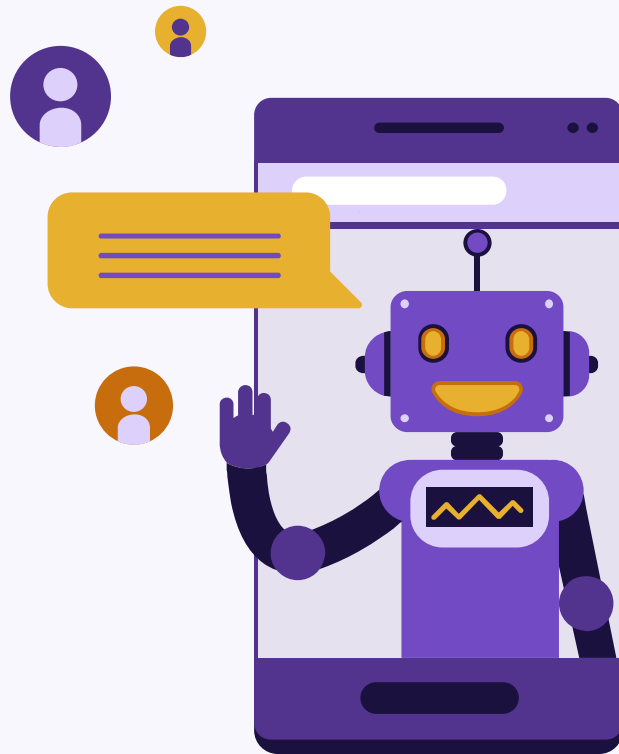


# Técnica RAG

Diego Moreno  
Johan Andrés Camilo  
Karen Vargas Hurtado



# Contenidos

**01** • • **Objetivos**

**04** • • **Requerimientos**

**02** • • **Descripción de  
la empresa**

**05** • • **Desarrollo del  
proyecto**

**03** • • **Requerimientos**

• •

07 • • Citas





## Objetivos

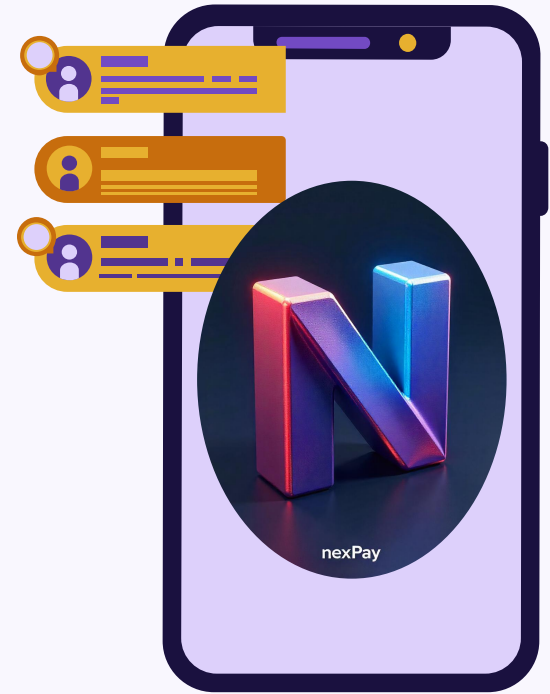


Implementar un RAG asistente que permita a sus usuarios consultar de manera eficiente y personalizada el documento cargado.

# Descripción

Historias Digitales requiere desarrollar un RAG (Retrieval Augmented Generation) referente a la historia de Colombia.

Esta tecnología mejoraría significativamente la experiencia de los usuarios, con la rapidez de encontrar la información requerida. Invertir en un RAG no sólo fortalecería el negocio, sino que también impulsará el avance de la investigación histórica.



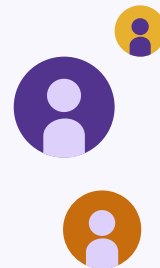
# Justificación

Historias Digitales, como líder en investigación histórica digital, busca mantenerse a la vanguardia de la tecnología. Al implementar un RAG, la empresa podrá:

Mejorar la eficiencia

Diferenciarse de la competencia

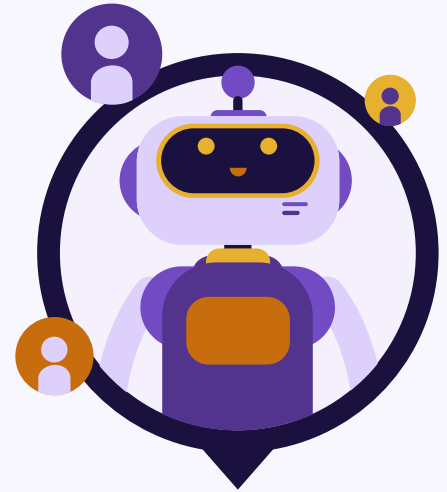
Innovar





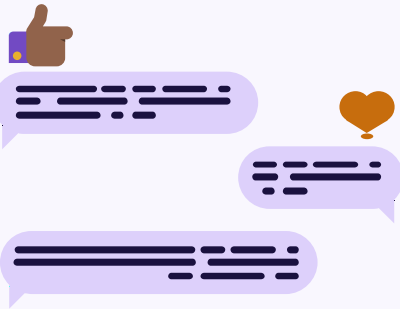
# Requerimientos Funcionales

- Consulta de documentación cargada.
- Consultas frecuentes
- Personalización de respuestas.
- Interacción en lenguaje natural.



# Requerimientos no Funcionales

- Seguridad y privacidad.
- Escalabilidad.
- Disponibilidad.
- Interfaz amigable y accesible.
- Compatibilidad tecnológica.





# Arquitectura

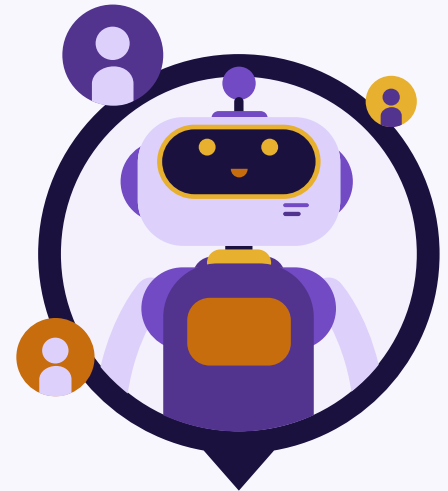
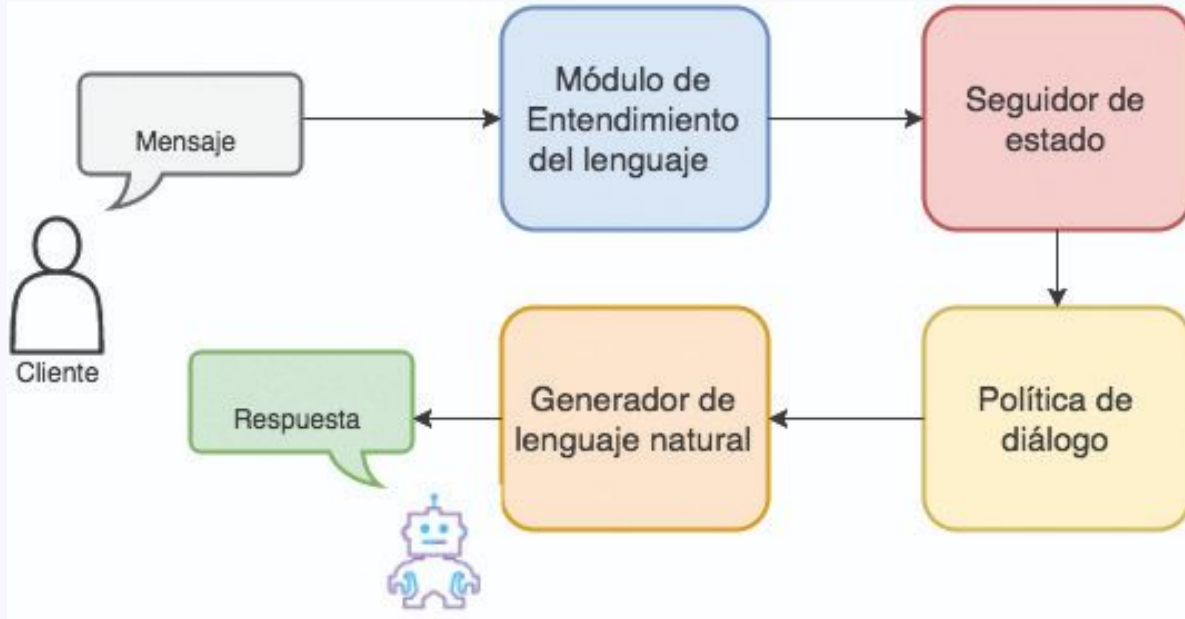


Figura 1: Esquema de un chatbot basado en inteligencia artificial  
Tomado de: <https://sg.com.mx/revista/53/chatbots-conversaci-n-natural-robots>

# Prompt

```
"""Retorna respuestas sobre la historia de Colombia. Se  
espera que la entrada sea una cadena de texto  
y retorna una cadena con el resultado más relevante. Si la  
respuesta con esta herramienta es relevante,  
no debes usar ninguna herramienta más"""  
compressed_docs = compression_retriever.invoke(text)  
resultado = compressed_docs[0].page_content  
return resultado
```

# RAG

## Conceptos clave de RAG

### Conocimiento del Modelo LLM

El Modelo puede generar alucinaciones si interactuamos directamente sobre un tema que no conoce en profundidad, pero nos permite una interacción natural.

