

RAGs: La Clave para Potenciar ChatGPT y los Modelos de Lenguaje



1



Clases teóricas



Casos de uso



Ejercicios



Laboratorios

2

Programación del curso

Programación Parte 1

- Introducción al curso
- Introducción a la **IA Gen** y LLMs
- **Evolución** de los LLMs
- Fundamentos de los RAGs
- **Herramientas** para desarrollar un RAG
- **CustomGPTs** para el desarrollo de RAGs
- Componentes del RAG

Programación Parte 2

- **Flowise** para el desarrollo de RAGs
- Indexing Pipeline
- **Embeddings**
- Almacenamiento en **BD Vectoriales**
- RAG Pipeline
- Modelos **Open-Source**
- (**Opcional**) Evaluación



Faiss

3

Mi experiencia

+ 5 años de experiencia en el **campo del Data Science y de la inteligencia Artificial**

Cloud MVP

Amplia **experiencia profesional** en el desarrollo de modelos de IA y LLM

Certificaciones en Data (Azure ML, Data Engineer, Azure Cognitive, etc)

Experiencia en dar **formación** (Coursera, Udemy, etc.)



4

2

¿Cómo te gustaría que fuera el curso?

1. Breve presentación de cada uno
2. ¿Cómo te gustaría que fuera el curso?
3. ¿Qué te gustaría aprender?
4. ¿Cómo lo vas a aplicar?



- 1) ¿Sabes cómo dotar a ChatGPT acceso a conocimiento especializado? Resultado:
 - 2) ¿Serías capaz de entrenar un LLM para que responda utilizando conocimientos de tu empresa? Resultado:
 - 3) ¿Serías capaz de desarrollar un modelo similar a ChatGPT con acceso a un conocimiento específico y que no comparta información? Resultado:

5

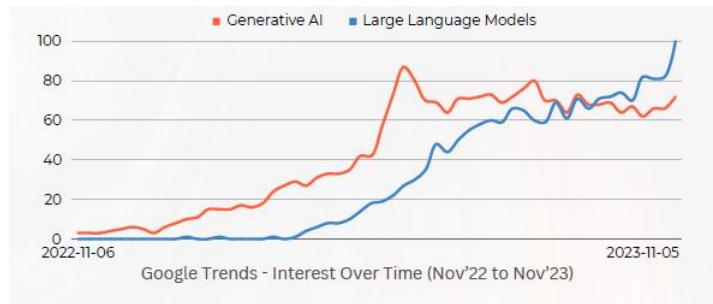


Introducción a la IA Gen y LLMs

6

Impacto de la IA Gen

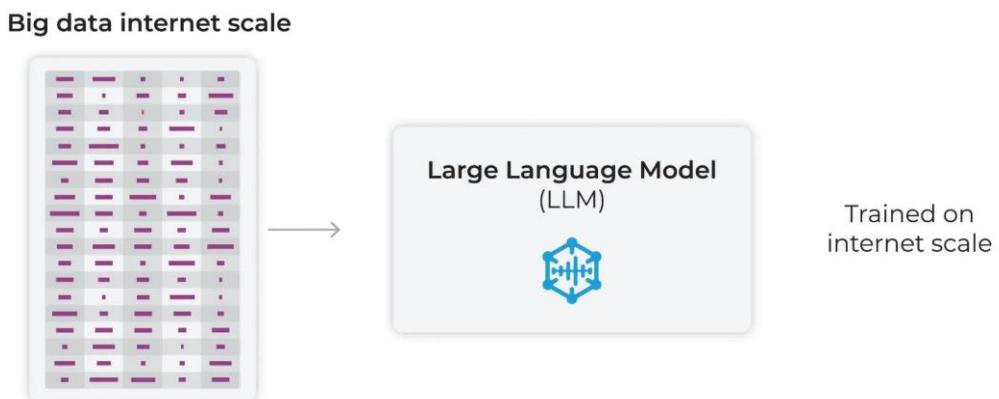
El **lanzamiento de ChatGPT** por OpenAI el 30 de noviembre de **2022** marcó un punto de inflexión en la inteligencia artificial, generando un interés sin precedentes en la IA Generativa y los Modelos de Lenguaje de Gran Escala.



7

¿Qué es un LLM?

Es un **gran modelo lingüístico** entrenado con enormes conjuntos de datos



8

Dos tipos de LLM

1. Base LLM

Predice la palabra siguiente a partir de los datos de entrenamiento del texto

[Érase una vez un unicornio](#)

que vivía en un bosque mágico con todos sus amigos unicornios.

[¿Cuál es la capital de Francia?](#)

¿Cuál es la ciudad más grande de Francia?

¿Cuál es la población de Francia?

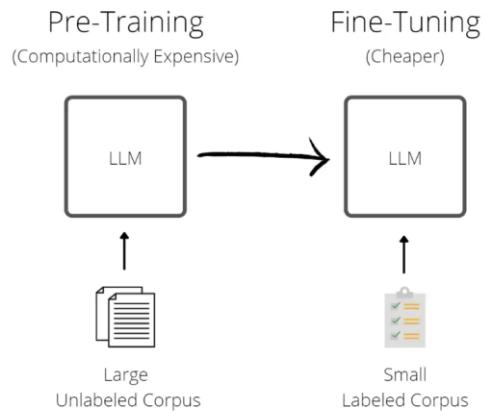
¿Cuál es la moneda de Francia?

2. Instrucción Tuned LLM

Preparados para seguir instrucciones (casi) arbitrarias.

[Dame 3 ideas para sabores de galletas.](#)

Chocolate, Matcha y Mantequilla de cacahuete



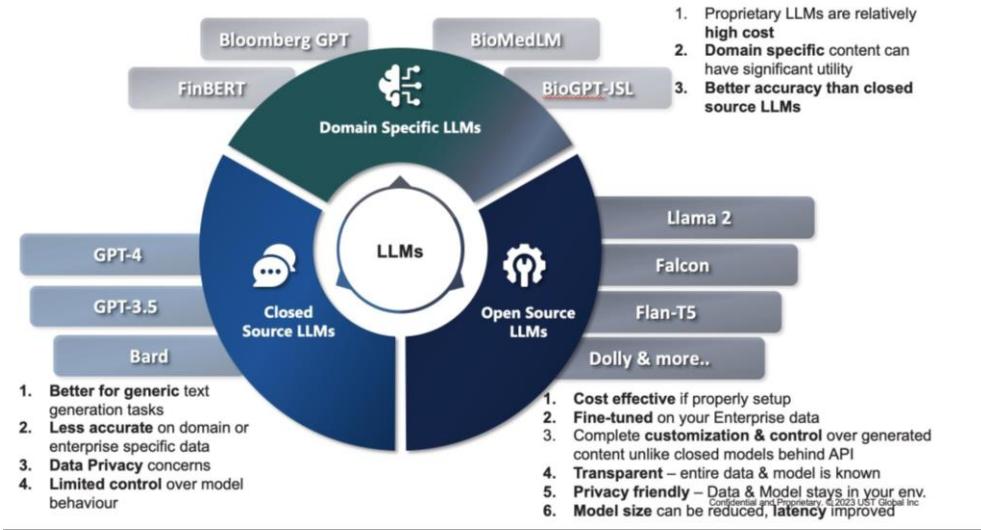
9

Modelos de Texto de IA Generativa



10

Tipos de modelos



11



**Evolución de los
LLMs**

12

Limitaciones de los LLM

Con el aumento del uso de ChatGPT, los usuarios empezaron a notar las principales **limitaciones** de los LLMs como son el hecho de que tienen un **conocimiento limitado** y que pueden sufrir “**Halucinaciones**”, además de enfrentarse a preocupaciones sobre derechos de autor, privacidad y seguridad.

Conocimiento Limitado



Halucinaciones



13

¿Qué significa alucinación?

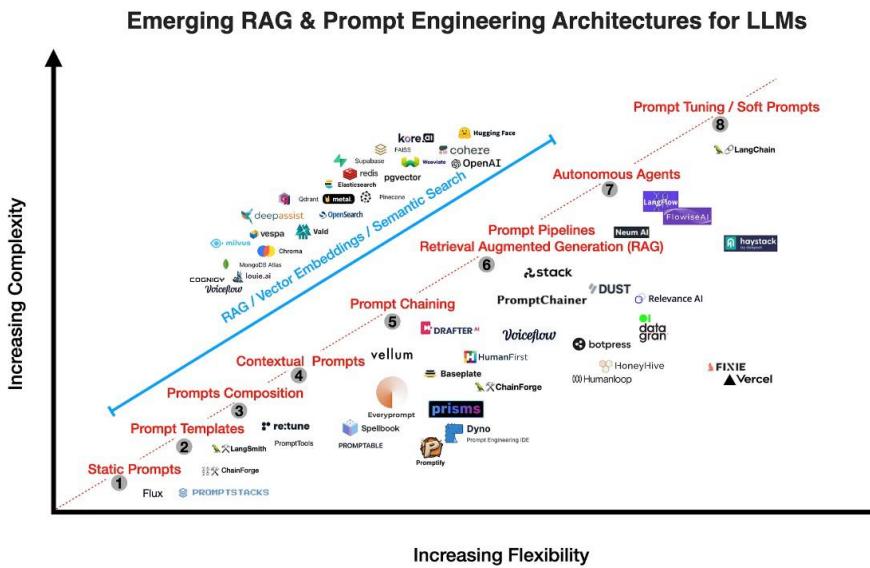
“El contenido generado **no tiene sentido** o no es **fiel** al contenido **fuente** ”.

Aunque dé la impresión de ser real y natural



14

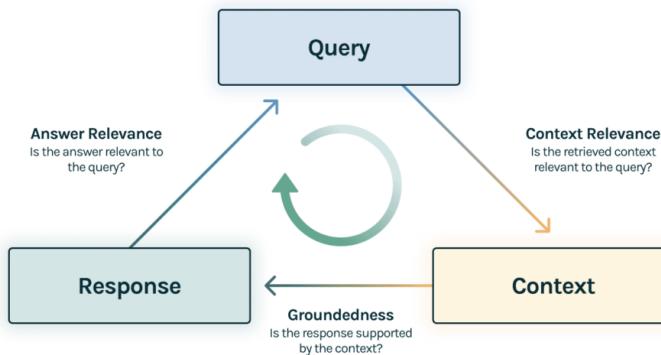
Desarrollo de Técnicas para la evolución de LLMs



15

Mitigar la alucinación a partir de RAGs

El **RAG** mejora los LLM con fuentes de conocimiento, reduciendo las alucinaciones al garantizar respuestas basadas en evidencias. Si falta **evidencia contextual**, RAG admite no saber. Sus limitaciones incluyen la necesidad de grandes bases de datos, que son costosas, y el riesgo de repetir hechos superficialmente



16

Fundamentos de los RAGS



17

El Contexto

Aunque el re-entrenamiento, Fine-tuning y aprendizaje por refuerzo pueden resolver muchos desafíos, suelen ser costosos y consumen tiempo, siendo inviables en muchos casos; en 2020, investigadores exploraron modelos que utilizaban **datos externos** para la generación de las respuestas.

