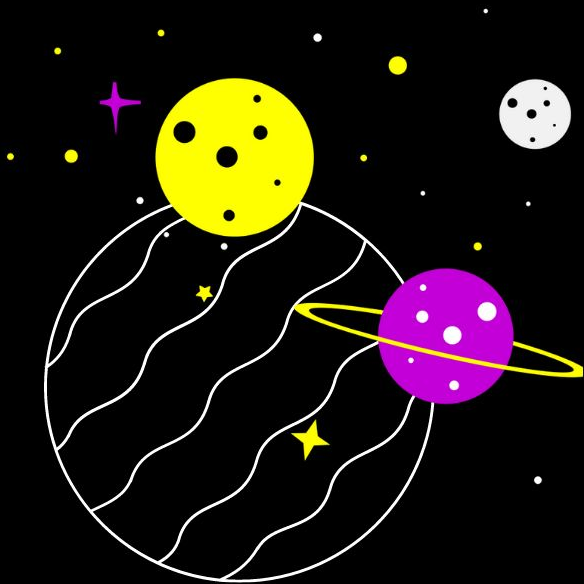


**HENRY**



# **Orquestación de Agentes : Método React**



# Agenda del día



- **Estructura de una clase**
  - Bibliografía
  - Repositorio
  - Definiciones preliminares
  - Agentes vs Chatbots
  - Anatomía de un workflow agéntico
  - Demo
- **Script de clase**



**HENRY**



# **Bibliografía**



[Langchain](#)

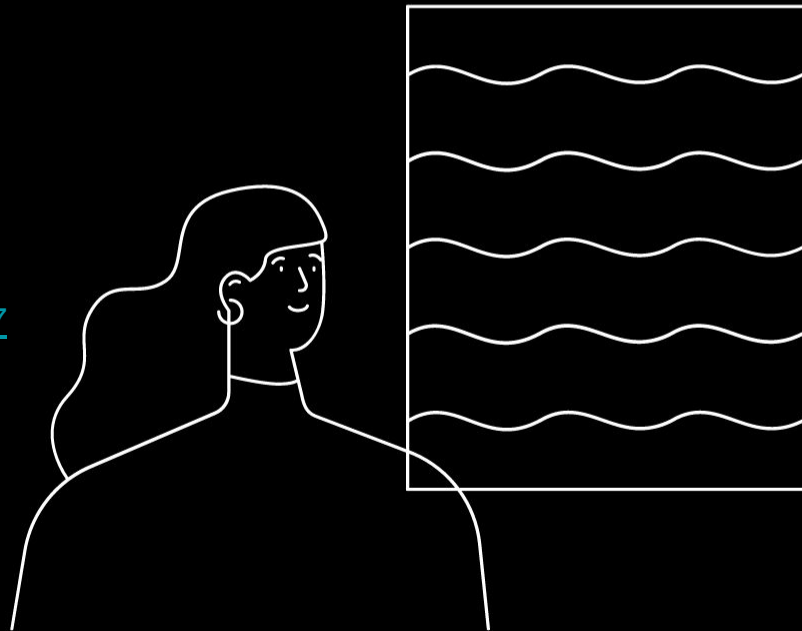
[Langgraph](#)

<https://newsletter.languagemodels.co/>

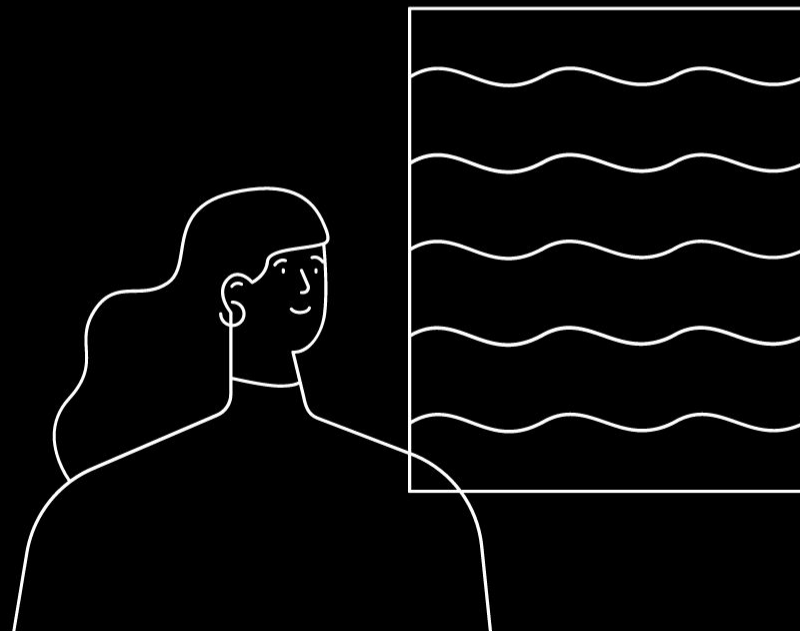
[OpenAI Models](#)

[Generative AI with LangChain](#)

