

SMART TRAVEL ASSISTANT

[HTTP://WWW.VOLINDO.COM](http://www.volindo.com)



TABLA DE CONTENIDO

- Objetivo del Proyecto
- Equipo de Trabajo
- Resumen de Ajustes a los Modelos
- Ajustes a la Etapa de Pre-procesamiento e Indexado
 - Ingestión de Archivos por País - Referencia
 - Ingestión de Archivos por Hotel - Nuevo
- Ajustes a la Etapa de Generación
 - Generación usando SDK de Python de AWS para Bedrock
 - Generación usando SDK de Python de LangChain para AWS Bedrock
 - Ajustes a los parámetros de recuperación de información
 - Ingeniería de Prompt
- Resultados
- Pasos Siguientes

OBJETIVO DEL PROYECTO

El objetivo principal de este proyecto es transformar la experiencia de planificación de viajes, haciendo que sea más intuitiva, personalizada y respaldada por datos. Las metas específicas incluyen mejorar la satisfacción del cliente mediante recomendaciones precisas y personalizadas, reducir el tiempo necesario para planificar un viaje y aumentar la eficiencia del proceso de selección de destinos y hoteles.



OBJETIVO DEL PROYECTO

Este proyecto se propone desarrollar un asistente de inteligencia artificial integrado en una plataforma de traveltech, que facilita la elección personalizada de hoteles y destinos para los usuarios. Utilizando la función "knowledge base" de Amazon Bedrock, el sistema integrará una extensa base de datos con información detallada sobre más de 2 millones de hoteles.



EL EQUIPO DE TRABAJO

Los integrantes del equipo para este proyecto son:

Joel Orlando Hernández Ramos

Juan Carlos Alvarado Carricarte

Juan Carlos Romo Cárdenas



A hiker in a dark jacket and cap, holding a trekking pole, stands on a rocky mountain trail. They are looking out over a vast, hazy mountain range under a clear blue sky. The foreground shows dry grass and rocks, while the background features layers of distant mountain peaks.

AJUSTES A LOS MODELOS

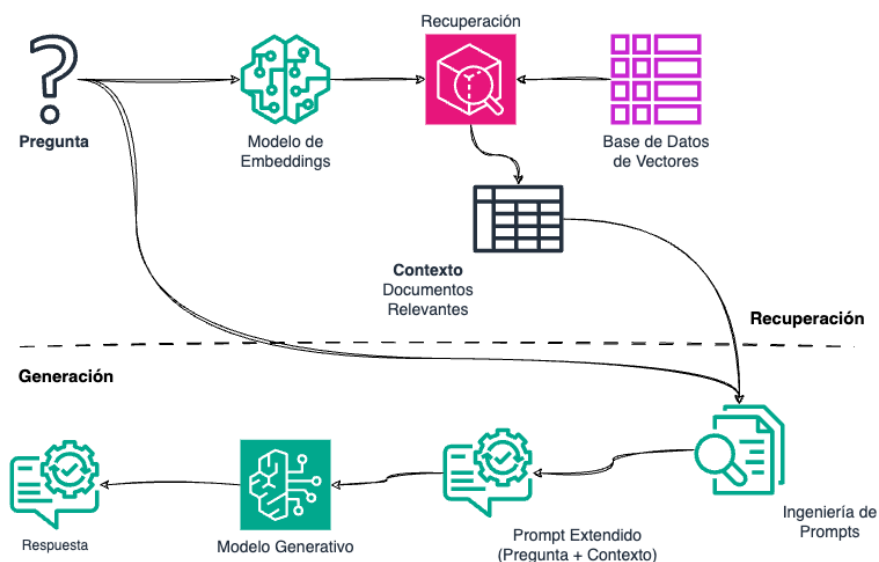
Retrieval Augmented Generation (RAG) - Conceptos

La técnica RAG hace uso de una base de conocimiento para extender o aumentar la pregunta, petición o sugerencia proveída por el usuario para mejorar la respuesta del modelo generativo (AWS, s.f.).