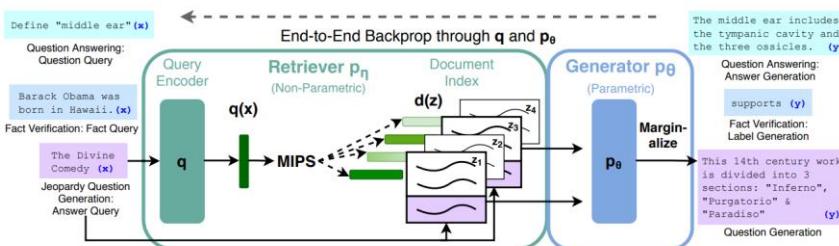
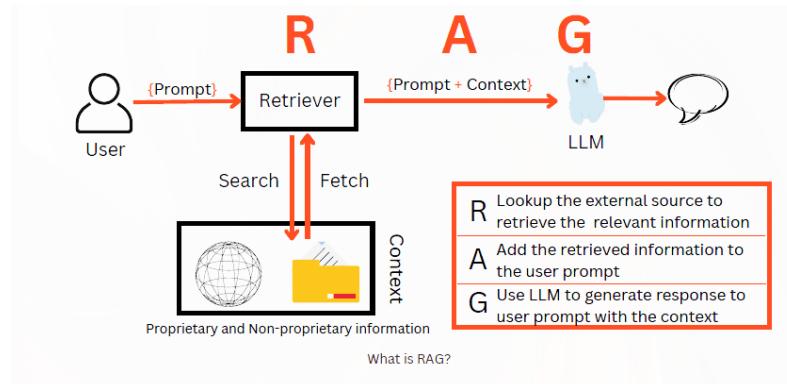


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder + Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x , we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i . For final prediction y , we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

18

Recuperación de Generación Aumentada (RAG)

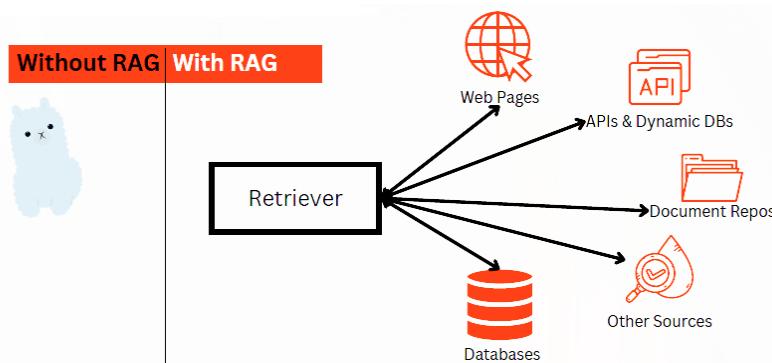
La Recuperación de Generación Aumentada (**RAG**) es una técnica que mejora la precisión de los LLM al **buscar datos actualizados antes de generar texto**, permitiendo aplicaciones más efectivas en campos especializados. RAG es un avance importante que permite a los modelos de IA superar sus limitaciones de conocimiento al integrar información relevante en tiempo real.



19

¿Cómo ayuda RAG?

RAG mejora los LLMs al proporcionar **acceso a información externa y actualizada**, lo que permite respuestas más precisas y menos propensas a errores.



20

Confianza en las respuestas

El uso de información adicional en los RAG aumenta la confianza en las respuestas de los LLMs, mejorando su precisión y relevancia **contextual**. Los sistemas LLM con RAG muestran una menor tendencia a producir **alucinaciones** comparados con aquellos sin esta funcionalidad, y ofrecen una mayor **transparencia** en sus respuestas.



21

Aplicaciones de los RAG



22

Caso de Uso: E-commerce

Un modelo de **LLM tradicional** no podría dar información sobre:

- Disponibilidad de productos
- Envío y entrega
- Reseñas de productos
- Promociones y descuentos

Un **RAG** tiene numerosas ventajas:

- Recuperación de datos en tiempo real
- Incorporación de reseñas de usuarios
- Promociones dinámicas
- Seguimiento de pedidos



23

**Herramientas
para desarrollar
un RAG**



24

Desarrollo de un RAG



25

Aternativas para crear un RAG

Existen diferentes alternativas para construir un RAG:

- Utilizar los **Custom GPTs** de OpenAi
- Utilizar la librería de Python de **LangChain**, LlamaIndex
- Utilizar una librería **No-code** (Flowise, LangFlow, GPT4ALL, etc)
- Herramientas externas como Writesonic, Run-llama, etc

Además, en función de la privacidad de datos que necesitemos podemos utilizar:

- Modelos de **Pago** (Open AI, etc)
- Modelos **Open-Source** (hugging face, gpt4all, etc)

26



CustomGPTs para el desarrollo de RAGs

27

Que son los Custom GPTs?

En noviembre de 2023, el CEO de OpenAI anunció la posibilidad de crear versiones personalizadas de ChatGPT, a las que se llama GPTs o CustomGPTs



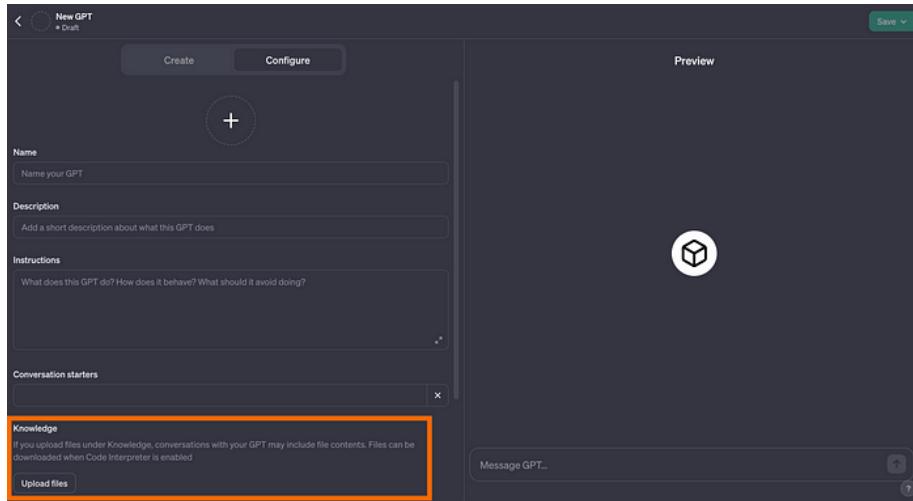
ChatGPT + Instrucciones + Conocimiento

→ **Acciones**

28

Custom GPTs

Los Custom GPTs actúan a modo de “RAG” ya que les podemos adjuntar conocimiento específico



29



30



31

Privacidad en ChatGPT

January 10, 2024

Introducing ChatGPT Team

We're launching a new ChatGPT plan for teams of all sizes, which provides a secure, collaborative workspace to get the most out of ChatGPT at work.

Enterprise privacy at OpenAI

Trust and privacy are at the core of our mission at OpenAI. We're committed to privacy and security for ChatGPT Team, ChatGPT Enterprise, and our API Platform.

Our commitments

Ownership: You own and control your data

- ✓ We do not own your business data (data from ChatGPT Team, ChatGPT Enterprise, or our API Platform)
- ✓ You own your inputs and outputs (where allowed by law)
- ✓ You control how long your data is retained (ChatGPT Enterprise)

Can OpenAI support my compliance with GDPR and other privacy laws?

Yes, we are able to execute a Data Processing Addendum (DPA) with customers for their use of ChatGPT Team, ChatGPT Enterprise, and the API in support of their compliance with GDPR and other privacy laws. Please complete our DPA form to execute a DPA with OpenAI.

What is ChatGPT Enterprise?

Who can view conversations and chat history in ChatGPT Enterprise?

What compliance standards does ChatGPT Enterprise meet?

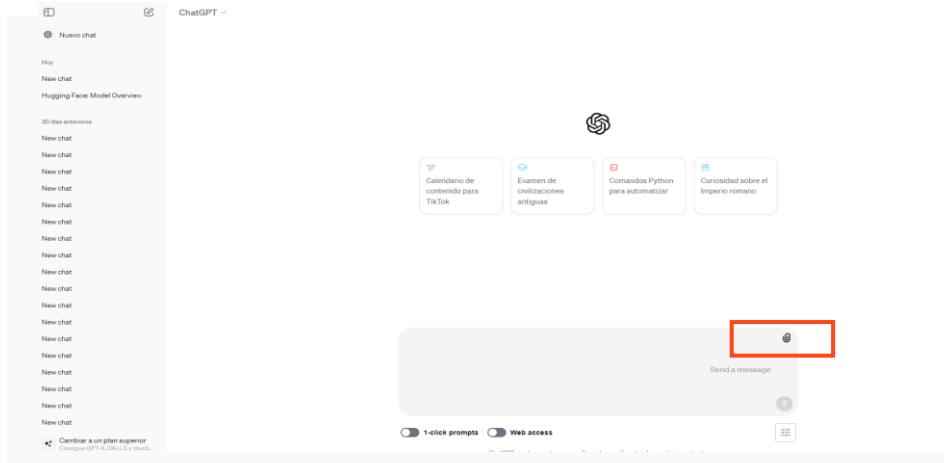
ChatGPT Enterprise has been audited and certified for SOC 2 Type 1 compliance (Type 2 coming soon). Read more in our [Trust Portal](#).

What is OpenAI's policy on data retention for ChatGPT Enterprise?

32

GPT-4o

Es la **nueva versión gratuita** y mejorada de ChatGPT. Tiene un número limitado de documentos que podemos adjuntar



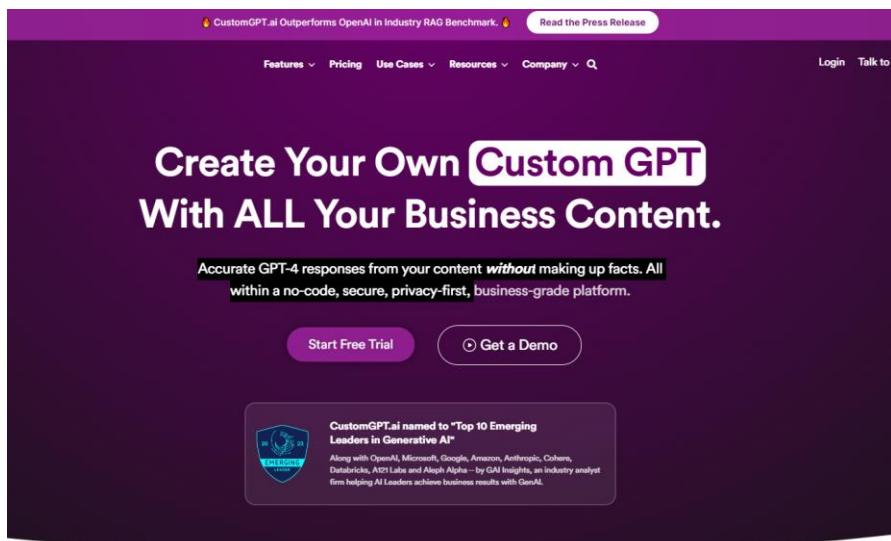
33

Alternativas: Writesonic



34

Alternativas: CustomGPT.ai



35

Privacidad en otras Plataformas

36

Desarrollo de un RAG Básico

Contexto

Deberás desarrollar un asistente para ayudar a las nuevas incorporaciones del departamento de RH

Ejercicio

Deberás crear un **RAG** Básico utilizando la plataforma de **Writesonic** o **GPT-4o** y el documento de "Guía de RH". Deberás comprobar que funciona adecuadamente y que utiliza la documentación a la hora de generar las respuestas

