# Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física
**Maestría en Ciencia de Datos**


## Smart sample selection for Retrieval Augmented Generation

---

**TRABAJO RECEPCIONAL** que para obtener el **GRADO** de
**Maestro en Ciencia de Datos**

Presenta:
**Diego Ramírez Barba**

Director:
**Mtro. Byron Michael Motta Bonilla**

Tlaquepaque, Jalisco, 5 de diciembre de 2024

# Smart sample selection for Retrieval Augmented Generation

Diego Ramírez Barba

## Resumen

El proyecto tiene como objetivo la optimización de una técnica emergente en el campo del machine learning llamada RAG (Retrieval-Augmented Generation). Esta técnica se utiliza para mejorar el procesamiento de datos en LLMs (Large Language Models) mediante diversos algoritmos, los cuales permiten aumentar la precisión y reducir el sesgo inherente al entrenamiento de estos modelos.

Decidimos utilizar una base de datos de imágenes de perros las cuales están categorizadas por raza, ya que el manejo de imágenes representa desafíos adicionales, como el manejo de las características de las imágenes para realizar su gestión y procesamiento.

Empleamos varios de los modelos aprendidos durante la maestría, como las redes neuronales para clasificación y random forest para evaluar la exactitud de los modelos. Después de realizar estas pruebas iniciales, decidimos comparar los resultados con los modelos comerciales más avanzados, utilizando la infraestructura de Nvidia AI Foundation como referencia (Neva22b, Kosmos2, Vila, Fuyu). Para el procesamiento de las imágenes, implementamos una serie de técnicas de preprocesamiento y transformación de datos que nos facilitaron su manejo.

Nuestro objetivo es optimizar tanto la exactitud como el procesamiento de los datos mediante el uso de vectores. Para ello, decidimos incorporar FAISS (Facebook AI Similarity Search), una herramienta diseñada para realizar búsquedas rápidas de similitud en grandes volúmenes de datos de alta dimensión. Esto nos permitió aplicar la técnica de búsqueda de vecinos más cercanos (nearest neighbor search) para procesar los vectores generados a partir de las imágenes.

En cuanto a la gestión del error, implementamos diversos métodos de muestreo, incluyendo muestreo aleatorio, sistemático y por clúster, con el fin de tomar decisiones más informadas y precisas durante el proceso de validación de nuestros modelos. Al obtener un balance entre el procesamiento y la exactitud de los modelos pudimos realizar la generación aumentada por recuperación (RAG), la cual nos permitió realizar consultas más precisas a los modelos de NVIDIA y así poder obtener mejores resultados con los modelos menos entrenados a través de nuestros vectores de búsqueda.

# Tabla de Contenidos

# Índice de figuras

**Página**

# Índice de tablas

**Página**

*Dedicado al Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) por el apoyo económico recibido a través de una beca. Adicionalmente, a la Empresa Oracle por los recursos proporcionados para el desarrollo de este Trabajo de Obtención de Grado*

# 1 Introducción

## 1.1 Contexto

En la actualidad podemos observar cómo cada día se generan nuevas soluciones utilizando inteligencia artificial (IA), con aplicaciones como ChatGPT, Siri, Alexa, Google Assistant, entre otras. Estas soluciones, que abarcan desde asistentes virtuales hasta sistemas avanzados de recomendación y búsqueda, comparten un factor común: el uso de grandes cantidades de datos para alimentar modelos de lenguaje de gran escala (LLMs).

A pesar de los avances alcanzados por los LLMs, siguen enfrentando retos importantes en cuanto a precisión. Uno de los principales problemas es el subentrenamiento o sobreentrenamiento de los modelos. El subentrenamiento ocurre cuando el modelo no se ha entrenado con suficientes datos, lo que limita su habilidad para generalizar correctamente. Por el contrario, el sobreentrenamiento se da cuando el modelo se adapta demasiado a los datos de entrenamiento, lo que puede llevarlo a hacer generalizaciones erróneas y a volverse sensible al ruido o a detalles irrelevantes. Estas dificultades afectan la calidad y exactitud de las respuestas generadas, lo cual es especialmente crítico en aplicaciones que requieren alta precisión y contexto detallado.

