

## Laboratorio 9

**Objetivo:** Familiarizar al estudiante con la implementación y uso de algoritmos de aprendizaje no supervisado.

**Enunciado:** Lea las instrucciones del documento y resuelva el enunciado en Python. Al inicio de su documento adjunte su nombre y carnet como un comentario. Este trabajo es de carácter individual, tampoco se aceptarán códigos que no hayan sido desarrollados por su persona.

Para este laboratorio se trabajará con imágenes de películas. Busque de 3 a 5 “cuadros” (*frames*) de sus películas favoritas. Almacene dichas imágenes, deberá entregarlas junto a su laboratorio y sus resultados.

El objetivo de este laboratorio consiste en utilizar algunos de los algoritmos de *clustering* vistos en clase con el fin de encontrar la paleta de colores principal de una imagen. Para ello realizaremos *clustering* de tipo *K-means* y *K-medoids* utilizando el algoritmo de Lloyd.

Para ello deberá implementar los siguientes métodos:

1. Método de carga `load_image(filename, resize)` que se encarga de cargar el archivo de imagen con nombre `filename`, cambia su tamaño al tamaño `resize` y la retorna como un arreglo multinivel de `numpy`.
2. Método de cálculo de distancias euclidianas `euclidean_distance(p1, p2)` que recibe dos puntos `p1` y `p2`, y retorna la distancia euclidiana entre ellos. Asuma que los puntos pueden ser `n`-dimensionales.
3. Método de cálculo de distancias de Manhattan `manhattan_distance(p1, p2)` que recibe dos puntos `p1` y `p2`, y retorna la distancia Manhattan entre ellos. La distancia Manhattan se calcula como la suma de las diferencias absolutas en cada una de las dimensiones para los puntos. Asuma que los puntos pueden ser `n`-dimensionales.
4. Método de selección de centroides `nearest_centroid(point, centroids, distance)` que recibe un punto `point`, la lista de centroides actuales `centroids` y una función `distance(p1, p2)` que calcula la distancia entre dos puntos. Esta función retorna una tupla (`idx_centroid, distance`) con el índice del centroide más cercano y su distancia respectiva.
  - a. Note que debe implementar su función de manera que funcione en un problema en que los datos pueden tener `n` dimensiones, no necesariamente 3.

5. Método `lloyd(data, k, iters, type, distance)` que recibe un conjunto de datos `n`-dimensionales `data`, un entero positivo `k` que especifica la cantidad de *clusters* en los que se desea agrupar la información, un entero positivo `iters` que especifica la cantidad de iteraciones del ciclo a ejecutar, una hilera `type` que puede ser “means” o “medioids” y una hilera `distance` que puede ser “euclidean” o “manhattan”. Debe implementar el algoritmo de Lloyd de manera que inicialice los puntos de manera aleatoria y luego ejecuta las iteraciones adecuadas del ciclo. Dependiendo del tipo deberá ejecutar el algoritmo como *K-means* o *K-medioids*, y deberá utilizar la función de distancia elegida por el usuario. Retorne los centroides obtenidos al final de la ejecución, así como el “error” total de dicho clustering (puede calcularlo como la suma de distancias o la suma de distancias cuadradas).
6. Utilice su algoritmo implementado para calcular la paleta de colores de las imágenes. Saque los 5 colores principales que componen cada imagen. Extraiga la paleta de colores probando con las 4 mezclas generadas por nuestro algoritmo: means/medioids con euclidean/manhattan para comprobar las variaciones que esto puede generar. Dado que la inicialización aleatoria puede llevar a soluciones locales, ejecute cada experimento 3 veces y tome los centroides de la solución con menor error. Finalmente para cada una de las imágenes, guarde una muestra de la imagen al lado de las 4 paletas de colores extraídas por los algoritmos. Para ello puede usar matplotlib o algún programa de imágenes.
  - a. Recomendación: trabajar con imágenes muy grandes resulta computacionalmente costoso, por lo que puede reducir sus imágenes a tamaños más manejables por medio del `resize` incluido en el método de carga. Sin embargo, si la reduce mucho se pierde mucha información por lo que se le recomienda no reducir la imagen abajo de 128x128.
  - b. Entregue junto con su código ejecutable todas las imágenes obtenidas.
  - c. Recuerde que las imágenes son tratadas como tipo de dato `uint8`, sin embargo esto genera errores de redondeo a la hora de hacer cálculos computacionales. Por lo que se le recomienda convertir sus tipos de datos a flotante.