Material: Laboratorios Prácticos/Desarrollo de RAG Básico/ Guía de RH.pdf

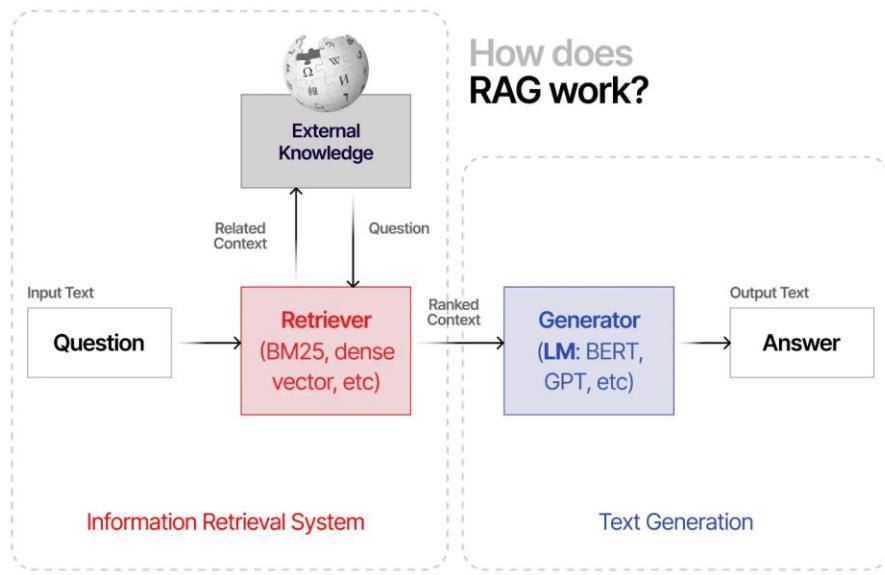
37



Componentes del RAG

38

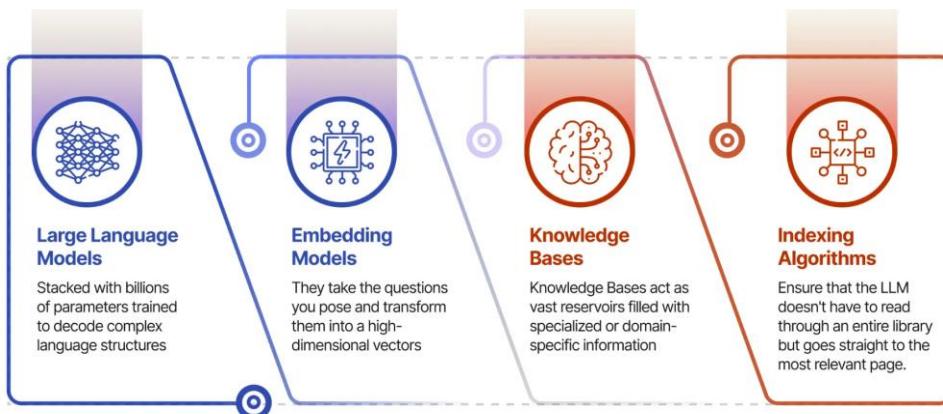
Arquitectura del RAG



39

Los Pilares

Los pilares incluyen **modelos de lenguaje** grandes como núcleo, modelos integrados (**embeddings**) como puente hacia el conocimiento especializado, **bases de conocimiento** como depósitos de información y **algoritmos de indexación** para una recuperación eficiente de datos.



40

Componentes del RAG

El sistema RAG implica dos flujos de trabajo clave:

- Indexing Pipeline:** Consiste en preparar las fuentes de conocimiento. Los datos se ingieren desde su origen y se indexan, lo cual incluye dividirlos, crear incrustaciones (embeddings) y almacenarlos.
- RAG Pipeline:** Se centra en el proceso de RAG propiamente dicho, donde se toma una consulta del usuario en tiempo real, se busca y recupera la información relevante del índice creado, y luego se pasa esa información al modelo para generar una respuesta



41



**LangChain para
el desarrollo de
RAGs**

42

Aternativas para crear un RAG

Existen diferentes alternativas para construir un RAG:

- Utilizar los **Custom GPTs** de OpenAi
- Utilizar la librería de Python de **LangChain**, LlamaIndex
- Utilizar una librería **No-code** (Flowise, LangFlow, etc)
- Herramientas externas como Bionic, Run-llama, etc

Además, en función de la privacidad de datos que necesitemos podemos utilizar:

- Modelos de **Pago** (Open AI, etc)
- Modelos **Open-Source** (hugging face, gpt4all, etc)

43

Aternativas para crear un RAG

 LangChain	 LlamaIndex
<p>Use cases: Good for applications that need enhanced AI capabilities, like language understanding tasks and more sophisticated text generation</p> <p>Features: Stands out for its versatility and adaptability in building robust applications with LLMs</p> <p>Agents: Makes creating agents using large language models simple through their agents API</p>	<p>Use cases: Good for tasks that require text search and retrieval, like information retrieval or content discovery</p> <p>Features: Excels in data indexing and language model enhancement</p> <p>Connectors: Provides connectors to access data from databases, external APIs, or other datasets</p>

44

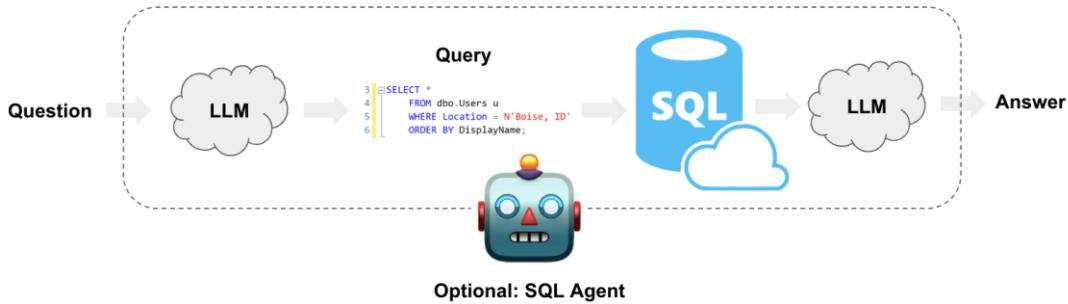
Langchain

LangChain es un marco de código abierto diseñado para el desarrollo de soluciones basadas en IA Generativa. Facilita la combinación de LLM como GPT-4 con fuentes externas de computación y datos (bases de datos, plataformas en la nube, API web, Blockchain, etc.).



45

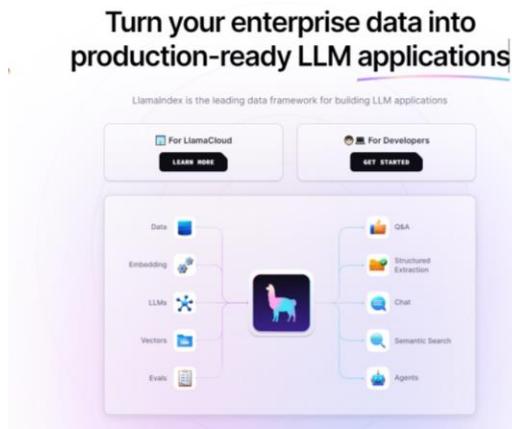
DEMO: RAG con acceso a BD



46

Llamaindex

Llamaindex es un framework diseñado para desarrollar **aplicaciones con LLMs**, gestionando efectivamente datos estructurados, semi-estructurados y no estructurados desde más de 160 fuentes. Ofrece diversas formas de indexación, facilitando las búsquedas y recuperación de información



47



**Flowise para el
desarrollo de
RAGs**

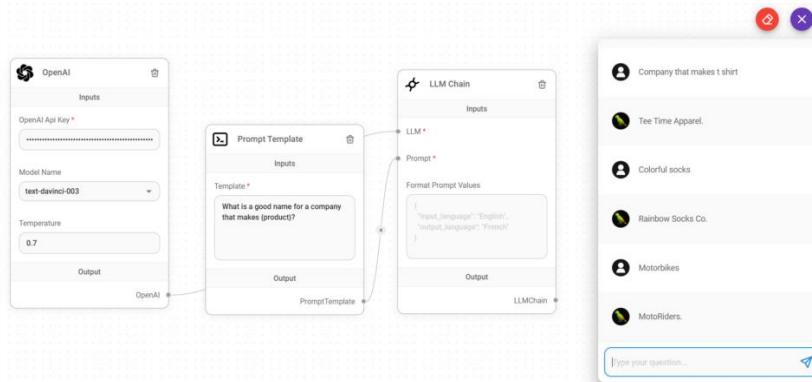
48



49

Que es Flowise?

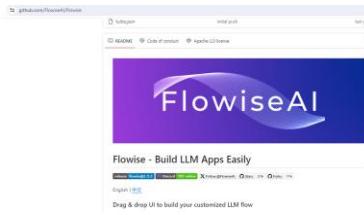
Flowise es una interfaz de arrastrar y soltar que te permite construir flujos personalizados de LLM utilizando LangchainJS, que es un marco de trabajo para desarrollar aplicaciones impulsadas por modelos de lenguaje como GPT-3.5 o GPT-4 de OpenAI.



50

Instalación de Flowise

- Acceder al GitHub: <https://github.com/FlowiseAI/Flowise>



- Acceder a CMD



3. Seguir el Quick Start

Quick Start

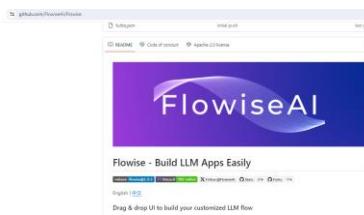
Download and Install [NodeJS](#) >= 18.15.0 3

- Install Flowise
`npm install -g flowise` 4
- Start Flowise
`npx flowise start` 5
- With username & password
`npx flowise start --FLOWISE_USERNAME=user --FLOWISE_PASSWORD=1234` 6
- Open <http://localhost:3000>

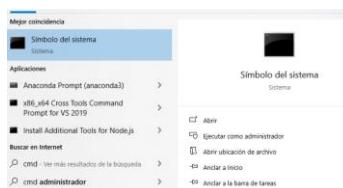
51

Acceso (una vez ya instalado)

- Acceder al GitHub: <https://github.com/FlowiseAI/Flowise>



- Acceder a CMD



3. Seguir el Quick Start

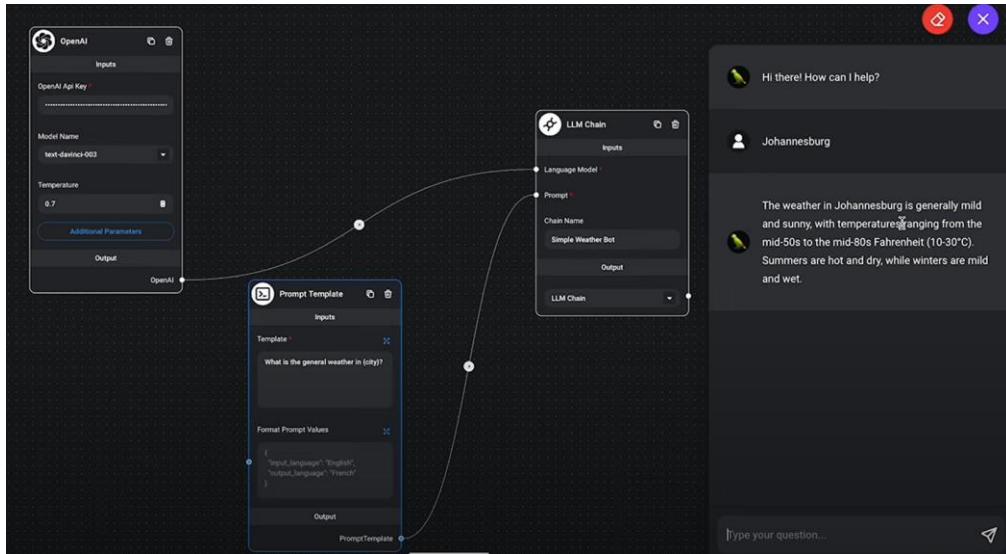
Quick Start

Download and Install [NodeJS](#) >= 18.15.0

- Install Flowise
`npm install -g flowise` 3
- Start Flowise
`npx flowise start` 4
- With username & password
`npx flowise start --FLOWISE_USERNAME=user --FLOWISE_PASSWORD=1234` 5
- Open <http://localhost:3000>

