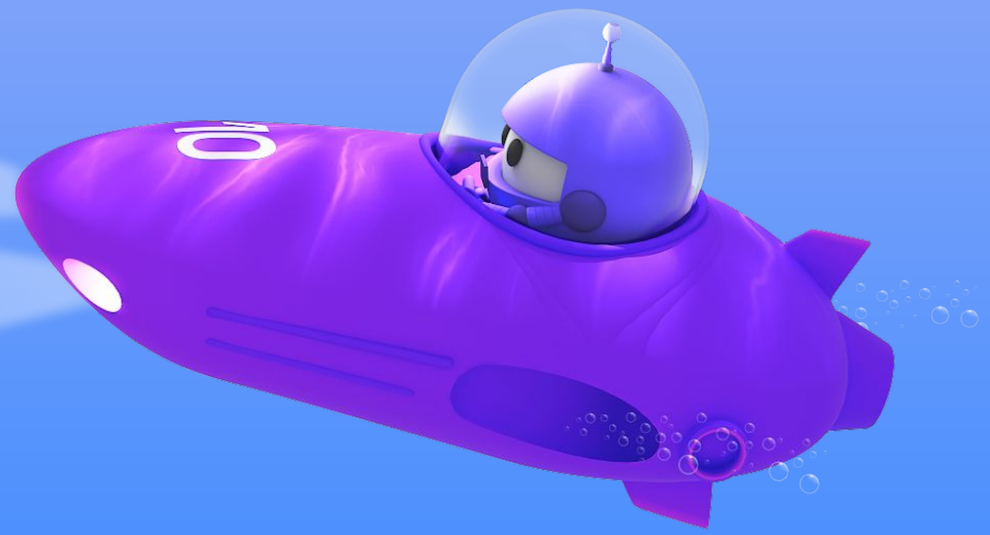
Extendiendo las capacidades de la IA con RAG en .Net 10



Ing. Marcos Polischuk

Software architect

Algunos Conceptos

LLM

Un modelo de lenguaje extenso (LLM) es un modelo estadístico de lenguaje, entrenado con una gran cantidad de datos, que se puede usar para generar y traducir texto y otros contenidos, así como realizar otras tareas de procesamiento de lenguaje natural (PLN).

Claude, Deepseek, Gemini, GPT, etc

Capacidades principales

Los LLM son una tecnología clave en la que se basan los chatbots inteligentes y otras aplicaciones El objetivo es crear bots que puedan responder a las preguntas de los usuarios en diversos contextos mediante referencias cruzadas de fuentes de conocimiento autorizadas

Limitaciones

Desafortunadamente, la naturaleza de la tecnología del LLM agrega imprevisibilidad en las respuestas del LLM. Además, los datos de entrenamiento del LLM son estáticos y agregan una fecha límite en los conocimientos que tienen.