✓ Interacción natural

### Pregunta del Usuario

El usuario puede obtener información de calidad de las BB.DD pero sin naturalidad, sin diálogo y conversación (NLG) que sí le aporta el Modelo LLM.

### BB.DD. externas

Conectar LLM y BB.DD. nos aporta 'ajuste fino' y conocimiento avanzado, pero sin las preguntas del usuario el Modelo ignora lo que de verdad es relevante.



Reference: Diseño propio JLFH - <https://www.linkedin.com/in/joseluis-hernandez/>

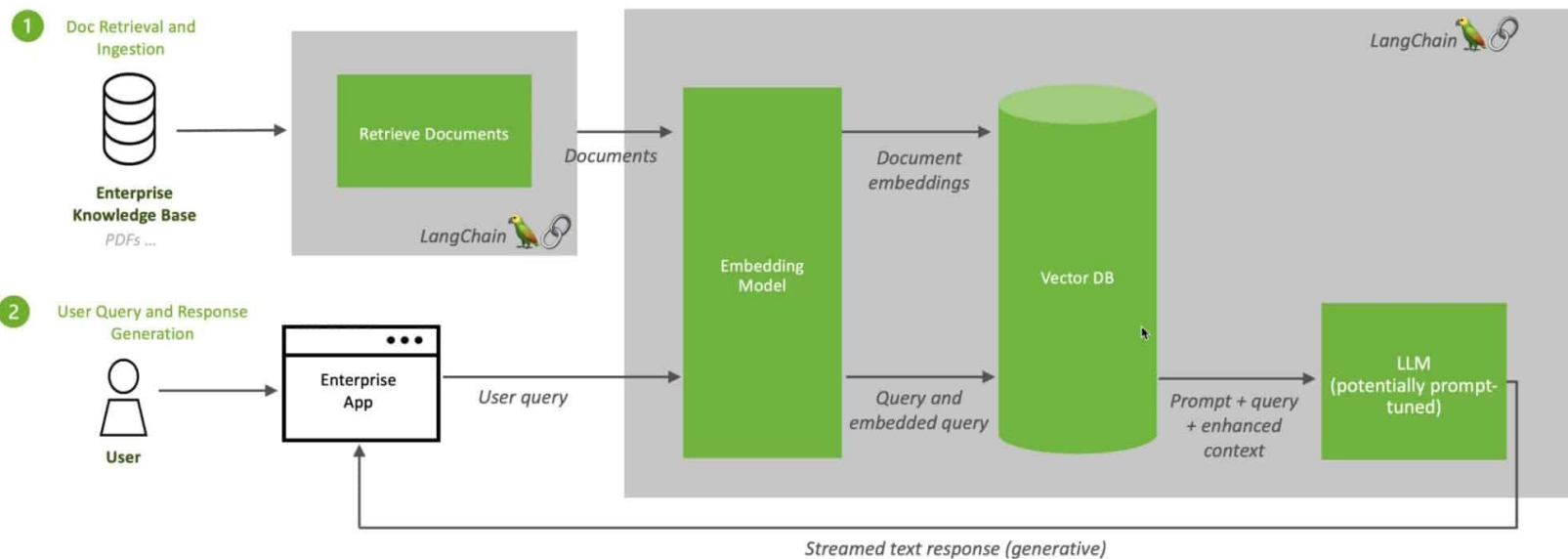
Una de las ventajas de implementar RAG es que puede ofrecer resultados precisos que aprovechan al máximo el conocimiento preexistente, pero también puede procesar y consolidar ese conocimiento para crear respuestas, instrucciones o explicaciones únicas y sensibles al contexto en un lenguaje similar al humano en lugar de simplemente resumir los datos recuperados.

