2026

Estado: MVP / Fase de Desarrollo

1. Resumen Ejecutivo

El "Coach Financiero" es una funcionalidad core de la aplicación que utiliza Inteligencia Artificial Generativa para analizar los patrones de gastos e ingresos del usuario. Su objetivo es transformar datos fríos (números en una base de datos) en consejos accionables, motivadores y estratégicos mediante procesamiento de lenguaje natural.

2. Estado Actual (Arquitectura MVP)

Actualmente, el módulo opera bajo una arquitectura "Client-Side" optimizada para desarrollo y pruebas rápidas.

2.1 Stack Tecnológico

- Proveedor de IA:** Google DeepMind / Google AI Studio.
- Modelo Activo:** gemini-2.5-flash-lite.
 - ¿Por qué este modelo? Es la versión más eficiente en costo-beneficio de 2026. Ofrece baja latencia (respuestas rápidas) y una cuota gratuita más generosa para desarrollo que los modelos "Pro".
- Conexión:** HTTP Directo (REST API v1).
 - Se eliminó la dependencia del SDK oficial de Flutter para evitar conflictos de versiones (v1beta vs v1) y tener control total sobre los *Headers* y *Body* de la petición.
- Formato de Respuesta:** Markdown.
 - La IA devuelve texto enriquecido con negritas, listas y emojis, renderizado en la App mediante `flutter_markdown`.

2.2 Lógica de Negocio Implementada

1. Análisis Diferenciado:

- **Semanal (Flash):** Prompt optimizado para respuestas cortas ("Tweets"), enfocado en corrección de rumbo inmediato.
 - **Mensual (Reporte):** Prompt detallado para análisis profundo de presupuesto, brecha de ahorro y proyección de metas.
2. **Sistema de Caché (Ahorro de Costos):**
 - Se implementó una lógica de persistencia local. Si el usuario consulta el Coach múltiples veces en el mismo periodo sin cambios drásticos en sus datos, la App muestra la última respuesta guardada **sin consumir peticiones a la API**.
 3. **Interfaz Adaptativa:**
 - Botones de selección con feedback visual (estado activo/inactivo).
 - Scroll dinámico para lecturas largas.
-

3. Limitaciones Actuales (Bloqueantes para Producción)

Para lanzar a la Play Store, debemos resolver las siguientes limitaciones críticas de la versión de desarrollo actual:

● 3.1 Seguridad de la API Key (Crítico)

- **Situación:** La llave de acceso (`API_KEY`) reside actualmente dentro del código de la aplicación móvil (archivo `.env` o código duro).
- **Riesgo:** Mediante ingeniería inversa, un atacante podría extraer nuestra llave y usar nuestra cuota para sus propios proyectos, lo que generaría costos inesperados o la suspensión del servicio.

● 3.2 Cuotas de Nivel Gratuito (Free Tier)

- **Situación:** Estamos usando el plan gratuito de Google AI Studio.
- **Limitación:** Este plan tiene límites estrictos de **Rate Limit** (ej. 15 peticiones por minuto) y **Daily Limit** (ej. 1,500 peticiones al día compartidas entre todos los usuarios).
- **Consecuencia:** Si 100 usuarios intentan usar el Coach al mismo tiempo, la mayoría recibirá un error `429 Resource Exhausted`.

● 3.3 Privacidad de Datos

- **Situación:** En el nivel gratuito, Google se reserva el derecho de usar los inputs (datos anónimos) para mejorar/entrenar sus modelos.
 - **Requisito:** Para una app financiera seria, debemos garantizar que los datos no se usen para entrenamiento ajeno (esto se logra pagando el servicio).
-

4. Estrategia de Despegue a Play Store (Roadmap)

A diferencia de una "suscripción mensual fija" (como Netflix), las APIs de IA empresarial funcionan bajo el modelo **Pay-as-you-go (Pago por uso)**. Solo pagamos por lo que nuestros usuarios consumen.

4.1 Arquitectura Propuesta: Firebase Vertex AI

Para solucionar el problema de seguridad, migraremos la conexión a **Vertex AI for Firebase**.

- **Cómo funciona:** La App no tiene la llave. La App se autentica con Firebase (Login de usuario) y Firebase gestiona la llamada a Gemini de forma segura.
- **Ventaja:** Permite poner límites por usuario (ej: "Usuario X solo puede hacer 5 consultas al mes").

4.2 Modelo de Costos (Estimación)

El modelo `gemini-2.5-flash` es extremadamente económico.

- **Unidad de cobro:** Por cada 1 millón de Tokens (palabras/caracteres).
- **Costo aproximado:** ~\$0.10 USD por 1 millón de tokens de entrada (aprox).
- **Escenario Real:**
 - Un análisis mensual consume ~500 tokens.
 - Con \$1.00 USD, podríamos procesar aproximadamente **20,000 análisis financieros**.
- **Conclusión:** El costo operativo es muy bajo, pero debe ser monitoreado.

4.3 Estrategia de Monetización (Sostenibilidad)

Para cubrir estos costos de API, tenemos tres caminos:

1. **Modelo Freemium (Recomendado):**
 - *Gratis:* Análisis Semanal (Breve, consume pocos tokens).
 - *Premium:* Análisis Mensual Profundo + Chat ilimitado con el Coach.
2. **Límite Duro:**
 - Todos los usuarios tienen 1 crédito de análisis semanal gratis. Si quieren más, deben ver un anuncio (Ads pagan la API).

5. Próximos Pasos para el Equipo

1. **Frontend (UI/UX):** Terminar la implementación del "Modo Bienvenida" para usuarios nuevos (sin datos históricos).
2. **Backend (Firebase):** Configurar proyecto en Google Cloud Console y habilitar Vertex AI.
3. **Legal:** Redactar Términos y Condiciones aclarando que el "Coach" es una IA y no un asesor financiero humano certificado (descargo de responsabilidad obligatorio).