Para poder abordar estos problemas, se ha propuesto la técnica de Retrieval-Augmented Generation (RAG), la cual busca mejorar la exactitud y precisión de las respuestas producidas por los LLMs [3]. RAG combina la recuperación de información de fuentes externas con la generación de texto, lo que mejora su capacidad para ofrecer respuestas. Este enfoque mejora la capacidad del sistema para ofrecer respuestas más precisas y contextualizadas.

Una herramienta clave para implementar RAG de manera eficiente es el uso de FAISS (Facebook AI Similarity Search) y la búsqueda vectorial (vector search) [4]. FAISS es una biblioteca que permite realizar búsquedas rápidas y escalables en grandes volúmenes de datos

a través de representaciones vectoriales. Mediante la conversión de la información en vectores, es posible comparar y recuperar datos de manera más eficiente, incluso cuando se manejan millones de documentos o fragmentos de texto. Esto mejora considerablemente la capacidad del sistema para localizar la información relevante de manera más rápida y precisa, lo que es crucial en aplicaciones donde se requiere información específica sobre un tema.

## 1.2 *Justificación*

El uso de técnicas avanzadas como RAG tiene una gran relevancia debido a los desafíos actuales en el ámbito de los LLMs, especialmente en lo que respecta a la precisión en la generación de respuestas. A pesar de los avances en la generación de texto, los modelos siguen enfrentando limitaciones cuando se trata de acceder a información actualizada o especializada, lo que puede afectar el desempeño en tareas como respuestas a preguntas, recomendaciones y procesamiento de imágenes. Otro de los grandes retos de los LLMs es el costo computacional, ya que a través de los años se han incrementado notablemente los casos de uso de los LLM como muestra la Figura 1.1 los cuales se traducen a más consumo de CPU/GPU incrementando los costos operativos.

Este trabajo se justifica en su intento por mejorar estos aspectos, utilizando un enfoque experimental que no solo optimiza el uso de RAG, sino que también explora su aplicabilidad en el procesamiento de imágenes, un área donde las técnicas de recuperación y generación aún presentan retos considerables.

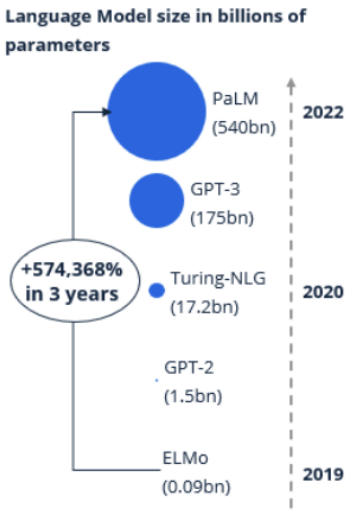


Figura 1.1: Incremento de los casos de uso en los LLMs a lo largo de los años [1]

## 1.3 Problema

El principal desafío que aborda este proyecto es la clasificación automática de imágenes de perros para determinar su raza, las redes neuronales han mostrado un rendimiento impresionante en tareas de clasificación de imágenes. Sin embargo, el desempeño de estos modelos depende en gran medida de la calidad de los datos y de la adecuada selección y configuración de los algoritmos utilizados.

Este proyecto se centra en la aplicación de diferentes algoritmos de clasificación para predecir la raza de un perro a partir de su imagen, utilizando como métrica principal la exactitud (accuracy) para evaluar el rendimiento de los modelos. Los algoritmos que se emplearán incluyen Redes Neuronales para clasificación con distintas configuraciones, como ReLU, Softmax, Linear y Tanh, con el fin de comparar su capacidad de generalización y exactitud. Además, se utilizará el algoritmo Random Forest con diferentes números de estimadores para analizar su capacidad de clasificación en este contexto.

Para complementar los modelos tradicionales, se explorarán modelos como Neva22b [5], Kosmos2 [6], Vila [7] y Fuyu [8], modelos de última generación diseñados para tareas de clasificación de imágenes y procesamiento de datos visuales. Estos modelos se evaluarán con el objetivo de identificar si superan a los enfoques convencionales en términos de precisión y eficiencia en la clasificación. Después se buscará mejorar su precisión y rendimiento utilizando FAISS y RAG.

