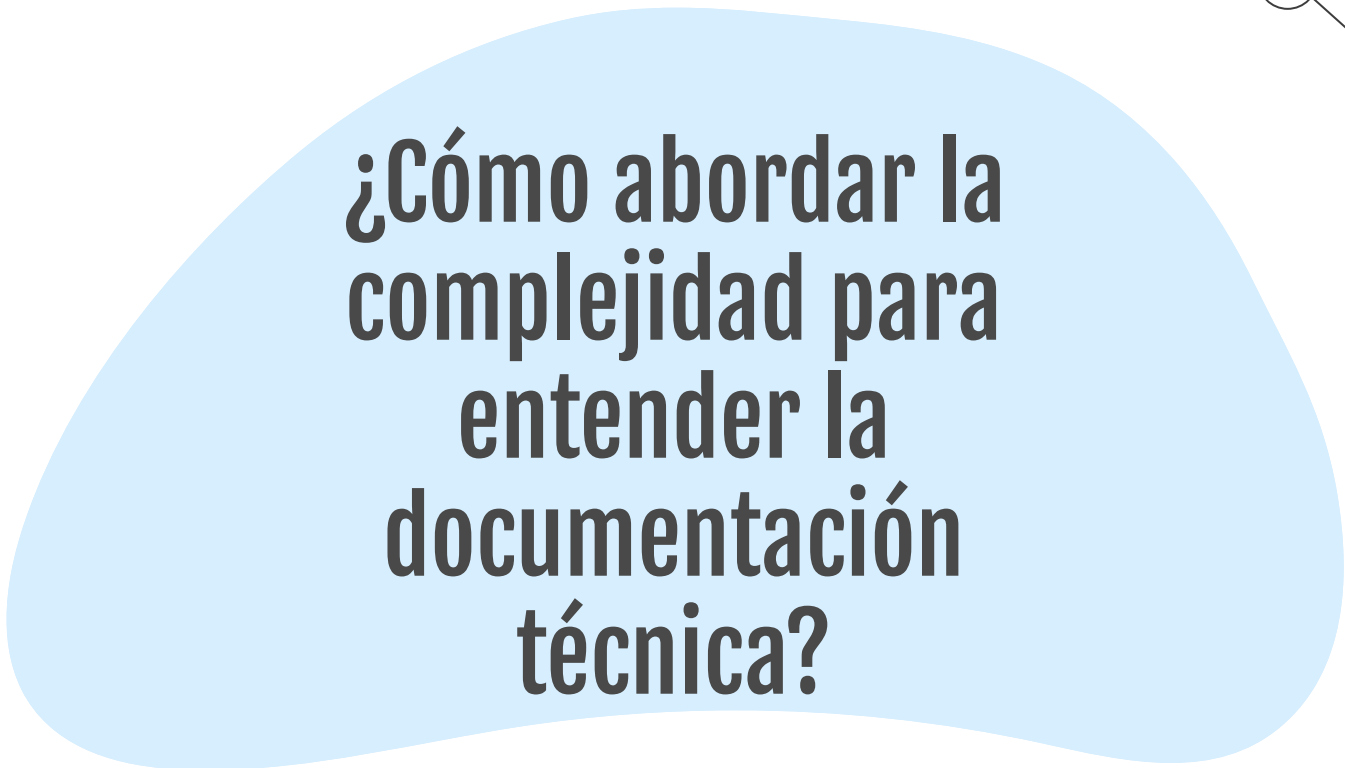
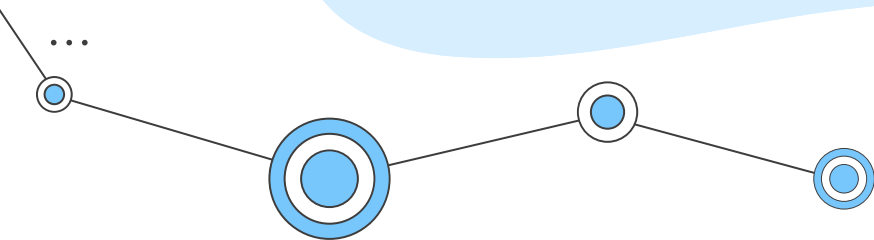
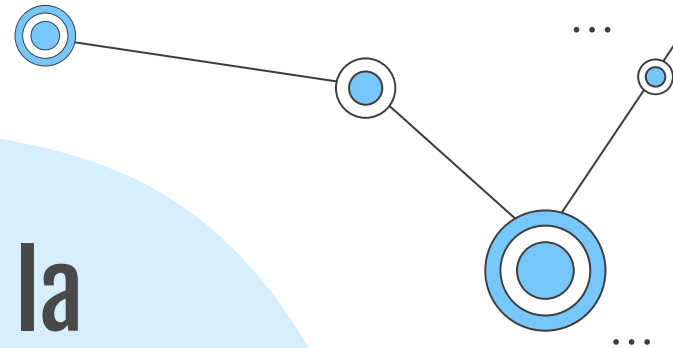


Uso de modelos de lenguaje para mejorar el entendimiento de documentación técnica

Andres Felipe Marcelo Rubiano
Juan Camilo Angel Hernandez
Juan Camilo Rojas Ortiz



**¿Cómo abordar la
complejidad para
entender la
documentación
técnica?**



Marco teorico



LLM

Modelos especializados de procesamiento lingüístico de propósito general



¿Que hace un LLM?

