



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

**ASISTENTE VIRTUAL (CHATBOT) MI CALI BELLA PARA LA AUTOMATIZACIÓN Y MEJORA DE GESTIÓN DE ENCUESTAS E INTERACCION CON LA IA, RESPECTO A CANTIDADES UTILIZANDO CONDICIONES Y EMITIENDO UN RESUMEN DONDE LA INTELIGENCIA ARTIFICIAL MACHINE LEARNING, RESPONDE DETALLADAMENTE.**

Corporación Universitaria Remington.  
Ingeniería De Sistemas.  
Seminario AI.

Estudiante: Daniel Alejandro Fajardo Gamboa.

Instructor: Danny López Segura

Opción de Trabajo de grado Seminario-Diplomado.  
2025.

### **Dedicatoria**

Dedicado A Todos Aquellos que Creyeron en mí, A mis padres que con sus Palabras y su apoyo me dieron la determinación de seguir. A mi esposa que dedico parte de su tiempo para escuchar mis ideas en proceso. A mi Hijo que se fascina cada vez que me desempeño en el área De sistemas. A mi Abuelo y mi Abuela, a mi Tía que sin su inmenso apoyo y motivación no hubiera sido posible terminar este Proyecto.

### **Agradecimientos**

A La Corporación Universitaria Remington Por los conocimientos Adquiridos A los profesores Compañeros de estudio que me acompañaron en el proceso y crecí de manera personal y profesional. A los Compañeros de Trabajo que con sus opiniones hicieron que este Proceso valiera la pena al Saber Que este Proyecto se Desarrolló para automatizar un proceso empresarial.

## Tabla de Contenidos

<a href="#">Resumen</a> .....	5
Palabra clave.....	6
Pregunta orientadora de la búsqueda .....	6
1. Mi CALIBELLA: Asistente AI para el análisis de encuestas Ciudadanas en Cali.....	7
2. objetivos.....	7
2.1 justificacion.....	7
2.2 alcance.....	8
2.3 plublico.....	8
3. Automatizacion de la consulta de Datos de Encuestas Usando la API de open AI.....	9
3.1 Automatizacion .....	9
3.1.1 Consulta de Datos de Encuesta.....	9
3.1.2 API de Open AI .....	10
3.1.3google Sheets. ....	10
3.2 Asistente Virtual Chat boot MICALIBELLA.....	10
Paso 1 .....	11
Paso 9 .....	16
3.3 pruebas.....	17
3.3.1 Visualizacion de la plataforma Web.....	18
Figuras y Tablas .....	22
Conclusiones.....	23
Referencias.....	24
Anexos .....	25

## Resumen

El presente trabajo de grado surge a partir de la necesidad de tener al alcance de la mano el análisis de encuestas ciudadanas en Cali, ante el incremento en la cantidad de datos y las limitaciones de los métodos tradicionales. Como respuesta, se plantea la integración de nuevas tecnologías que agilicen, automaticen y mejoren la eficiencia del procesamiento de información.

ChatBot MI CALI BELLA Asistente Virtual creado con machine Learning con la Api de OpenAI, Para la aplicación de inteligencia artificial en la automatización, aconseja y da mejoras de la gestión de encuestas según las cantidades obtenidas o resultados de las mismas.

Este proyecto tiene como objetivo hacer más fácil y rápida la gestión de encuestas, no solo obteniendo los resultados o cantidades si no que machine Learning detalla y aconseja cuales son los puntos referentes a mejorar y o a mantener. Lo que quiero lograr es automatizar el proceso para que, en lugar de responder manualmente a cada encuesta, la inteligencia artificial pueda buscar la información en las hojas de Google y dar respuestas automáticas al menú de preguntas que se han identificado son concurrentes y necesitan respuesta inmediata. Esto ayudara a que todo el proceso sea más eficiente, ahorrando tiempo y evitando errores; obteniendo el concejo de la IA para diseñar estrategias que nos ayuden a mejorar.

El objetivo principal es reflexionar sobre las estrategias metodológicas que, mediante el uso de tecnologías de la información como lo es hoy la inteligencia artificial, En este lugar, se presenta el proyecto MI CALI BELLA un asistente virtual “ChatBot” conectada a una base de datos de encuestas por medio de la API Google Sheets y la API de Open AI (ChatGPT-3.5-turbo) diseñada para interpretar e interactuar con resultados de encuesta de satisfacción de manera fácil entendible de manera natural e inteligente.

La metodología implementada es utilizar condiciones de Algoritmos a la Base de datos entregando un resumen y machine Learning interprete los datos y responda detalladamente diciendo donde se realizó mayor y menor cantidad de encuestas. La IA nos recomienda técnicas para aumentar el nivel de encuestas para futuras mejoras esta tiene una conexión a la base de datos por medio de la API donde toma las encuestas como referencia y el modelo de inteligencia artificial GPT-3.5-turbo el cual es un modelo ya entrenado responde de manera acertada con un lenguaje comprensible de lo que queremos saber.

Los resultados esperados es que se pueda analizar grandes volúmenes de datos para mejorar la consulta de estos en el menor tiempo posible y de manera exacta.

que las cantidades de encuestas realizadas, la IA las interprete nos detalle y nos aconseje el lugar de mayor y menor impacto de encuestas siendo un sistema capaz de entender y aconsejar posibles mejoras.

Impacto: académico y práctico esta propuesta contribuye en lo académico demostrando que las nuevas tecnologías y la inteligencia artificial puede transformar la manera de trabajar con bases de datos volviéndolo más cómodo y seguro para trabajar.

En lo práctico Mi Cali bella ayuda a procesar las encuestas de manera rápida y confiable para reducir el tiempo de consulta que se requiera, y mitigar la pregunta recurrente del cliente en decir cuantas encuestas llevamos y que día se hicieron.

### **Palabras clave**

Nuevas tecnologías, inteligencia artificial, Chatbot, automatización, análisis, encuesta de satisfacción, machine Learning

### **Pregunta orientadora de la búsqueda**

Desarrollar la problemática

En mi entorno laboral se trabaja con encuestas de satisfacción sobre el espacio público de la ciudad de Cali. Donde la dificultad constante está relacionada con el seguimiento y la contabilización diaria de encuestas de satisfacción.

Diariamente, se realiza la misma pregunta: ¿cuántas encuestas se han llevado a cabo hasta el momento y en qué lugar y fecha? Este proceso, aunque aparentemente simple, implica revisar manualmente bases de datos en hojas de cálculo. Replanteando que con la respuesta nos demos cuenta de qué lugar se necesita visitar más o menos. Ya que se debe tener en cuenta todo el territorio de la ciudad de Cali para que con ello puedan atender lugares que no se visitan regularmente y poder tener un porcentaje casi igual en todos los sectores donde se realizan las encuestas.

