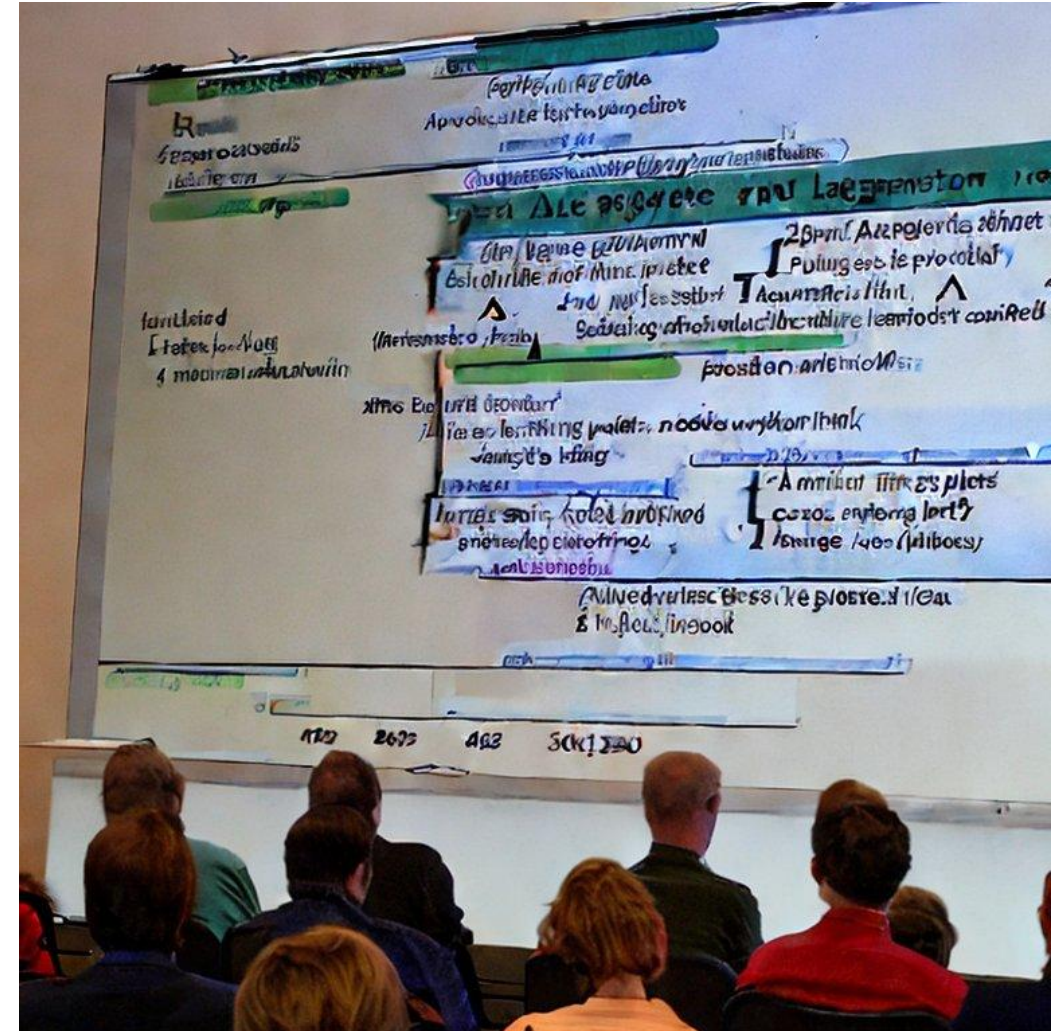


# LLMs e IA Generativa

## Clase 7 – Agentes y Fine Tuning

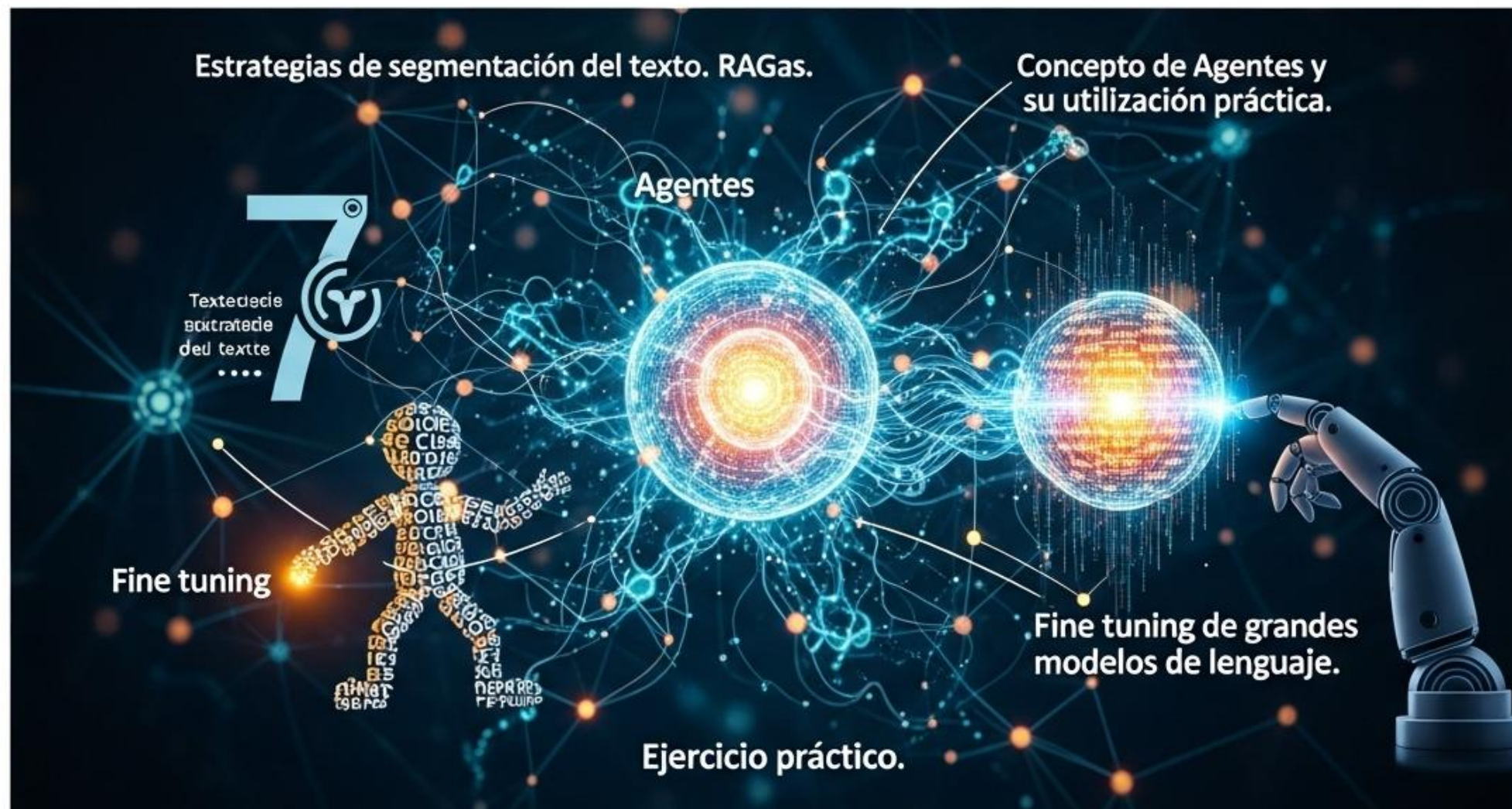
Mg. Ing. Ezequiel Guinsburg  
ezequiel.guinsburg@gmail.com





Universidad de  
Buenos Aires

## Procesamiento de Lenguaje Natural II



# Temas:

- Pendientes de clase 7: Estrategias de segmentación del texto. RAGas.
- Concepto de Agentes y su utilización práctica.
- Fine tuning de grandes modelos de lenguaje.
- Ejercicio práctico.



## Referencias:

- [1] Wei et al. [“Chain of thought prompting elicits reasoning in large language models.”](#) NeurIPS 2022
- [2] Yao et al. [“Tree of Thoughts: Deliberate Problem Solving with Large Language Models.”](#) arXiv preprint arXiv:2305.10601 (2023).
- [3] Liu et al. [“Chain of Hindsight Aligns Language Models with Feedback “](#) arXiv preprint arXiv:2302.02676 (2023).
- [4] Liu et al. [“LLM+P: Empowering Large Language Models with Optimal Planning Proficiency”](#) arXiv preprint arXiv:2304.11477 (2023).
- [5] Yao et al. [“ReAct: Synergizing reasoning and acting in language models.”](#) ICLR 2023. (Google Research)
- [6] Ejemplo completo de aplicación de agentes: <https://github.com/Significant-Gravitas/AutoGPT>

