

---

# Pre TP2: Procesamiento y Agrupamiento de Imágenes de Flores

---

**Ricardo G. Amarilla**

Data Mining en Ciencia y Tecnología

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

amarilla\_ricardo@hotmail.com

**19 de junio de 2025**

## Abstract

Este trabajo tiene como objetivo familiarizarse con técnicas básicas de procesamiento de imágenes, aplicadas a un conjunto de datos compuesto por imágenes de flores. Para ello, se realizaron distintas manipulaciones orientadas al preprocesamiento del dataset, incluyendo estandarización, visualización por especie, conversión a escala de grises y binarización, así como la aplicación de filtros de suavizado. Se calcularon imágenes promedio globales y por especie, en color y en blanco y negro, con el fin de identificar patrones visuales comunes. Además, se aplicó reducción de dimensionalidad mediante análisis de componentes principales (PCA) y agrupamiento no supervisado utilizando la biblioteca clustimage. Estas técnicas permitieron explorar la estructura interna del conjunto y detectar agrupamientos naturales en base a similitudes visuales.

## 1. Introducción

El procesamiento de imágenes es una herramienta fundamental dentro del análisis de datos visuales. En este trabajo se busca familiarizarse con distintas técnicas básicas de manipulación de imágenes digitales, como parte del preprocesamiento necesario para tareas posteriores de análisis no supervisado. A través de estas manipulaciones, se prepara el conjunto de datos para la exploración de patrones y la detección de agrupamientos naturales, permitiendo así identificar similitudes estructurales entre las imágenes y evaluar posibles estructuras latentes en el conjunto.

## 2. Preprocesamiento de los Datos

### 2.1. Carga y estandarización de imágenes

Las imágenes fueron cargadas en formato .png, junto con sus respectivas etiquetas. Para asegurar la comparabilidad entre ellas, se estandarizaron las siguientes características:

- **Color:** se mantuvo el formato original RGBA, incluyendo el canal alfa.
- **Tamaño:** todas las imágenes fueron ajustadas a 128x128 píxeles.
- **Tipo de dato:** se verificó que todas utilizan el tipo uint8.
- **Rango:** los valores de píxeles se encuentran en el intervalo [0, 255].

Se detectó una única imagen con un tamaño distinto: 0208.png, de 208x208 píxeles. Esta fue redimensionada a 128x128 utilizando la función `resize()` de la biblioteca PIL, con el fin de garantizar la uniformidad del conjunto. Ver tabla 1.

Tamaño	Canales	Tipo de dato	Cantidad	Mínimo_global	Máximo_global
128x128	4	uint8	209	0	255
208x208	4	uint8	1	0	255

Cuadro 1: Resumen de características técnicas de las imágenes utilizadas.

## 2.2. Exploración visual por especie

También se exploraron los subconjuntos de imágenes por especie, visualizando varias imágenes por clase.

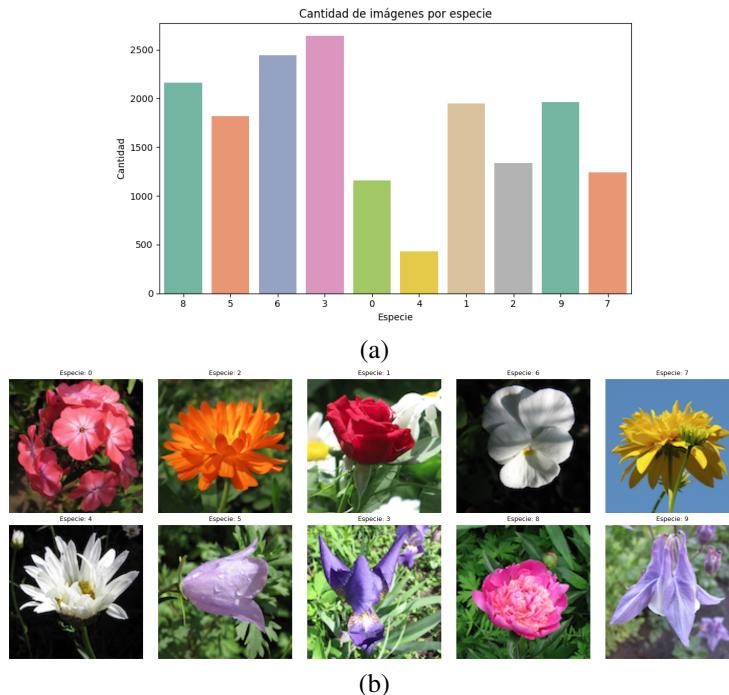


Figura 1: Visualización de (a) Cantidad de imágenes por especie y (b) Subconjuntos de flores por especie.