El problema: Conocimiento fragmentado, costes crecientes

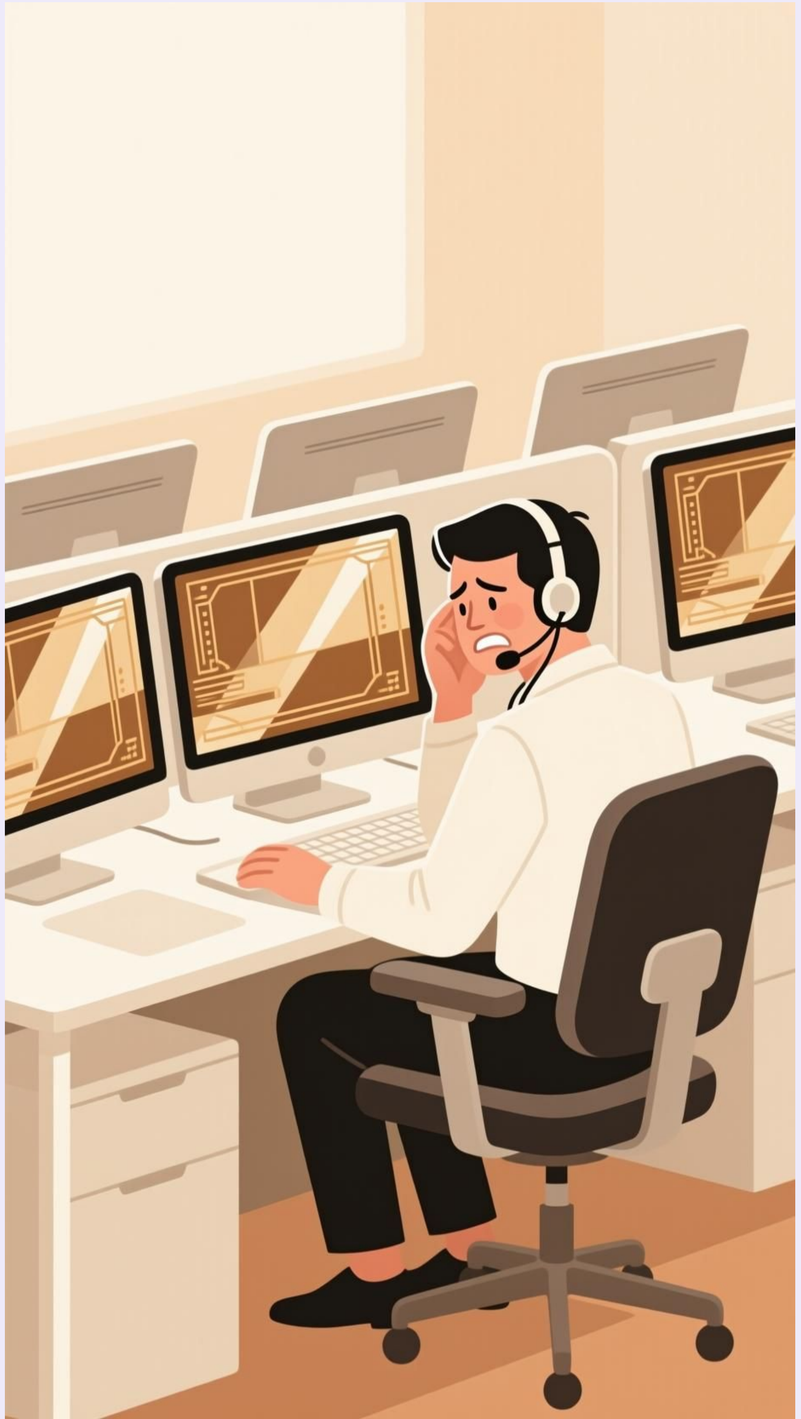
Múltiples fuentes desfragmentadas

- Información dispersa por distintos sistemas
- Información desactualizada
- Imposibilidad de tracking
- Respuestas inexactas debido a confusiones terminológicas

Impacto en las métricas

- **High AHT:** Los agentes pierden tiempo buscando.
- **Low FCR:** Información incompleta produce callbacks
- **Stressed CSAT:** Respuestas lentas e inconsistentes
- **Compliance risk:** Sin rastro de citaciones

La fragmentación del conocimiento genera respuestas inconsistentes entre los agentes, complica la incorporación de nuevos datos y hace prácticamente imposible la escalabilidad por temporadas. Cada consulta se convierte en una investigación manual a través de sistemas desconectados.

