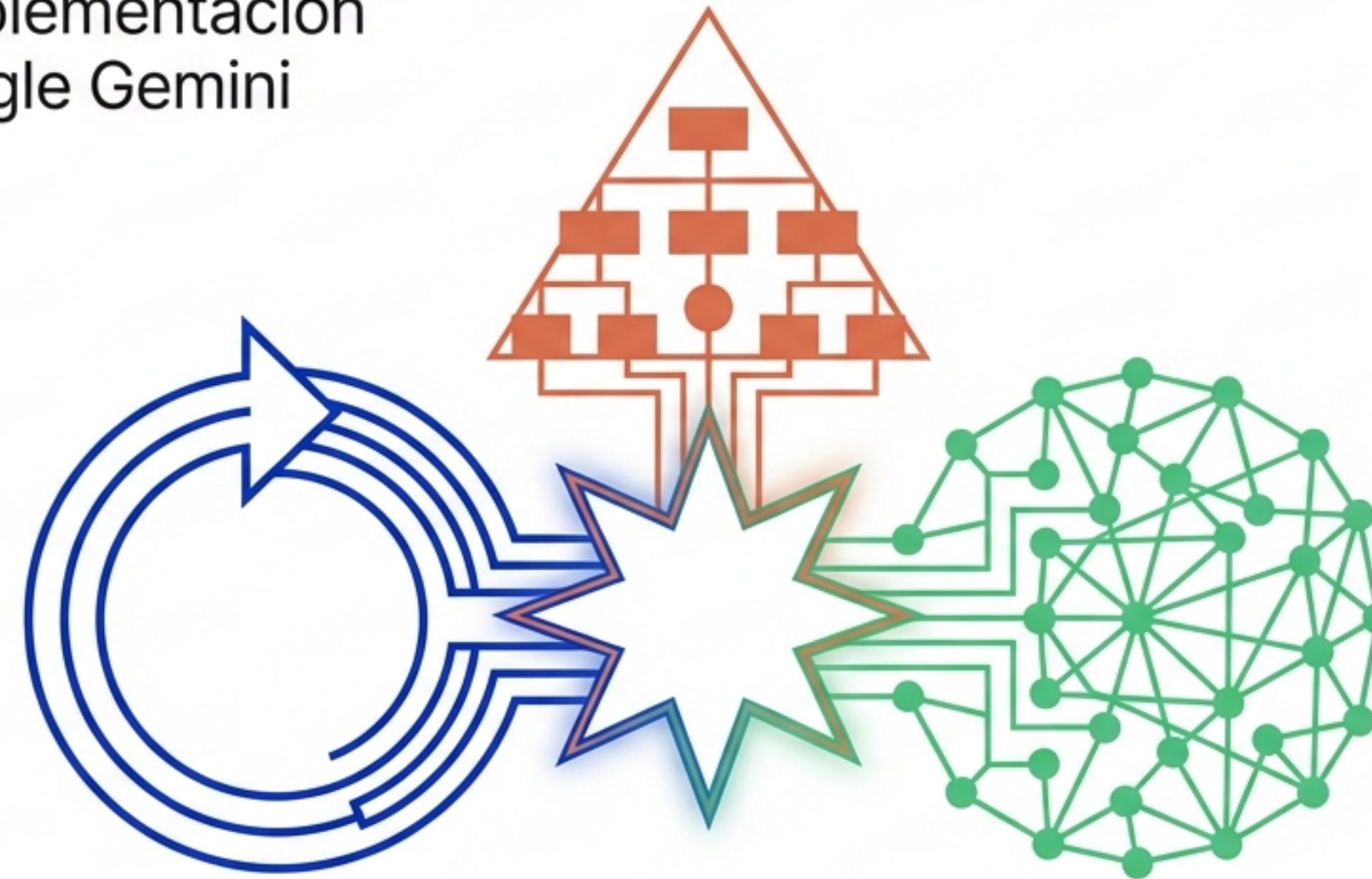


Orquestación de Sistemas Multi-Agente: LangGraph, CrewAI y AutoGen

Módulo 7.3 — Implementación
práctica con Google Gemini



El Desafío de la Orquestación: Tres Paradigmas

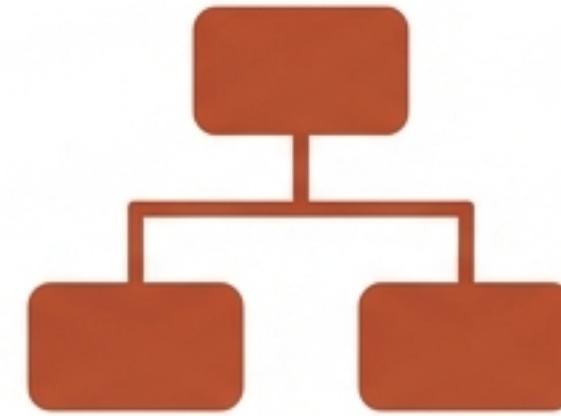
Un solo modelo de IA no es suficiente para flujos de trabajo complejos. Para escalar, necesitamos coordinar equipos de agentes utilizando estructuras lógicas distintas.



