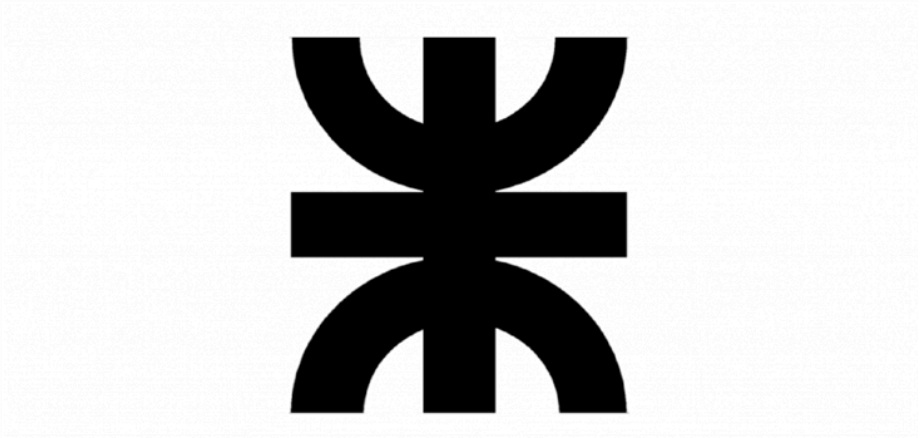
UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL CÓRDOBA



**DEPARTAMENTO DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN**

# IA - Inteligencia Artificial

---

**2do Cuatrimestre 2025**

**Curso 5K2**

## Integrantes Grupo N° 2:


- Bordino Coniglio, Tobías Martín - 93611
- Caffaro, Santiago - 90364
- Moreno, Tomás Agustín - 90365
- Suarez, Emiliano Fabricio - 91134

# Índice

# Análisis del Dataset

El objetivo de este apartado es comprender las características principales del conjunto de datos utilizado denominado **Pascal VOC 2012** , tanto desde el punto de vista cuantitativo (dimensiones, tamaños, canales, etc.) como desde su distribución de clases y su estructura visual. Se realizan tres etapas: cálculo de estadísticas básicas, generación de visualizaciones descriptivas y un análisis exploratorio mediante clustering no supervisado.

## Estadísticas básicas

En primer lugar, se calculan atributos relevantes para cada imagen, como el nombre del archivo, las dimensiones en píxeles, el número de canales de color y el tamaño en disco.

| | filename | width | height | channels | size_kb |
|---|---|---|---|---|---|
| 0 | 2007_005304.jpg | 500 | 400 | 3 | 69.343750 |
| 1 | 2009_002094.jpg | 500 | 375 | 3 | 120.733398 |
| 2 | 2011_006475.jpg | 500 | 375 | 3 | 158.150391 |
| 3 | 2011_000611.jpg | 500 | 375 | 3 | 131.470703 |
| 4 | 2009_004042.jpg | 500 | 375 | 3 | 147.988281 |

```
Cantidad total de imágenes: 17125
              width        height       size_kb
count  17125.000000  17125.000000  17125.000000
mean     466.797547    389.507620    109.604073
std       61.931367     65.497125     45.656307
min      142.000000     71.000000      7.147461
25%      499.000000    338.000000     77.677734
50%      500.000000    375.000000    109.863281
75%      500.000000    400.000000    138.762695
max      500.000000    500.000000    835.206055
```

El dataset contiene un total de 17125 imágenes. Las mismas presentan una estructura relativamente homogénea: Ancho promedio de 500 píxeles, altura promedio de 390 píxeles y un tamaño promedio de 110KB con una gran dispersión entre la imagen más pequeña (7KB) y (835KB).

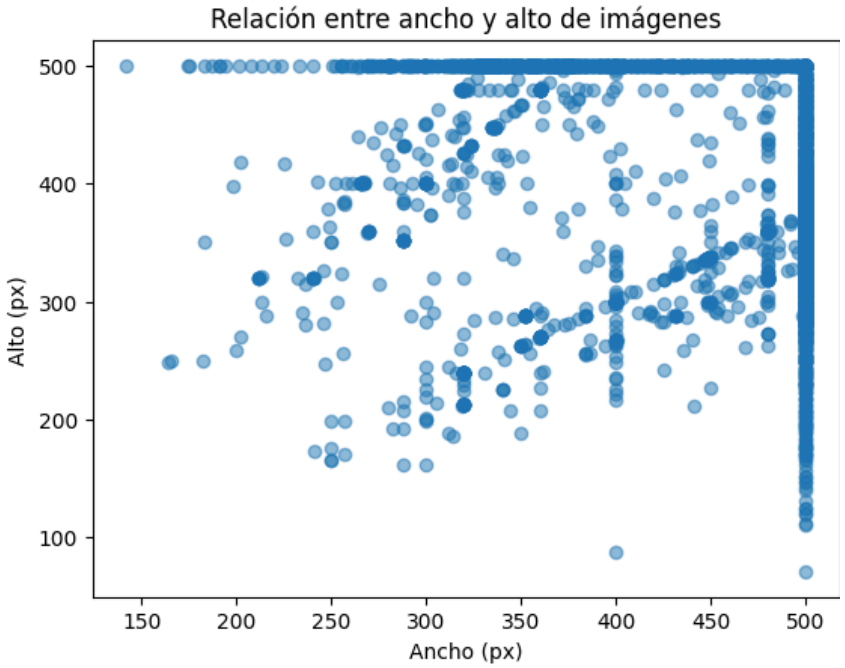
Esto sugiere un formato de entrada estandarizado, posiblemente con ancho fijo y altura variable según la proporción original. La variabilidad en el tamaño puede atribuirse al nivel de detalle y la compresión JPEG utilizada.

# Visualizaciones relevantes

Para explorar la distribución de características se generan los siguientes gráficos:

## Gráfico de dispersión

Muestra la correspondencia entre el ancho y el alto de cada imagen en píxeles.