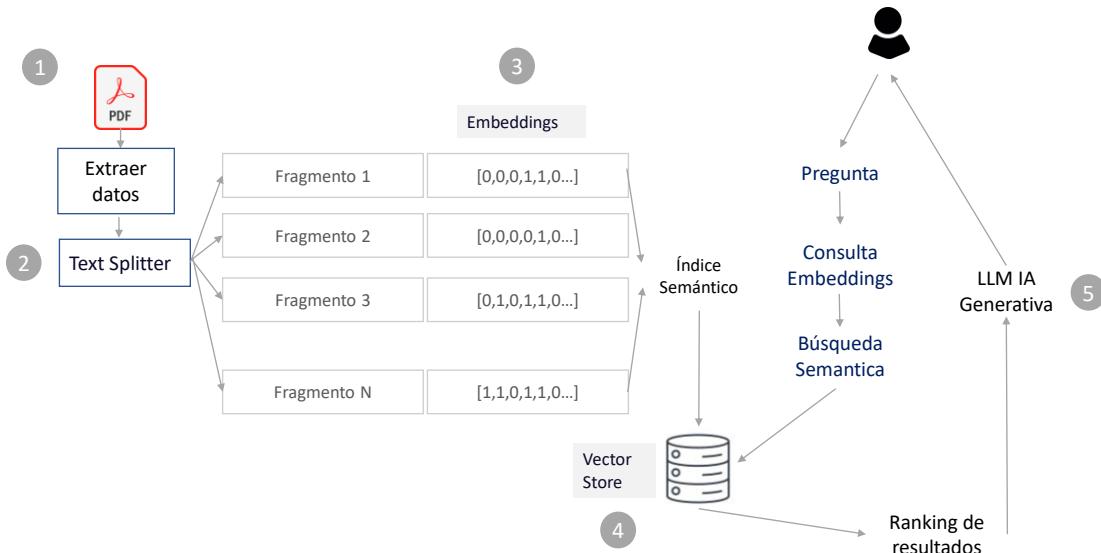
52

Cómo funciona Flowise?



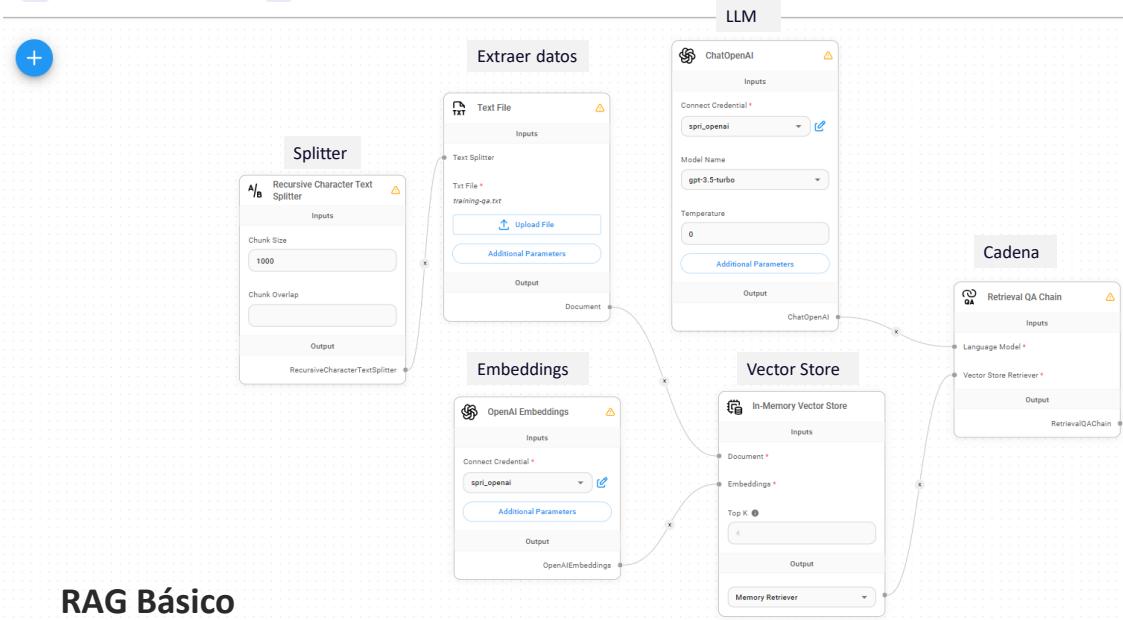
53

Arquitectura del RAG



54

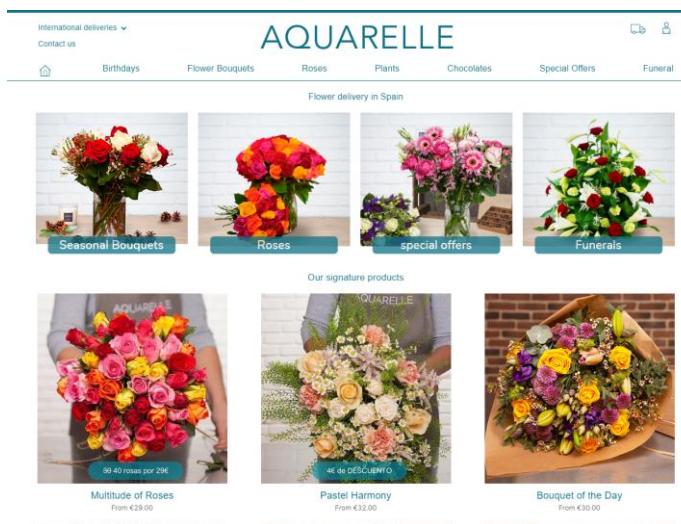
Ejercicio 3.1 RAG



RAG Básico

55

Proyecto Aplicado: RAG para un E-Commerce



Para: <https://www.aquarelle.es/shop/home>

56

Desarrollo de un ChatFlow Básico

Escenario

Una empresa de venta de flores on-line quiere desarrollar un RAG para dar soporte a sus clientes 24x7

Ejercicio

Deberás desarrollar un **RAG Básico** con Flowise. Deberá incluir el modelo de ChatGPT y una Base de Datos vectorial en Memoria (In Memory Vector Store)

Documentación: Laboratorios Prácticos/Desarrollo de RAG con Flowise/
Aquarelle_data.txt

57

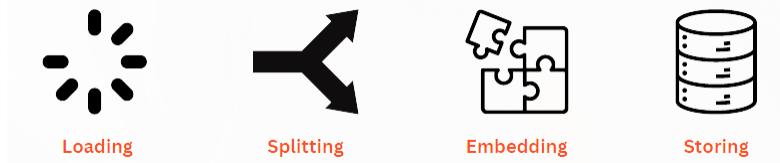


Indexing Pipeline

58

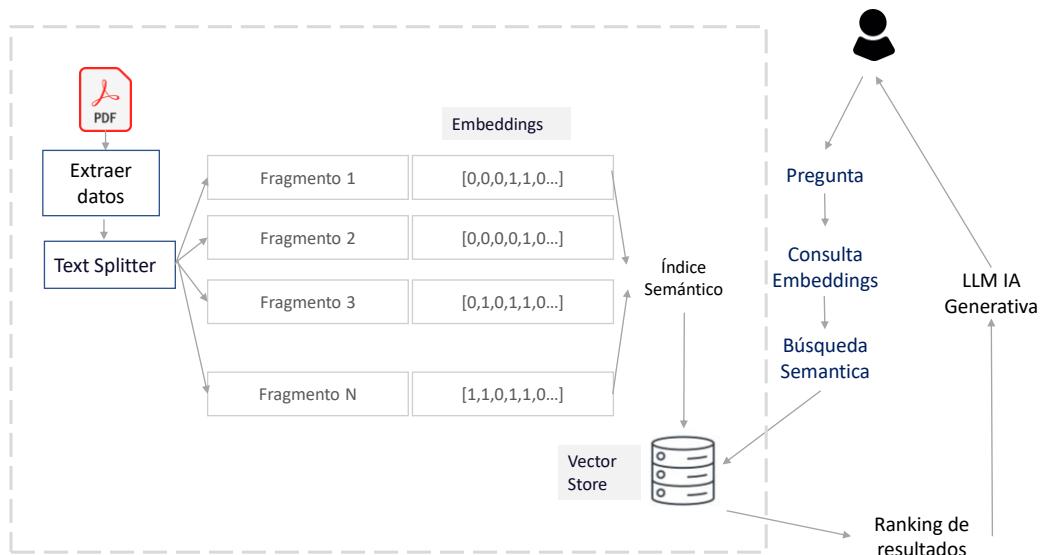
Pipeline de Indexación

El **Pipeline de Indexación** incluye la carga de datos desde diversas fuentes, la división en fragmentos manejables, la conversión a representaciones vectoriales (embeddings) y el almacenamiento en bases de datos vectoriales.



59

Arquitectura del RAG



60

Carga de Datos

La utilidad de RAG radica en acceder a **datos de variadas fuentes** como sitios web, documentos (Word, PDF), código (Python, Java), datos en formatos (JSON, CSV), APIs, directorios de archivos y bases de datos.

El primer paso es extraer la información de estas fuentes.

The screenshot shows the LangChain interface with two tabs at the top: 'LangChain' (selected) and 'LlamaIndex' (BETA). The main area is titled 'Chat Models' with a dropdown arrow. Below it is a section titled 'Document Loaders' with a collapse/expand arrow. A list of loaders is shown with icons and descriptions:

- Airtable**: Load data from Airtable table
- API Loader**: Load data from an API
- Apify Website Content Crawler**: Load data from Apify Website Content Crawler
- Cheerio Web Scraper**: Load data from webpages
- Confluence**: Load data from a Confluence Document
- CSV File**: Load data from CSV files
- Docx File**: Load data from Docx files

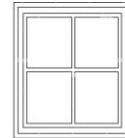
61

Fragmentación del Documento

La división de documentos en **fragmentos manejables** es un paso clave para la indexación en RAG, mejorando la búsqueda y adaptándose al tamaño del contexto de los LLMs.



