

GPTur

Claudia Hernández Pérez, Joel Aparicio Tamayo, and Kendry Javier Del Pino Barbosa

Facultad de Matemática y Computación, La Habana, Cuba

Resumen. GPTur es un sistema diseñado con una arquitectura de Generación Aumentada por Recuperación (RAG) con agentes especializados para cada funcionalidad y una base de datos vectorial para la recuperación de la información. Posee, además, varios agentes encargados de tópicos específicos como gastronomía, vida nocturna, alojamiento y lugares históricos, que comparten datos en una pizarra central. Estos agentes están dirigidos por un guía y un planificador. El primero se encarga de responder consultas mientras el segundo utiliza algoritmos para planear itinerarios ajustados a presupuestos. Este agente resuelve un problema de satisfacción de restricciones utilizando la técnica de precocido simulado. También, el programa preprocesa la consulta para obtener ciertas entidades, palabras claves y sentimientos que puedan influir en los resultados esperados. Todo eso, permite conformar una respuesta acertada, acorde a las necesidades del usuario, validadas experimentalmente.

Palabras clave: RAG, agente, problema de satisfacción de restricciones, precocido simulado

1 Introducción

La industria turística enfrenta desafíos únicos en contextos con limitaciones en la infraestructura digital y actualización constante de servicios. Cuba, como destino turístico emblemático, requiere soluciones innovadoras que superen estas barreras mediante inteligencia artificial. *GPTur*, un sistema de agentes inteligentes que funciona como guía turístico virtual, está diseñado específicamente para ser una alternativa viable en ese escenario.

La aplicación consiste en un chatbot capaz de responder consultas sobre destinos turísticos cubanos y sus peculiaridades, así como de generar itinerarios para *tours* ajustándose a un presupuesto. El sistema está programado en *Python*, aprovechando la comodidad del lenguaje y la diversidad de bibliotecas útiles que ofrece. La interfaz está diseñada con *streamlit* (framework de *Python* para *frontend*), pues ofrece un estilo sencillo, moderno y fácil de programar para entornos web. El código fuente de la aplicación se puede encontrar en la plataforma GitHub: <https://github.com/ClaudiaHdezPerez/GPTur>.

2 Detalles de Implementación

En esta sección, se abordarán los distintos temas asociados a la implementación de la solución propuesta, que incluyen la arquitectura RAG y multiagente, funcionamiento del *crawler* automatizado y recopilación del corpus, base de datos vactorial y representación semántica del conocimiento mediante *embeddings*, procesamiento de lenguaje natural, algoritmos basados en metaheurísticas, entre otros.

2.1 Arquitectura RAG (*Generación Aumentada por Recuperación*)

Un sistema de Generación Aumentada por Recuperación combina generación de texto con recuperación de información para mejorar la calidad, precisión y actualidad de las respuestas obtenidas de los modelos de lenguaje. Está compuesto por una entrada, un recuperador, un generador y una salida. La implementación propuesta es bastante sencilla, ajustada a la definición:

1. Entrada: constituida por la consulta del usuario en lenguaje natural.
2. Recuperador: agente que recibe la consulta y le pide a la base de datos vectorial que recupere la información relacionada.
3. Generador: agente que utiliza la consulta y la información recuperada para decidir si la respuesta debe ser generada por el guía o el planificador. Para tomar esa decisión le pide a un modelo de lenguaje que detecte la intención del usuario.
4. Salida: respuesta generada por el agente inteligente encargado de elaborarla.

2.2 Arquitectura multiagente

Un sistema multiagente está constituido por un conjunto de agentes, cada uno con determinadas capacidades y recursos para la solución de problemas inherentemente distribuidos. Los agentes interactúan entre ellos, determinan y coordinan tareas a realizar según sus capacidades y recursos.

GPTur cuenta con dos agentes fundamentales: el guía y el planificador. Existen, además, otros especializados en distintos tópicos: gastronomía, vida nocturna, alojamiento y lugares históricos. Todos ellos

Información adicional Encabezados de tercer nivel en 10pt negrita.

3 Validación y Experimentación

Esta sección fue un elemento esencial en la etapa de desarrollo y, en muchos casos, influyó en la toma de decisiones. Se realizaron varios experimentos, que incluyen valoraciones de la respuesta del sistema y evaluación de los algoritmos de metaheurísticas.