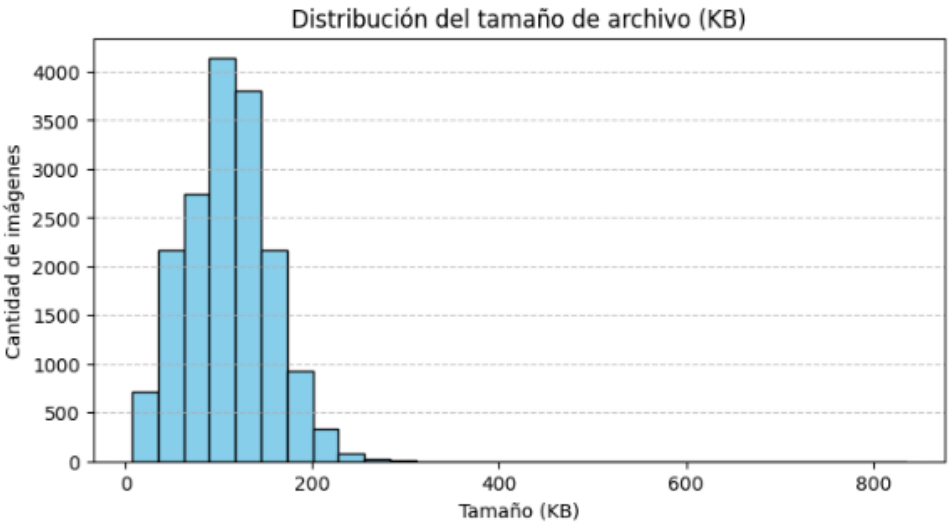

La mayoría de los puntos se agrupan en torno a un ancho de 500 píxeles, lo que indica que muchas imágenes comparten una misma dimensión horizontal. La altura, en cambio, presenta una variabilidad mayor, con valores que oscilan principalmente entre 300 y 500 píxeles.

Se observa una franja densa en la parte superior del gráfico (altura ≈ 500px), lo que sugiere que varias imágenes son cuadradas o recortadas al tamaño máximo permitido.

Algunas imágenes presentan alturas atípicamente bajas (por debajo de 150px), que podrían corresponder a imágenes con proporciones inusuales o errores en el procesamiento.

## Histograma

Representa la distribución del tamaño de los archivos de imagen en kilobytes (KB).



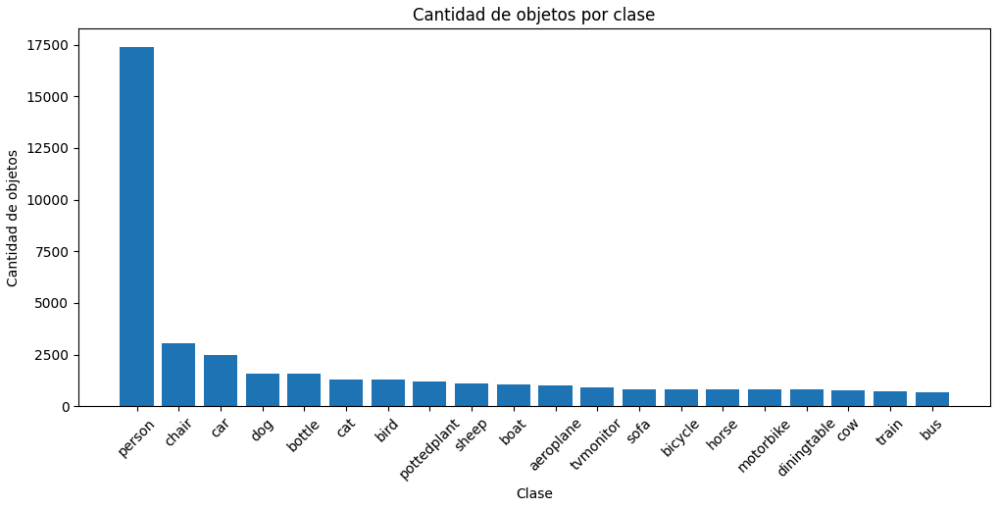

La mayor parte de los archivos se concentran entre 50 KB y 200 KB, con una clara asimetría positiva (cola hacia la derecha), lo que indica que la mayoría de las imágenes son livianas, pero existen algunos casos con tamaños mucho mayores.

El valor modal (pico máximo) se ubica en torno a los 100–120 KB, correspondiente a una compresión estándar de imágenes JPEG con buena calidad visual.

Algunos archivos superan los 800 KB, lo que probablemente esté asociado a menor compresión o a contenido visual más complejo (por ejemplo, imágenes con muchos detalles, texturas o colores).

## Distribución por clases

A partir de los archivos XML de anotaciones, se extrae la frecuencia de aparición de cada clase de objeto:

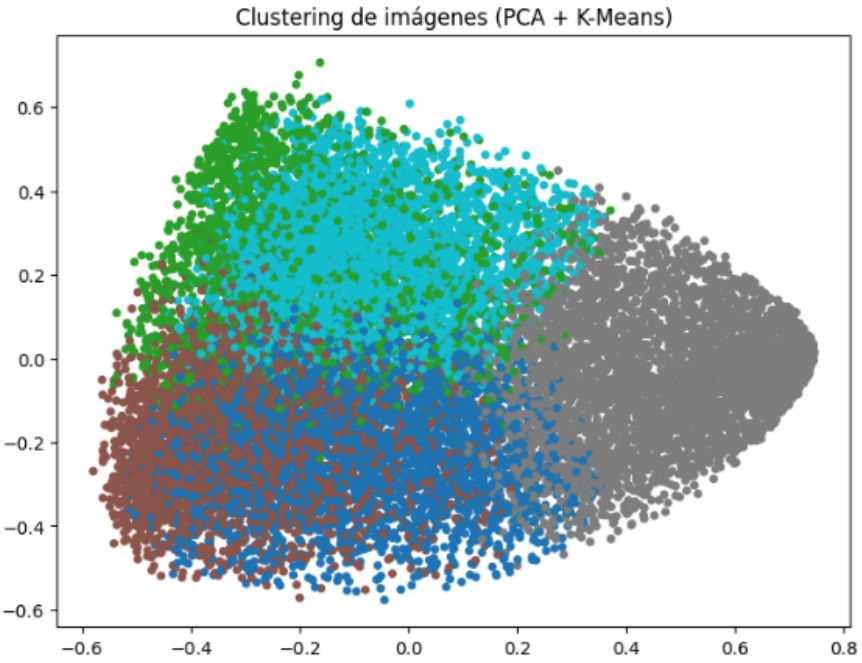


La clase "person" domina claramente con más de 17,000 instancias, seguida de chair, car y dog.