Taken from: <https://www.linkedin.com/pulse/quieres-comprender-qu%C3%A9-es-rag-y-su-relaci%C3%B3n-con-los-llm-hern%C3%A1ndez-ley3f/>



# RAG

## Retrieval Augmented Generation (RAG) Sequence Diagram



**RAG**

# Desarrollo del proyecto

Taken from: <https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/>



localhost:8502

# Consulta Histórica de Colombia con LangChain

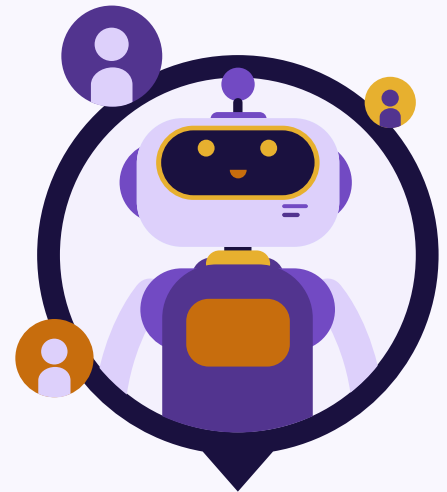
Pregunta sobre la historia de Colombia, y el modelo te responderá.

Introduce tu pregunta:

Que paso en Colombia en 1821

Respuesta:

```
{
  "input": "Que paso en Colombia en 1821"
  "chat_history": ""
  "output":
    "In 1821, Colombia was part of the Spanish colonial empire. This was a significant year in Colombian history as it marked the beginning of the Colombian War of Independence. The war ultimately led to Colombia gaining its independence from Spain in 1821."
}
```

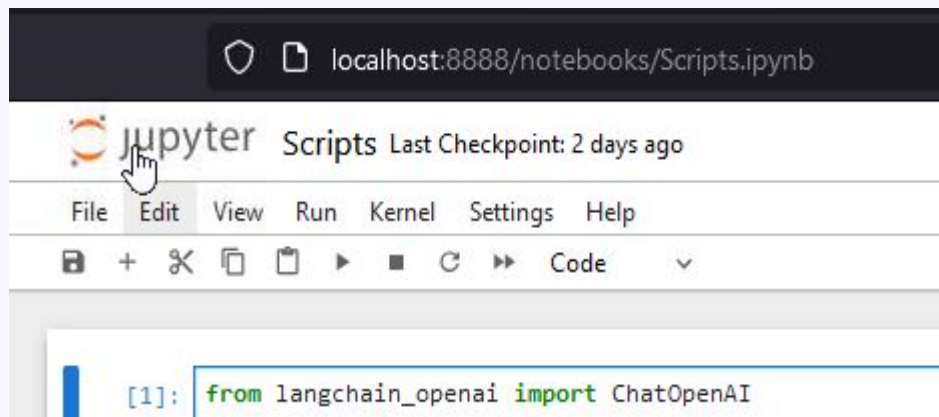


# LangChain RAG



# LangChain RAG

**Jupyter**, para este caso por temas de memoria desinstalamos anaconda e instalamos solo jupyter Instalación librería Langchain en cmd : `pip install langchain`, en cmd : `pip install openai` en cmd : `pip install langchain_openai`

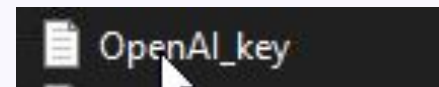
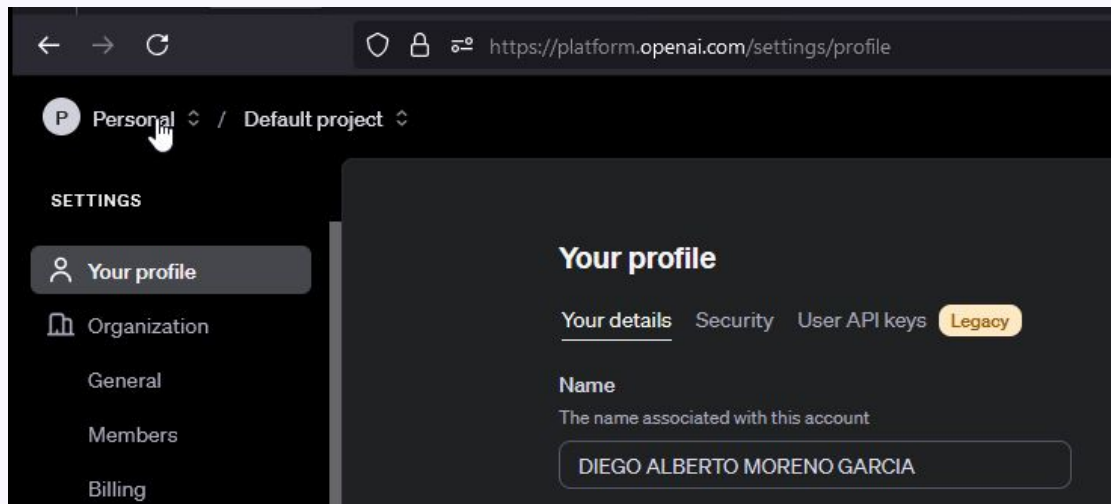




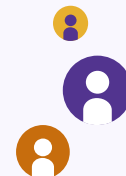
# LangChain RAG



## Cuenta Open Ai y Api Key



- **Auto recharge is off**  
When your credit balance reaches \$0, your API requests will stop working. Enable automatic recharge to automatically keep your credit balance topped up.  
[Enable auto recharge](#)



# LangChain RAG



**LangChain**, si lo traducimos podría ser “Cadena de lenguaje”

Es un framework para desarrollar aplicaciones que interactúan con modelos de lenguaje grandes (LLMs).