# Segmentación de texto pre-embeddings:

Distintas estrategias:

- División en párrafos u oraciones (depende máximo de Tokens que acepte el modelo de embeddings (ej.: 512 K, 2048 K, etc.).
- Segmentación por Chunking, largo fijo.
- Segmentación por ventanas deslizantes por contexto semántico.
- Segmentación por contexto semántico (Smart parsing). [Referencia](#).
- Por elementos lógicos (ej.: en código por funciones o clases)

# RAGas (Retrieval-Augmented Generation Assessment)

# Framework para evaluar sistemas RAG, utilizando métricas basadas en LLMs y tradicionales.

Repo: <https://github.com/explodinggradients/ragas>

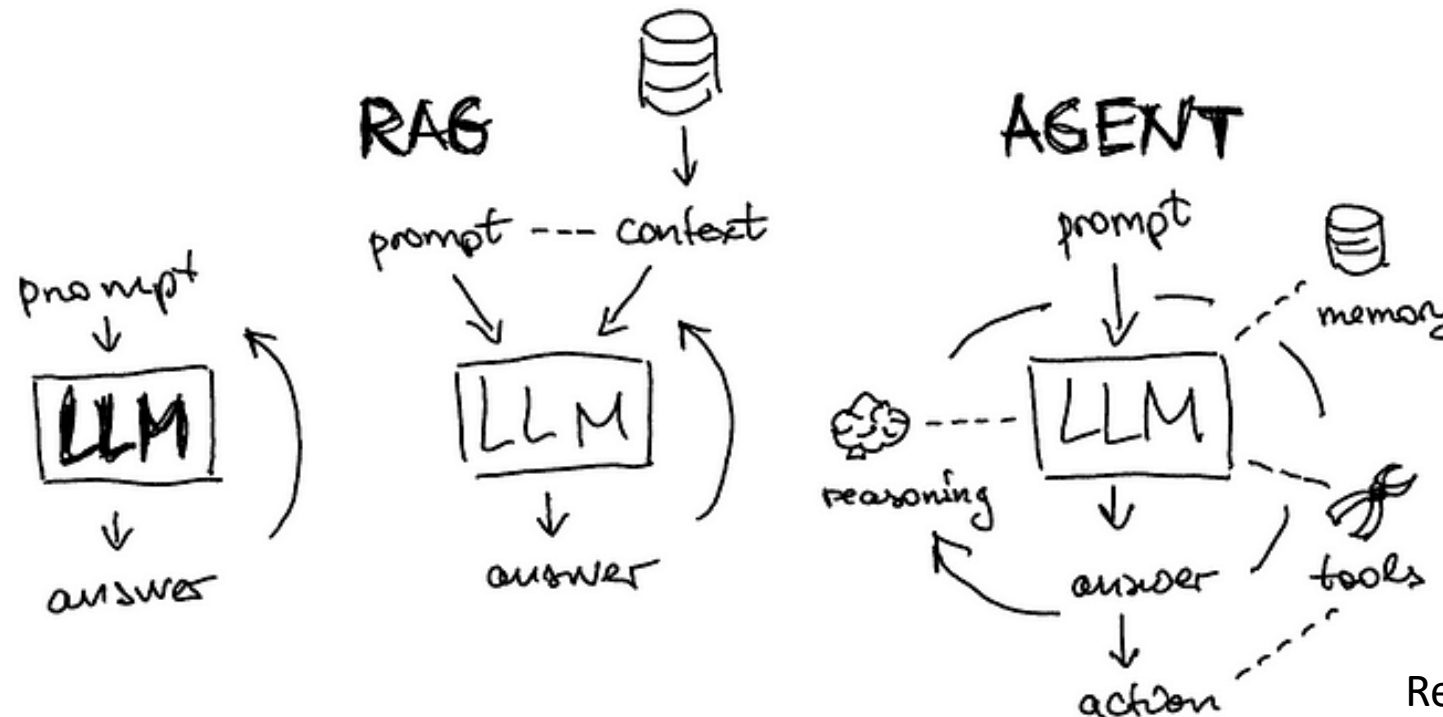
Ejemplo: <https://towardsdatascience.com/evaluating-rag-applications-with-ragas-81d67b0ee31a>