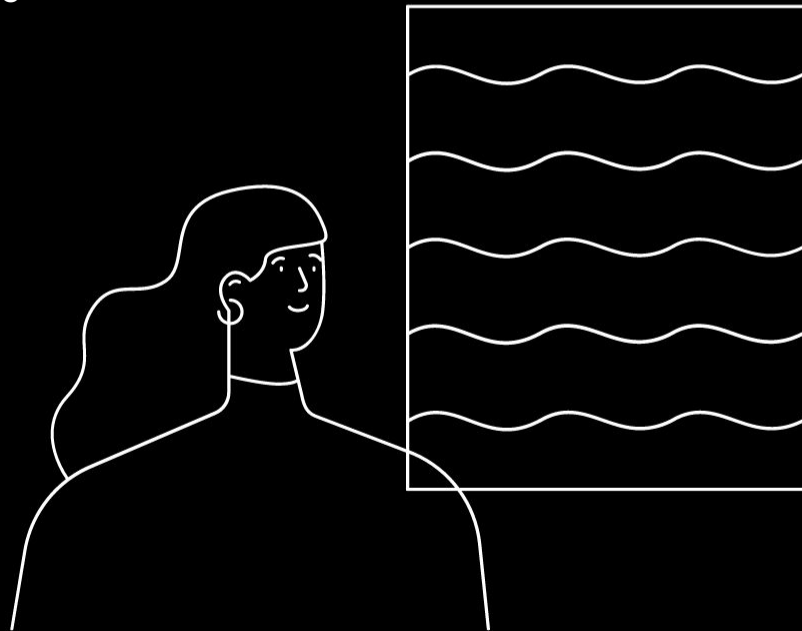
<http://oreilly.com/library/view/ai-engineering/9781098166298/>



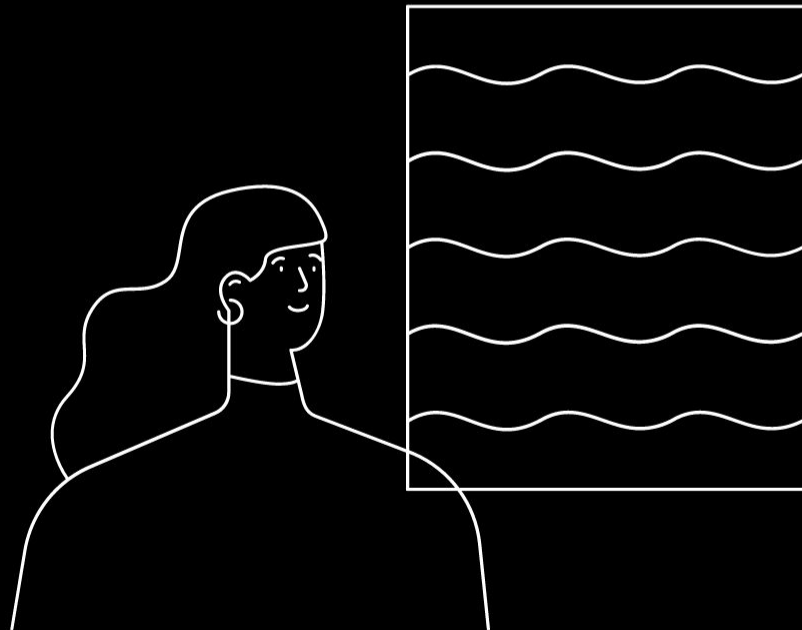
[Repo](#)



Anotación No. 1 : En esta clase no se hablara de Imágenes o AI enfocada a las imágenes.



Anotación No. 2 : Esta no es una presentación hecha con ChatGPT -> Pero saben qué es un GPT?



