Universidad del Caribe
2000 CANCUN, QUINTANA ROO, MÉXICO
CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

Asignatura:

Teoria de la información

Tema:
Ai- enginnering-tolking

Trabajo para realizar:
Elegir una tool en https://github.com/Sumanth077/ai-engineering-toolkit y presentar un demo el dia 4 de Febrero 2026

https://transapp-e56b5.web.app/

Alumno:

Pérez Barahona Pedro Luis

Programa educativo:

Ingeniería en datos e inteligencia organizacional

Presentado a:

Jiménez Sánchez Ismael

Fecha:

04/02/2026

Chatbot de Generación Aumentada de Recuperación (RAG)

Verba es un asistente personal totalmente personalizable que utiliza Generación Aumentada de Recuperación (RAG) para consultar e interactuar con sus datos, **ya sea localmente o implementados en la nube** . Resuelva dudas sobre sus documentos, cruce referencias de múltiples puntos de datos o extraiga información de las bases de conocimiento existentes. Verba combina técnicas de RAG de vanguardia con la base de datos contextual de Weaviate. Elija entre diferentes marcos de RAG, tipos de datos, técnicas de fragmentación y recuperación, y proveedores de LLM según su caso de uso.

El uso que le di, fue colocar un reglamento de transito de benito juarez para retroalimentar el RAG y así ser una consulta más rápida para el usuario común que quiera consultar algo.

Asistente Inteligente de Tránsito con RAG (Retrieval-Augmented Generation)

https://transapp-e56b5.web.app/

1. Descripción general del sistema

El asistente inteligente de tránsito es una aplicación web que permite a los usuarios realizar consultas en lenguaje natural sobre **leyes, reglamentos y normas de tránsito**, recibiendo respuestas precisas y contextualizadas. El sistema está basado en una arquitectura **RAG (Retrieval-Augmented Generation)**, la cual combina recuperación de información normativa con modelos de lenguaje avanzados para generar respuestas confiables y actualizadas.

El asistente opera de forma segura mediante **autenticación con Google**, y está diseñado para funcionar como una **plataforma escalable**, accesible desde cualquier navegador moderno.

2. Arquitectura de funcionamiento (RAG)

El flujo técnico del sistema es el siguiente:

1. **Interacción del usuario**
   El usuario ingresa una consulta por texto o voz desde la interfaz web.
2. **Backend en Firebase Functions**
   La consulta se envía a una función serverless que actúa como intermediario seguro entre el frontend y los servicios de IA.
3. **Módulo de recuperación (Retrieval)**
   - Las leyes y reglamentos de tránsito se encuentran previamente **fragmentados (chunking)** y almacenados en una base de conocimiento.

- Se utilizan **embeddings semánticos** para buscar los fragmentos más relevantes relacionados con la pregunta del usuario.
- Solo la información normativa pertinente es seleccionada para el siguiente paso.

4. **Generación de respuesta (Generation)**
   - El contexto recuperado se inyecta en el prompt del modelo de lenguaje (OpenAI).
   - El modelo genera una respuesta fundamentada exclusivamente en la información normativa recuperada, reduciendo al mínimo respuestas inventadas (hallucinations).
5. **Respuesta enriquecida**
   - La respuesta se devuelve al usuario en texto.
   - Opcionalmente, se activa un **narrador por síntesis de voz**, mejorando la accesibilidad.

3. Tecnologías y herramientas utilizadas

Frontend

- **React + Vite**
  Para una interfaz rápida, modular y altamente mantenible.
- **CSS puro**
  Animaciones ligeras y diseño visual optimizado sin dependencias externas.
- **Web Speech API**
  Reconocimiento de voz y narración de respuestas.
- **Firebase Authentication**
  Inicio de sesión con Google para control de acceso y trazabilidad.

Backend

- **Firebase Cloud Functions (Node.js)**
  Arquitectura serverless para manejar consultas sin administrar servidores.
- **Express.js**
  Manejo de rutas y peticiones HTTP.
- **OpenAI API**
  Generación de respuestas con modelos de lenguaje.
- **dotenv / variables de entorno**
  Gestión segura de claves y configuraciones sensibles.

Base de conocimiento (RAG)

- Documentos legales de tránsito (PDF, texto plano, normativas oficiales).
- Preprocesamiento:
    - Limpieza del texto legal.
    - Segmentación en fragmentos semánticos.
    - Generación de embeddings.
- Almacenamiento:
    - Puede utilizarse Firestore, bases vectoriales o servicios externos especializados.

4. Seguridad y control

- La **clave de OpenAI nunca se expone al cliente**.
- El acceso está restringido mediante autenticación.
- Firebase Functions limita llamadas no autorizadas.
- Posibilidad de implementar rate limiting y monitoreo por usuario.

5. Escalabilidad y sostenibilidad a largo plazo

El sistema está diseñado para crecer de forma progresiva:

Escalabilidad técnica

- **Serverless**: Firebase Functions escala automáticamente según la demanda.
- **Separación de capas**: frontend, backend y base de conocimiento desacoplados.
- **Fácil actualización legal**: las leyes pueden actualizarse sin modificar la interfaz.

Escalabilidad funcional

- Soporte para múltiples jurisdicciones (municipal, estatal, federal).
- Inclusión de versiones históricas de leyes.
- Integración con mapas, multas, semáforos inteligentes o cámaras urbanas.
- Implementación futura de análisis estadístico de consultas ciudadanas.

Escalabilidad institucional

- Uso en dependencias de tránsito.
- Asistencia a ciudadanos, policías y personal administrativo.
- Integración con portales gubernamentales y apps móviles.

6. Valor agregado del enfoque RAG

- Respuestas **basadas en documentos oficiales**, no en conocimiento genérico.
- Reducción de errores y desinformación.
- Fácil auditoría y trazabilidad de las fuentes legales.
- Adaptación rápida a cambios normativos.

7. Conclusión técnica

Este asistente de tránsito con RAG representa una solución moderna, robusta y escalable para la consulta de normativas legales. Su arquitectura serverless, combinada con recuperación semántica y modelos de lenguaje, permite ofrecer respuestas precisas, seguras y accesibles, con una clara proyección a largo plazo para entornos gubernamentales y ciudadanos.

Modelos de lenguaje e IA generativa

OpenAI. (2024). *OpenAI API documentation*. https://platform.openai.com/docs

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., … Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474. https://arxiv.org/abs/2005.11401

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901. https://arxiv.org/abs/2005.14165

Arquitectura RAG y recuperación semántica

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., … Yih, W. (2020). Dense passage retrieval for open-domain question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781. https://arxiv.org/abs/2004.04906

Liu, J., Yuan, W., Fu, J., & Liu, X. (2023). Retrieval-augmented generation: A survey. *arXiv preprint*. https://arxiv.org/abs/2302.00083

Firebase y arquitectura serverless

Google LLC. (2024). *Firebase documentation*. https://firebase.google.com/docs

Google Cloud. (2024). *Cloud Functions documentation*. https://cloud.google.com/functions/docs

Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C. C., Khandelwal, A., Pu, Q., … Stoica, I. (2019). Cloud programming simplified: A Berkeley view on serverless computing. *arXiv preprint*. https://arxiv.org/abs/1902.03383

Autenticación y seguridad

Google LLC. (2024). *Firebase Authentication documentation*. https://firebase.google.com/docs/auth

OWASP Foundation. (2023). *OWASP Top Ten Web Application Security Risks*. https://owasp.org/www-project-top-ten/

Procesamiento de lenguaje natural y embeddings

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. https://arxiv.org/abs/1908.10084