2. Mi CALIBELLA: Asistente AI para el análisis de encuestas Ciudadanas en Cali

Asistente virtual (ChatBot) mi CALIBELLA para la automatización y mejora de gestión de encuestas e interacción con la IA, respecto a cantidades utilizando condiciones y emitiendo un resumen donde la inteligencia artificial machine Learning, responde detalladamente, y nos da su enfoque diciendo el lugar de mayor y menor donde se han realizado las encuestas de satisfacción.

### **3. Objetivo General**

El proyecto MI CALI BELLA tiene como propósito ofrecer una solución interactiva basada en inteligencia artificial para analizar encuestas ciudadanas recolectadas en la ciudad de Cali. Esta herramienta permite a los funcionarios interpreten datos de forma amigable, rápida y eficaz, sin necesidad de conocimientos técnicos avanzados en bases de datos “Tener a la mano la información”.

#### **2.1 Justificación**

Actualmente, los resultados de encuestas ciudadanas suelen quedar relegados a bases de datos complejas o informes poco accesibles para el personal de la empresa. MI CALI BELLA busca implementar un Chatbot, brindando un canal conversacional mediante el cual cualquier persona de la empresa puede hacer preguntas específicas sobre los datos recolectados y recibir respuestas.

#### **2.2 Alcance**

MI CALI BELLA analiza información almacenada en bases de datos de encuestas, detalla y proporciona respuestas informadas. Esta primera versión se enfoca en datos de la ciudad de Cali, pero su estructura permite escalar el sistema para otras regiones o temáticas.

### **2.3 Público Objetivo**

- ✚ Personal de empresa
- ✚ Investigadores, docentes y estudiantes
- ✚ Entidades públicas (UAESP) unidad administrativa de servicios públicos que requieren herramientas para visualizar el sentir ciudadano
- ✚ El personal de comunicaciones el cual desea acceder a datos y poder representarlo en sus flayers de manera interactiva

## **4. Automatización de la Consulta de Datos de Encuestas Usando machine learning la API de OpenAI y Google Sheets**

Este proyecto se centra en **automatizar** el proceso de consulta de datos de encuestas de satisfacción, es decir, en reemplazar las tareas manuales de recopilación y análisis con un sistema que realice estas funciones de manera automática y eficiente.

**3.1. Automatización** : Se busca reducir la intervención nosotros como departamento de TI en consulta de los datos, agilizando el proceso mediante el uso de tecnologías de la información tal como la AI (Inteligencia Artificial). En lugar de revisar hojas de cálculo y

procesar los datos manualmente, el ChatBot MICALIBELLA automatiza Responde y detalla los resultados sobre el contenido de las encuestas.

**3.1.1. Consulta de Datos de Encuestas:** Las encuestas de satisfacción son una herramienta clave para conocer la opinión del ciudadano. Estas son tomadas en la Ciudad de Cali en formato físico y luego llegan a la oficina donde son digitalizadas, estas encuestas necesitan una validación de datos para unificar nombre de barrios, comunas, quitar espacios vacíos haciendo de ella una base de datos sólida que nos brindara la información necesaria para que la inteligencia artificial detalle e informe puntos de mayor y menor encuestas y técnicas para mejorar en un lenguaje profesional y entendible.

**3.1.2. API de OpenAI:** La inteligencia artificial juega un papel esencial en este proceso. Utilizando la **API de OpenAI**, usando el Modelo Chat Gpt-3.5-Turbo modelo económico y eficaz el cual puede comprender preguntas en lenguaje natural, procesar los datos correspondientes y proporcionar respuestas en tiempo real, sin necesidad de intervención adicional.

**3.1.3. Google Sheets:** Este sistema de hojas de cálculo colaborativo se utiliza como la base de datos donde se almacenan las respuestas de las encuestas. Google Sheets ofrece accesibilidad en la nube, lo que permite que múltiples usuarios puedan interactuar con los datos y que la IA pueda acceder a la información de manera eficiente a través de su API. La integración entre Google Sheets y la API de OpenAI garantiza que los datos se mantengan actualizados y accesibles de forma constante.

### 3.2 Asistente Virtual “Chatbot Mi cali bella”

Para desarrollar el proyecto trabajamos en el laboratorio de Google Colab, iniciamos instalando Open AI que será el modelo de inteligencia artificial y Gradio, la interfaz grafica web que nos permitirá visualizar nuestro Chatbot de una manera interativa y fácil de comprender.

```

colab.research.google.com/drive/1f7u4BaysmVMxhBWGTHdgvRbyDL4AQ5Yj
IAPRO2025
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Comandos + Código + Texto

!pip install openai==0.28
!pip install Gradio

Requirement already satisfied: openai==0.28 in /usr/local/lib/python3.11/dist-packages (0.28.0)
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.11/dist-packages (from openai==0.28) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from openai==0.28) (4.67.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from openai==0.28) (3.11.15)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.20->openai==0.28) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.20->openai==0.28) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.20->openai==0.28) (2.4.0)
Requirement already satisfied: certifi=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.20->openai==0.28) (2025.4.26)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (2.6.1)
Requirement already satisfied: aiosignal=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (25.3.0)
Requirement already satisfied: frozenlist=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (1.6.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (6.4.3)
Requirement already satisfied: propcache=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (1.20.0)
Collecting Gradio
  Downloading gradio-5.29.0-py3-none-any.whl.metadata (16 kB)
Collecting aiofiles<25.0,>=22.0 (from Gradio)
  Downloading aiofiles-24.1.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages (from Gradio) (4.9.0)
Collecting fastapi<1.0,>=0.115.2 (from Gradio)
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Collecting ffmpy (from Gradio)
  Downloading ffmpy-0.5.0-py3-none-any.whl.metadata (3.0 kB)
Collecting gradio-client=1.10.0 (from Gradio)
  Downloading gradio_client-1.10.0-py3-none-any.whl.metadata (7.1 kB)
Collecting groovy~=0.1 (from Gradio)
  Downloading groovy-0.1.2-py3-none-any.whl.metadata (6.1 kB)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-packages (from Gradio) (0.28.1)
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.11/dist-packages (from Gradio) (0.30.2)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages (from Gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist-packages (from Gradio) (3.0.2)

Requirement already satisfied: mdurl=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->Gradio) (0.1.2)
Downloading gradio-5.29.0-py3-none-any.whl (54.1 MB)
 54.1/54.1 MB 19.2 MB/s eta 0:00:00
Downloading gradio_client-1.10.0-py3-none-any.whl (322 kB)
 322.9/322.9 kB 25.0 MB/s eta 0:00:00
Downloading aiofiles-24.1.0-py3-none-any.whl (15 kB)
 15.0/15.0 kB 7.4 MB/s eta 0:00:00
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
 95.2/95.2 kB 7.4 MB/s eta 0:00:00
Downloading groovy-0.1.2-py3-none-any.whl (14 kB)
 14.0/14.0 kB 7.4 MB/s eta 0:00:00
Downloading python_multipart-0.0.20-py3-none-any.whl (24 kB)
 24.0/24.0 kB 7.4 MB/s eta 0:00:00
Downloading ruff-0.11.8-py3-none-manylinux_2_17_x86_64_muslinux2014_x86_64.whl (11.5 MB)
 11.5/11.5 MB 81.1 MB/s eta 0:00:00
Downloading safetextpx-0.1.6-py3-none-any.whl (8.7 kB)
 8.7/8.7 kB 7.4 MB/s eta 0:00:00
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
 15.0/15.0 kB 7.4 MB/s eta 0:00:00
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
 72.0/72.0 kB 5.5 MB/s eta 0:00:00
Downloading tomkit-0.13.2-py3-none-any.whl (37 kB)
 37.0/37.0 kB 5.5 MB/s eta 0:00:00
Downloading uvicorn-0.34.2-py3-none-any.whl (62 kB)
 62.5/62.5 kB 4.9 MB/s eta 0:00:00
Downloading ffmpy-0.5.0-py3-none-any.whl (6.0 kB)
 6.0/6.0 kB 4.9 MB/s eta 0:00:00
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
 32.0/32.0 kB 4.9 MB/s eta 0:00:00
Installing collected packages: pydub, uvicorn, tomkit, semantic-version, ruff, python-multipart, groovy, ffmpy, aiofiles, starlette, safetextpx, gradio-client, fastapi, Gradio
Successfully installed Gradio-5.29.0 aiofiles-24.1.0 fastapi-0.115.12 ffmpy-0.5.0 gradio-client-1.10.0 groovy-0.1.2 pydub-0.25.1 python-multipart-0.0.20 ruff-0.11.8 safetextpx-0.1.6 semantic-version-2.10.0

