A picture containing text

Description automatically generated

Proyecto Integrador

Dra. Grettel Barceló Alonso

Dr. Luis Eduardo Falcón Morales

Avance 6 – Conclusiones Clave

Junio 09 2024

**Equipo 37**

A00759664 - Joel Orlando Hernández Ramos

A01793486 - Juan Carlos Alvarado Carricarte

A00260430 - Juan Carlos Romo Cárdenas

Contenido

[Introducción 3](#_Toc168866553)

[Implementación del Modelo Seleccionado 3](#_Toc168866554)

[Recomendaciones de Preprocesamiento 3](#_Toc168866555)

[Recomendaciones de Implementación 4](#_Toc168866556)

[Plataforma de Servicios en la Nube 6](#_Toc168866557)

[Referencias 8](#_Toc168866558)

# Introducción

En la era digital actual, la personalización y eficiencia en la planificación de viajes se ha convertido en una prioridad para los consumidores globales. Respondiendo a esta demanda, nuestro proyecto ha desarrollado un asistente de inteligencia artificial avanzado, integrado en una innovadora plataforma de traveltech, diseñada para transformar la experiencia de selección de hoteles y destinos. Este sistema inteligente, apoyado por la robusta "knowledge base" de Amazon Bedrock, incorpora información detallada sobre más de 2 millones de hoteles.

El objetivo principal del proyecto ha sido hacer que la planificación de viajes sea más intuitiva y basada en datos confiables, mejorando significativamente la satisfacción del cliente y optimizando los tiempos de búsqueda y selección de alojamientos y destinos. Durante las fases iniciales, nuestro equipo se centró en el preprocesamiento de datos y el desarrollo de una base de conocimientos sólida, estableciendo un modelo base que sirvió como punto de partida para futuras iteraciones y mejoras.

Con el propósito de alcanzar una mayor eficiencia, nuestro equipo desarrolló tres modelos alternativos al modelo de base. Estos modelos fueron evaluados cualitativamente para determinar su eficacia y precisión en la generación de recomendaciones. El modelo que destacó por sus capacidades avanzadas, empleando técnicas de generación de texto asistida por recuperación de información, fue seleccionado para desarrollo y evaluación adicionales. Este modelo no solo mejoró la calidad de las recomendaciones, sino que también proporcionó una base más sólida para las interacciones intuitivas y personalizadas con los usuarios.

En esta fase final del proyecto, hemos aplicado métricas cuantitativas para evaluar exhaustivamente el rendimiento de nuestro modelo seleccionado. Este informe presenta los hallazgos de estas evaluaciones y del desarrollo de este y los otros modelos, subrayando cómo nuestro asistente inteligente cumple con los objetivos propuestos.

# Implementación del Modelo Seleccionado

El modelo seleccionado ya fue presentado a los patrocinadores del proyecto en Volindo, y es su opinión que esta versión del modelo esta lista para su introducción controlada, siguiendo una estrategia de Canario. De acuerdo a esta estrategia, se hará una introducción controlada de un chatbot a un 10% de los usuarios y se evaluará su desempeño. Dependiendo de los resultados de la evaluación se pueden agregar más usuarios y/o hacer ajustes al modelo o a los datos. De esta manera se puede desplegar la solución de una manera ágil y se puede mejorar el producto de una manera iterativa.

Los hallazgos y recomendaciones que siguen están encaminados a soportar este objetivo.

## Recomendaciones de Preprocesamiento

Volindo cuenta con una base de datos de más de un millón de registros con información de hoteles. Esta información incluye nombre del hotel, dirección, ciudad y país y nivel de estrellas entre otros datos. Para la construcción del prototipo solo se usaron registros de hoteles con un nivel de 4 o más estrellas, lo que redujo el tamaño de la base de conocimiento a un poco más de 145 mil registros, lo que limitó la calidad de las respuestas generadas en términos de opciones presentadas.

