

## Descripción del proyecto

- Aplicación móvil Android (Java) para registrar "Experiencias" (producto, servicio, costo, categoría, descripción, favorito).

- Sincronización bidireccional con Google Calendar mediante n8n: cuando se crea/edita una experiencia se crea/actualiza un evento en Google Calendar; cuando hay cambios en Calendar, se envían a la app/servicio para actualizar la experiencia.

- Comunicación por webhooks HTTP entre tu backend/servicio en Railway y los workflows de n8n.

- Endpoints públicos en tu despliegue Railway:

- `https://primary-production-6c81c.up.railway.app/webhook-test/experiencias/webhook`

- **Usa webhooks** para comunicación entre tu app y n8n
- **Maneja errores** tanto en la app como en n8n

2. En la app:

- Abrir proyecto en Android Studio, compilar.

- Configurar `DatabaseHelper` si usa campos: `synced` boolean, `calendarEventId` string.

- Implementar o verificar `WebhookManager` para que haga POST al endpoint Railway con JSON y header de autorización (`X-Webhook-Secret` ).

3. En Railway / backend:

- Asegurar que `POST` a `.../experiencias/webhook` acepte y valide un header `X-Webhook-Secret` .

- Responder rápido con 200 OK para webhooks y procesar en background.

4. En Google Cloud:

- Crear credenciales OAuth y configurar redirect URLs de n8n (si n8n maneja OAuth).

- Habilitar Google Calendar API.

## Integración con n8n — explicación

### 1. Preparar credenciales en n8n:

- En n8n crear credencial Google Calendar con OAuth client ID/secret. Probar conexión.

### 2. Workflow: sincronización app -> n8n -> Google Calendar

#### - Nodo 1: HTTP Request / Webhook Trigger

- Usar un nodo `Webhook` (HTTP Request incoming) si quieres que la app llame a n8n directamente, o `HTTP Request` node si n8n debe llamar a tu Railway endpoint.
  - Si la app llama n8n: exponer un webhook n8n y configurar la app para `POST` allí.
  - Si n8n llama a Railway: usar `HTTP Request` en n8n apuntando a `https://primary-production-6c81c.up.railway.app/webhook-test/experiencias/webhook` .

#### - Nodo 2: Set/Function

- Mapear campos de la experiencia al formato de Google Calendar:

- título: `producto` o `servicio`
- descripción: `descripcion + categoría + costo + favorito`
- start / end: calcular (puede usarse fecha actual o datos provistos)

#### - Nodo 3: Google Calendar (Create / Update)

- Si payload contiene `calendarEventId` -> `Update` evento.
- Si no -> `Create` evento.

#### - Nodo 4: Set / HTTP Request (respuesta)

- Enviar resultado (event id) de vuelta al backend/app para guardar `calendarEventId` .

### 3. Workflow: Google Calendar -> n8n -> app/servicio (bidireccional)

#### - Nodo 1: Google Calendar Trigger (watch events)

- Configurar para `on event created/updated/deleted` .

- Nodo 2: Filter / Function
  - Filtrar cambios originados por la propia integración (para evitar loops). Ejemplo: usar en la descripción una etiqueta `prox0-sync:id=...` .
- Nodo 3: Mapear datos a formato de experiencia.
- Nodo 4: HTTP Request a `https://primary-production-6c81c.up.railway.app/webhook-test/experiencias/webhook` con header `X-Webhook-Secret` .
  - Método: POST con JSON.
- Nodo 5: Error handling / retry (ver abajo).