Realizar una gran variedad de tareas de procesamiento de lenguaje natural



Base datos vectorizada ¿Que bases de datos hay?

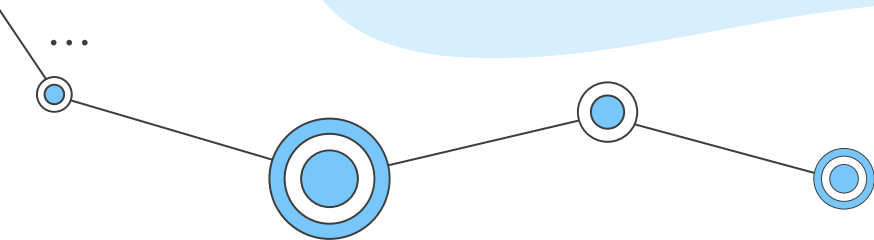
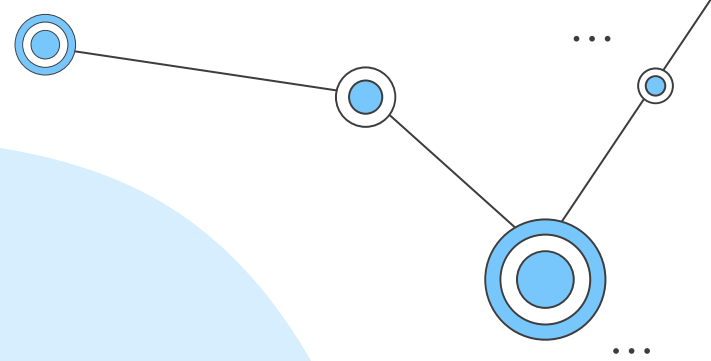
Base de datos que almacena y maneja vectores soportando operaciones como la búsqueda por similitud



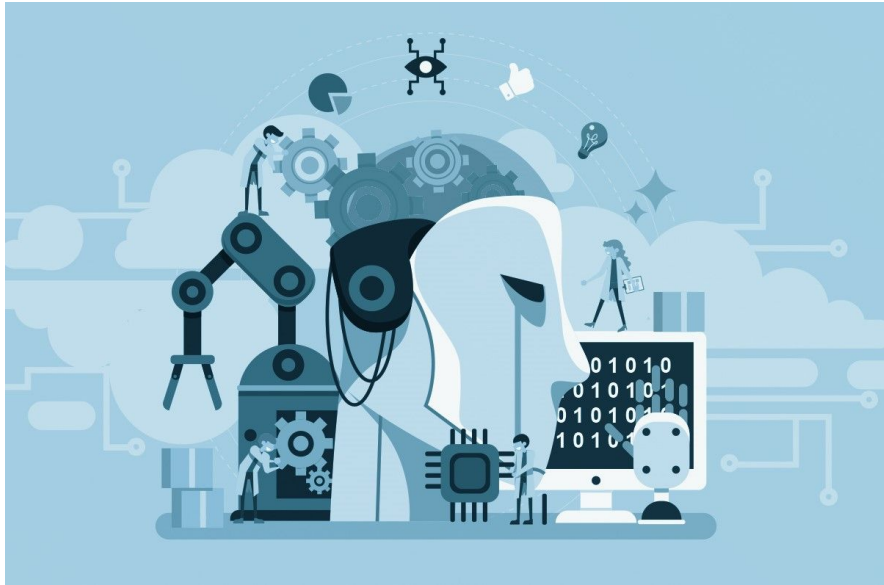
Contamos con proveedores como Elasticsearch y Pinecone



Arquitectura



Solución propuesta



- Diseñar un sistema globalmente distribuido capaz de dar respuesta a las dudas de los usuarios sobre la documentación técnica presente.
- Uso de grandes modelos de lenguaje junto con bases de datos vectorizadas para acceder a la documentación.
- Mejora en la comprensión de ésta a través de preguntas, ahorrando tiempo y aumentando la productividad.

