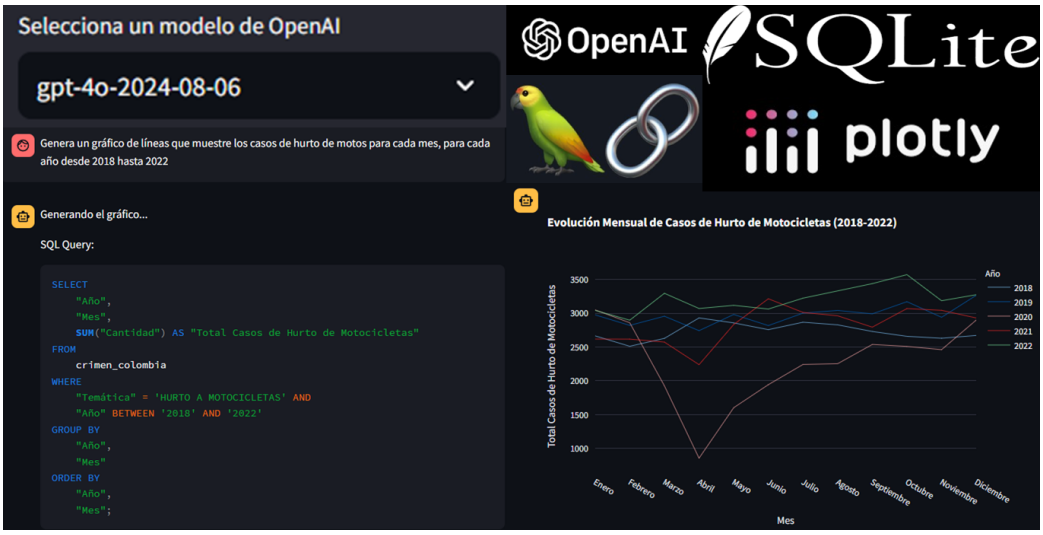
Graphical Abstract

Desarrollo de un sistema a la medida de generación aumentada por recuperación para la consulta interactiva de información especializada de estadística delictiva en Colombia

Nicolás Forero Baena, Hector Javier Hortua Orjuela



Highlights

Desarrollo de un sistema a la medida de generación aumentada por recuperación para la consulta interactiva de información especializada de estadística delictiva en Colombia

Nicolás Forero Baena, Hector Javier Hortua Orjuela

- Se desarrolló y desplegó una aplicación web de Streamlit, fundamentada en GPT4o de OpenAI, LangChain y LangGraph, que muestra dataframes y gráficas con estadísticas delictivas de la Policía Nacional de Colombia de los últimos 20 años, a partir de consultas del usuario en lenguaje natural transformadas a consultas de SQL.
- La aplicación web (https://tesiscrimen.streamlit.app/) muestra buenos resultados en general, tiempos de inferencia (10-20 segundos) y costos razonables (2-5 centavos de dolar/pregunta), permitiendo consultar de una forma atractiva e interactiva la estadística delictiva colombiana.

Desarrollo de un sistema a la medida de generación aumentada por recuperación para la consulta interactiva de información especializada de estadística delictiva en Colombia

Nicolás Forero Baena[a], Hector Javier Hortua Orjuela[b]

[a]*Estudiante Maestría Estadística Aplicada y Ciencia de Datos, nforeroba@unbosque.edu.co, Bogotá, Colombia*
[b]*Docente Maestría Estadística Aplicada y Ciencia de Datos, hhortuao@unbosque.edu.co, Bogotá, Colombia*

Abstract

Los modelos de lenguaje de gran tamaño, o LLMs por sus siglas en inglés, han mostrado capacidades formidables en la generación de texto articulado. Actualmente, son la base de varios tipos de aplicaciones en la escena IA, e.g. chatbots, sistemas QA, recomendadores, etc. Sin embargo, estos LLMs pueden mostrar imprecisiones convincentes sobre temas especializados o recientes, lo que se conoce como 'alucinaciones'. Por tal motivo, la generación aumentada por recuperación, o RAG por sus siglas en inglés, se ha mostrado útil en la producción de contenido factualmente correcto por parte de los LLMs, al brindarles una base de conocimiento o una ventana contextual verídica, e.g. texto, base de datos, etc. Se desarrolló una aplicación web con los marcos de trabajo Langchain y LangGraph, mediante la cual un usuario hace consultas en lenguaje natural sobre estadística delictiva de Colombia; un modelo como GPT4o interpreta dicha consulta, la traduce en lenguaje SQL, ejecuta la consulta sobre una base de conocimiento especializada construída a partir de sábanas de datos del SIEDCO (Sistema de Información Estadístico, Delincuencial Contravencional y Operativo) de la Policía Nacional de Colombia, recupera la información y la retorna al usuario en forma de gráfico o dataframe, todo lo anterior en una interfaz gráfica sencilla de Streamlit. La aplicación web muestra buenos resultados en general y tiempos de inferencia razonables, con fallos de interpretación ocasionales, permitiendo consultar de una forma interactiva esta información coyuntural de interés general.

Preprint submitted to Repositorio Biblioteca Universidad El Bosque *January 21, 2025*

Keywords: RAG (Retrieval-Augmented Generation), (LLMs) Large Language Models, GPT (Generative Pre-training Transformer), OpenAI, Langchain, LangGraph, SQL (Structured Query Language), Text-to-SQL, Streamlit, Estadística delictiva, SIEDCO

---

1. Introducción

Los modelos de lenguaje de gran tamaño

Los modelos de lenguaje de gran tamaño (Large Language Models - LLMs) están pre-entrenados con billones de parámetros y han mostrado capacidades formidables en la producción de texto articulado y cohesivo, a veces indistinguible del contenido generado por el ser humano (Auffarth, 2023).