Del análisis de la figura 1, apartado (a), se observa que la distribución de imágenes por especie no es uniforme. La especie con menor cantidad de imágenes es la número 4, mientras que la más representada corresponde a la especie 3. Por otro lado, en el apartado (b), se graficó una imagen por especie, seleccionada de manera aleatoria. Estas imágenes no fueron elegidas por su representatividad morfológica, sino únicamente con fines ilustrativos.

## 3. Manipulación de Datos

### 3.1. Escala de grises y binarización

Una imagen fue convertida a escala de grises utilizando PIL. Para binarizarla, es necesario aplicar un umbral: todos los píxeles por debajo se vuelven negros (0), y los demás blancos (255). Esto se puede hacer con `cv2.threshold()`. Los resultados se pueden observar en la figura 2.

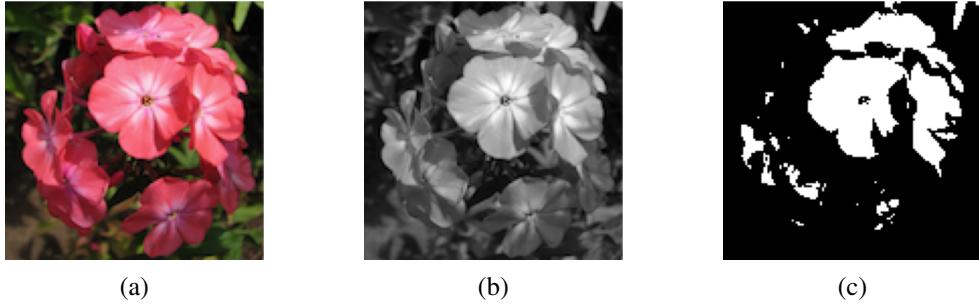


Figura 2: Comparación visual: (a) original, (b) en escala de grises, (c) binarizada.

### 3.2. Imágenes aleatorias

En este trabajo se implementaron dos técnicas básicas de manipulación de imágenes digitales. En primer lugar, se aplicó una mezcla aleatoria de píxeles dentro de una misma imagen, utilizando la función `np.random.shuffle`, lo que permitió desestructurar su composición visual sin alterar la información de color. En segundo lugar, se generaron composiciones híbridas mediante la combinación de secciones (patches) provenientes de diferentes imágenes, uniendo fragmentos como cuadrantes para formar una nueva imagen, los resultados puede ver en la figura 3.

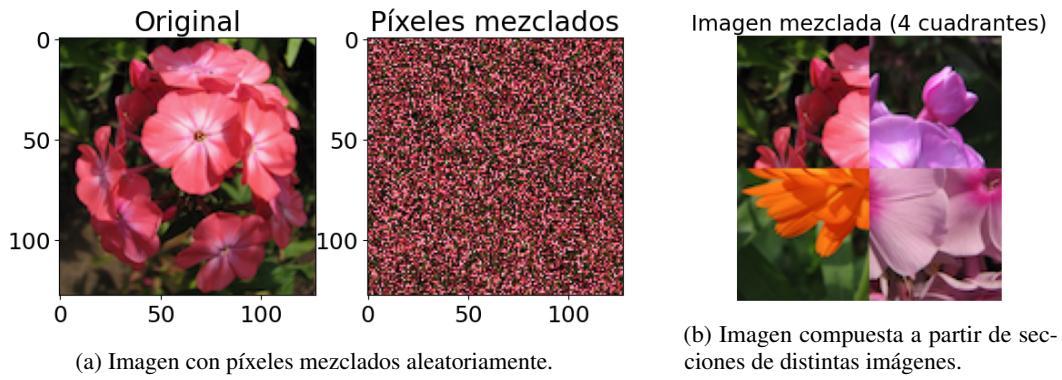


Figura 3: Comparación entre dos métodos de mezcla de imágenes: (a) por píxeles y (b) por secciones.