Facilidad de Búsqueda



Tamaño del Contexto de los LLMs

62

Tipos de Fragmentadores (Splitters)

1. **División por Carácter:** Cortar texto basándose en un carácter específico, con fragmentos de un tamaño determinado por el número de caracteres.
2. **División Recursiva por Carácter:** Similar a la división por carácter, pero utilizando una lista de caracteres para dividir el texto de manera jerárquica.
3. **División por Tokens:** Utiliza tokenizadores específicos para dividir el texto en fragmentos, cuyo tamaño se mide por el número de tokens.
4. **División Especializada:** Centrada en mantener juntos textos con contexto común, útil para respetar la estructura de documentos específicos como HTML o Markdown.
5. **Compresión Contextual:** Reducir documentos a las partes más relevantes para la búsqueda.

63

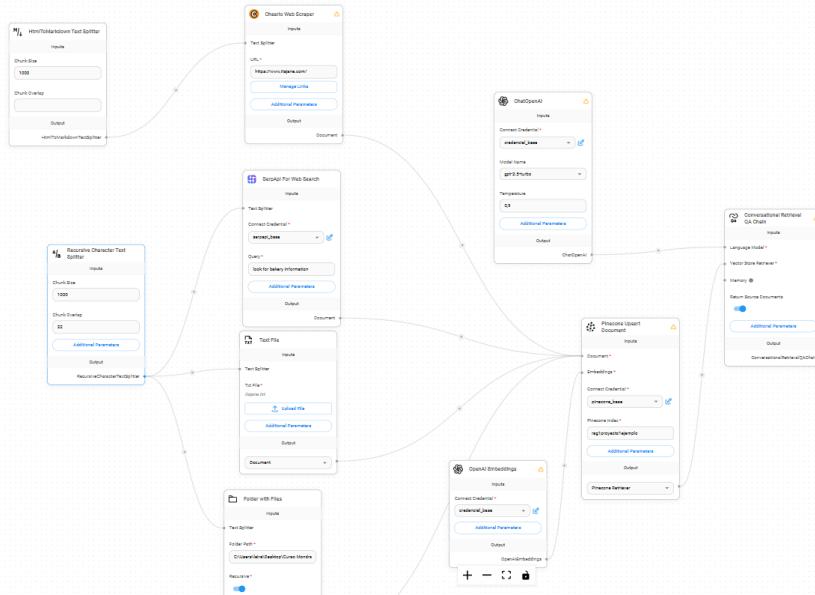
Pasos a seguir

Para asegurar la calidad de los datos en la división de documentos, sigue estos pasos:

- **Procesamiento Previo:** Antes de determinar el tamaño óptimo de los fragmentos, procesa los datos. Esto puede incluir eliminar etiquetas HTML o elementos que generan ruido, especialmente en datos de la web.
- **Consideración de Factores:** Toma en cuenta la naturaleza del contenido (como mensajes cortos o documentos largos), las características del modelo de embedding y los límites de tokens.
- **Pruebas y Almacenamiento:** Experimenta con diferentes tamaños de fragmentos, crea embeddings para estos tamaños y guárdalos en tu índice o índices.
- **Evaluación:** Realiza una serie de consultas para evaluar la calidad y comparar el rendimiento de los distintos tamaños de fragmentos.

64

DEMO



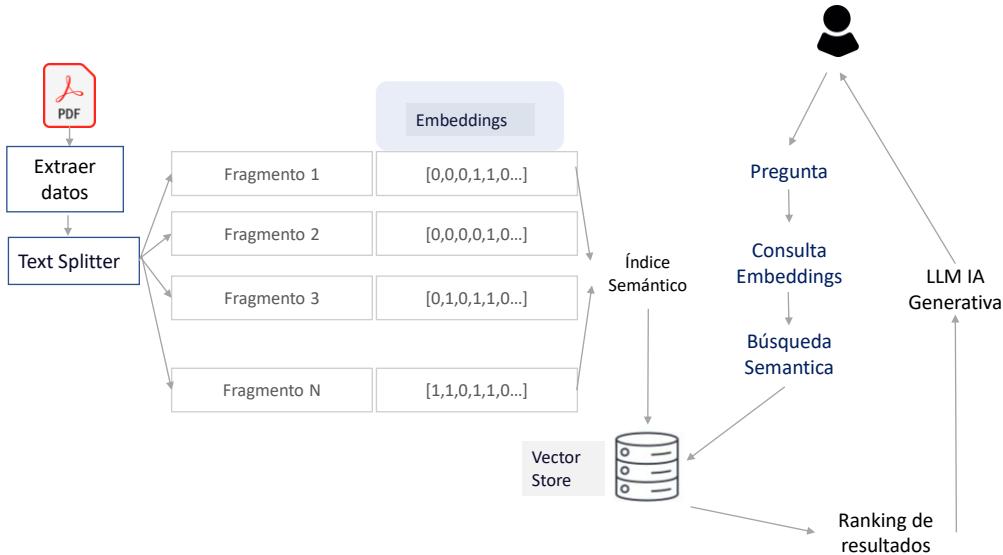
65



Embeddings

66

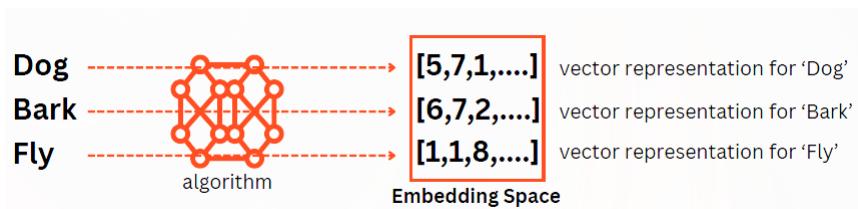
Arquitectura del RAG



67

Embeddings

Los **embeddings** son representaciones vectoriales que reflejan las **relaciones** significativas entre los datos. Se utilizan modelos como word2vec, GLOVE y BERT para generar estos vectores numéricos. Estos vectores se almacenan luego en bases de datos vectoriales para su eficiente recuperación y uso.



68

Ranking de Embeddings

Spaces = mteb leaderboard ⚡ like 2.2k Running

Massive Text Embedding Benchmark (MTEB) Leaderboard. To submit, refer to the [MTEB GitHub repository](#). Refer to the [MTEB paper](#) for details on metrics, tasks and models.

Overall Bitext Mining Classification Clustering Pair Classification Re-ranking Retrieval STS Summarization

English Chinese French Polish

Overall MTEB English leaderboard 🌐

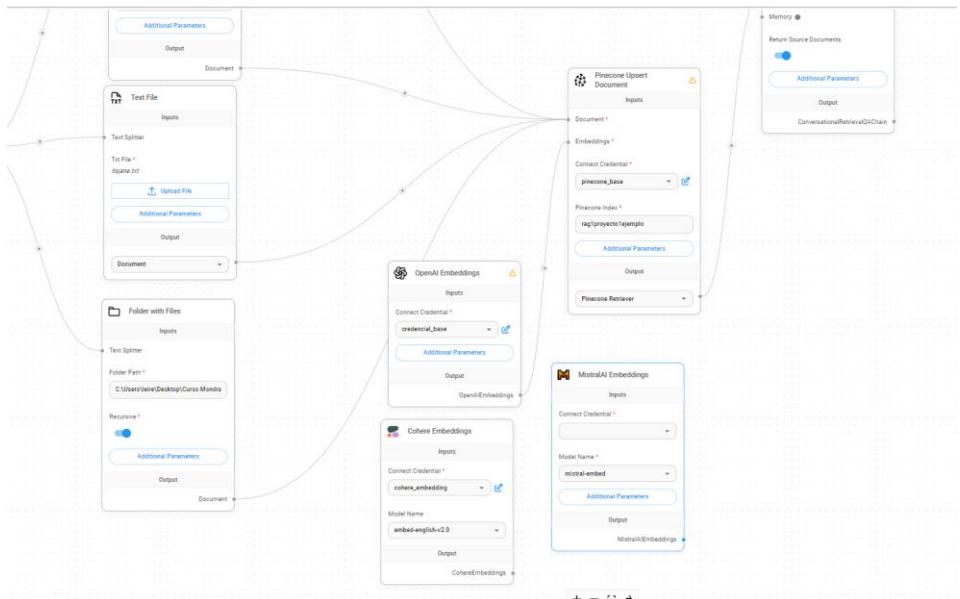
- Metric: Various, refer to task tabs
- Languages: English

Rank	Model	Model Size (GB)	Embedding Dimensions	Max Tokens	Average (56 datasets)	Classification Average (12 datasets)	Clustering Average (11 datasets)	Pair Classification Average (3 datasets)	Re-ranking Average (4 datasets)	Retrieval Average (15 datasets)	S A C d
1	SFR-Embedding-Mistral	14.22	4096	32768	67.56	78.33	51.67	88.54	68.64	59	8
2	voyage-lite-02-instruct	1024	4000	67.13	79.25	52.42	86.87	58.24	56.6	8	8
3	GritLM-7B	14.48	4096	32768	66.76	79.46	50.61	87.16	68.49	57.41	8
4	e5-mistral-7b-instruct	14.22	4096	32768	66.63	78.47	50.26	88.34	68.21	56.89	8
5	GritLM-Bx7B	93.41	4096	32768	65.66	78.53	50.14	84.97	59.8	55.09	8
6	echo-mistral-7b-instruct-lai	14.22	4096	32768	64.68	77.43	46.32	87.34	58.14	55.52	8
7	mxbai-embed-large-v1	0.67	1024	512	64.68	75.64	46.71	87.2	68.11	54.39	8
8	UAE-large-V1	1.34	1024	512	64.64	75.58	46.73	87.25	59.88	54.66	8
9	text-embedding-3-large	3072	8192	64.59	75.45	49.01	85.72	59.16	55.44	8	8
10	voyage-lite-01-instruct	1024	4000	64.49	74.79	47.4	86.57	59.74	55.58	8	8
11	Cohere-embed-english-v3.0	1024	512	64.47	76.49	47.43	85.84	58.01	55	8	8
12	multilingual-e5-large-instruct	1.12	1024	512	64.41	77.56	47.1	86.19	58.58	52.47	8

