

# Inteligencia Artificial Generativa Para la Ciencia de Datos



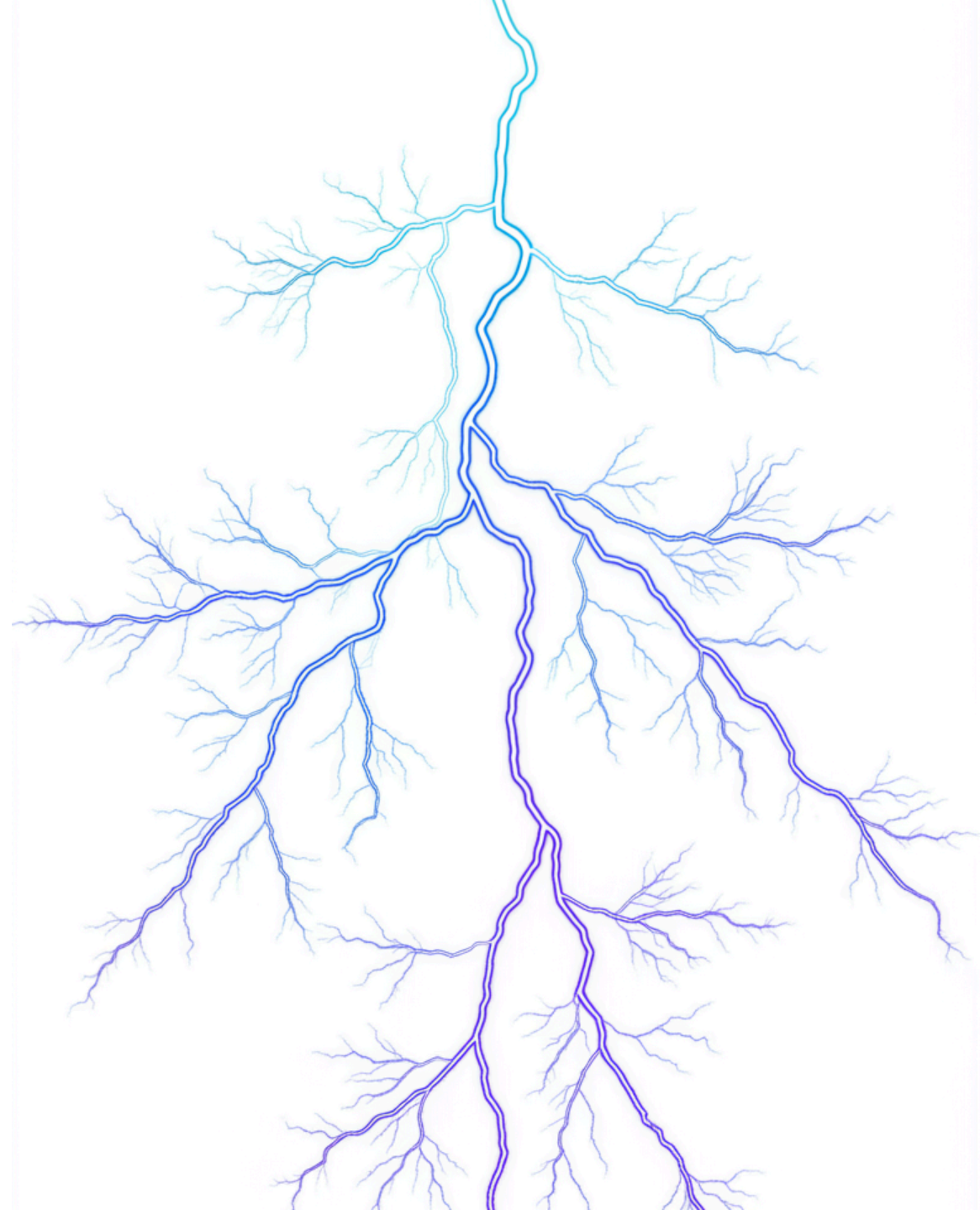
Profesor

Juan Camilo Vega Barbosa

*Consultor IA - Ingeniero IA/ML*



LinkedIn




GenBI

Interfaces conversacionales de  
datos con LLM



# La revolución conversacional en Business Intelligence

Los LLM han transformado fundamentalmente cómo interactuamos con los datos . Ya no necesitamos memorizar queries SQL complejos ni navegar por múltiples dashboards. Podemos simplemente preguntar "¿cuáles fueron nuestras ventas del último trimestre por región?" y obtener respuestas inmediatas.

Esta transformación democratiza el análisis de datos, permitiendo que cualquier persona en la organización acceda a insights sin conocimientos técnicos avanzados.