```

Figura No1 Instalacion librerias correspondientes a Open AI y Gradio.

## Paso 2: Conexión a OpenAI

La segunda parte del código se encarga de conectar a la **API de OpenAI**, utilizando la clave de API, que nombre `openai_key`. Mediante esta conexión, se puede hacer uso de los modelos de lenguaje de OpenAI.

La API de OpenAI permite generar respuestas inteligentes y coherentes basadas en texto, lo que se convierte en una herramienta buena para asistir por medio del chatbot a los usuarios en la obtención de respuestas sin intervención manual.

```
import json
from google.colab import userdata
from google.oauth2 import service_account
from googleapiclient.discovery import build
import openai
import pandas as pd
import gradio as gr
import os
```

*Figura No2 Instalacion librerias necesarias en el proyecto*

`import json`: Necesaria para key o llave

`from google.colab import userdata`: Accede a datos privados.

`from google.oauth2 import service_account`: nos permite conexión con Google

`from googleapiclient.discovery import build`: se Conecta con Google sheet las credenciales de servicio.

`import openai`: Modelo entrenado

`import pandas as pd`: Utilizado en python para la visualizacion o esquema tipo excel.

`import gradio as gr`: Interfaz grafica web que nos permite la visualizacion de nuestro chatbot para interactuar con el usuario de manera facil.

`import os`: Manipula el entorno del sistema.

### Paso 3. Secret Colab (llave API)

Para las claves API e implementado en el proyecto usar la llave secreta que nos permite Colab una forma fácil y segura de autenticación para ello voy al menú izquierdo la opción de la llave creo nuevo valor al cual le asigno el nombre APIbella, en el valor colocamos el contenido de la key generada por Google guardo los cambios y lo mismo realizo para la API de open AI.

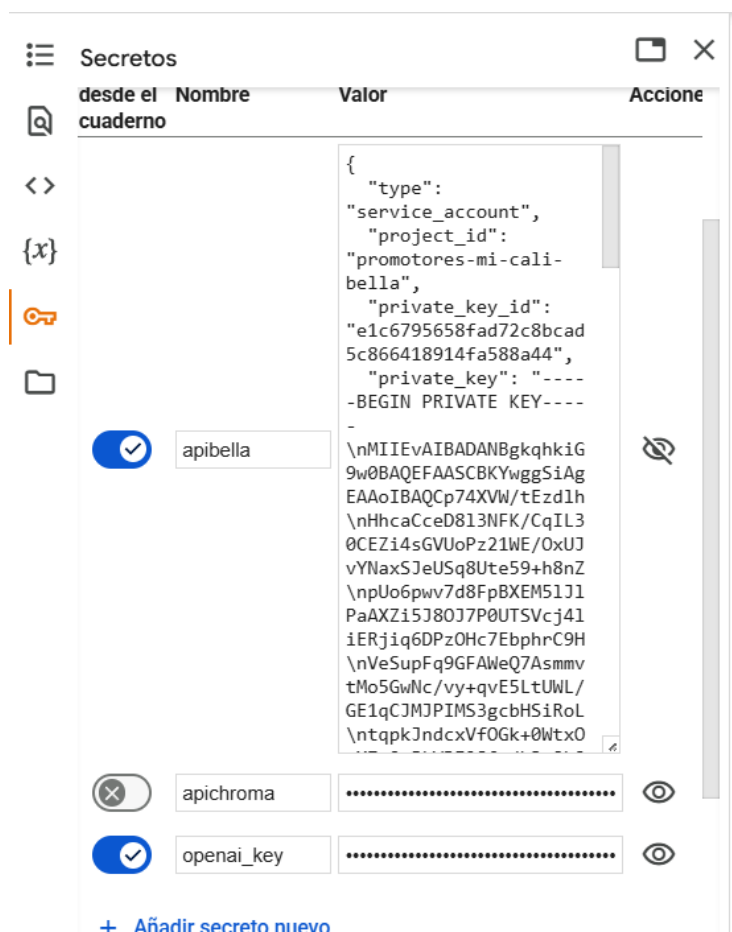


Figura No3 Secret Colab

## Paso 4. Autenticación API Google Sheet y Open AI

La primera fase del código indicamos donde obtener las API apibella y Openai\_key según la parte del código nos indican las credenciales del servicio donde las obtiene Google Sheet en la tercera parte debemos colocar el ide de nuestra hoja Google Sheet, el nombre de la hoja y el nombre de la hoja de la cual queremos obtener los datos.

The screenshot shows a Google Colab notebook interface. On the left, there is a 'Secrets' panel with a table of secrets:

Acceso desde el cuaderno	Nombre	Valor	Acciones
<input checked="" type="checkbox"/>	apibella	.....	
<input type="checkbox"/>	apichroma	.....	
<input checked="" type="checkbox"/>	openai_key	.....	