question	contexts	answer	ground_truths	context_precision	context_recall	faithfulness	answer_relevance
0 What did the president say about Justice Breyer?	['Tonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Breyer. And I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. A former top litigator in private practice. A former federal public defender. And from a first-class family of public servants.']	The president thanked Justice Breyer for his service and referred to him as an Army veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. The president also mentioned that he nominated Circuit Court of Appeals Judge Ketanji Brown Jackson, who will continue Justice Breyer's legacy of excellence.	['The president said that Justice Breyer has dedicated his life to serve the country and thanked him for his service.']	0.50	1.00	1.00	0.85
1 What did the president say about Intel's CEO?	['But that's just the beginning. Intel's CEO, Pat Gelsinger, who is here tonight, told me that this is where Intel, the American company that helped build Silicon Valley, is going to go. For the past 40 years we were told that if we gave tax breaks to those at the very top,']	The president did not mention Intel's CEO specifically in the given context.	['The president said that Pat Gelsinger is ready to increase Intel's investment to \$100 billion.']	0.00	1.00	0.50	0.83
2 What did the president say about gun violence?	['And I ask Congress to pass proven measures to reduce gun violence. Pass universal background checks. As I said last year, especially to our younger transgender Americans, I will always have your back. Let's stop seeing each other as enemies, and start seeing each other for who we really are.']	The president called for Congress to pass measures to reduce gun violence, including universal background checks and a ban on assault weapons and high-capacity magazines. He also mentioned the need to repeal the liability shield for gun manufacturers.	['The president asked Congress to pass proven measures to reduce gun violence.']	1.00	1.00	1.00	0.91

# Agentes:

- Concepto:

Sistemas o programas que interactúan con un entorno o realizan tareas específicas de manera autónoma, tomando decisiones basadas en la información que tienen a su disposición.



Referencia: [Paper REACT](#)

## Características de los Agentes:

- **Alta autonomía:** pueden planificar, tomar decisiones y ejecutar acciones sin supervisión constante.
- **Capacidad para manejar tareas complejas:** gestionan flujos de trabajo y problemas que requieren múltiples pasos.
- **Aprendizaje y adaptación:** emplean aprendizaje automático para mejorar su rendimiento con la experiencia.
- **Uso de herramientas y acceso a datos externos:** pueden integrar APIs, bases de datos y sistemas corporativos para cumplir objetivos.
- **Razonamiento contextual:** interpretan la situación, el entorno y el historial para tomar decisiones más acertadas.