#### 4. Estructura de payload recomendada (app -> n8n y n8n -> app)

```
```json {
  "action": "create|update|delete",
  "experience": {
    "id": "local-id-123",
    "title": "Nombre del producto/servicio",
    "category": "🍽️ Comidas & Restaurantes",
    "product": "Hamburguesa",
    "service": "",
    "cost": 12.5,
    "description": "Descripción...",
    "favorite": true,
    "calendarEventId": "google-event-id-abc"
  },
  "timestamp": "2025-10-25T12:00:00Z"
}
```

#### 6. Manejo de errores (app y n8n)

- En la app:

- Enviar webhook en background (WorkManager) con reintentos exponenciales si falla.
- Marcar en DB estado `pending\_sync` y campo `last\_sync\_error` .
- Mostrar notificación al usuario si falla permanentemente.
- En n8n:
  - Usar nodo `Error Trigger` o `Catch` para crear un flujo de manejo de errores.
  - Reintentos automáticos en nodos `HTTP Request` (configurar `Retry` y `Timeout` ).
  - Guardar logs de errores (puede enviar a un endpoint `/logs` o a un Google Sheet/DB).
  - Filtrar y evitar loops: cuando n8n crea un evento en Calendar, incluir una marca en la descripción (ej. `prox0-sync:local-id-123` ) y al recibir triggers de Calendar ignorar eventos con esa marca si vienen del mismo origen.

## 7. Pruebas y verificación

- Paso 1: Crear experiencia en app, verificar que:
  - Se inserta en DB local.
  - `WebhookManager` hace POST a Railway (ver logs en Railway).
  - n8n recibe payload y crea event en Google Calendar.
  - Respuesta de n8n con `calendarEventId` actualiza el registro local/servidor.
- Paso 2: Crear evento manual en Google Calendar, verificar que n8n detecta el cambio y hace POST a tu endpoint para crear/actualizar la experiencia.