Dominios



Negocio

Aumento de productividad
(Onboarding - Soporte técnico - Gestión del conocimiento)



Información

Amplia base de conocimiento con información técnica
(Vectorial DB)



Aplicación

API que permite al usuario realizar preguntas sobre la información presente

Componente de actualización y entrenamiento de Datos en base a repositorios enviados



Tecnología

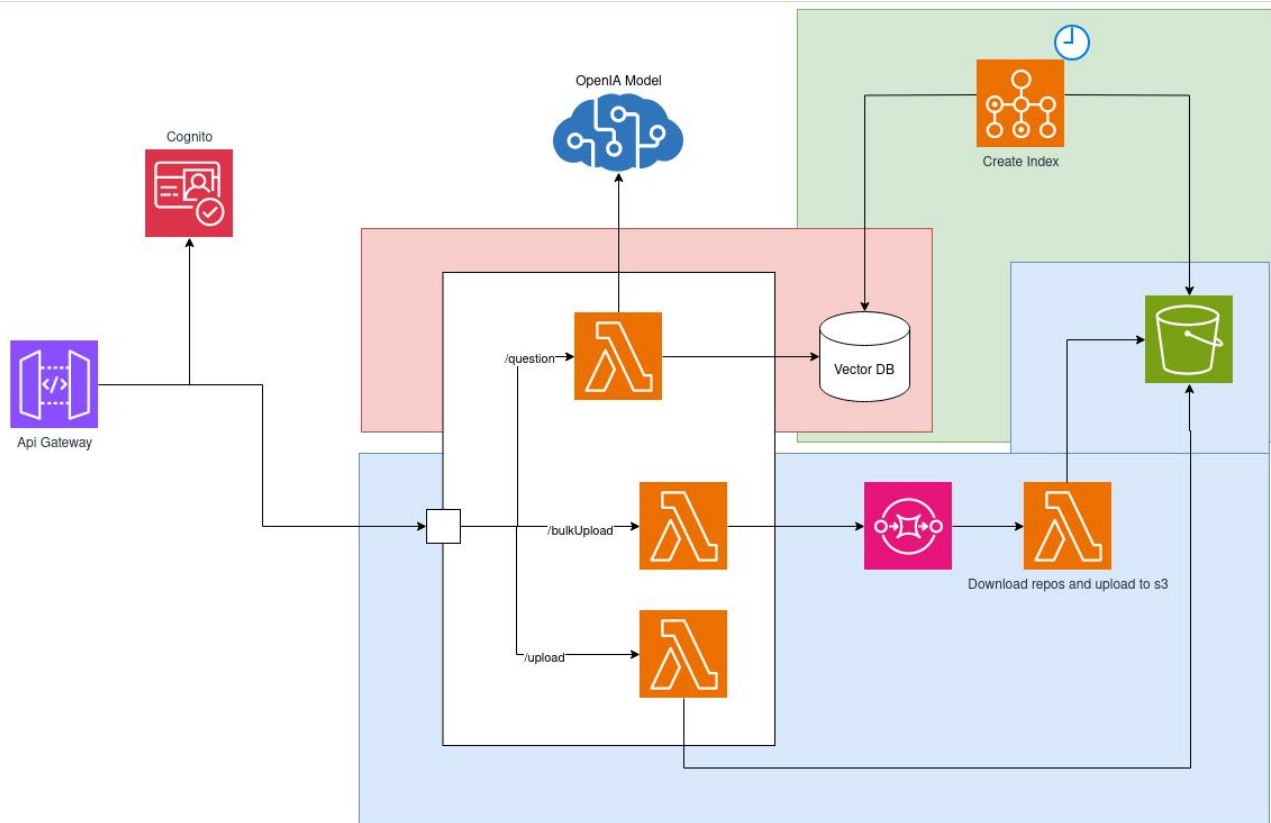
Infraestructura en la nube con mecanismos de escalamiento



