



BOT DE TELEGRAM CON INTELIGENCIA ARTIFICIAL

Desarrollado por Tomas Cano Nieto – Alumno en Prácticas

(Organismo Autónomo Informática del Ayuntamiento de Madrid)

Escuela Madrid Talento (IAM)

Documentación de Uso de la API SIC del Ayuntamiento de Madrid en el Bot

1. Descripción General

Este proyecto utiliza la API SIC (Sistema de Información Ciudadana) del Ayuntamiento de Madrid en su entorno de **preproducción** (PRE) para **enviar avisos y peticiones** mediante peticiones HTTP POST. La finalidad es **probar la correcta recepción y procesamiento de incidencias** por parte del sistema municipal.

2. Fuentes y Herramientas

- **Web de pruebas para creación de avisos:**
<https://sic-frontpre.madrid.es/management/list>
- **Documentación oficial de la API SIC (PRE):**
<https://servpubpre.madrid.es/AVSICAPIINT/doc/>
- **Inspección de red en navegador:**
Se ha utilizado la herramienta de desarrolladores del navegador (apartado "Network") para capturar las estructuras JSON generadas por la web de pruebas.

3. Funcionamiento del Proyecto

3.1 Clasificación del mensaje

1. El usuario envía un **mensaje o imagen**.
2. Se clasifica usando **OpenAI** para extraer:
 - a. Tipo: aviso o petición
 - b. Categoría
 - c. Subcategoría
3. Se valida esta clasificación usando diccionarios locales (AVISOS_PRUEBA y PETICIONES_PRUEBA).

3.2 Ubicación y Datos

4. Se solicita la **ubicación GPS del usuario**.
5. Se formatea un **payload JSON** compatible con la API SIC, incluyendo:
 - a. Coordenadas

- b. Descripción traducida al español
- c. Información adicional simulada (como barrio, distrito, código postal)

3.3 Envío a la API SIC

6. Se envía una petición POST al endpoint:

POST
https://servpubpre.madrid.es/AVSICAPIINT/requests?jurisdiction_id=es.madrid&return_data=false

7. Se incluye el encabezado Authorization: Bearer {TOKEN} y Content-Type: application/json.

3.4 Validación

- Se valida que la ubicación esté **dentro de la ciudad de Madrid**.
- Se comprueba que la petición haya sido **aceptada correctamente** por la API.

4. Particularidades Técnicas

- El ID del service_id usado para simulación es fijo: "591b36544e4ea839018b4653".
- El JSON enviado incluye campos que **fueron extraídos manualmente** desde la interfaz web de PRE mediante inspección.
- La estructura del payload cambia dependiendo de si se han definido todos los campos extendidos (como barrio, distrito, etc.).
- Se emplean diccionarios locales para vincular subcategorías a sus IDs correspondientes.

5. Archivos y Dependencias

- `diccionarios.py`: contiene los diccionarios de pruebas (AVISOS_PRUEBA, PETICIONES_PRUEBA).
- `claves.py`: contiene las claves necesarias para operar (API_KEY, TOKEN, etc.).
- `openai`, `telegram`, `requests`: librerías principales usadas para procesamiento, interacción y envío de datos.

6. Conclusión

El desarrollo aprovecha el entorno de PRE de la API SIC para realizar pruebas completas de ciclo de reporte automatizado de incidencias ciudadanas. Se ha hecho ingeniería inversa del formulario web de avisos para replicar el formato JSON correcto. El sistema permite pruebas seguras y controladas sin impactar el entorno de producción. También se reduce el tiempo a la hora de enviar avisos/peticiones respecto a '**Madrid Móvil**' o en la web de '<https://avisos.madrid.es/>' que puedes tardar entre 2-5 min en completar un reporte y que puede que el mismo reporte que se envíe con esas plataformas se envíen de manera errónea porque no corresponde con la categoría/subcategoría.