

SmartTour Cuba: Sistema Inteligente de Planificación y Recomendación Turística

Sistema de Gestión Turística Avanzada

Proyecto de Investigación en Inteligencia Artificial Aplicada al Turismo

Resumen SmartTour Cuba es un sistema integral de planificación turística que combina técnicas avanzadas de inteligencia artificial, incluyendo algoritmos metaheurísticos (ACO y PSO), sistemas RAG (Retrieval-Augmented Generation), web crawling inteligente y chatbots conversacionales. El sistema proporciona recomendaciones personalizadas, planificación optimizada de itinerarios hoteleros y acceso a información turística actualizada sobre Cuba. Este trabajo presenta la arquitectura completa del sistema, sus módulos funcionales, las tecnologías implementadas y los resultados experimentales obtenidos en diferentes escenarios de uso.

Keywords: Turismo inteligente · Metaheurísticas · RAG · Planificación de itinerarios · Chatbots · Web crawling

1. Introducción

1.1. Contexto y Motivación

El turismo en Cuba representa uno de los sectores económicos más importantes del país, recibiendo millones de visitantes anuales que requieren servicios de planificación eficientes y personalizados. La complejidad de coordinar alojamiento, transporte, actividades culturales y restricciones presupuestarias presenta desafíos significativos tanto para turistas como para operadores turísticos.

SmartTour Cuba surge como respuesta a esta necesidad, integrando tecnologías de vanguardia en inteligencia artificial para ofrecer un sistema completo de planificación turística. El sistema combina múltiples enfoques computacionales: optimización metaheurística para la planificación de itinerarios, procesamiento de lenguaje natural para interacciones conversacionales, y técnicas de recuperación de información para proporcionar datos actualizados y relevantes.

1.2. Alcance del Sistema

El alcance de SmartTour Cuba abarca las siguientes funcionalidades principales:

- **Planificación Optimizada de Itinerarios:** Utiliza algoritmos ACO (Ant Colony Optimization) y PSO (Particle Swarm Optimization) para generar itinerarios hoteleros óptimos considerando presupuesto, calidad de servicios y preferencias del usuario.

- **Sistema RAG Conversacional:** Implementa un sistema de Recuperación Aumentada por Generación que combina bases de conocimiento locales con modelos de lenguaje para responder consultas turísticas específicas.
- **Web Crawling Inteligente:** Extrae información actualizada de sitios web turísticos oficiales, manteniendo una base de datos dinámica de ofertas hoteleras y destinos.
- **Recomendaciones Personalizadas:** Genera sugerencias adaptadas al perfil individual del usuario, considerando preferencias, presupuesto y tipo de experiencia turística deseada.

1.3. Contribuciones Técnicas

Las principales contribuciones técnicas del sistema incluyen:

1. Implementación de algoritmos metaheurísticos optimizados específicamente para planificación hotelera, con parámetros calibrados experimentalmente.
2. Desarrollo de un sistema RAG híbrido que combina conocimiento estructurado y no estructurado para respuestas contextuales.
3. Arquitectura modular que permite escalabilidad y mantenimiento eficiente del sistema.
4. Integración de múltiples fuentes de datos turísticos con procesamiento en tiempo real.

2. Arquitectura del Sistema

2.1. Diseño General

SmartTour Cuba sigue una arquitectura modular basada en microservicios, donde cada componente principal opera de forma independiente pero coordinada. La estructura general se organiza en las siguientes capas:

- **Capa de Presentación:** Interfaces de usuario (Streamlit GUI)
- **Capa de Lógica de Negocio:** Módulos especializados (Planificador, RAG, Chatbot, etc.)
- **Capa de Datos:** Repositorios, crawlers y bases de conocimiento
- **Capa de Servicios:** Conectores externos (Ollama, OpenRouter)

2.2. Tecnologías Principales

El sistema integra las siguientes tecnologías y bibliotecas:

Categoría	Tecnologías
Frontend	Streamlit, HTML/CSS, JavaScript
Backend	Python, FastAPI, Uvicorn
IA/ML	Transformers, FAISS, Sentence-Transformers
Optimización	Optuna, NumPy, SciPy
LLMs	Ollama, OpenRouter API
Web Scraping	Selenium, BeautifulSoup, Requests
Datos	Pandas, JSON, CSV
Vectorización	MiniLM, OpenAI Embeddings

Cuadro 1: Stack tecnológico de SmartTour Cuba

3. Módulo de Planificación de Itinerarios

3.1. Funcionalidad

El módulo de planificación constituye el núcleo del sistema, utilizando algoritmos metaheurísticos para generar itinerarios hoteleros óptimos. El sistema considera múltiples variables: presupuesto disponible, número de noches, destino seleccionado, preferencias de calidad y minimización de cambios de hotel.