LangGraph

Control Cílico

Grafos de Estado



CrewAI

Roles Especializados

Estructura Organizacional



AutoGen

Conversación Dinámica

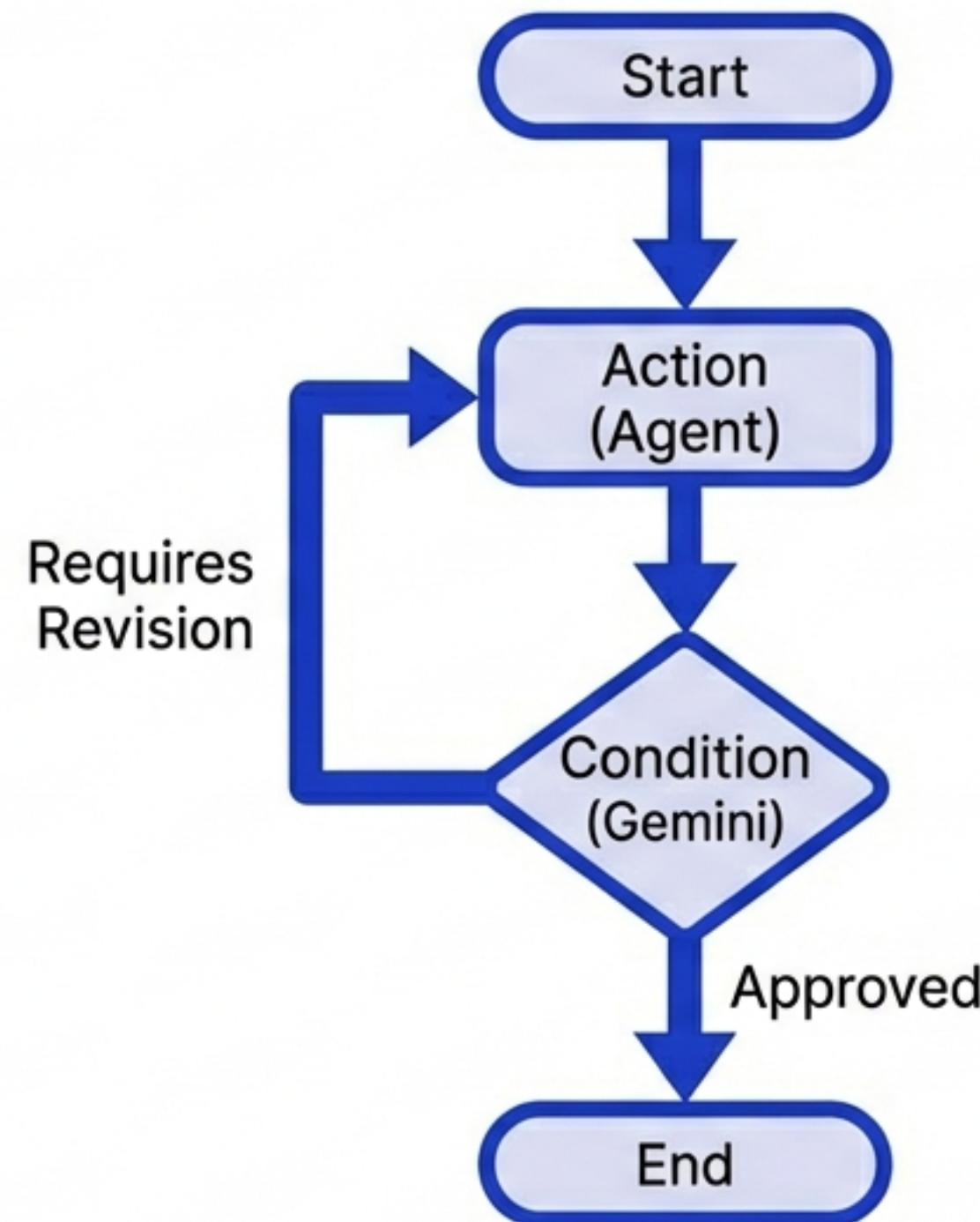
Colaboración vía Chat

Todos los paradigmas utilizan **Google Gemini** como motor de razonamiento subyacente.

LangGraph: Arquitectura Basada en Grafos de Estado

Extensión de LangChain diseñada para permitir ciclos, bifurcaciones y persistencia en la ejecución de agentes.

Ideal para workflows donde el agente necesita recordar, iterar y corregirse a sí mismo antes de entregar un resultado final.