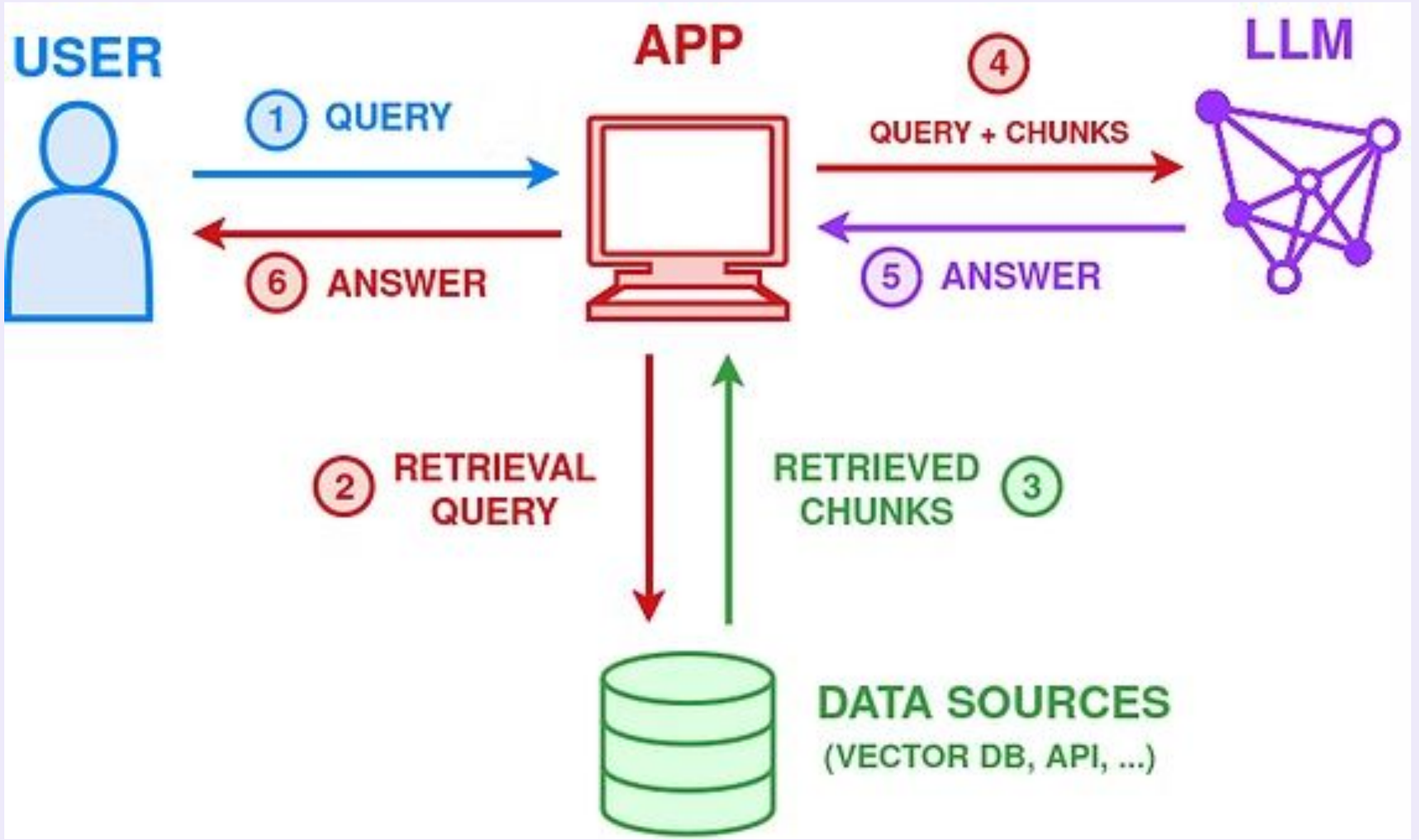


RAG al rescate!

La generación aumentada por recuperación (**RAG - R**etrieval **A**ugmented **G**eneration) es un framework de IA que combina las ventajas de los sistemas tradicionales de recuperación de información (como la búsqueda y las bases de datos) con las capacidades de los LLM.

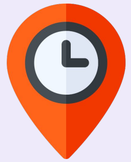
Al combinar tus datos y el conocimiento del mundo con las habilidades lingüísticas de los LLMs, la generación fundamentada es **más precisa, actualizada y relevante** para tus necesidades específicas.

RAG al rescate!



Ejemplo de escenario de caso de uso

Sistema de tracking de pedidos



Cuando llega?

Requiere verificar los plazos de la póliza estándar con los datos de seguimiento en tiempo real del transportista para obtener horas de llegada estimadas precisas, no estimaciones genéricas.

Llegó dañado

Debe hacer referencia a los plazos de notificación establecidos en las políticas, los requisitos de evidencia (carga de fotos) y los flujos de trabajo específicos de reemplazo vs reembolso.

Cambié la dirección

Incluye restricciones específicas del operador, límites de tiempo, posibles tarifas y limitaciones del sistema; todo ello documentado por separado.

Demoras en transporte

Se necesita una función de búsqueda de SLA por operador, protocolos de comunicación proactivos y posibles políticas de compensación basadas en la gravedad de la demora.

Cada escenario demuestra la amplitud de las dependencias de datos y la complejidad a la que se enfrentan los agentes cuando la información reside en múltiples sistemas no conectados.

Propuesta de solución para el escenario

App que extiende soporte de LLM con RAG

Orquestación de múltiples fuentes

Realiza consultas inteligentes en paralelo usando políticas de PDF, informacion de registros en bases de datos SQL y API externa de seguimiento.

