Reconocimiento facial - Caso Práctico 2

Librerias necesarias

In [ ]:
```
#instalación de librerías necesarias
%pip install face-recognition
%pip install opencv-contrib-python  #Adición de módulos avanzados para ma
%pip install tabulate
```

```
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: face-recognition in /home/walmache/.local/l
ib/python3.10/site-packages (1.3.0)
Requirement already satisfied: Click>=6.0 in /usr/lib/python3/dist-package
s (from face-recognition) (8.0.3)
Requirement already satisfied: numpy in /home/walmache/.local/lib/python
3.10/site-packages (from face-recognition) (1.26.4)
Requirement already satisfied: face-recognition-models>=0.3.0 in /home/wal
mache/.local/lib/python3.10/site-packages (from face-recognition) (0.3.0)
Requirement already satisfied: Pillow in /usr/lib/python3/dist-packages (f
rom face-recognition) (9.0.1)
Requirement already satisfied: dlib>=19.7 in /usr/local/lib/python3.10/dis
t-packages/dlib-19.24.99-py3.10-linux-x86_64.egg (from face-recognition)
(19.24.99)
Note: you may need to restart the kernel to use updated packages.
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: opencv-contrib-python in /home/walmache/.lo
cal/lib/python3.10/site-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.19.3 in /home/walmache/.local/lib/
python3.10/site-packages (from opencv-contrib-python) (1.26.4)
Note: you may need to restart the kernel to use updated packages.
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: tabulate in /home/walmache/.local/lib/pytho
n3.10/site-packages (0.9.0)
Note: you may need to restart the kernel to use updated packages.
```

Paso 1.- Extracción de rostros para entrenamiento posterior del modelo

In [ ]:
```
#Importar módulos necesarios
import cv2  #Manejo de imagenes
import os   #Manejo de archivos/directorios
import face_recognition        # Linreria de reconocimiento facial
from datetime import datetime   # Para el registro de fecha y hora de asi
from tabulate import tabulate   #Tabulación de resultados de aistencia

import imghdr  #Para validar si un archivo es imagen
import sys     #Manejo de comandos de S.O.
import matplotlib.pyplot as plt   #Para gráficos
```

In [ ]:
```
#Constantes a usar
RED = "\033[91m"
```

```
GREEN = "\033[92m"
YELLOW = "\033[93m"
BLUE = "\033[94m"
RESET = "\033[0m"
```

In [ ]:

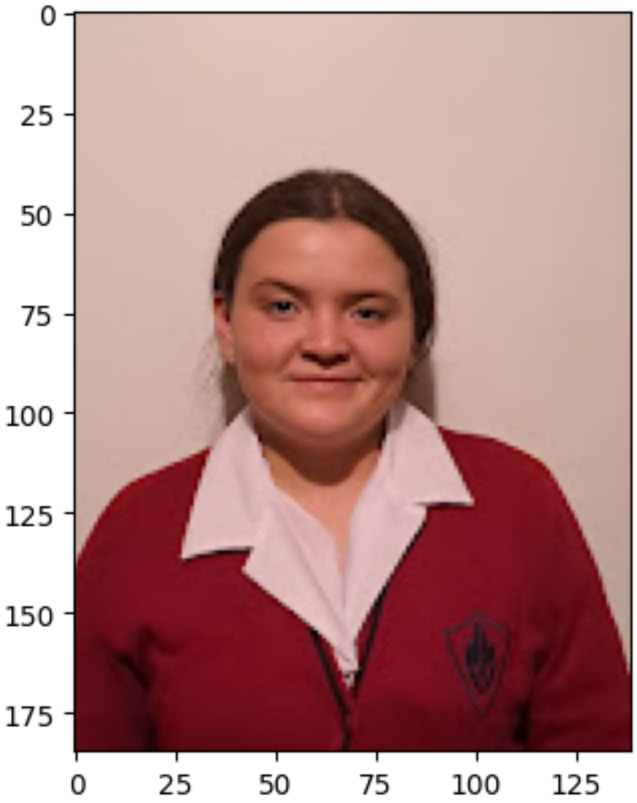
```
#Definición de funciones necesarias
#Evalua si un archivo es imagen válida
def isImage(filePath):
    return imghdr.what(filePath) is not None

#Evalua existencia de directorios
def existsDirectory(directoryPath):
    return os.path.exists(directoryPath) and os.path.isdir(directoryPath)

#Visualizasión de mensajes
def message(text, color):
    print(f"{color}{text}{RESET}")

#Registro de Asistencias
def attendanceRegister(student):
    if student!='Desconocido':
        for registro in attendance:
            # Verificar si el estudiante ya está registrado
            if registro['nombre'] == student:
                ##message(f"{student} ya está registrado.",RED)
                return
        hora_registro = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        registro = {'nombre': student, 'hora': hora_registro}
        attendance.append(registro)

def printReport():
    tabla_asistencia = [[record['nombre'], record['hora']] for record in
    message("\nRegistros de asistencia:".center(100),BLUE)
    print(tabulate(tabla_asistencia, headers=["Nombre del Estudiante", "H
```