Para mejorar el desempeño del modelo se recomienda lo siguiente para la etapa de procesamiento:

* Seleccionar uno de los tres conjuntos de datos disponibles para actualizar la base de conocimiento en Pinecone:
  + Conjunto de datos con todos los hoteles disponibles con un poco más de un millón de registros
  + Conjunto de datos con los hoteles con 2 o más estrellas con un poco más de 600 mil registros
  + Conjunto de datos con los hoteles con 3 o más estrellas con un poco más de 500 mil registros

Está decisión es importante y debe de estar alineada con los principios de calidad de información y servicio de Volindo. Si se usan el conjunto de datos de 2 o más estrellas, o 3 o más estrellas, se pueden restringir las opciones que el modelo puede proporcionar a los usuarios. Por otro lado, si se selecciona el conjunto de datos con todos los hoteles se pueden incluir opciones de alojamiento de baja calidad. Las implicaciones en términos de costo de esta decisión no son significativas y están principalmente limitados a la base de datos de vectores de Pinecone.

* Agregar el nombre del hotel y la ciudad y el país donde se localiza a la descripción. Estos datos ayudan a generar mejores respuestas. Como se tiene que agregar el resto de los datos este proceso forma parte de las actividades de preprocesamiento que se tienen que realizar.
* Actualizar y preprocesar la base de datos de hoteles de manera periódica para mantener la relevancia del modelo.

## Recomendaciones de Implementación

Mientras que el modelo se desempeñó bien en la evaluación con preguntas sintéticas y en la evaluación de las preguntas de referencia también se notó que:

* Preguntas con puntos de referencia o características preferidas producen mejores resultados
* Algunas veces mucha información acerca de la localización preferida puede afectar negativamente la selección de opciones. Ciudad y país parece ser suficiente

Para mejorar el desempeño del modelo se recomienda lo siguiente para la etapa de implementación:

* Proveer una guía básica de uso. Aunque solo hay dos recomendaciones de momento la guía se puede extender con recomendaciones de los usuarios
* Proveer un mecanismo de retroalimentación para que el usuario pueda dar recomendaciones y una calificación del 1 al 5, tal y como se hizo en el análisis cuantitativo del modelo en el Avance 4 del proyecto
* Capturar las preguntas proveídas por el usuario y los contextos recabados para hacer una evaluación de dispersión del espacio de embeddings, como se hizo en el Avance 5. Las preguntas del usuario deben de pasarse por un proceso de anonimato para remover datos personales. La evaluación de la dispersión ayudara a mejorar la base de conocimiento
* Las preguntas capturadas se pueden usar para ingeniería de prompts. El modelo pasa una serie de instrucciones al modelo generativo y estas se pueden mejorar con características comunes observadas en las preguntas hechas
* Hay dos versiones de la librería de LangChain, una para Python y una para JavaScript/Node.JS. Volindo puede hacer uso de cualquiera de ellas de acuerdo con sus estándares de programación
* Mover la licencia de Pinecone al nivel Standard.

# Plataforma de Servicios en la Nube

Volindo usa actualmente Amazon Web Services para desplegar todos los recursos de la aplicación de la empresa, y esta es la razón principal por las que la compañía decidió desarrollar este proyecto en AWS Bedrock Titan y Knowledge Base. Mantener todos los componentes en un solo proveedor de servicios en la nube simplifica administración de recursos y cuentas.

Aunque Volindo no está cerrado a usar otros proveedores de servicios en la nube, las ventajas deberían de poder justificar el trabajo de migración, conexión entre redes y mantenimiento de más de un proveedor.

Siguiendo la estrategia delineada por Volindo la implementación de este modelo se hará en Amazon Web Services.