# Clustering exploratorio

## Basado en Histogramas de Color

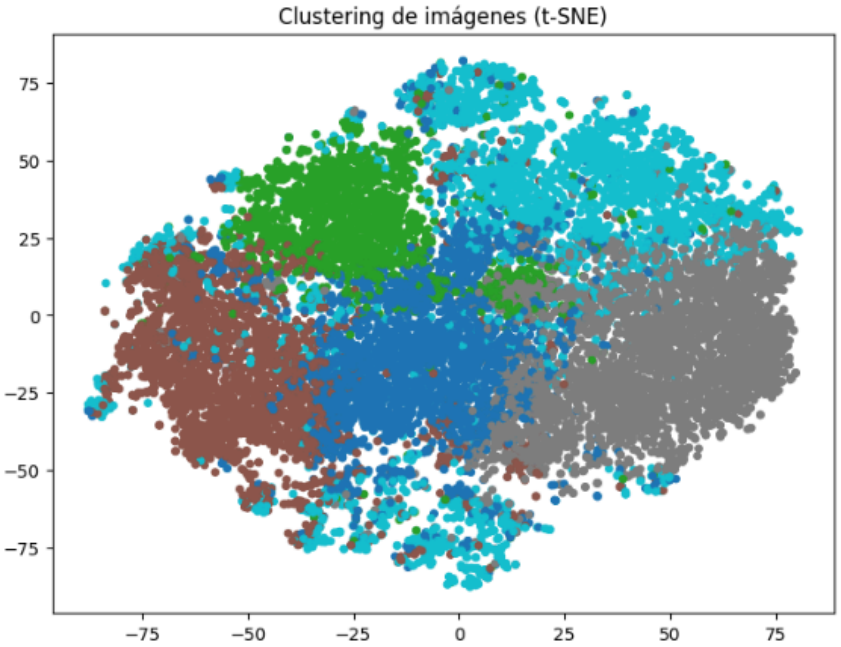
Para explorar la estructura visual del conjunto de imágenes sin utilizar etiquetas, se empleó un enfoque no supervisado basado en histogramas de color. Cada imagen fue representada mediante un vector de características construido a partir de histogramas RGB normalizados, lo que permitió describir la distribución cromática de sus píxeles. Posteriormente, se aplicó **K-Means** (k=5) para agrupar las imágenes según su similitud visual, y las dimensiones fueron reducidas mediante **PCA** y **t-SNE** para facilitar la visualización.




Se observan cinco agrupaciones principales (representadas con distintos colores), lo que confirma que el algoritmo fue capaz de separar el espacio de características en regiones relativamente bien definidas.

Sin embargo, los bordes entre grupos no son completamente nítidos, indicando cierta superposición entre clusters. Esto es esperable, ya que las características utilizadas (histogramas de color) capturan solo información cromática y no semántica.

El **PCA** mantiene las relaciones lineales y permite ver que algunos clusters (por ejemplo, los grises y celestes) presentan una transición suave entre sí, lo que podría corresponder a imágenes con paletas de color similares.




En este caso, las agrupaciones aparecen más compactas y mejor diferenciadas, evidenciando que **t-SNE** logra resaltar las afinidades locales entre imágenes.

Algunos clusters (por ejemplo, el verde y el marrón) se separan claramente, lo que indica que existen subconjuntos de imágenes con características cromáticas y texturales muy específicas.

Se aprecian también zonas de mezcla parcial, coherentes con transiciones entre tipos de contenido visual (por ejemplo, entre fondos naturales y objetos artificiales).

Este análisis exploratorio sugiere que el dataset contiene subgrupos coherentes de imágenes, determinados principalmente por la distribución del color y la textura.

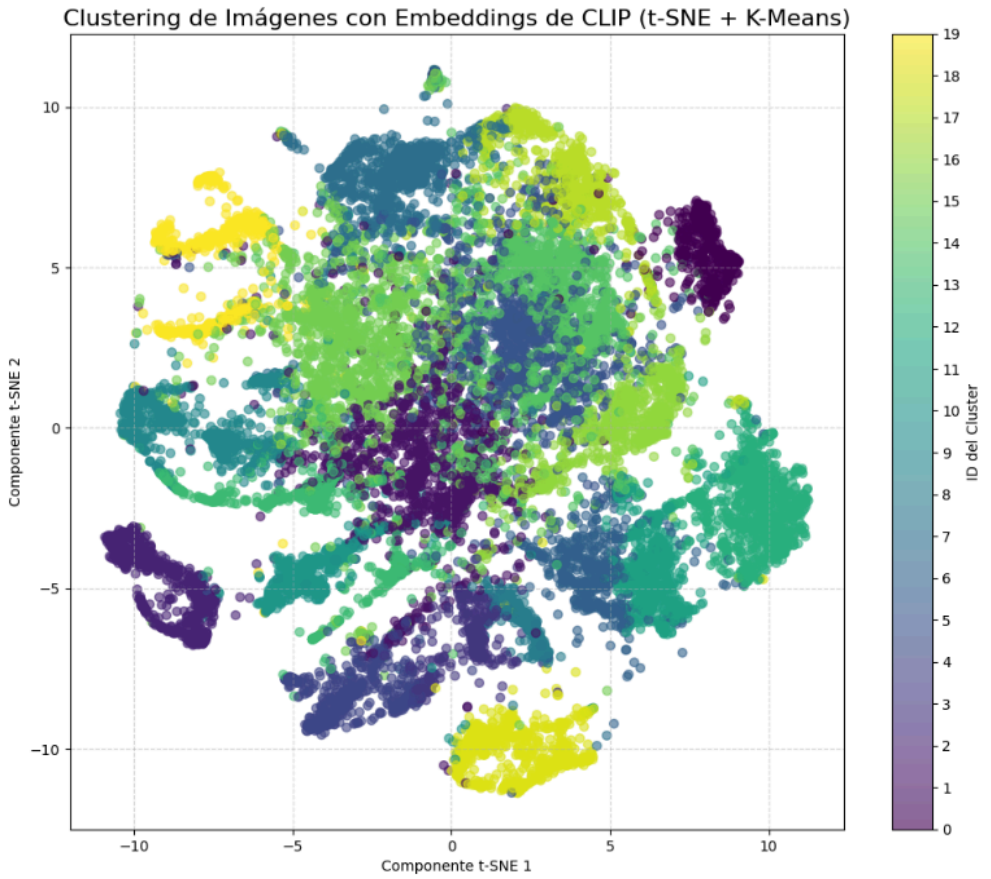
No obstante, al no incorporar información semántica (como objetos o etiquetas), los clusters no se alinean necesariamente con las clases reales del conjunto de datos.

## Basado en Embeddings

Podemos transformar las imágenes en embeddings para representarlas de una forma más compacta y significativa dentro de un espacio numérico. Estos embeddings condensan la información visual y semántica de cada imagen, permitiendo comparar y agrupar contenidos de manera más inteligente. A diferencia de los histogramas de color (que solo describen la distribución cromática), los embeddings generados por modelos como **CLIP** capturan patrones visuales más profundos, como formas, texturas y relaciones conceptuales (por ejemplo, entender que un “auto rojo” y un “auto azul” son instancias del mismo objeto).

De esta forma, realizar una clusterización sobre embeddings permite agrupar imágenes por similitud de contenido, no solo por color o apariencia superficial. Esto produce grupos más

coherentes desde el punto de vista semántico y reduce la influencia de factores como la iluminación o el fondo.




Los colores corresponden a los 20 clusters obtenidos con **K-Means**, donde cada color representa un grupo distinto de imágenes con características compartidas. Se observan zonas donde los puntos del mismo color se concentran con claridad (indicando agrupaciones bien definidas) y otras donde los colores se mezclan parcialmente, reflejando transiciones suaves entre categorías.