**HENRY**



**Definizione**

**S**

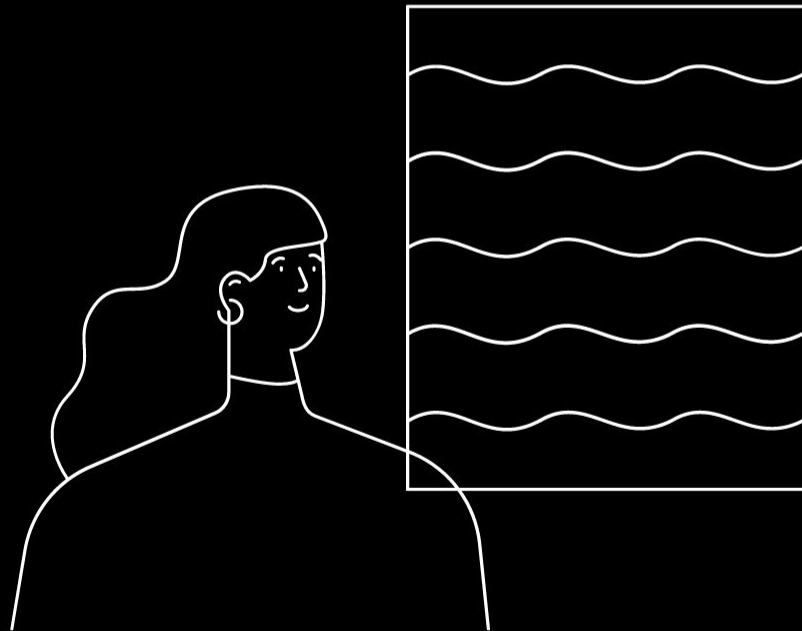
**Preliminare**

**S**

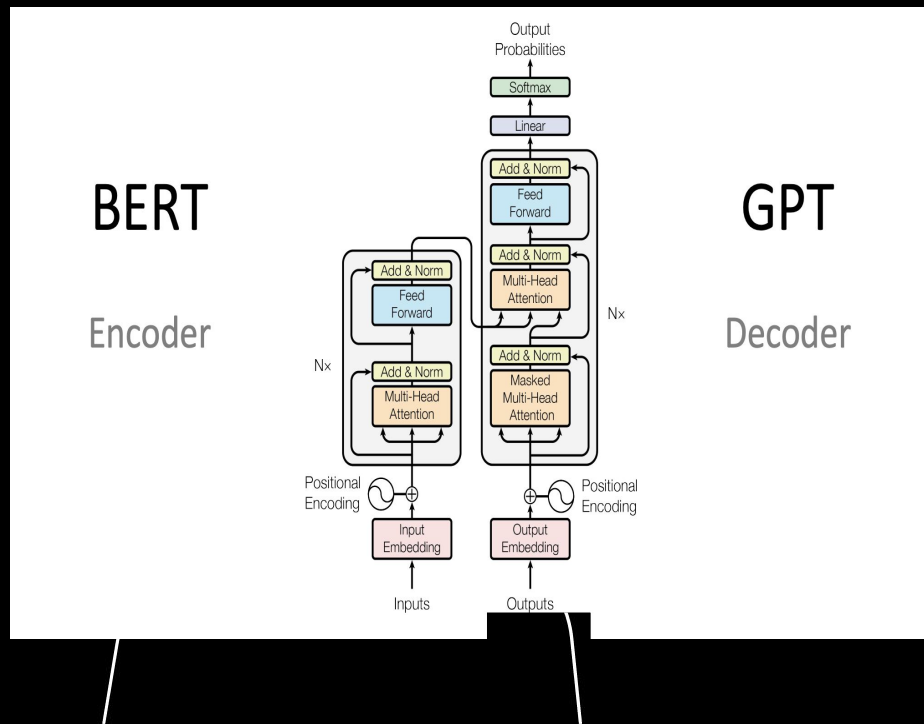




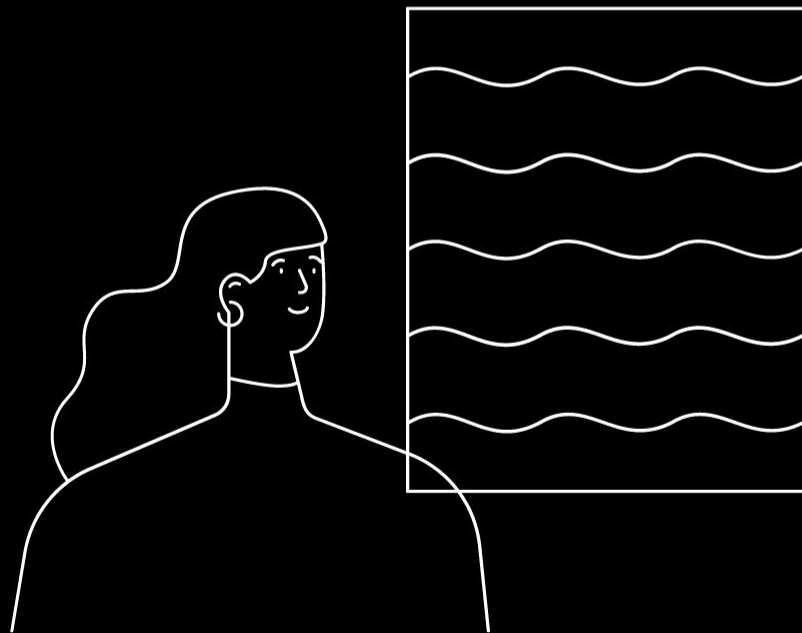
# **Generative Pre-training Transformers**