Componentes Clave de LangGraph



StateGraph

Contenedor del grafo que define la estructura del workflow.

Inter



Nodes

Funciones ejecutables que procesan y modifican el estado.

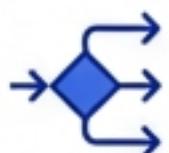
Inter



Edges

Conexiones directas para flujo secuencial.

Inter



Conditional Edges

Bifurcaciones dinámicas (Routing) basadas en lógica o decisión del LLM.

Inter



Cycles

Conexiones de retorno para refinamiento iterativo.

Inter

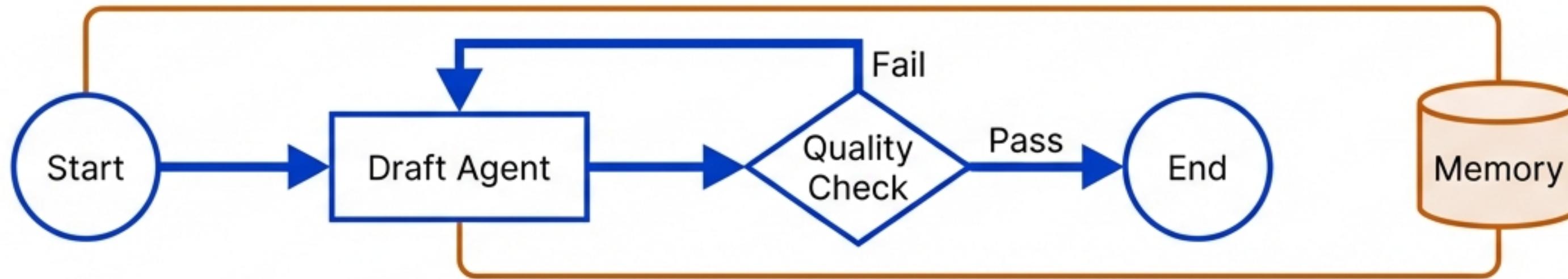


Checkpointer

Sistema de persistencia para memoria a largo plazo entre sesiones.

Inter

Implementación: Grafo con Memoria Persistente



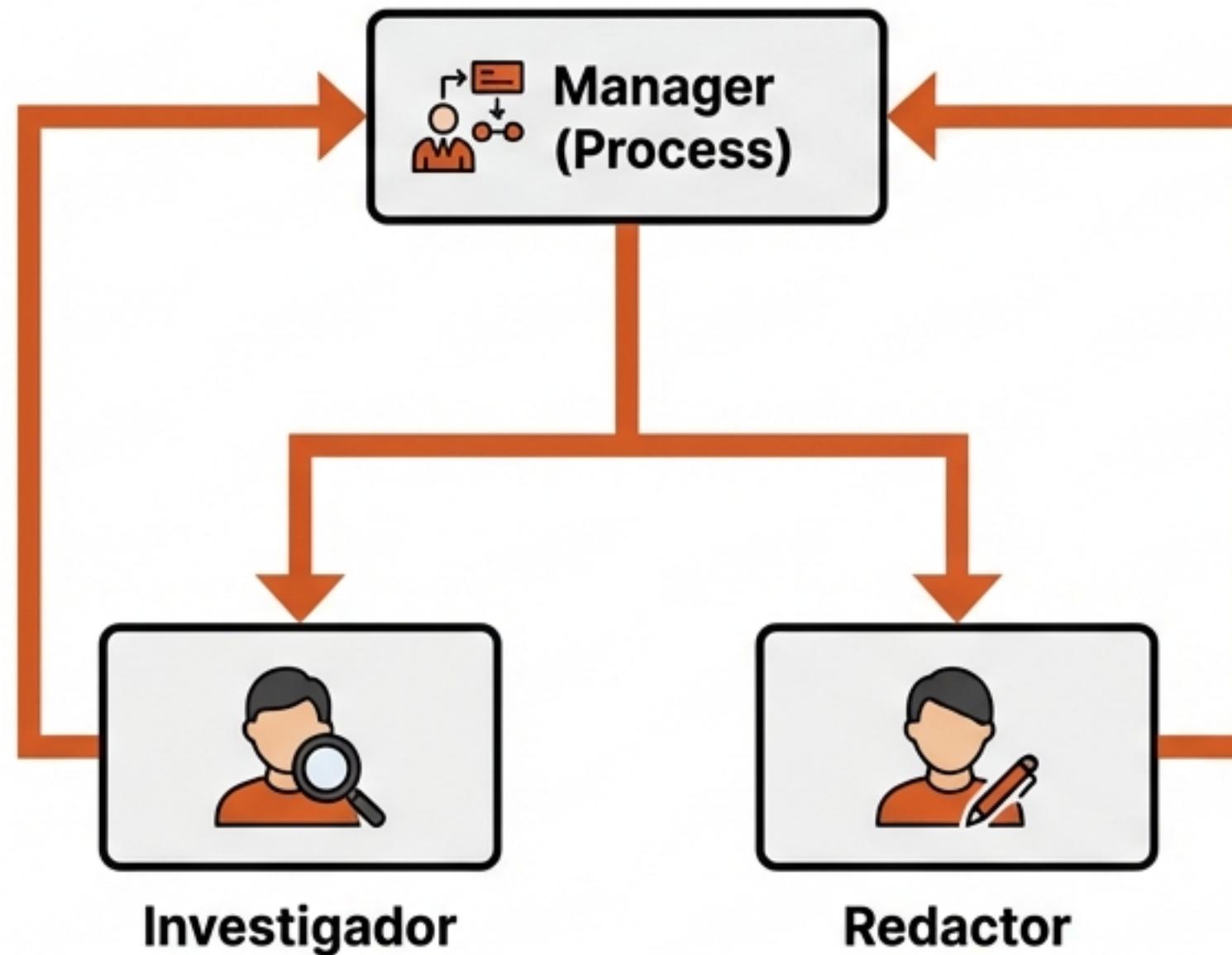
```
# Configuración del Grafo
workflow = StateGraph(AgentState)

# Añadir Nodos
workflow.add_node("draft", draft_agent)

# Lógica Condicional (Ciclo)
workflow.add_conditional_edges(
    "draft",
    should_continue,
    {
        "continue": "draft",
        "end": END
    }
)

# Compilar con Memoria
app = workflow.compile(checkpointer=memory)
```

