

---

# Detección no supervisada de imágenes generadas por IA mediante agrupamiento de atributos visuales

---

**Patricio Gerpe Pablo Silvarredonda Ricardo Amarilla**

Data Mining en Ciencia y Tecnología

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

**06 de Julio de 2025**

## Abstract

La creciente disponibilidad de imágenes generadas mediante inteligencia artificial (IA) plantea nuevos desafíos en la detección de fraudes digitales, auditoría visual y verificación de autenticidad. Este trabajo evalúa la capacidad de separación no supervisada de algoritmos de clustering aplicados a vectores de atributos visuales para distinguir entre imágenes naturales y aquellas manipuladas o generadas artificialmente. Como antecedente, se retoman enfoques previos en detección de anomalías mediante clustering. Se utiliza el modelo VGG16 como extractor de características mediante transfer learning sobre el conjunto Natural Images, que incluye ocho clases visuales. Primero se analizó la separabilidad de las etiquetas reales sobre el conjunto de datos original probando K-Means, DBSCAN y Clustering Jerárquico Aglomerativo y luego utilizamos el Clustering Jerárquico (el que encontramos más apropiado al dataset) aplicado a una versión “inoculada” con imágenes falsas generadas por IA (IFGIA), con modificaciones sutiles y grotescas. Los resultados muestran que el algoritmo con anomalías sigue separando coherentemente las etiquetas reales, pero existen limitaciones de diferenciación cuando las anomalías son más sutiles ya que estas se agrupan en los mismos clusters que las reales mientras que anomalías más grotescas se aglutan en diferentes clusters distribuidos.

## 1. Introducción

El avance de los modelos generativos ha multiplicado la disponibilidad de contenidos visuales sintéticos, desde imágenes hiperrealistas creadas con inteligencia artificial hasta memes, deepfakes y artefactos visuales grotescos como los denominados "brainrots". Esta proliferación plantea desafíos importantes para la seguridad informática, la auditoría de contenidos, el control de calidad en flujos digitales y la confianza en los datos. En paralelo, la detección de anomalías sigue siendo un problema clave en contextos críticos como el control industrial o el monitoreo automatizado, donde distinguir entre contenidos genuinos y sintéticos es cada vez más relevante. Por un lado, la detección de anomalías visuales tiene un papel clave en aplicaciones críticas como el **control industrial**, donde identificar defectos o patrones inusuales de forma automática puede prevenir fallos o pérdidas. Por otro, el uso creciente de **imágenes sintéticas generadas por IA** —como deepfakes, memes manipulados, bran rots, entre otros— plantea desafíos importantes en cuanto a la veracidad, auditoría, seguridad y calidad de los datos visuales. En este contexto, se vuelve necesario desarrollar herramientas capaces de distinguir de forma confiable imágenes reales de imágenes sintéticas o alteradas.

La detección de anomalías visuales ha sido abordada desde distintos enfoques dentro del campo del aprendizaje profundo, especialmente en contextos industriales y en el análisis de contenidos generados artificialmente. Entre los trabajos más destacados se encuentran los métodos que combinan autoencoders con aprendizaje por transferencia (transfer learning), así como aquellos que aplican arquitecturas Transformer para la detección no supervisada. Más recientemente, han comenzado a

emergen estudios centrados específicamente en los riesgos asociados a los contenidos generados por inteligencia artificial, conocidos como FAIGC (Fake Artificial Intelligence Generated Content). A continuación, se describen cuatro contribuciones clave en la literatura que sirven como base para este trabajo. Uno de los enfoques más relevantes en detección semi-supervisada es el propuesto por Saeedi y Giusti (2021), quienes desarrollan un método basado en autoencoders convolucionales entrenados únicamente con imágenes normales, y complementado con un extractor de características previamente entrenado. A partir de esta representación, entrena un clasificador de una sola clase para detectar anomalías. Este enfoque es particularmente útil cuando no se dispone de ejemplos etiquetados de imágenes anómalas, y resulta aplicable tanto a imágenes reales con defectos sutiles como a imágenes sintéticas que alteran ligeramente la estructura de clases visuales conocidas. Su trabajo aporta una estrategia robusta para la construcción de mapas de anomalía sin supervisión directa sobre las clases anómalas, lo cual se alinea con la motivación de este estudio.

Por otro lado, Yang y Guo (2022) presentan un método completamente no supervisado para la detección de anomalías en imágenes industriales, utilizando una arquitectura basada en Vision Transformers (ViT). Su propuesta aprovecha la capacidad de los Transformers para capturar relaciones globales entre bloques de imagen, superando las limitaciones de los autoencoders convolucionales tradicionales, que tienden a centrarse en patrones locales. Además, incorporan un módulo de memoria y mecanismos de atención coordinada para reforzar la diferenciación entre muestras normales y anómalas. Este trabajo resulta especialmente relevante como antecedente técnico para explorar el uso de transfer learning en combinación con métodos de agrupamiento, como se propone en la presente investigación. Además, el relevamiento realizado por Yu et al. (2024) ofrece una visión general integral sobre el fenómeno emergente de los contenidos falsos generados por IA (FAIGC). A través de una clasificación sistemática de métodos de generación, modalidades implicadas (texto, imagen, video, voz) y estrategias de detección, el trabajo establece un marco conceptual para comprender los riesgos asociados a la proliferación de este tipo de contenidos. Asimismo, destaca los desafíos específicos que plantea la detección de imágenes sintéticas altamente realistas en contextos como redes sociales, periodismo y ciberseguridad. Este artículo justifica la necesidad de desarrollar enfoques que, como el aquí propuesto, puedan diferenciar de forma no supervisada entre imágenes genuinas y aquellas alteradas o generadas por modelos generativos avanzados. Finalmente, la evaluación comparativa realizada por Sánchez Vinces et al. (2025) aporta una perspectiva valiosa sobre los métodos de detección de anomalías basados en clustering, un enfoque tradicional que ha sido poco explorado en los últimos años frente al auge de los métodos basados en aprendizaje profundo. Su estudio demuestra que, cuando se aplican correctamente, las técnicas de agrupamiento como KMeans-pueden igualar o incluso superar a métodos más sofisticados en términos de eficiencia, calidad de resultados y escalabilidad. Además, destacan su bajo costo computacional y mayor interpretabilidad, cualidades especialmente relevantes en contextos de detección no supervisada. Esta línea de trabajo apoya la viabilidad del enfoque que se explora en esta investigación: utilizar algoritmos de clustering aplicados a características visuales extraídas por modelos previamente entrenados para distinguir entre imágenes reales y generadas. En conjunto, estos antecedentes demuestran la evolución de la investigación en detección de anomalías visuales, desde entornos industriales controlados hasta escenarios abiertos con presencia de contenido sintético. También evidencian la pertinencia de aplicar técnicas de clustering no supervisado sobre atributos visuales extraídos mediante transfer learning, como una vía prometedora para enfrentar los desafíos actuales en verificación de autenticidad de imágenes.

**Pregunta de investigación.** ¿Es posible identificar, sin supervisión ni etiquetas, imágenes sintéticas generadas mediante inteligencia artificial como *outliers visuales*, utilizando *clustering jerárquico* sobre vectores de características extraídas con *VGG16*, y lograr su separación estructural respecto a las imágenes naturales, especialmente en presencia de alteraciones visuales evidentes?

**Hipótesis.** Las imágenes sintéticas generadas mediante inteligencia artificial, especialmente aquellas con alteraciones visuales grotescas o incongruentes con las clases naturales, se comportan como *outliers* dentro del espacio de características extraído con *VGG16*. En consecuencia, pueden ser detectadas como *anomalías* y agrupadas en clústeres diferenciables mediante *clustering jerárquico no supervisado*, sin necesidad de etiquetas externas. Se espera que la capacidad de separación sea más efectiva ante manipulaciones evidentes, y limitada ante modificaciones sutiles.

## 2. Métodos Generales

### 2.1. Dataset

Se trabajó con un dataset de 6899 imágenes que fue sometido a un sampleo estratificado. El dataset consta de imágenes previamente clasificadas en 5 grupos: airplane (727), car (968), cat (885), dog (702), flower (843), fruit (1000), motorbike (788) y personas (986).

Cuadro 1: Resumen del dataset de imágenes

| Métrica                   | Descripción  |
|---------------------------|--|
| Tamaños únicos de imagen  | 3602 resoluciones distintas                          |
| Tamaño más común          | 100x100 con 1968 imágenes                            |
| Canales                   | Todas las imágenes tienen 3 canales (RGB)            |
| Tipo de dato              | Solo se usa uint8 (valores entre 0 y 255)            |
| Rango global de píxeles   | Mínimo = 0, Máximo = 255                             |
| Tamaños con solo 1 imagen | La gran mayoría de tamaños son únicos (Cantidad = 1) |

En el cuadro 1 se puede observar un resumen de las principales propiedades del dataset bajo estudio.

### Ejemplos de imágenes por tipo



Figura 1: Ejemplo de imágenes naturales que componen al dataset

En la figura de 1 se puede observar un grupo de imágenes de ejemplos de cada tipo del dataset.

### Promedio de las imágenes

Se generaron promedios tanto generales como específicos por cada clase de imagen natural a partir de las muestras disponibles. Estos promedios se calcularon tomando en cuenta los valores de los píxeles en las imágenes, produciendo una representación promedio de los colores y características visuales de cada grupo.