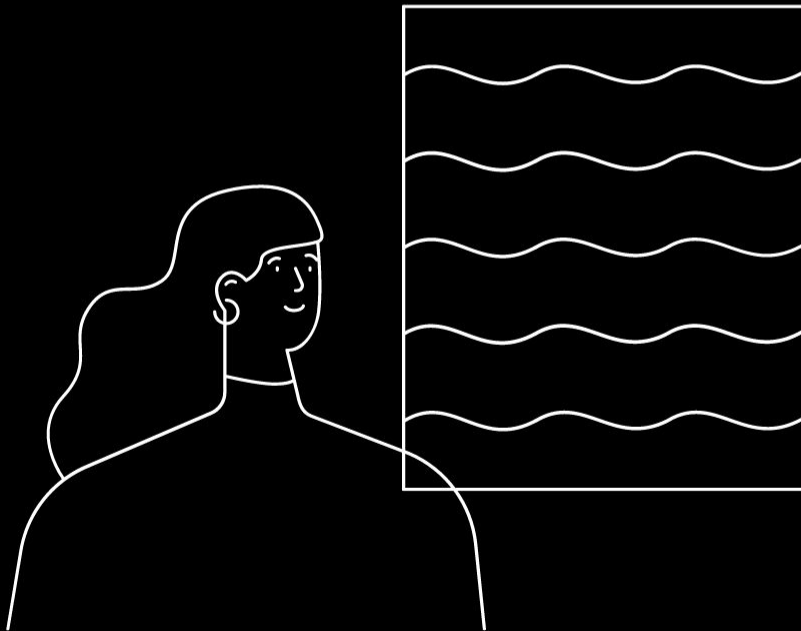
# Generative Pre-training Transformers



1. **No es un buscador, es un creador:** A diferencia de Google, que te da una lista de enlaces, GPT construye respuestas nuevas combinando ideas de forma coherente.
2. **Aprende patrones, no datos:** Imagina que es un experto en "adivinar" cuál es la siguiente palabra más lógica en una oración basándose en el contexto que le das.
3. **Memoria de contexto:** Tiene un contexto gracias a la lógica matemática y de almacenamiento que hay entre sus características
4. **Versatilidad total:** Tiene la capacidad de hacer todo lo que por palabras se pueda expresar y representar.

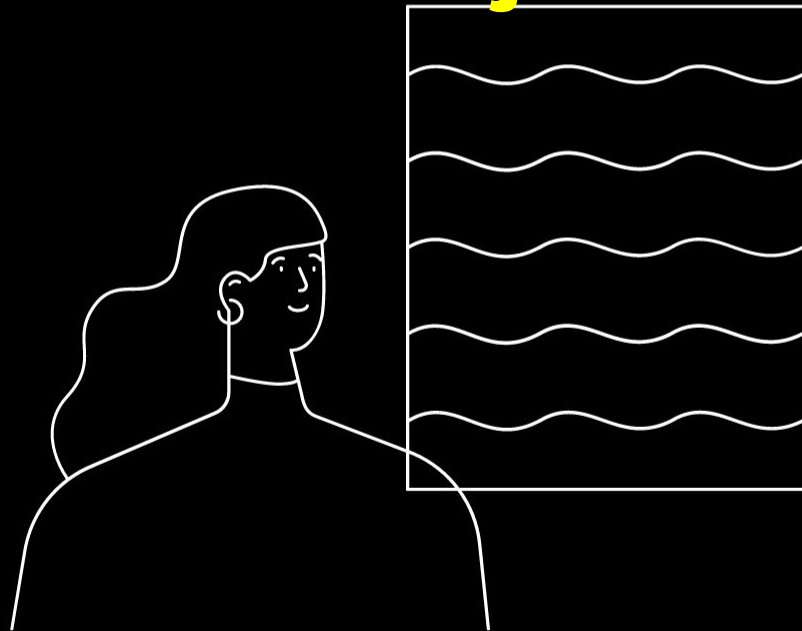


1. **Recuerde lo siguiente:** Una unidad de medida a la hora de trabajar con textos son los **tokens**.
2. **Masked Language Models:** Predice en una secuencia de texto el **Token** faltante , ejemplo : El libro La — levedad del ser , de Kundera -> El libro La **Insoportable** levedad del ser , de Kundera.
3. **Autoregressive Language Models:** Predice el **token** consecuente, ejemplo: Mi película favorita es — | Mi película favorita es **El secreto de sus ojos**.



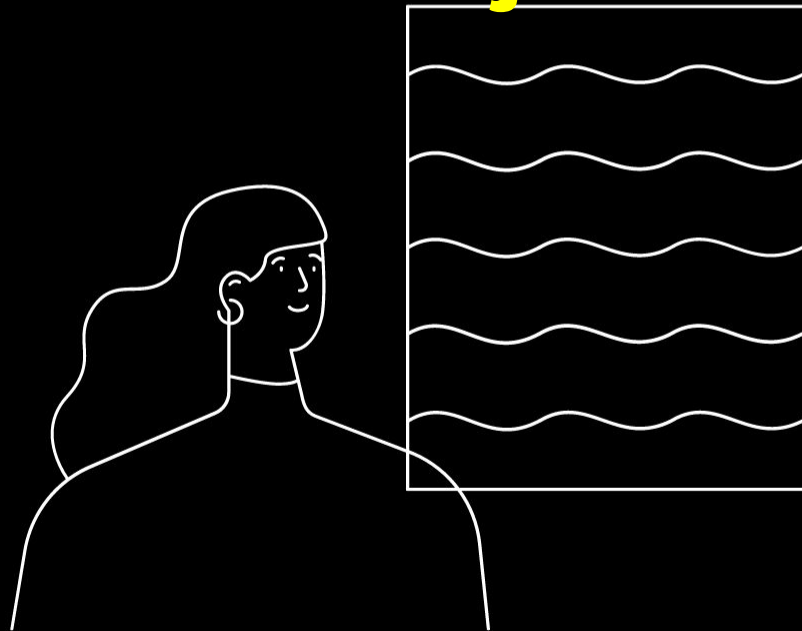
## En cuál lado de la ciencia estaremos trabajando?