El progreso en modelos de lenguaje de gran tamaño ha evolucionado rápidamente, desde los modelos T5 de Google en 2019 hasta todo un abanico de modelos tanto libres como propietarios en el presente (figura 1, Naveed et al. (2024)). Algunos tienen la capacidad de procesar lenguaje natural en múltiples idiomas, e incluso tienen capacidades multimodales, i.e. procesan texto, imágenes, documentos, audio, video. Gracias a esto, se están implementando en aplicaciones como chatbots, asistentes virtuales, resumidores automáticos, agentes de inteligencia de negocio, generadores de código, 'copilotos', etc (Auffarth, 2023). Con marcos de trabajo como LangChain y LangGraph, es posible construir estas aplicaciones.

La arquitectura de transformador

En 2017, en el reconocido artículo 'Attention Is All You Need', Vaswani y colegas propusieron la arquitectura de transformador (figura 2), la cual está basada únicamente en mecanismos de auto-atención, prescindiendo por completo de la recurrencia y las convoluciones de otros modelos (Vaswani et al., 2017). Las redes neuronales recurrentes (RNNs) tradicionales tienen dificultades con dependencias de largo plazo debido al problema de desvanecimiento de gradiente, y las redes LSTM (long-short term memory), aunque mantienen la información a lo largo de secuencias más extensas, siguen recurriendo al procesamiento secuencial de la información (Bouchard and Peters, 2024). Mediante experimentos de traducción de lenguaje, mostraron

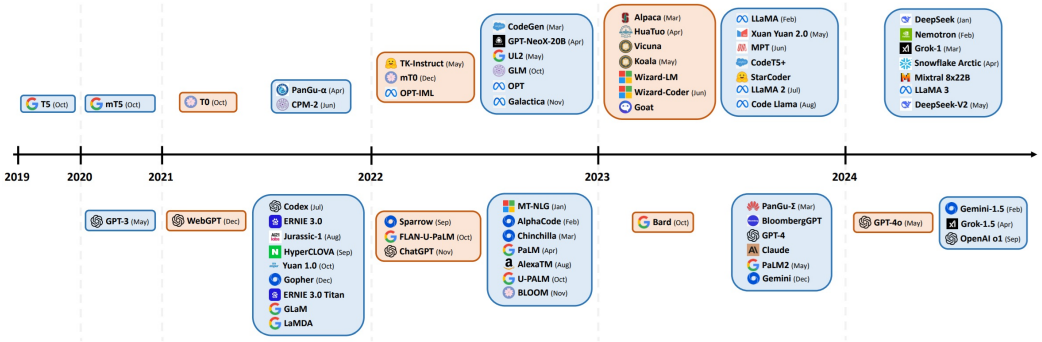


Figure 1: Línea de tiempo de lanzamientos de varios LLMs hasta septiembre 2024, tomado de Naveed et al. (2024). Sobre la línea, modelos libres, e.g. LlaMA, Nemotron, StarCoder; bajo la línea, modelos propietarios, e.g. Gemini, Claude, GPT-4.

que estos modelos son superiores en calidad, al tiempo que son más ***paralelizables*** y requieren significativamente menos tiempo de entrenamiento (Vaswani et al., 2017).

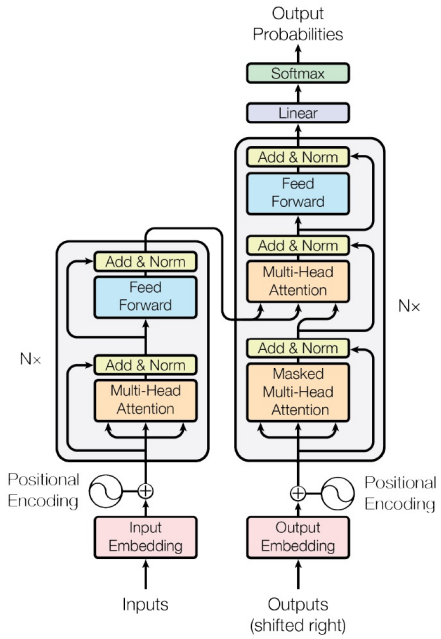


Figure 2: La arquitectura del transformador, tomada de Vaswani et al. (2017). Su base es el sistema de auto-atención multicabeza.

La auto-atención es un mecanismo que permite que cada token de una frase tenga en cuenta a los demás tokens, sin importar la distancia entre ellos. Este proceso permite que las palabras sean codificadas en función de su contexto, lo cual es útil en situaciones donde la misma palabra puede tener distintos siginificados o connotaciones dependiento de las palabras que la rodean (Amidi and Amidi, 2024).

Los modelos GPT (Generative Pre-trained Transformer)

Los modelos GPT de OpenAI, que son la base de la popular herramienta ChatGPT lanzada en noviembre del 2022, están fundamentados en la arquitectura de transformador. Éstos solo contienen modulos de decodificación (decoder-only) junto a un mecanismo de auto-atención emparejado con una red neuronal de alimentación hacia adelante totalmente conectada por posición (position-wise fully linked feed-forward network) en cada capa de la arquitectura (Bouchard and Peters, 2024).

A octubre del 2024, modelos como GPT-4o y GPT-4o mini también se pueden utilizar mediante su API, para tareas complejas y multimodales o ligeras respectivamente, con tarifas específicas por token.

Generación aumentada por recuperación

Pese a su evidente impacto en la forma en la que las personas interacturan con la información, todos los modelos de lenguaje, incluyendo GPT, pueden mostrar imprecisiones sobre muchos temas especializados o recientes con los que estén o no familiarizados, lo que se conoce como 'alucinaciones', generando contenido que es inconsistente con hechos reales o con la información proporcionada por el usuario (Huang et al., 2023).