# Metodología

En esta etapa se utilizan los embeddings de imágenes generados con CLIP, que representan de forma compacta y semántica el contenido visual. A partir de ellos, se implementan dos pipelines: uno de búsqueda básica, basado en la similitud directa entre consultas e imágenes, y otro de búsqueda avanzada, que incorpora reformulación con modelo de lenguaje y reranking para interpretar consultas complejas y mejorar la precisión de los resultados.

## Baseline

En la primera etapa, denominada búsqueda baseline, se implementa un motor de recuperación basado en el modelo **CLIP** (Contrastive Language–Image Pre-training) y el índice **FAISS**. **CLIP** se encarga de transformar tanto las imágenes como las consultas textuales en un mismo espacio de representación vectorial, donde la similitud entre vectores refleja la coherencia semántica entre ambos dominios. Los embeddings de las imágenes se generan previamente y se almacenan en un índice **FAISS** de tipo **IndexFlatIP**, que utiliza el producto interno como medida de similitud. Como los vectores se normalizan con **L2**, este producto interno equivale a la similitud de coseno. Cuando el usuario introduce una consulta textual, el sistema genera su embedding correspondiente mediante **CLIP**, lo normaliza y realiza una búsqueda de los top-k vectores más similares dentro del índice, devolviendo las imágenes más relacionadas semánticamente con la consulta. Esta etapa constituye una línea base simple, eficiente y totalmente determinista, adecuada para consultas directas y sin ambigüedades.

## Reformulación con LLM

La segunda etapa introduce un componente de comprensión semántica mediante un proceso de reformulación con modelo de lenguaje (**LLM**). Dado que los modelos de recuperación visual no manejan bien expresiones complejas, idiomas distintos al inglés o frases con negaciones, se integra un pipeline que primero traduce el texto de entrada al inglés utilizando el modelo **Helsinki**, seleccionado por su alta precisión en traducciones automáticas multilingües y su eficiencia al manejar estructuras gramaticales complejas.

Posteriormente, se aplican reglas lingüísticas para detectar posibles estructuras de negación (como “no”, “sin”, “not”, “without”, “except”, “non-”). Luego, se utiliza el modelo **TinyLlama** para producir un objeto JSON con dos campos clave: main_term, que representa el concepto central de búsqueda, y negative_term, que representa aquello que el usuario desea excluir. **TinyLlama** se eligió por su ligereza y velocidad de inferencia, lo que permite ejecutar el proceso de reformulación de manera eficiente en entornos con recursos limitados, manteniendo una calidad semántica competitiva respecto a modelos de mayor tamaño.

Por ejemplo, para la consulta “perro sin correa”, el sistema extrae main_term = "dog" y negative_term = "leash". Si no se detecta ninguna negación, el campo negative_term permanece vacío y la búsqueda continúa de forma estándar. Esta etapa permite reinterpretar consultas ambiguas o en lenguaje natural y estructurarlas en un formato controlado, mejorando la comprensión del modelo.

## Reranking

Finalmente, en la tercera etapa, denominada búsqueda con reranking, se combina la información reformulada para refinar los resultados. Primero, se realiza una búsqueda ampliada en **FAISS** utilizando únicamente el término principal (main_term), recuperando un

número mayor de candidatos que el solicitado inicialmente. Este aumento, controlado mediante el parámetro **recall_multiplier**, asegura que se consideren suficientes imágenes potencialmente relevantes antes del filtrado. Luego, se calculan las similitudes entre cada imagen candidata y los embeddings tanto del término principal como del término negativo, y se ajusta la puntuación final según la expresión:

$$s = s_{main} - \lambda_{neg} \times s_{neg}.$$

Donde *smain* es la similitud con el término principal, *sneg* la similitud con el término a excluir, y *λneg* un factor de penalización. Las imágenes se reordenan según esta puntuación corregida, de modo que aquellas que se asemejen al término negativo sean penalizadas, y las más alineadas con el concepto principal queden en las primeras posiciones.

Este mecanismo permite que el sistema interprete correctamente consultas con negaciones, reduzca falsos positivos y genere un ranking final más coherente con la intención del usuario.

## Decisiones de diseño

**Prompt LLM:** El prompt utilizado para **TinyLlama** está diseñado para guiar al modelo de manera precisa y minimizar la ambigüedad en su salida. Se compone de dos partes principales: un mensaje de sistema que establece las reglas de comportamiento y el texto traducido del usuario como entrada. En el mensaje de sistema se instruye al modelo a convertir una consulta de búsqueda en inglés en un objeto JSON que contenga exactamente dos claves: main_term y negative_term. Se le indica que si existe una negación en la consulta (por ejemplo, en expresiones como “a non-red car” o “car not red”) debe colocar el término negado en negative_term; de lo contrario, este campo debe permanecer vacío. Además, se enfatiza que no debe agregar ningún texto antes o después del JSON, garantizando así una salida limpia y estructurada.

El formato del prompt sigue la convención de entrenamiento de **TinyLlama**, empleando las etiquetas <|system|>, <|user|> y <|assistant|> para delimitar los roles conversacionales, lo que mejora la consistencia de la respuesta.

**Índice FAISS:** se seleccionó **IndexFlatIP** con embeddings normalizados, lo que permite que el producto interno equivalga a la similitud de coseno. Esta configuración ofrece resultados exactos (no aproximados) y un comportamiento determinista adecuado para conjuntos de datos medianos.

**Modelo de embeddings:** El modelo seleccionado para la generación de embeddings fue **CLIP ViT-B/32**, desarrollado por OpenAI. Esta arquitectura combina una red visual basada en Vision Transformer (ViT) con un codificador textual entrenado conjuntamente mediante aprendizaje contrastivo. El resultado es un espacio semántico compartido donde imágenes y texto pueden compararse directamente.

La variante **ViT-B/32** fue elegida por su balance entre rendimiento y costo computacional. Modelos más grandes (como **ViT-L/14**) ofrecen una ganancia marginal en precisión pero

requieren recursos significativamente mayores. **ViT-B/32** produce embeddings de 512 dimensiones, lo que proporciona una representación suficientemente rica para capturar relaciones semánticas sin generar un costo excesivo en memoria o en tiempo de inferencia.

**Normalización L2:** Se optó por la normalización **L2** porque es la más adecuada cuando se utiliza **FAISS** con producto interno (**IndexFlatIP**), ya que convierte el producto punto entre vectores en una medida directamente equivalente a la similitud de coseno, la más común para comparar embeddings de **CLIP**. Otras normalizaciones (como L1 o min–max) no preservan la dirección del vector ni la proporcionalidad entre componentes, lo que distorsionaría la geometría del espacio semántico y afectaría la coherencia de las comparaciones. La **L2**, en cambio, mantiene la orientación del vector (que es lo que realmente codifica la información semántica) y elimina solo la influencia de su magnitud. De esta forma, dos embeddings con distinta intensidad numérica pero significado similar seguirán considerándose cercanos. Además, esta normalización es computacionalmente eficiente, está implementada de forma nativa en **FAISS** y es el estándar en la mayoría de sistemas basados en similitud de coseno, lo que garantiza estabilidad, interpretabilidad y compatibilidad con el modelo **CLIP**.