1. **AI** = Creación de agentes que piensan, actúan y se pueden automatizar
  - a. Los agentes son modelos de lenguaje que pueden tener acciones , más que responder una pregunta.
2. **ML** = Algoritmos que aprenden patrones de los datos ( generalmente tabulades)
3. **Deep Learning** = Algoritmos de machine learning que aprenden patrones de datos complejos o no estructurados como textos , imágenes o sonidos.
4. **GenAI** = Genera nuevos datos a partir de inputs o patrones pasados para así poder construir un contenido.
5. **LM** = Modelos estadísticos destinados a entender el comportamiento y el orden de las palabras dentro de un contexto
6. **LLM** = Lo anterior pero con + datos y llamados apificados



## En cuál lado de la <sup>02</sup> ciencia estaremos trabajando?

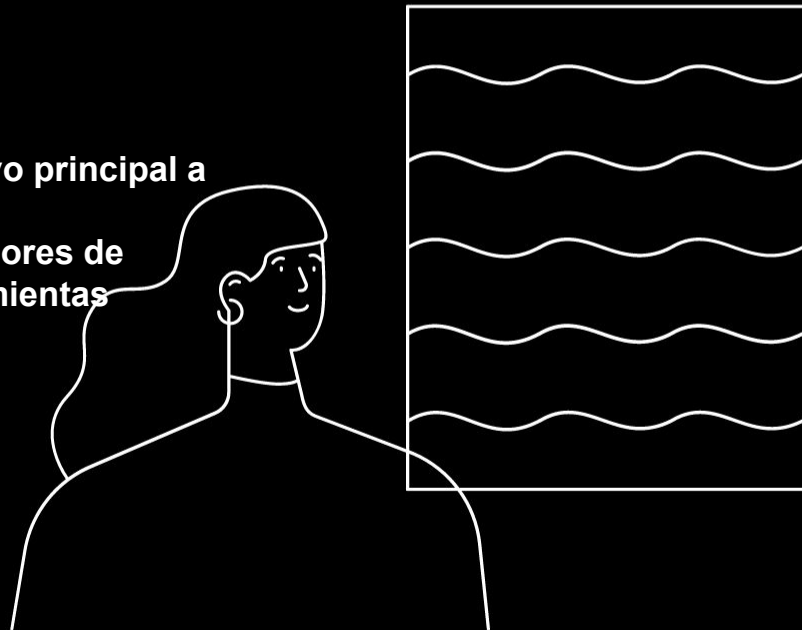
1. AI = Creación de agentes que piensan, actúan y se pueden automatizar
  - a. Los agentes son modelos de lenguaje que pueden tener acciones, más que responder una pregunta.



- Es un framework de Python
- Permite tener arquitectura modular
- Acceder a servicios
- Y está orientado a objetivos claros

- Agentes y memorias son el objetivo principal a trabajar
- Vea a los agentes como orquestadores de proceso los LLMS + uso de herramientas externas

- El core de Langchain es : **Dividir una tarea de manera** tal que un modelo de lenguaje pueda resolver uno a uno los substeps o task y agregar estos resultados de manera tal que el margen de error en el output sea menor.



```
from langchain_openai import ChatOpenAI

llm = ChatOpenAI(
    model=config['model_llm']['name'],
    temperature=config['model_llm']['temperature'],
    max_tokens=config['model_llm']['max_tokens']
)
answer = llm.invoke("¿Cuál es la capital de Colombia?")
print(answer.content)
# Bogotá
```

```
model_llm :
  name: "gpt-4"
  temperature: 0.01
  max_tokens: 1000
```



**HENRY**



# **Agentes Vs Chatbots**



## CHATBOT: El Conversador Reactivo



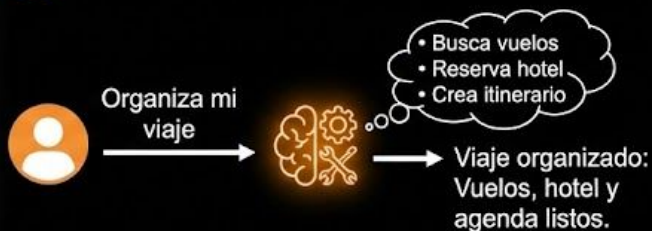
- ✓ Responde preguntas específicas.
- ✓ Sigue un guion predefinido.
- ✓ Espera tu próxima instrucción.



## AGENTE IA: El Solucionador Proactivo



- ✓ Entiende objetivos y contexto.
- ✓ Planifica y ejecuta múltiples pasos.
- ✓ Actúa y aprende autónomamente.