Por tal motivo, la generación aumentada por recuperación de información, conocida en inglés como Retrieval-Augmented Generation o RAG Lewis et al. (2021), se ha mostrado útil en la producción de contenido factualmente correcto a partir de una base de conocimiento o una ventana contextual proporcionada por el usuario. Dependiendo de la ventana de contexto (información estructurada, no estructurada o ambas), se requiere diseñar el sistema RAG óptimo para poder interactuar con la información (Nguyen et al., 2024).

Ventana de contexto con datos estructurados

Para trabajar con un contexto especializado en forma de data estructurada, e.g. las tablas de una base de datos relacional, un LLM puede transformar consultas en lenguaje natural del usuario a consultas en lenguaje SQL (text-to-sql), las cuales se ejecutan sobre la base de datos mediante herramientas dentro del marco de trabajo LangChain mencionado previamente (Alto, 2024). En retorno, el usuario recibiría información de la base de datos acorde a su consulta en lenguaje natural (figura 3).

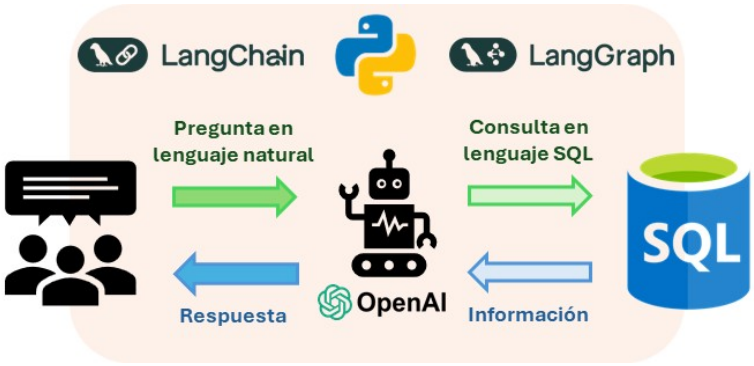


Figure 3: Esquema general de un sistema donde un LLM genera respuestas con base en información recuperada a partir de consultas SQL ejecutadas sobre una base de datos relacional, a partir de la pregunta de un usuario en lenguaje natural.

Empresas como Pinterest (redes sociales, USA) y Swiggy (pedido y distribución de comida, India) han desarrollado e implementado herramientas de analítica de datos text-to-sql con LLMs. El equipo de ingeniería de Pinterest (Obeng et al., 2024) potenció su herramienta in-house de analítica *Querybook* con esta tecnología, mejorando la productividad de sus analistas de datos y otros usuarios de la base de datos. Así mismo, Swiggy (Amaresh and Reddy, 2024) desarrolló *Hermes*, una herramienta con IA generativa que permite hacer una pregunta en lenguaje natural, obtener una consulta SQL y recibir los resultados de la ejecución de la consulta en Slack (una herramienta de comunicación empresarial propiedad de Salesforce), evitando dashboards, intermediarios, conocimiento profundo del esquema de base de datos o SQL.

El presente trabajo tiene como objetivo desplegar una aplicación web,

fundamentada en GTP4o, text-to-sql, Langchain y LangGraph, capaz de responder preguntas asociadas a una base de datos de estadística delictiva de Colombia con estadísticas delictivas de los últimos 20 años.

2. Origen de la información y análisis descriptivo

Origen de la información

Inicialmente, se solicitaron a la Secretaría distrital de seguridad, convivencia y justicia datos de estadística delictiva de Bogotá (vía derecho de petición del 20 de marzo del 2024, a través de la plataforma 'Bogotá te escucha'). El 10 de abril del 2024, se recibió respuesta de la entidad con información consolidada y totalizada en excel, pero también informaron del traslado de la solicitud de información a la Policía Metropolitana de Bogotá, puesto que la información proviene del SIEDCO (Sistema de Información Estadístico, Delincuencial, Contravencional y Operativo) de la Policía Nacional.

El 19 de abril de 2024, la Policía Metropolitana de Bogotá respondió con un link a la información de hurto, homicidio y otros crímenes a nivel nacional (https://www.policia.gov.co/estadistica-delictiva). El 3 de julio se escribió un correo a la Dirección de Investigación Criminal e INTERPOL de la DIJIN en referencia al acceso al sistema SIEDCO. Pese a que respondieron el 9 de julio que el acceso directo al SIEDCO no era posible, brindaron un enlace a otro portal de datos delictivos:

https://portalsiedco.policia.gov.co:4443/extensions/PortalPublico/index.html#/home

Éste es más actualizado y robusto que el anterior, a partir del cual se descargaron las sábanas de datos para cada una de las 24 tipologías de crimen. Posteriormente, el 14 de noviembre, se consultó nuevamente el portal SIEDCO, encontrándose sábanas de datos para 10 tipologías de crimen adicionales.

En python, mediante las librerías pandas y sqlite3, se construyó la base de datos crimen.db, que consta de la tabla crimen_colombia, la cual se consolidó a partir de las 34 sábanas de datos del portal público SIEDCO, con

datos desde 2003 hasta 2024.

Análisis exploratorio

La tabla crimen_colombia consta de 10'210.484 filas y 16 columnas (tabla 1):

Table 1: Resumen de variables de la tabla `crimen_colombia`.