**Parámetros del reranking:** El reranking constituye la etapa final del pipeline, en la cual los resultados iniciales recuperados por **FAISS** se reordenan para reflejar con mayor fidelidad la intención del usuario, especialmente en consultas que incluyen negaciones o exclusiones semánticas.

En esta etapa se establecieron los valores **recall_multiplier** = **5** y **lambda_neg** = **0.8**, seleccionados tras un análisis orientado a equilibrar precisión y eficiencia.

El valor de **recall_multiplier** = **5** amplía significativamente el conjunto de candidatos recuperados antes del reranking (quintuplicando el número solicitado por el usuario) , lo que permite compensar posibles omisiones del modelo **CLIP** ante consultas complejas o con términos negados. Este factor asegura una exploración más amplia del espacio de búsqueda sin aumentar de forma considerable la latencia, manteniendo un compromiso adecuado entre cobertura y rendimiento.

Por su parte, se utilizó el parámetro **lambda_neg** = **0.8**, la intención fue aplicar una penalización más fuerte a los resultados asociados al término negativo, priorizando la exclusión estricta sobre la flexibilidad semántica. Este valor se adoptó en escenarios de prueba donde las negaciones eran muy claras (por ejemplo, consultas como “auto no rojo” o “persona sin máscara”) y el modelo baseline tendía a incluir numerosos falsos positivos (imágenes con el atributo que debía excluirse). Al fijar **lambda_neg = 0.8**, la similitud con el término negado adquiere un peso mayor en la resta del puntaje final, desplazando de las primeras posiciones las imágenes que presentan una correlación alta con dicho atributo.

**Fallback de robustez:** se implementó un mecanismo de respaldo que garantiza que el sistema siempre devuelva una respuesta válida, incluso si el modelo de lenguaje **TinyLlama** falla o produce un resultado incorrecto.

Si **TinyLlama** no logra generar el JSON esperado o no detecta correctamente una negación, el pipeline recurre automáticamente a las reglas lingüísticas aplicadas previamente (como “no X”, “sin X”, “without X”) para reconstruir la consulta en forma estructurada. De este modo, ante frases como “auto no rojo” o “perro sin correa”, el sistema puede seguir funcionando correctamente y devolver pares coherentes como {"main_term": "car", "negative_term": "red"} o {"main_term": "dog", "negative_term": "leash"}, asegurando continuidad operativa y coherencia semántica aunque el modelo principal falle.

# Resultados

Los resultados muestran diferencias sustanciales entre ambos métodos. La búsqueda baseline mantiene una mayor velocidad y simplicidad, pero sufre caídas de precisión ante consultas con negaciones o formulaciones ambiguas. La búsqueda con reformulación y reranking presenta una mejora notable en la precisión y el ordenamiento de resultados relevantes, especialmente en consultas con exclusiones explícitas o semántica compleja, a costa de un incremento moderado en la latencia debido al procesamiento adicional de reformulación y reranking. En términos de comportamiento general, la versión reformulada logra un mejor equilibrio entre recall y precisión, dado que amplía el conjunto inicial de candidatos y filtra posteriormente los falsos positivos.

## Comparativas visuales Baseline vs Reformulación

En los presentes ejemplos, se visualiza el comportamiento comparativo entre el sistema baseline (**CLIP** + **FAISS**) y la versión avanzada con reformulación y reranking, ante consultas que incorporan negaciones o exclusiones semánticas, tales como “un auto no rojo”, “perro no negro” y “moto sin persona”.

En el caso de la consulta “un auto no rojo”, el sistema baseline devuelve principalmente automóviles de color rojo o con tonos similares. Esto ocurre porque el modelo no interpreta la negación presente en el texto; el embedding generado para la frase “un auto no rojo” conserva un peso semántico dominante sobre el término “auto”, sin diferenciar el modificador “no rojo”. En consecuencia, el motor de búsqueda prioriza imágenes con alta similitud al concepto de “auto”, incluso cuando exhiben el color que debía ser excluido, generando múltiples falsos positivos.

Por el contrario, la versión Reformulada + Reranking implementa un pipeline extendido que integra un modelo de lenguaje (**LLM**) para la reformulación semántica de la consulta y un mecanismo de penalización vectorial sobre el término negado. El proceso detecta automáticamente la estructura negativa (“no rojo”), identifica dos componentes:

- main_term: el concepto principal a buscar (por ejemplo, “auto”)
- negative_term: el atributo a excluir (por ejemplo, “rojo”)

A partir de esta detección, el sistema realiza primero una búsqueda ampliada utilizando únicamente el término principal y, posteriormente, aplica una penalización proporcional a la similitud con el término excluido. En este trabajo se utilizó una penalización de **λ = 0.8**, desplazando hacia posiciones inferiores las imágenes más asociadas al color rojo.

Visualmente, se observa un cambio notorio: los autos de color rojo desaparecen de las primeras posiciones y son reemplazados por vehículos de colores blanco, gris, negro y verde, lo cual refleja una mejor alineación entre la intención del usuario y los resultados obtenidos.


En la consulta “perro no negro”, el fenómeno se repite. El sistema baseline incluye perros negros en varias de las primeras posiciones del ranking, demostrando nuevamente su incapacidad para manejar negaciones. Sin embargo, el sistema avanzado logra penalizar correctamente las imágenes con tonos oscuros, priorizando perros de color blanco, marrón o beige. Esto confirma que la combinación de reformulación lingüística y penalización vectorial permite al modelo interpretar adecuadamente el contexto negativo de la consulta.

Finalmente, en la consulta “moto sin persona”, el baseline tiende a devolver motocicletas con personas visibles, ya que el modelo asocia fuertemente el concepto “moto” con escenas típicas del dataset donde hay individuos conduciendo o posando junto al vehículo.

La versión reformulada, en cambio, detecta la estructura “sin persona”, identifica main_term = moto y negative_term = persona, y aplica la misma estrategia de penalización semántica.

El resultado es un conjunto de imágenes mucho más coherente con la intención de búsqueda, mostrando motocicletas sin presencia humana o con personas parcialmente visibles y en planos secundarios.


En conjunto, estos tres ejemplos demuestran las limitaciones semánticas del modelo base frente a consultas con negaciones o condiciones compuestas, y a su vez, evidencian la efectividad del enfoque reformulación + reranking para corregir dichos sesgos.