CrewAI: Estructura de Roles y Metas



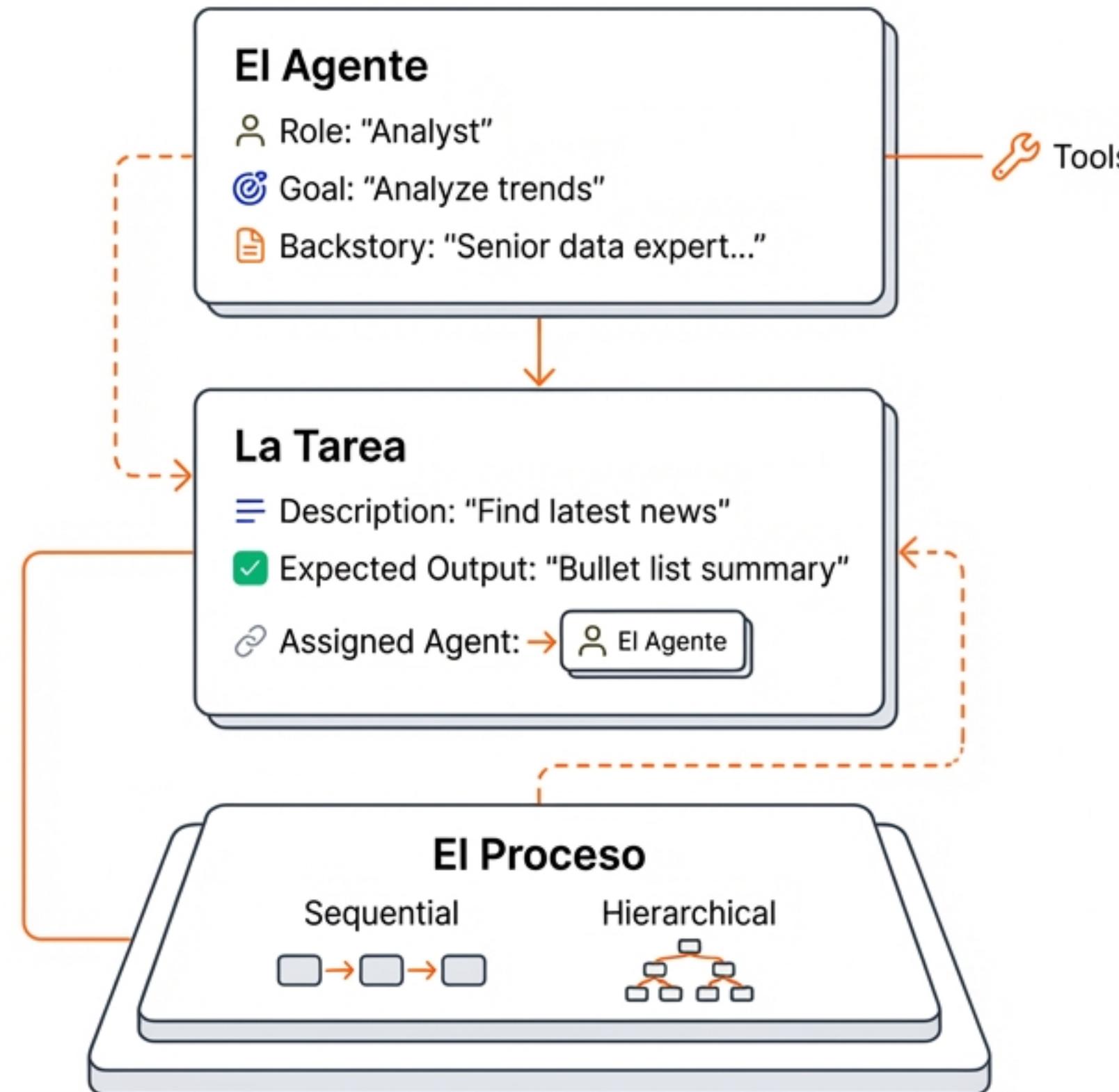
Concept: Modelado de agentes como “empleados” con descripciones de trabajo claras, orquestados para cumplir objetivos específicos.