### 3.3. Filtros aplicados

En la figura 4 se presenta una comparación visual entre una imagen original y sus versiones filtradas mediante dos técnicas básicas de procesamiento digital: el filtro de media y el filtro de mediana. El filtro de media suaviza la imagen calculando el promedio de los valores de píxeles vecinos, lo cual resulta útil para reducir ruido suave o granular, aunque puede producir una pérdida de detalles, especialmente en los bordes. Por otro lado, el filtro de mediana reemplaza cada píxel por el valor central dentro de una ventana ordenada de píxeles, lo que lo hace especialmente eficaz para eliminar ruido impulsivo del tipo "sal y pimienta", preservando mejor las estructuras y contornos de la imagen original.

Esta comparación permite observar cómo cada filtro responde a distintos tipos de ruido, y destaca la importancia de elegir la técnica adecuada según las características específicas de la imagen y del problema a resolver.



Figura 4: Comparación de Filtros de Media y Mediana en Imágenes en Escala de Grises

La documentación para llevar a cabo los filtros fue obtenido de [4].

### 3.4. Imagen promedio

Para obtener las imágenes promedio por especie en blanco y negro, se definió una función que convierte cada imagen a escala de grises y luego aplica binarización mediante umbralado fijo (`cv2.threshold`). A partir de estas versiones binarizadas, se calculó el promedio por especie promediando los píxeles correspondientes. El resultado permite observar la estructura binaria media de cada clase, resaltando contornos y formas dominantes independientemente del color.

Para comparación, también se calcularon los promedios por especie en formato color. En este caso, se promediaron directamente los valores RGB de todas las imágenes asociadas a cada clase, reconstruyendo así una imagen representativa del contenido visual promedio de cada especie.

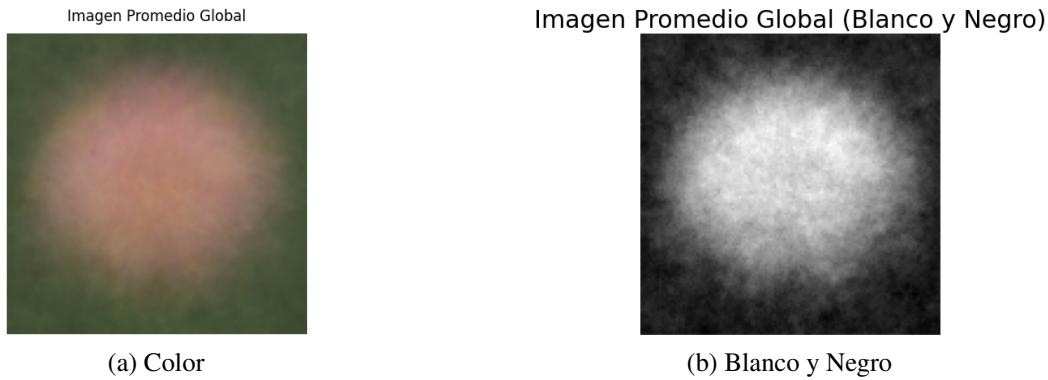


Figura 5: Comparación de imagen promedio global: (a) en color, (b) en escala de grises.

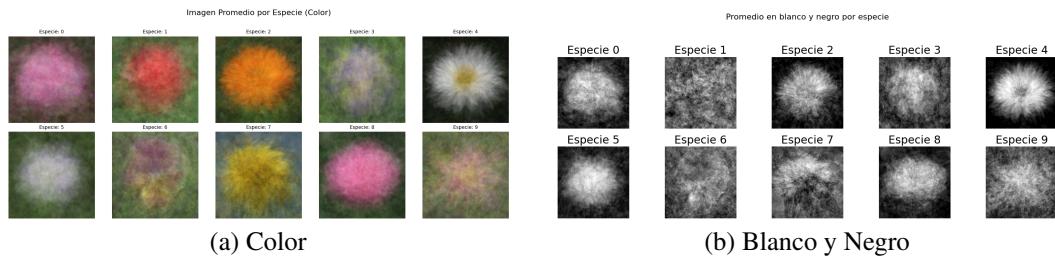


Figura 6: Imágenes promedio por especie: (a) en color, (b) en escala de grises.