## Métricas cuantitativas

Para validar objetivamente el rendimiento de los sistemas, la evaluación se dividió en dos partes. Primero se definieron cuatro métricas estándar de Information Retrieval (IR). Posteriormente se calcularon sobre los resultados arrojados por los modelos tanto para consultas simples como para consultas complejas.

**Precisión promedio (Average Precision - AP):** Mide la calidad del ranking teniendo en cuenta tanto la cantidad de imágenes relevantes recuperadas como la posición en la que aparecen. Un valor alto de AP indica que el sistema ordena correctamente las imágenes relevantes al principio del listado. Se utilizó AP@10 para evaluar las primeras 10 imágenes.

**Precisión (Precision@10):** Representa la proporción de imágenes relevantes dentro de las primeras 10 recuperadas. Indica la exactitud del sistema, es decir, qué porcentaje de los resultados mostrados son correctos. Un valor alto de precisión implica pocos falsos

positivos.

**Cobertura o Recall (Recall@10):** Indica la proporción de imágenes relevantes totales que el sistema logra recuperar dentro del top-10. Mide la capacidad del modelo para encontrar todos los resultados relevantes. Como se evalúa sólo sobre las primeras posiciones el recall tiende a ser bajo aunque el sistema tenga buen desempeño global.

**F1-Score (F1@10):** Es una métrica combinada que equilibra precisión y recall mediante su media armónica. Un valor alto de F1 indica que el sistema mantiene un buen balance entre exactitud y cobertura.

A continuación se presentan los resultados de las métricas, que muestran el desempeño de los sistemas baseline y avanzado al evaluar las consultas simples correspondientes a las clases del dataset Pascal VOC, considerando únicamente las 10 primeras imágenes recuperadas para cada consulta.

Este tipo de evaluación busca medir la calidad del ranking en las primeras posiciones, que es donde realmente se percibe la relevancia desde la perspectiva del usuario.

| Métrica | Baseline | Avanzado | Δ (A - B) |
|---|---|---|---|
| AP@10 | 0.9786 | 0.9528 | -0.0259 |
| Prec@10 | 0.9900 | 0.9700 | -0.0200 |
| Rec@10 | 0.0147 | 0.0145 | -0.0003 |
| F1@10 | 0.0290 | 0.0284 | -0.0005 |

Estos valores indican que ambos sistemas mantienen un rendimiento alto y estable, con una ligera ventaja del modelo Baseline.

Las diferencias negativas pequeñas (Δ) reflejan que el modelo avanzado tiende a obtener resultados muy similares al Baseline, aunque en algunos casos ordena de forma ligeramente menos precisa las imágenes relevantes dentro del top-10.

| Clase | AP@10Baseline | AP@10Avanzado | Prec@10Baseline | Prec@10Avanzado | Rec@10Baseline | Rec@10Avanzado | F1@10Baseline | F1@10Avanzado |
|---|---|---|---|---|---|---|---|---|
| person | 1.0000 | 0.8900 | 1.0000 | 0.9000 | 0.0010 | 0.0009 | 0.0021 | 0.0019 |
| sofa | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0135 | 0.0135 | 0.0266 | 0.0266 |
| bottle | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0123 | 0.0123 | 0.0243 | 0.0243 |
| tvmonitor | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0155 | 0.0155 | 0.0305 | 0.0305 |
| cat | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0089 | 0.0089 | 0.0176 | 0.0176 |
| pottedplant | 1.0000 | 0.8789 | 1.0000 | 0.9000 | 0.0163 | 0.0147 | 0.0321 | 0.0289 |
| horse | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0190 | 0.0190 | 0.0373 | 0.0373 |
| car | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0078 | 0.0078 | 0.0155 | 0.0155 |
| dog | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0075 | 0.0075 | 0.0148 | 0.0148 |
| train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0170 | 0.0170 | 0.0334 | 0.0334 |
| bicycle | 0.8154 | 1.0000 | 0.9000 | 1.0000 | 0.0149 | 0.0166 | 0.0294 | 0.0326 |
| aeroplane | 1.0000 | 0.8354 | 1.0000 | 0.9000 | 0.0140 | 0.0126 | 0.0275 | 0.0248 |
| diningtable | 0.7571 | 1.0000 | 0.9000 | 1.0000 | 0.0130 | 0.0145 | 0.0257 | 0.0285 |
| motorbike | 1.0000 | 0.4509 | 1.0000 | 0.7000 | 0.0174 | 0.0122 | 0.0342 | 0.0239 |
| chair | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0073 | 0.0073 | 0.0145 | 0.0145 |
| cow | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0294 | 0.0294 | 0.0571 | 0.0571 |
| bus | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0214 | 0.0214 | 0.0419 | 0.0419 |
| bird | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0123 | 0.0123 | 0.0244 | 0.0244 |
| boat | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0182 | 0.0182 | 0.0358 | 0.0358 |
| sheep | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0280 | 0.0280 | 0.0545 | 0.0545 |

Al revisar los resultados por clase se aprecia que en la mayoría de las categorías (por ejemplo, person, sofa, dog, cat, car, train), los valores de AP@10 y Precision son muy altos (cercanos a 1.0) en ambos sistemas.

Esto demuestra que el modelo Baseline ya tiene un desempeño excelente en consultas simples, donde el texto coincide directamente con una categoría del dataset.

Solo en algunas clases aisladas, como “pottedplant” o “motorbike”, el modelo avanzado presenta pequeñas variaciones negativas, posiblemente debido a que el reranking reordenó algunos resultados relevantes fuera del top-10.

Como conclusión, los resultados indican que el sistema *Baseline* ya logra un nivel de precisión casi perfecto en las consultas simples, y que el modelo avanzado no aporta mejoras significativas en este escenario.

Esto se explica porque las consultas son directas y semánticamente claras, por lo que el modelo inicial ya identifica fácilmente las imágenes correctas.

El reranking, diseñado para consultas más complejas o con negaciones, no tiene impacto positivo en este tipo de búsquedas básicas e incluso puede alterar levemente el orden ideal del ranking.

Posteriormente, se evaluó el desempeño de los sistemas Baseline y avanzado ante consultas complejas, es decir, aquellas que combinan condiciones, negaciones o relaciones entre objetos (por ejemplo: “un auto no rojo”, “un gato junto con un perro negro”). Para lograr esto se construyó un conjunto de ground truth para consultas complejas, que contiene imágenes correspondientes a distintas descripciones con negaciones o relaciones entre objetos.