Below the table are buttons for '+ Añadir secreto nuevo' and 'Claves de API de Gemini'. Below that, a text prompt says 'Accede a tus claves de secretos en Python a través de:' followed by a code block:

```
from google.colab import userdata
userdata.get('secretName')
```

On the right, a code cell contains the following Python code:

```
# Secret Colab
json_key = userdata.get("apibella")
openai_api_key = userdata.get("openai_key")

# Guardar archivo temporal para Google Sheets
with open("apibella.json", "w") as f:
    f.write(json_key)

# Autenticación Google Sheets
SCOPES = ['https://www.googleapis.com/auth/spreadsheets']
credentials = service_account.Credentials.from_service_account_file("apibella.json", scopes=SCOPES)
service = build('sheets', 'v4', credentials=credentials)
sheets = service.spreadsheets()

# Leer datos desde la hoja
sheet_id = '1nuARCPY8efS3cvJgITE_fcmWlnQx0xHb0ZfWutVDIf0'
range_name = 'PEDAGOGIA A PIE DE CALLE'
result = sheets.values().get(sheet_id=sheet_id, range=range_name).execute()
values = result.get('values', [])
```

Figura No4 Autenticación API Google Sheet y Open AI.

## Paso 5. Pandas Dataframe

Este código indica la forma limpia de convertir los datos obtenidos desde Google Sheets en un Data Frame de pandas.

Donde

- La primera fila (values[0]) contiene los **encabezados de columna**.
- Las filas siguientes (values[1:]) contienen los **datos reales**.
- Es igual a las columnas al título y a las filas.

Con este Dataframe mas adelante podemos obtener vista de tabla modo Excel con visualización estadística

```
# DataFrame
headers = values[0]
data_rows = values[1:]
df = pd.DataFrame(data_rows, columns=headers)
```

Figura No 5 Dataframe Pandas.

## Paso 6. Condicional – Resumen o filtro - Machine Learning

Aquí en esta figura podrán ver la manera como colocamos condicionales para filtrar y extraer cantidad de datos por columnas para que así la inteligencia artificial tenga un resumen de las cantidades y con lenguaje natural pueda detallar mayor y menor de encuestas además de aconsejarnos para mejorar.

Con el siguiente código tuve como base para los otros condicionales que realice llevan la misma estructura solo cambia la columna y el enfoque como quiero que responda la inteligencia artificial.

if opcion == "Cantidad de encuestas por BARRIO CUADRANTE":

    if 'BARRIO CUADRANTE' in df.columns:

        resumen = df['BARRIO CUADRANTE'].value\_counts()

        prompt = f"Soy un analista de datos. Aquí está el resumen de cantidad de encuestas por BARRIO CUADRANTE:\n{resumen.to\_string()}\n\n¿Dame un análisis o interpretación identificando el mayor y menor sobre estos datos Y DALE ESTRATEGIAS PARA OBTENER MAS ENCUESTAS O MANTENER EL NIVEL?"

    else: return "la columna 'BARRIO CUADRANTE' no existe en los datos."

```
# GRABO
def analizar_datos(opcion):
    try:
        resumen = None
        prompt = ""

        if opcion == "Cantidad de encuestas por BARRIO CUADRANTE":
            if 'BARRIO CUADRANTE' in df.columns:
                resumen = df['BARRIO CUADRANTE'].value_counts()
                prompt = f"Soy un analista de datos. Aquí está el resumen de cantidad de encuestas por BARRIO CUADRANTE:\n{resumen.to_string()}\n\n¿Dame un análisis o interpretación identificando el mayor y menor sobre estos datos Y DALE ESTRATEGIAS PARA OBTENER MAS ENCUESTAS O MANTENER EL NIVEL?"
            else:
                return "La columna 'BARRIO CUADRANTE' no existe en los datos."

        elif opcion == "Cantidad de encuestas por COMUNA":
            if 'COMUNA' in df.columns:
                resumen = df['COMUNA'].value_counts()
                prompt = f"Soy un analista de datos. Aquí está el resumen de cantidad de encuestas por COMUNA:\n{resumen.to_string()}\n\n¿Dame un análisis o interpretación identificando el mayor y menor sobre estos datos?"
            else:
                return "La columna 'COMUNA' no existe en los datos."

        elif opcion == "Cantidad de encuestas por FECHA":
            if 'FECHA' in df.columns:
                resumen = df['FECHA'].value_counts()
                prompt = f"Soy un analista de datos. Aquí está el resumen de cantidad de encuestas por FECHA:\n{resumen.to_string()}\n\n¿Podrías darme un análisis o interpretación identificando el mayor y menor sobre estos datos?"
            else:
                return "La columna 'FECHA' no existe en los datos."

        elif opcion == "Cantidad de encuestas por OBJETIVO":
            if 'OBJETIVO' in df.columns:
                resumen = df['OBJETIVO'].value_counts()
                prompt = f"Soy un analista de datos. Aquí está el resumen de cantidad de encuestas por OBJETIVO:\n{resumen.to_string()}\n\n¿Dame un análisis o interpretación identificando el mayor y menor sobre estos datos?"
            else:
                return "La columna 'OBJETIVO' no existe en los datos."

        else:
            return "Opción no válida."

    if not prompt:
        return "Error: el prompt está vacío."
```

Figura No 6 Condicional- Machine learning

## Paso 7. Estructura modelo Chat GPT-3.5-turbo

Tenemos la estructura GPT-3.5-turbo el cual toma el resumen de los condicionales que hemos creado en el paso anterior y responde e interpreta ejecutando un resultado con base a las encuestas de satisfacción.

Lo realizamos de esta forma debido a que la API de open AI es paga y consumir el Qwery en una base de datos tan extensa seria costoso, en cambio con ayuda de los condicionales filtramos y obtenemos información resumida que openai o GPT-3.5-turbo modelo entrenado responderá de manera adecuada.

```
# openai
response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": "Eres un experto analista de datos urbanos especializado en encuestas ciudadanas."},
        {"role": "user", "content": prompt}
    ],
    max_tokens=500,
    temperature=0.3
)
interpretacion = response['choices'][0]['message']['content'].strip()

resultado = f"### Resumen de datos:\n```\n{resumen.to_string()}\n```\n\n### MICALIBELLA ANALIZA Y TE DICE... :\n{interpretacion}"
return resultado

except Exception as e:
    return f"! Error al obtener análisis: {str(e)}"
```

*Figura No 7 GPT-3.5-turbo que obtiene el resumen e interpreta y da respuesta.*

## Paso 8. Tabla de Pandas

En los anteriores pasos creamos el dataframe ese es indispensable para ejecutar el siguiente comando que nos permitirá una vista tipo Excel sobre los datos recolectados de encuestas de satisfacción. Esta nos permite imprimir la cantidad de filas que requerimos en este caso colocamos 6 filas.

```
# display pandas

print("\n Vista previa de los datos:")
display(df.head(6))
# cantidad filas
total_encuestas = len(df)

print(f" El total de encuestas registradas es: {total_encuestas}")
```

Vista previa de los datos:

ID	FECHA	NOMBRE DEL PROMOTOR	NOMBRE DEL ENCUESTADO	TÉLEFONO	E-MAIL	BARRIO CUADRANTE	Latitude	Longitud	COMUNA	CUADRANTE	INFORMACIÓN MENORADA ES ÓPTIL PARA SENSIBILIZAR Y EDUCAR	INFORMACIÓN PRE CLARA, PRECISA Y PERTINENTE	TRATO FUE AMABLE Y RESPETUOSO	SENTIDO DE APROPIACIÓN DEL ESPACIO PÚBLICO POR PARTE DE LA COMUNIDAD DEL SECTOR	FIN O ACTIVIDAD	OBJETIVO			
0	2025-02-26	MARIN MEZA NATHALY	MARIANA MEJIA	3152970339	N/A	Santa Teresita	3.448480000	-76.552435700	2	CARRERA 2A OESTE	4	4	4	4	4	REQUERIMIENTO POR PARTE DE SECRETERIA DE SEGUR...			
1	2025-02-26	MARIN MEZA NATHALY	SHARID SOLANO	3155746157	N/A	Santa Teresita	3.448480000	-76.552435700	2	CARRERA 2A OESTE	4	4	4	4	3	PEDAGOGICO	REQUERIMIENTO POR PARTE DE SECRETERIA DE SEGUR...		
2	2025-02-26	MARIN MEZA NATHALY	CRISTIAN LOZANO	318224702	N/A	Santa Teresita	3.448480000	-76.552435700	2	CARRERA 2A OESTE	4	4	4	4	4	4	PEDAGOGICO	REQUERIMIENTO POR PARTE DE SECRETERIA DE SEGUR...	
3	2025-02-26	MARIN MEZA NATHALY	ANDRES REBDOH	318326716	N/A	Santa Teresita	3.448480000	-76.552435700	2	CARRERA 2A OESTE	4	4	4	4	4	3	PEDAGOGICO	REQUERIMIENTO POR PARTE DE SECRETERIA DE SEGUR...	
4	2025-02-26	MARIN MEZA NATHALY	LUIS TOLOZA	3218138402	N/A	Santa Teresita	3.448480000	-76.552435700	2	CARRERA 2A OESTE	4	4	4	4	4	4	4	PEDAGOGICO	REQUERIMIENTO POR PARTE DE SECRETERIA DE SEGUR...
5	2025-02-26	MARIN MEZA NATHALY	MANUEL QUINTERO	3180102114	N/A	Santa Teresita	3.448480000	-76.552435700	2	CARRERA 2A OESTE	4	4	4	4	4	4	4	PEDAGOGICO	REQUERIMIENTO POR PARTE DE SECRETERIA DE SEGUR...

El total de encuestas registradas es: 6119

Figura No 8 Representación tablas Pandas.

## Paso 9. Interfaz Grafica web Gradio

Gradio es una herramienta que nos permite representar nuestra información en una pagina web rápida y fácil de comprender donde colocamos nuestro menú de opciones y botón para ejecutar MI CALI BELLA Asistente Virtual Chatboot.

```
# Interfaz Gradio
with gr.Blocks() as demo:
    gr.Markdown("# CHATBOT MICALIBELLA - Análisis de Encuestas Asistente Virtual")
    gr.Markdown(f" Total de encuestas cargadas: {len(df)}")

    opcion_dropdown = gr.Dropdown(
        choices=[
            "Cantidad de encuestas por BARRIO CUADRANTE",
            "Cantidad de encuestas por COMUNA",
            "Cantidad de encuestas por FECHA",
            "Cantidad de encuestas por OBJETIVO"
        ],
        label="Selecciona una opción"
    )

    resultado_texto = gr.Markdown()

    analizar_btn = gr.Button("Analizar con BELLA LA IA DE MICALIBELLA")

    analizar_btn.click(
        fn=analizar_datos,
        inputs=opcion_dropdown,
        outputs=resultado_texto
    )

# Lanzar app Gradio
demo.launch()
```

Figura No 9 Estructura Gradio Representación Grafica Web.

Página principal

Beneficios

28	620
13, 7, 8	665
4, 9	461
7, 0	420
1	366
48765	230
8, 12	380
45046	380
41132	120
ÁREA DE EXPANSIÓN URBANA	160

MICALIBELLA ANALIZA Y TE DICE...:

Claro, basándonos en la cantidad de encuestas por comuna, podemos identificar lo siguiente:

- La comuna con la mayor cantidad de encuestas es la 14, con un total de 4675 encuestas.
- La comuna con la menor cantidad de encuestas es el Área de Expansión Urbana, con solo 100 encuestas.

Esto nos indica que la comuna 14 es la que ha recibido la mayor atención en términos de encuestas, posiblemente debido a su importancia o relevancia dentro de la ciudad. Por otro lado, el bajo número de encuestas en el Área de Expansión Urbana sugiere que esta zona puede no haber sido tan prioritaria en términos de recolección de datos o análisis de la opinión de los ciudadanos.

Analizar con BELLA LA IA DE MICALIBELLA

Usar el API

Construido con Gradio

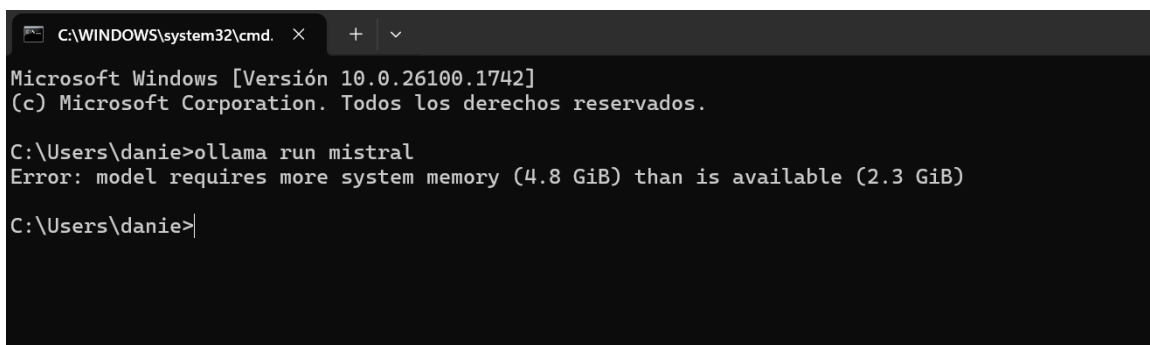
Configuración

### 3.3 Pruebas

#### Capítulo 1: Introducción al Proyecto MI CALI BELLA

ChatBot MI CALI BELLA Asistente Virtual creado con machine Learning con la Api de OpenAI, Para la aplicación de inteligencia artificial en la automatización, aconseja y da mejoras de la gestión de encuestas según las cantidades obtenidas o resultados de las mismas.