Variable	Tipo de dato	Número de valores únicos
Temática	object	34
Año	object	22
Mes	object	12
Departamento	object	32
Código Dane	object	6389
Municipio	object	1022
Armas / Medios	object	74
Agrupa Edad Persona	object	4[a]
Género	object	3[b]
Zona	object	4[c]
Clase de Sitio	object	499
Día	object	7
Cantidad	float64	Contador de casos o de gramos
Clase Bien	object	65
Descripción Conducta	object	89
Unidad Medida	object	1[d]

[a] ('ADULTOS', 'MENORES', 'JOVENES', 'NR')
[b] ('M', 'F', 'NR')
[c] ('URBANA', 'RURAL', 'OTRAS', 'CARRETERA')
[d] ('GRAMO')

Sin duda, la variable más importante es *Temática*, donde se tienen las siguientes 34 categorías de delito: abigeato, amenazas, automotores recuperados, capturas, delitos contra administracion publica, delitos contra el medio ambiente, delitos informaticos, delitos sexuales, derechos de autor, estafa, extorsión, homicidios, homicidios en accidentes de tránsito, hurto a automotores, hurto a comercio, hurto a entidades financieras, hurto a motocicletas, hurto a personas, hurto a residencias, incautacion armas de fuego,

incautación de estupefacientes, injuria y calumnia, lesiones en accidentes de tránsito, lesiones personales, maltrato animal, motocicletas recuperadas, pirateria terrestre, secuestro extorsivo, secuestro simple, terrorismo, trafico migrantes, trata personas, violencia a servidor publico y violencia intrafamiliar.

A continuación se muestran algunos resultados para tipologías de interés general:

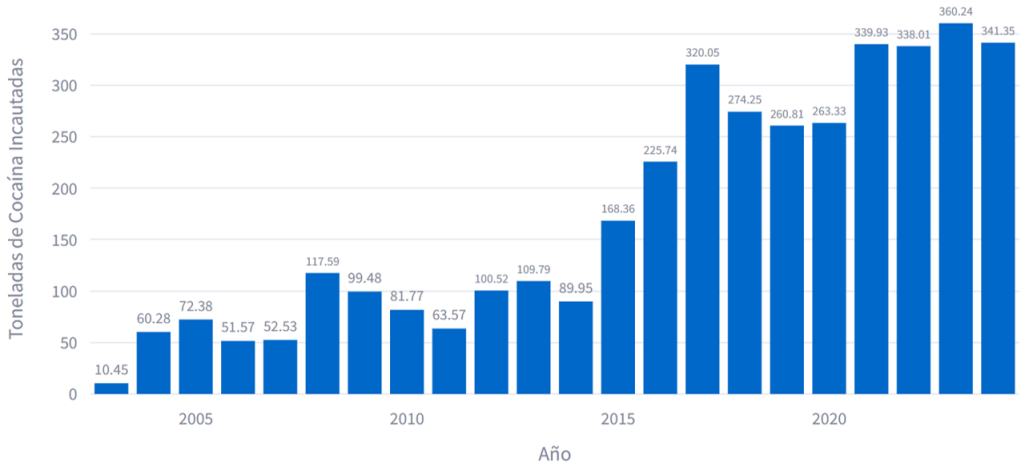


Figure 4: Toneladas de cocaina incautadas por año, 2003 - 2024.

El narcotráfico es un problema serio en nuestro país, en especial la producción de cocaina. Se observa un punto de inflexión en la incautación de cocaina (figura 4) entre 2014 y 2015, y una disminución en la incautación entre 2018 y 2020 (año de pandemia). El ritmo se recuperó en 2021 hasta el presente, con más de 330 toneladas incautadas por año.

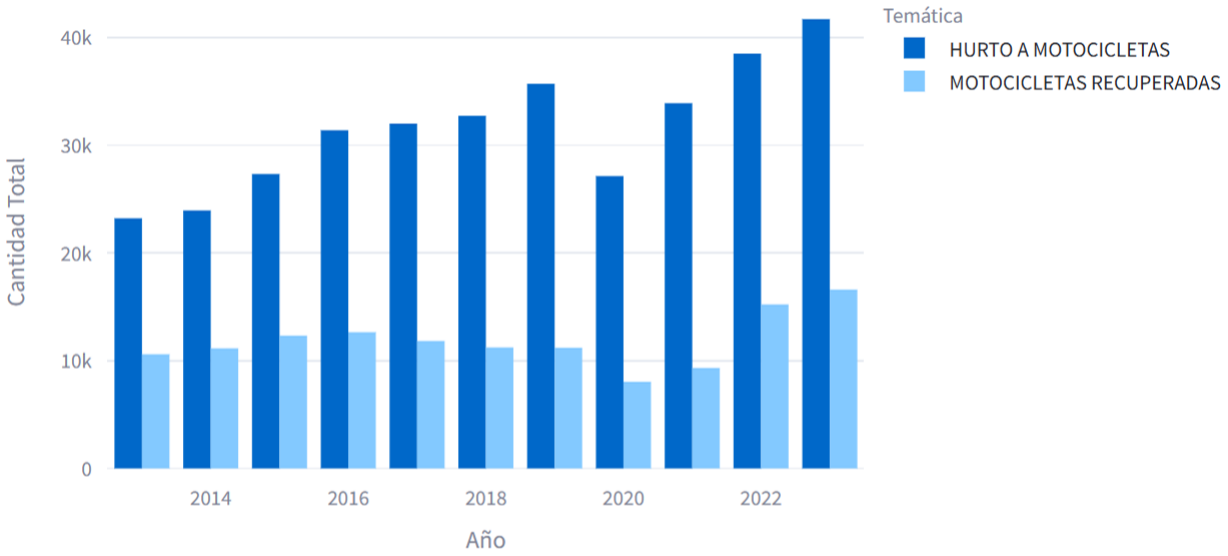


Figure 5: Comparativo motos hurtadas contra motos recuperadas por año, 2013 - 2023.