## 1.4 Objetivos

### 1.4.1 Objetivo general

Desarrollar un sistema de clasificación automática de imágenes para determinar la raza de un perro, utilizando diferentes algoritmos de clasificación, como redes neuronales y Random Forest, y evaluando su rendimiento en términos de precisión (accuracy). Además, se explorarán modelos de última generación (Neva22b, Kosmos-2, Vila y Fuyu) para comparar su efectividad en la clasificación de imágenes, con el objetivo de optimizar su precisión y eficiencia a través del uso de técnicas avanzadas de búsqueda de vectores como FAISS y RAG.

### 1.4.2 *Objetivos específicos*

1. Implementar y evaluar redes neuronales para clasificación de imágenes: Aplicar diferentes configuraciones de redes neuronales (ReLU, Softmax, Linear, Tanh) para evaluar su capacidad de generalización y precisión en la clasificación de razas de perros.

2. Evaluar el rendimiento de Random Forest: Aplicar el algoritmo Random Forest con diferentes números de estimadores para comparar su capacidad de clasificación con respecto a las redes neuronales, y medir su desempeño en términos de precisión.

3. Comparar modelos de última generación: Evaluar modelos avanzados como Neva22b, Kosmos-2, Vila y Fuyu en el contexto de clasificación de imágenes de razas de perros, con el objetivo de identificar cuál de ellos ofrece el mejor rendimiento en términos de precisión y eficiencia computacional.

4. Optimizar el rendimiento de los modelos utilizando técnicas de búsqueda de vectores: Implementar FAISS (Facebook AI Similarity Search) para mejorar la precisión de la clasificación mediante la búsqueda eficiente de vecinos más cercanos, y analizar cómo esta optimización impacta el rendimiento de los modelos.

5. Optimizar la calidad de los datos y la configuración de los algoritmos: Ajustar los parámetros y la configuración de los algoritmos para maximizar la precisión del sistema, después aplicar RAG para optimizar las llamadas a los LLMs haciendo un manejo predictivo del error.

# 2 Metodología

## 2.1 Descripción de los datos

La base de datos **Stanford Dogs** contiene un total de 20,580 imágenes que pertenecen a 120 razas de perros diferentes. Cada raza de perro está representada por un directorio que lleva el nombre de la raza, y dentro de cada directorio se encuentran las imágenes correspondientes a esa raza. Los datos están organizados de la siguiente manera:

- Número de razas/etiquetas: 120
- Número total de imágenes: 20,580
- Tamaño total de las imágenes: 757MB
- Formato de las imágenes: JPEG (con extensión .jpg)
- Estructura de directorios: Cada raza tiene un directorio individual, cuyo nombre corresponde a la raza de perro, y dentro de este directorio se encuentran las imágenes asociadas a esa raza.

## 2.2 Análisis exploratorio

Comenzamos realizando un análisis de las distribuciones de las razas de perros como muestra la Figura 2.1 la cual muestra que las razas con un mayor numero de imagenes son el maltés, el galgo afgano y el galgo escocés como se muestra en la Figura 2.2. Observamos que la base de datos no contenía valores duplicados, nulos ni NaN (valores numéricos indefinidos o inexistentes [9]). Luego, procesamos y normalizamos las imágenes utilizando un formato de tamaño estándar de 224x224, lo cual nos permitirá optimizar operaciones como el uso de ResNet50. Posteriormente, realizamos una segunda normalización en un rango de [0, 1] para mejorar la convergencia de los datos y asegurarnos de que todos estén en una escala similar.