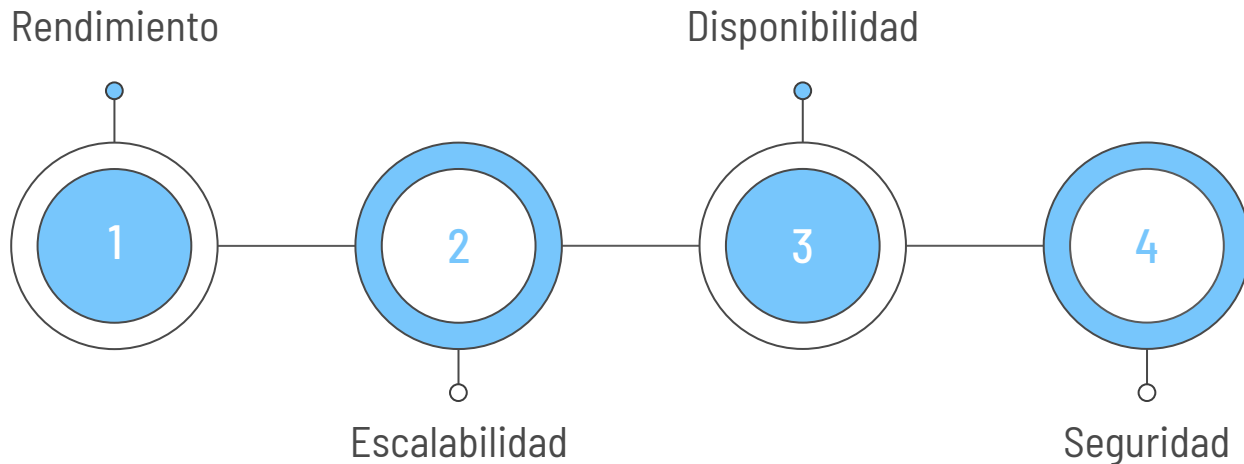
Seguridad

Mecanismos de autenticación (Oauth2) para proteger aplicación y Datos

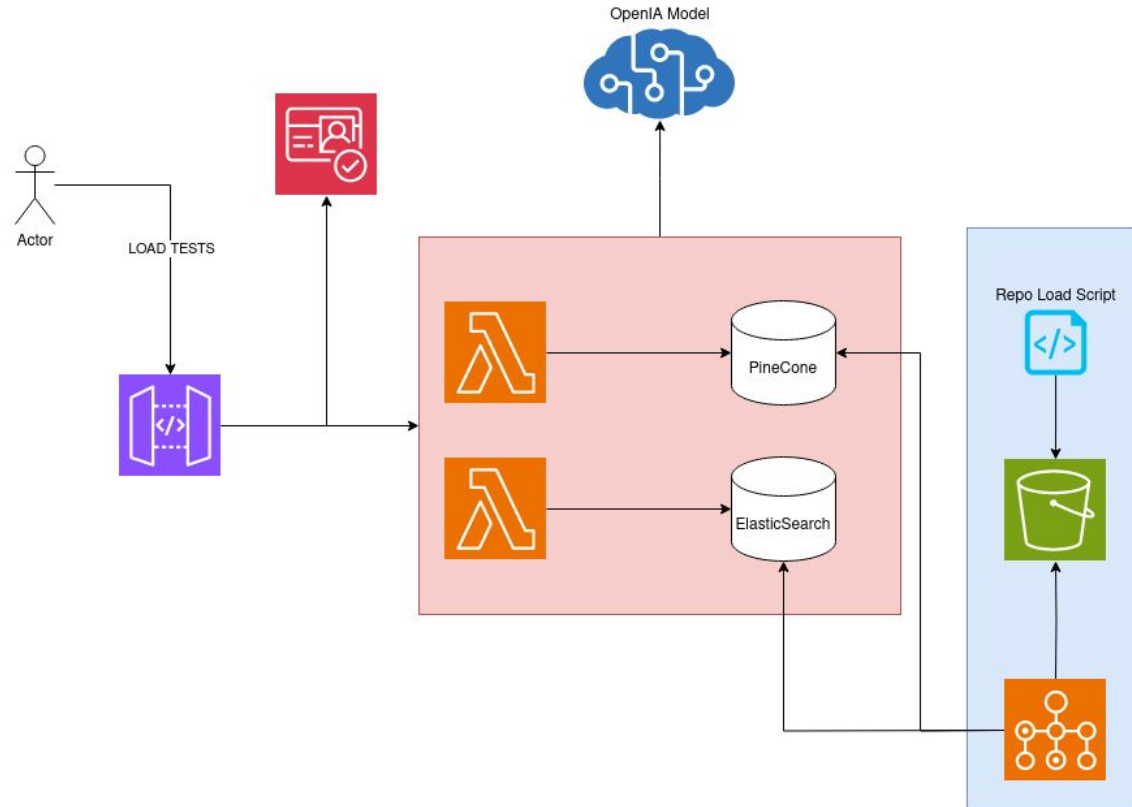
Arquitectura propuesta



Atributos de calidad



Diseño de prototipo



Data



Origen

Documentos .md
pertenecientes a la
organización
aws-samples



Contenido

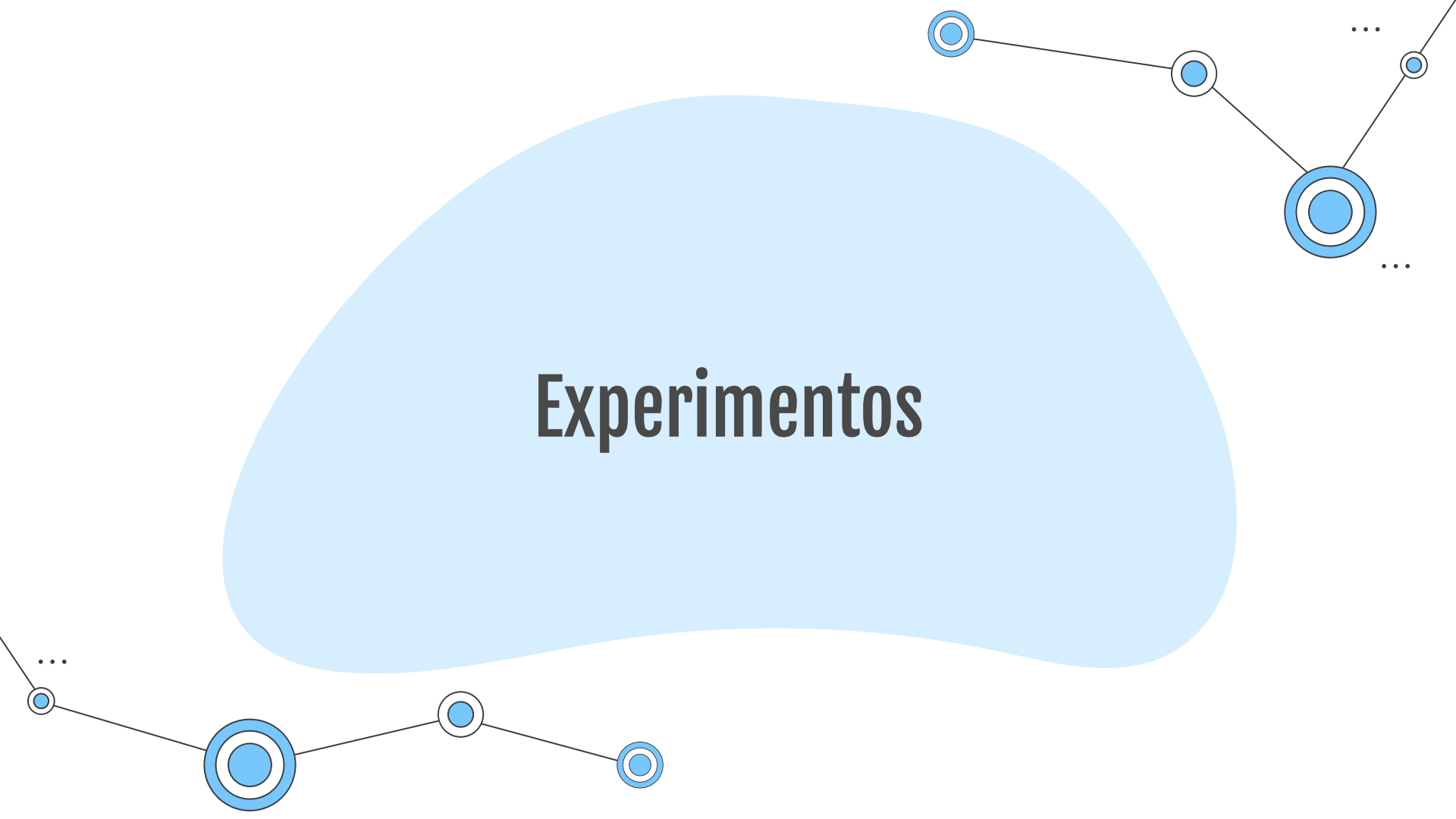
Información sobre
una variedad de
temas de ingeniería
y arquitectura de
software.



Cantidad

3500 documentos
READMEs de
arquitectura de
software (AWS)

Experimentos



Rendimiento de entrenamiento

- Vectores generados en ambos proveedores : 7920.
- Ambos proveedores generaron vectores de 1536 dimensiones
- El costo de cada ejecución con elasticSearch fue 0.38USD sobre la api de OpenAI