69

Demo: Embeddings

RAG: Proyecto de Ejemplo



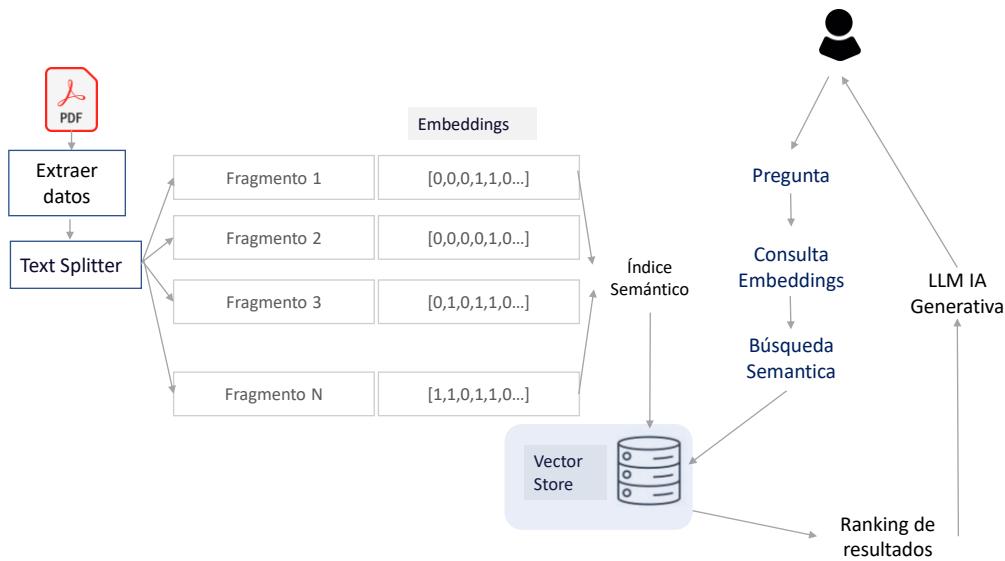
70



Almacenamiento en BD Vectoriales

71

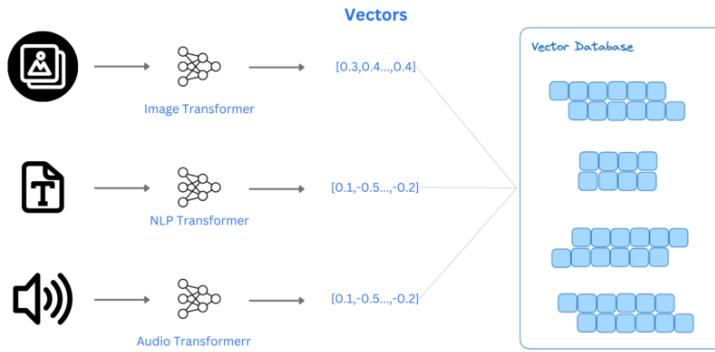
Arquitectura del RAG



72

Base de Datos Vectorial

Una **base de datos vectorial** es un sistema especializado en almacenar y gestionar vectores de características, los cuales son representaciones numéricas de datos en espacios multidimensionales. Estos vectores suelen derivarse de objetos de **datos complejos**, como textos, imágenes o sonidos, mediante procesos de modelado de aprendizaje automático.

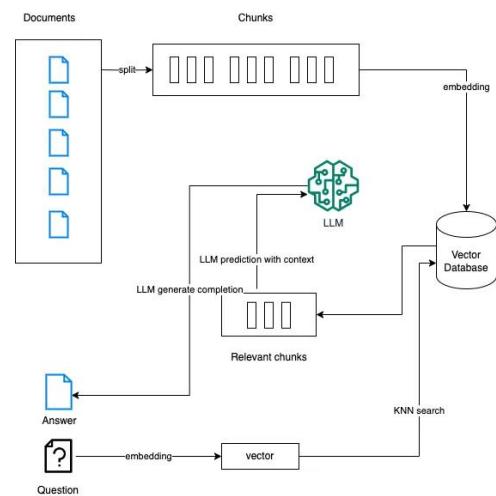


73

Función de la BD vectorial

Las **bases de datos vectoriales** optimizan las **búsquedas** de similitud mediante cálculos de distancia entre vectores para identificar elementos similares.

En **sistemas RAG**, estas bases almacenan embeddings de datos, permitiendo que, durante una consulta, se recupere información relevante de forma rápida y eficiente para generar respuestas.



74

Tipos de BD Vectoriales

Pure vector databases	Text search databases	
Milvus, zilliz, Pinecone, Weaviate, vespa, marqo, qdrant, Chroma, Vald, LanceDB	elasticsearch, Apache LUCENE, OpenSearch, Apache Solr	
Vector-capable NoSQL databases	Vector libraries	Vector-capable SQL databases
cassandra, neo4j, MongoDB, DataStax Astra, Azure Cosmos DB, redis, ROCKSET	FAISS, Annoy, Hnswlib	SingleStore, Pgvector for Postgres, ClickHouse, kinetica

75

Mas populares



Facebook AI Similarity search is a vector index released with a library in 2017



Weaviate is an open source vector database that stores both objects and vectors



Pinecone is one of the most popular managed Vector DB for large scale



Chromadb is also an open source vector database.

76

Comparativa

	Weaviate	Elastic	Milvus	Qdrant	Pinecone	Chroma
Type	Managed/Self-hosted/Open-source	Managed/Self-hosted	Self-hosted/Open-source	Managed/Self-hosted/Open-source	Managed	In-memory single-process oriented library/Open-source
Purpose-built for Vectors	✓	✗	✓	✓	✓	✓
Database rollback	✓	✗	✓	✓	✓	✗
Consistency	Tunable	Eventual	Tunable	Eventual & Tunable	Eventual	N/A
Support for both stream and batch of vector data	✓	✓	✓	✗	✗	✗
Binary Vector support	✓	✓	✓	✗	✓	✓
SDK	Python, Java, Go	Python, Java, Go, C++, Node, Rust, Ruby, .NET, PHP	Python, Java, Go, C++, Node	Python, Go, Rust	Python, Node	Python, Node

Source: zilliz.com, objectbox.io, press.ai, elastic.co



77

Consideraciones Importantes



BD Vectoriales Persistentes

Plataformas diseñadas para almacenar y gestionar de manera eficiente grandes volúmenes de vectores de características a largo plazo.



Índices Vectoriales Temporales

Estructuras de almacenamiento de corta duración adecuadas para datos que cambian frecuentemente o tienen una relevancia temporal.



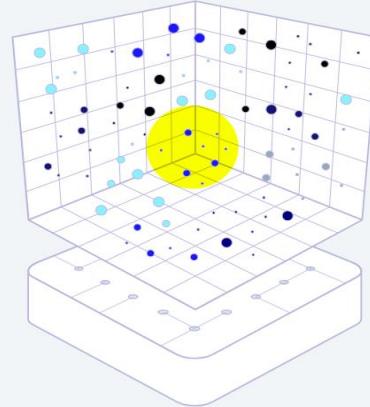
Small Data

Implica el manejo de volúmenes de datos relativamente pequeños, donde los procesos son menos complejos y demandan menos recursos.

78

Long-Term Memory for AI

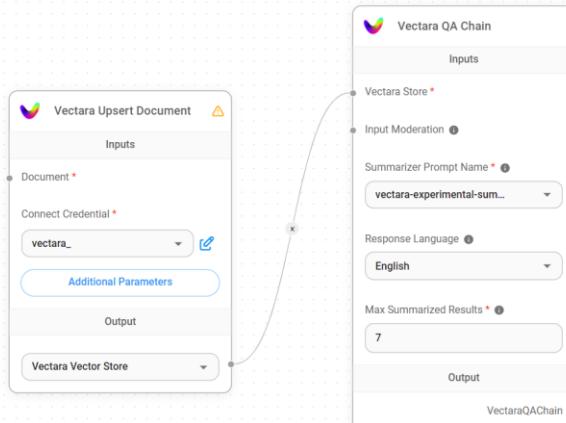
Transform your business with high-performance AI applications. Pinecone's [vector database](#) is fully-managed, developer-friendly, and easily scalable.

[Get Started](#)
[Contact Sales](#)


79

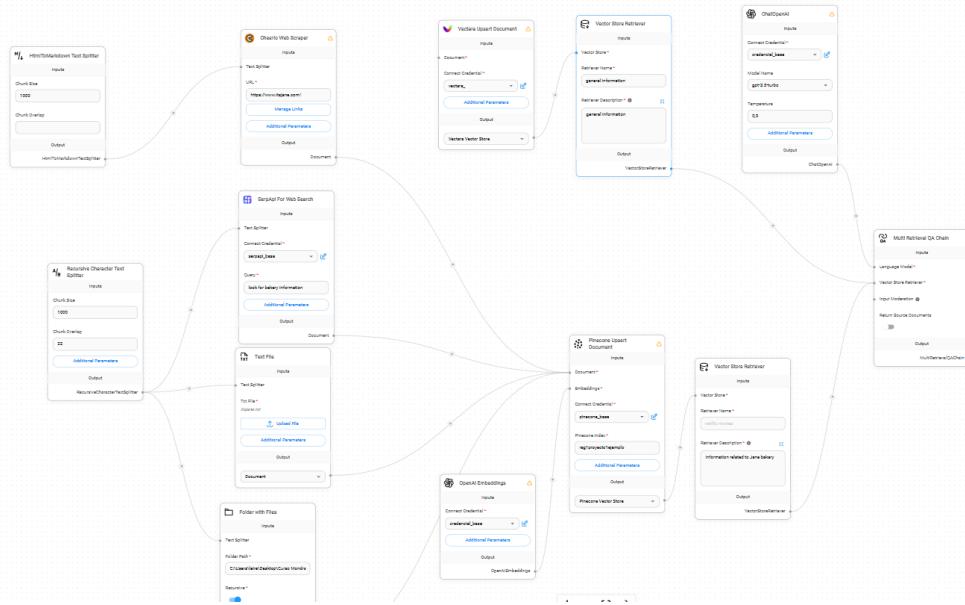
Vectara

Vectara es una plataforma que utiliza LLM y redes neuronales para mejorar la precisión y el contexto de las búsquedas, superando las limitaciones de las búsquedas convencionales. Facilita la integración de capacidades de búsqueda semántica y AI conversacional en aplicaciones.



80

Demo: RAG Avanzado



81

Desarrollo de un RAG

Escenario

Una empresa de venta de flores on-line quiere desarrollar un RAG para dar soporte a sus clientes 24x7

Ejercicio

Deberás desarrollar un **RAG** con Flowise con un Indexing Pipeline avanzado. Deberá ser capaz de obtener información de la web en **tiempo real** de <https://www.aquarelle.es/> además del PDF de Aquarelle.pdf. Además, la información procesada deberá almacenarse en **Pinecone** y deberá utilizarse el modelo de Embedding de **Cohere**

Documentación: Laboratorios Prácticos/Desarrollo de RAG con Flowise/Aquarelle

82



RAG Pipeline

83

RAG Pipeline

El "RAG Pipeline" (Retrieval-Augmented Generation Pipeline) es un proceso clave en sistemas de inteligencia artificial que combinan modelos de lenguaje con capacidades de recuperación de información.

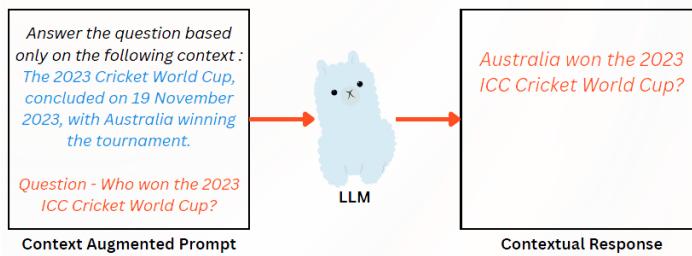


84

Aumentación y Generación

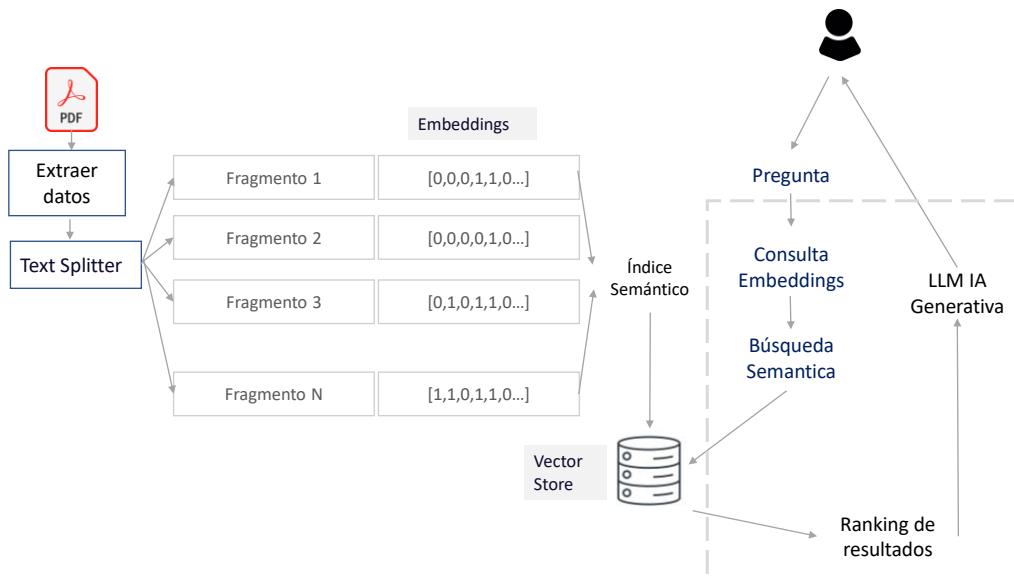
En el proceso RAG, "Augmentation & Generation" comprenden:

- Aumentación:** Combina la consulta del usuario con información relevante recuperada para enriquecer el prompt que se proporcionará al modelo de lenguaje.
- Generación:** Utiliza el prompt enriquecido para generar una respuesta con el modelo de lenguaje, buscando mayor precisión y relevancia gracias al contexto adicional.