Antes de elegir el modelo *GPT-3.5-turbo* trabaje de manera local con ollama e instale el modelo mistral pero mi equipo se quedo sin recursos no lo soportaba para ello minimo debo tener 16 gigas de memoria y GPU osea graficos que permitan que la aplicación o modelo se ejecute satisfactoriamente.



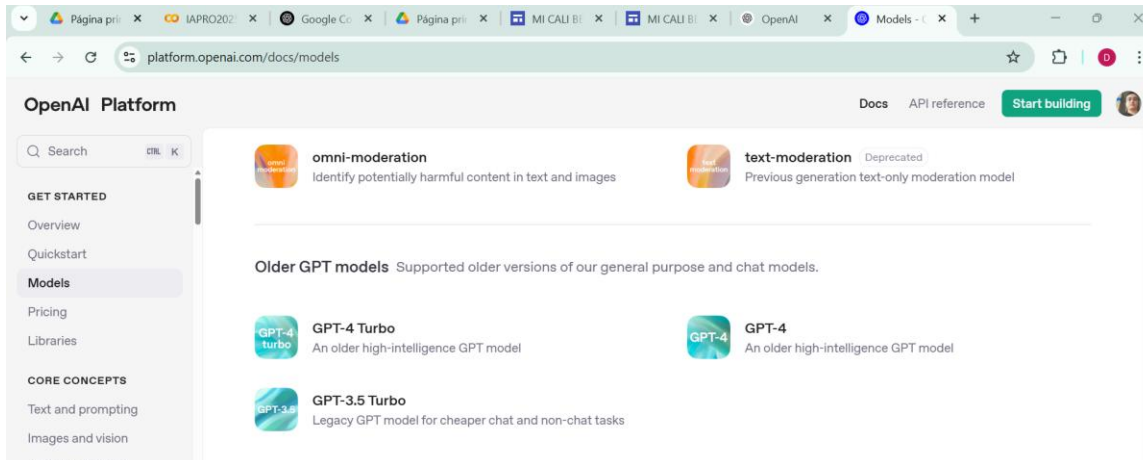
```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Versión 10.0.26100.1742]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\danie>ollama run mistral
Error: model requires more system memory (4.8 GiB) than is available (2.3 GiB)

C:\Users\danie>
```

luego instale el modelo de ollama (tinillama) un modelo mucho mas basico pero me daba respuestas con terminos que no requeria para las encuestas de sastifaccion en definida cuentas pague por utilizar los recursos de Open AI que son muy completos y eficientes que me ayudaron a desempeñar el buen funcionamiento e interpretacion de mi chatbot

CALI BELLA.



### Modelos 3.5-turbo disponibles en Open AI

Escogi GPT-3.5-Turbo basico y economico disponible para Chatboot, luego Se trabajo en google colab donde se instalaron librerias requeridas primero para importar la API de google, API de open AI, utilice la opcion de llave secreta de Colab para guardar el archivo.json que contiene la clave de acceso a la base de datos de Google Sheet y tambien la llave de open AI, e instale la librería.

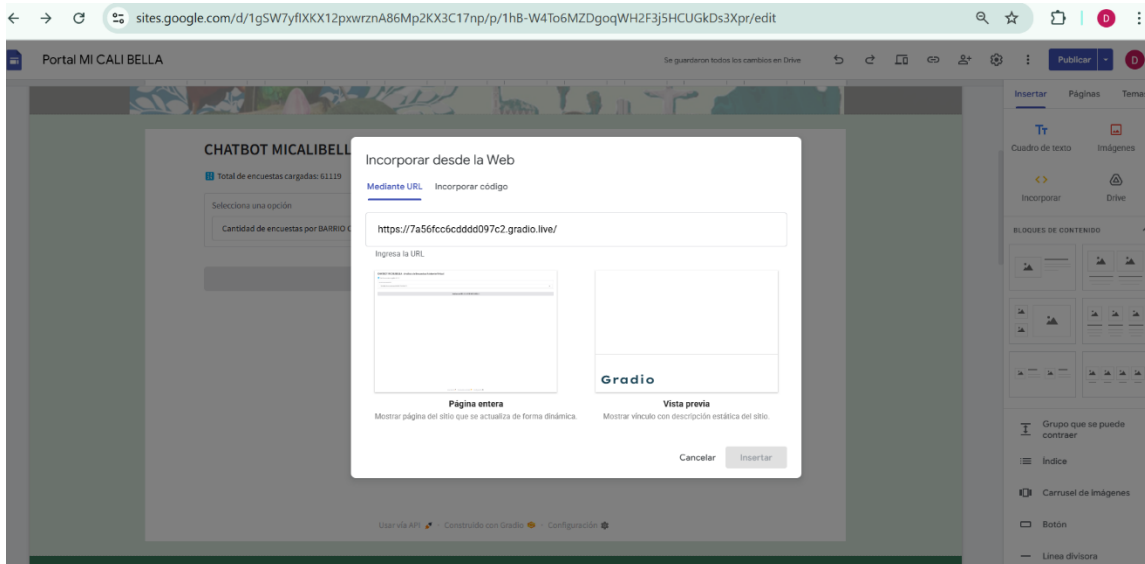
*Me benefició utilizar el modelo cualitativo GPT -3.5-turbo en una interpretación rápida de 4 segundos de respuesta, análisis profundo ya que interpreta lo que quiero saber por medio de otras palabras, ahorro de tiempo y dinero, facilita la toma de decisiones ya que se puede tener un informe detallado y poder interpretarlo.*

A continuación, Ejecución desde Colab.

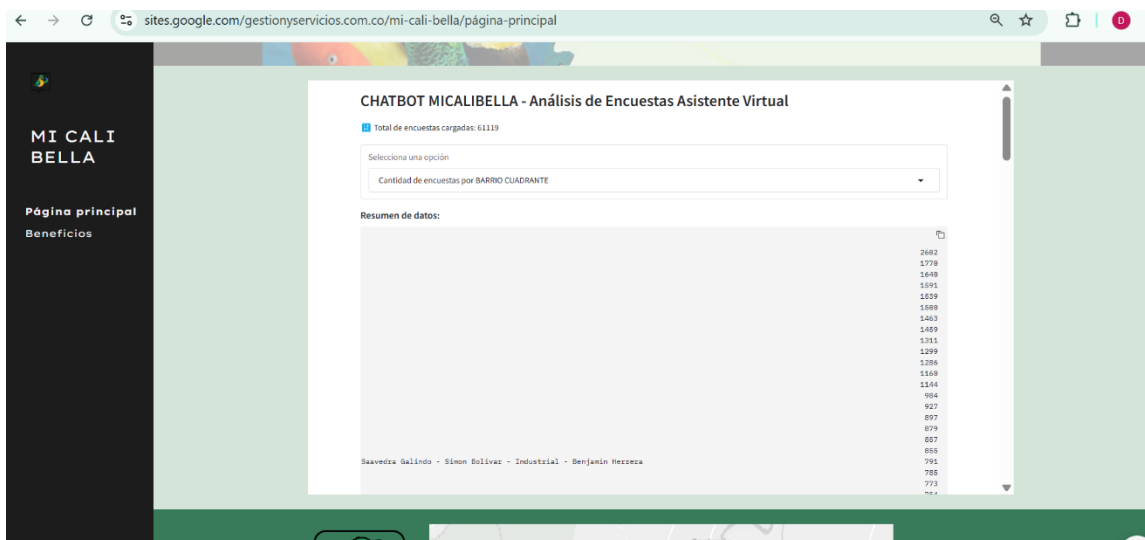
1. Me muestra Tablas o display de Pandas.
2. Me muestra la Cantidad de encuestas Contando el Numero de Filas.
3. Se ejecuta Gradio.