Las motocicletas son un medio de transporte popular entre los colombianos, al ser relativamente económicas y facilitar la movilidad en medio de la congestión vehicular y el tráfico. Sin embargo, eso las hace blanco del hurto (figura 5). Entre 2013 y 2023, se observa una tendencia creciente en el hurto de motocicletas (interrumpida por el confinamiento en 2020), no siendo así para la cantidad recuperada, que suele ser entre un tercio y un medio de la cantidad robada de estos vehículos.

Table 2: Top 5 de ciudades con más casos de violencia intrafamiliar entre 2019 y 2023.

Municipio	Casos
BOGOTÁ D.C. (CT)	180,065
MEDELLÍN (CT)	47,357
CALI (CT)	28,391
SOACHA	14,343
CARTAGENA (CT)	13,317

Entre 2019 y 2023, la ciudad con más casos de violencia intrafamiliar fue Bogotá DC (tabla 2), seguida por Medellín y Cali. Estos 3 centros urbanos, en ese orden, son los más poblados del país.

Sería difícil hacer un análisis descriptivo exhaustivo de este dataset tan extenso y variado, por lo cual se invita al lector a que utilice la herramienta presentada en este artículo para explorar las tipologías de crimen que más le interesen, pero se mostraron algunos resultados para tipologías de interés general.

3. Metodología

Para la elaboración de la herramienta fueron fundamentales el capítulo 8 del libro referenciado en Alto (2024) y las clases del profesor Matt Dancho sobre IA generativa (https://university.business-science.io/). También fue necesario crear una cuenta de desarrollador en OpenAI y cargar crédito para utilizar sus modelos, e.g. GPT-4o, GPT-4o mini, etc.

La herramienta, fundamentada en LangChain y LangGraph, procesa preguntas de usuarios para generar tablas o gráficas a partir de la base de datos SQL. Está compuesta por los siguientes elementos o eslabones, donde el modelo de lenguaje recibe distintos roles y contexto en lenguaje natural mediante una plantilla (from langchain.prompts import PromptTemplate):

a) **Enrutador**

- Formatea la pregunta del usuario, removiendo elementos innecesarios, e.g. saludos, cortesías, etc, para que GPT4o haga la transformación text-to-SQL
- Infiere si se debe generar tabla o gráfico (tabla por default)
- Produce: pregunta formateada y decisión de visualización

b) **Generador SQL (text-to-sql)**

- Crea una consulta SQL sintácticamente correcta a partir de la pregunta formateada
- Produce: consulta/query SQL

c) **Generador de dataframe**

- Ejecuta la consulta SQL sobre la base de datos

- Convierte los resultados de la ejecución de la consulta en un diccionario
- Produce: datos en formato diccionario

d) **Punto de Bifurcación**

- Determina si se genera una tabla o un gráfico. Si es tabla, se muestra la información al usuario y concluye el proceso. Si es gráfico, se procede con los siguientes eslabones

e) **Instructor del Generador de Gráficos**

- Genera las instrucciones básicas para crear el gráfico de plotly más adecuado para representar lo que pide el usuario
- Define títulos y etiquetas con base en la pregunta y los datos
- Produce: instrucciones para el gráfico

f) **Generador de Gráficos**

- Convierte el diccionario con los datos producto de la ejecución de la consulta SQL a dataframe
- Genera el código de Python para producir el gráfico a partir del dataframe. El código se ejecuta con REPL (Read-Eval-Print Loop en langchain_experimental.utilities)
- Produce: código de Python y gráfico en formato JSON

El progreso en consola se puede seguir mediante la función `state_printer`, que muestra el estado de cada eslabón en la cadena de procesamiento.

La aplicación web se desplegó mediante Streamlit en un link público (https://tesiscrimen.streamlit.app/), donde puede explorarla y hacer consultas.

Las figuras 6 y 7 muestran los flujos de trabajo desde la pregunta hasta el resultado, tabla y gráfica respectivamente, tanto a nivel de la interfaz gráfica de Streamlit como a nivel de la consola.