85

Arquitectura del RAG



86

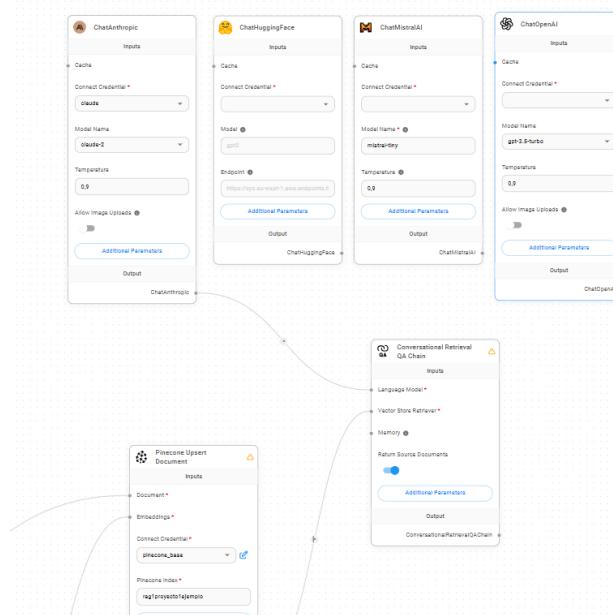
Diferentes LLMs

Flowise soporta diferentes tipos de modelos:

- **Azure ChatOpenAI:** Integrado con servicios de Azure, permite desplegar modelos de IA como GPT-4 y GPT-3.5 Turbo, ideal para aplicaciones empresariales
- **ChatGoogleGenerativeAI:** Proporciona acceso a las capacidades de IA de Google, enfocándose en la generación de texto
- **ChatLocalAI:** Permite ejecutar modelos de lenguaje de gran escala localmente o en servidores propios, adecuado para entornos donde la privacidad de los datos es crucial
- **Google VertexAI:** Parte de la suite de Google Cloud, ofrece herramientas para desplegar y gestionar modelos de IA a gran escala, con énfasis en la flexibilidad e integración con otros servicios de Google
- **ChatMistralAI:** Soporta configuraciones versátiles, ideal para flujos de trabajo complejos que incluyen sistemas de preguntas y respuestas
- **ChatOllama:** Admite modelos de lenguaje grandes y multimodales de código abierto para tareas locales que requieren manejo de texto e imágenes

87

DEMO: Diferentes LLMs



88



Open-Source

89



90

Beneficios Open-Source

Hay numerosos **beneficios** de usar modelos de lenguaje grande **open-source**:

- **Privacidad de datos**
- **Personalización**
- **Asequibilidad**
- **Democratización de la IA**



91

Qué LLM open-source es mejor?

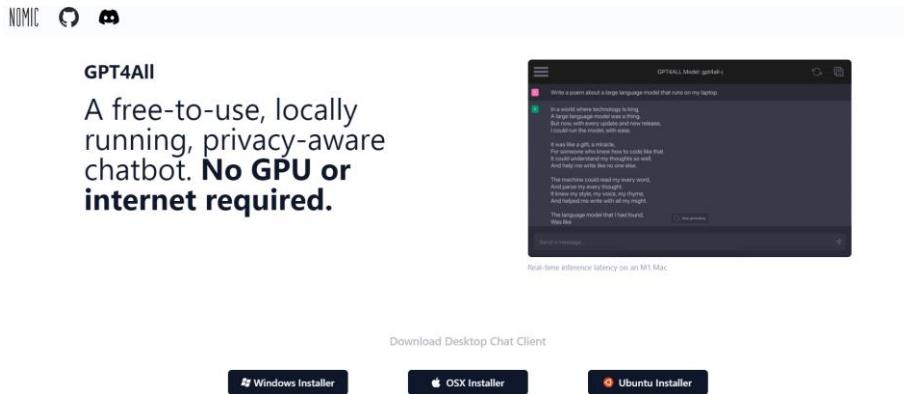
<https://lmsys.org/blog/2023-05-03-arena/>

Rank	Model	Elo Rating	Description
1	 vicuna-13b	1169	a chat assistant fine-tuned from LLaMA on user-shared conversations by LMSYS
2	 koala-13b	1082	a dialogue model for academic research by BAIR
3	 qasst-pythia-12b	1065	an Open Assistant for everyone by LAION
4	 alpaca-13b	1008	a model fine-tuned from LLaMA on instruction-following demonstrations by Stanford
5	 chatglm-6b	985	an open bilingual dialogue language model by Tsinghua University
6	 fastchat-t5-3b	951	a chat assistant fine-tuned from FLAN-T5 by LMSYS
7	 dolly-v2-12b	944	an instruction-tuned open large language model by Databricks
8	 llama-13b	932	open and efficient foundation language models by Meta
9	 stablelm-tuned-alpha-7b	858	Stability AI language models

92

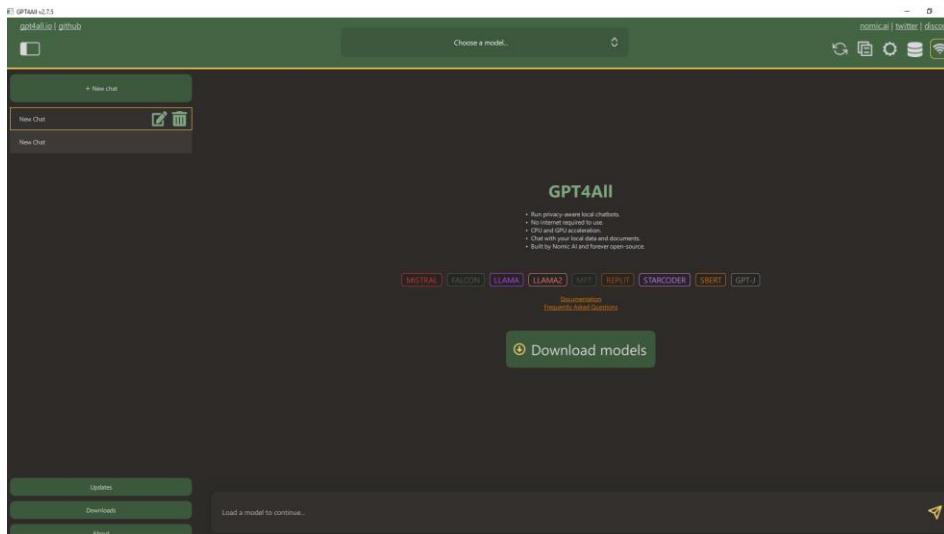
GPT4LL

GPT4All es un ecosistema de software open-source creado por Nomic AI que permite ejecutar localmente LLMs en hardware convencional, como las CPUs de ordenadores portátiles y de escritorio. Este sistema facilita la personalización y el entrenamiento de modelos potentes sin necesidad de GPUs avanzadas, asegurando **accesibilidad y privacidad**.



93

Aplicación de GPT4LL



94

LLM Open-source en Flowise

Para utilizar modelos LLM Open-Source en Flowise hay que seguir los siguientes pasos:

1. Instalación de **Docker** y **Git**
2. Crear una carpeta local y acceder desde **CMD**
3. Clonar con **Git** el proyecto de Local AI con el comando git clone <https://github.com/go-skynet/LocalAI.git>
4. Descargar desde **GPT4ALL** el modelo
5. Copiar el modelo en la carpeta de /models
6. Ejecutar el comando docker-compose up -d --pull always
7. En Flowise añadir el componente de ChatLocalAI
8. Añadir a ChatLocalAI el Base Path de http://localhost:8080/v1

95

Desarrollo de un RAG

Escenario

Una empresa de venta de flores on-line quiere desarrollar un RAG para dar soporte a sus clientes 24x7

Ejercicio

Utiliza la aplicación de **GPT4ALL** para desarrollar un RAG open-source y privado con la documentación del PDF de Aquarelle

Documentación: Laboratorios Prácticos/Desarrollo de RAG con Flowise/Aquarelle

96



Evaluación

97

La Evaluación

La **evaluación** en sistemas RAG es crucial para asegurar su efectividad, enfocándose en aspectos como relevancia, precisión y tendencia a errores de generación. Para esto, se utilizan diversos marcos y herramientas especializadas.



Evaluación de la Búsqueda y Recuperación

Se evalúa cuán precisa y relevante es la recuperación del contexto de la base de datos vectorial en relación con la consulta.



Evaluación de la Generación

Se examina la calidad de la respuesta generada, evaluando si está fundamentada en el contexto proporcionado y si es relevante para la consulta original.

98

Métricas de Evaluacion

Métricas de Evaluación:

- **Fidelidad** (Faithfulness): Mide en qué medida la respuesta se basa en los hechos del contexto recuperado.
- **Relevancia de la Respuesta** (Answer Relevance): Evalúa si la respuesta es pertinente a la consulta o prompt.
- **Relevancia del Contexto** (Context Relevance): Determina si el contexto recuperado es relevante para el prompt.
- **Precisión del Contexto** (Context Precision): Evalúa si los ítems relevantes en los contextos están clasificados adecuadamente.
- **Similitud Semántica de la Respuesta** (Answer Semantic Similarity): Compara la respuesta generada con la respuesta correcta conocida en términos de similitud semántica.

99

RAGAS

Ragas es un marco que facilita la evaluación y optimización de pipelines de RAG en modelos de lenguaje avanzados. Ofrece herramientas para medir la efectividad del texto generado y puede integrarse con sistemas CI/CD para mantener el rendimiento de manera continua.