Es como un conjunto de bloques que permiten crear aplicaciones inteligentes que entiendan y generen lenguaje natural.

**Conecta los puntos:** Une diferentes componentes como modelos de lenguaje, bases de datos y otras herramientas para crear aplicaciones más completas.

**Es flexible:** Puedes personalizarla para adaptarla a tus necesidades específicas.

**Es eficiente:** Optimiza el uso de los modelos de lenguaje, lo que puede reducir costos.

**Es fácil de usar:** Incluso si no eres un experto en programación, puedes crear aplicaciones interesantes.



# LangChain RAG



## Componentes:

**Modelo IO** entradas y salidas de LLM, el cual es estandarizado lo que facilita cambiar entre LLM.

**Conectores de datos** conecta un modelo LLM a una fuente de datos, intercambia fácilmente almacenes de vectores, gran variedad de fuentes, CSV, PDF, AWS.

**Cadenas Chains** permite vincular la salida de un modelo para la entrada de otro. Encadena fácilmente diferentes llamadas de LLM para separar el trabajo.

**Memoria** retiene el contexto histórico de interacciones anteriores, permite guardar fácilmente conversaciones historicas.

**Agentes** crea agentes personalizados con pocas lineas de código.



# LangChain RAG



## LangChain, Base de datos vectorial.

Scikit-learn es una colección de código abierto de algoritmos de aprendizaje automático,  
que incluye algunas implementaciones de los k vecinos más cercanos .  
SKLearnVectorStore Envuelve esta implementación y agrega la posibilidad de conservar el almacén de vectores  
en formato json, bson (json binario) o Apache Parquet.

<https://python.langchain.com/docs/integrations/vectorstores/sklearn/>



# LangChain RAG



LangChain, Base de datos vectorial.

```
# Importar librerías
```

```
from langchain_openai import OpenAIEmbeddings  
from langchain.text_splitter import CharacterTextSplitter  
from langchain.document_loaders import TextLoader
```



# LangChain RAG



LangChain

LangChain, Base de datos vectorial.

```
# Carga de documento |
```

```
# Cargar el documento  
loader = TextLoader('C:/ProyectoChatOpenAI/RAG/Historia Colombia.txt', encoding="utf8")  
documents = loader.load()
```



# LangChain RAG



LangChain, Base de datos vectorial.

```
# Dividir en chunks
text_splitter = CharacterTextSplitter.from_tiktoken_encoder(chunk_size=500) #Otro método de split basándose en tokens
docs = text_splitter.split_documents(documents)
```

```
Created a chunk of size 528, which is longer than the specified 500
Created a chunk of size 579, which is longer than the specified 500
Created a chunk of size 508, which is longer than the specified 500
Created a chunk of size 525, which is longer than the specified 500
Created a chunk of size 511, which is longer than the specified 500
```



# LangChain RAG



LangChain, Base de datos vectorial.

```
# SKLearn Vector Store
```

[+ Código](#)[+ Markdown](#)

```
from langchain_community.vectorstores import SKLearnVectorStore #Se debe tener instalado pip install scikit-learn / pip install
```

```
persist_path="C:/ProyectoChatOpenAI/RAG/Vectorial_db" #ruta donde se guardará la BBDD vectorizada
```

```
#Creamos la BBDD de vectores a partir de los documentos y la función embeddings
```

```
vector_store = SKLearnVectorStore.from_documents(  
    documents=docs,  
    embedding=funcion_embedding,  
    persist_path=persist_path,  
    serializer="parquet", #el serializador o formato de la BD lo definimos como parquet  
)
```





# LangChain RAG



## LangChain, Base de datos vectorial.

```
#Creamos un nuevo documento que será nuestra "consulta" para buscar el de mayor similitud en nuestra Base de Datos de Vectores y devolverlo
consulta = "dame información de la Epoca precolombina"
docs = vector_store.similarity_search(consulta)
print(docs[0].page_content)
```

Época precolombina

Artículos principales: Colombia precolombina y Poblamiento de Colombia.