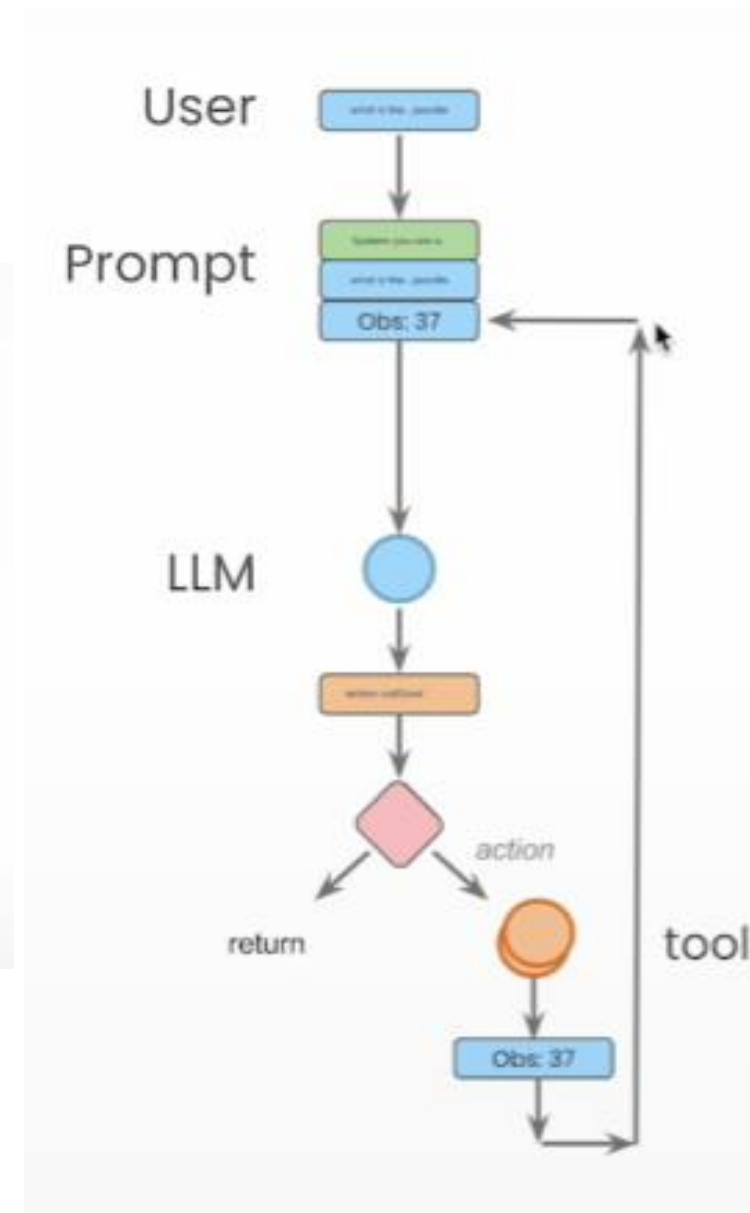
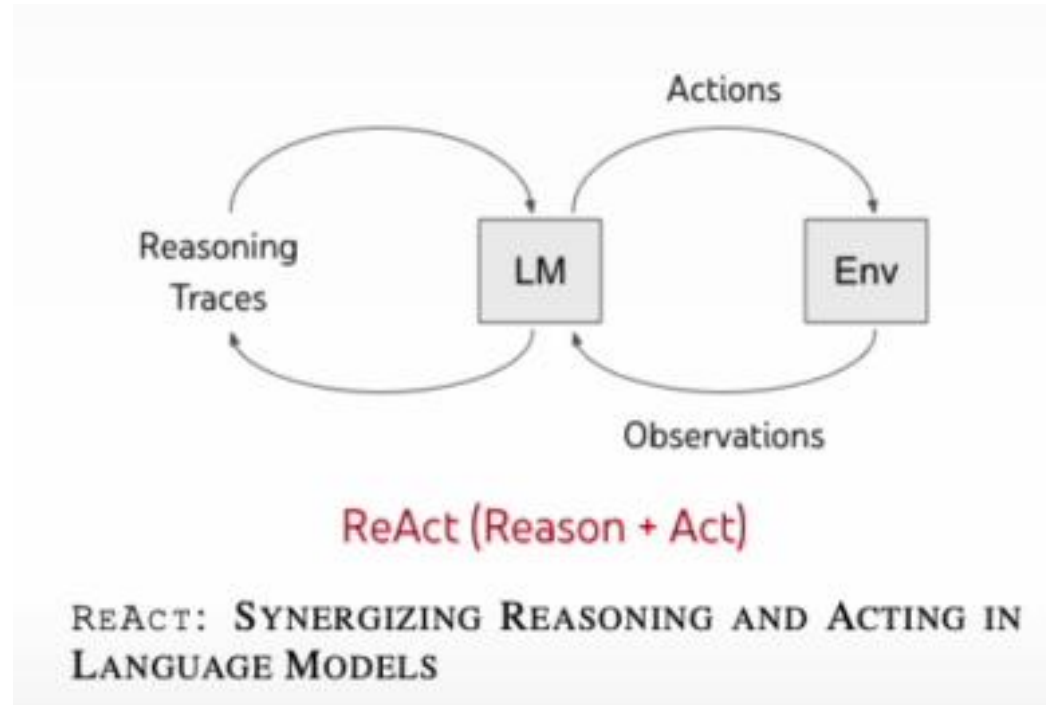


# Gestión de las tareas de los Agentes:

- **Agente único** con herramientas (ReAct). Un AgentExecutor corre un bucle percepción→razonamiento→acción.El agente (LLM + prompt + memoria/scratchpad) elige qué tool llamar en cada paso y cuándo parar.Control centralizado en el agente.
- **Router**: Otro LLM decide a qué cadena/agente enviar cada input (por dominio o por intención).Patrón tipo “supervisor simple”: delega a X o Y según la consulta.
- **Plan-and-Execute** Un planner (agente) descompone el objetivo en subtareas (tasks) en una lista.Un executor (agente) ejecuta cada subtask con herramientas.Puede re-priorizar/insertar nuevas.
- **Orquestación** (LangGraph): Grafo de nodos (agentes, cadenas, verificadores) y un supervisor que elige el siguiente nodo según el estado.Permite ciclos controlados, políticas de reintento, límites de pasos, paralelismo y checkpoints.
- **Paralelización de** ejecuciones. (funciones Async). La decisión dentro de cada agente sigue siendo secuencial por diseño.

