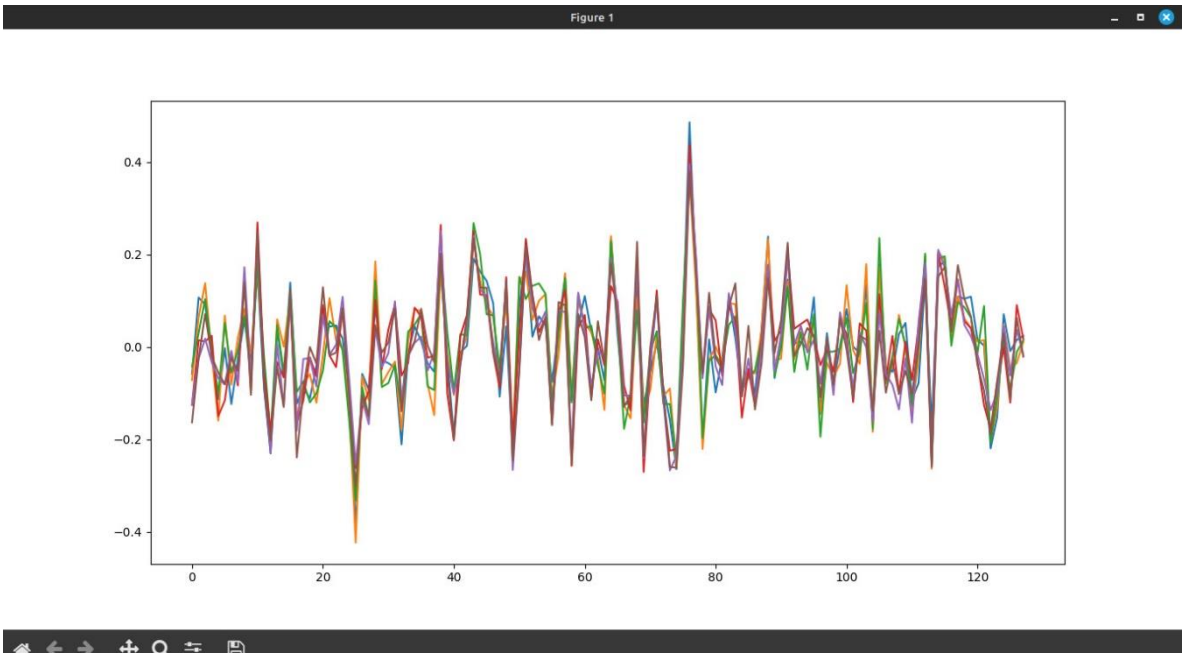
```
#Librerias
import face_recognition
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Load image:
img5 = face_recognition.load_image_file("5.jpg")
img6 = face_recognition.load_image_file("6.jpg")
img7 = face_recognition.load_image_file("7.jpg")
img8 = face_recognition.load_image_file("8.jpg")
img9 = face_recognition.load_image_file("9.jpg")
img10 = face_recognition.load_image_file("10.jpg")

# Calculate the encodings for every face of the image:
encodings5 = face_recognition.face_encodings(img5)
encodings6 = face_recognition.face_encodings(img6)
encodings7 = face_recognition.face_encodings(img7)
encodings8 = face_recognition.face_encodings(img8)
encodings9 = face_recognition.face_encodings(img9)
encodings10 = face_recognition.face_encodings(img10)

# Show the first encoding:

print(len(encodings5))
print(encodings5[0])
print('')
print(len(encodings6))
print(encodings6[0])
print('')
print(len(encodings7))
print(encodings7[0])
print('')
print(len(encodings8))
print(encodings8[0])
print('')
print(len(encodings9))
print(encodings9[0])
print('')
print(len(encodings10))
print(encodings10[0])
print('')
plt.plot(encodings5[0])
plt.plot(encodings6[0])
plt.plot(encodings7[0])
plt.plot(encodings8[0])
plt.plot(encodings9[0])
plt.plot(encodings10[0])
plt.show()

# Create data (encodings of the images):
data_A = np.vstack((encodings5[0], encodings6[0], encodings7[0]))
```

```
data_B = np.vstack((encodings8[0], encodings9[0], encodings10[0]))

# Convertir datos a tipo float32
data_A = np.float32(data_A)
data_B = np.float32(data_B)

# Reorganizar los datos para tener solo 2 dimensiones
data_A = data_A.reshape(-1, 1)
data_B = data_B.reshape(-1, 1)

# Ejecutar K-means clustering para grupo A
criteria_A = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 20,
1.0)
ret_A, label_A, center_A = cv2.kmeans(data_A, 2, None, criteria_A, 10,
cv2.KMEANS_RANDOM_CENTERS)

# Ejecutar K-means clustering para grupo B
criteria_B = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 20,
1.0)
ret_B, label_B, center_B = cv2.kmeans(data_B, 2, None, criteria_B, 10,
cv2.KMEANS_RANDOM_CENTERS)

# Obtener los índices de cada grupo
indices_A = np.where(label_A.ravel() == 0)[0]
indices_B = np.where(label_B.ravel() == 0)[0]

# Obtener los puntos correspondientes a cada grupo
A1 = data_A[indices_A]
A2 = data_A[np.where(label_A.ravel() == 1)[0]]
B1 = data_B[indices_B]
B2 = data_B[np.where(label_B.ravel() == 1)[0]]

# Plot the results
plt.plot(A1[:, 0], 'b-', linewidth=1, label='Group A1')
plt.plot(A2[:, 0], 'b-', linewidth=1, label='Group A2')
plt.plot(B1[:, 0], 'g-', linewidth=1, label='Group B1')
plt.plot(B2[:, 0], 'g-', linewidth=1, label='Group B2')

  plt.scatter(center_A[:, 0], center_A[:, 0], s=100, c='r',
  label='Centers A')
  plt.scatter(center_B[:, 0], center_B[:, 0], s=100, c='m',
  label='Centers B')
  plt.legend()
  plt.show()
```