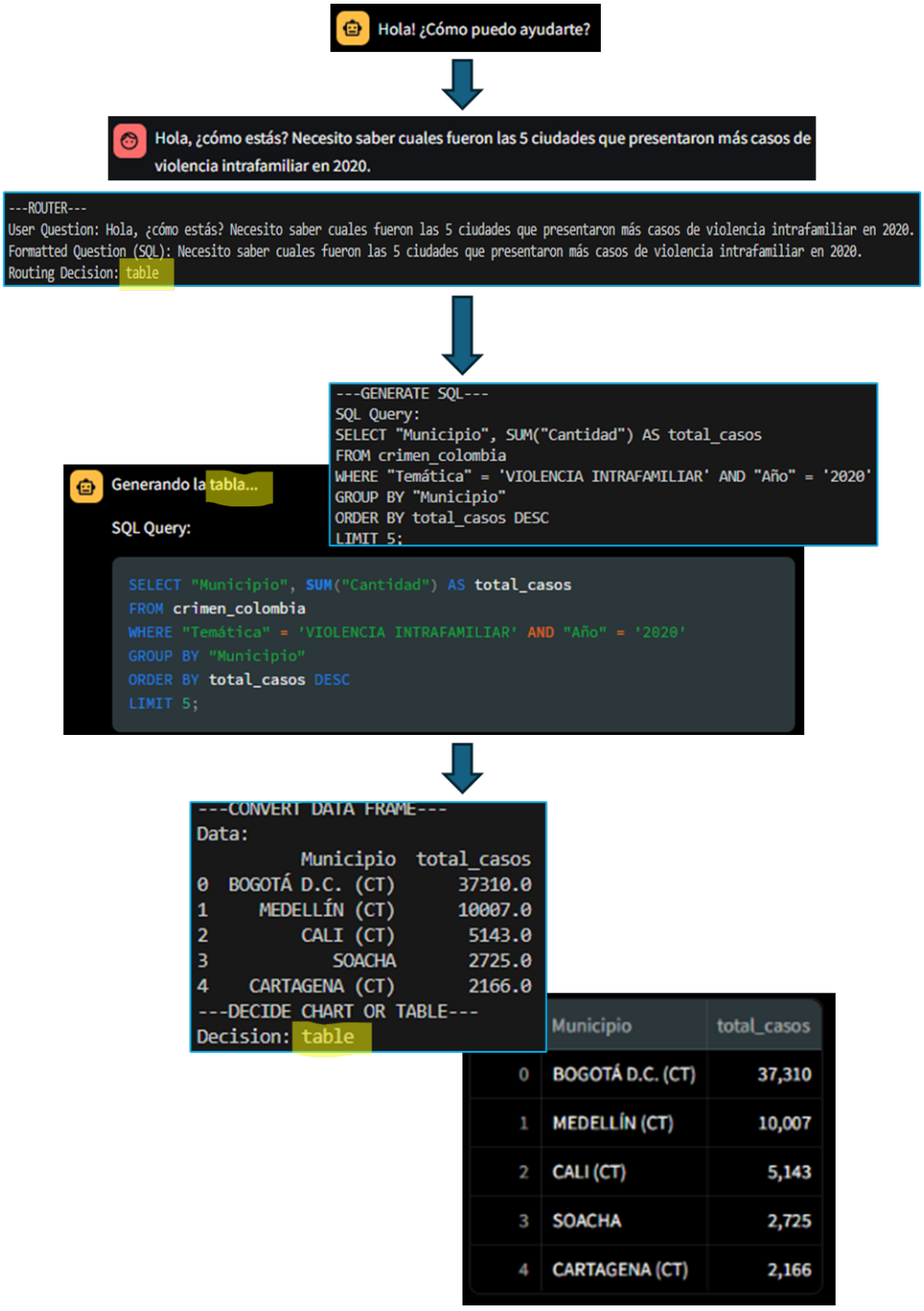


Figure 6: Resumen del proceso de generación de respuesta en forma de dataframe.

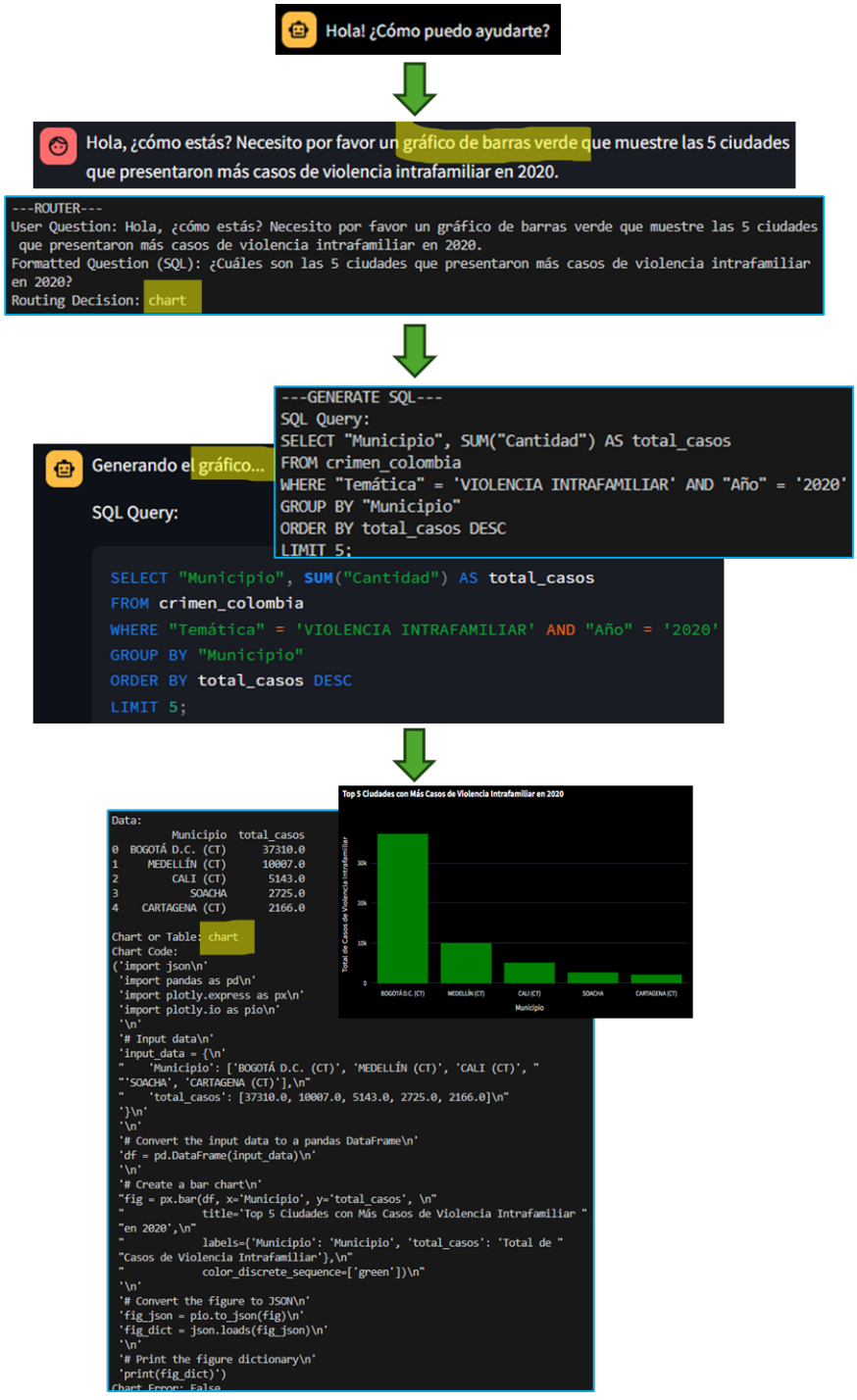


Figure 7: Resumen del proceso de generación de respuesta en forma de gráfica.