- El costo de cada ejecución con pinecone fue 0.34USD sobre la api de OpenAI
- Se encontraron diferencias de precisión entre los vectores de cada proveedor
 - Pinecone : -0.000817436317
 - ElasticSearch: -0.0008254778434388454

Run	Total Execution Time (seconds)
1	720.0
2	730.0
3	722.5
4	728.5
5	724.0
Avg	~725.0

TABLE I


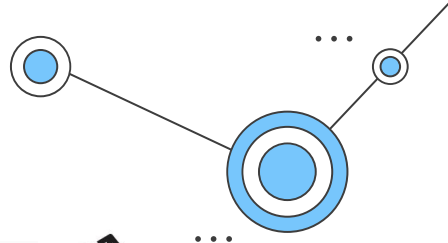
EXECUTION TIME FOR ELASTICSEARCH TRAINING EXPERIMENTS

Run	Total Execution Time (seconds)
1	680.0
2	675.5
3	678.0
4	680.5
5	675.0
Avg	~677.4

TABLE II

EXECUTION TIME FOR PINECONE TRAINING EXPERIMENTS

Rendimiento de entrenamiento – Resultados en nube



Amazon S3 > Buckets > aygo-is-project > data/

data/

Objects Properties

To enable sorting in the table below, use the search to reduce the size of the list to 999 objects or fewer.