3.1 Respuesta del Sistema

Durante las distintas etapas de desarrollo, fue importante evaluar el comportamiento de la aplicación. Para estos experimentos se utilizó un modelo de lenguaje para generar 30 preguntas relacionadas con el corpus y obtener las respuestas de GPTur automáticamente. Luego, se creó una consulta para un modelo de lenguaje pidiendo una evaluación entre 0 y 5 puntos de la respuesta del sistema en cuanto a:

- Precisión de la información
- Coherencia y relevancia
- Cobertura del tema
- Calidad de las recomendaciones
- Ajuste a la pregunta realizada

Después de obtener los resultados de la muestra, se hacía necesario evaluar muestras mucho más grandes, pero eso requería demasiado cómputo, así que como alternativa se utilizó la técnica conocida como *bootstrapping*, pudiendo estudiar muestras de tamaño 300 con 1000 simulaciones. A continuación se muestran los resultados obtenidos en las distintas etapas.

Sistema con Agentes Guía y Planificador Solamente La Tabla [Tab. 1](#) resume las métricas clave obtenidas:

Table 1. Resultados del análisis de bootstrapping (IC 95%)

Métrica	Tamaño=30	Tamaño=300
Puntuación Media	3.7	3.7013...
Desviación Estándar	0.4582...	0.0261...

Remark 1. Se obtuvo un intervalo de confianza de $[3.6499..., 3.75]$, lo que indica unos resultados aceptables. La Figura [Fig. 1](#) muestra la distribución bootstrap de la puntuación media y la figura [Fig. 2](#) muestra un mapa de calor con las puntuaciones otorgadas a las respuestas del sistema a las 30 preguntas iniciales.

Sistema con Agentes Especializados La Tabla [Tab. 2](#) resume las métricas clave obtenidas:

Table 2. Resultados del análisis de bootstrapping (IC 95%)

Métrica	Tamaño=30	Tamaño=300
Puntuación Media	4.1333...	4.1329...
Desviación Estándar	0.6182...	0.0349...

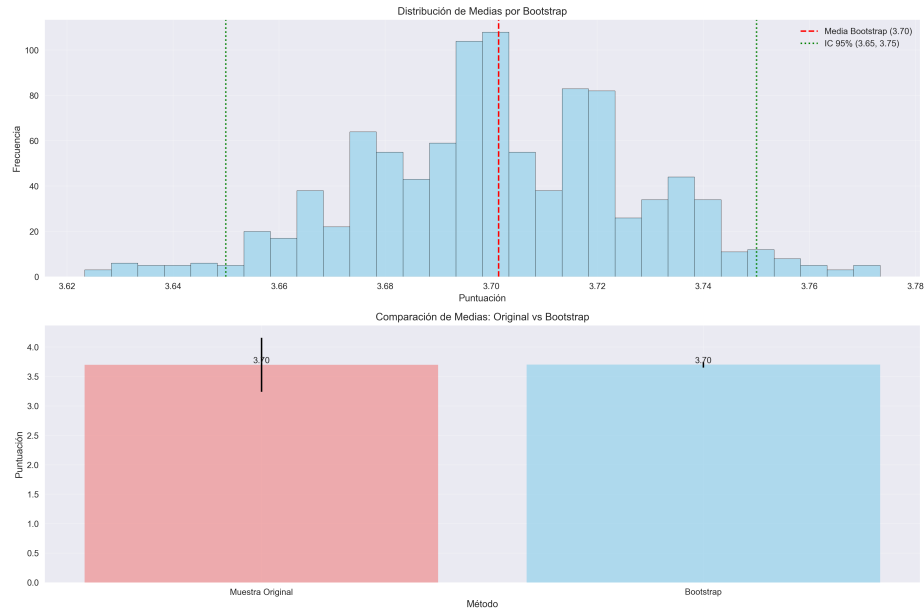


Fig. 1. Distribución bootstrap de la puntuación media ($N = 1000$). La línea discontinua roja indica la media bootstrap (3.70)

Remark 2. Se obtuvo un intervalo de confianza de $[4.0666..., 4.1966...]$, lo que indica unos resultados bastante buenos. La Figura Fig. 3 muestra la distribución bootstrap de la puntuación media y la figura Fig. 4 muestra un mapa de calor con las puntuaciones otorgadas a las respuestas del sistema a las 30 preguntas iniciales.