Focus Points

- Enfoque en el ‘Quién’ (Rol)
- Enfoque en el ‘Qué’ (Meta)
- Abstracción del ‘Cómo’ (Grafo subyacente)

Context: Facilitación de equipos colaborativos donde cada agente tiene un Backstory y un Goal definido.

Anatomía de un “Crew”



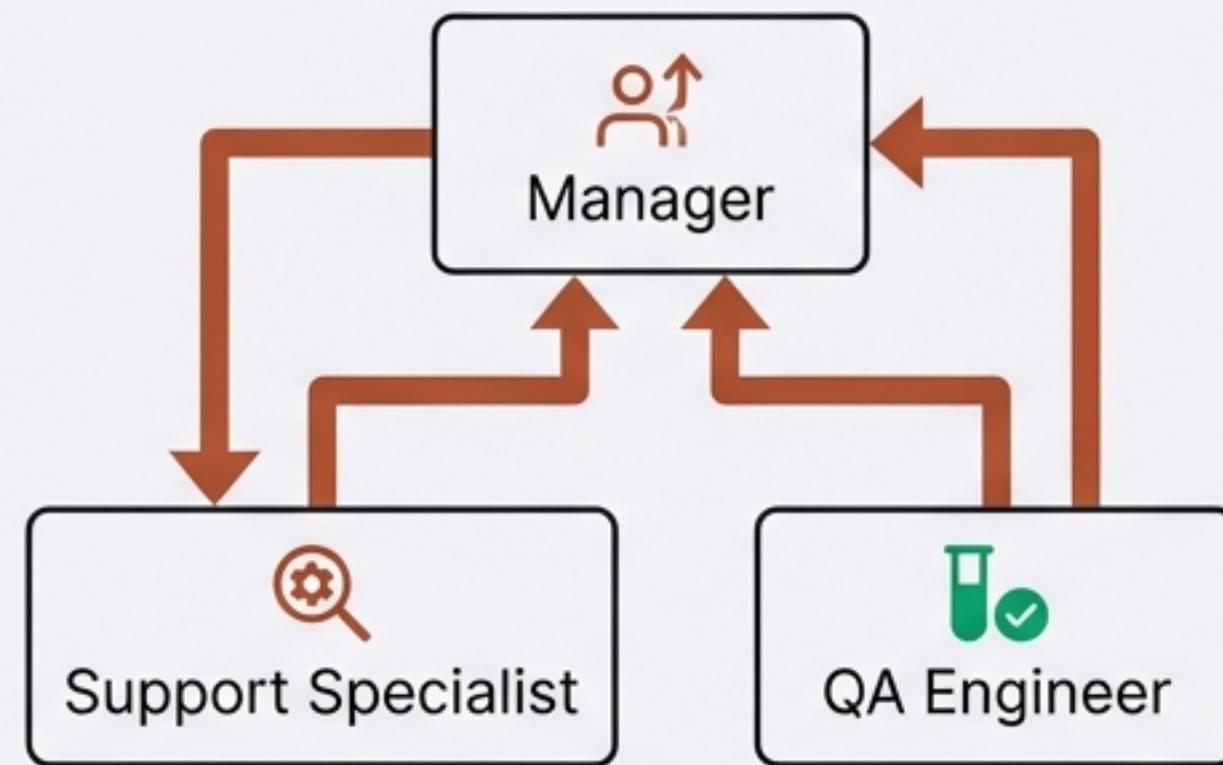
Definición en Python:

```
agent = Agent(role='Analyst',  
goal='...', tools=[SearchTool])
```

Escenarios de Uso: Configuración de Equipos

Soporte Técnico

Process: Hierarchical



Goal: Resolución de tickets y debugging.

Marketing Content

Process: Sequential



Goal: Creación de campaña viral.

La estructura del equipo define la calidad del resultado.

AutoGen: La Conversación como Computación

Session: Data Analysis

UserProxy: Plot the stock price of NVDA for 2023.