Pre-Procesamiento e Indexado de Documentos

El objetivo de esta etapa es extraer información de las fuentes de conocimiento y dividirlos en pedazos uniformes de contenido con metadatos. Los pedazos son procesados por un modelo de embeddings y almacenados, con sus metadatos asociados en una base de datos de vectores.



Recuperación de Documentos y Generación de Respuesta

El objetivo de esta etapa es recuperar información relevante de la base de datos de vectores en base a la pregunta presentada. La información recuperada es usada como contexto a la pregunta y ambas son presentadas al modelo generativo para producir la respuesta.

RESUMEN DE AJUSTES

En metodología RAG existen dos facetas operativas y hay ajustes pertinentes para cada una de ellas (Boudier, 2024):

- Pre-procesamiento e Indexado
 - Limpieza de Datos e Ingeniería de Características
 - Tamaño de los pedazos, o *chunks* a procesar
- Generación
 - Ajustes a los parámetros de recuperación
 - Ingeniería de Prompts

En este proyecto se realizaron todos estos ajustes.

AJUSTES A LA ETAPA DE PRE- PROCESAMIENTO E INDEXADO



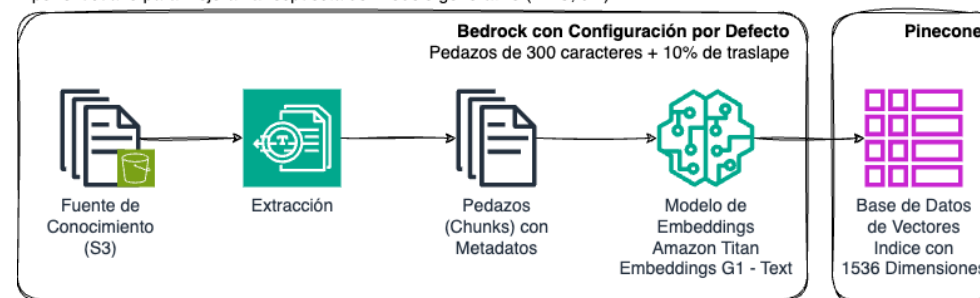
DATOS DE REFERENCIA

Para la creación de la versión 1.00 de la base de conocimiento se hizo lo siguiente:

- Limpieza de datos
- Ingeniería de características
- Reducción del conjunto de datos para incluir únicamente hoteles con un nivel de 4 estrellas o más
- Generación de archivos maestros con datos preparados y archivos con datos para la ingestión por país
- Ingestión de los archivos por país en segmentos de 300 caracteres
- Generación de embeddings con 1536 dimensiones
- Almacenamiento de los embeddings en una base de datos de vectores.

Retrieval Augmented Generation (RAG) - KB V1.00

La técnica RAG hace uso de una base de conocimiento para extender o aumentar la pregunta, petición o sugerencia proveída por el usuario para mejorar la respuesta del modelo generativo (AWS, s.f.).

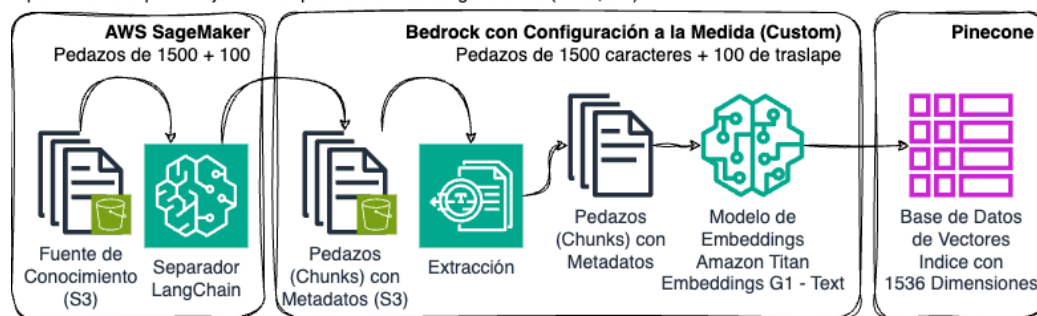


Pre-Procesamiento e Indexado de Documentos

El objetivo de esta etapa es extraer información de las fuentes de conocimiento y dividirlas en pedazos uniformes de contenido con metadatos. Los pedazos son procesados por un modelo de embeddings y almacenados, con sus metadatos asociados en una base de datos de vectores.