Ubicación de las culturas precolombinas de Colombia

Estatuas en un monumento sepulcral. La Cultura San Agustín tuvo dos períodos de desarrollo entre los Siglos X y IX a. C..15

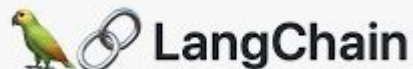
La Balsa muisca cuya creación ha sido estimada entre los años 600 y 1600 d. C.16

Ciudad Perdida construida hacia el 700 en la Sierra Nevada de Santa Marta. Hacia 1600 la ciudad quedó abandonada cayendo en el olvido.17

La teoría más aceptada expone que, siguiendo a los grandes animales a través del Istmo de Panamá, llegaron los primeros pobladores a América del Sur.18 Sin embargo, el des



# LangChain RAG



LangChain, Base de datos vectorial.

```
# Cargar la BD de vectores (uso posterior una vez tenemos creada ya la BD)
```

```
vector_store_connection = SKLearnVectorStore(  
    embedding=funcion_embedding, persist_path=persist_path, serializer="parquet"  
)  
print("Una instancia de la BBDD de vectores se ha cargado desde ", persist_path)
```

Una instancia de la BBDD de vectores se ha cargado desde [C:/ProyectoChatOpenAI/RAG/Vectorial\\_db](#)

+ Código

+ Markdown

```
vector_store_connection
```

```
<langchain_community.vectorstores.sklearn.SKLearnVectorStore at 0x14b5a6d3fe0>
```



# LangChain RAG



LangChain

## LangChain, trabajando con el RAG

```
#Importar librerías iniciales
from langchain.openai import ChatOpenAI
from langchain.prompts import PromptTemplate, SystemMessagePromptTemplate, ChatPromptTemplate, HumanMessagePromptTemplate
from langchain.agents import load_tools, initialize_agent, AgentType, create_react_agent, AgentExecutor
f = open('C:/ProyectoChatOpenAI/Scripts/Api_key.txt')
api_key = f.read()
llm = ChatOpenAI(openai_api_key=api_key, temperature=0) #Recomendable temperatura a 0 para que el LLM no alucine
```

```
#Cargamos la BD Vectorial y el compresor se establece memoria
from langchain.memory import ConversationBufferMemory
#Ponemos una clave a la memoria "chat_history"
memory = ConversationBufferMemory(memory_key="chat_history")
```

C:\Users\DMoreno\AppData\Local\Temp\ipykernel\_12920\2028269873.py:3: LangChainDeprecationWarning: Please see the migration guide at [https://python.langchain.com/docs/migration](#): `ConversationBufferMemory` is deprecated in favor of `ConversationBufferWindowMemory`.  
memory = ConversationBufferMemory(memory\_key="chat\_history") #ponemos una denominada clave a la memoria "chat\_history"



# LangChain RAG



## LangChain, trabajando con el RAG

```
from langchain_community.vectorstores import SKLearnVectorStore
from langchain_openai import OpenAIEmbeddings
from langchain.retrievers import ContextualCompressionRetriever
from langchain.retrievers.document_compressors import LLMChainExtractor

funcion_embedding = OpenAIEmbeddings(openai_api_key=api_key)
persist_path="C:/ProyectoChatOpenAI/RAG/Vectorial_db"
vector_store_connection = SKLearnVectorStore(embedding=funcion_embedding, persist_path=persist_path, serializer="parquet")
compressor = LLMChainExtractor.from_llm(llm)
compression_retriever = ContextualCompressionRetriever(base_compressor=compressor, base_retriever=vector_store_connection.as_retriever())
```

```
#Creamos una nueva herramienta a partir de la BD Vectorial para obtener resultados optimizados
from langchain.agents import tool
```

```
@tool
def consulta_interna(text: str) -> str:
    """Retorna respuestas sobre la historia de Colombia. Se espera que la entrada sea una cadena de texto
    y retorna una cadena con el resultado más relevante. Si la respuesta con esta herramienta es relevante,
    no debes usar ninguna herramienta más"""
    compressed_docs = compression_retriever.invoke(text)
    resultado = compressed_docs[0].page_content
    return resultado
```



# LangChain RAG



LangChain

## LangChain, trabajando con el RAG

```
tools = load_tools(["wikipedia","llm-math"],llm=llm)
```

```
tools=tools+[consulta_interna]
```

```
#Creamos el agente y lo ejecutamos
```

```
agent = initialize_agent(tools, llm, agent=AgentType.CONVERSATIONAL_REACT_DESCRIPTION,memory=memory,verbose=True)
```

```
agent.invoke("¿Qué·aso·en·Colombia·en·1821?")
```



