

Ejercicio #3 — LangChain

Definición de LangChain y su importancia

LangChain es un framework de código abierto diseñado para desarrollar aplicaciones que utilizan modelos de lenguaje (LLMs), como GPT, de forma estructurada e integrada con datos externos. Su objetivo es conectar modelos de IA con fuentes de datos, APIs y herramientas externas para crear sistemas más útiles y contextuales.

Importancia:

- Facilita la creación de chatbots, asistentes y sistemas de IA personalizados.
- Permite conectar el LLM con bases de datos, documentos, APIs y memoria.
- Ahorra tiempo de desarrollo gracias a su arquitectura modular.
- Es ampliamente utilizado en la industria de IA generativa.

Componentes principales de LangChain

1. LLMs (Modelos de Lenguaje): Es el motor principal. Puede ser GPT, Claude, Gemini, etc.
2. Prompts (Indicaciones): Son las instrucciones o plantillas que se le dan al modelo para guiar sus respuestas.
3. Chains (Cadenas): Secuencias de pasos que combinan prompts, modelos y salidas para crear flujos de trabajo complejos.
4. Memory (Memoria): Permite que el sistema recuerde información previa.
5. Agents (Agentes): Son componentes inteligentes que pueden decidir qué acción tomar.
6. Tools (Herramientas): Conectan el modelo con fuentes externas como bases de datos o internet.

Casos de uso reales o ejemplos de aplicación

1. Chatbots personalizados para empresas: Asistentes que responden preguntas con datos internos.
2. Análisis de documentos legales o financieros: Busca, resume y explica información compleja.
3. Asistentes educativos o de programación: Tutores que ayudan a escribir código o explicar temas.

Ejemplo de código con explicación

```
from langchain.chat_models import ChatOpenAI from langchain.chains import LLMChain from langchain.prompts import PromptTemplate # Definimos el modelo de lenguaje (GPT-3.5 o GPT-4) llm = ChatOpenAI(temperature=0.7) # Creamos una plantilla de prompt plantilla = PromptTemplate( input_variables=["tema"], template="Explica de forma sencilla qué es {tema} y da un ejemplo." ) # Creamos una cadena (Chain) cadena = LLMChain(llm=llm, prompt=plantilla) # Ejecutamos la cadena con una entrada respuesta = cadena.run("la inteligencia artificial") print(respuesta)
```

Explicación: Se importa LangChain y se define un modelo de lenguaje. Luego se crea una plantilla de prompt con una variable (tema). La cadena conecta el modelo con el prompt, y al

ejecutar, el modelo devuelve una explicación sobre el tema.