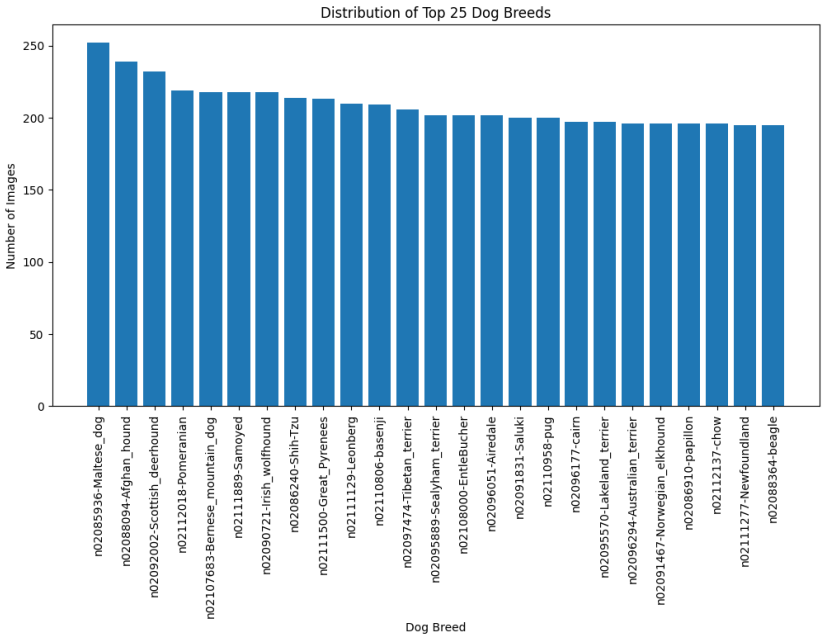


Figura 2.1: Distribución de las 25 razas de perros con más imágenes

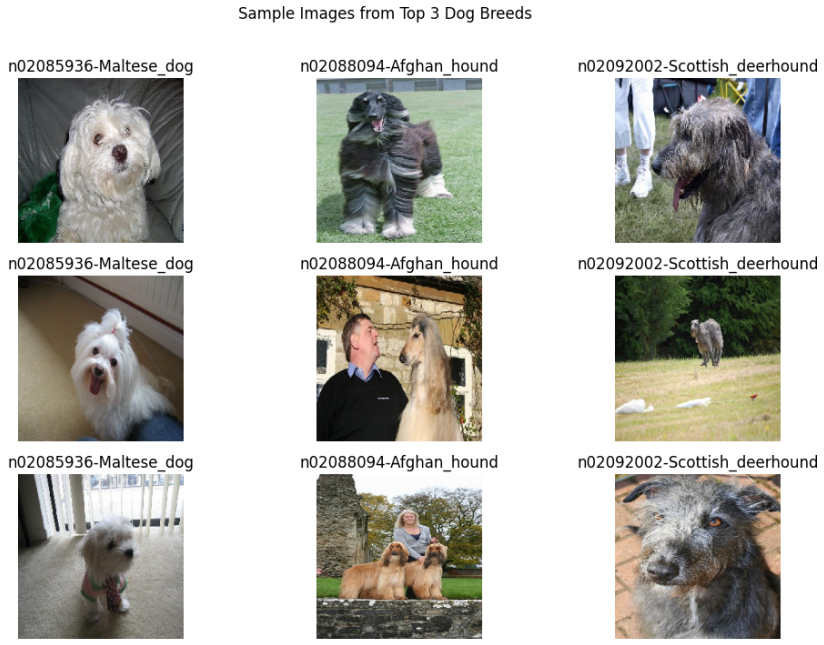


Figura 2.2: Las 3 razas de perros con mas imágenes

## 2.3 Descripción de los modelos

Para desarrollar nuestros modelos, optamos por utilizar redes neuronales para clasificación debido a la complejidad de los patrones presentes en nuestros datos, que incluyen imágenes, y la flexibilidad que ofrecen al trabajar con diferentes conjuntos de datos. Elegimos diferentes funciones de activación, ya que juegan un papel fundamental en la determinación de los objetivos de la red neuronal y son cruciales para obtener buenos resultados. Las funciones de activación que utilizamos son:

- ReLU (Rectified Linear Unit): ReLU es ampliamente utilizada en las capas ocultas debido a su simplicidad y efectividad. Introduce no linealidad al generar un valor de salida cero para entradas negativas y transmitir los valores positivos sin cambios como muestra la Figura 2.3 . ReLU ayuda a mitigar el problema del gradiente desvanecido, donde los gradientes se vuelven extremadamente pequeños durante la retropropagación, lo que permite una convergencia más rápida durante el entrenamiento.

