

# Sistema Multiagente para Turismo Inteligente

Gabriel Herrera Carrazana, Adrian A Souto Morales, and Lauren Peraza

Universidad de la Habana

**Abstract.** Se presenta el desarrollo de un sistema multiagente orientado a la simulación y recomendación turística, integrando técnicas de crawling paralelo y generación aumentada por recuperación (RAG). El sistema fue implementado en Python y se encuentra disponible en: <https://github.com/Ghavvvo/Proyecto-IA-SIM-SRI>.

## 1 Introducción

El turismo representa uno de los sectores económicos más importantes a nivel mundial, generando millones de empleos y contribuyendo significativamente al PIB de numerosos países. En la era digital, la industria turística ha experimentado una transformación profunda, evolucionando hacia el concepto de "turismo inteligente" que aprovecha las tecnologías de la información y comunicación para mejorar la experiencia del viajero y optimizar la gestión de destinos.

El turismo moderno se caracteriza por la abundancia de información disponible en línea, desde reseñas de hoteles y restaurantes hasta guías de destinos y recomendaciones personalizadas. Sin embargo, esta riqueza informativa presenta desafíos significativos: la sobrecarga de información, la dificultad para encontrar contenido relevante y personalizado, y la necesidad de procesar grandes volúmenes de datos de manera eficiente. Los turistas actuales demandan experiencias personalizadas, información actualizada y recomendaciones que se adapten a sus preferencias específicas, perfil de viaje y contexto situacional.

## 2 Autores

Gabriel Herrera Carrazana  
Adrian A Souto Morales  
Lauren Peraza

## 3 Descripción del tema

El sistema implementado integra diversas tecnologías y enfoques avanzados para optimizar la gestión de información especializada en el dominio turístico. Se adoptó una arquitectura multiagente que permite la coordinación y colaboración entre componentes especializados, alineada con la arquitectura genérica del modelo Retrieve-Augmented Generation (RAG).

Los requisitos clave del sistema incluyen: experimentación mediante procesos iterativos de prueba y validación; procesamiento del lenguaje natural para interpretar y mejorar datos textuales; aplicación de metaheurísticas para abordar problemas de optimización; diseño de una estructura de representación del conocimiento que organice la información de manera accesible; e implementación de un crawler automatizado que opera en dos fases: una fase inicial de recopilación y almacenamiento en repositorio vectorial, y una fase dinámica de actualización automática basada en consultas específicas.

El sistema debe facilitar la recuperación eficiente de información turística y proporcionar recomendaciones personalizadas, integrando técnicas de inteligencia artificial para mejorar continuamente la experiencia del usuario en el contexto del turismo inteligente.

## 4 Soluciones implementadas

El sistema desarrollado se compuso de una arquitectura multiagente modular y escalable, conformada por ocho agentes especializados: crawler, RAG, coordinador, interfaz, contexto, simulador de turistas, agente de rutas y guía turístico. Cada agente cumplió funciones específicas y se comunicó mediante mensajes, facilitando la escalabilidad y el mantenimiento.

El agente crawler (CrawlerAgent) implementó crawling paralelo con hasta 10 hilos configurables, permitiendo explorar grandes volúmenes de páginas en menor tiempo. Se incorporó un filtrado semántico avanzado de URLs basado en palabras clave, sinónimos y patrones relevantes para turismo, mejorando la calidad de los datos almacenados. El agente integró tres modalidades de operación: crawling inicial con exploración sistemática de URLs semilla y procesamiento Mistral opcional; crawling por palabras clave con búsqueda dirigida basada en términos específicos; y exploración ACO que integra el Algoritmo de Colonia de Hormigas con búsquedas en DuckDuckGo para descubrir fuentes novedosas guiadas por feromonas y heurísticas de relevancia.

El agente RAG (RAGAgent) implementó un sistema de generación aumentada por recuperación que inicializa con colecciones ChromaDB y procesa consultas mediante recuperación semántica. El sistema utiliza embeddings vectoriales para encontrar documentos relevantes y genera respuestas contextualizadas usando modelos de lenguaje, proporcionando respuestas precisas basadas en la información almacenada en la base de conocimiento vectorial.

Se desarrolló un agente de simulación de turistas (TouristSimulationAgent) que utilizó lógica difusa para modelar la satisfacción, cansancio y experiencia de diferentes perfiles de usuario (exigente, relajado, promedio). El sistema implementa ocho reglas difusas que evalúan clima, aglomeración, atención, tiempo de espera e interés, junto con generación estocástica de condiciones climáticas, niveles de crowding y tiempos de espera. Soporta simulación multi-día con descanso nocturno, acumulación de cansancio y genera visualizaciones automáticas de satisfacción y evolución del cansancio.

El agente de contexto (ContextAgent) empleó Mistral para analizar y mejorar las consultas del usuario, contextualizándolas según el historial de conversación. Implementa análisis conversacional con detección de continuaciones temáticas, extracción inteligente de ubicaciones turísticas en respuestas, evaluación automática de cuándo ofrecer optimización de rutas, y mantenimiento de historial contextual con límites configurables. El agente coordinador (CoordinatorAgent) actúa como núcleo del sistema, orquestando múltiples flujos: análisis contextual de consultas, recuperación RAG, evaluación de utilidad de respuestas usando Mistral, modo especializado de planificación de vacaciones, y detección automática de solicitudes de rutas.