A diferencia de las consultas simples, estas pruebas buscan medir la capacidad del sistema para interpretar consultas más semánticas y menos literales, lo que pone a prueba su comprensión contextual.

| Métrica | Baseline | Avanzado | Δ (A - B) |
|---|---|---|---|
| AP@10 | 1.0000 | 1.0000 | +0.2679 |
| Prec@10 | 0.1500 | 0.4500 | +0.3000 |
| Rec@10 | 0.1528 | 0.4556 | +0.3028 |
| F1@10 | 0.0545 | 0.0545 | +0.3013 |

El modelo avanzado muestra un aumento importante en todas las métricas, especialmente en Precisión y Recall, con incrementos de aproximadamente +30% respecto al Baseline.

Esto indica que el reranking y la reformulación de consultas mejoran significativamente la capacidad del sistema para identificar resultados relevantes dentro de las primeras 10 posiciones.

Resumen por consulta (ordenado por ΔAP10):

| Consulta | APBaseline | APAvanzado | PrecBaseline | PrecAvanzado | RecBaseline | RecAvanzado | F1Baseline | F1Avanzado |
|---|---|---|---|---|---|---|---|---|
| un auto no rojo | 0.0000 | 0.1619 | 0.0000 | 0.3000 | 0.0000 | 0.3000 | 0.0000 | 0.3000 |
| un barco no rojo | 0.0100 | 0.9000 | 0.1000 | 0.9000 | 0.1000 | 0.9000 | 0.1000 | 0.9000 |
| un gato junto con un perro negro | 0.3071 | 0.2095 | 0.4000 | 0.4000 | 0.4000 | 0.4000 | 0.4000 | 0.4000 |
| a non-black cat | 0.0123 | 0.1296 | 0.1000 | 0.2000 | 0.1111 | 0.2222 | 0.1053 | 0.2105 |

En la tabla individual se puede destacar que “un auto no rojo”: pasa de AP=0.0000 en el baseline a AP=0.1619 en el avanzado, mostrando una mejora significativa. El sistema con reranking logra detectar y excluir correctamente autos rojos, identificando más casos relevantes de autos de otros colores.

Después en la consulta de “un barco no rojo”: el AP aumenta drásticamente (de 0.0100 a 0.9000), indicando que el modelo avanzado interpreta bien la negación y recupera imágenes coherentes con la condición “no rojo”.

Luego, en “un gato junto con un perro negro”: el rendimiento se mantiene estable, lo cual es esperable dada la complejidad relacional de la consulta (dos objetos y un atributo de color).

Por último, con respecto a la consulta “a non-black cat”: el modelo avanzado logra captar la negación parcial, subiendo el AP de 0.0123 a 0.1296, lo que demuestra una mejor comprensión semántica en inglés.

En conjunto, las mejoras son consistentes con el objetivo del sistema avanzado, se manejan mejor las consultas con negaciones o atributos combinados, donde el baseline no logra filtrar adecuadamente los términos contradictorios.

Como conclusión, el análisis de las consultas complejas muestra que el modelo avanzado supera al baseline en todas las métricas clave, especialmente en Precision y Recall, lo que indica una mejor comprensión semántica y capacidad de refinamiento en el top-10.

Mientras que el sistema base tiende a devolver resultados visualmente similares sin entender las restricciones lingüísticas, el reranking avanzado interpreta con mayor precisión las condiciones de la consulta (como exclusiones de color o combinaciones de objetos).

## Ventajas

1. La métrica de Average Precision (AP) es ampliamente utilizada en tareas de recuperación de información, ya que permiten evaluar tanto la precisión como la posición de los elementos relevantes dentro del ranking de resultados. En este caso, su empleo resulta apropiado porque el objetivo principal no es clasificar imágenes en categorías discretas, sino medir la relevancia de los resultados obtenidos frente a una consulta textual.
2. Se obtiene una comparación directa entre dos enfoques: el modelo baseline (sin reranking) y el avanzado (con reranking u optimización). Esta estructura posibilita cuantificar objetivamente las mejoras logradas, no sólo a nivel global mediante el mAP, sino también a nivel de cada clase o consulta específica. La inclusión de la diferencia Δ (Avanzado - Baseline) en la tabla facilita identificar las categorías donde el nuevo enfoque aporta mayor valor.
3. La evaluación individual por clase (para consultas simples) y por consulta textual (en consultas complejas) ofrece información sobre el comportamiento del sistema. Este

análisis permite detectar clases difíciles o mal representadas, así como comprender cómo el modelo responde a distintos tipos de consultas (visuales simples versus combinaciones semánticas complejas).
4. Se contempla tanto consultas simples (directamente asociadas a clases del dataset Pascal VOC) como consultas complejas que combinan atributos o condiciones semánticas (“auto no rojo”, “gato junto con perro negro”). Esta dualidad demuestra la flexibilidad del método para evaluar sistemas de búsqueda tanto en entornos controlados como en contextos más realistas.

## Limitaciones

1. Cantidad limitada de imágenes evaluadas: En ambos esquemas se utilizó un número reducido de imágenes por consulta (K=10), lo que restringe el análisis a las primeras posiciones del ranking. Esto puede ocultar comportamientos distintos del modelo en rankings más extensos, donde podrían aparecer más resultados relevantes.
2. Ground truth acotado y parcial: El conjunto de referencia (ground truth) para las consultas complejas fue definido manualmente y con pocas imágenes por categoría. Esto introduce un sesgo subjetivo y limita la representatividad de los resultados, ya que puede no incluir todas las imágenes relevantes del dataset.
3. Homogeneidad en consultas simples: En las consultas simples, el rendimiento es muy alto y las diferencias entre sistemas son mínimas, lo que reduce la sensibilidad de las métricas. Esto ocurre porque las consultas directas (“dog”, “car”, “person”, etc.) están bien representadas en el dataset y ambos modelos responden correctamente.
4. Restricciones semánticas en consultas complejas: Las consultas con negaciones o relaciones entre objetos (como “un gato junto con un perro negro”) dependen fuertemente de la comprensión semántica del modelo. Sin un procesamiento lingüístico o visual avanzado, es difícil que el sistema capture correctamente esas combinaciones, afectando los valores de precisión y recall.
5. Dependencia del top-K elegido: Las métricas (AP@K, Precision@K, Recall@K y F1@K) dependen directamente del valor de K. Cambiar el tamaño del conjunto evaluado podría modificar los resultados y alterar las conclusiones sobre qué sistema es superior.
6. No se considera el rendimiento computacional: El análisis se centró únicamente en la calidad de los resultados (precisión y recall), sin contemplar el tiempo de cómputo, velocidad de búsqueda ni eficiencia en la carga o procesamiento de imágenes. Esto implica que no se evalúa la escalabilidad del sistema ni su desempeño en escenarios reales donde la latencia y el tiempo de respuesta son factores críticos.