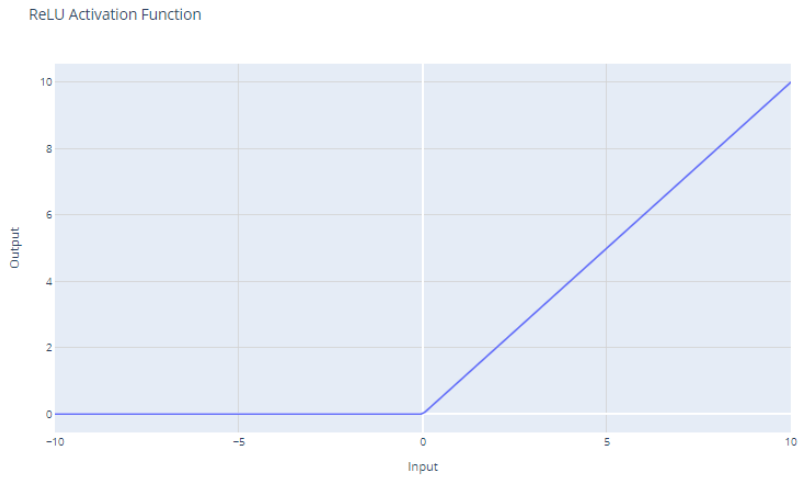


Figura 2.3: Función de activación ReLU [2]

- Softmax: Softmax se utiliza comúnmente en la capa de salida de tareas de clasificación multiclase. Convierte los puntajes de salida crudos de la red en probabilidades, asegurando que la suma de todas las probabilidades de salida sea igual a uno. Esto lo hace adecuado para tareas en las que el modelo debe asignar probabilidades a múltiples clases, facilitando así la interpretación de los resultados utilizando una función de activación como muestra la Figura 2.4.

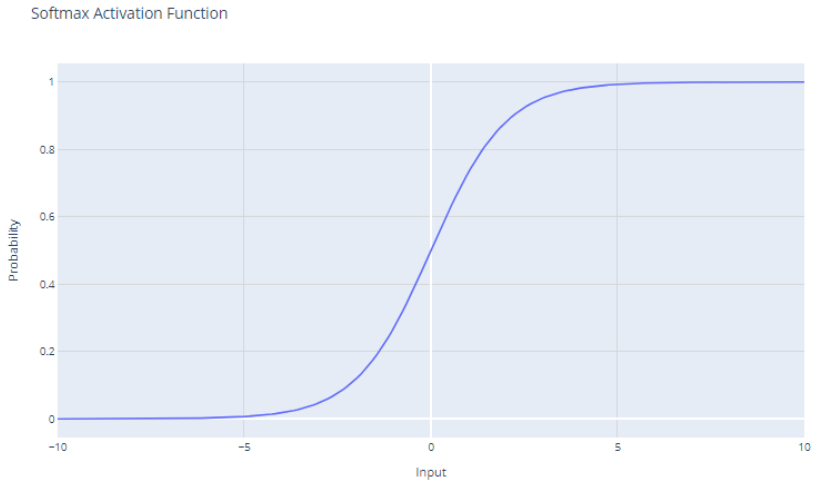


Figura 2.4: Función de activación Softmax [2]

- Lineal: En algunos casos, se utilizan funciones de activación lineales en tareas de regresión, donde la red debe predecir valores continuos. Las funciones de activación lineales preservan el valor de entrada sin transformarlo, lo que las hace apropiadas para tareas donde la salida debe ser directamente proporcional a la entrada como muestra la Figura 2.5.

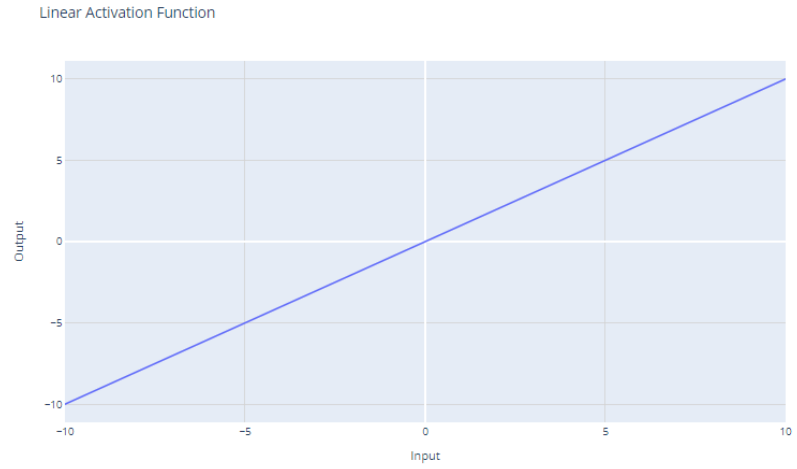


Figura 2.5: Función de activación Lineal [2]