3.2. Algoritmos Implementados

Búsqueda en Profundidad (DFS) Implementado como método de referencia para problemas de tamaño pequeño (< 7 noches), garantiza la solución óptima mediante exploración exhaustiva del espacio de búsqueda.

Optimización por Colonia de Hormigas (ACO) El algoritmo ACO simula el comportamiento de hormigas buscando rutas óptimas mediante deposición y evaporación de feromonas. Parámetros optimizados experimentalmente:

- Número de hormigas: 48
- Tasa de evaporación: 0.125
- Factor de influencia de feromonas (α): 1.0
- Factor de información heurística (β): 1.0

La ecuación de probabilidad de selección de hotel es:

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in \text{válidos}} [\tau_{ik}]^{\alpha} \cdot [\eta_{ik}]^{\beta}} \quad (1)$$

Donde τ_{ij} representa la feromona y $\eta_{ij} = \frac{\text{estrellas}}{\text{precio}}$ la información heurística.

Optimización por Enjambre de Partículas (PSO) PSO optimiza posiciones de partículas en el espacio de soluciones mediante actualización de velocidades basada en experiencia personal y colectiva.

Parámetros optimizados:

- Número de partículas: 42
- Coeficiente de inercia (w): 0.7
- Aceleración cognitiva (c_1): 1.5
- Aceleración social (c_2): 1.5

3.3. Función de Fitness

La función objetivo combina tres componentes normalizados:

$$\text{fitness} = \alpha \cdot \text{stars_norm} + \beta \cdot (1 - \text{cost_norm}) + \gamma \cdot (1 - \text{changes_norm}) \quad (2)$$

Donde:

$$\text{stars_norm} = \frac{\sum \text{estrellas}}{\text{noches} \times \text{max_stars}} \quad (3)$$

$$\text{cost_norm} = \min \left(\frac{\text{costo_total}}{\text{presupuesto}}, 1 \right) \quad (4)$$

$$\text{changes_norm} = \frac{\text{cambios_hotel}}{\text{noches} - 1} \quad (5)$$

3.4. Resultados Experimentales

Algoritmo	Tiempo (s)	Fitness Promedio	Óptimo (%)
DFS	0.15	0.95	100
ACO	2.3	0.92	87
PSO	1.8	0.89	82

Cuadro 2: Comparativo de rendimiento de algoritmos (7 noches, 50 hoteles)

3.5. Simulaciones Realizadas

Para validar el desempeño del planificador, se realizaron múltiples simulaciones utilizando el módulo de simulación implementado en `modules/simulation/planner`. Estas simulaciones permitieron comparar los algoritmos bajo diferentes escenarios de usuario, parámetros y restricciones.

Configuración de las Simulaciones Se definieron escenarios variando los siguientes parámetros:

- **Duración del viaje:** 3 a 14 noches
- **Presupuesto:** \$500 a \$3000 USD
- **Destino:** Ejemplo, “La Habana”, “Varadero”, “Trinidad”
- **Preferencias de usuario:** Peso relativo de calidad, costo y cambios de hotel (α , β , γ)
- **Dataset:** CSV de hoteles reales extraído mediante crawling

Ejemplo de Simulación A continuación se muestra un ejemplo de simulación ejecutada para un usuario con las siguientes preferencias:

- **Destino:** La Habana
- **Noches:** 7
- **Presupuesto:** \$1500
- **Parámetros:** $\alpha = 2,5$, $\beta = 1,0$, $\gamma = 1,0$

Los resultados obtenidos fueron:

Método	Latencia (s)	Estrellas	Costo (\$)	Cambios
Clásico (DFS)	0.18	29	1420	2
Metaheurística (ACO)	2.4	28	1390	3
Metaheurística (PSO)	1.9	27	1350	4

Cuadro 3: Resultados de simulación para un escenario típico

Análisis Comparativo Las simulaciones muestran que el método clásico (DFS) es óptimo para instancias pequeñas, pero su tiempo de cómputo crece exponencialmente con el número de noches. Los métodos metaheurísticos (ACO y PSO) ofrecen soluciones cercanas al óptimo con tiempos de respuesta mucho menores en escenarios más grandes.

Se observó que:

- **ACO** tiende a balancear mejor la calidad y el costo, con menor cantidad de cambios de hotel.
- **PSO** explora soluciones más diversas, a veces sacrificando calidad por menor costo.
- El ajuste de los parámetros α , β y γ permite personalizar el itinerario según las preferencias del usuario.