La diferencia clave: Los chatbots *\*hablan\**, los agentes *\*hacen\**.

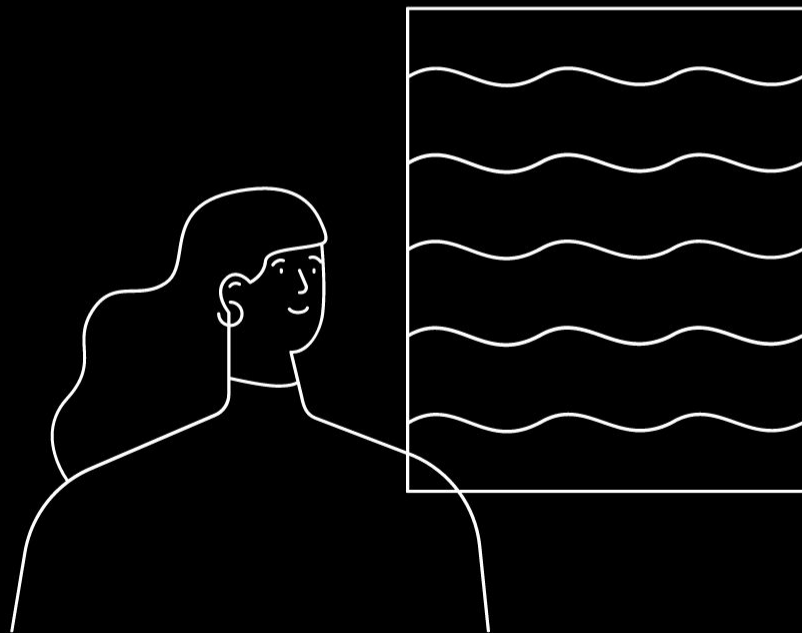
### Anatomía de un modelo

Nombre del  
modelo |  
Proveedor

```
llm = ChatOpenAI(  
    model=config['model_llm']['name'],  
    temperature=config['model_llm']['temperature'],  
    max_tokens=config['model_llm']['max_tokens']  
)
```

Qué tanto  
puede gastar  
el modelo ?

Qué tan  
creativo es el  
modelo ?



## Anatomia de un modelo

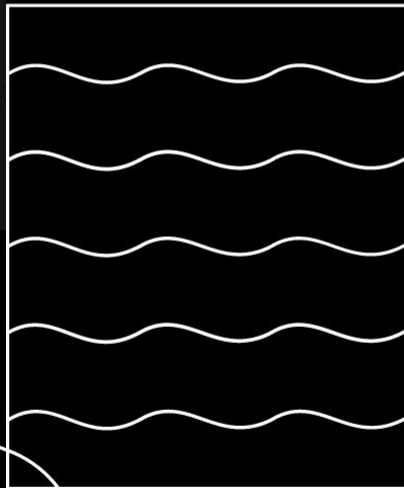
```
from langchain_core.messages import HumanMessage, SystemMessage, AIMessage
chat_output = llm.invoke([
    SystemMessage(content="Eres un asistente útil para programación en python."),
    HumanMessage(content="haz un hello world en python con typing"),
])
print(chat_output.content)
```

```
from typing import Any

def hello_world(name: str) -> str:
    return f"Hello World, {name}"

def main() -> Any:
    print(hello_world("Python"))

if __name__ == "__main__":
    main()
```



```
from langchain_core.tools import tool
from typing import List
import requests

@tool("get_weather", description="Get the weather of a city")

def get_weather(city: str):
    response = requests.get(f"https://geocoding-api.open-meteo.com/v1/search?name={city}&count=1")
    data = response.json()
    latitude = data["results"][0]["latitude"]
    longitude = data["results"][0]["longitude"]
    response = requests.get(f"https://api.open-meteo.com/v1/forecast?latitude={latitude}&longitude={longitude}&current_weather=true")
    data = response.json()
    response = f"The weather in {city} is {data["current_weather"]["temperature"]}C with {data["current_weather"]["windspeed"]}km/h of wind."
    return response

get_weather.invoke({"city": "Barranquilla"})
#The weather in Barranquilla is 28.2C with 24.7km/h of win
```

```
system_prompt = """
Eres un experto en clima y que actividades puede hacer una familia de padres y una hija dada la
temperatura del momento:
```

```
### Rol
Definir si es buen clima, muy frio o calido o caliente para hacer alguna actividad
recomenda la actividad
```

```
### Tools
- get_weather: Obtiene el clima actual de una ciudad dada
"""
```

```
messages = [
    ('system', system_prompt),
    ('user', "Que actividades puedo hacer en familia en barranquilla hoy?")
]
llm_with_tools = llm.bind_tools([get_weather])
response = llm_with_tools.invoke(messages)
```

```
f response.tool_calls:
    # Agregar la respuesta del modelo (con el tool_call)
    messages.append(response)

    # Ejecutar cada tool call
    for tool_call in response.tool_calls:
        # Ejecutar la herramienta
        tool_result = get_weather.invoke(tool_call["args"])

        # Agregar el resultado como ToolMessage
        messages.append(ToolMessage(
            content=tool_result,
            tool_call_id=tool_call["id"]
        ))
```

```
# Llamar al modelo de nuevo con el resultado
final_response = llm_with_tools.invoke(messages)
print(final_response.content)
else:
    print(response.content)
```