- Tanh (Tangente Hiperbólica): Tanh es similar a la función sigmoide, pero su rango va de -1 a 1, ofreciendo una salida simétrica alrededor de cero como muestra la Figura 2.6. Introduce no linealidad y se utiliza comúnmente en las capas ocultas para manejar datos tanto negativos como positivos de manera efectiva. Tanh es especialmente útil cuando los datos de entrada están normalizados entre -1 y 1,

ya que ayuda a prevenir la saturación de los gradientes y facilita el aprendizaje.

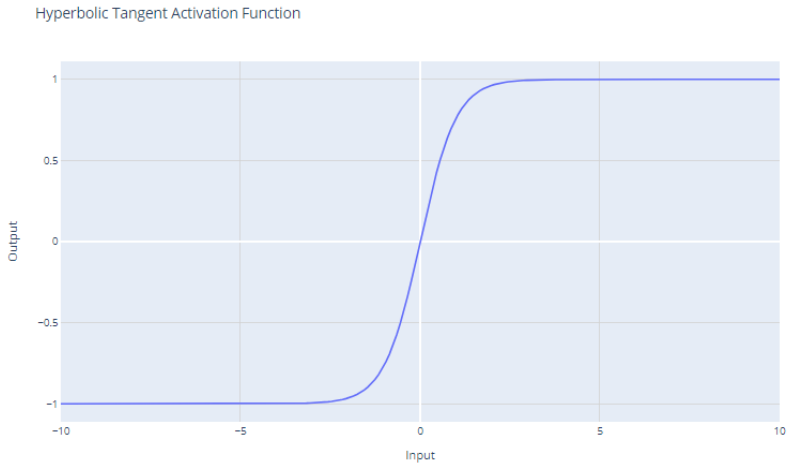


Figura 2.6: Función de activación Tanh [2]

Además de las redes neuronales tradicionales, decidimos incorporar modelos más modernos que son ampliamente utilizados en la industria y tienen un alto rendimiento en tareas complejas. Estos modelos son:

- Random Forest: Es un modelo compuesto por varios árboles de decisión. Cada árbol se construye con un subconjunto aleatorio de los datos, aunque todos usan los mismos predictores. Esta aleatoriedad introduce variabilidad en las predicciones. Para mejorar el rendimiento, se ajusta la correlación entre los árboles. El modelo genera múltiples predicciones (m predicciones) para una sola muestra.
- Neva22b: Este modelo se utiliza debido a su capacidad para manejar grandes volúmenes de datos y su rendimiento excepcional en tareas de clasificación de imágenes complejas. Es particularmente eficiente en la extracción de características significativas a partir de datos no estructurados.
- Kosmos2: es un modelo que se destaca por su capacidad de adaptarse rápidamente a nuevos datos, lo que lo convierte en una opción ideal para aplicaciones en las que los datos cambian con frecuencia.
- Vila: es un modelo de aprendizaje profundo que sobresale en la clasificación de imágenes multiclase y en la generalización de patrones complejos.
- Fuyu: es un modelo diseñado para trabajar con grandes cantidades de datos no etiquetados y se destaca en tareas de clasificación y

segmentación. Su arquitectura permite una mayor precisión en la identificación de patrones sutiles dentro de las imágenes.

- RAG: Este modelo combina técnicas de búsqueda con generación de respuestas, lo que lo hace particularmente adecuado para aplicaciones en las que se necesita obtener información de grandes bases de datos o colecciones de imágenes. Su capacidad para generar resultados precisos y relevantes lo convierte en una herramienta valiosa en el mercado actual.
- FAISS: es una biblioteca optimizada para realizar búsquedas de vectores en grandes bases de datos. Utilizamos este modelo para mejorar la eficiencia en la recuperación de imágenes similares en tareas de clasificación y búsqueda de imágenes, un componente esencial en aplicaciones comerciales.