# Agentes:

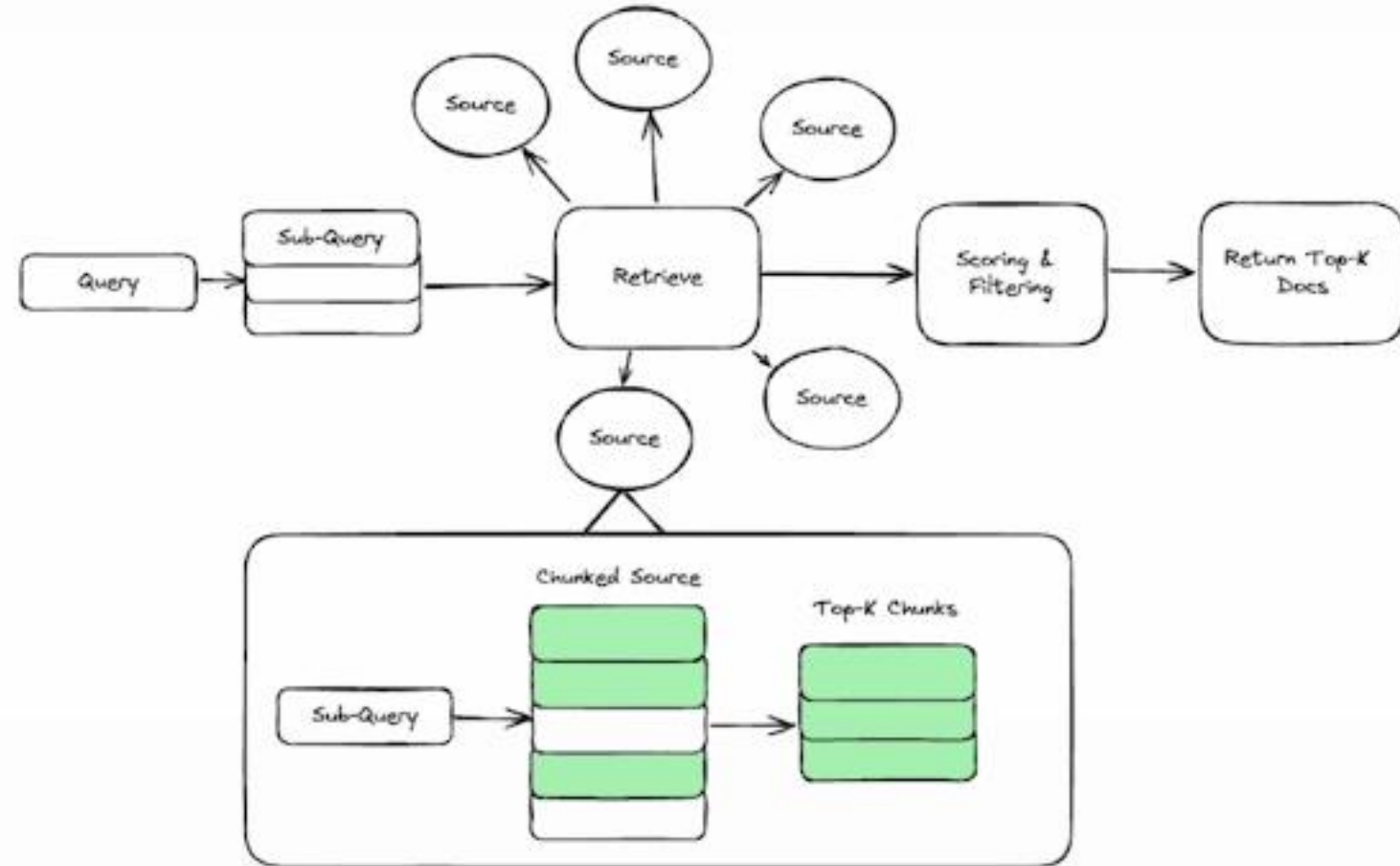
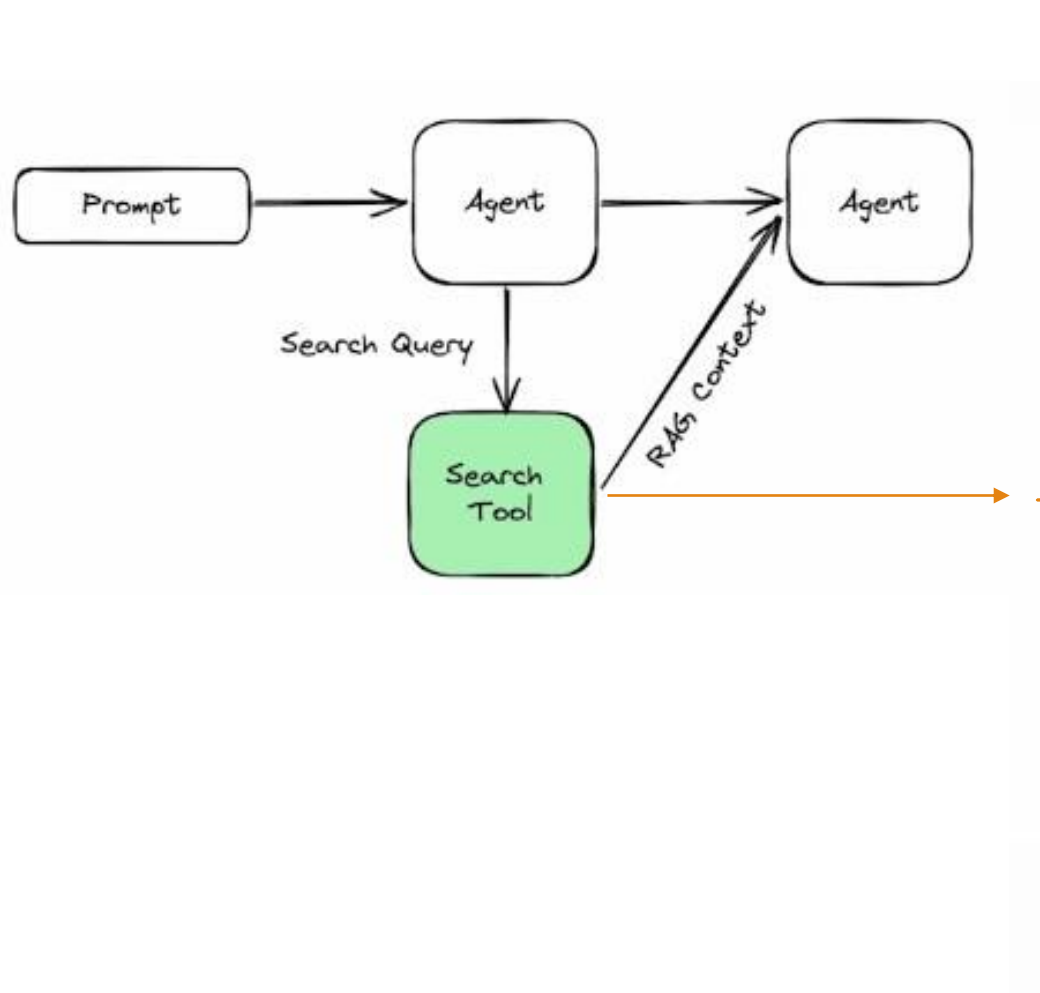
- Ejemplo: ReAct