Retrieval Augmented Generation (RAG) - KB V2.00

La técnica RAG hace uso de una base de conocimiento para extender o aumentar la pregunta, petición o sugerencia proveída por el usuario para mejorar la respuesta del modelo generativo (AWS, s.f.).



Pre-Procesamiento e Indexado de Documentos

El objetivo de esta etapa es extraer información de las fuentes de conocimiento y dividirlas en pedazos uniformes de contenido con metadatos. Los pedazos son procesados por un modelo de embeddings y almacenados, con sus metadatos asociados en una base de datos de vectores.

AJUSTES DE LOS DATOS

El análisis de los resultados del modelo de referencia mostro que los documentos de referencia tenían información mezclada de uno o más hoteles. Para remediar esta situación se hizo lo siguiente:

- Generar archivos de datos y metadatos por hotel
- Generar los archivos de datos con un tamaño de segmento, o *chunk*, predeterminado de 1500 caracteres, con una tolerancia de 100 caracteres por traslape
- Configurar una nueva base de conocimiento, v2.00, con segmentos a la medida. En esta configuración Bedrock asume que el archivo ingerido tiene el tamaño adecuado.

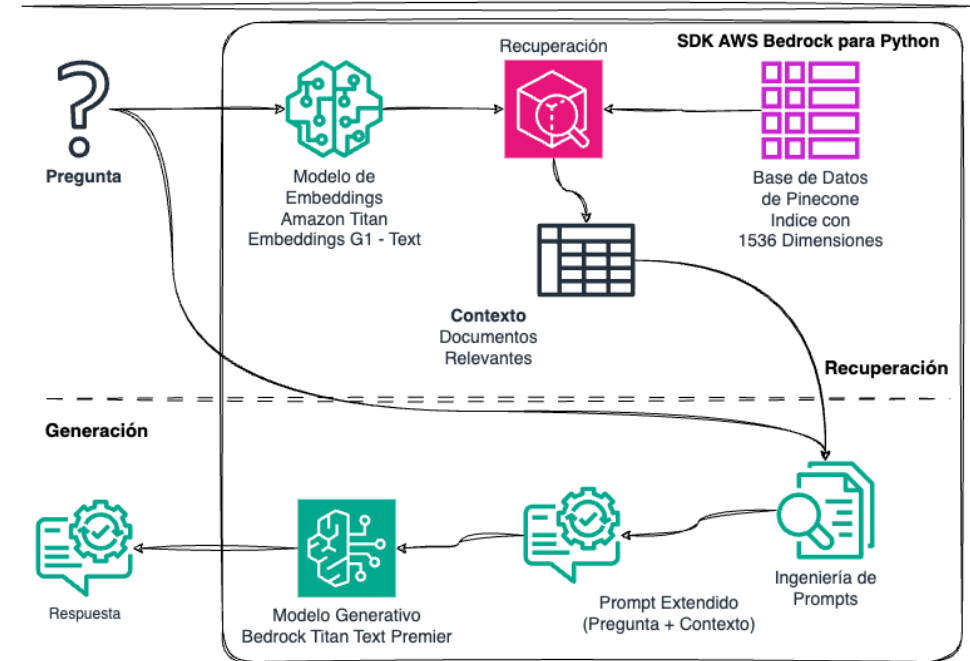
AJUSTES A LA ETAPA DE GENERACIÓN



GENERACIÓN DE REFERENCIA

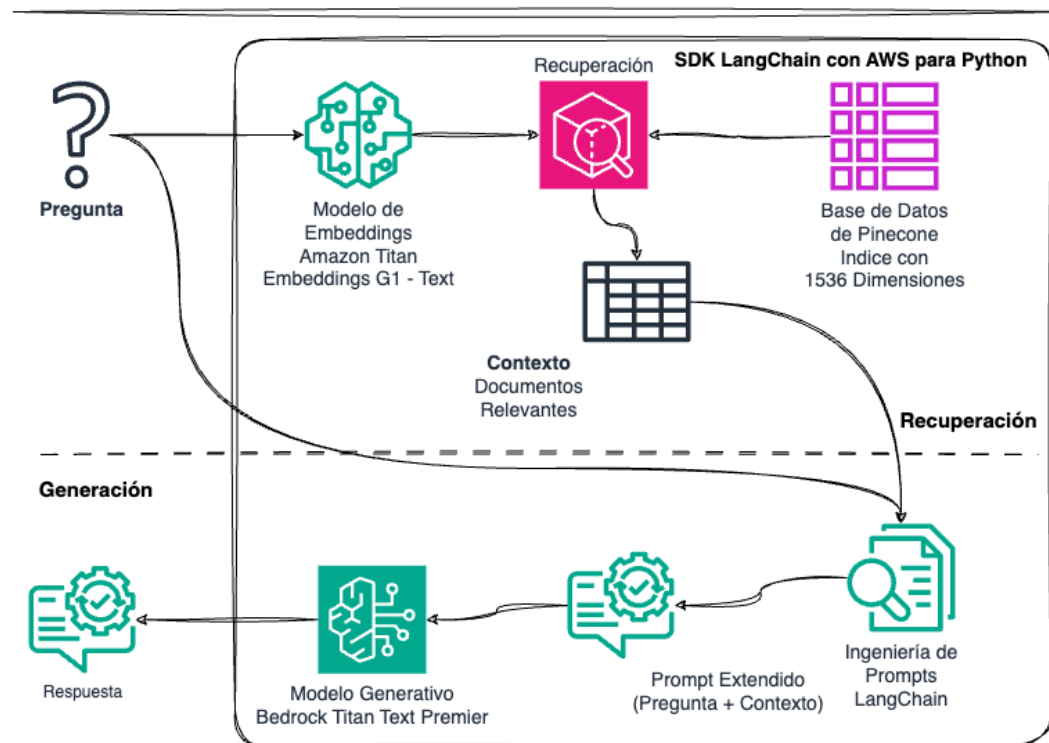
En la versión 1.00 se usaron las librerías, o SDK, de AWS para Bedrock para la generación de respuestas a las preguntas presentadas:

- La calidad y precisión de la respuesta variaba con el contenido de la pregunta, sin un patrón claro de mejora o deterioro
- Si bien los resultados eran prometedores también eran erráticos.



Recuperación de Documentos y Generación de Respuesta

El objetivo de esta etapa es recuperar información relevante de la base de datos de vectores en base a la pregunta presentada. La información recuperada es usada como contexto a la pregunta y ambas son presentadas al modelo generativo para producir la respuesta.



Recuperación de Documentos y Generación de Respuesta

El objetivo de esta etapa es recuperar información relevante de la base de datos de vectores en base a la pregunta presentada. La información recuperada es usada como contexto a la pregunta y ambas son presentadas al modelo generativo para producir la respuesta.

