

**UNIVERSIDAD MARIANO GALVÉZ DE GUATEMALA**

**CAMPUS JUTIAPA**

**FACULTAD INGENIERÍA EN SISTEMAS**

# **TUMATCHAPP**



**INTEGRANTES:**

**LITZI YASMIRA PINEDA CERMEÑO**

**ADRIANA STEPHANIA LÓPEZ LARA**



## DOCUMENTACIÓN DEL FLUJO DE INTEGRACIÓN REGISTRO DE USUARIOS

### 1. Descripción general

Este flujo lo desarrollamos con el propósito de automatizar el registro de usuarios que provienen directamente de nuestra aplicación móvil, creada en Android Studio.

Cuando una persona se registra desde la app, la información se envía automáticamente al Webhook, que actúa como punto de entrada del sistema.

A partir de ahí, todo el proceso se ejecuta de manera automática:

validación, registro en Google Sheets y envío de un correo mediante Gmail.

Este flujo combina nuestra app móvil con herramientas en la nube para garantizar que cada registro quede documentado, validado y notificado sin intervención manual.

### 2. Componentes principales

- **Aplicación Android (origen de los datos):** Desarrollamos nuestra app en Android Studio. Desde allí, los usuarios completan un formulario con sus datos personales (nombre, correo, teléfono, etc.).  
Una vez que presionan “Registrar”, la aplicación envía los datos al Webhook a través de una solicitud HTTP POST.  
La app fue configurada para manejar posibles errores de conexión y confirmar visualmente al usuario cuando su registro ha sido procesado.
- **Webhook:** El Webhook recibe los datos enviados por la aplicación móvil. Es el punto de conexión entre nuestra app y el flujo automatizado en la nube. Cada vez que se recibe una solicitud, el Webhook inicia el proceso de validación y registro.
- **Registro\_usuarios (Bot principal)** Este componente coordina todas las acciones del flujo. Recibe los datos desde el Webhook, los valida con ayuda de un modelo de OpenAI, y ejecuta las siguientes tareas:
  - Guarda el registro en una hoja de Google Sheets.
  - Envía un correo de confirmación mediante Gmail.
- **OpenAI Chat Model:** Lo utilizamos para analizar y validar los datos recibidos desde la app. Este modelo nos permite aplicar una validación más contextual (por ejemplo, detectar si un correo está mal escrito o si un número de teléfono tiene formato incorrecto). Además, genera textos personalizados para los correos de confirmación.

- **Gmail:** Se encarga de enviar automáticamente un correo de confirmación al usuario registrado o una notificación interna si ocurre algún error. Esto garantiza una comunicación inmediata y profesional.
- **Google Sheets:** Sirve como registro principal de todos los usuarios que se registran desde la app. Cada fila contiene los datos validados, junto con la fecha y hora en que se procesaron.

### 3. Flujo general del proceso

1. El usuario completa el formulario en la **app Android**.
2. La app envía los datos al **Webhook** mediante una solicitud POST.
3. El **Webhook** recibe la información y la pasa al bot **Registro\_usuarios**.
4. El bot solicita al **modelo de OpenAI** que valide los datos y genere el mensaje de confirmación.
5. Si los datos son válidos:
  - a) Se guardan en **Google Sheets**.
  - b) Se envía un correo a través de **Gmail**.
6. Finalmente, la app recibe una respuesta de éxito o error para informar al usuario.

### 4. Diagramas de funcionamiento

#### 4.1. Diagrama de flujo

```

A[App Android (Registro de usuario)] --> B[Webhook (POST)]
B --> C[Bot: Registro_usuarios]
C --> D[OpenAI Chat Model]
C --> E[Google Sheets: Guardar registro]
C --> F[Gmail: Enviar correo]
D --> C
  
```


#### 4.2. Diagrama de secuencia

```

sequenceDiagram
    participant Usuario
    participant App as App Android
    participant Webhook
    participant Bot as Registro_usuarios
    participant Modelo as OpenAI Chat Model
    participant Hoja as Google Sheets
    participant Correo as Gmail
  