In [ ]:

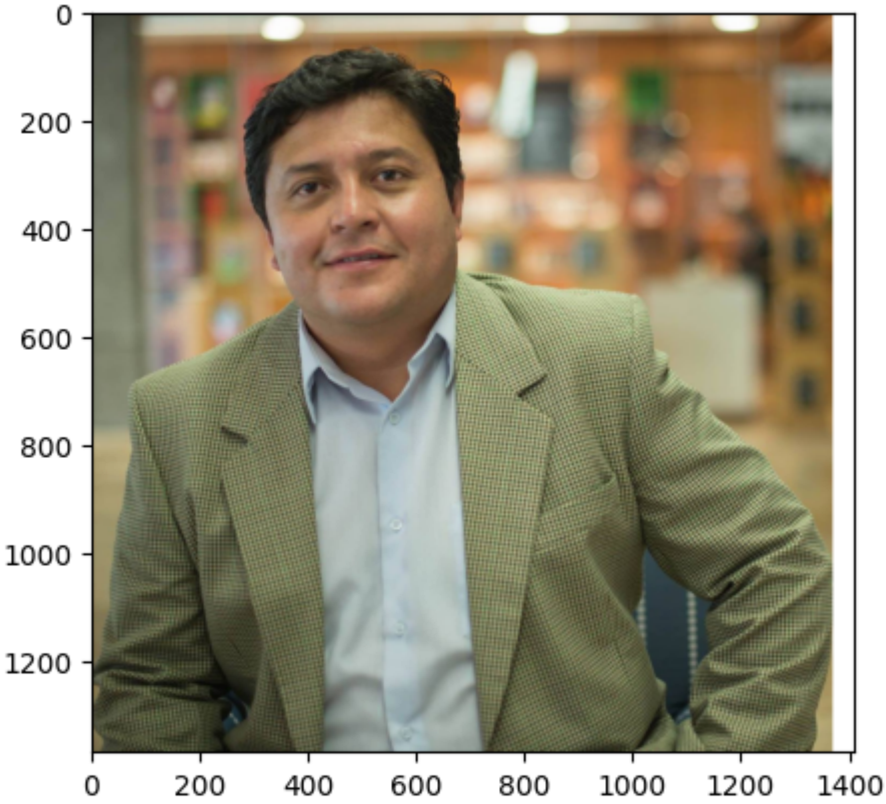
```
# Reconocimiento y alamacenamiento de rostros de imagenes aleatorias

sourceImages = "sourceImages"  #Directorio que almacena imagenes que cont
facesRepo    = "facesRepo"     #Direcorio donde se almacenaran los rostro

#Existe directorio que contiene imagenes fuente?
message("Inicio Proceso de reconocimiento de rostros en imagenes.", GREEN
if not existsDirectory(sourceImages):
    message("Repositorio de imagenes fuente no existe",RED)
    message("PROCESAMIENTO TERMINADO",RED)
    sys.exit()

#Existe directorio contenedore de rostros?
if not existsDirectory(facesRepo):
    os.makedirs(facesRepo)
    message("Se crea carpeta contenedora de rostros.", GREEN)

#Inicio delproceso
faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_
i = 0

for imageName in os.listdir(sourceImages):
    message("Procesando archivo "+imageName, GREEN)
    if not isImage(sourceImages + "/" + imageName):
        message("Archivo "+imageName+ " no es imagen",RED)
    else:
        #Carga la imagen a memoria
        image = cv2.imread(sourceImages + "/" + imageName)

        #Visualizar imagen a procesar
        imgRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        plt.imshow(imgRGB)
        plt.show()

        faces = faceClassif.detectMultiScale(image, 1.1, 5)
        for (x, y, w, h) in faces:
            cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
            face = image[y:y + h, x:x + w]
            face = cv2.resize(face, (150, 150))
            cv2.imwrite(facesRepo + "/" + str(i) + ".jpg", face)
            i += 1
            cv2.imshow("Rostros", face)
            cv2.waitKey(0)
        cv2.imshow("Image", image)
        cv2.waitKey(0)
    cv2.destroyAllWindows()
```

```
Inicio Proceso de reconocimiento de rostros en imagenes.
Procesando archivo foto3.jpg
```

```
Procesando archivo proyectoErick.docx
Archivo proyectoErick.docx no es imagen
Procesando archivo 238.jpg
```



```
Procesando archivo foto.png
```



```
Procesando archivo WA0007.jpg
```