AJUSTES DE LAS LIBRERÍAS

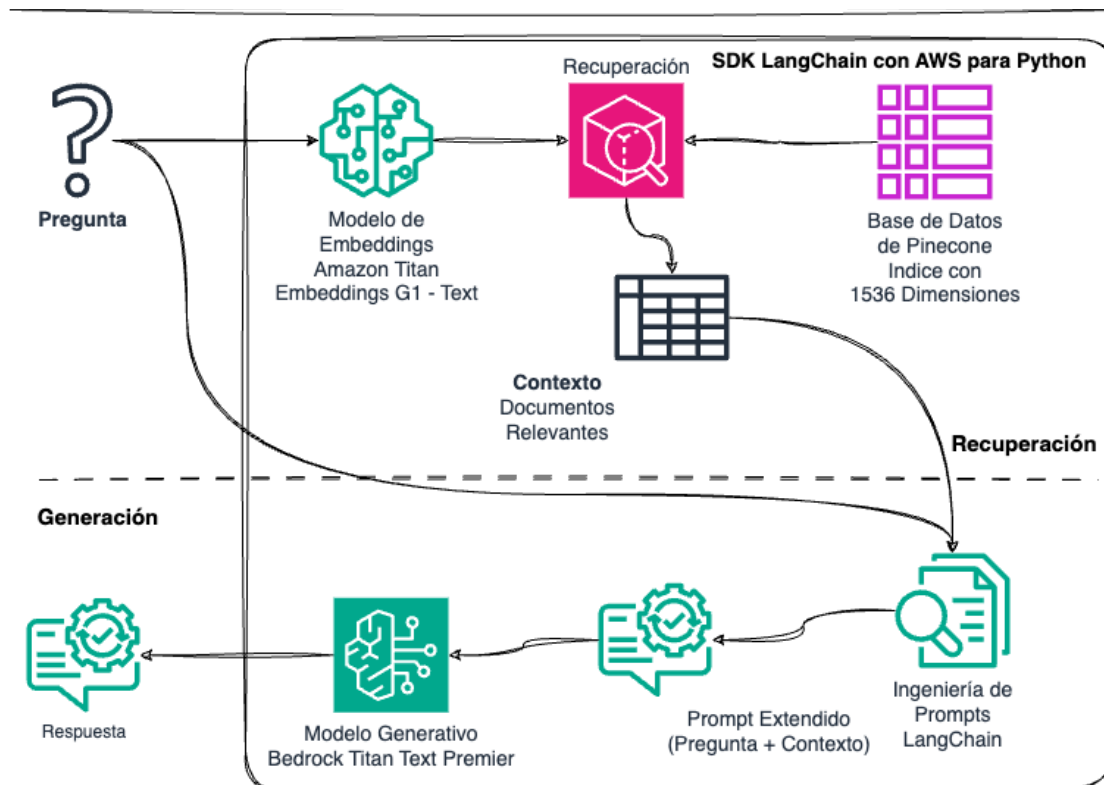
El uso de las librerías de LangChain permitieron:

- Mejorar la presentación de las respuestas generadas
- Tener más control de los hiper-parámetros para la recuperación de información
- Tener la habilidad para probar diferentes prompts.

AJUSTES DE LOS PARÁMETROS

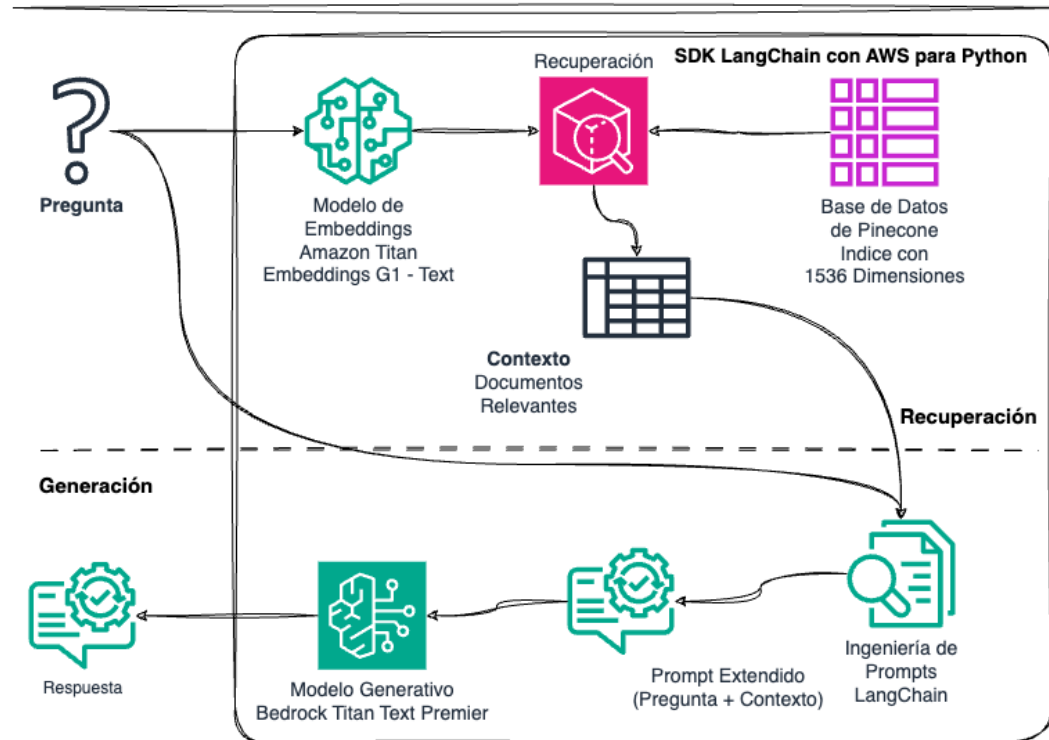
Los principales ajustes a los parámetros del modelo generativo Titan Premier Text fueron:

- *temperature* = 0 para forzar al modelo a seleccionar respuestas con altas probabilidades
- *topP* = 0.6 para limitar la selección de opciones a las que son más probables de ocurrir.



Recuperación de Documentos y Generación de Respuesta

El objetivo de esta etapa es recuperar información relevante de la base de datos de vectores en base a la pregunta presentada. La información recuperada es usada como contexto a la pregunta y ambas son presentadas al modelo generativo para producir la respuesta.



Recuperación de Documentos y Generación de Respuesta

El objetivo de esta etapa es recuperar información relevante de la base de datos de vectores en base a la pregunta presentada. La información recuperada es usada como contexto a la pregunta y ambas son presentadas al modelo generativo para producir la respuesta.

INGENIERÍA DE PROMPTS

Aún y con el aumento en la fidelidad de los datos fue necesario probar diferentes prompts para mejorar la precisión y relevancia de las respuestas. El nuevo prompt

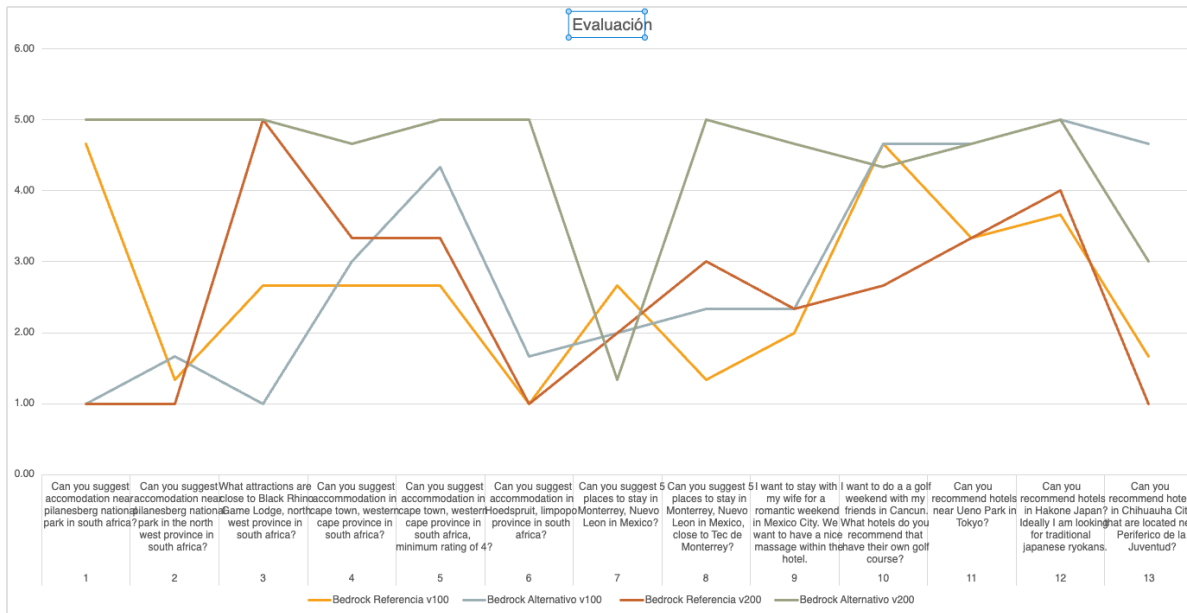
- Especifica el comportamiento esperado al preparar la respuesta
- Indica como seleccionar las mejores partes del contexto en términos de cercanía un lugar o atracción
- Solicita la elaboración de respuestas concisas, pero informativas y,
- Solicita que diga “No Se” si no podía llegar a una respuesta adecuada.