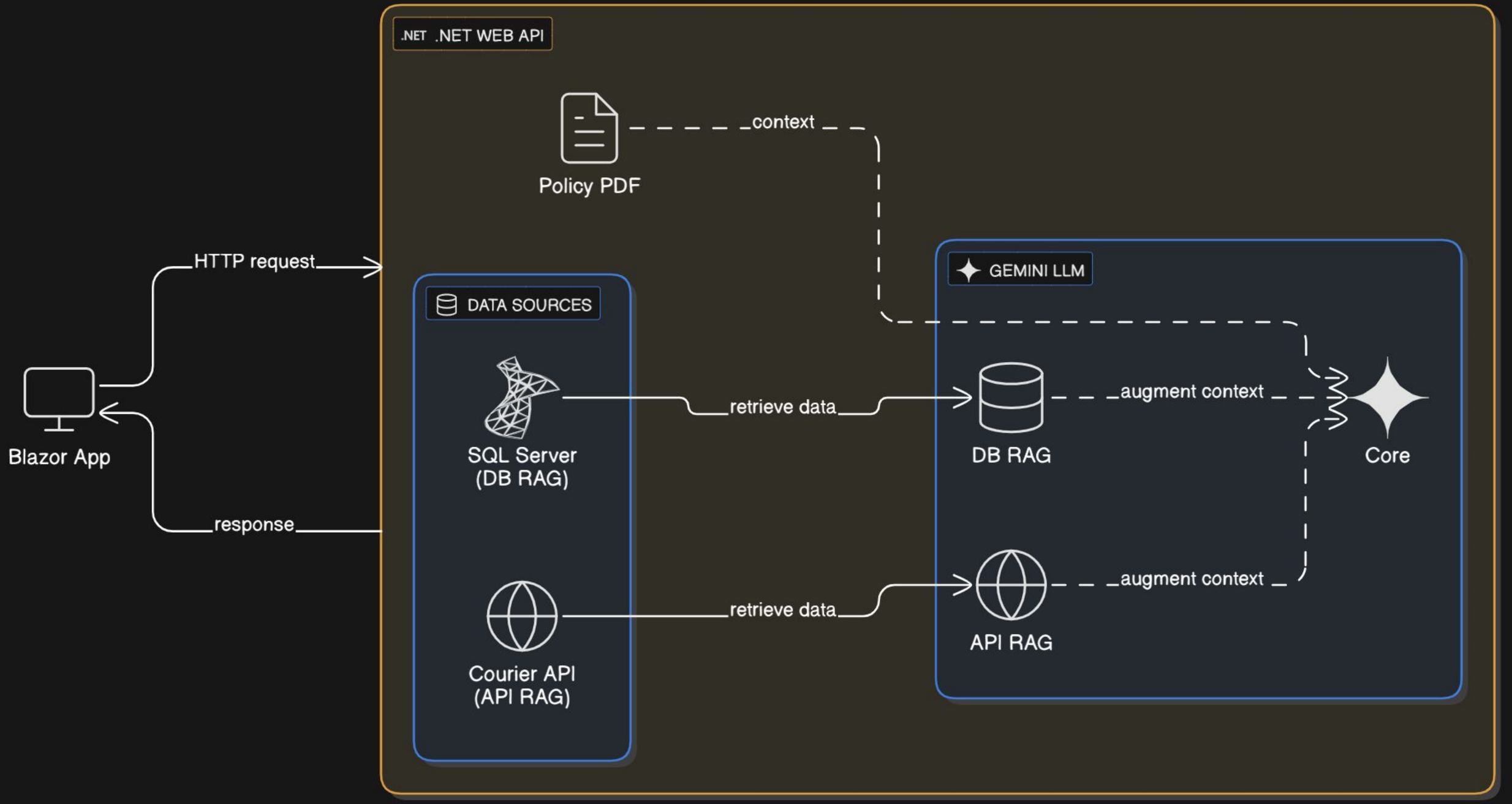
Procesamiento usando Gemini

Sintetiza la evidencia recuperada en respuestas coherentes y contextuales.

Respuestas trazables

Devuelve respuestas coherentes con citas verificables para cada hecho.

Un asistente de IA que reúne fuentes de conocimiento fragmentadas, elimina la búsqueda manual y ofrece respuestas fiables con total trazabilidad, todo ello a través de una única interfaz intuitiva.


.NET .NET WEB API
Policy PDF
context
HTTP request
Blazor App
response
DATA SOURCES
SQL Server
(DB RAG)
retrieve data
Courier API
(API RAG)
retrieve data
GEMINI LLM
DB RAG
augment context
Core
API RAG
augment context

SmartSupport

POC demo

¿Preguntas?

Muchas gracias

@markfab182

/marcos-fabian-polischuk

https://github.com/MPolischuk/smartsupport