Del análisis de las Figuras 5 y 6 se observa que, en formato color, los promedios permiten identificar formas generales que remiten a flores, aunque con bordes poco definidos y cierta opacidad visual. Al

convertir las imágenes a blanco y negro mediante binarización, no se obtiene una mejora perceptible en cuanto a nitidez o reconocimiento estructural. Desde esta perspectiva, el promedio en color conserva mayor información visual y facilita, aunque de manera limitada, la interpretación de las formas florales.

## 4. Búsqueda de Features

### 4.1. Distribución de píxeles por especie

Se analizaron histogramas de valores RGB por especie. En algunos casos, se observaron diferencias en rangos de color que podrían ser indicativos de clase.

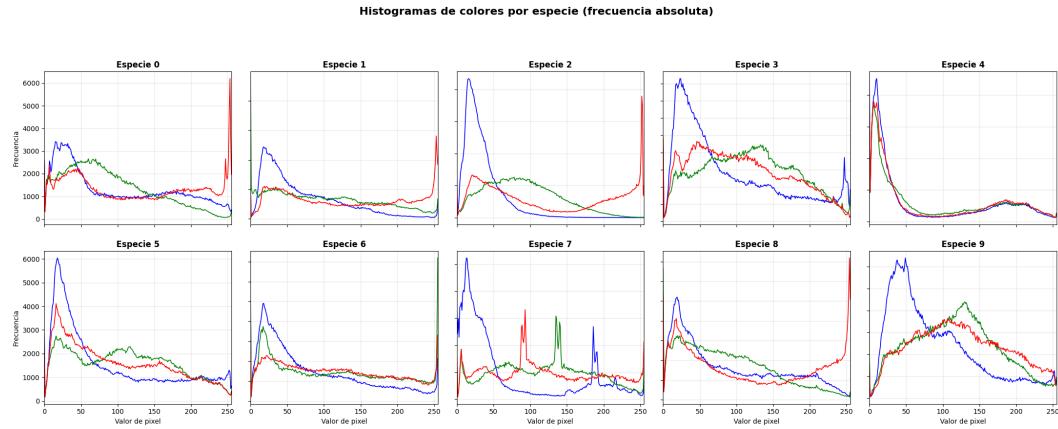


Figura 7: Histogramas de colores por especie

Al observar las distribuciones de valores de píxeles por especie figura 7, se identifican ciertos patrones que permiten distinguir a algunas especies en rangos específicos de color. La especie 0 presenta un pico muy pronunciado en el canal rojo entre 240 y 255, acompañado de una caída en los canales verde y azul, lo que sugiere una fuerte presencia de tonos rojizos. La especie 1 tiene un comportamiento similar, aunque con un pico rojo menos acentuado y valores de rojo y verde más estabilizados, indicando una coloración más equilibrada.

Por otro lado, la especie 4 muestra picos en los tres canales (R, G y B) en el rango de 0 a 50, seguidos de una caída continua hacia los valores más altos, lo que indica una predominancia de tonos oscuros. La especie 5 comparte este patrón, pero la disminución de intensidad hacia el final es más suave, lo que sugiere una transición de color menos abrupta.

Estas diferencias permiten concluir que sí es posible distinguir algunas especies en función de sus distribuciones de color, particularmente en los rangos extremos de los valores RGB. Este tipo de análisis puede ser útil para tareas de clasificación automática basadas en información cromática.

### 4.2. PCA (Componentes Principales)

Se aplicó Análisis de Componentes Principales (PCA) al conjunto de datos de imágenes aplastadas (210 muestras con 49152 variables) con el objetivo de reducir la dimensionalidad y conservar la mayor cantidad posible de varianza. El gráfico del codo que se observa en la figura (a) de 8 indicó que un número reducido de componentes de 153 que capturan la mayoría de la información, destacando un punto de inflexión a partir del cual los componentes adicionales aportan menor valor.