Simulaciones de Sensibilidad Se realizaron simulaciones variando sistemáticamente los parámetros de importancia de calidad, costo y cambios de hotel. Los resultados muestran que:

- Aumentar α prioriza hoteles de mayor calidad, elevando el costo.
- Aumentar β reduce el costo total, pero puede disminuir la calidad.
- Aumentar γ minimiza los cambios de hotel, a costa de menor flexibilidad.

Conclusión de las Simulaciones El simulador permite evaluar rápidamente el impacto de diferentes configuraciones y restricciones, facilitando la selección del algoritmo y parámetros más adecuados para cada perfil de usuario. Los resultados experimentales y de simulación confirman la robustez y flexibilidad del módulo de planificación de SmartTour Cuba.

4. Sistema RAG (Retrieval-Augmented Generation)

4.1. Arquitectura del Sistema RAG

El sistema RAG de SmartTour Cuba combina recuperación de información basada en similitud semántica con generación de texto mediante modelos de lenguaje. La arquitectura incluye:

- **Base de Conocimiento:** Repositorio de información turística sobre Cuba
- **Motor de Vectorización:** MiniLM para generar embeddings semánticos
- **Índice FAISS:** Búsqueda eficiente de documentos similares
- **Generador LLM:** Modelos Ollama locales para respuestas contextuales

4.2. Base de Conocimiento

La base de conocimiento se estructura en categorías temáticas:

- **Historia y Cultura:** Información sobre sitios históricos, personajes relevantes, tradiciones
- **Geografía y Destinos:** Descripciones de provincias, ciudades, atracciones naturales
- **Información Práctica:** Transporte, moneda, requisitos de visa, seguridad
- **Gastronomía:** Platos típicos, restaurantes recomendados, especialidades regionales

4.3. Procesamiento de Archivos ZIM

Para enriquecer la base de conocimiento, el sistema incluye procesamiento de archivos ZIM de Eured:

4.4. Resultados de Evaluación

Métrica	Con RAG	Sin RAG
Precisión de respuestas	89 %	67 %
Relevancia contextual	92 %	45 %
Tiempo de respuesta (s)	3.2	1.8
Satisfacción usuario	4.3/5	3.1/5

Cuadro 4: Evaluación comparativa del sistema RAG

4.5. Simulaciones Realizadas

Para evaluar el comportamiento del módulo RAG, se desarrolló un simulador específico (`modules/simulation/rag`) que permite ejecutar consultas típicas de usuarios sobre la base de conocimiento y comparar el desempeño de diferentes modelos y configuraciones.

Metodología de Simulación Se definió un conjunto de preguntas frecuentes sobre turismo en Cuba (por ejemplo: “¿Cuáles son los lugares turísticos más importantes de Santiago de Cuba?”, “¿Qué historia tiene el Malecón de La Habana?”, “¿Dónde puedo probar el mejor café cubano?”). Estas consultas se procesaron automáticamente usando el simulador, tanto con RAG activado como desactivado, y con distintos modelos de lenguaje (por ejemplo, OpenHermes y Gemma2).

Parámetros Evaluados Las simulaciones midieron:

- **Latencia de respuesta:** Tiempo promedio de generación de respuesta.
- **Longitud de respuesta:** Número de palabras generadas.
- **Fuente de información:** Si la respuesta provino de la base de conocimiento, Wikipedia/Ecured o solo del modelo.
- **Calidad percibida:** Evaluación manual de relevancia y precisión.

Resultados de Simulación A continuación se muestra un resumen de los resultados obtenidos en una simulación típica con 15 consultas:

Configuración	Latencia Prom. (s)	Longitud Prom. (palabras)	Fuente Principal
RAG + OpenHermes	2.8	110	Base de Conocimiento
RAG + Gemma2	3.1	105	Base de Conocimiento
Sin RAG + OpenHermes	1.7	85	Modelo LLM

Cuadro 5: Resultados de simulación automática sobre el módulo RAG

Las simulaciones muestran que el uso de RAG incrementa la relevancia y precisión de las respuestas, aunque con un ligero aumento en la latencia. Además, la integración de la base de conocimiento permite respuestas más contextualizadas y extensas.

Conclusión de las Simulaciones El simulador facilita la evaluación sistemática del módulo RAG ante diferentes escenarios y modelos, permitiendo ajustar parámetros y validar mejoras en la recuperación y generación de información turística relevante para el usuario.

5. Chatbot Conversacional

5.1. Funcionalidad

El chatbot de SmartTour Cuba utiliza modelos de lenguaje avanzados para mantener conversaciones naturales con usuarios, extrayendo información de perfiles turísticos y proporcionando recomendaciones personalizadas.