  - Verificar casos de error (rechazo 401 por secret faltante, timeouts, formato inválido).

### Checklist corto

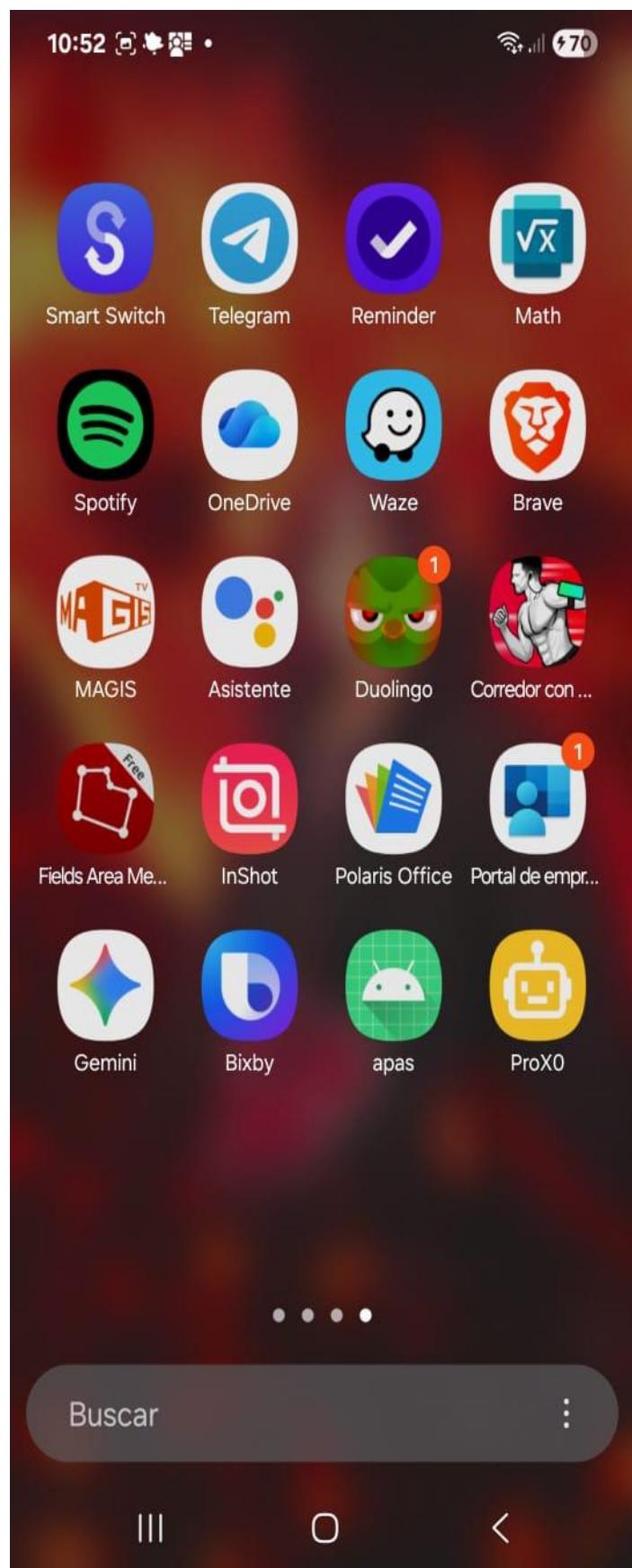
- [ ] Credenciales Google en n8n configuradas y probadas.
- [ ] Webhook secret en Railway y n8n igual.

- [ ] `WebhookManager` implementa POST con retries y callbacks (usar WorkManager).
- [ ] DB local tiene `calendarEventId` y `syncState` .
- [ ] n8n workflow: trigger (Calendar), mapping, HTTP Request a Railway, manejo de errores.
- [ ] Pruebas end-to-end realizadas.

#### Resumen final

- Comunicación por webhooks entre app/backend y n8n; n8n actúa como orquestador entre la app y Google Calendar.
- Asegurar validación de secret, reintentos y marca anti-loop para sincronización bidireccional.
- Implementar reintentos en la app (WorkManager) y usar nodos de manejo de errores en n8n.

Ejemplos concretos de payload, endpoints y recomendaciones ya están incluidos arriba para integrar rápidamente.



¡Bienvenido Abner Mateo!



Registrar Experiencia



Ver Historial



Consultas IA



Favoritos



Compartir por WhatsApp

Registrar Experiencia

📁 Selecciona una categoría:

➕ Comidas & Restaurantes

🛍️ Compras & Productos

✈️ Viajes & Lugares

🎬 Entretenimiento

Costo (\$)

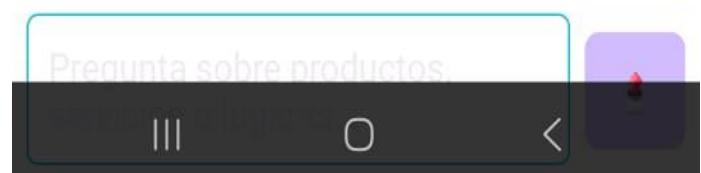
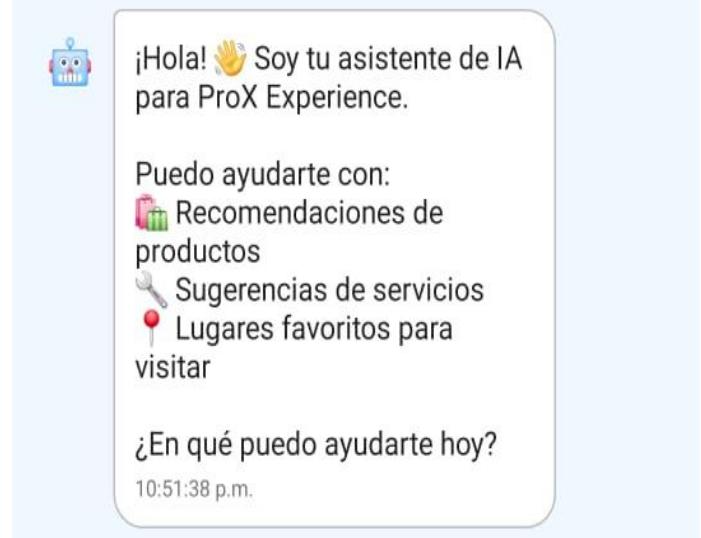
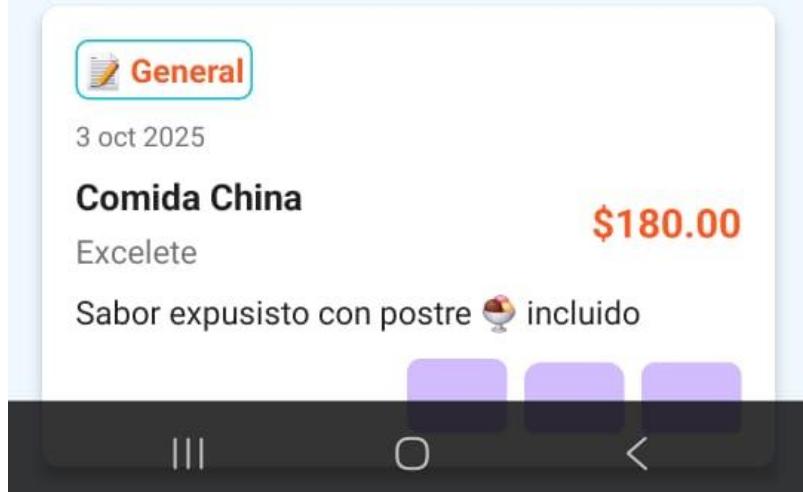
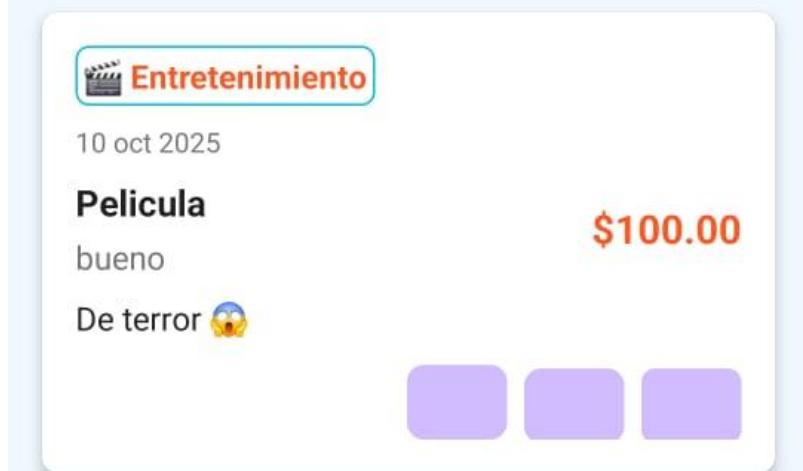
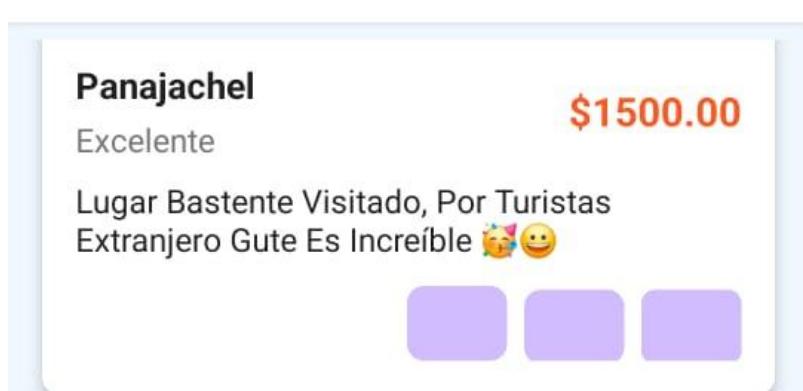
Descripción de la experiencia

Guardar Experiencia

⭐ Favorito

💬 Compartir en WhatsApp





10:51

70

## Historial de Experiencias

IA