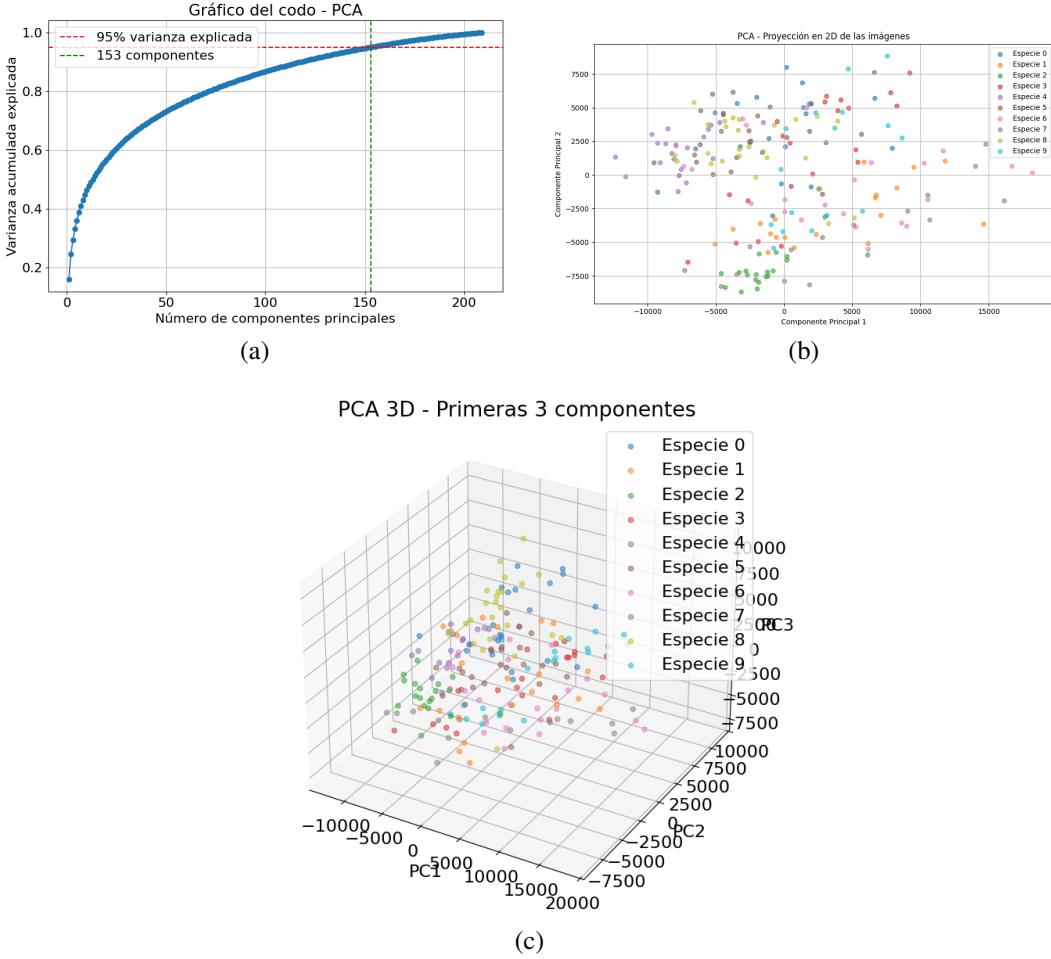


Figura 8: Análisis de componentes principales (PCA): (a) gráfico del codo, (b) proyección con 2 componentes principales, (c) proyección con 3 componentes.

Al proyectar las muestras sobre los dos primeros componentes (PC1 y PC2) y los tres primeros componentes (PC1, PC2 y PC3) colorearlas por especie (b) y (c) de la figura 8, no se observaron patrones de agrupamiento claramente definidos; sin embargo, en algunos casos se notaron leves tendencias de separación, aunque difusas, lo que indica que la información relevante para clasificar las especies podría estar distribuida en componentes más allá de los dos primeros.

## 5. Agrupamiento de Imágenes con clustimage

El análisis inicial se basó en un agrupamiento externo utilizando etiquetas conocidas, lo cual representa un enfoque supervisado. Con el objetivo de contrastar esta clasificación con una estrategia no supervisada, se aplicó la herramienta clustimage, que permite agrupar imágenes según su similitud visual sin necesidad de etiquetas previas. Esta herramienta extrae características visuales de las imágenes, aplica un algoritmo de clustering (como k-means) y determina automáticamente el número óptimo de clústeres utilizando métricas internas como el silhouette score. En este caso, se identificaron cinco clústeres principales.