Objects (999+) info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
serverless-disaster-recovery-with-global-accelerator-contents-README.md	md	December 7, 2023, 02:43:58 (UTC-05:00)	5.0 KB	Standard
serverless-dotnet-demo-contents-README.md	md	December 7, 2023, 02:43:58 (UTC-05:00)	21.2 KB	Standard

Chatbot application indexes



[Back to Indexes](#)

pinecone-index

Free Tier

METRIC	DIMENSIONS	POD TYPE	HOST
cosine	1536	starter	https://pinecone-index-mo12ilo.svc.gcp-starter.pinecone.io
CLOUD	REGION	ENVIRONMENT	VECTOR COUNT
GCP	Iowa (us-central1)	gcp-starter	7920

[BROWSER](#)

METRICS

NAMESPACES (1)

Namespace
(Default)

Query by Vector

vector 0.71,0.38,0.47,0.81,0.61,0.46,0.68,0.24,0.27,0.06,0.13,0.37,0.2

Top K

50

Query

Metadata Filter

elastic-index

Search Applications

[Overview](#)

[Documents](#)

[Index mappings](#)

[Pipelines](#)

Ingestion type
API

Document count
7920

How create a kinesis data stream to receive data, give me detailed explanation

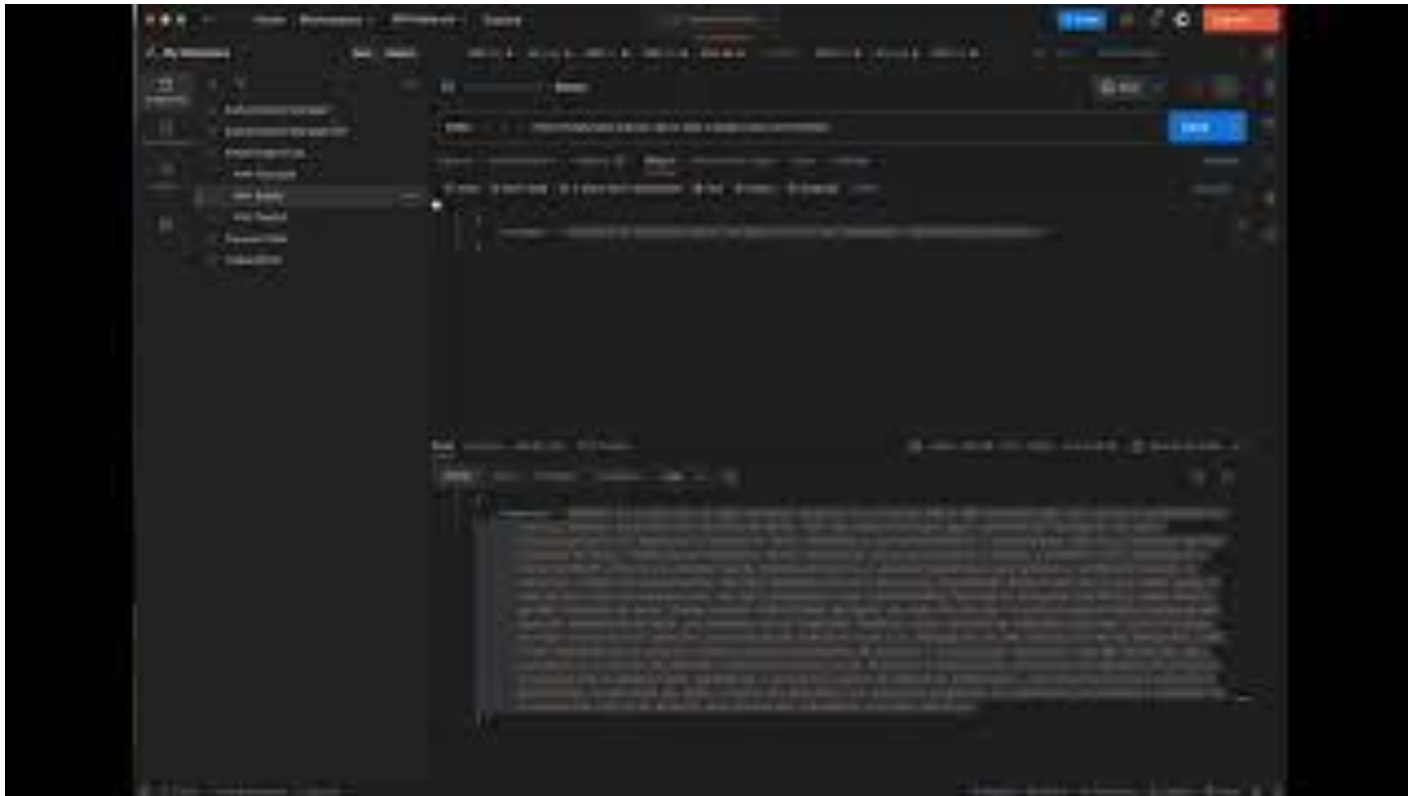
Pinecone

ElasticSearch

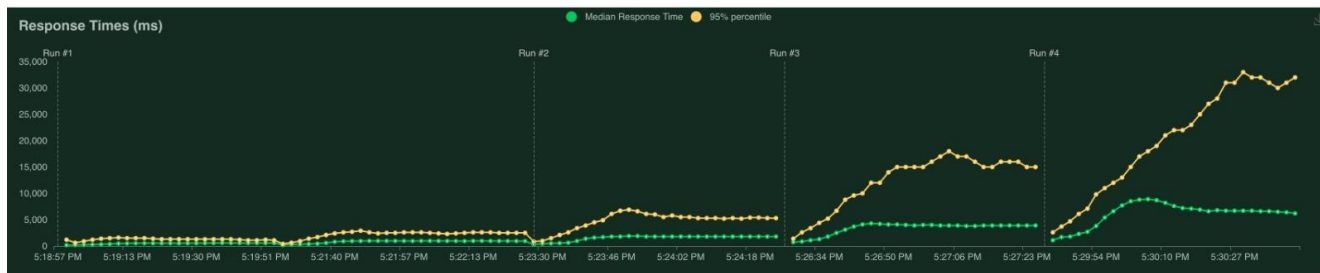
To create a Kinesis Data Stream, you can use the AWS Management Console, the AWS Command Line Interface (CLI), or the AWS SDK. Using the AWS Management Console: 1. Log in to the AWS Management Console and open the Amazon Kinesis console. 2. Choose Create data stream. 3. Enter a name for the data stream. 4. Choose the number of shards for the data stream. 5. Choose Create data stream. Using the AWS CLI: 1. Run the create-stream command. 2. Enter the name of the data stream. 3. Enter the number of shards for the data stream. 4. Run the command. Using the AWS SDK: 1. Create an Amazon Kinesis client. 2. Create a CreateStreamRequest object. 3. Set the name and number of shards for the data stream. 4. Call the createStream method of the Amazon Kinesis client.