En resumen, se puede observar que al añadir agentes especializados al sistema, mejoró la respuesta del modelo, lo cual demostró que la decisión tomada estuvo acertada.

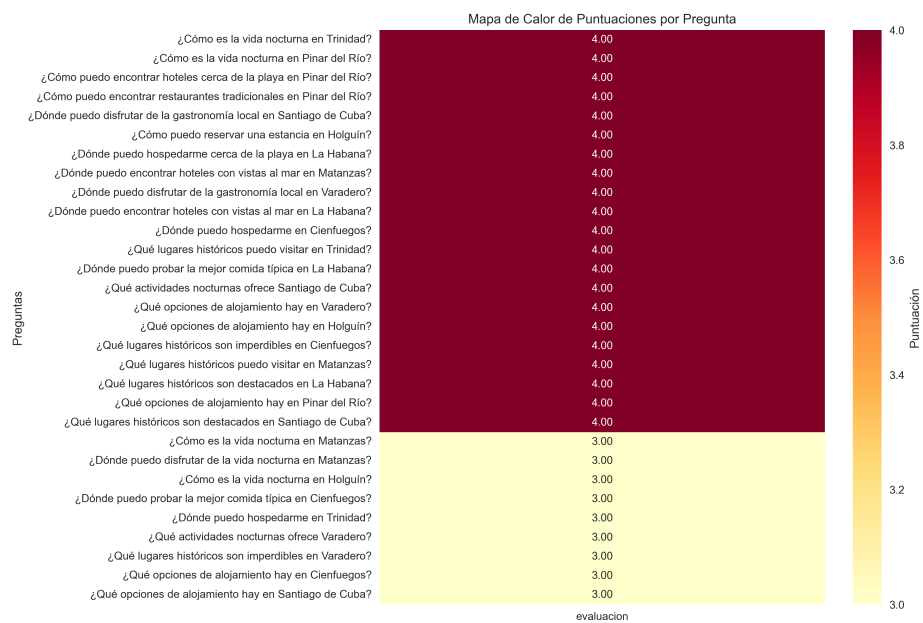


Fig. 2. Evaluaciones obtenidas para la respuesta a cada pregunta en la muestra más pequeña (tamaño 30)

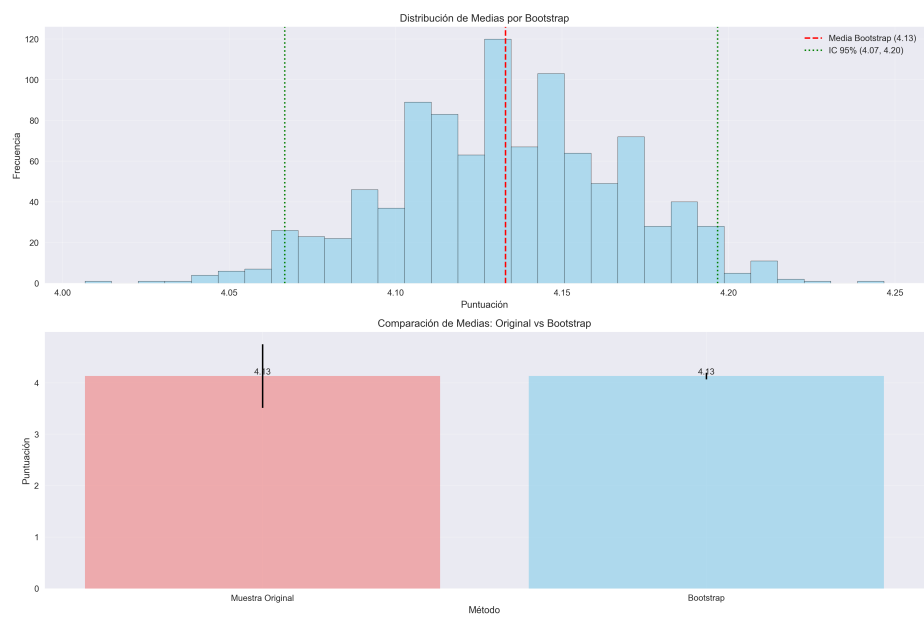


Fig. 3. Distribución bootstrap de la puntuación media ($N = 1000$). La línea discontinua roja indica la media bootstrap (4.13)



Fig. 4. Evaluaciones obtenidas para la respuesta a cada pregunta en la muestra más pequeña (tamaño 30)