5.2. Extracción de Información de Usuario

El sistema utiliza esquemas JSON para validar y estructurar la información extraída:

5.3. Integración con Modelos de Lenguaje

El chatbot puede utilizar diferentes proveedores de LLM:

- **OpenRouter:** Acceso a modelos como Mistral-7B, GPT-3.5, Claude
- **Ollama Local:** Modelos ejecutados localmente para privacidad
- **Fallback:** Sistema de respaldo en caso de fallas de conectividad

5.4. Simulaciones Realizadas

Para validar la capacidad del chatbot en la extracción de información de perfiles turísticos, se desarrolló un simulador (`modules/simulation/chatbot`) que automatiza conversaciones con perfiles de usuario generados aleatoriamente y mide la calidad de la extracción.

Metodología de Simulación El simulador ejecuta múltiples conversaciones, donde el chatbot interactúa con perfiles sintéticos que contienen campos como nombre, edad, intereses, destinos, presupuesto, duración del viaje y preferencias adicionales. En cada paso, el bot solicita información relevante y se registra la respuesta, el valor extraído y la latencia.

Métricas Evaluadas Las simulaciones permiten calcular:

- **Similitud de extracción:** Se utiliza la similitud de coseno entre el perfil original y el perfil extraído por el bot.
- **Latencia promedio:** Tiempo medio de respuesta por interacción.
- **Tasa de extracción completa:** Porcentaje de campos correctamente identificados.

Resultados de Simulación En una evaluación típica con 30 perfiles aleatorios:

- **Similitud promedio:** 0.89 (coseno)
- **Latencia promedio:** 1.2 segundos por interacción
- **Extracción completa:** 93 % de los campos correctamente identificados

Métrica	Valor Promedio	Desviación Estándar
Similitud de coseno	0.89	0.04
Latencia (s)	1.2	0.3
Extracción completa (%)	93	5

Cuadro 6: Resultados de simulación del chatbot conversacional

Conclusión de las Simulaciones Las simulaciones demuestran que el chatbot es capaz de extraer información relevante de perfiles turísticos con alta precisión y eficiencia, validando su utilidad para la personalización de recomendaciones y planificación en SmartTour Cuba.

6. Web Crawler Inteligente

6.1. Objetivos del Crawler

El módulo de web crawling mantiene actualizada la base de datos de ofertas hoteleras mediante extracción automatizada de información del sitio oficial cuba.travel.

6.2. Configuración y Cumplimiento

El crawler respeta estrictamente las directrices de robots.txt:

- **User-Agent:** Identificación clara del bot
- **Crawl Delay:** Pausa entre solicitudes para minimizar carga del servidor
- **Rutas Prohibidas:** Exclusión de directorios administrativos y privados
- **Límites de Tasa:** Control de frecuencia de solicitudes

6.3. Estructura de Datos Extraídos

Campo	Descripción
name	Nombre del hotel
stars	Clasificación por estrellas (1-5)
address	Dirección física
cadena	Cadena hotelera
tarifa	Tipo de plan (Todo Incluido, etc.)
price	Precio por noche
hotel_url	URL de detalles

Cuadro 7: Estructura de datos de ofertas hoteleras

7. Sistema de Recomendaciones

7.1. Algoritmo de Recomendación

El sistema de recomendaciones utiliza filtrado colaborativo y basado en contenido para sugerir destinos y actividades personalizadas.

7.2. Implementación Actual

El módulo de recomendación se basa en técnicas de representación vectorial de perfiles de usuario y ofertas turísticas. Para ello, se emplea el modelo `all-MiniLM-L6-v2` de *Sentence Transformers* para generar embeddings semánticos tanto del perfil del usuario como de las ofertas disponibles.

- **Perfil de Usuario:** Se representa como un vector generado a partir de la descripción plana y recursiva de todos los campos del perfil (nombre, intereses, destinos, presupuesto, etc.), permitiendo capturar tanto información explícita como implícita.
- **Ofertas:** Cada oferta turística (hotel, actividad, destino) se vectoriza a partir de sus atributos relevantes (nombre, descripción, servicios, ubicación, etc.).
- **Similitud:** La recomendación se realiza calculando la similitud de coseno entre el vector del usuario y los vectores de las ofertas, retornando las más afines.
- **Carga de Ofertas:** El sistema soporta la carga de ofertas desde archivos JSON y CSV, procesando automáticamente los datos y generando los embeddings correspondientes.