4. Resultados y discusión

La tabla 3 muestra el resumen del desempeño de la aplicación en algunas preguntas concretas, entendiendo que no constituye un muestreo exhaustivo ni abarca todas las posibles combinaciones de temáticas, años, medios, etc, tanto por tiempo como por costos. Por ello, se invita al lector a que pruebe la aplicación web, con preguntas propias o como se muestran en la tabla 3, para dimensionar el alcance de la herramienta. La verificación de los resultados se hizo con código de Pandas para aquellas consultas donde el aplicativo respondió efectivamente con una tabla o con una gráfica (figura 8).

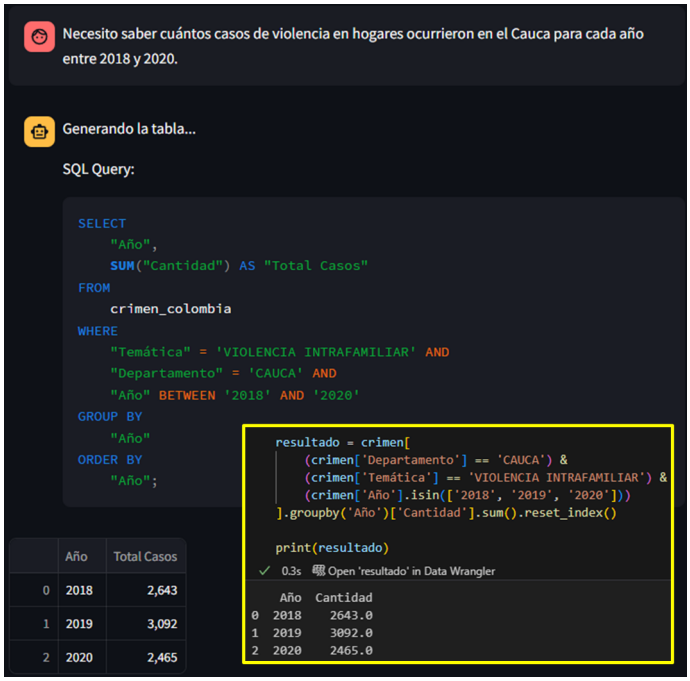


Figure 8: Respuesta del aplicativo a una pregunta y verificación del resultado con Pandas.

Sin embargo, aún hay preguntas para las cuales la herramienta no puede proporcionar una respuesta, en general porque no entiende el requerimiento, el gráfico solicitado es muy complejo o porque no cuenta con el contexto ni con los datos necesarios, e.g. un usuario pregunta por delitos en el norte de un departamento/municipio, pero no se tiene información oficial a nivel de cardinalidad. Tampoco proporciona respuestas a preguntas específicas como crímenes perpetrados por asesinos seriales, o fechas particulares.

Table 3: Resumen del desempeño de la aplicación.

Pregunta	**Formato Resultados**	**¿Coincide con resultado en Pandas?**	**Costo (centavos de USD)**
Hola. Necesito saber cuántos casos de homicidio ocurrieron en Becerril para cada año entre 2018 y 2022	Tabla	Sí	2
Necesito saber cuántos casos de violencia en hogares ocurrieron en el Cauca para cada año entre 2018 y 2020	Tabla	Sí	2
Necesito saber el top 5 de departamentos en cuanto a gramos de marihuana incautados en 2015	Tabla	Sí	2
Necesito saber el top 3 de departamentos en cuanto a kilogramos de cocaína incautados en 2021	Tabla	Sí	3
Necesito el top 5 de departamentos en cuanto a robo de motos en 2023	Tabla	Sí	3
Necesito un gráfico de barras que muestre el total de robos de ganado por año desde 2003 hasta 2023	Gráfico	Sí	4
Para el año en que más motocicletas robaron, quiero saber cuántas motos fueron recuperadas*	Tabla	Sí	2x3
Necesito un gráfico de pastel que muestre el total de lesiones personales respecto al día de la semana en Medellín	Gráfico	Sí	4

*La aplicación mostró el resultado solicitado, pero no fue mostrado el año en que más motos fueron robadas (condición filtro). Para fines de verificación, se hicieron las consultas 'en qué año fueron robadas más motocicletas' y 'cuántas motocicletas fueron recuperadas en 2023', coincidiendo con el resultado de la consulta original y los resultados con Pandas.

Los investigadores han desarrollado benchmarks para evaluar el rendimiento de los LLMs en problemas text-to-sql, e.g. BIRD (BIg Bench for LaRge-scale Database Grounded Text-to-SQL Evaluation, Li et al. (2024)), donde GPT-4o se destaca en la tabla de líderes de precisión de ejecución (tabla 4).