Paso 2.- Reconocer Rostros en tiempo real

In [ ]:
```
#Guardar embeding nombres y registros asistencia
facesEncodings = []
facesNames = []
attendance = []
```

In [ ]:
```
#Bucle para obtener los codigos de codificacion facial de los rostros alm
for face in os.listdir(facesRepo):
    image = cv2.imread(facesRepo + "/" + face)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    f_coding = face_recognition.face_encodings(image, known_face_location
    facesEncodings.append(f_coding)
    facesNames.append(face.split(".")[0])
message("Embeding de los rostros", GREEN)
print(facesEncodings)
message("Nombres de rostros a buscar", GREEN)
print(facesNames)

#Lectura del video
##stream = cv2.VideoCapture(0, cv2.CAP_DSHOW) backen para windows
stream = cv2.VideoCapture(0,cv2.CAP_V4L2)  #backend para captura de video

#Modelo para la detección Facial
faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_

#Bucle de captura del video hasta presionar escape
while True:
    ret, frame = stream.read()

    if ret == False:   #Evaluar la lectura del frame desde el video
        message("Error al leer el video",RED)
        print(ret)
        break

    #Voltea el frame horizontalmente para efecto espejo
    frame = cv2.flip(frame, 1)
    orig = frame.copy()

    #Búsqueda de rostros en el frame
    faces = faceClassif.detectMultiScale(frame, 1.1, 5)

    #Proceso de recuadro y etiqueta de los rostros
    # Itera sobre cada rostro detectado
    for (x, y, w, h) in faces:
        # Extrae la región de interés (el rostro) del frame original
        face = orig[y:y + h, x:x + w]
        face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)

        #obtener los codigos de codificacion facial de los rostros almace
        actual_face_encoding = face_recognition.face_encodings(face, know

        #Compara y busca similitudes en los rostros
        result = face_recognition.compare_faces(facesEncodings, actual_fa

        if True in result:
            index = result.index(True)
            name = facesNames[index]
            color = (125, 220, 0)
            ##message("Registrar asistencia para "+name, GREEN)
            ###attendanceRegister(name)
        else:
            name = "Desconocido"
            color = (50, 50, 255)
        #recuadro y etiqueta del rostro detectado
        cv2.rectangle(frame, (x, y + h), (x + w, y + h + 30), color, -1)
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, name, (x, y + h + 25), 2, 1, (255, 255, 255),
        ##message("Registrar asistencia para "+name, GREEN)
        attendanceRegister(name)

    cv2.imshow("Frame", frame)

    k = cv2.waitKey(1) & 0xFF
    if k == 27:  #tecla escape
        break
stream.release()
cv2.destroyAllWindows()
printReport()
```

```
Embeding de los rostros
[array([-0.12143265,  0.19625102,  0.08020499, -0.05858349, -0.09441917,
        0.0089132 , -0.03373381, -0.1276917 ,  0.25013059, -0.16357608,
        0.15654157, -0.07490052, -0.27351776,  0.07977211, -0.03840117,
        0.18672575, -0.09343993, -0.14617914, -0.13552435, -0.04133531,
       -0.03567626, -0.00746801,  0.01788495,  0.08826931, -0.02541375,
       -0.40300402, -0.07097854, -0.05955545,  0.00432406, -0.01362286,
       -0.08300193,  0.03576885, -0.16693628, -0.06621212,  0.04101455,
        0.06591708, -0.15637375, -0.06480979,  0.20494595, -0.03600967,
       -0.23680773, -0.05564757,  0.0625725 ,  0.24871862,  0.25732535,
        0.07471912,  0.02216042, -0.16442882,  0.14318113, -0.24764875,
        0.01136388,  0.20363633,  0.0874915 ,  0.09438957,  0.08067404,
       -0.14382546,  0.04135977,  0.10174824, -0.19360857, -0.05150409,