### Promedio General:

La imagen promedio, como se muestra en la Figura , fue construida a partir del total de las 6.899 imágenes del dataset. Esta imagen proporciona una vista general de las características comunes presentes en el conjunto completo. El promedio global ayuda a identificar patrones visuales dominantes y tendencias generales en cuanto a color y forma. Ver figura 3

### Promedios por Clase:

Además, se calcularon imágenes promedio para cada una de las clases del dataset (airplane, car, cat, dog, flower, fruit, motorbike y person). Estas imágenes promedio, mostradas en la Figura 2, permiten observar características visuales particulares de cada categoría, como diferencias en color, estructura y distribución espacial.



Figura 2: Imágenes promedios por tipo

Figura 3: Imágenes Global promedios

La observación de estas imágenes revela que, si bien algunas clases comparten características comunes (por ejemplo, formas difusas en clases como flower o fruit), otras muestran patrones visuales más definidos (como en airplane o car), lo cual puede resultar útil para tareas de clasificación automática basada en aspectos visuales.

#### Análisis de color por clase con imágenes homogenizadas:

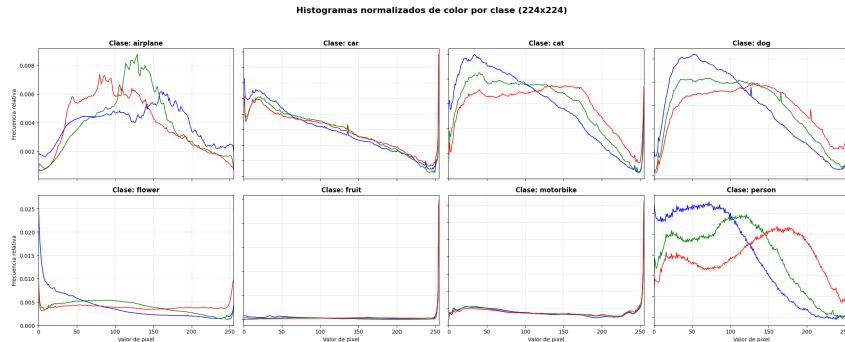


Figura 4: Histogramas normalizados de color por clase (224x224)

Se redimensionaron todas las imágenes a 224×224×3 píxeles para asegurar consistencia espacial y compatibilidad con arquitecturas como VGG16. Luego, se calcularon histogramas RGB por clase utilizando cv2.calcHist, y se normalizaron para obtener frecuencias relativas. Esto permitió comparar de forma justa las distribuciones de color entre clases, eliminando el sesgo por tamaño o cantidad de imágenes.

Para la extracción de features de estas imágenes se utilizó un modelo pre-entreando de redes neuronales conocido como VGG16, un modelo de Transfer Learning que posee 16 capas (con pesos) pre entrenado para clasificación de imágenes. Dicho modelo está implementado en Keras y por default trabaja con imágenes de 224x224. Al estar trabajando un dataset de imágenes de distintos tamaños, las mismas debieron ser redimensionadas antes de aplicar el modelo. Como output, el modelo devuelve 4092 features para cada una de las imágenes que luego serán las utilizadas. Se utilizó el conjunto **Natural Images**, compuesto por 6.899 imágenes distribuidas en 8 clases visuales. Como paso previo, se aplicó un proceso de **preprocesamiento** que incluyó:

- **Normalización** de los valores de pixel.
- Redimensionamiento uniforme de las imágenes a **224x224** píxeles.
- Conservación de **etiquetas reales** para análisis posteriores (aunque no se usan durante el clustering ni entrenamiento del autoencoder).

Estas transformaciones fueron necesarias para adaptar las imágenes al formato de entrada requerido por el modelo **VGG16**, utilizado posteriormente para la extracción de atributos visuales.

#### 2.2. Extracción de atributos visuales (VGG16)

Para representar cada imagen como un vector de características, se empleó un modelo **VGG16 preentrenado en ImageNet**, removiendo la capa final de clasificación. Las activaciones intermedias

de una capa convolucional fueron extraídas como **vectores de atributos visuales** de alta dimensión, lo que permite capturar representaciones profundas sin necesidad de entrenamiento adicional. Esta etapa permitió convertir cada imagen en una representación numérica útil para tareas no supervisadas.

### 2.3. Reducción de la dimensionalidad

Se optó por conservar el 70 % de la varianza acumulada mediante PCA como umbral de corte, ya que permitió reducir los vectores de 4096 dimensiones a solo 155 componentes principales. Este valor fue seleccionado por representar un equilibrio razonable entre compresión y representatividad: aunque umbrales mayores como el 95 % requerían hasta 1416 componentes, con incrementos marginales de información, el 70 % retiene buena parte de la estructura de los datos con una reducción significativa en la dimensionalidad.

### 2.4. Clustering sobre imágenes reales

De manera tal de poder determinar el algoritmo de agrupamiento más apropiado para realizar los posteriores experimentos sobre el conjunto de datos, se aplicaron técnicas de **clustering no supervisado** sobre los componentes de los vectores de atributos extraídos. El objetivo fue evaluar en qué medida las separaciones generadas por cada algoritmo preservan la estructura natural del dataset. Se utilizaron los algoritmos **K-Means**, **DBSCAN** y **Clustering Jerárquico** como métodos principales de agrupamiento, siendo el Clustering Jerárquico aquel algoritmo que pudo separar con mayor coherencia las etiquetas reales del conjunto de datos. Esta etapa permite establecer una línea base estructural para contrastar posteriormente con las anomalías generadas.

### 2.5. Generación de anomalías sintéticas

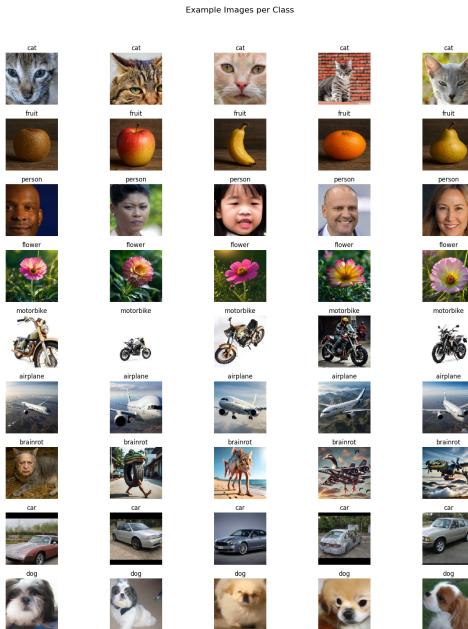


Figura 5: Ejemplos de anomalías sintéticas generadas por clase.

Se diseño un conjunto de datos de anomalías visuales. Estas están formadas por:

- **Imágenes del dataset Artifact:** Se utilizaron imágenes del dataset público *Artifact*, que contiene gran cantidad de imágenes reales y sintéticas. Se seleccionaron principalmente imágenes de perros, gatos, aviones, motos y personas generadas por modelos de difusión como *Stable Diffusion*, *Latent Diffusion*, y GANs como *StyleGAN2* y *StyleGAN3*.<sup>1</sup>
- **Imágenes sintéticas sutiles:** Para completar categorías subrepresentadas (como flores y frutas) se generaron imágenes adicionales mediante **IA generativa** (OpenAI, modelos *ChatGPT O4* y *Qwen*), con el prompt: "Generá una imagen hiperrealista de <clase>".

<sup>1</sup><https://www.kaggle.com/datasets/awsa49/artifact-dataset>

- **Imágenes meme públicas:** Se creó la clase *brainrot* con imágenes bizarras tomadas de internet, que combinan elementos de distintas clases (por ejemplo, un cocodrilo con un avión).<sup>2</sup>
- **Imágenes meme generadas:** Para esta misma clase se generaron imágenes adicionales mediante **IA generativa** (OpenAI, modelos *ChatGPT O4* y *Qwen*), combinando dos imágenes del dataset original con el siguiente prompt :

*Use these two input images to create a highly realistic and convincing surreal composite image. It blends a {class\_a} with a {class\_b} in a bizarre and unnatural way. The integration should be seamless enough to appear photographically real, despite the impossible nature of the combination. Both elements ({class\_a} and {class\_b}) should be clearly identifiable within the unified entity. Use expert photorealistic lighting, shadows, and textures to enhance the sense of reality in this impossible scene. Neutral or abstract background.*

La siguiente tabla ilustra la distribución de las clases (la cuál busca cierto balance con respecto al dataset original):

| Clase | Etiqueta    | Cantidad de Registros | Proporción |
|-------|-------------|-----------------------|------------|
| 0     | gato        | 16                    | 14 %       |
| 1     | fruta       | 6                     | 5 %        |
| 2     | persona     | 18                    | 16 %       |
| 3     | flor        | 10                    | 9 %        |
| 4     | motocicleta | 13                    | 12 %       |
| 5     | avión       | 9                     | 8 %        |
| 6     | brainrot    | 10                    | 9 %        |
| 7     | auto        | 18                    | 16 %       |
| 8     | perro       | 13                    | 12 %       |

Cuadro 2: Distribución por clase de imágenes falsas generadas.

Consideramos a una anomalía como grotesca o clase *brainrot* cuando no es simplemente una imagen falsa realista de una clase existente (gato, perro, etc) si no, una imagen surrealista que combina elementos de una clase con otra (como mezclar un cocodrilo con un avión).