The screenshot shows a Google Colab notebook. The top part displays a pandas DataFrame with columns: ID, FECHA, NOMBRE DEL PROPRIETARIO, NOMBRE DEL ENCUESTADO, TELEFONO, E-MAIL, BARRIO CUADRANTE, Latitude, Longitud, COMUNA, CUADRANTE, INFORMACIÓN BREVEDAD ES ÚTIL PARA MANEJABILIDAD Y EDUCAR, INFORMACIÓN FUE CLARA, PRECISA Y PERTINENTE, TRATO FUE AMABLE Y RESPETUOSO, SENTIDO DE APROPIACIÓN DEL ESPACIO PÚBLICO POR PARTE DE LA COMUNIDAD DEL SECTOR, ETC O ACTIVADO, and OBJETIVO. Below the table, there's a Gradio chatbot interface titled 'CHATBOT MICALIBELLA - Análisis de Encuestas Asistente Virtual'. It shows a dropdown menu for 'Cantidad de encuestas por BARRIO CUADRANTE' and a button 'Analizar con BELLA IA DE MICALIBELLA'.

En la empresa tengo una pagina en Google sites la cual me permite dejar el enlace de Gradio y Visualizar en ChatBot en ella. Lo dejo así Porque esta pagina cuenta su fondo de pantalla, tema, imágenes y toda la visualización requerida.



Aquí están las opciones disponibles ya sea cantidad de encuestas por fecha, por barrio cuadrante, o por objetivo.



s Fundadores

38  
35  
31  
30  
30  
17  
15

---

**MICALIBELLA ANALIZA Y TE DICE... :**

Para realizar un análisis de los datos de cantidad de encuestas por barrio cuadrante, primero identificamos el barrio con la mayor cantidad de encuestas y el barrio con la menor cantidad de encuestas:



- o Mayor cantidad de encuestas: Santa Monica Residential - Cerro Tres Cruces (2602 encuestas)
- o Menor cantidad de encuestas: Urbanizacion Rio Lili (15 encuestas)

**Análisis e interpretación:**


El barrio Santa Monica Residential - Cerro Tres Cruces es claramente el más popular entre los encuestados, lo que puede deberse a diversos factores como su tamaño, densidad poblacional, nivel socioeconómico, entre otros. Por otro lado, Urbanizacion Rio Lili presenta una cantidad muy baja de encuestas, lo que puede indicar una menor participación de los residentes en actividades de encuestas ciudadanas.

**Estrategias para obtener más encuestas o mantener el nivel:**

1. **Promoción y difusión:** Realizar campañas de promoción y difusión de las encuestas en los barrios con menor participación, utilizando medios de comunicación locales, redes sociales, carteles en lugares estratégicos, entre otros.
2. **Incentivos:** Ofrecer incentivos a los residentes para que participen en las encuestas, como sorteos de premios, descuentos en servicios locales, o reconocimientos públicos.
3. **Participación comunitaria:** Involucrar a líderes comunitarios, organizaciones locales y centros educativos en la promoción de las encuestas y en la recolección de datos.
4. **Facilitar el acceso:** Asegurarse de que las encuestas sean accesibles para todos los residentes, utilizando plataformas en línea, encuestas en papel en lugares públicos, y ofreciendo asistencia en caso de necesidad.