### Recomendaciones de IA

1. Producto: Samsung Galaxy Watch, Servicio: excelente, Costo: \$2000.0

Este producto podría interesarle ya que le gustó mucho el Samsung A54 y le pareció bueno el servicio. Además, le gusta invertir en tecnología de calidad.

2. Producto: Libro de cocina guatemalteca, Servicio: excelente, Costo: \$120.0

Dado que le gustó mucho la comida china y le pareció excelente el servicio, podría disfrutar de explorar nuevas recetas de comida guatemalteca. Además, le gusta invertir en experiencias culinarias.

3. Producto: Tour a Antigua Guatemala, Servicio: excelente, Costo: \$300.0

Considerando que le gustó mucho Panajachel y le pareció excelente el servicio, podría disfrutar de un tour a la hermosa ciudad de Antigua Guatemala. Además, le gusta invertir en experiencias de viaje y conocer nuevos lugares.

**Compartir****Entendido****\$1500.00**

Excelente

Lugar Reciente Visitado Por Turistas  
Extranjero Gute Es Increible

10:51

70



A

+502 3990 0128 (Tú)

Envía mensajes a este mismo número.

### Recomendaciones de IA - ProX Experience

1. Producto: Samsung Galaxy Watch, Servicio: excelente, Costo: \$2000.0

Este producto podría interesarle ya que le gustó mucho el Samsung A54 y le pareció bueno el servicio. Además, le gusta invertir en tecnología de calidad.

2. Producto: Libro de cocina guatemalteca, Servicio: excelente, Costo: \$120.0

Dado que le gustó mucho la comida china y le pareció excelente el servicio, podría disfrutar de explorar nuevas recetas de comida guatemalteca. Además, le gusta invertir en experiencias culinarias.

3. Producto: Tour a Antigua Guatemala, Servicio: excelente, Costo: \$300.0