RESULTADOS

Al concluir todos los ajustes se evaluaron dos versiones de la base de conocimiento con dos conjuntos de algoritmos diferentes y dos tipos de prompt, básico y extendido agrupados en cinco libretas de Jupyter:

- Avance4_Equipo37_S3_Upload_Job.ipynb – libreta para preparar los datos para la version 2.00 de la base de conocimiento
- Avance3_Equipo37_Bedrock_Baseline_v1_00.ipynb – libreta de referencia
- Avance3_Equipo37_Bedrock_Baseline_v2_00.ipynb – libreta similar a la de referencia que usa la V2.00 de la base de conocimiento
- Avance4_Equipo37_Bedrock_Alternative_v1_00.ipynb y Avance4_Equipo37_Bedrock_Alternative_v2_00.ipynb – libretas con las nuevas librerías, los nuevos parámetros y el prompt mejorado, usando versión 1.00 y 2.00 de la base de conocimiento respectivamente.

Como se puede notar en la gráfica, **Avance4_Equipo37_Bedrock_Alternative_v2_00** fue la que mejor se desempeñó en casi todas las preguntas.



PASOS SIGUIENTES

El proyecto va a proseguir a la etapa de preparación del modelo final con métricas de desempeño.




GRACIAS

[HTTP://WWW.MARGIESTRAVEL.COM/](http://www.margiestravel.com/)



 **Equipo 37**

 +1 (589) 555-0199

 victoria@margiestravel.com



BIBLIOGRAFÍA

- AWS. (s.f.). What is RAG? - Retrieval-Augmented Generation Explained. Amazon Web Services, Inc. Recuperado 25 de abril, 2024 de <https://aws.amazon.com/what-is/retrieval-augmented-generation/>
- Apache.org, (s.f.). File Format. Apache Parquet. Recuperado 25 de abril, 2024. <https://parquet.apache.org/docs/file-format/>
- Boudier, C. (2024, 4 abril). From Sketch to Success: Strategies for Building & Evaluating an Advanced RAG System. Blog.dataiku.com. <https://blog.dataiku.com/strategies-for-building-evaluating-an-advanced-rag-system>
- ml-ops.org. (s.f.). CRISP-ML(Q). The ML Lifecycle Process. ml-ops.org. Recuperado 28 de abril, 2024, de <https://ml-ops.org/content/crisp-ml>
- WS (s.f.). Amazon Titan Text models - Amazon Bedrock. docs.aws.amazon.com. Recuperado 20 de mayo 26, 2024, de <https://docs.aws.amazon.com/bedrock/latest/userguide/model-parameters-titan-text.html>
- AWS. (s.f.). Inference parameters - Amazon Bedrock. docs.aws.amazon.com. Recuperado 20 de mayo 26, 2024, de <https://docs.aws.amazon.com/bedrock/latest/userguide/inference-parameters.html#:~:text=lop%20P%20%E2%80%93%20The%20percentage%20of>
- LangChain. (s.f.). Bedrock (Knowledge Bases). python.langchain.com. Recuperado 20 de mayo 26, 2024, de <https://python.langchain.com/v0.1/docs/integrations/retrievers/bedrock/>
- LangChain. (s.f.). ChatBedrock. python.langchain.com. Recuperado 20 de mayo 26, 2024, de <https://python.langchain.com/v0.1/docs/integrations/chat/bedrock/>
- LangChain. (s.f.). langchain.chains.retrieval_qa.base.RetrievalQA LangChain 0.1.12. api.python.langchain.com. https://api.python.langchain.com/en/latest/chains/langchain.chains.retrieval_qa.base.RetrievalQA.html