Un conjunto de Consultas o Prompts para evaluación



Contexto Recuperado para cada prompt



Respuesta Correspondiente o Respuesta del Modelo de Lenguaje de Gran Escala (LLM)

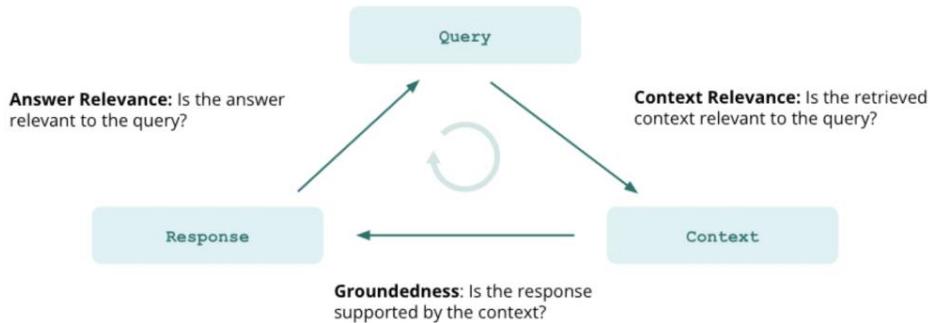


Verdad de Campo o respuesta correcta conocida

100

The RAG Triad (TruLens)

El **triángulo RAG** es un marco propuesto por TruLens para evaluar las alucinaciones a lo largo de cada arista de la arquitectura RAG.



101

Thank
You



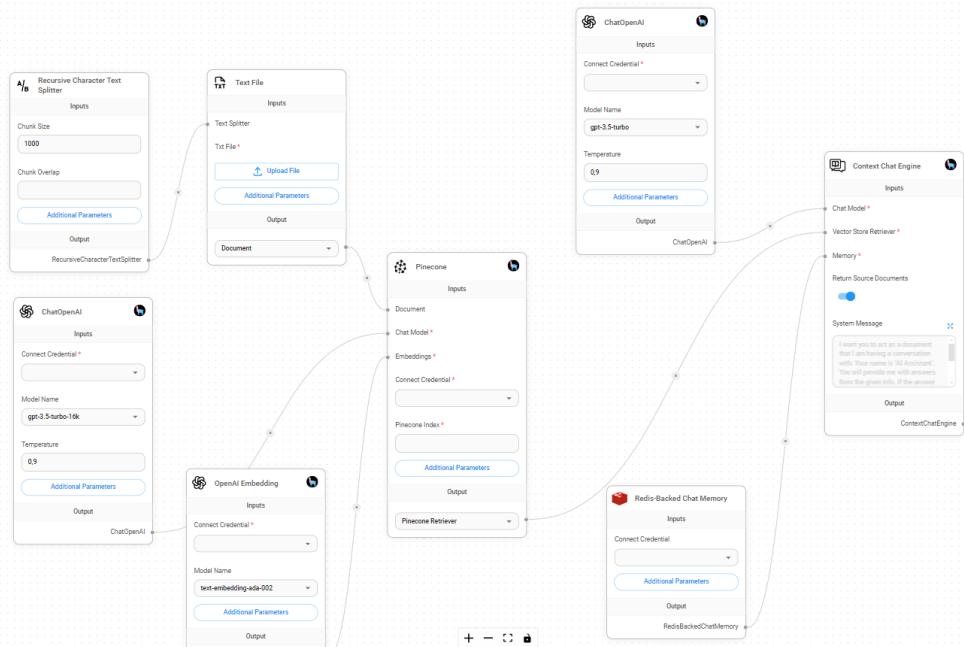
102



LlamaIndex

103

RAG: Llamaindex



104



Ollama

105

Qué es Ollama?

Ollama es una plataforma de IA Gen que permite la ejecución de modelos de **IA localmente**, utilizando infraestructura de hardware propio, lo cual mejora la privacidad y el control sobre los datos. Ofrece modelos preentrenados y opciones de personalización.



Get up and running with large language models.

Run [Llama 3](#), [Phi 3](#), [Mistral](#), [Gemma](#), and other models. Customize and create your own.

[Download !](#)

Available for macOS, Linux, and Windows (preview)

106

Qué es Ollama?

Ollama es una plataforma de inteligencia artificial generativa que permite ejecutar modelos de IA en dispositivos locales. Sus características clave son:

- **Ejecución local:** Ejecuta modelos de IA en dispositivos propios, mejorando la privacidad y el control de datos.
- **Interfaz amigable:** Fácil de usar para desarrolladores y usuarios sin experiencia técnica.
- **Modelos pre-entrenados:** Ofrece modelos de IA preentrenados para diversas aplicaciones.
- **Personalización:** Permite ajustar y entrenar modelos según necesidades específicas.

107

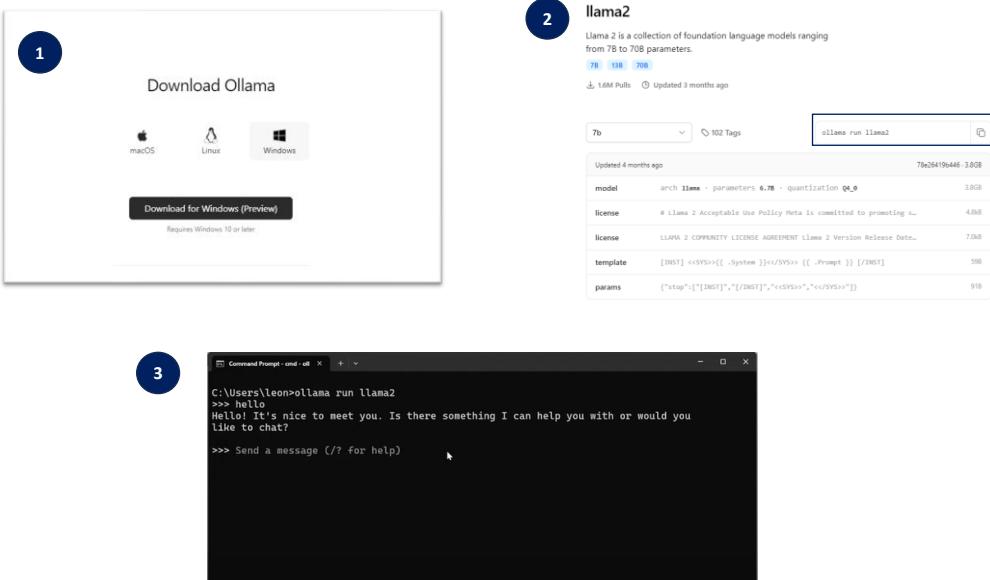
Modelos disponibles en Ollama

Los modelos disponibles en Ollama son:

- **Llama 3:** Modelo avanzado de Meta para generación y comprensión de texto, con tamaños de 8B y 70B parámetros.
- **Phi 3:** Modelo ligero de 3.8B parámetros de Microsoft, optimizado para eficiencia y rendimiento.
- **Mistral:** Modelo de 7B parámetros, excelente en tareas de razonamiento y comprensión de lenguaje.
- **Gemma:** Modelos ligeros de Google DeepMind, disponibles en versiones de 2B y 7B parámetros.
- **Command R (R+):** Modelos de 35B y 104B parámetros optimizado para interacción conversacional y tareas de contexto largo.
- **LLaVA:** Modelo multimodal que combina visión y comprensión de lenguaje.
- **Dolphin-Llama 3:** Variante de Llama 3 con habilidades avanzadas en instrucciones, conversación y codificación.
- **Stable Code:** Modelo de 3B parámetros para generación y completado de código.
- **WizardCoder:** Modelo de generación de código, disponible en versiones de 7B, 13B, 33B, y 34B parámetros.
- **Neural-Chat:** Modelo basado en Mistral afinado para múltiples dominios e idiomas, con 7B parámetros.
- **Falcon:** Modelo para resumen, generación de texto y chatbots, con versiones de 7B, 40B y 180B parámetros

108

Instalación de Ollama



109

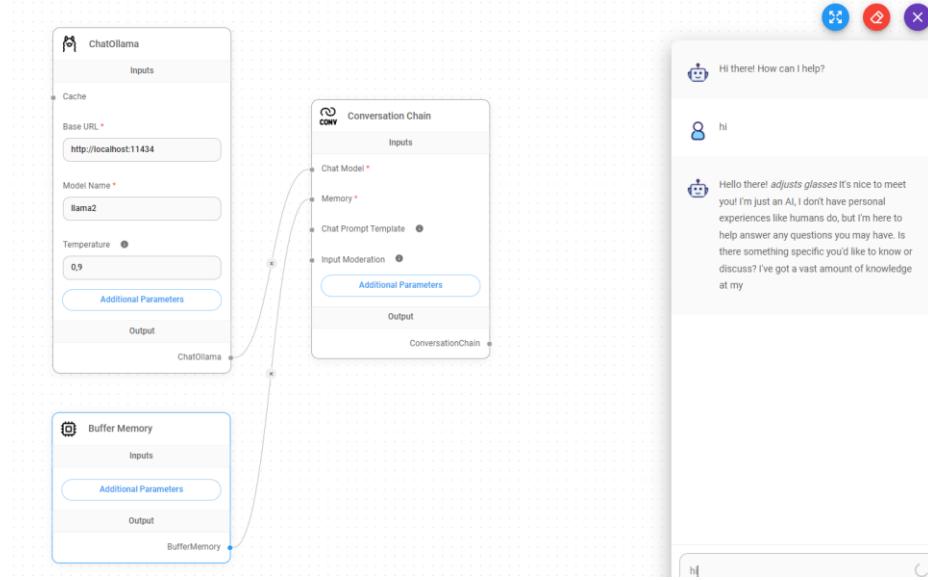
Acceso desde CMD

```
Selezionar Símbolo del sistema - ollama run llama2
Microsoft Windows [Versión 10.0.19045.4412]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\leire>ollama run llama2
>>> hellow
Hello! It's nice to meet you. How are you today?
>>> Send a message (/? for help)
```

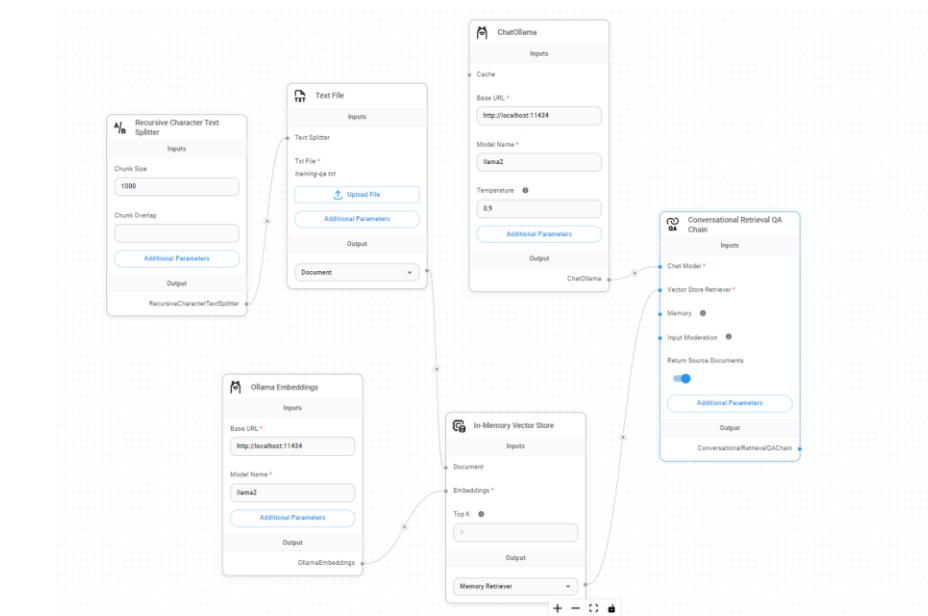
110

Integración de Ollama en Flowise



111

RAG de Ollama



112