Considerando que le gustó mucho Panajachel y le pareció excelente el servicio, podría disfrutar de un tour a la hermosa ciudad de Antigua Guatemala. Además, le gusta invertir en experiencias de viaje y conocer

**Mensaje**

The screenshot shows a Google Calendar view for Saturday, October 25, 2025. The sidebar displays the month of October 2025. The main calendar area shows several overlapping events from 2 AM to 7 AM. A tooltip for one event reads:

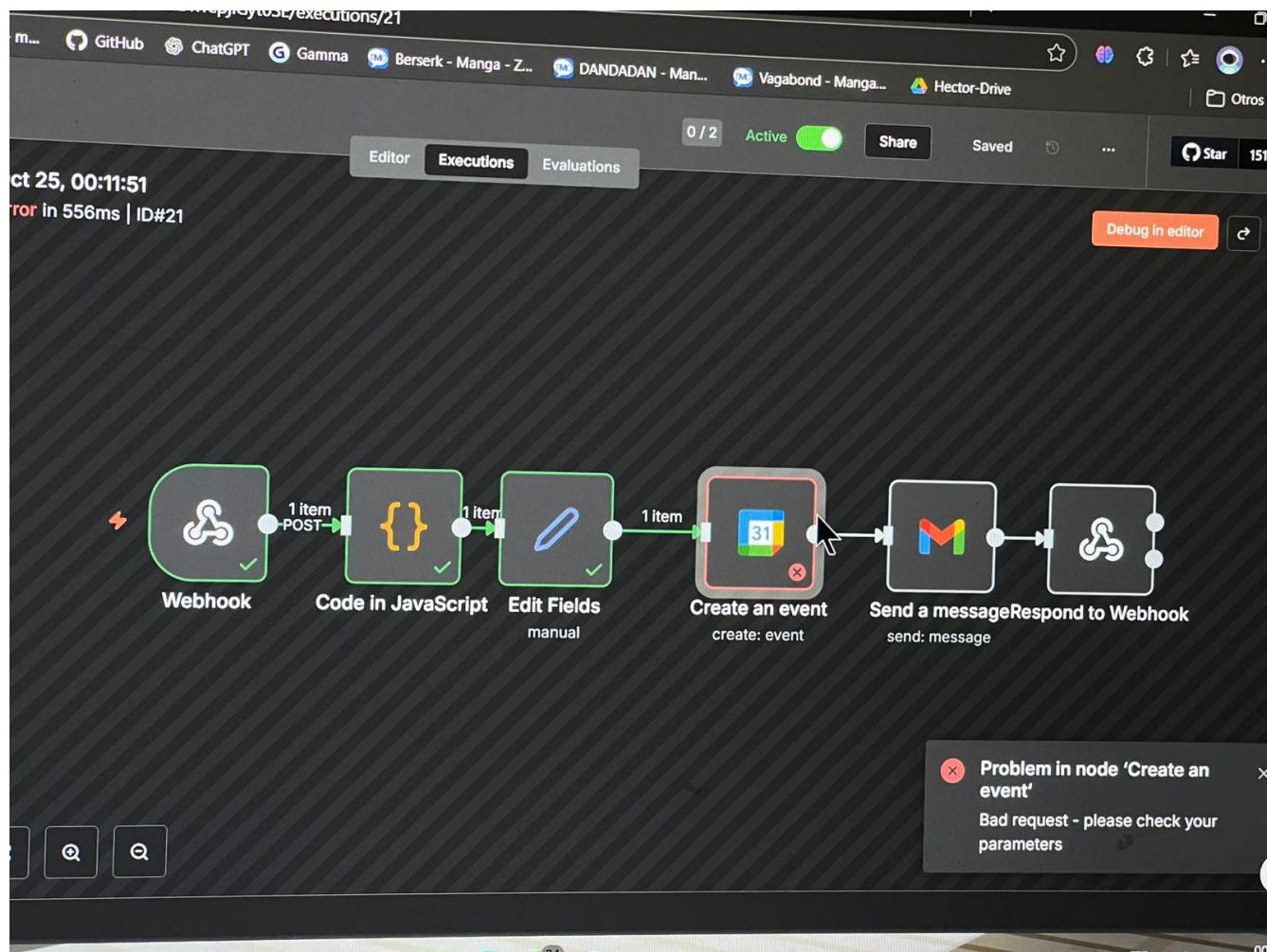
- (Sin título)** Sábado, 25 de octubre · 1:52 - 2:52am
- Tomar notas de la reunión** Crea un documento para tomar notas
- Categoría: General
- Producto:
- Servicio:
- Costo:
- Descripción:

Below the tooltip, a reminder is shown: **10 minutos antes**.

The screenshot shows a workflow editor interface for a specific execution on October 25, 2025, at 03:03:16. The execution was successful in 1.96s. The workflow consists of the following steps:

```
graph LR; A[Webhook] -- "2 artículos" --> B["Código en JavaScript"]; B -- "2 artículos" --> C["Crear un evento  
crear: evento"]; C -- "2 artículos" --> D["Responder al webhook"]
```

The left sidebar lists previous executions and a question about which ones are stored. The bottom navigation bar includes links for Históricos destacados and Buscar.



En la integración también nos daba muchos errores los cuales supimos resolver con la ayuda de la inteligencia artificial muchas ocasiones era por que los parámetros no eran los correcto, la verdad un reto bastante bonito para poder aprender y enriquecer nuestros conocimientos en proyectos futuros