Referencia [5] Yao et al. [“ReAct: Synergizing reasoning and acting in language models.”](#) ICLR 2023.

# Agentes:

- Search Tool



# LLM Model Fine Tuning:

- Cuando usarlo?! (prompting Vs. RAG Vs. Fine tuning) -> Clase 8
- Huggingface Transformers / Spacy models fine tuning:
  - On Premise – Open Source
  - En general tamaños chicos de LLMs
  - Privacidad de datos
- OpenAI fine tuning
  - Por medio de API. OpenAI gestiona la infraestructura
  - Se utiliza en modelos grandes
  - El costo se calcula por token
  - El modelo queda guardado por OpenAI

# LLM Model Fine Tuning - Huggingface Transformers / Spacy models fine tuning

Ejemplos de uso:

- Clasificación de texto
- Respuestas a preguntas
- Generación de texto
- Traducción
- NER (Named Entity Recognition)
- Resumen (summarization)



# LLM Model Fine Tuning: OpenAI fine tuning

[\(https://platform.openai.com/docs/guides/fine-tuning/\)](https://platform.openai.com/docs/guides/fine-tuning/)

Disponibles: GPT 4.1, 4o, 4o mini, 3.5 turbo

Casos de uso:

- a) Establecer el estilo, tono, formato u otros aspectos cualitativos.
- b) Mejorar la confiabilidad al generar una salida deseada.
- c) Corregir fallos al seguir indicaciones complejas.
- d) Manejar situaciones específicas.
- e) Generar una nueva habilidad o tarea que sea difícil de articular en una indicación.

