

## Toma de reclamos, derivación y respuesta

### Manual operativo

Reclamos desde Google Forms: clasificación por IA, derivación por correo y respuesta automática al vecino

#### Objetivo

Automatizar el circuito de reclamos ingresados por un formulario. Cada envío se clasifica por tema con IA, se deriva por email al área correspondiente y, si el vecino dejó un correo válido, recibe una respuesta breve de recepción con pedido de datos faltantes cuando aplique.

#### Descripción funcional del script

El script se ejecuta automáticamente al enviarse el formulario mediante un disparador de tipo onFormSubmit. Toma los valores enviados, clasifica el reclamo con OpenAI en un JSON estructurado (tema, subtema, intención, urgencia, resumen y confianza), envía un correo al área municipal destino con todos los datos del reclamo y la clasificación, y luego envía un acuse de recepción al vecino.

#### Acciones que realiza en cada envío

1. Lee los campos del formulario (marca temporal, nombre, teléfono, email, dirección, barrio, descripción y URL de foto).
2. Clasifica el reclamo con IA usando descripción, dirección y barrio.
3. Resuelve el destinatario interno (área) a partir del tema clasificado y el mapa DESTINOS.
4. Envía un correo al área con todos los datos del reclamo y la clasificación IA.
5. Si el vecino dejó un email válido, genera una respuesta breve con IA (máximo 70 palabras) y se la envía.
6. Si falta un dato relevante (por ejemplo dirección, CUIT, teléfono) lo detecta y lo solicita en la respuesta al vecino.

## Resultado esperado

Para cada reclamo enviado por el formulario, el área municipal recibe un email con el caso y su clasificación. El vecino recibe un acuse breve, y la organización opera con una mesa de entradas más ordenada, sin intervención manual.

## Requisitos

Cuenta Google gratuita (Gmail, Forms, Sheets y Apps Script).

Una API key de OpenAI para clasificación y redacción de respuestas.

## Permisos

En la primera ejecución Google solicitará permisos. Este script suele requerir Gmail (enviar correos) y UrlFetch (llamadas a OpenAI). Para el taller es recomendable usar una cuenta nueva y exclusiva para esta automatización, y otorgar permisos completos.

## Recursos a crear

### Formulario de Google

Crear un Google Form para reclamos. El script asume un orden específico de columnas en la respuesta. Ese orden debe coincidir con los índices configurados en el código.

### Hoja de respuestas del formulario

Vincular el formulario a una hoja de cálculo de respuestas. El disparador onFormSubmit se configura en el proyecto Apps Script ligado a esa hoja.

### Proyecto de Apps Script

Abrir la hoja de respuestas y entrar por Extensiones -> Apps Script. Pegar el script y guardar.

## Configuración del script

### Mapa de destinos por tema

Completar el objeto DESTINOS con los correos reales de cada área. El tema clasificado debe coincidir con las claves del mapa.

**DESTINOS:** Mapa tema -> email del área. Incluye 'Otros' como fallback.

### Índices de columnas del formulario

El script toma los campos de e.values en posiciones fijas. Si cambias el formulario o reordenas columnas, ajusta los índices.

1. 0 Marca temporal
2. 1 Nombre y apellido
3. 2 Teléfono
4. 3 Correo electrónico
5. 4 Dirección (calle y número)
6. 5 Barrio
7. 6 Descripción del reclamo
8. 7 Agregar foto (1 como máximo) (URL o vacío)

### **OpenAI API key en Apps Script**

El script lee la API key desde Propiedades del script con el nombre OPENAI\_API\_KEY. No se recomienda pegar la clave dentro del código.

#### **Cómo cargar OPENAI\_API\_KEY**

1. Abrir el editor de Apps Script.
2. Ir a Configuración del proyecto (o Proyecto -> Propiedades del proyecto).
3. Entrar a Propiedades del script.
4. Agregar una propiedad con clave OPENAI\_API\_KEY y pegar el valor sk-...
5. Guardar.

### **Activación automática con disparador**

Este script está pensado para ejecutarse en el momento del envío del formulario. La función principal es onFormSubmit(e).

#### **Crear el disparador onFormSubmit**

1. En Apps Script, abrir Activadores.
2. Añadir activador.
3. Elegir función: onFormSubmit.
4. Origen del evento: Desde hoja de cálculo (o Desde formulario, según interfaz).

5. Tipo de evento: Al enviar formulario (On form submit).

6. Guardar y autorizar cuando Google lo solicite.

## Pruebas rápidas

Realizar varios envíos de prueba con descripciones distintas y datos incompletos para observar pedidos de información y derivaciones.

## Casos sugeridos

1. Reclamo de bache con dirección completa y barrio (esperado: Obras).
2. Luminaria o recolección con dirección (esperado: Servicios Públicos).
3. Semáforo o señalización sin intersección (esperado: Tránsito y pedido de ubicación precisa).
4. Habilitación comercial sin CUIT/razón social (esperado: Habilitaciones y pedido de CUIT o razón social).
5. Asistencia social o salud sin teléfono (esperado: Desarrollo Social o Salud y pedido de teléfono).

## Qué verificar

1. El área destino recibe un email con asunto 'Reclamo <Tema> | <Urgencia> | <Barrio>' y cuerpo con datos y clasificación.
2. El vecino recibe respuesta si el email es válido (contiene '@').
3. La respuesta al vecino no supera 70 palabras.
4. Cuando faltan datos, la respuesta solicita el dato faltante adecuado.

## Criterios de clasificación y respuesta

### Temas e intención

Temas esperados: Obras, Servicios Públicos, Tránsito, Habilitaciones, Desarrollo Social, Salud, Atención al Vecino, Otros. Intención esperada: Reclamo, Consulta, Denuncia, Turno, Sugerencia, Otro.

### Detección de dato faltante

Reglas principales: si falta dirección se pide dirección; en Habilitaciones se pide CUIT o razón social si no aparece; en Tránsito se pide calle y altura o intersección si falta precisión; en Desarrollo Social o Salud se pide teléfono si no está informado.

## Errores frecuentes y resolución

### Falta OPENAI\_API\_KEY

Síntoma: error 'Falta OPENAI\_API\_KEY en Script Properties'. Solución: cargar OPENAI\_API\_KEY en Propiedades del script.

### Orden de columnas distinto al esperado

Síntoma: datos cruzados (por ejemplo barrio en lugar de dirección). Solución: ajustar los índices row[0..7] para que coincidan con el orden real en la hoja de respuestas.

### Email inválido o vacío

Síntoma: no se envía respuesta al vecino. Solución: asegurar campo de email y validación en el formulario; el script ya controla que incluya '@'.

### Destinos incompletos

Síntoma: derivación a 'Otros' cuando no existe una clave para el tema. Solución: completar DESTINOS con todas las áreas y mantener 'Otros' como fallback.

## Checklist de implementación

1. Formulario creado y vinculado a hoja de respuestas.
2. Proyecto Apps Script creado desde la hoja de respuestas.
3. DESTINOS actualizado con correos reales de áreas.
4. OPENAI\_API\_KEY cargada en Propiedades del script.
5. Disparador onFormSubmit configurado y autorizado.
6. Envíos de prueba verificados (correo interno y respuesta al vecino).

**Script:** Ver archive Reclamos\_derivacion\_y\_respuesta.txt