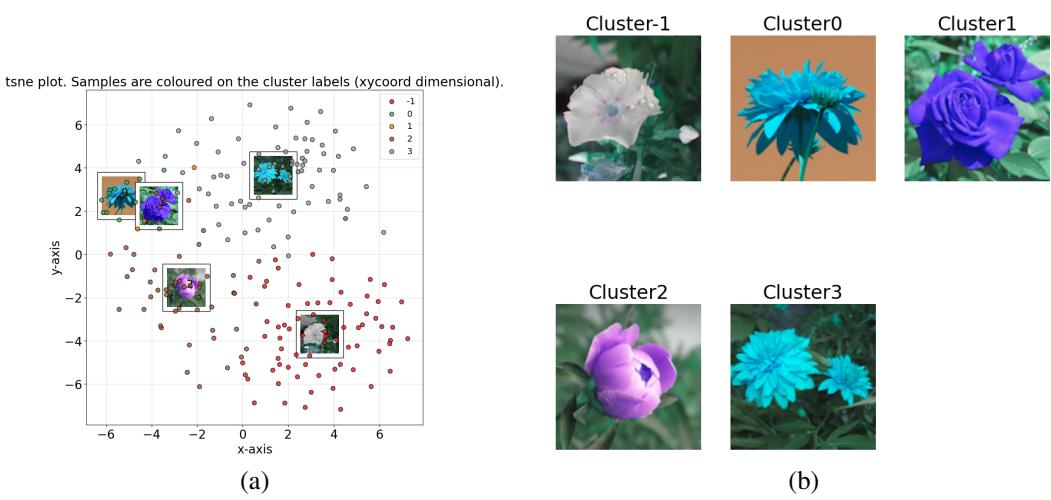


Figura 9: Resultados del agrupamiento de imágenes utilizando `clustimage`: (a) proyección visual de grupos detectados, (b) promedio representativo por clúster.

La distribución de las imágenes se visualizó mediante un gráfico de dispersión en un espacio reducido por PCA, donde se destacan las cinco imágenes más representativas (centrodes visuales) de cada grupo. Esta visualización permite explorar la coherencia interna de los grupos formados y analizar su correspondencia o divergencia respecto a las clases originales, aportando una nueva perspectiva sobre la estructura latente de los datos. Los resultados se pueden observar en la figura 9

La biblioteca utilizada para la extracción de características fue `clustimage` y la documentación es [3], [1] y [2].

## 6. Conclusiones

A partir de las distintas etapas del procesamiento de imágenes aplicadas en este trabajo, fue posible explorar visualmente la estructura interna del conjunto de datos. El preprocessamiento permitió estandarizar las imágenes y resaltar patrones comunes, mientras que las imágenes promedio ofrecieron una visión general de las características compartidas por especie. La reducción de dimensionalidad mediante PCA facilitó la visualización de posibles agrupamientos, y el uso de `clustimage` permitió detectar estructuras naturales en los datos sin utilizar etiquetas. Si bien algunas especies mostraron patrones visuales y cromáticos distintivos, otras resultaron más difíciles de diferenciar, lo que evidencia tanto el potencial como las limitaciones del análisis visual y no supervisado para tareas de clasificación.

## Referencias

- [1] Abdullah Erdogant. `clustimage` 1.3.3 - pypi. <https://pypi.org/project/clustimage/1.3.3/>, 2023. Accessed: 2025-06-19.
- [2] Abdullah Erdogant. `clustimage` documentation and examples. <https://erdogant.github.io/clustimage/pages/html/Examples.html>, 2023. Accessed: 2025-06-19.
- [3] Abdullah Erdogant. `clustimage` github repository. <https://github.com/erdogant/clustimage>, 2023. Accessed: 2025-06-19.
- [4] scikit-image contributors. Image filters - scikit-image tutorials. [https://scikit-image.org skimage-tutorials/lectures/1\\_image\\_filters.html](https://scikit-image.org skimage-tutorials/lectures/1_image_filters.html), 2023. Accessed: 2025-06-19.