AssistantAgent (Gemini): Here is the Python code to retrieve and plot the data.

```
import yfinance as yf  
import matplotlib.pyplot as plt  
# ... (code snippet)
```

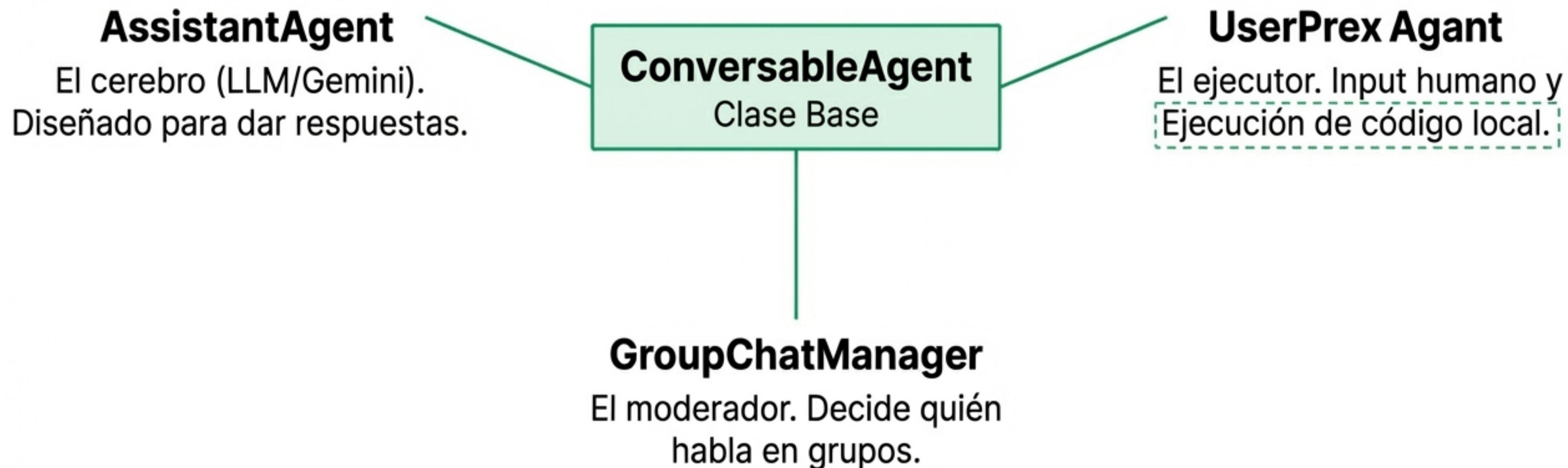
UserProxy: Executing code...
Error: Module 'yfinance' not found.

AssistantAgent: Apologies. I will install the library first. Here is the corrected code.

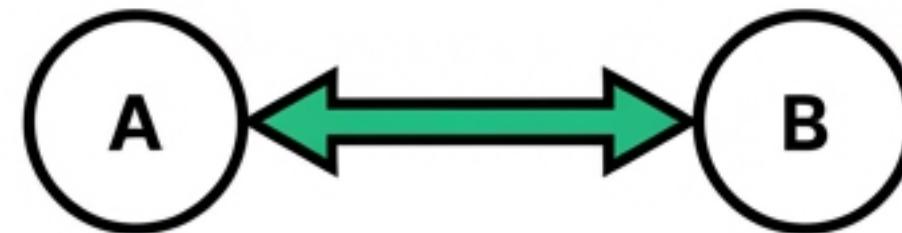
```
!pip install yfinance  
import yfinance as yf  
# ... (code snippet)
```

Automatización a través del diálogo natural, debate y ejecución de código.