```

Usuario->>App: Completa formulario de registro





App->>Webhook: Envía datos del usuario (POST)  
Webhook->>Bot: Transfiere la información  
Bot->>Modelo: Valida datos y genera mensaje  
Modelo-->>Bot: Devuelve validación y texto  
Bot->>Hoja: Registra el usuario  
Bot->>Correo: Envía correo de confirmación  
Bot-->>Webhook: Devuelve estado del proceso  
Webhook-->>App: Respuesta (registro exitoso)  
App-->>Usuario: Muestra mensaje de confirmación

## 5. Decisiones técnicas

1. **Uso del Webhook:** Optamos por esta arquitectura porque es la forma más directa de conectar nuestra aplicación móvil con los servicios externos. Nos permite recibir la información en tiempo real y activar el flujo inmediatamente después de que el usuario se registra.
2. **Integración con OpenAI Chat Model:** Incorporamos este modelo para validar los datos con mayor precisión y naturalidad. Nos da la ventaja de detectar errores semánticos y generar mensajes de correo coherentes y personalizados.
3. **Registro en Google Sheets:** Decidimos almacenar los datos en una hoja de cálculo porque ofrece una visualización rápida y colaborativa. Podemos revisar los registros sin depender de herramientas técnicas y mantener trazabilidad inmediata.
4. **Uso de Gmail:** El envío de correos mediante Gmail fue una elección práctica: es confiable, fácil de integrar y mantiene la comunicación en el mismo ecosistema de Google.

## 6. Seguridad y Validación

Desde el desarrollo en Android hasta la ejecución del flujo, implementamos medidas de seguridad básicas:

- **Validación en la app:** la aplicación revisa que los campos obligatorios estén completos antes de enviar los datos.
- **Protección del Webhook:** solo acepta solicitudes con credenciales válidas.
- **Gestión segura de claves:** las credenciales de Google y OpenAI se almacenan de forma segura, nunca dentro del código.
- **Control de duplicados:** cada registro incluye un identificador único (por ejemplo, el ID del dispositivo o timestamp).

## 7. Manejo de errores

Diseñamos el flujo para detectar y responder ante fallos:

- Si la conexión desde la app falla, se muestra un mensaje claro al usuario.
- Si el modelo devuelve una validación negativa, no se guarda el registro.
- Si ocurre un error con Google Sheets o Gmail, el flujo puede reintentar automáticamente.
- Todos los errores quedan registrados para análisis posterior.

## 8. Monitoreo

Cada registro se identifica de forma única, lo que nos permite seguir su recorrido completo:

- Desde que el usuario lo envía desde la app.
- Hasta que se guarda en Google Sheets y se envía el correo.

Esto facilita detectar errores, medir el rendimiento del flujo y garantizar que todos los registros fueron procesados correctamente.

## 9. Beneficios del diseño

- **Integración directa con nuestra app Android.**
- **Automatización completa** del registro y la notificación.
- **Validación inteligente** de datos mediante IA.
- **Trazabilidad total**, gracias al registro en Google Sheets.
- **Comunicación inmediata** con el usuario a través de Gmail.

## 10. Lecciones aprendidas

- Documentar mientras desarrollamos fue clave para mantener claridad técnica y funcional.
- La conexión entre la app Android y el webhook debe manejar correctamente los errores de red.
- El modelo de IA mejora mucho la calidad de los datos, pero requiere prompts bien diseñados.
- Google Sheets funciona bien para comenzar, aunque planeamos migrar a una base de datos si el sistema crece.

## 11. Posibles mejoras futuras

- Implementar una base de datos más robusta para el almacenamiento
- Incorporar métricas en tiempo real sobre los registros.
- Añadir control de versiones a los prompts del modelo de IA.
- Mejorar la interfaz en la app para mostrar detalles del estado del registro.
- Integrar un panel web para monitorear el flujo completo.



## 12. Conclusión

Nuestro flujo de **Registro de Usuarios** conecta directamente una **app Android** con un sistema automatizado en la nube que valida, registra y notifica cada registro sin intervención manual.

El diseño prioriza la **eficiencia, seguridad y trazabilidad**, permitiendo mantener la calidad de los datos y la comunicación constante con los usuarios.

Documentamos cada etapa mientras desarrollábamos, asegurando que las decisiones técnicas queden registradas y justificadas para futuras mejoras o auditorías.