## 2.4 *Descripción de las métricas*

Para evaluar el rendimiento de nuestros modelos, utilizamos dos métricas clave que nos permiten medir tanto la precisión de las predicciones como la efectividad en la comparación y búsqueda de vectores dentro de la base de datos. Estas métricas son:

- Precisión (Accuracy): Es la proporción de predicciones correctas sobre el total de predicciones realizadas. En nuestro caso, utilizamos la precisión para medir qué tan bien nuestros modelos de clasificación logran predecir las etiquetas correctas en los conjuntos de datos de imágenes.
- Distancia Euclidiana: La distancia euclidiana se emplea para medir la similitud entre vectores, y es especialmente útil cuando se trabaja con representaciones vectoriales de imágenes.

## 2.5 *Descripción de los experimentos o simulaciones*

Comenzamos utilizando árboles de clasificación y evaluamos el rendimiento de las redes neuronales, donde los resultados mostraron una precisión máxima del 6 %. Ante este desempeño, se decidió utilizar el algoritmo Random Forest para intentar mejorar la precisión. A pesar de aumentar linealmente el número de estimadores, solo conseguimos una precisión máxima del 4 %. Posteriormente, se utilizaron modelos preentrenados de Nvidia, los cuales mostraron una precisión significativamente más alta en comparación con los modelos que generamos manualmente.

Después de obtener los resultados de los modelos de Nvidia como referencia, decidimos utilizar FAISS para generar un índice con los vectores de las imágenes y calcular sus distancias utilizando un índice tipo L2, que emplea la distancia euclidiana para la búsqueda de similitudes. Posteriormente, modificamos la búsqueda de vectores similares alternando el nivel de búsqueda k. Al incrementar k buscábamos más similitudes, lo que reducía la precisión; mientras que al disminuir k buscábamos menos similitudes, lo que aumentaba la precisión. Para nuestro caso general de uso, se determinó que **k = 3** era la mejor opción. En cuanto al uso de RAG con una arquitectura como se muestra en la Figura 2.7, decidimos emplear el vector con mayor similitud para la búsqueda de la raza del perro y luego generamos un manejo interno de errores para saber cuándo llamar a los modelos de Nvidia.

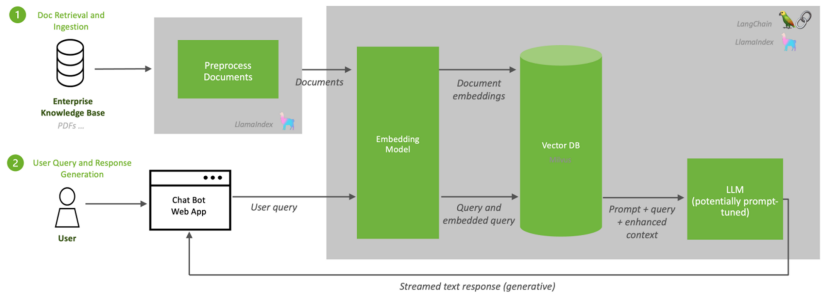


Figura 2.7: Arquitectura RAG [3]

# 3 Resultados y discusión

## 3.1 Resultados y discusión

Pudimos observar que los modelos convencionales, como los árboles de regresión y Random Forest, no fueron suficientes para mejorar la precisión del modelo. Sin embargo, al utilizar modelos preentrenados, vimos que muchos de los modelos comerciales, como Neva22b y Fuyu, no arrojaron resultados positivos. En cambio, Kosmos2 y Fuyu se desempeñaron bastante bien, como se muestra en la Tabla 3.1.

| **Modelo** | **Precisión** |
|---|---|
| NN Relu/Softmax | 0.06 |
| NN Relu/Linear | 0.01 |
| NN Tanh/Relu | 0.01 |
| RF 50 Estimadores | 0.04 |
| RF 100 Estimadores | 0.03 |
| Neva22b | 0.1 |
| Kosmos-2 | 0.8 |
| Vila | 0.9 |
| Fuyu | 0.2 |