# LLM Model Fine Tuning: OpenAI fine tuning

(<https://platform.openai.com/docs/guides/fine-tuning/>)

## Ejemplo de preparación de datos (json):

Validación del dataset: [https://cookbook.openai.com/examples/chat\\_finetuning\\_data\\_prep](https://cookbook.openai.com/examples/chat_finetuning_data_prep)

```
{  
  "messages": [  
    {"role": "system", "content": "Marv es un chatbot factual que también es sarcástico."},  
    {"role": "user", "content": "¿Cuál es la capital de Francia?"},  
    {"role": "assistant", "content": "París, como si todo el mundo no lo supiera ya."}  
  ]  
}
```

Cargar datos ->

```
from openai import OpenAI  
client = OpenAI()  
  
client.files.create(  
  file=open("mydata.jsonl", "rb"),  
  purpose="fine-tune"  
)
```

# LLM Model Fine Tuning: OpenAI VISION fine tuning

Modelo de dato de  
entrenamiento --->

```
{
  "messages": [
    { "role": "system", "content": "Eres un asistente que identifica quesos poco comunes." },
    { "role": "user", "content": "¿Qué es este queso?" },
    { "role": "user", "content": [
      {
        "type": "image_url",
        "image_url": {
          "url": "https://upload.wikimedia.org/wikipedia/commons/3/36/Danbo_Cheese.jpg"
        }
      }
    ]
  },
  { "role": "assistant", "content": "Danbo" }
]
```

Restricciones: Máximo 50k ejemplos. Cada ejemplo máximo 10 imágenes. Cada imagen máximo 10 MB

Formatos: JPEG, PNG, WEBP (RGB / RGBA).

Policy: So se puede entrenar con imágenes de personas, caras, chicos ni CAPTCHAs

# LLM Model Fine Tuning: OpenAI fine tuning

Crear modelo --->

```
from openai import OpenAI  
client = OpenAI()  
  
client.fine_tuning.jobs.create(  
    training_file="file-abc123",  
    model="gpt-4o-mini-2024-07-18"  
)
```

Hiperparámetros--->

- Batch Size (cuanto mayor menor es la varianza)
- Learning Rate Multiplier (Factor de escala del learning rate)
- n\_epochs (un epoch es un ciclo entero del dataset)

# LLM Model Fine Tuning: precios OpenAI / modelos Llama

Precios OpenAI ->

**gpt-4o-2024-08-06**

\$10.00 / 1M output tokens

\$5.00 / 1M output tokens

\$2.50 / 1M input tokens

\$1.25 / 1M input tokens

\$1.25 / 1M cached\*\* input tokens

\$10.00 / 1M output tokens

\$5.00 / 1M output tokens

**Model**

**Pricing**

**Pricing with Batch API\***

**gpt-4o-2024-08-06**

\$3.750 / 1M input tokens

\$1.875 / 1M input tokens

\$1.875 / 1M cached\*\* input tokens

\$15.000 / 1M output tokens

\$7.500 / 1M output tokens

Framework de fine tuning  
para modelos Llama ->

<https://github.com/pytorch/torch tune>



## Trabajo práctico Nro 3:

- Sobre el trabajo de la clase 6: implementar un sistema de agentes para que responda de manera eficiente dependiendo de qué persona se está preguntando (1 agente por persona).
- **Entregables:** Link a repo público y captura de video de chatbot consultando los CVs de los integrantes del equipo. Por defecto, cuando no se nombra a nadie en la query, utilizar el Agente del alumno.  
Atención: Si se consulta por más de un CV, traer el contexto de cada uno y responder de manera acorde. El video debe mostrar todas estas funcionalidades!!
- Link para entregar: <https://forms.gle/roAtRx2rXawaftjm7>
- Se evalúa el código y la presentación (Repo con documentación).

## Ejercicio en clase:

- Pasos:
  1. Esquematizar el diagrama flujo que debe tener la aplicación.
  2. Definir el “conditional Edge”, tomador de la decisión. Consejo: utilizar librería **re** y su método **match**.
  3. Implementar cada uno de los pasos y compilar el diagrama de flujo.