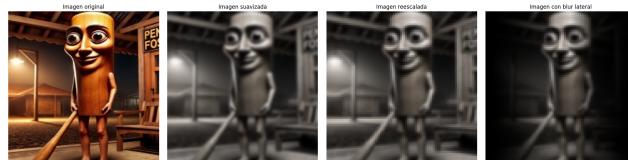


Figura 6: Ejemplo de una anomalía sintética de clase *brainrot*

A modo de poder ejemplificar la caracterización de este tipo de anomalía, la siguiente figura ilustra la aplicación de Connected-Component Labelling y Clustering Espectral. Para facilitar la detección de la entidad, suavizamos la imagen, escalamos a 90x90 y aplicamos un blur lateral tal como se observa en la figura anterior.

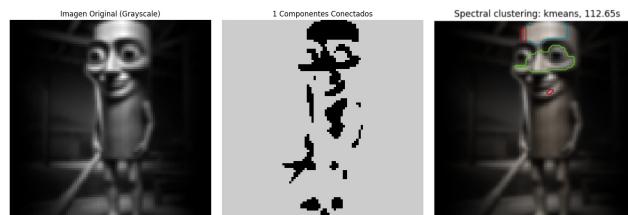


Figura 7: Aplicación de Connected-Component Labelling y Clustering Espectral sobre el ejemplo

<sup>2</sup><https://italian-brainrot.org/>

Todas las imágenes del conjunto de datos de anomalías se encuentran normalizadas a 224x244 y en formato JPG. Se duplicó el dataset original y estas imágenes fueron introducidas en el dataset como **anomalías visuales** en cada categoría correspondiente, reemplazando imágenes originales por cada anomalía insertada en cada clase previamente existente. Para la clase "brainrot", no presente en el dataset original, se agregaron todas las imágenes, sin reemplazarla por ninguna del dataset original, quedando un total de 6909 imágenes. De manera de poder tener una comparación homogénea, este conjunto de datos pasó por extracción de atributos vía VGG16 y luego reducción de la dimensionalidad por PCA. Los componentes de los vectores de atributos fueron el insumo para los algoritmos de agrupamiento. Conservamos las etiquetas de clase (car, cat, dog, etc.) y tipo (fake o real) para posterior validación. Estas etiquetas no fueron utilizadas en el proceso de agrupamiento dada la naturaleza no supervisada del proceso.

## 2.6. Experimentos: Agrupamiento ante la inoculación de imágenes falsas

Tomando de referencia las experiencias de Clustering sobre dataset original (K-Means, DBSCAN y Clustering Jerárquico Aglomerativo), procedemos a evaluar en una nueva experiencia la capacidad del Clustering Jerárquico de separar, sin conocimiento a priorístico de las etiquetas reales, las imágenes normales de las anómalas.

- **Experiencias A:** K-Means, DBSCAN y Clustering jerárquico sobre el dataset sin inoculación de anomalías
- **Experiencia B:** Clustering jerárquico sobre el dataset con inoculación de anomalías

Para el dataset de anomalías solo realizamos Clustering Jerárquico ya que era el que encontramos más performante en cuanto a una separación coherente de las etiquetas reales del dataset.

## 2.7. Método de evaluación

A partir de las etiquetas reales, se utilizó una **matriz de confusión** para validar el rendimiento del modelo en ambas situaciones. Con ella, se buscó identificar si la separación de los clusters coincidía o no con las etiquetas reales del conjunto de datos, observando si tal separación difería cuando las imágenes eran falsas o reales.

# 3. Resultados y discusión

## 3.1. Análisis Exploratorio preliminar

### 3.1.1. Extracción de atributos con modelo VGG16

Para esta investigación se trabajó con un conjunto de 6.899 imágenes naturales, distribuidas en 8 categorías visualmente distintas: aviones, autos, gatos, flores, perros, frutas, motos y personas. Esta diversidad proporciona un escenario ideal para explorar representaciones visuales profundas y aplicar técnicas no supervisadas como reducción de dimensionalidad y agrupamiento.

Se empleó el modelo VGG16 preentrenado en ImageNet para la extracción de características visuales. En particular, se utilizó la capa densa fc1, que genera vectores de 4096 dimensiones por imagen. Estas representaciones condensan información abstracta de alto nivel relacionada con formas, texturas, patrones y estructuras semánticas, aprendidas a partir de millones de imágenes.

Cuadro 3: Resumen del dataset de imágenes y estadísticas de los features

| Descripción                  | Valor   |
|------------------------------|---------|
| Número de imágenes           | 6899    |
| Dimensiones por imagen       | 4096    |
| Media global de los features | 0.4730  |
| Desviación estándar global   | 1.1470  |
| Valor mínimo                 | 0.0000  |
| Valor máximo                 | 28.4520 |

El análisis estadístico que puede observarse en la figura 3 de los vectores arrojó una media global de 0.473, una desviación estándar de 1.147, y valores en el rango de 0 a 28.45. Estas diferencias de escala entre dimensiones reflejan una activación desigual entre componentes.

Además, considerando la alta heterogeneidad visual entre clases —como personas frente a vehículos o animales—, **se concluyó que es fundamental aplicar una estandarización basada en Z-score**. Esto permite que todas las características contribuyan de manera equilibrada a los algoritmos posteriores.

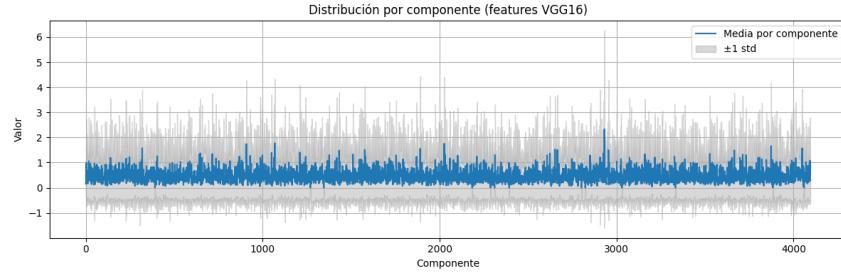


Figura 8: Distribución por componente (features VGG16)

res, evitando que ciertas dimensiones dominen el análisis debido a diferencias de escala o activación. Ver Figura 8.

**Reducción de Dimensionalidad para Clustering** Posteriormente, se aplicó el Análisis de Componentes Principales (PCA) como técnica de reducción de dimensionalidad, con el objetivo de preservar la mayor parte de la información visual original, reduciendo al mismo tiempo la carga computacional. Ver figura 9

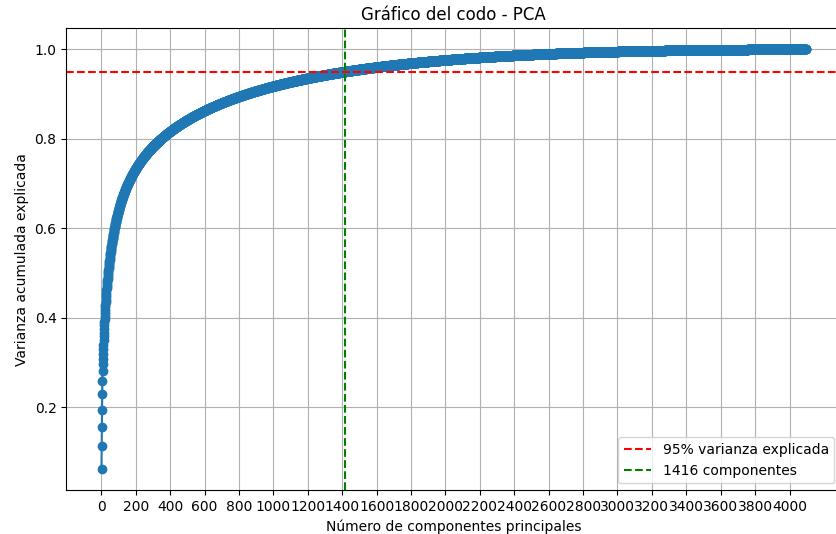


Figura 9: Varianza acumulada explicada

Se evaluó la varianza explicada acumulada por los componentes principales, obteniéndose los siguientes resultados:

- Para explicar el **95 %** de la varianza, fueron necesarios **1.416 componentes**.
- Para capturar el **80 %** de la varianza, se necesitaron **348 componentes**.
- Para alcanzar el **75 %** de la varianza, bastaron **229 componentes**.
- Para retener el **70 %** de la varianza, fueron suficientes **155 componentes**.

Por equilibrio entre eficiencia y preservación de información relevante, se redujo la dimensionalidad a 155 componentes, lo que representa aproximadamente el 70 % de la varianza total. Esta elección se considera razonable, ya que en las visualizaciones del espacio reducido (tanto en 2D como en 3D), los datos mantienen una estructura coherente, permitiendo identificar patrones y posibles agrupamientos. Esto sugiere que la reducción conserva información suficiente para el análisis posterior (ver Figuras 10 y 11).