iHola! Acabo de consultar el clima en Barranquilla y aquí está mi análisis:

```
## 🌤️ Clima Actual en Barranquilla
- **Temperatura:** 28.1°C (Clima cálido/caliente)
- **Viento:** 25.2 km/h (brisa moderada)
```

```
## 🌤️ Evaluación del Clima
El clima está cálido, típico de Barranquilla. Es un día agradable pero hay que tomar precauciones con
el sol y mantenerse hidratados.
```

```
## 🚶‍♀️ Actividades Recomendadas para la Familia
```

```
### Actividades al Aire Libre (preferiblemente temprano en la mañana o al atardecer):
1. Visitar el Malecón del Río - Paseo familiar con brisa del río Magdalena
2. Ir a la playa (Puerto Colombia o Salgar) - Perfecta para refrescarse con este clima
3. Paseo en bicicleta por la Vía 40 - Aprovechando la brisa
```

```
### Actividades para Evitar el Calor:
1. Centro Comercial Buenavista o Viva - Aire acondicionado, cine, juegos para niños
2. Museo del Caribe - Educativo y climatizado, ideal para toda la familia
3. Visitar el Zoológico de Barranquilla - Ir temprano (antes de las 10am) para evitar el calor fuerte
```

```
### Actividades Acuáticas (¡Ideales para este clima!):
1. Piscina en un club o hotel
2. Parque acuático
```

```
## ⚠️ Recomendaciones:
- Llevar protector solar
- Mantenerse hidratados (mucha agua)
- Usar ropa ligera y sombreros
- Evitar actividades intensas al aire libre entre 11am y 3pm
```

¿Te gustaría más detalles sobre alguna de estas actividades? 🙋

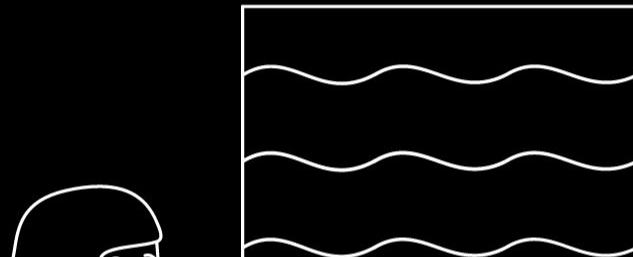
**HENRY**



# **Anatomía de un workflow**



- El orquestador de los agentes es langgraph , la cual ayuda a razonar hasta llegar a una respuesta correcta