7.3. Factores de Recomendación

- **Perfil de Usuario:** Edad, intereses, presupuesto, duración de viaje
- **Características de Destino:** Tipo de turismo, clima, actividades disponibles
- **Restricciones:** Médicas, dietéticas, de accesibilidad

7.4. Visualización de Resultados

Las ofertas recomendadas se presentan al usuario con un formato amigable, mostrando los atributos clave de cada opción. El sistema permite adaptar la visualización según el contexto y las preferencias del usuario.

7.5. Ventajas y Limitaciones

- **Ventajas:** El enfoque vectorial permite recomendaciones personalizadas incluso ante perfiles y ofertas heterogéneas, y es escalable a grandes volúmenes de datos.
- **Limitaciones:** La calidad de la recomendación depende de la riqueza semántica de los datos y de la capacidad del modelo de embeddings para capturar matices relevantes.

8. Interfaz de Usuario

8.1. Arquitectura de Frontend

SmartTour Cuba utiliza Streamlit para crear una interfaz web moderna y responsiva. La aplicación principal ([‘main.py’](file:///c %3A/Users/HP/Desktop/IA/Proyecto/tour-guide-cuba/GUI/main.py)) implementa un sistema de navegación modular.

8.2. Características de la Interfaz

- **Diseño Responsivo:** Adaptable a diferentes tamaños de pantalla
- **Navegación Intuitiva:** Menú principal con iconografía clara
- **Chat Interactivo:** Interfaz tipo WhatsApp para conversaciones
- **Visualizaciones Dinámicas:** Gráficos y mapas interactivos
- **Controles Modernos:** Elementos UI con estilo contemporáneo

8.3. Módulos de Interfaz

Módulo	Funcionalidad Principal
Chatbot	Conversación con extracción de datos
Recomendador	Sugerencias personalizadas
Planificador	Generación de itinerarios optimizados
Recuperador	Consultas RAG sobre información turística
Base de Conocimiento	Gestión de información turística
Usuario	Perfil y preferencias
Exportar	Descarga de itinerarios

Cuadro 8: Módulos de la interfaz de usuario

9. Integración del Sistema

9.1. Flujo de Trabajo Completo

El sistema integrado de SmartTour Cuba opera mediante el siguiente flujo:

1. **Adquisición de Datos:** El crawler actualiza periódicamente la base de datos de hoteles
2. **Interacción Inicial:** El usuario interactúa con el chatbot para definir preferencias
3. **Extracción de Perfil:** El sistema extrae y valida información del usuario
4. **Recomendaciones:** Se generan sugerencias basadas en el perfil
5. **Planificación:** Los algoritmos metaheurísticos optimizan itinerarios
6. **Consultas RAG:** El usuario puede hacer preguntas específicas sobre destinos
7. **Exportación:** El itinerario final se presenta en formato descargable

10. Resultados Experimentales

10.1. Evaluación de Rendimiento

11. Conclusiones y Trabajo Futuro

11.1. Logros Principales

SmartTour Cuba representa una solución integral para la planificación turística inteligente, demostrando la viabilidad de combinar múltiples técnicas de IA en un sistema cohesivo. Los principales logros incluyen:

1. **Optimización Efectiva:** Los algoritmos metaheurísticos muestran resultados consistentes con fitness promedio superior al 85 %
2. **Interacción Natural:** El sistema RAG proporciona respuestas contextuales con 89 % de precisión
3. **Escalabilidad:** Arquitectura modular que soporta crecimiento incremental
4. **Usabilidad:** Interfaz intuitiva con alta satisfacción del usuario (4.2/5)

11.2. Limitaciones Identificadas

- **Dependencia de Datos:** La calidad de recomendaciones depende de la actualización constante de información
- **Escalabilidad de LLM:** Los modelos de lenguaje grandes requieren recursos computacionales significativos
- **Personalización:** El sistema requiere interacción mínima para generar perfiles efectivos

11.3. Trabajo Futuro

Las siguientes mejoras están planificadas para versiones futuras:

1. **Aprendizaje Adaptativo:** Implementación de algoritmos de aprendizaje por refuerzo para optimización continua
2. **Integración IoT:** Incorporación de datos en tiempo real de sensores y dispositivos
3. **Realidad Aumentada:** Desarrollo de funcionalidades AR para guías interactivas
4. **Blockchain:** Sistema de reputación descentralizado para hoteles y servicios
5. **Análisis Predictivo:** Modelos de predicción de demanda y precios dinámicos

12. Agradecimientos

Este trabajo fue desarrollado como parte de un proyecto de investigación en inteligencia artificial aplicada al sector turístico. Agradecemos a las instituciones y organizaciones que proporcionaron datos y apoyo para el desarrollo de este sistema.