To create a Kinesis Data Stream to receive data, you can follow these steps: 1. Open the AWS Management Console and navigate to the Kinesis service. 2. Click on "Create data stream" to start the creation process. 3. Provide a name for your data stream. This should be a unique name within your AWS account. 4. Specify the number of shards you want for your data stream. Shards determine the capacity of your stream to handle incoming data. Each shard can handle up to 1 MB of data per second (including both reads and writes) or 1000 records per second for writes. 5. Choose the appropriate settings for data retention. This determines how long the data will be stored in the stream before it is automatically deleted. You can choose a retention period of up to 7 days. 6. Click on "Create data stream" to create the stream. Once the data stream is created, you can start sending data to it using the Kinesis API or SDKs. You can also configure consumers, such as AWS Lambda functions or Kinesis Data Firehose, to process and analyze the data from the stream. It's important to note that creating a Kinesis Data Stream incurs costs based on the number of shards and the amount of data ingested and stored. Make sure to monitor and manage your stream to optimize costs and performance.

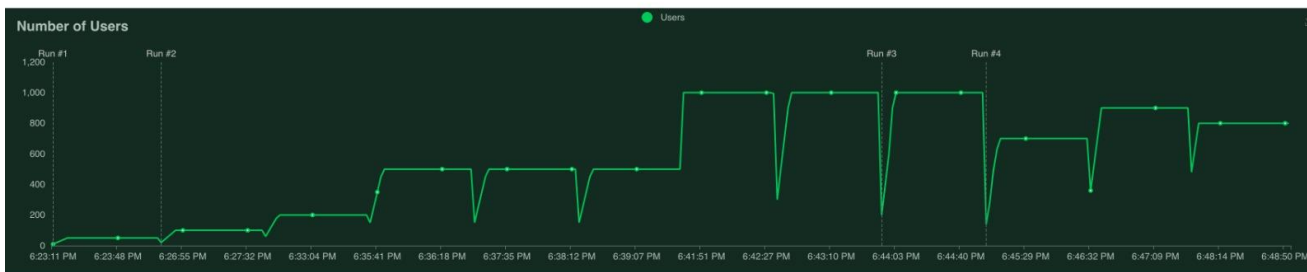
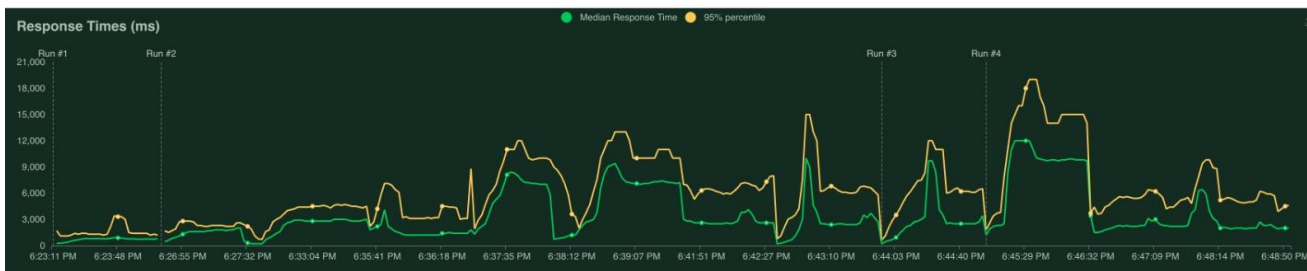
API Demo