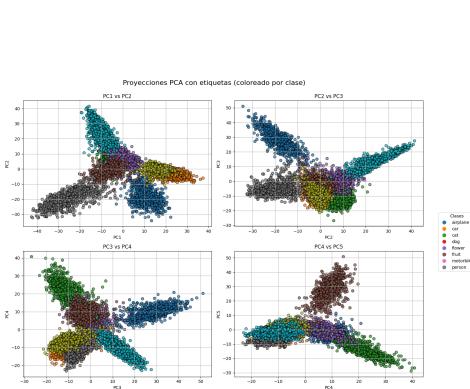


Figura 10: Proyección PCA 2D con etiquetas (coloreado por clase)

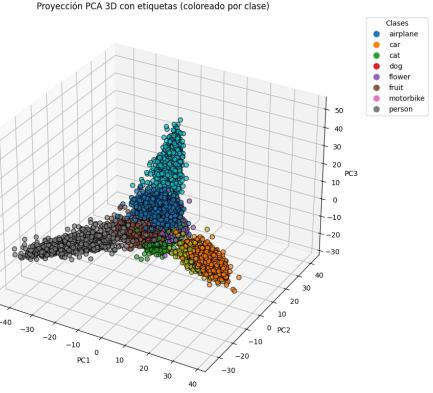


Figura 11: Proyección PCA 3D con etiquetas (coloreado por clase)

### 3.1.2. Clustering

Como parte del análisis exploratorio, se aplicaron técnicas de clustering no supervisado con el objetivo de analizar la estructura interna del conjunto de imágenes reales, sin utilizar etiquetas. Esto permite identificar patrones visuales latentes y establecer una línea base estructural frente a la cual evaluar, posteriormente, la incorporación de imágenes generadas por inteligencia artificial, en línea con la hipótesis del trabajo. Se utilizaron tres algoritmos con enfoques complementarios: KMeans, DBSCAN y Clustering Jerárquico. KMeans brinda una partición eficiente basada en distancia euclidiana, útil como referencia inicial; el método jerárquico permite explorar similitudes a distintos niveles sin fijar el número de grupos; y DBSCAN detecta agrupamientos según densidad, ideal para formas irregulares y detección de outliers. Esta combinación permite capturar estructuras diversas en los datos naturales. Para evaluar la calidad de los agrupamientos se utilizaron métricas internas como Silhouette y SSE, junto con métricas externas como los índices de Rand ajustado y de van Dongen, complementadas con visualizaciones en espacios reducidos mediante PCA, T-SNE, ISOMAP y UMAP. Cabe señalar que los agrupamientos se realizaron sobre una versión reducida de los datos por PCA (70 % de varianza retenida), lo cual, al tratarse de una técnica lineal, podría implicar cierta pérdida de información que afecte la precisión de los resultados de clustering.

**Algoritmo de Clustering K-means** El algoritmo KMeans representa una estrategia inicial eficaz para realizar agrupamiento, ya que segmenta los datos en grupos mediante la minimización de distancias a centroides definidos. A pesar de requerir la especificación previa del número de clusters, su simplicidad computacional y su interpretación clara lo convierten en una herramienta valiosa para obtener una primera aproximación a la estructura del conjunto de datos.

#### Aplicación del algoritmo y determinación de parámetros.

Se aplicó el algoritmo K-Means sobre el conjunto de atributos extraídos mediante VGG16 y reducidos dimensionalmente mediante PCA. Para determinar el número óptimo de clusters, se utilizaron dos métricas clave: el Silhouette Score y la Suma de Errores Cuadrados (SSE). Ver figuras 12 y 13

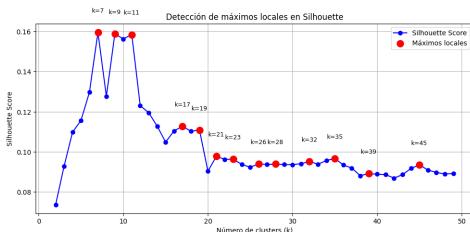


Figura 12: Detección de máximos locales en Silhouette

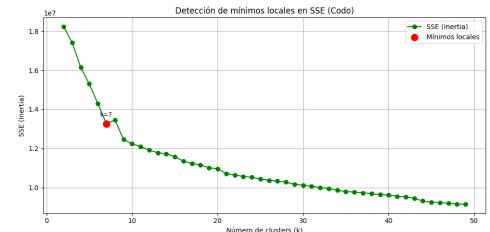


Figura 13: Detección de mínimos locales en SSE (Codo)

Los resultados obtenidos muestran que para  $k = 7$  se observa un mínimo local pronunciado en la curva de SSE, lo que sugiere un posible punto de “codo”, indicando un buen equilibrio entre compacidad y complejidad del modelo. A su vez, este mismo valor de  $k$  coincide con el máximo local más alto en el coeficiente de silueta (0,1595), lo cual refuerza su validez como una opción sólida para el agrupamiento. Si bien se identificaron otros máximos locales en la métrica de silueta para valores más altos de  $k$  (como  $k = 9$ ,  $k = 11$  y  $k = 17$ ), la mejora en el score es marginal o decreciente, y podría representar sobresegmentación sin una ganancia clara en la calidad del clustering. Por tanto,  $k = 7$  se perfila como una elección razonable para representar la estructura interna de los datos con un buen compromiso entre simplicidad y coherencia de los grupos.

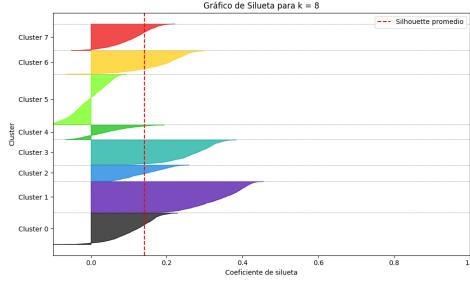


Figura 14: Gráfico de Silhouette para  $k = 8$

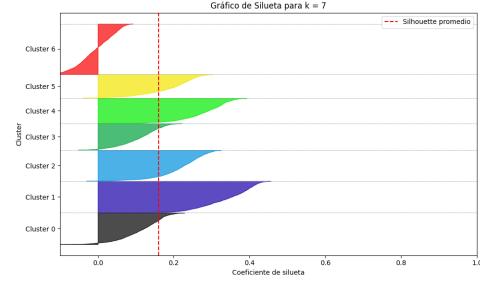


Figura 15: Gráfico de Silhouette para  $k = 7$

Dado que el conjunto de imágenes cuenta con 8 clases reales, se consideró relevante evaluar el comportamiento del agrupamiento para  $k = 8$  utilizando un gráfico de silueta. Esta visualización permite analizar la validez interna de los clusters generados por KMeans, sin necesidad de recurrir a las etiquetas durante el proceso de agrupamiento. Al comparar los resultados de  $k = 7$  (valor sugerido por el análisis de SSE y Silhouette) con  $k = 8$  (coincidente con la cantidad de clases reales), se busca identificar si el número natural de categorías del conjunto se refleja en la estructura no supervisada de los datos. Si bien un valor de  $k$  coincidente con las clases reales podría parecer ideal, su validez se sustenta únicamente si el agrupamiento resultante presenta separación clara entre grupos y valores altos de silueta, lo que se evalúa visualmente en este gráfico. Ver figura 14 y 15

#### Bootstrapping: SSE y Silhouette Score

Con el objetivo de obtener estimaciones más robustas y confiables para la selección del número óptimo de clusters, se implementó un procedimiento de validación mediante bootstrap con 500 repeticiones, evaluando valores de  $k$  en el rango de 2 a 15. En cada iteración se generó una muestra con reemplazo del conjunto de datos reducidos por PCA, se aplicó K-Means y se calcularon tanto el Silhouette Score como el SSE (Sum of Squared Errors). Esto permitió estimar la media y desviación estándar de ambas métricas para cada valor de  $k$ , brindando así una visión más estable y estadísticamente respaldada del comportamiento del modelo. Los resultados obtenidos pueden observarse en la Tabla 4 y la Figura 16.

Cuadro 4: Resumen de métricas Silhouette y SSE para distintos valores de  $k$

| $k$ | Silhouette (mean) | Silhouette (std) | SSE (mean)    | SSE (std)   |
|-----|-------------------|------------------|---------------|-------------|
| 2   | 0.0787            | 0.0276           | 18513296.1000 | 229020.7174 |
| 3   | 0.0804            | 0.0214           | 17376571.2500 | 251301.7037 |
| 4   | 0.0934            | 0.0192           | 16322651.7060 | 321846.8243 |
| 5   | 0.1121            | 0.0172           | 15329635.3840 | 356152.1431 |
| 6   | 0.1286            | 0.0188           | 14502203.9900 | 427357.8625 |
| 7   | 0.1437            | 0.0215           | 13763950.7220 | 470525.2268 |
| 8   | 0.1546            | 0.0224           | 13072519.6540 | 427061.3057 |
| 9   | 0.1573            | 0.0215           | 12663256.0700 | 357470.3219 |
| 10  | 0.1562            | 0.0185           | 12346426.7860 | 278597.7151 |
| 11  | 0.1501            | 0.0195           | 12130191.1620 | 220444.5384 |
| 12  | 0.1438            | 0.0204           | 11947626.3120 | 175388.3986 |
| 13  | 0.1388            | 0.0206           | 11780520.3220 | 145673.0204 |
| 14  | 0.1321            | 0.0197           | 11634909.8380 | 130036.9878 |
| 15  | 0.1271            | 0.0183           | 11502970.7540 | 143393.9048 |
| 16  | 0.1236            | 0.0182           | 11367824.5560 | 119204.0198 |

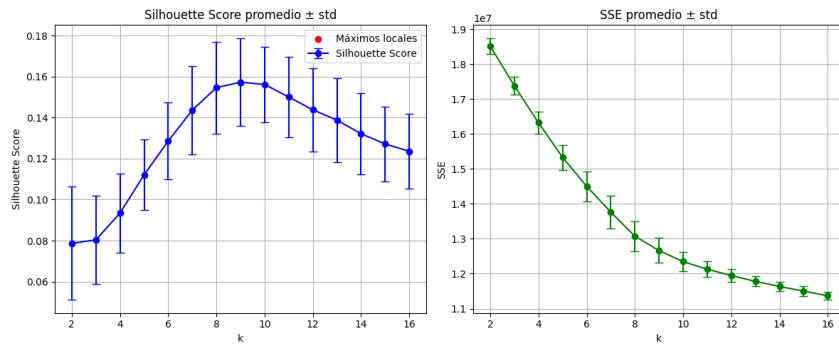


Figura 16: Bootstrapping: SSE y Silhouette Score

Los resultados muestran una evolución esperable en ambas métricas: el SSE disminuye progresivamente a medida que aumenta el número de clusters, indicando una mayor compacidad interna, aunque con ganancias decrecientes. Por otro lado, el Silhouette Score mejora hasta alcanzar un máximo local en  $k = 9$  ( $0.1573 \pm 0.0215$ ), lo que sugiere que esta configuración ofrece el mejor equilibrio entre cohesión intra-cluster y separación entre grupos dentro del rango evaluado. No obstante, los valores para  $k = 7$  y  $k = 8$  también son elevados (0.1437 y 0.1546 respectivamente), y presentan un comportamiento estable. Esto refuerza que el agrupamiento en torno a 7–9 clusters es razonable, y permite seleccionar  $k = 7$  como solución final, priorizando interpretabilidad y simplicidad, sin sacrificar calidad estructural.