Table 4: Porcentaje de precisión de ejecución text-to-sql (EX) de modelos con GPT-4/4o, medida con BIRD-bench*

Puesto	**Modelo**	**%EX**
Desempeño humano†		92.96
3	AskData + GPT-4o *(AT&T - CDO)*	72.39
4	OpenSearch-SQL, v2 + GPT-4o *(Alibaba Cloud)*	72.28
5	Distillery + GPT-4o *(Distyl AI Research)*	71.83
8	PURPLE + RED + GPT-4o *(Fudan University + Transwarp Technology)*	70.21
13	E-SQL + GPT-4o	66.29
14	Arcwise + GPT-4o *(Arcwise)*	66.21
15	AskData + GPT-4o *(AT&T - CDO)*	65.62
16	MCS-SQL + GPT-4 *(Dunamu)*	65.45
18	OpenSearch-SQL,v1 + GPT-4 *(Alibaba Cloud)*	64.95

†Ingenieros de datos + Estudiantes de base de datos

*Tomado de https://bird-bench.github.io/, consultado por última vez el 10/11/2024. El 3/11/2024, el primer lugar fue alcanzado por CHASE-SQL + Gemini (*Google Cloud*), con una EX de 74.06%. Nota: el %EX para GPT-4o (via API o ChatGPT) en su forma original no aparece reportado.

Así mismo, autores reportan a GPT-4o como uno de los LLMs de mejor desempeño en producción junto a otros modelos tope de gama como Claude 3.5 Sonnet de Anthropic, en aspectos tales como entendimiento del esquema de base de datos, complejidad de las consultas, eficiencia de tokens y tiempo de inferencia (Aluri, 2024). Esto lo hace un modelo adecuado para la con-

strucción de herramientas para la consulta de bases de datos en lenguaje natural, sin dejar de lado los modelos libres, cada vez más capaces.

5. Conclusiones y perspectivas

Se desarrolló y desplegó una aplicación web de Streamlit, fundamentada en GPT4o de OpenAI, LangChain y LangGraph, que muestra tablas y gráficas con estadísticas delictivas de la Policía Nacional de Colombia de los últimos 20 años, a partir de consultas del usuario en lenguaje natural transformadas a consultas de SQL. La aplicación web (https://tesiscrimen.streamlit.app/) muestra buenos resultados en general, tiempos de inferencia (10-20 segundos) y costos razonables (2-5 centavos de dolar/pregunta), permitiendo consultar de una forma atractiva e interactiva la estadística delictiva colombiana.

Se han identificado varias oportunidades de mejora:

- Que además de generar tablas o gráficas, proporcione una breve interpretación amigable de las mismas, a manera de resumen ejecutivo.
- Si el dato solicitado es puntual y no hace parte de una serie de tiempo, que proporcione un texto con la información solicitada en vez de una tabla con una sola fila.
- Integrar aspectos de geolocalización y cardinalidad, e.g. mapas con puntos de calor. También información a nivel de barrios y localidades.
- Un uso a gran escala requeriría de un presupuesto proporcional al tráfico o volumen de consultas esperado, ya sea para seguir usando los modelos de OpenAI a través de su API, o de modelos de alto poder como Claude 3.5 Sonnet vía la API de Anthropic o Llama 3.1 70B / 405B en plataformas en la nube como AWS Bedrock.
- Sería ideal una integración directa con la base de datos SIEDCO, puesto que las sábanas de datos solo se pueden exportar en formato .xlsx (fue problemático para tipologías con muchas filas como 'capturas' y 'hurto a personas'), y es dispendioso descargarlas frecuentemente para tener la base de datos actualizada.

6. Agradecimientos

A mi esposa y gatos, a mis padres, a la Policía Nacional de Colombia y DIJIN, a la Secretaría Distrital de Seguridad Convivencia y Justicia, al programa de maestría en EACD de la UEB y todos sus gestores, al profesor Hector Hortua, al profesor Matt Dancho, a los autores de los libros en las referencias Valentina Alto y Ben Auffarth, entre otros, a los videos del ingeniero Santiago Valderrama (YouTube @underfitted) y del ingeniero Lance Martin de LangChain (YouTube @LangChain), a mis compañeros de maestría, etc.

References

Alto, V., 2024. Building LLM Powered Applications: Create intelligent apps and agents with large language models. Packt Publishing.

Aluri, R., 2024. Text-to-sql excellence: An evaluation of sonnet 3.5 and gpt-4o. URL: https://blog.waii.ai/text-to-sql-excellence-an-evaluation-of-sonnet-3-5-and-gpt-4o-c52af5206ffc.

Amaresh, Reddy, R., 2024. Hermes: A text-to-sql solution at swiggy. URL: https://bytes.swiggy.com/hermes-a-text-to-sql-solution-at-swiggy-81573fb4fb6e.

Amidi, A., Amidi, S., 2024. Super Study Guide: Transformers Large Language Models. Afshine Amidi and Shervine Amidi.

Auffarth, B., 2023. Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs. Packt Publishing.

Bouchard, L.F., Peters, L., 2024. Building LLMs for Production: Enhancing LLM Abilities and Reliability with Prompting, Fine-Tuning, and RAG. Towards AI.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T., 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. URL: https://arxiv.org/abs/2311.05232, arXiv:2311.05232.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., Kiela, D., 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. URL: https://arxiv.org/abs/2005.11401, arXiv:2005.11401.

Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Geng, R., Huo, N., et al., 2024. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. Advances in Neural Information Processing Systems 36. URL: https://arxiv.org/pdf/2305.03111.

Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., Mian, A., 2024. A comprehensive overview of large language models. URL: https://arxiv.org/abs/2307.06435, arXiv:2307.06435.

Nguyen, Z., Annunziata, A., Luong, V., Dinh, S., Le, Q., Ha, A.H., Le, C., Phan, H.A., Raghavan, S., Nguyen, C., 2024. Enhancing qa with domain-specific fine-tuning and iterative reasoning: A comparative study. URL: https://arxiv.org/abs/2404.11792, arXiv:2404.11792.

Obeng, A., Zhong, J., Gu, C., 2024. How we built text-to-sql at pinterest. URL: https://medium.com/pinterest-engineering/how-we-built-text-to-sql-at-pinterest-30bad30dabff.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. URL: https://arxiv.org/abs/1706.03762, arXiv:1706.03762.