El paradigma cambió de "buscar datos" a "conversar con datos", creando una experiencia más natural e intuitiva que acelera la toma de decisiones empresariales y elimina las barreras entre usuarios de negocio y la información que necesitan.

# Chat de documentos: Consulta conversacional

Interrogar bibliotecas completas mediante conversación natural reemplaza la búsqueda manual en PDFs y reportes técnicos.

## Casos de uso clave

- **Manuales técnicos:** Configuración de sistemas
- **Reportes financieros:** Análisis de factores de impacto
- **Políticas corporativas:** Procesos y aprobaciones

**Impacto:** 70% reducción en tiempo de búsqueda de información técnica

## Análisis conversacional de datos

**LLM eliminan barreras técnicas** en análisis de datos: SQL, Python y herramientas complejas de visualización quedan obsoletas para el usuario final.

De "Muéstrame las tendencias de churn por segmento" a análisis estadístico, visualizaciones y recomendaciones automáticas.

**Impacto:** equipos de negocio acceden directamente a insights sin depender de equipos técnicos.

# Frameworks para aplicaciones LLM

## Gradio

- **Prototipado rápido:** Interfaces en minutos
- **Deploy automático:** Hugging Face Spaces
- **Ideal para:** MVP y demos

## Streamlit

- **Ecosystem maduro:** Comunidad amplia
- **Multi-page apps:** Aplicaciones complejas
- **Ideal para:** Dashboards empresariales

## Chainlit

- **Chat especializado:** Optimizado para conversaciones
- **Async nativo:** Performance óptima para LLM
- **Ideal para:** Asistentes conversacionales

**Selección:** MVP → Gradio | Empresa → Streamlit | Chat → Chainlit

# Gradio: Prototipado instantáneo

Crear interfaces conversacionales con código mínimo, ideal para validación rápida de conceptos.

## Ventajas para GenBI

- **gr.ChatInterface:** Chat nativo
- **File processing:** Upload automático CSV/PDF
- **Plot rendering:** Matplotlib, Plotly integrado
- **Deploy automático:** Hugging Face Spaces

```
import gradio as gr

def chat_fn(message, history):
    # Procesar consulta
    return response

gr.ChatInterface(
    fn=chat_fn,
    title="GenBI Assistant"
).launch()
```

# Streamlit: Aplicaciones empresariales

Para aplicaciones complejas con múltiples páginas y componentes avanzados.

## Capacidades clave

- `st.chat_message`: Sistema de chat robusto
- **Session state**: Persistencia entre interacciones
- **Custom components**: Extensibilidad total

**Arquitectura típica:** - Sidebar: Configuración DB - Main: Chat con datos - Tabs: Visualizaciones



# Chainlit: Chat especializado

Diseñado específicamente para aplicaciones conversacionales con LLM, con funcionalidades que otros frameworks requieren desarrollo custom.

## Características clave

- **Message streaming:** Respuestas en tiempo real
- **Multi-modal:** Texto, imágenes, archivos
- **LangChain native:** Integración directa
- **Observability:** Tracking automático de llamadas

**Ideal para:** Asistentes de datos conversacionales y sistemas RAG complejos

# Modelos especializados: Text-to-SQL

Modelos que transforman preguntas naturales en consultas SQL ejecutables, eliminando la principal barrera técnica en análisis de datos.

## Modelos destacados

- [Qwen-2.5-3b-Text\\_to\\_SQL](#): Multilingüe, optimizado
- [text-to-sql-with-table-schema](#): Manejo de esquemas complejos
- [CodeLlama SQL variants](#): Performance empresarial

Ejemplo: "¿Top 5 productos más vendidos?" ↓

```
SELECT producto, SUM(cantidad)
FROM ventas
GROUP BY producto
ORDER BY SUM(cantidad) DESC
LIMIT 5;
```

# Generación de visualizaciones

En la práctica, no existen modelos especializados únicamente en generar gráficos. Los sistemas GenBI usan prompt engineering avanzado con modelos generales.

## Enfoques efectivos

- **GPT-4/Claude:** Con prompts específicos de visualización
- **CodeLlama/DeepSeek-Coder:** Modelos de código general
- **Prompt engineering:** Templates para chart types

**Pipeline típico:** 1. Análisis de datos (tipos, distribuciones) 2. Detección de intent ("tendencias", "comparación") 3. Generación código Plotly/Matplotlib 4. Renderizado en interface

# Recursos del curso

## Enlaces esenciales

## Frameworks

- [Gradio Documentation](#)
- [Chainlit GitHub](#)
- [Streamlit for LLMs](#)

## Modelos especializados

- [Qwen Text-to-SQL](#)
- [HuggingFace Text-to-SQL Collection](#)

## GitHub del curso

## Contacto:

[LinkedIn - Camilo Vega](#)