Cabe destacar que, si bien el conjunto cuenta con 8 clases reales, el modelo no supervisado sugiere que 7 clusters son suficientes para capturar la organización más representativa de los datos. Esta diferencia puede deberse a la existencia de clases visualmente similares, solapamientos entre categorías o la presencia de outliers, todos factores que pueden influir en el proceso de agrupamiento sin etiquetas.

## Evaluación del agrupamiento respecto a las etiquetas reales.

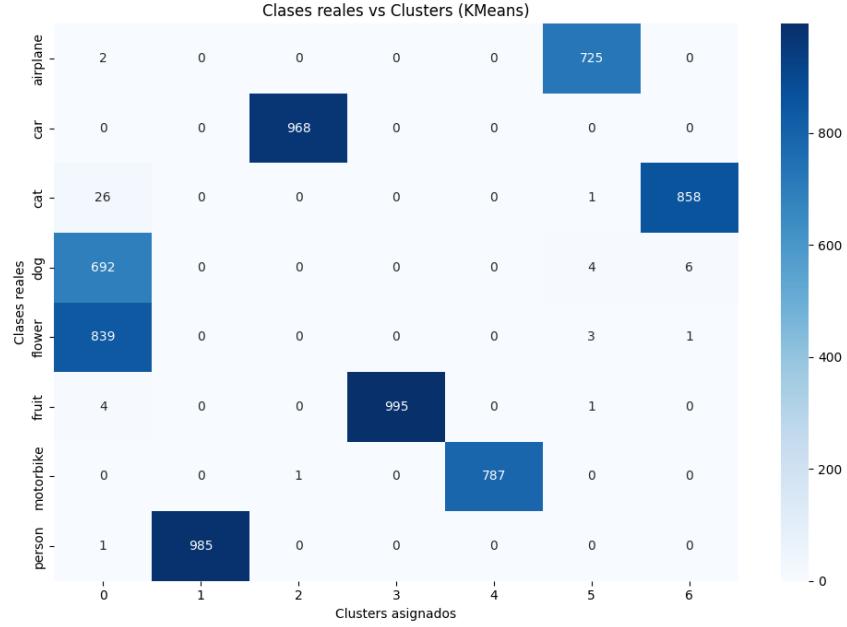


Figura 17: Matriz de confusión

La matriz de confusión (ver figura 17) obtenida al comparar los clusters generados por el algoritmo K-Means con  $k = 7$  frente a las etiquetas reales revela una correspondencia elevada entre agrupamientos no supervisados y categorías originales. Clases como *car*, *person*, *fruit* y *motorbike* fueron agrupadas de forma casi perfecta, con una precisión superior al 99.5 % en un único cluster. También se observó una asignación dominante para *airplane* y *cat*, aunque con leves dispersiones. Sin embargo, el cluster 0 agrupó conjuntamente imágenes de las clases *dog* y *flower*, evidenciando una confusión estructural que podría atribuirse a similitudes visuales o a la pérdida de información durante la reducción de dimensionalidad por PCA. A pesar de esta superposición, la segmentación general fue consistente y precisa. Esta conclusión se refuerza con las métricas de evaluación: el índice de Rand Ajustado (ARI) alcanzó un valor de 0.8800, lo que indica una fuerte coincidencia entre los clusters y las etiquetas reales; mientras que el índice de van Dongen fue de 0.0574, reflejando una baja disimilitud estructural. En conjunto, estos resultados validan que K-Means logró una agrupación eficaz en relación con la estructura real del conjunto de datos.

## Visualización de los resultados en espacios reducidos.

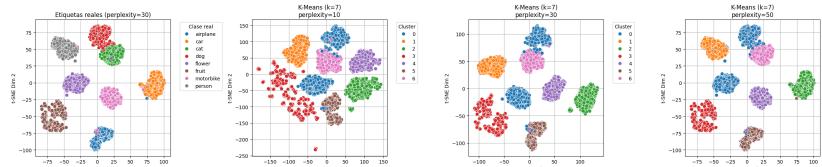


Figura 18: Visualización con T-SNE 2D con cambios en perplexity

Se realizó una visualización en dos dimensiones mediante t-SNE con el objetivo de comparar la estructura descubierta por K-Means ( $k = 7$ ) con las etiquetas reales del conjunto de datos. La figura incluye cuatro paneles: el primero muestra las proyecciones t-SNE coloreadas según las clases reales, y los tres restantes representan los agrupamientos generados por K-Means bajo distintos valores de perplexity (10, 30 y 50). Esta variación permite analizar cómo se proyectan los datos según diferentes escalas de vecindad. Entre las configuraciones evaluadas, se observó que perplexity=50 ofreció la

representación más clara y estructurada de los clusters, revelando mejor separación entre grupos y mayor coherencia visual con las clases originales. Esta visualización cualitativa aporta evidencia adicional sobre la correspondencia entre los grupos descubiertos y la estructura real del conjunto. Ver figura 18

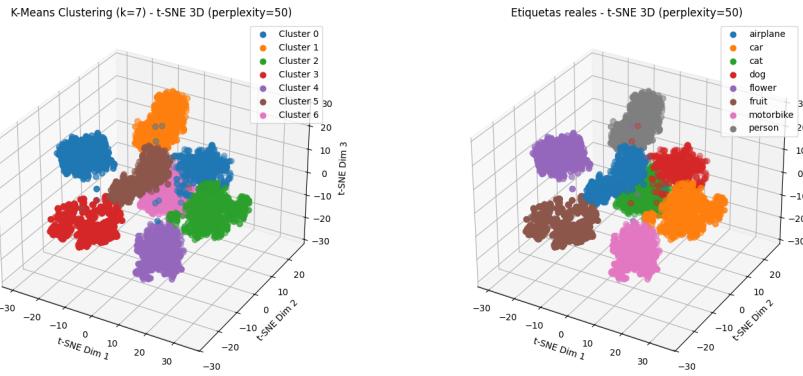


Figura 19: Visualización con T-SNE 3D

Se construyó una visualización tridimensional del espacio reducido mediante t-SNE con 3 componentes y perplexity=50, a partir de los datos transformados por PCA. El objetivo fue comparar la organización de los clusters generados por K-Means ( $k = 7$ ) con la distribución real de las clases en el conjunto de imágenes. La figura resultante contiene dos gráficos: el primero muestra los grupos asignados por el algoritmo K-Means, mientras que el segundo presenta las etiquetas reales. Esta representación permite evaluar visualmente el grado de coincidencia entre ambas estructuras, identificando posibles solapamientos, separaciones claras o inconsistencias. La elección de perplexity=50 resultó adecuada para capturar tanto relaciones locales como globales en el conjunto, y facilitó una observación más fluida de la distribución espacial de los datos en tres dimensiones.

#### Discusión e interpretación de los resultados.

El análisis con **K-Means** permitió explorar la estructura interna del conjunto de imágenes reales de forma no supervisada, tras una reducción de dimensionalidad por **PCA** que conservó el 70 % de la varianza. Para determinar el número óptimo de clusters, se evaluaron los valores de *Silhouette Score* y *SSE*, validados por **bootstrap** con 500 repeticiones. El valor  $k = 7$  emergió como el mejor compromiso entre cohesión intra-cluster y separación entre grupos. Visualizaciones con **t-SNE** en 2D y 3D mostraron cierta correspondencia entre los clusters generados y las clases reales, aunque con solapamientos evidentes. Esto fue confirmado por la **matriz de confusión**, donde varios clusters contienen imágenes de distintas clases reales, lo cual sugiere similitudes visuales entre categorías o limitaciones propias del modelo y la reducción lineal aplicada. Estos hallazgos resultan clave en el marco de la hipótesis del trabajo, ya que permiten establecer un **baseline estructural** sobre datos reales. A partir de esta línea base, será posible evaluar en etapas posteriores si las imágenes generadas por inteligencia artificial se desvían significativamente de esta organización, aportando evidencia sobre su carácter anómalo o no.