“Encodings” de las imágenes, una grafica que junta todo el “encodings” de todas las imágenes juntas

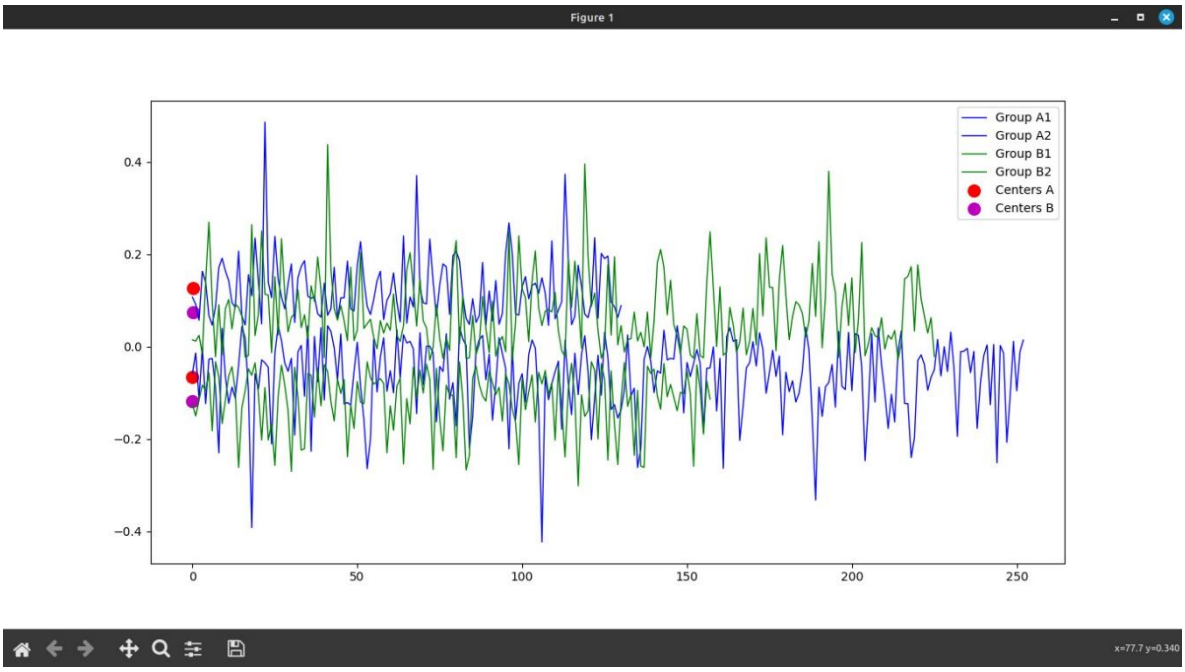


Imagen usando el algoritmo K-means para separar en closters cada grupo, el grupo A1 Y A2 son las fotos del profesor, los grupos B1 y B2 son mis fotos, le intente dar un color a cada grupo para que hubieran líneas, pero como están muy juntos los puntos no se logran apreciar al completo y se ve con poca legibilidad.

Lo primero que hacemos es cargar las imágenes para poder sacar su “encoding”, después creamos dos matrices data_A y data_B, que contienen los encodings de las imágenes que deseas clasificar en los grupos A y B, respectivamente.

Se utiliza np.float32() para convertir los datos data_A y data_B al tipo float32. Luego, se utiliza reshape() para reorganizar los datos de manera que tengan solo 2 dimensiones.

Luego, aplicamos el algoritmo de clustering K-means separadamente para cada grupo. Creamos dos conjuntos de criterios diferentes, criteria_A y criteria_B, y utilizamos estos criterios para ejecutar cv2.kmeans en data_A y data_B, respectivamente.

Después de obtener las etiquetas y los centros para cada grupo, separamos los datos en subconjuntos A1, A2, B1 y B2 según las etiquetas.

Los valores A1, A2, B1 y B2 representan los subconjuntos de puntos obtenidos después de aplicar el algoritmo de K-means y separar los datos en los grupos A y B.

Finalmente, trazamos los puntos de los grupos A y B, junto con los centros de cada grupo, en un gráfico para visualizar los resultados. Asegúrate de ajustar los colores y el estilo de trazado según tus preferencias.