Arquitectura	¿Qué es?	Decisión: Úsala si...
Router	Un clasificador de entrada.	El camino es 100% predecible según el tema.
ReAct (Ciclo)	Un agente con herramientas.	Necesitas que la IA "piense" y use herramientas por sí misma.
Paralelismo	Tareas simultáneas.	El tiempo de respuesta importa y las tareas no dependen entre sí.
Multi-Agente	Varios ReAct colaborando.	Una sola "mente" de IA se pierde con tanta complejidad o herramientas.



```
from typing import Annotated, TypedDict, Sequence
from langchain_core.messages import BaseMessage, HumanMessage, AIMessage, ToolMessage
from langgraph.graph import StateGraph, END
from langgraph.prebuilt import ToolNode
from langgraph.graph.message import add_messages

# Definir el State
class AgentState(TypedDict):
    messages: Annotated[Sequence[BaseMessage], add_messages]

# Crear el model con las tools
llm_with_tools = llm.bind_tools([get_weather])

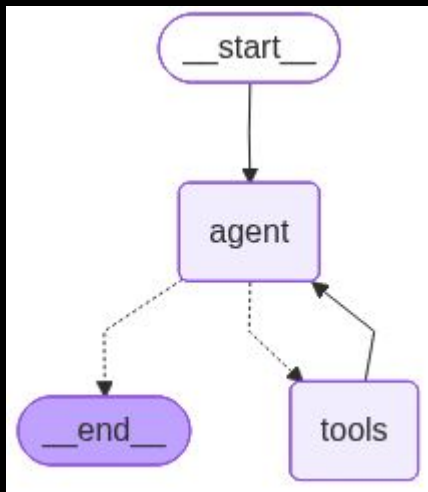
# Definir los nodos
def call_model(state: AgentState):
    """Nodo que llama al modelo con las herramientas"""
    messages = state["messages"]
    response = llm_with_tools.invoke(messages)
    return {"messages": [response]}

def should_continue(state: AgentState):
    """Decide si continuar ejecutando tools o terminar"""
    messages = state["messages"]
    last_message = messages[-1]

    # Si el modelo hace tool calls, continuar
    if hasattr(last_message, 'tool_calls') and last_message.tool_calls:
        return "tools"
    # Si no, terminar
    return END
```

```
# Visualizar el grafo
from IPython.display import Image, display

try:
    display(Image(graph.get_graph().draw_mermaid_png()))
except Exception as e:
    print(f"No se pudo generar la imagen: {e}")
    print("\nEstructura del grafo:")
    print(graph.get_graph().draw_ascii())
```



```
initial_state = {
    "messages": [
        ("system", ""Eres un experto en clima y que
actividades puede hacer una familia de padres y una hija
dada la temperatura del momento:
```

```
### Rol
```

```
Definir si es buen clima, muy frio o calido o caliente
para hacer alguna actividad
recomenda la actividad
```

```
### Tools
```

```
- get_weather: Obtiene el clima actual de una ciudad
dada
```

```
""),
```

```
        ("user", "Que actividades puedo hacer en
familia en barranquilla hoy?")
```

```
    ]
```

```
}
```

```
result = graph.invoke(initial_state)
print(result["messages"][-1].content)
```

```
iHola! El clima actual en **Barranquilla** es de
**26.4°C** con vientos de 31.4 km/h.

### 🌤 Evaluación del clima:
Es un clima **cálido y agradable**, típico de
Barranquilla. La temperatura es perfecta para
actividades al aire libre, aunque los vientos moderados
pueden hacer que se sienta un poco más fresco, lo cual
es ideal para esta época.

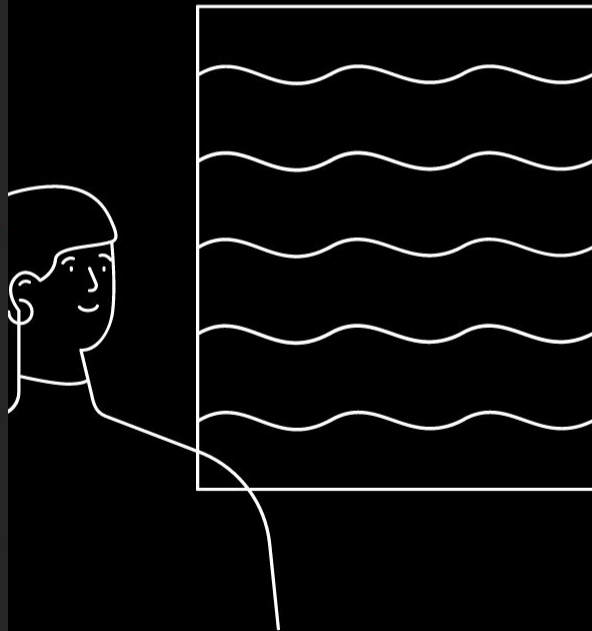
### 🏠 Actividades recomendadas para la familia:

**Al aire libre:**
1. **Visitar el Malecón del Río** - Pasear por la
orilla del río Magdalena, disfrutar de los espacios
verdes y la brisa. ¡El viento hará el paseo muy
refrescante!
2. **Ir a la playa** - Puerto Colombia o Salgar están
cerca. Con este clima cálido, un día de playa sería
perfecto.
3. **Parque Venezuela o Parque Suri Salcedo** - Ideales
para un picnic familiar, jugar con su hija y disfrutar
del aire libre.
4. **Zoológico de Barranquilla** - Excelente opción
para que la niña conozca animales y disfrute en
familia.

**En interiores (si prefieren evitar el sol fuerte):**
1. **Centro Comercial Buenavista o Portal del Prado** -
Cine, juegos para niños y restaurantes.
2. **Museo del Caribe** - Interactivo y educativo,
perfecto para toda la familia.

### 💡 Consejos:
- Lleven agua y protector solar si salen al aire libre
- Aprovechen las horas de la mañana o después de las
4pm para evitar el sol más intenso
- El viento puede ser útil para actividades como volar
cometas con su hija

¿Les gustaría más detalles sobre alguna de estas
actividades? 🌿
```



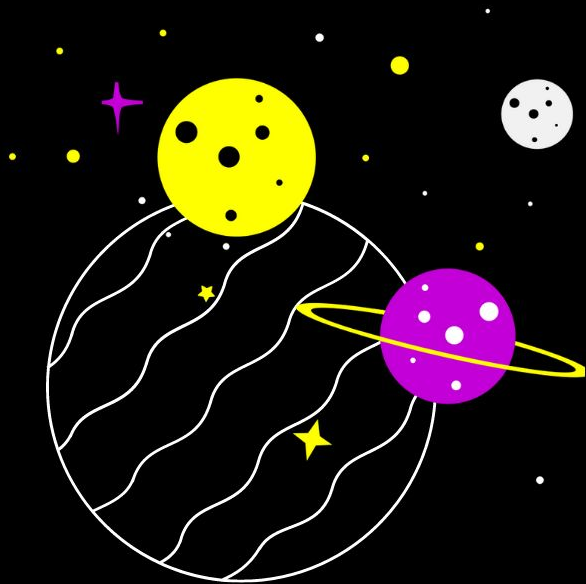
**HENRY**



**Demo**



**HENRY**



**¡Muchas gracias!**