## CHATBOT MICALIBELLA - Análisis de Encuestas Asistente Virtual

 Total de encuestas cargadas: 61119

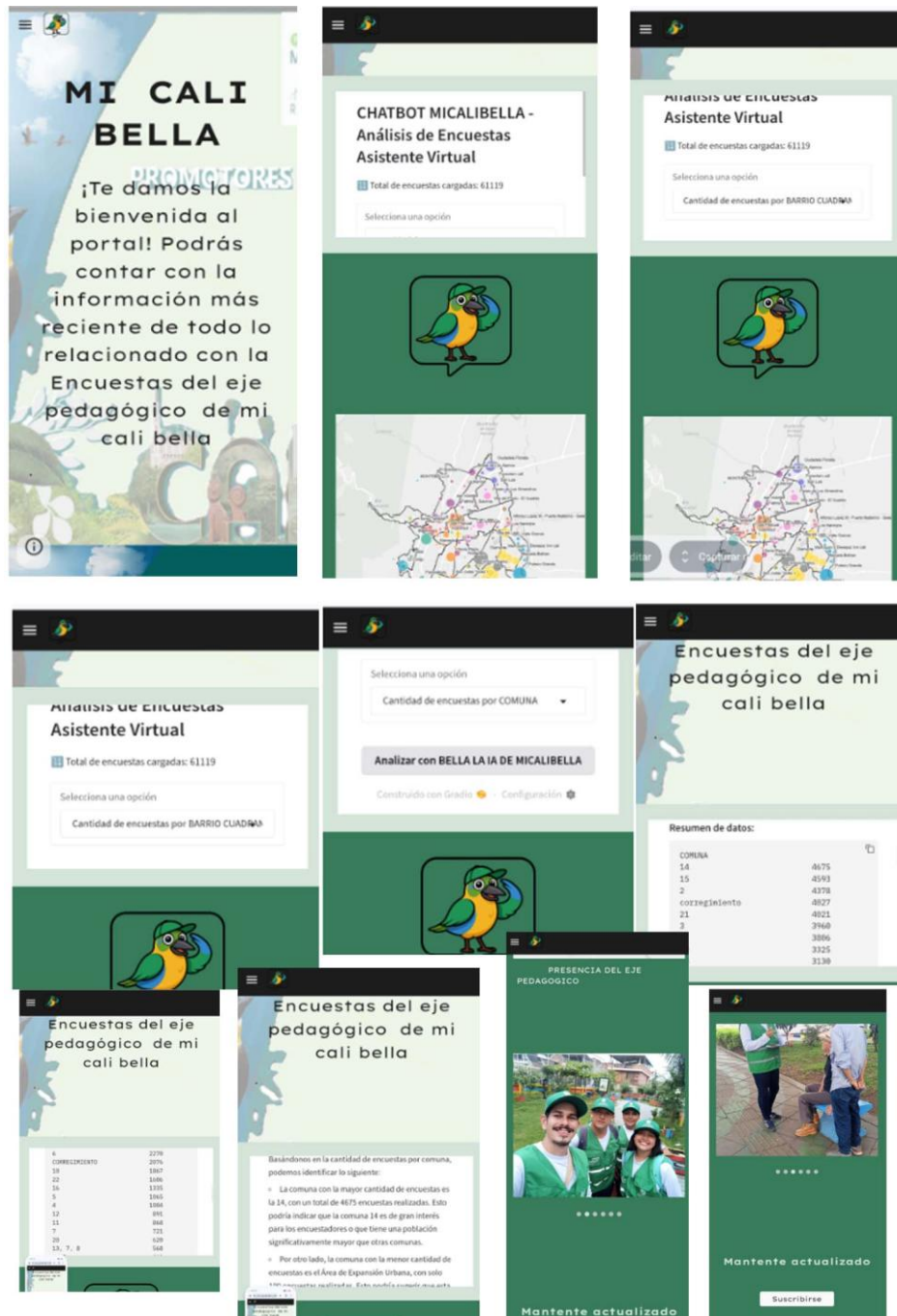
Selecciona una opción

Cantidad de encuestas por COMUNA ▼

### Resumen de datos:

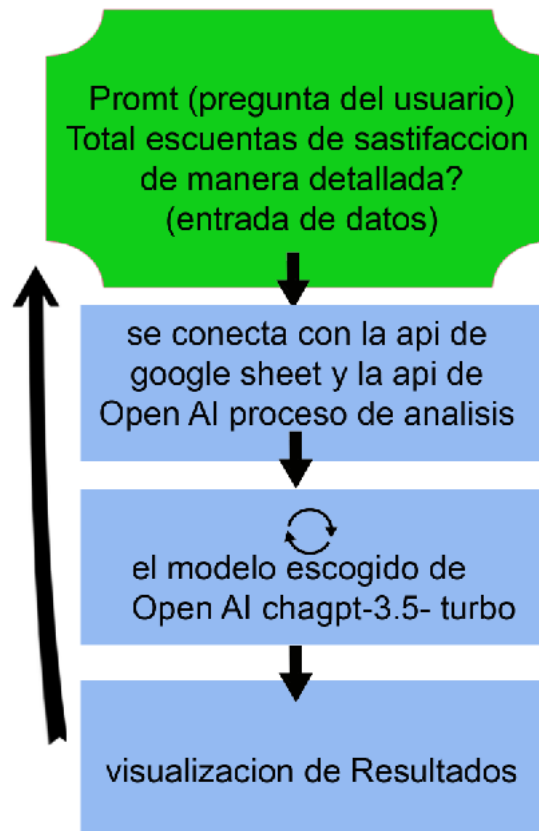
COMUNA	
14	4675
15	4593
2	4378
corregimiento	4027
21	4021
3	3960
9	3806
19	3325
18	3130
13	2759
17	2591
8	2566
6	2270
CORREGIMIENTO	2076
10	1867
22	1606
16	1335
5	1065
4	1004
12	891
11	868
7	721

### 3.3.1 Visualización de la Plataforma Web



**Título 4.** Flujo de consulta Detallada.

Figura 1. Formas y descripción de las formas.



## Conclusiones

Concluimos que la inteligencia artificial Esta Marcando y siendo tendencia para solucionar preguntas recurrentes automatizando procesos haciendo eficaz para nuestro trabajo y labores diarias Donde Dimos Solución a la necesidad de tener a la mano información confiable donde Machine Learning interpreta y nos da su opinión de manera profesional y en lenguaje natural dando a entender la cantidad de menor y mayor encuestas encuestas según la dependencia ya sea por barrio, fecha o objetivo . Cada vez que ingresan datos a la base de datos de google sheets la api nos trae la información donde pasa por unas condiciones que generan un resumen detallado de totales de encuestas don la IA Opina referente a las técnicas y forma para tener más encuestas y poder atender aquellos lugares donde hay poca afluencia de publico.

Al final por medio de Gradio nos funciono como interfaz web para el usuario el cual incorporamos a Google sites.

## Referencias

Modulo seminario AI universidad Remington, Google Colab (laboratorio de Google donde trabajamos el codigo)

Normas Apá <https://normasapa.in/>

**Abid, A., Tejani, A., Horn, M., & Zou, J.** (2019). *Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild*. In *Proceedings of the 36th International Conference on Machine Learning (ICML) 2019 - Demo Track*. Obtenido de <https://www.gradio.app/docs>

Google Cloud. (2024). *Using OAuth 2.0 to Access Google APIs* [Guía técnica]. Obtenido de <https://cloud.google.com/docs/authentication>

OpenAI. (2024). *OpenAI API Documentation* [Manual en línea]. Documentación oficial sobre cómo usar la API de OpenAI, incluyendo GPT-3.5, GPT-4, embeddings y asistentes. Obtenido de <https://platform.openai.com/docs>

McKinney, W. (2022). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media. Obtenido de <https://pandas.pydata.org/docs/>

Google Research. (2024). *Colaboratory – Google Colab* [Manual en línea]. <https://colab.research.google.com/>

### Anexos

ID	Fecha	Nombre del Promotor	Nombre del Encuestado	Teléfono	E-Mail	Barrio/Cuadrante	Latitud	Longitud	Comuna	Cuadrante	Información Brindada es útil para sensibilizar y educar	Información Fecunda	Tratamiento amable y respetuoso	Información del espacio físico de la comuna	EEO Actividad	Objetivo
52492	52491	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	SANTAGO RODRIGUEZ	3333433504	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	3	PEDAGOGICO	ATENCION SEMANA SANTA
52493	52492	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	LAILA MORALES	3337334378	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52494	52493	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	DUSTAVO MORALES	3002733356	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52495	52494	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	LEIDY SANCHEZ	3366998943	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52496	52495	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	ANDREA MORALES	3007304397	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52497	52496	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	ANDREA LOPEZ	378639637	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	3	PEDAGOGICO	ATENCION SEMANA SANTA
52498	52497	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	LAURA AGUIRRE	3077482021	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52499	52498	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	LISETH CARVAJAL	387717779	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52500	52499	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	MARITHA HERRERA	3022464454	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52501	52500	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	JULIA V LOPEZ	3232020071	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52502	52501	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	LUCIA CANTOR	3333763306	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA
52503	52502	2025-04-19	VALENCIA BEPARRICIZ ANIELA TATIANA	MAVERLY QUEVAS	3339593030	NA	3.467864	-78.545478	concepcion	DEPPO-DRS-TOPEV	4	4	4	4	PEDAGOGICO	ATENCION SEMANA SANTA

Base De Datos