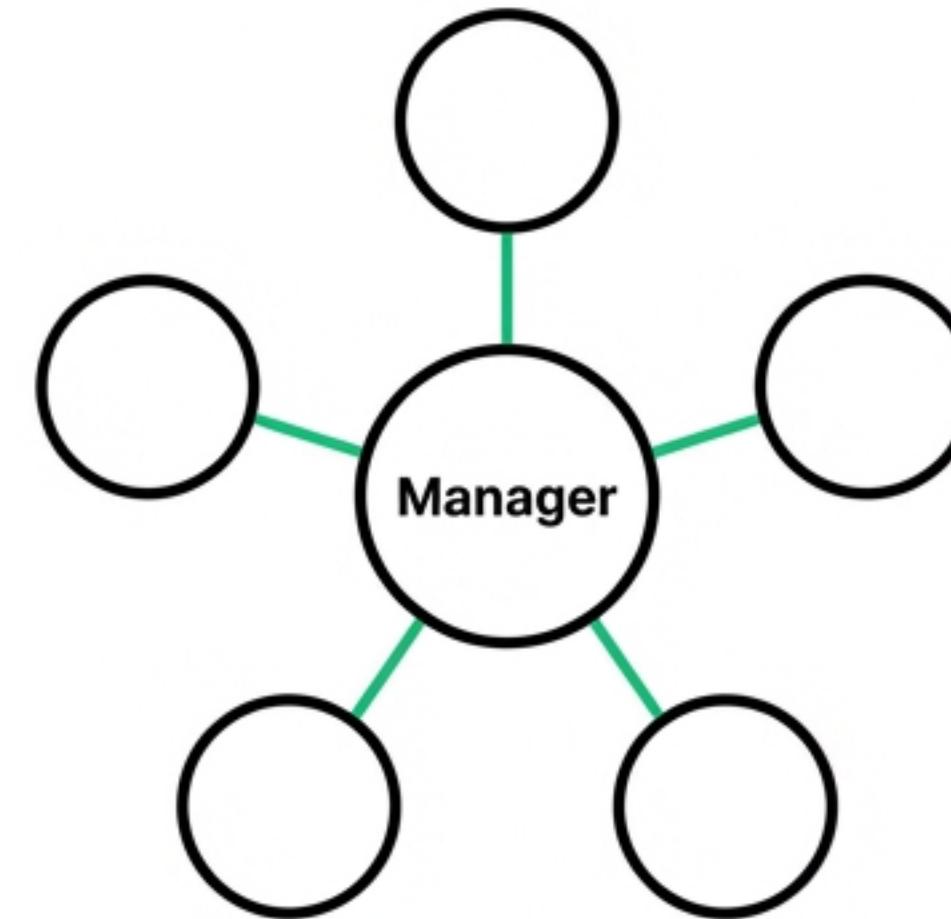
Ecosistema AutoGen: Clases Principales



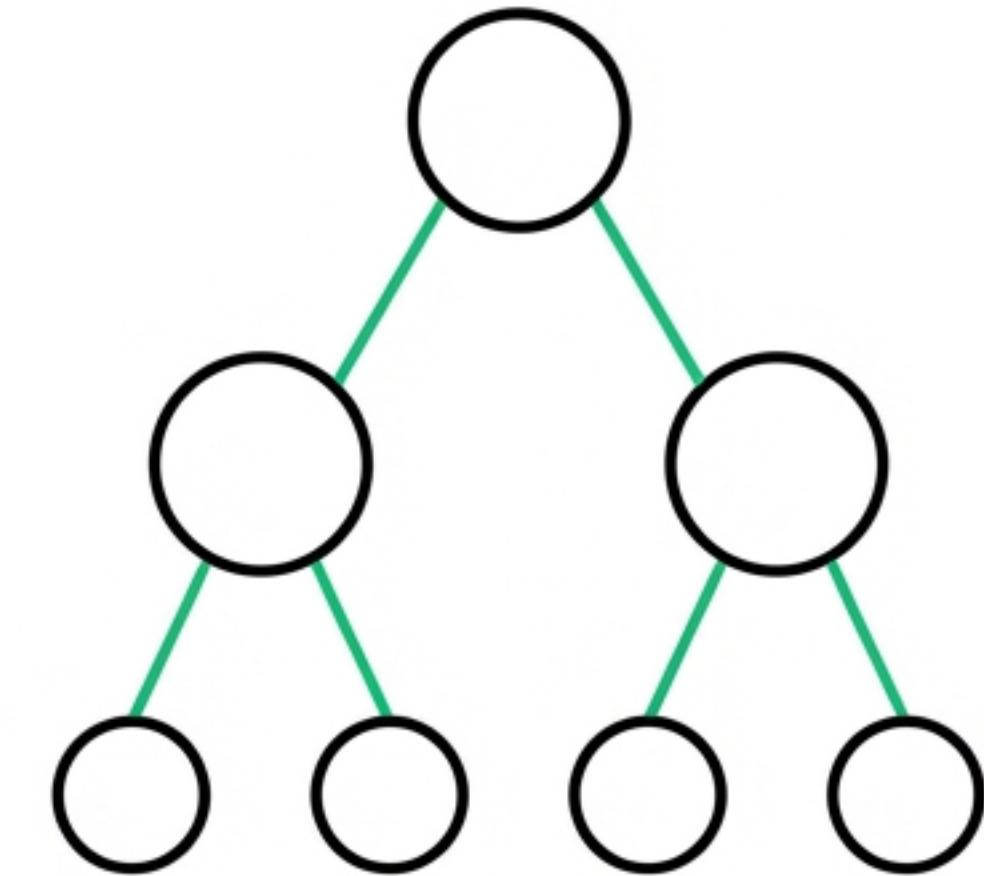
Patrones de Conversación Avanzados



Uno a Uno



Panel de Expertos
(Hub & Spoke)



Jerárquico

Gemini provee el razonamiento para que el AssistantAgent debata eficazmente en cualquier topología.