**Clustering Jerárquico** El clustering jerárquico resulta especialmente útil para datos visuales, ya que permite explorar relaciones entre imágenes a distintos niveles de similitud sin requerir un número fijo de grupos. Esta capacidad de revelar jerarquías implícitas lo hace valioso para establecer una estructura base en imágenes reales, que luego puede compararse con la distribución de imágenes sintéticas, en línea con la hipótesis de detección de anomalías visuales.

#### Aplicación del algoritmo y determinación de parámetros.

Se aplicó clustering jerárquico aglomerativo sobre una matriz de datos previamente normalizada y reducida mediante PCA. Se evaluaron distintos métodos de enlace: ward, complete, average y single, variando la cantidad de clusters en un rango de  $k=2$  a  $k=49$ .

Para cada configuración, se calcularon las métricas de Silhouette Score y SSE (Suma de Errores Cuadráticos). Los resultados fueron graficados para comparar visualmente el comportamiento de los diferentes métodos

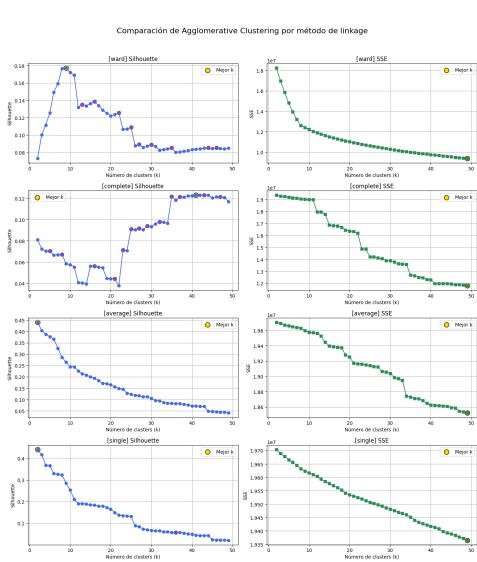


Figura 20: Comparación de Agglomerative Clustering

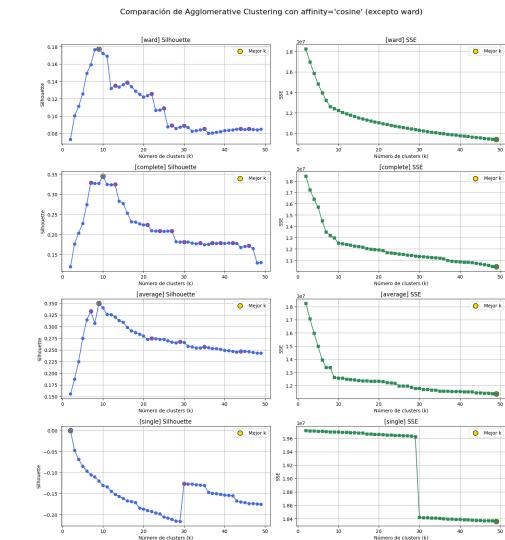


Figura 21: Comparación de Agglomerative Clustering con affinity='cosine' (excepto ward)

Adicionalmente, se realizó un análisis comparativo utilizando distancias euclidianas y de coseno en los métodos de enlace que lo permiten (complete, average, single). Se observó que el uso de la distancia coseno mejoró los valores de Silhouette en comparación con la distancia euclidiana. El método ward se mantuvo evaluado únicamente con distancia euclidiana por su restricción teórica. Ver figuras 20 y 21

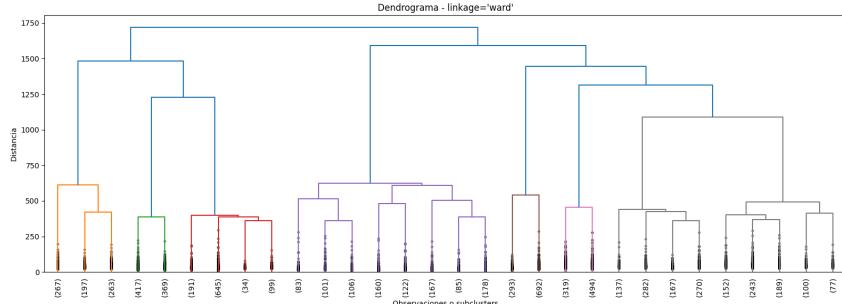


Figura 22: Dendrograma

La figura 22 muestra el Dendrograma asociado al Dataset que puede dar alguna idea de donde puede hacer un corte que mejor ajuste, aunque no siempre es obvia la respuesta.

## Evaluación del agrupamiento respecto a las etiquetas reales.

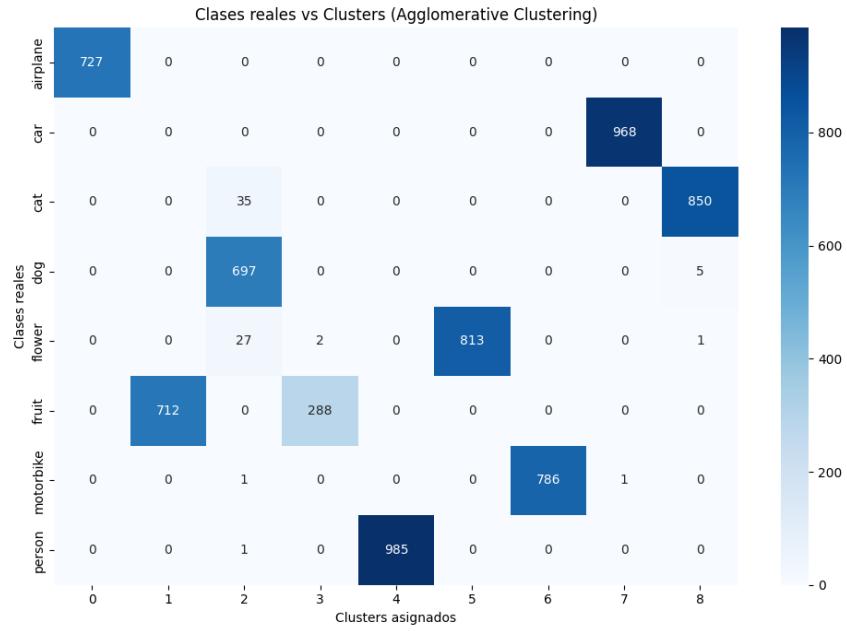


Figura 23: Matriz de confusión

Posteriormente, se aplicó Agglomerative Clustering con enlace tipo *ward* y  $k = 9$ , y se evaluó la segmentación resultante comparando los clusters asignados con las etiquetas reales del conjunto de datos. Para ello, se construyó una tabla de contingencia, un heatmap y se calcularon métricas de evaluación externas. El modelo logró una segmentación precisa en la mayoría de las clases, con un índice de Rand ajustado (ARI) de 0.9379 y un V-Measure Score de 0.9564, lo que indica una fuerte correspondencia entre los agrupamientos no supervisados y las categorías reales. No obstante, se observó que la clase *fruit* se dividió principalmente entre dos clusters: 712 muestras en el cluster 1 y 288 en el cluster 3. Esta división sugiere una variabilidad interna significativa dentro de la categoría, posiblemente asociada a subgrupos estructurales captados por el modelo jerárquico. Ver figura 23

## Visualización de los resultados en espacios reducidos.

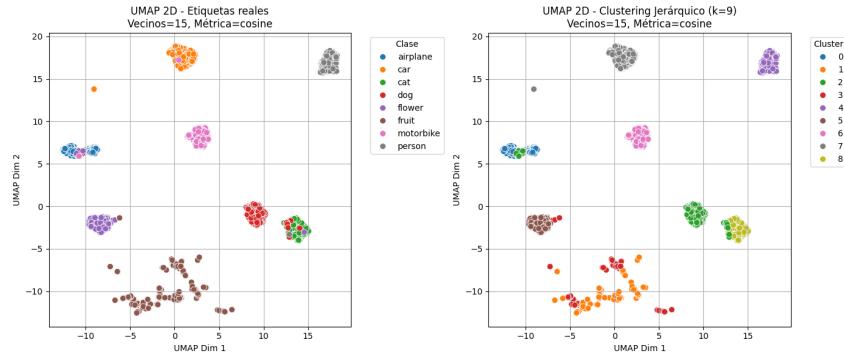


Figura 24: Visualización con UMAP 2D

Finalmente, se utilizó UMAP con metric='cosine' para reducir los datos a 2 y 3 dimensiones, permitiendo visualizar las asignaciones de cluster y las clases reales en un espacio latente.

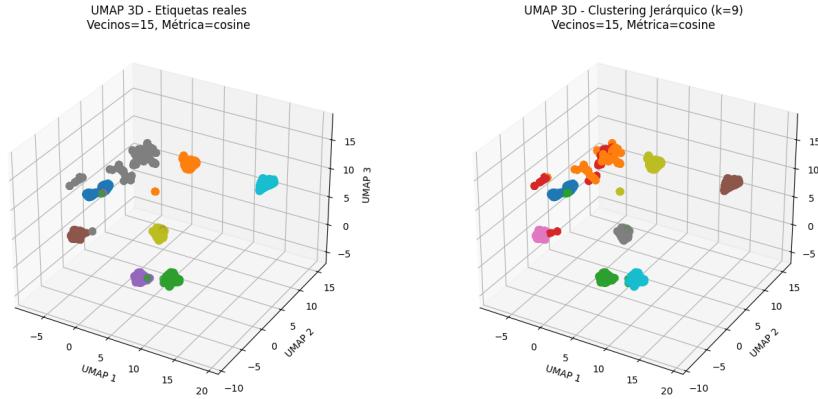


Figura 25: Visualización con UMAP 3D

**Algoritmo de Clustering DBSCAN** El algoritmo DBSCAN es especialmente útil en el análisis de datos visuales por su capacidad para identificar agrupamientos de forma arbitraria basados en densidad, sin necesidad de especificar el número de clusters previamente. Esta propiedad le permite detectar estructuras complejas y separar outliers de manera natural, lo cual es valioso en conjuntos de imágenes reales que pueden contener clases poco definidas o ruido visual. Su aplicación permite establecer una segmentación estructural robusta que sirve como base para contrastar la distribución de imágenes sintéticas, en coherencia con la hipótesis de detección de anomalías visuales.