# LangChain RAG



## LangChain, trabajando con el RAG

```
> Entering new AgentExecutor chain...  
Thought: Do I need to use a tool? Yes  
Action: consulta_interna  
Action Input: ¿Qué aso en Colombia en 1821?  
Observation: Independencia de Colombia  
En 1819 un ejército republicano comandado por Simón Bolívar cruzó las montañas que separan Casanare de Tunja y Santa Fe, Tras La Batalla del Pantano de Vargas y La Batalla  
Thought: Do I need to use a tool? No  
AI: En 1821, en Colombia, se proclamó La República de Colombia, que ya había sido firmada en el Congreso de Angostura en febrero del mismo año. Esto condujo a la creación
```

```
> Finished chain.
```

```
{  
  'input': '¿Qué aso en Colombia en 1821?',  
  'chat_history': 'Human: ¿Qué periodo abarca cronológicamente en Colombia la época precolombina?\nAI: La época precolombina en Colombia abarca desde la llegada de los prime  
  'output': 'En 1821, en Colombia, se proclamó la República de Colombia, que ya había sido firmada en el Congreso de Angostura en febrero del mismo año. Esto condujo a la ci
```



# LangChain RAG



LangChain

## LangChain, trabajando con el RAG

```
agent.invoke("¿El congreso que dispuso durante este periodo?") #Gracias a tener memoria compara en la pregunta anterior
```

> Entering new AgentExecutor chain...

*Thought: Do I need to use a tool? Yes*

*Action: consulta\_interna*

*Action Input: El congreso que dispuso durante este periodo*

*Observation: El congreso designó como presidente a Ramón González Valencia.*

*Thought: Do I need to use a tool? No*

*AI: Durante este periodo, el congreso designó como presidente a Ramón González Valencia.*

> Finished chain.

```
{'input': '¿El congreso que dispuso durante este periodo?',
```

```
 'chat_history': 'Human: ¿Qué periodo abarca cronológicamente en Colombia la época precolombina?\nAI: La época precolombina en Colombia abarca desde la llega
```

```
 'output': 'Durante este periodo, el congreso designó como presidente a Ramón González Valencia.'}
```

◀





# LangChain RAG



LangChain

## LangChain, trabajando con el RAG

```
¿Cuáles son las marcas de vehículos más famosas hoy en día?agent.invoke("") #Pregunta que no podemos responder con nuestra BD Vectorial
```

> Entering new AgentExecutor chain...

Thought: Do I need to use a tool? Yes

Action: wikipedia

Action Input: Most famous vehicle brands today

Observation: Page: Chrysler (brand)

Summary: Chrysler is an American brand of automobiles and division owned by Stellantis North America. The automaker was founded in 1925 by Walter Chrysler from the remains of the Maxwell Motor Company. The brand has been historically popular. However starting in the Late 2010s, the brand has been overshadowed by other brands owned by Stellantis yet continues to have a large loyalty following.

Page: Matchbox (brand)

Summary: Matchbox is a toy brand which was introduced by Lesney Products in 1953, and is now owned by Mattel, Inc, which purchased the brand in 1997. The brand was given its name because the company's first product was a matchbox. During the 1980s, Matchbox began to switch to the more conventional plastic and cardboard "blister packs" that were used by other die-cast toy brands such as Hot Wheels. By the 2000s, the company's products currently marketed under the Matchbox name include scale model plastic and die-cast vehicles, and toy garages.

Page: Brand

Summary: A brand is a name, term, design, symbol or any other feature that distinguishes one seller's good or service from those of other sellers. Brands are used in business, marketing, and advertising. The practice of branding—in the original literal sense of marking by burning—is thought to have begun with the ancient Egyptians, who are known to have engaged in livestock branding and branding of tools. In the modern era, the concept of branding has expanded to include deployment by a manager of the marketing and communication techniques and tools that help to distinguish a company or product. Brand equity is the measure of a brand's value.

Thought: Do I need to use a tool? No

AI: Some of the most famous vehicle brands today include Toyota, Ford, Chevrolet, Honda, Volkswagen, BMW, Mercedes-Benz, and Tesla, among others. These brands are well-known for their quality and reliability.



# LangChain RAG



**Streamlit**, es una herramienta de Python de código abierto que te permite construir aplicaciones web interactivas y basadas en datos de una manera rápida y sencilla.

**Sencillez:** Con una sintaxis similar a Python, es fácil de aprender y utilizar, incluso si no tienes experiencia en desarrollo web.

**Velocidad:** Puedes crear prototipos y aplicaciones funcionales en cuestión de minutos.

**Interactividad:** Permite crear aplicaciones con interfaces de usuario intuitivas, incluyendo gráficos, sliders, mapas y más.