El agente de rutas (RouteAgent) se especializó en la planificación y optimización de itinerarios turísticos utilizando algoritmos de búsqueda y optimización para generar rutas eficientes entre destinos. Este agente consideró múltiples factores como distancias, tiempos de viaje, horarios de apertura de atracciones, preferencias del usuario y restricciones temporales para crear itinerarios personalizados. Integró información geográfica y de transporte para proporcionar recomendaciones de rutas que maximizaran la experiencia turística mientras minimizaban los tiempos de desplazamiento y costos asociados.

El agente guía turístico (TouristGuideAgent) se implementó como agente conversacional especializado en recopilar preferencias de viaje mediante diálogo natural usando Mistral. Implementa fases estructuradas de conversación (saludo, destino, intereses, detalles, resumen), extracción inteligente de preferencias en mensajes del usuario, detección de intenciones para reconocer cuándo proceder con información limitada, y mapeo de intereses a palabras clave para búsquedas especializadas.

## 5 Consideraciones de implementación

Se priorizó la eficiencia y robustez en la recolección de datos mediante crawling paralelo y heurísticas de filtrado semántico. La arquitectura multiagente permitió desacoplar responsabilidades y facilitar la integración de nuevos módulos, como el simulador de turistas y el optimizador ACO. Se consideró la escalabilidad, la actualización dinámica de la base de conocimiento y la adaptabilidad ante consultas fuera del alcance inicial. La integración de técnicas bioinspiradas y de optimización genética respondió a la necesidad de mejorar la relevancia y diversidad de la información recuperada.

## 6 Justificación teórico-práctica

La elección de un enfoque multiagente se fundamentó en la literatura sobre sistemas distribuidos y turismo inteligente [1]. El uso de crawling paralelo y heurísticas avanzadas se justificó por la necesidad de recopilar grandes volúmenes de datos en tiempo reducido y con alta pertinencia [2]. La integración de ACO permitió optimizar la exploración de fuentes novedosas, mientras que el uso de RAG con optimización genética y métricas avanzadas respondió a la tendencia de

combinar recuperación y generación para mejorar la relevancia de las respuestas [3]. La simulación de turistas con lógica difusa permitió modelar la experiencia de usuario de manera realista y adaptable.

## 7 Evaluación cuantitativa y cualitativa

Cuantitativamente, el sistema logró explorar hasta 200 páginas web con una profundidad de 2 y 10 hilos en paralelo, reduciendo el tiempo de crawling a menos de 5 minutos por sesión. El módulo ACO permitió descubrir rutas óptimas de exploración y priorizar fuentes poco cubiertas, incrementando la diversidad de la base de datos. El sistema de simulación de turistas ejecutó hasta 30 réplicas por itinerario, generando análisis estadísticos agregados con intervalos de confianza y métricas de variabilidad. Cualitativamente, se observó una mejora en la pertinencia de las recomendaciones, una interacción fluida entre agentes y una simulación de turistas capaz de reflejar diferencias de perfil y contexto, con visualizaciones automáticas de resultados.

## 8 Autocrítica y propuestas de mejora

Se reconoció que la cobertura de fuentes web podría incrementarse mediante la integración de APIs de plataformas turísticas establecidas como TripAdvisor, Booking.com y Google Places, complementando el crawling web tradicional. La dependencia de modelos generativos externos podría mitigarse implementando sistemas de caché más sofisticados y técnicas de prompt engineering para optimizar las consultas y reducir costos.

La simulación de turistas puede enriquecerse incorporando datos demográficos reales y patrones de comportamiento basados en estudios de mercado turístico. Se propone implementar un sistema de retroalimentación del usuario que permita ajustar automáticamente los parámetros de recomendación y mejorar la personalización sin requerir intervención manual.

Como mejoras técnicas inmediatas, se sugiere implementar un sistema de monitoreo de la calidad de las fuentes web para detectar automáticamente sitios obsoletos o de baja calidad, y desarrollar métricas de evaluación automática de la coherencia y relevancia de las respuestas generadas. Además, se recomienda ampliar la evaluación con usuarios reales mediante encuestas de satisfacción y análisis de patrones de uso para validar la efectividad del sistema en escenarios reales.

## 9 Conclusiones

El desarrollo del sistema multiagente para turismo inteligente ha demostrado exitosamente la viabilidad de integrar múltiples tecnologías avanzadas en una arquitectura cohesiva. La combinación de crawling paralelo, técnicas RAG y

optimización metaheurística ha resultado en un sistema capaz de procesar información turística de manera eficiente y generar recomendaciones personalizadas.

La arquitectura multiagente ha probado ser especialmente valiosa, permitiendo la especialización de funciones y facilitando el mantenimiento del sistema. La integración de técnicas bioinspiradas como el Algoritmo de Colonia de Hormigas y la optimización genética ha aportado valor significativo, mejorando tanto la exploración de fuentes web como la selección de información relevante.

El proyecto ha establecido una base sólida para el desarrollo de sistemas de turismo inteligente, proporcionando un marco de trabajo escalable que puede ser extendido con nuevas funcionalidades. La experiencia adquirida demuestra el potencial de los enfoques híbridos para abordar problemas complejos del mundo real.

## 10 Referencias

### References

1. Buhalis, D.,  
Amaranggana, A.: Smart tourism destinations. In: Information and Communication Technologies in Tourism 2015, pp. 553–564. Springer (2015)
2. Olston, C.,  
Najork, M.: Web crawling. Foundations and Trends in Information Retrieval 4(3), 175–246 (2010)
3. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems 33, 9459–9474 (2020)