Tabla 3.1: Precisión de diferentes modelos de clasificación

Al aplicar la búsqueda por vectores utilizando FAISS, logramos obtener una mejor precisión. Al incorporar RAG y añadir un manejo interno del error mediante la clasificación de razas, conseguimos una precisión constante en los modelos, como se muestra en la Tabla 3.2. Podemos observar que hubo una pequeña pérdida de precisión en los modelos Kosmos-2 y Vila. Esto se debe a que nuestros vectores aún requieren más entrenamiento.

| **Modelo** | **Precisión** |
|---|---|
| FAISS - VectorSearch | 0.7 |
| RAG - Neva22b | 0.7 |
| RAG - Kosmos-2 | 0.7 |
| RAG - Vila | 0.7 |
| RAG - Fuyu | 0.7 |

Tabla 3.2: Precisión de modelos con búsqueda de vectores y RAG

# 4 Conclusiones y trabajo futuro

## 4.1 Conclusiones

En general, pudimos observar cómo logramos obtener una precisión constante en los modelos a través del uso de nuestros vectores. Esto se traduce en obtener mejores resultados con menos entrenamiento, optimizando el rendimiento de nuestros modelos y, de esta manera, reduciendo el costo operativo.

Personalmente, pensé que tendríamos una precisión menor debido a las limitaciones de hardware con el que entrené los vectores de FAISS. Sin embargo, esto deja abierta la posibilidad de que, con algunas optimizaciones adicionales, podríamos lograr una precisión aún más alta. Actualmente, estos métodos están cambiando la forma en que se entrenan los LLMs, ya que permiten evitar tanto el subentrenamiento como el sobreentrenamiento mediante nuevas políticas que emplean RAG. Esto podría compararse con las optimizaciones a nivel computacional que ocurrieron después de la invención de la caché.

## 4.2 Trabajo futuro

Para obtener mejores resultados, sería necesario utilizar un entrenamiento por lotes (batch training) para mejorar los vectores de FAISS y emplear un modelo orientado a objetos para hacer el código más legible. Otro requisito importante para obtener resultados más confiables sería utilizar conjuntos de entrenamiento adicionales, lo que permitiría probar el modelo de manera más exhaustiva y generar predicciones con mayor certeza.

# *Bibliografía*


[1] J. Lucas, "Llm use cases: 574,368 % growth reshapes ai landscape," 2024. Available at: https://www.virtasant.com/ai-today/llm-use-cases-growth#:~:text=In%202019%2C%20the%20largest%20LLM,%2C%20achieving%20a%20574%2C368%25%20increase, Accessed: November 17, 2024.

[2] Datacamp, "Introducción a las funciones de activación en las redes neuronales," 2024. Available at: https://www.datacamp.com/es/tutorial/introduction-to-activation-functions-in-neural-networks, Accessed: November 29, 2024.

[3] R. Merritt, "What is retrieval-augmented generation, aka rag," 2023. Available at: https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/, April 19, 2024.

[4] FacebookResearch, "Faiss (facebook ai similarity search)," 2024. Available at: https://ai.meta.com/tools/faiss/, Accessed: November 29, 2024.

[5] NvidiaFoundations, "Neva-22b," 2024. Available at: https://build.nvidia.com/nvidia/neva-22b, Accessed: November 29, 2024.

[6] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei, "Kosmos-2: Grounding multimodal large language models to the world," 2023. Available at: https://www.microsoft.com/en-us/research/publication/kosmos-2-grounding-multimodal-large-language-models-to-the-world/, Accessed: November 29, 2024.

[7] NvidiaFoundations, "Vila," 2024. Available at: https://build.nvidia.com/nvidia/vila, Accessed: November 29, 2024.

[8] R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, and S. Taşırlar, "Vila," 2023. Available at: https://www.adept.ai/blog/fuyu-8b, Accessed: November 29, 2024.

[9] DocsPython, “math - mathematical functions,” 2024. Available at: https://docs.python.org/3/library/math.html#math.isnan, Accessed: November 29, 2024.