**Integración con otras bibliotecas:** Funciona muy bien con otras bibliotecas de Python populares como NumPy, Pandas, Matplotlib, y frameworks de machine learning como Scikit-learn y TensorFlow.

**Compartir:** Puedes compartir tus aplicaciones fácilmente con otros, ya sea de forma local o desplegándose en la nube.



# LangChain RAG



LangChain

LangChain, Agente

## Asistente de Historia de Colombia

Pregunta sobre la historia de Colombia y obtén respuestas de la base de datos.

Introduce tu pregunta:

¿Qué paso en Colombia en 1821?

Respuesta:

```
{
  "input" : "¿Qué paso en Colombia en 1821?"
  "chat_history" : ""
  "output" :
    "En 1821, en Colombia se estableció la Constitución de 1821, se llevó a cabo el Congreso Constituyente de 1821 y se promulgó la Ley de Libertad de Vientres."
}
```



# LangChain RAG



## LangChain, Agente

### Asistente de Historia de Colombia ↔

Pregunta sobre la historia de Colombia y obtén respuestas de la base de datos.

Introduce tu pregunta:

¿El congreso que dispuso durante este periodo ?

Respuesta:

```
{
  "input" : "¿El congreso que dispuso durante este periodo ?"
  "chat_history" : ""
  "output" :
    "El congreso que dispuso durante este periodo en Colombia fue el Congreso de
    Cúcuta."
}
```



# LangChain RAG



LangChain

## LangChain, Agente

### Asistente de Historia de Colombia

Pregunta sobre la historia de Colombia y obtén respuestas de la base de datos.

Introduce tu pregunta:

¿Cuáles son las marcas de vehículos más famosas hoy en día?

Respuesta:

```
{  
  "input" : "¿Cuáles son las marcas de vehículos más famosas hoy en día?"  
  "chat_history" : ""  
  "output" :  
    "The most famous vehicle brands today include Toyota, Volkswagen, and  
    Chrysler."  
}
```



# LlamaIndex RAG



## Chat with the Streamlit docs



Ask me a question about de energía comprimida



de que trata el documento



El documento parece ser un informe o reporte sobre la realización del proyecto "Aire comprimido".

El documento parece ser un informe o reporte sobre la realización del proyecto "Aire comprimido".




ok, cual es la deficion de energia comprimida segun el documento





# LlamaIndex RAG





Chat with the Streamlit docs

 Ask me a question about de energía comprimida

 de que trata el documento

 El documento parece ser un informe o reporte sobre la realización del proyecto "Aire comprimido".

 ok, cual es la deficion de energia comprimida segun el documento

 La definición de energía comprimida no está explícitamente mencionada en el texto proporcionado. Sin embargo, se puede inferir que se refiere a la procesamiento o compresión del aire para producir un gas adicional para la generación de energía, como se describe en el capítulo 5 sobre viabilidad económica.



# Conclusiones

- La integración de RAG ha sido un punto diferencial. Al permitir que el modelo acceda a una base de conocimiento externa, hemos mejorado significativamente la precisión y la relevancia de las respuestas. RAG garantiza que el chatbot pueda proporcionar información actualizada y específica, superando las limitaciones de los modelos entrenados en datos estáticos.



# Artículos consultados

Rodríguez Parrales, M. S. (2024). Desarrollo de una aplicación web para la gestión bibliotecaria de un Instituto Tecnológico aplicando un asistente virtual Chatbots con reconocimiento de voz (Bachelor's thesis, La Libertad: Universidad Estatal Península de Santa Elena, 2024.).

<https://repositorio.upse.edu.ec/bitstream/46000/10930/1/UPSE-TTI-2024-0013.pdf>

Lalwani, Tarun and Bhalotia, Shashank and Pal, Ashish and Rathod, Vasundhara and Bisen, Shreya, Implementation of a Chatbot System using AI and NLP (May 31, 2018). International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3, May-2018, Available at SSRN: <https://ssrn.com/abstract=3531782> or <http://dx.doi.org/10.2139/ssrn.3531782>.

Huaman Hilari, J. Z., & Quispe Ramos, M. A. (2019). Modelo de búsqueda de productos alimenticios en supermercados online categoría abarrotes utilizando asistente virtual de tipo chatbot y extracción de datos con web scraping.

Expósito García, Y. (2019). Bullbot: asistente virtual para la gestión de servicios de la Biblioteca de la ULL

[https://python.langchain.com/v0.1/docs/use\\_cases/sql/agents/](https://python.langchain.com/v0.1/docs/use_cases/sql/agents/)

<https://streamlit.io/>





# Thanks

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), and infographics & images by [Freepik](#)

Please keep this slide for attribution