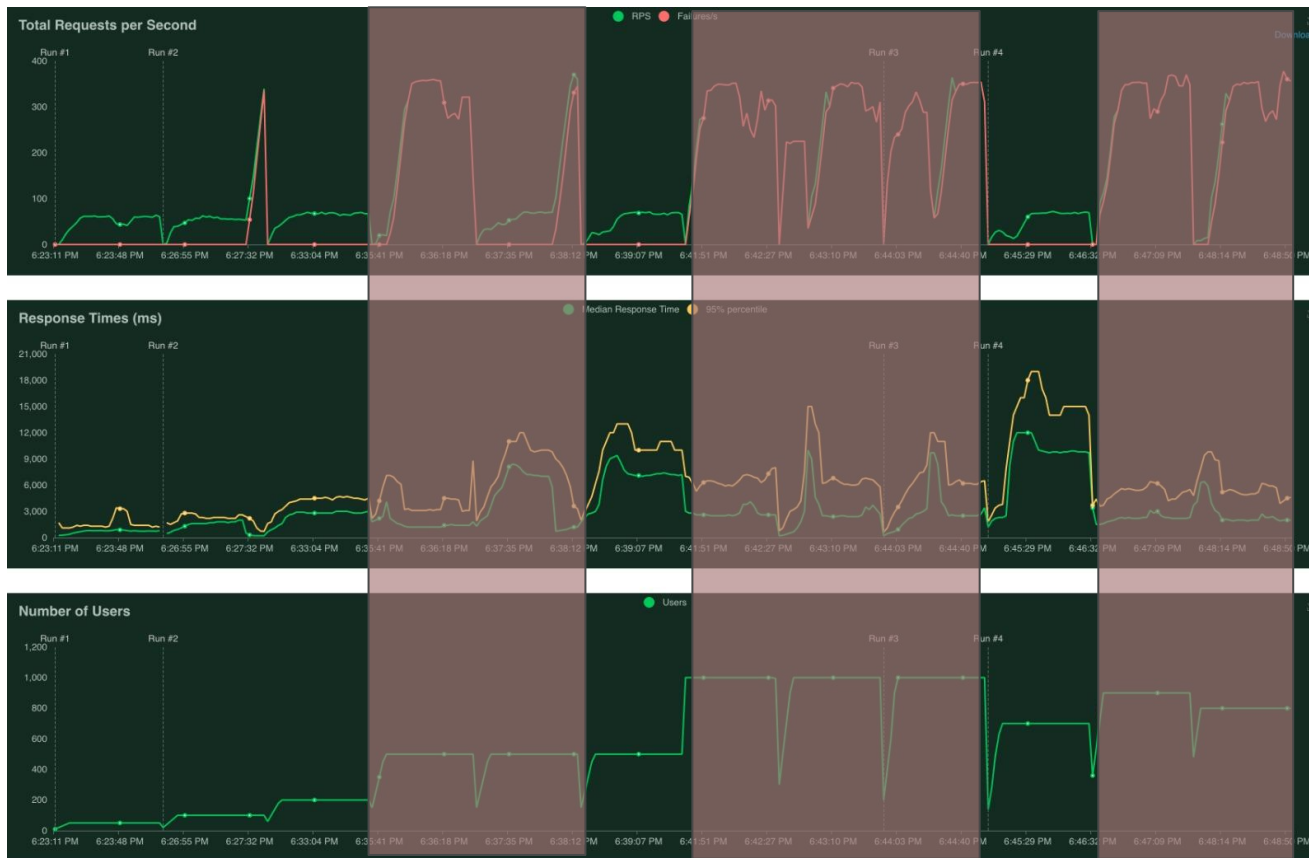
Pruebas de carga – Pinecone



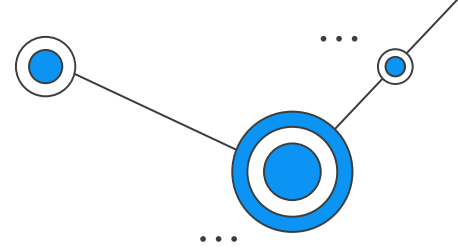
Pruebas de carga – Elasticsearch



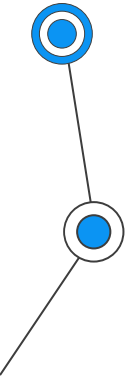
Pruebas de carga – Elasticsearch



Conclusiones



- Se evidenció el potencial de los modelos LLM + Bases de datos de vectores para generar análisis sobre documentos propios no incluidos en el entrenamiento de los modelos
- Cada proveedor de base de datos almacena los vectores con diferentes niveles de precisión y esto afecta el comportamiento del modelo
- Elasticsearch almacena vectores con mayor precisión y esto genera respuestas con mayor calidad al sacrificar rendimiento
- Pinecone tiene menos precisión pero presenta mejor rendimiento
- Para el contexto del problema consideramos más importante la calidad de las respuestas que el rendimiento, por lo que el proveedor de preferencia es elasticSearch





Muchas Gracias!

¿Preguntas?