## Aplicación del algoritmo y determinación de parámetros.

Se aplicó el algoritmo DBSCAN para segmentar las imágenes en base a sus vectores de características reducidos. El análisis comparó el desempeño del algoritmo utilizando dos métricas de distancia: euclíadiana y coseno.

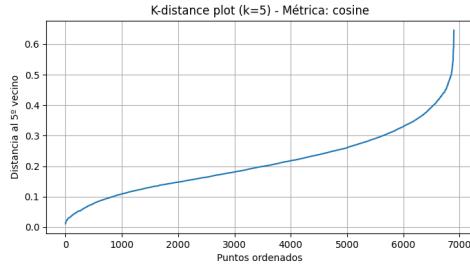


Figura 26: K-distance plot (k=5) - Métrica: cosine

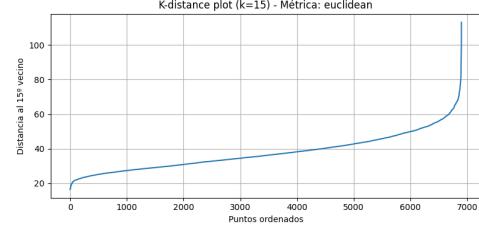


Figura 27: K-distance plot (k=5) - Métrica: euclidean

Con la métrica euclíadiana, DBSCAN no logró identificar agrupamientos significativos. En todas las configuraciones probadas, casi la totalidad de los puntos fueron etiquetados como outliers y los valores del Silhouette Score fueron negativos, lo que indica una pobre calidad de clustering.

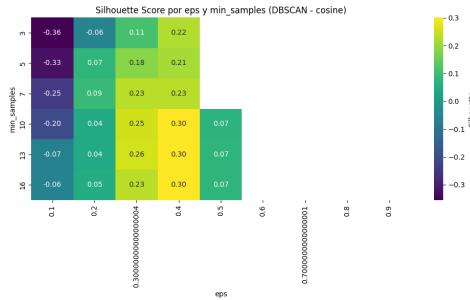


Figura 28: Silhouette Score por eps y min\_samples - Métrica: cosine)

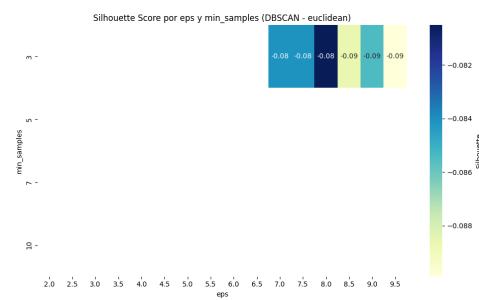


Figura 29: Silhouette Score por eps y min\_samples Métrica: euclidean

En cambio, al usar la distancia coseno, los resultados mejoraron notablemente. Se formaron varios clusters con una baja proporción de outliers y valores de Silhouette Score en torno a 0.30, lo que refleja una estructura de grupos más definida. Esto sugiere que la distancia coseno es más adecuada para capturar la similitud entre imágenes en este espacio de representación. Ver figuras 26, 28, 27 y 29 y el cuadro 5

Cuadro 5: Mejores configuraciones de DBSCAN según Silhouette Score

| # | eps | min_samples | Clusters | Outliers | Silhouette |
|---|-----|-------------|----------|----------|------------|
| 1 | 0.4 | 16          | 6        | 280      | 0.3007     |
| 2 | 0.4 | 13          | 6        | 236      | 0.3002     |
| 3 | 0.4 | 10          | 6        | 205      | 0.2999     |
| 4 | 0.3 | 13          | 10       | 1099     | 0.2554     |
| 5 | 0.3 | 10          | 10       | 984      | 0.2537     |

En resumen, DBSCAN combinado con la métrica coseno permitió detectar agrupamientos útiles, mientras que con la métrica euclíadiana no se logró segmentar de manera efectiva

**Evaluación del agrupamiento respecto a las etiquetas reales.** Se aplicó DBSCAN para agrupar las imágenes en función de sus características, obteniendo resultados muy positivos. El modelo logró formar clusters bien definidos, con un índice de Rand ajustado de 0.7836 y un V-Measure de

0.9088, lo que indica una alta concordancia con las etiquetas reales y una buena separación entre grupos. Además, DBSCAN logró asignar clusters al 96 % de las imágenes, dejando solo un pequeño porcentaje como outliers. Ver figura 30

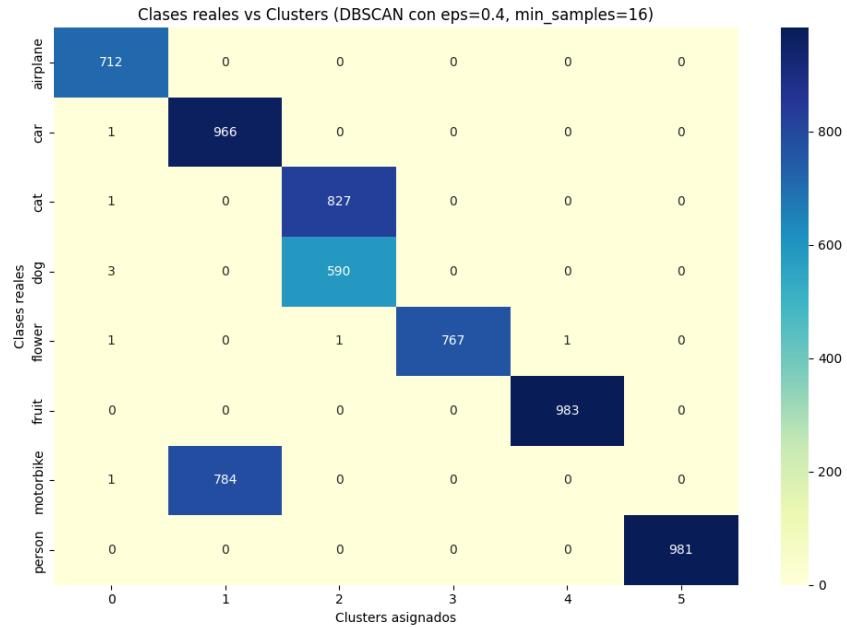


Figura 30: Matriz de confusión

Al observar la tabla de contingencia, se nota que la mayoría de las clases fueron bien agrupadas: por ejemplo, las clases airplane, fruit, flower y person aparecen en clusters casi puros. Sin embargo, destaca el Cluster 2, que agrupa simultáneamente una gran cantidad de imágenes de cat y dog, lo que sugiere que estas dos clases no fueron bien diferenciadas por el modelo. Esta superposición puede deberse a similitudes visuales entre ambas categorías o a limitaciones en las representaciones utilizadas.

En resumen, DBSCAN mostró un rendimiento sólido en la segmentación no supervisada del conjunto de imágenes, con una agrupación precisa en la mayoría de las clases y solo una confusión importante entre gatos y perros.

### Visualización de los resultados en espacios reducidos.

Se utilizó la técnica DBSCAN para agrupar las imágenes en el espacio reducido por PCA, empleando los parámetros  $\text{eps}=0.4$  y  $\text{min\_samples}=16$ , con distancia basada en el coseno. Esta configuración fue elegida por presentar un buen equilibrio entre número de clusters y proporción de outliers, además de obtener un alto Silhouette Score.

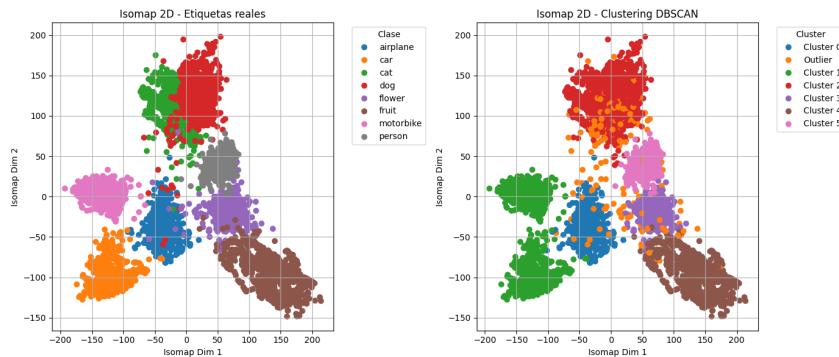


Figura 31: Visualización con ISOMAP 2D

Para visualizar los resultados, se aplicó una reducción adicional mediante Isomap a dos dimensiones. En la figura resultante se comparan las etiquetas reales con los clusters generados por DBSCAN. Se observa que, aunque algunos grupos coinciden parcialmente con las clases originales, también hay una proporción considerable de puntos etiquetados como *outliers*, reflejando la sensibilidad de DBSCAN a la densidad local. Esta representación permite analizar la estructura de los datos desde una perspectiva no supervisada y evaluar la capacidad del modelo para detectar regiones bien definidas en el espacio de características. Ver figuras 31 y 32