Matriz de Decisión: ¿Qué Framework elegir?

	LangGraph	CrewAI	AutoGen
Estilo de Control	Control Granular (Grafo)	Roles y Procesos	Conversación Libre
Complejidad Setup	Alta	Media	Variable
Mejor Caso de Uso	Workflows con ciclos y validación estricta.	Procesos secuenciales tipo “línea de ensamblaje”.	Resolución creativa y generación de código.

Resumen del Módulo 7.3



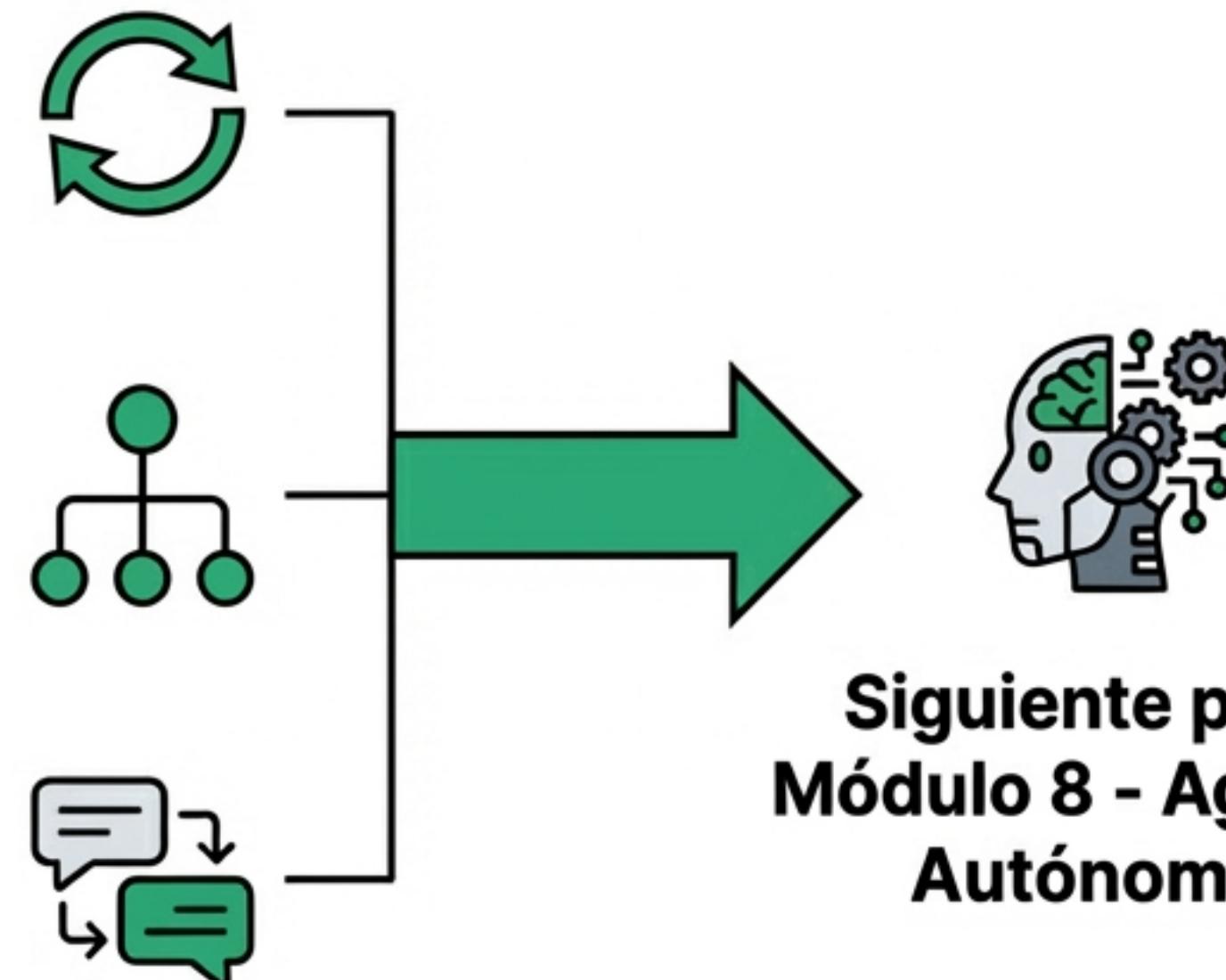
LangGraph: Implementación de ciclos de refinamiento y memoria persistente.



CrewAI: Orquestación de agentes basada en roles y tareas secuenciales.



AutoGen: Creación de sistemas multi-agente basados en diálogo y ejecución de código.



Siguiente paso:
Módulo 8 - Agentes Autónomos



Google Gemini: Orquestación de Inteligencia

Arquitectura de Datos y Desarrollo de IA

Documentación Técnica Módulo 7.3