

Brandon David Rambauth Cespedes

Harol Yesid Rambauth Cespedes

ARQUITECTURAS AGÉNTICAS SECUENCIALES PARA INTELIGENCIA DE MERCADOS Y ANALÍTICA WEB

I. Abstract

Este documento describe el diseño e implementación de un Sistema Multi-Agente (MAS) destinado a la Inteligencia de Mercados mediante la analítica de texto y la adquisición de datos web. La arquitectura propuesta sigue un Patrón Secuencial/Jerárquico, orquestando un conjunto de Agentes especializados mediante la librería crewai. El sistema integra un Large Language Model (LLM) (Gemini) con la herramienta de búsqueda Tavily para adquirir información fresca en la web, analizarla, asignar una calificación justificada y generar un informe final estructurado. Los resultados demuestran la capacidad del MAS para automatizar el ciclo de vida completo de la analítica, desde la consulta en lenguaje natural hasta la entrega de insights accionables, superando las limitaciones de los scrapers monolíticos.

I. Introducción

La evolución de la Inteligencia Artificial (IA) se dirige hacia la construcción de Sistemas Agénticos capaces de problemas que están más allá de la capacidad de un agente individual. Modelos de Lenguaje Grande (LLMs): Gemini se utiliza como núcleo de razonamiento de los agentes.

interactuar con el entorno y tomar decisiones. Este proyecto aborda la necesidad de automatizar el proceso de Text & Web Analytics para fines de Inteligencia de Mercados, específicamente en el análisis comparativo de productos.

El objetivo principal es diseñar un Sistema Multi-Agente (MAS) que: 1) Interprete una consulta compleja en lenguaje natural (ej. 'relojes sumergibles baratos'); 2) Adquiera datos en tiempo real de la web utilizando herramientas determinísticas; 3) Procese la información mediante análisis de texto (asignación de score y justificación); y 4) Genere un resultado estructurado y conciso.

II. Marco Teórico

Esta sección explica los fundamentos teóricos que soportan la arquitectura del proyecto:

Sistemas Multi-Agente (MAS): Se define el concepto de MAS como un conjunto de agentes que interactúan para resolver

Patrones Agénticos: Se usó el Patrón Secuencial/Jerárquico.

Web & Text Analytics: Incluye análisis de texto y herramientas como Tavily.

III. Metodología y Arquitectura

La solución fue implementada en Python utilizando el framework crewai, que permite la definición, especialización y coordinación de agentes autónomos. Cada agente opera con un objetivo específico y se integra con herramientas externas como el motor de búsqueda Tavily para completar tareas de adquisición y análisis de información en tiempo real.

A. Arquitectura del Sistema

El sistema está compuesto por cuatro agentes especializados, diseñados para trabajar dentro de un flujo de procesamiento estrictamente lineal:

Agente Orquestador (Task Orchestrator)

Coordina la ejecución completa del sistema. Recibe la consulta del usuario, inicializa la crew y desencadena las tareas secuenciales.

Agente Rastreador Web (Web Scraping / Tavily Agent)

Utiliza la herramienta TavilySearchTool para obtener información actualizada desde la web. Su función es recolectar descripciones, precios, características y enlaces relevantes según la consulta del usuario.

Agente Analista de Texto (Text Analyst Agent)

Procesa el contenido entregado por el rastreador. Emplea el LLM para realizar un análisis comparativo, asignar un score

justificado a cada producto y estructurar la información conforme a criterios definidos.

Agente Generador del Informe Final (Report Generator Agent)

Recibe la lista analizada y calificada. Selecciona, ordena y sintetiza el Top 3, generando una salida clara, concisa y orientada a decisiones.

El diseño sigue un Patrón Secuencial/Jerárquico, donde la salida de cada agente alimenta directamente al siguiente. Esto asegura trazabilidad, coherencia en el flujo de datos y evita inconsistencias entre fases.

B. Justificación del Patrón de Diseño

Se seleccionó el Patrón Secuencial debido a que el proceso de Web & Text Analytics imita un modelo de cadena de valor:

adquisición → 2) análisis → 3) síntesis.

Este patrón implementado mediante `process = Process.sequential` garantiza:

- Control total sobre el orden de ejecución.
- Dependencias claras entre agentes.
- Minimización de errores durante la transferencia de información.
- Un flujo reproducible y fácil de depurar.

Este enfoque es ideal para tareas donde cada fase depende directamente de los resultados de la anterior, como ocurre en el análisis automatizado de productos.

IV. Resultados y Discusión

Se probó la consulta: 'dame el top 3 de los relojes sumergibles baratos que me recomiendes'.

V. Conclusiones

Para validar el comportamiento del sistema, se ejecutó la consulta:

“Dame el top 3 de los relojes sumergibles baratos que me recomiendes.”

El sistema realizó:

Adquisición automática de información desde múltiples fuentes web mediante Tavily.

Procesamiento analítico, comparando precios, resistencia al agua, calidad y valor percibido.

Cálculo de un score justificado, permitiendo priorización objetiva entre alternativas.

Síntesis estructurada del Top 3, presentada en un formato final claro y accionable.

Este resultado confirma que el MAS no solo obtiene información, sino que además es capaz de razonar, evaluar y sintetizar, generando insights útiles para decisiones de compra o benchmarking competitivo.