## Posibles Mejoras

1. Evaluación en múltiples valores de K: Actualmente, la evaluación se centra en un único valor de corte del ranking. Incorporar distintos valores de K (por ejemplo, 5, 10, 20 y full) permitiría observar cómo varía el rendimiento del sistema a diferentes

profundidades de búsqueda. Esto es importante porque en muchos escenarios el usuario solo analiza los primeros resultados, mientras que en otros podría interesarle la calidad del ranking completo. Un análisis multiescala proporcionaría una visión más completa y realista del comportamiento del modelo.

2. Inclusión de métricas complementarias: Para obtener una evaluación más equilibrada, podrían incorporarse métricas adicionales como Recall@K (que mide la cobertura de resultados relevantes) o NDCG@K (Normalized Discounted Cumulative Gain), que considera la posición de cada resultado relevante y los grados de relevancia posibles. Estas métricas enriquecerían la interpretación, permitiendo distinguir entre modelos que priorizan la precisión temprana y aquellos que ofrecen un ranking más uniforme.
3. Ponderación de clases en el cálculo del mAP: En lugar de promediar todas las clases por igual, se podría implementar una versión ponderada del mAP que tome en cuenta la cantidad de ejemplos o la importancia relativa de cada clase. De esta forma, las métricas reflejarían mejor el rendimiento general del sistema en datasets desbalanceados y evitarían que clases poco representadas distorsionen el promedio global.
4. Ampliación y validación del ground truth para consultas complejas: Las consultas complejas, que combinan atributos o relaciones (“un gato junto a un perro negro”), podrían evaluarse con mayor rigurosidad construyendo un ground truth más sistemático. Una opción sería generar subconjuntos automáticos mediante la intersección de clases y atributos (por ejemplo, “auto” ∩ “no rojo”), o bien realizar una validación manual revisada por más de un evaluador. Esto incrementaría la objetividad y reproducibilidad de los resultados.
5. Evaluación de la diversidad en los resultados recuperados: Una mejora conceptual importante sería incluir métricas que evalúen la diversidad interna de los resultados, como la intra-list diversity. Esta métrica analiza cuán distintos son los elementos dentro del top-K recuperado. De esta manera, se podría evaluar no solo la relevancia de los resultados, sino también su variedad, aspecto especialmente valioso en sistemas donde se busca fomentar la exploración o cubrir múltiples perspectivas visuales.
6. Análisis visual complementario: Más allá de las métricas numéricas, el uso de representaciones visuales (como curvas Precision–Recall, gráficos de Precision@K o comparaciones visuales entre resultados del baseline y del modelo avanzado) ayudaría a interpretar de forma más intuitiva las diferencias entre sistemas. Este enfoque mixto, cuantitativo y cualitativo, fortalecería el análisis y facilitaría la comunicación de los hallazgos.
7. Evaluación de aspectos de eficiencia y escalabilidad: Una extensión futura del esquema podría incorporar métricas relacionadas con la eficiencia temporal, el uso de recursos y la escalabilidad del modelo frente a conjuntos de datos mayores. Esto permitiría evaluar el equilibrio entre precisión y costo computacional, ofreciendo una visión más integral del rendimiento real del sistema en contextos operativos.

# Discusión y Análisis Crítico

El sistema de búsqueda desarrollado mostró un buen desempeño en la interpretación y recuperación de imágenes a partir de consultas textuales. La etapa baseline, basada en el modelo **CLIP** y el índice **FAISS**, funcionó correctamente para consultas directas y simples, demostrando la eficacia de **CLIP** al proyectar texto e imágenes en un mismo espacio semántico. Este enfoque permitió obtener resultados coherentes y consistentes en términos de similitud visual y conceptual.

La extensión hacia una búsqueda avanzada, incorporando traducción automática, reformulación con modelo de lenguaje (**TinyLlama**) y reranking con penalización por negación, aportó una mejora notable en la comprensión semántica de consultas complejas.

El sistema logró interpretar frases con negaciones o estructuras más naturales del lenguaje, ajustando los resultados de manera más alineada con la intención del usuario. Esta integración demostró que el uso combinado de modelos multimodales y lingüísticos puede mejorar sustancialmente la precisión semántica en la recuperación de imágenes.

Sin embargo, algunos intentos no alcanzaron el rendimiento esperado. En particular, las consultas con múltiples negaciones o relaciones complejas entre términos presentaron dificultades, lo que sugiere que la reformulación automática aún es sensible a errores de interpretación o a ambigüedades del lenguaje natural. Asimismo, el proceso de reranking depende fuertemente de la calidad de los embeddings y de la correcta detección del término negativo; pequeñas inconsistencias en esa etapa pueden afectar el orden final de las imágenes.

Entre las principales limitaciones del enfoque, se destaca el hecho de que el sistema depende de modelos pre entrenados (**CLIP** y **TinyLlama**), por lo que su rendimiento está condicionado por los sesgos y el vocabulario de esos modelos. Además, la búsqueda avanzada implica un mayor costo computacional, especialmente en el cálculo de similitudes y la generación de embeddings textuales durante la reformulación. A pesar de ello, el sistema logró mantener un equilibrio entre precisión y eficiencia, demostrando la viabilidad del enfoque propuesto.

# Referencias y herramientas utilizadas

Este trabajo fue desarrollado con fines académicos y de investigación, utilizando modelos, datasets y herramientas disponibles públicamente. En particular, se empleó el dataset Pascal VOC 2012 para la evaluación del sistema de recuperación de imágenes; el modelo CLIP (Contrastive Language–Image Pre-training) de OpenAI para la generación de embeddings multimodales; y la librería FAISS para la indexación y búsqueda eficiente por similitud vectorial.

Para la reformulación semántica de consultas se utilizaron modelos de lenguaje de código abierto, incluyendo TinyLlama y modelos de traducción automática desarrollados por

Helsinki-NLP. Asimismo, se utilizó asistencia de modelos de lenguaje como apoyo en tareas de redacción, revisión y reformulación conceptual.

El diseño del sistema, la implementación, el análisis de resultados y las conclusiones son originales y responsabilidad exclusiva de los autores.

# Trabajo Futuro

Se podría optimizar la etapa de reformulación mediante el uso de modelos de lenguaje más robustos, capaces de manejar con mayor precisión las negaciones, comparaciones y relaciones lógicas presentes en las consultas. Esto permitiría mejorar la interpretación de expresiones complejas y reducir los errores en la generación de los términos principales y negativos.

También se plantea la incorporación de un módulo de retroalimentación del usuario, que permita ajustar los resultados en función de las selecciones o correcciones realizadas durante la búsqueda, favoreciendo un proceso de aprendizaje iterativo que refine la calidad del ranking.

Desde el punto de vista técnico, sería valioso experimentar con modelos visuales más avanzados (como **CLIP ViT-L** o **SigLIP**) y estrategias de indexación más eficientes en **FAISS**, con el fin de mejorar la precisión y reducir los tiempos de respuesta en grandes volúmenes de datos.