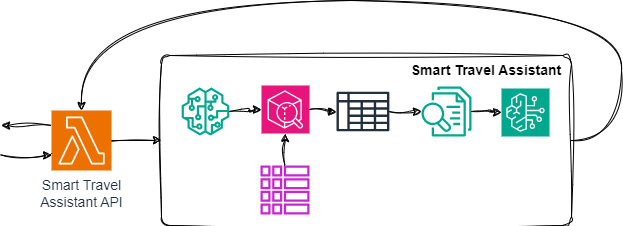


Ilustración 1 - Implementación del Modelo en AWS

Sin embargo, y a manera comparativa, se hará un breve análisis de la posible implementación del asistente de viajes en Azure y Google Cloud.

| **Proveedor** | **Componentes Básicos** | **Comentarios** |
| --- | --- | --- |
| Amazon Web Services | Bedrock Titan, Bedrock Knowledge Base, OpenSearch o Aurora Data Base | La configuración básica de Bedrock para Knowledge Base usa AWS OpenSearch como buscador almacén y buscador de la base de conocimiento. Esta es una configuración cara en el corto plazo. Por esta razón se decidió usar Pinecone como base de datos para Knowledge Base.  Mientras que Bedrock ofrece APIs para ejecutar la cadena de RAG, durante el desarrollo del proyecto se encontró que el uso de LangChain como LLM Pipeline mejoró el desempeño de la solución (AWS, s.f., LangChain, s.f.). |
| Azure | Azure OpenAI, Azure AI Search, Pipeline LLM | Azure ofrece AI Search como una base de datos de vectores, con un esquema de suscripción basado en rangos de espacio usado.  Para implementar RAG es necesario es necesario usar un pipeline LLM para coordinar la recuperación de información, la formulación del prompt y la generación de la respuesta. Esto permite usar otra base de datos de vectores si así se prefiere. |
| Google Cloud | Vertex AI, Vertex AI Search, Pipeline LLM | Como en el caso de AWS Bedrock, GCP ofrece Vertex AI Search como buscador de las fuentes de la base de conocimiento, que pueden ser fuentes estructuradas o no estructuradas.  Como en el caso de Azure, GCP requiere de un pipeline de LLM para coordinar la recuperación de información, la formulación del prompt y la generación de la respuesta. Esto permite usar otra base de datos de vectores si así se prefiere. |

Como se puede apreciar en la comparación anterior, un pipeline de LLM como LangChaing juega un papel crucial en la implementación de un sistema de generación de respuestas asistidas por recuperación de información. En este sentido, considerando que la integración de este Pipeline es prácticamente el mismo en Amazon Bedrock que en los otros servicios en la nube analizados, no se vislumbra una ventaja competitiva clara que justifique el esfuerzo de migrar a otro servicio en la nube.

La implementación de nuestro modelo ha demostrado ser eficaz dentro del ecosistema de Amazon, y replicar este proceso en otra plataforma no solo implicaría una duplicación de esfuerzos, sino que también conllevaría riesgos asociados al cambio y la necesidad de una nueva curva de aprendizaje para el equipo. Esta decisión se alinea con nuestra estrategia de optimización de recursos y minimización de riesgos, asegurando así la continuidad y la eficiencia operativa sin sacrificar la calidad del servicio ofrecido.

# Referencias

AWS. (s.f.). Foundation Model for Generative AI - Amazon Titan. Amazon Web Services, Inc. <https://aws.amazon.com/bedrock/titan/>

AWS. (s.f.). ‌Retrieval Augmented Generation (RAG) - Knowledge Bases for Amazon Bedrock. Amazon Web Services, Inc. <https://aws.amazon.com/bedrock/knowledge-bases/>

‌Steen, H., Walhin, D. (2023, 19 de octubre). RAG and generative AI - Azure Cognitive Search. learn.microsoft.com. <https://learn.microsoft.com/en-us/azure/search/retrieval-augmented-generation-overview>

GCP. (s.f.). ‌Vertex AI APIs for building search and RAG experiences. Google Cloud. Recuperado 3 de junio, 2024, de <https://cloud.google.com/generative-ai-app-builder/docs/builder-apis?_gl=1>

Pinecone. (s.f.). Integrations. Pinecone Docs. Recuperado 3 de junio, 2024, de <https://docs.pinecone.io/integrations/overview>

‌RAGAS. (s.f.). Introduction Ragas. docs.ragas.io. Recuperado 3 de junio, 2024, de <https://docs.ragas.io/en/stable/>