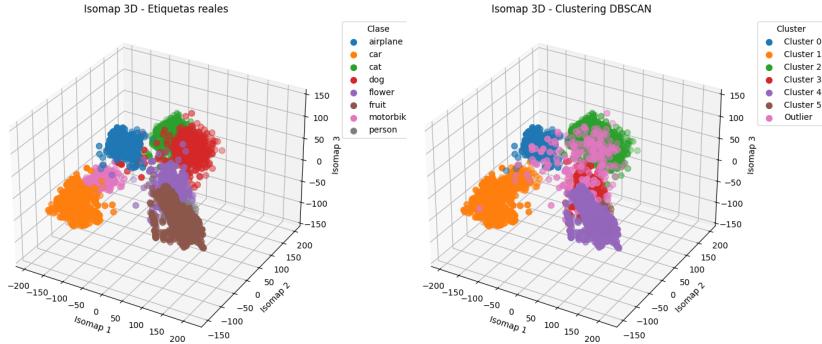


Figura 32: Visualización con ISOMAP 3D

### Comparación métodos de clustering

El algoritmo K-Means permitió identificar una segmentación efectiva del conjunto de imágenes reales, mostrando agrupamientos coherentes en clases como car, person, fruit y motorbike, con una precisión superior al 99.5 % en algunos casos. Sin embargo, presentó solapamientos entre clases como dog y flower, probablemente debido a similitudes visuales o a la pérdida de información causada por la reducción de dimensionalidad con PCA. Además, es importante remarcar que K-Means requiere definir el número de clusters de antemano y es sensible a la presencia de outliers, lo que puede afectar negativamente la estabilidad del agrupamiento.

Por otro lado, el clustering jerárquico aglomerativo con enlace ward obtuvo los mejores resultados globales, con un ARI de 0.9379 y un V-Measure de 0.9564. No solo logró una alineación precisa con las etiquetas reales, sino que también captó subestructuras internas relevantes, como la división de la clase fruit en dos agrupamientos bien definidos. Esta capacidad de representar jerarquías lo convierte en una herramienta particularmente útil para explorar categorías visuales complejas y heterogéneas.

En cuanto a DBSCAN, su enfoque basado en densidad permitió detectar agrupamientos sólidos en clases como airplane, fruit y person, además de identificar outliers de forma natural. No obstante, mostró dificultades para separar las clases cat y dog, posiblemente debido a representaciones visuales similares. En el contexto de este trabajo, donde se busca detectar anomalías visuales generadas artificialmente, estas técnicas de clustering no supervisado ofrecen una base estructural inicial para contrastar con la distribución de imágenes sintéticas. Así, contribuyen a validar la hipótesis de que las imágenes anómalas pueden diferenciarse estructuralmente sin necesidad de etiquetas previas.

Entre los tres algoritmos evaluados, el clustering jerárquico y DBSCAN son los que mejor respaldan la hipótesis del trabajo, ya que permiten identificar patrones estructurales y detectar desviaciones sin requerir etiquetas previas. El clustering jerárquico facilita una exploración detallada de las relaciones entre imágenes reales, lo que resulta útil como línea base para contrastar la presencia de anomalías. Por su parte, DBSCAN destaca por su capacidad para detectar outliers y agrupamientos de forma arbitraria, lo que lo vuelve especialmente valioso para identificar imágenes sintéticas que no encajan en la estructura natural del conjunto. En cambio, K-Means, aunque útil como aproximación inicial, resulta menos adecuado en este contexto debido a su sensibilidad a outliers y la necesidad de fijar un número de clusters a priori.

### 3.1.3. Clustering Jerarquico con anomalias

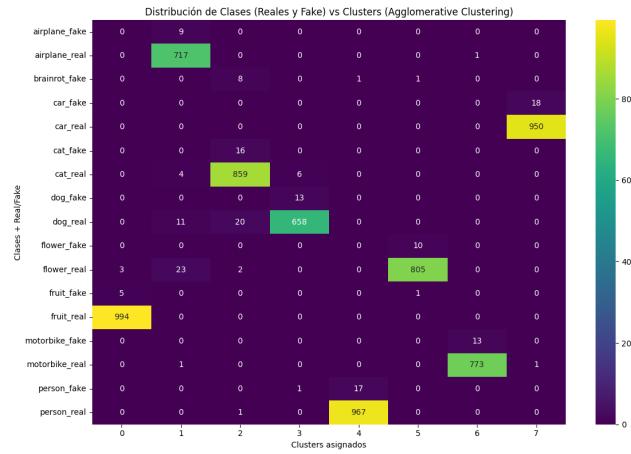


Figura 33: Matriz de Confusión

Dado que el clustering jerárquico aglomerativo tuvo mejor performance de acuerdo a los parámetros de validación anteriormente presentados decidimos aplicarlo pero al dataset inoculado con anomalías para evaluar su comportamiento. La Figura 30 ilustra cómo se distribuyen las etiquetas reales en los distintos clusters generados de manera no supervisada mediante clustering jerárquico en este nuevo dataset. El objetivo de esta visualización es evaluar si los agrupamientos obtenidos coinciden con las clases originales y, además, examinar si el algoritmo logra diferenciar entre imágenes reales y sintéticas (anomalías). En línea con los resultados sobre el conjunto original sin anomalías, observamos que el algoritmo logra separar de manera efectiva muchas de las clases reales en agrupamientos coherentes. Ahora bien, como podría esperarse para las anomalías "sútiles", es decir aquellas son imitaciones realistas, estas son agrupadas dentro de los mismos clusters que sus contrapartes reales. El escenario es otro cuando se trata de agrupar las anomalías de clase "brainrot", pues no se diferencian en un cluster único sino que se van solapando en distintos agrupamientos de manera altamente diversificada. Esto sugiere que el algoritmo no logró ser resiliente ante ese tipo de inoculaciones ni diferenciar imitadores realistas de las imágenes originales.

## Conclusiones

Los resultados obtenidos a partir de los tres algoritmos de clustering evaluados —K-Means, DBSCAN y clustering jerárquico aglomerativo— sobre los componentes de vectores de atributos (extraídos de VGG16) de las imágenes permiten validar parcialmente la hipótesis del trabajo. En particular, el clustering jerárquico y DBSCAN demostraron ser los enfoques más adecuados para detectar estructuras internas dentro del conjunto de imágenes naturales sin anomalías.

K-Means, aunque útil como aproximación inicial, presentó limitaciones notables: su sensibilidad a *outliers* y la necesidad de definir previamente el número de clústeres afectaron negativamente su desempeño, en especial en clases con alta variabilidad visual.

El clustering jerárquico aglomerativo se destacó por su capacidad para representar relaciones jerárquicas y captar subestructuras dentro de clases complejas, como se evidenció en la división interna de la categoría *fruit*. Además, obtuvo los mejores resultados cuantitativos del análisis, con un ARI de 0,9379 y un V-Measure de 0,9564, lo que respalda su utilidad como herramienta sólida para explorar agrupamientos visuales.

A partir de estos resultados, se seleccionó el clustering jerárquico como único algoritmo para el segundo experimento, en el cual se introdujeron imágenes sintéticas (IFGIA) al conjunto original. En esta fase, se observó que las anomalías visuales sutiles —aquellas que imitan de forma realista elementos naturales— tienden a agruparse junto a sus contrapartes reales, lo que refleja una limitación del enfoque no supervisado ante modificaciones visuales de baja intensidad. En cambio, las anomalías grotescas (o “brainrot”) no formaron un clúster único ni se separaron de forma consistente, sino que se dispersaron entre distintos grupos, lo que evidencia que la resiliencia del algoritmo frente a este tipo de alteraciones aún es limitada.

En conjunto, se concluye que el clustering jerárquico aglomerativo si bien resultó muy eficaz para separar clases naturales, al exponerlo a anomalías se presentan resultados diversos. Por un lado, las más sutiles (imitaciones realistas de las imágenes originales) se agrupan en los mismos clusters que la de las clases originales. Por otro, las anomalías grotescas se distribuyen pero sin un patrón claro de homegenidad. Futuras investigaciones podrían contemplar un *k* más grande para evaluar si existe separabilidad en sub-clusters que puedan diferenciar, por ejemplo, gatos reales de gatos falsos.

## Referencias

- [1] Saeedi, J., & Giusti, A. (2021). *Semi-supervised visual anomaly detection based on convolutional autoencoder and transfer learning*. Dalle Molle Institute for Artificial Intelligence (IDSIA USI-SUPSI). arXiv preprint arXiv:2107.11265.
- [2] Yang, Q., & Guo, R. (2022). *An unsupervised method for industrial image anomaly detection with vision transformer-based autoencoder*. Sensors, 22(24), 9540. <https://doi.org/10.3390/s22249540>
- [3] Yu, X., Wang, Y., Chen, Y., Tao, Z., Xi, D., Song, S., Niu, S., & Li, Z. (2024). *Fake artificial intelligence generated contents (FAIGC): A survey of theories, detection methods, and opportunities*. arXiv preprint arXiv:2405.00711.
- [4] Sánchez Vinces, B. V., Schubert, E., Zimek, A., & Cordeiro, R. L. F. (2025). *A comparative evaluation of clustering-based outlier detection*. Data Mining and Knowledge Discovery, 39(13). <https://doi.org/10.1007/s10618-024-01086-z>