        0.06941391, -0.07890394, -0.06997906, -0.06115335,  0.29649469,
        0.18859772, -0.09985739, -0.13930112,  0.28375718, -0.18836422,
       -0.01020743,  0.0322182 , -0.14070928, -0.18417253, -0.23926395,
       -0.0338513 ,  0.42271015,  0.15667905, -0.0930523 ,  0.03004029,
       -0.11364448,  0.00313011,  0.00934813,  0.09445199, -0.1447169 ,
       -0.08832539, -0.01962383, -0.00296845,  0.21689296, -0.04746201,
       -0.02661706,  0.13452148,  0.02290166, -0.0564867 ,  0.02693324,
        0.04706161, -0.16020441, -0.0470264 , -0.23217943, -0.09671292,
       -0.02013266, -0.0301282 ,  0.00271144,  0.1560344 , -0.2756516 ,
        0.17424661, -0.01707152, -0.03042506,  0.04168707,  0.02629026,
       -0.05590067, -0.07901034,  0.15282919, -0.19916479,  0.16593006,
        0.13413388,  0.0076067 ,  0.14238922,  0.03671142,  0.1015249 ,
        0.00660877, -0.01902001, -0.18318041, -0.11306642,  0.0881359 ,
        0.02402743,  0.03710941,  0.01484852]), array([-9.55536962e-02,
4.06087935e-02,  4.41179462e-02,  2.22713780e-02,
       -1.71126649e-01, -8.37334022e-02, -4.35462110e-02, -9.65142846e-02,
        1.69186473e-01, -1.80669650e-01,  1.80732489e-01, -5.06329387e-02,
       -2.31953323e-01, -5.17832376e-02,  2.77460124e-02,  1.13614768e-01,
       -9.11910757e-02, -1.60435051e-01, -1.24122098e-01, -1.24210022e-01,
        3.72011540e-03, -1.21307261e-02, -2.34917458e-02,  4.30925749e-02,
       -1.05854914e-01, -3.52892667e-01, -5.22318333e-02, -1.25303313e-01,
        3.32733840e-02, -7.71403089e-02, -7.39459917e-02, -1.34352213e-02,
       -1.74719647e-01, -1.60668671e-01,  1.94317270e-02,  1.13869675e-01,
       -1.32946875e-02, -3.92621458e-02,  2.06102937e-01, -5.62832803e-02,
       -2.05779299e-01, -1.41278030e-02,  5.33160008e-02,  2.26676568e-01,
        1.42509416e-01,  7.01146051e-02,  2.02594418e-02, -3.08032539e-02,
        1.55381277e-01, -2.82614678e-01,  8.30982104e-02,  7.33142495e-02,
        1.67735234e-01,  2.69729644e-03,  1.31623402e-01, -1.54071093e-01,
        1.27992770e-02,  9.90563929e-02, -1.96696565e-01,  5.55365756e-02,
        8.54705349e-02, -5.79518601e-02, -7.00965747e-02,  2.72012614e-02,
        2.96806484e-01,  1.60463050e-01, -3.96664403e-02, -9.30511355e-02,
        2.21139476e-01, -1.91101477e-01, -4.75439057e-02,  5.62998690e-02,
       -4.22049686e-02, -1.32073313e-01, -2.30879202e-01,  2.49017589e-02,
        5.11169195e-01,  1.10428005e-01, -2.27381840e-01, -1.51778627e-02,
       -6.88402951e-02, -2.48233788e-02,  1.45836115e-01, -6.03636401e-03,
       -4.93013337e-02, -1.40562980e-02, -1.39976621e-01,  4.10501510e-02,
        2.50262082e-01, -2.14178860e-02, -2.98263207e-02,  1.50488392e-01,
        7.85649419e-02,  2.50926893e-02,  1.11194171e-01,  2.25747824e-02,
       -1.11105174e-01, -4.68956977e-02, -1.73610047e-01, -3.57284136e-02,
        1.62880391e-01, -4.41301912e-02, -1.31389834e-02,  6.08448535e-02,
       -1.68611526e-01,  1.12794548e-01, -3.72063071e-02, -7.12425262e-02,
       -9.07352008e-03,  2.61655636e-02, -1.07230499e-01, -3.26920636e-02,
        1.63706347e-01, -2.06010565e-01,  1.86414972e-01,  1.20067433e-01,
       -4.56652977e-02,  1.61128744e-01,  1.15740988e-02,  9.53902854e-05,
        4.62498628e-02, -5.81197105e-02, -1.73027173e-01, -1.71041444e-01,
        3.66740003e-02, -3.92946303e-02,  9.67395455e-02,  4.48312610e-0
2]), array([-6.53717965e-02,  1.14239633e-01,  6.14230484e-02, -1.83346868
```

```
e-03,
       -2.95813289e-02, -1.11663513e-01, -4.48337831e-02, -1.00183599e-01,
        2.05059662e-01, -7.49980137e-02,  1.41689420e-01, -2.63717473e-02,
       -2.82279968e-01,  2.09885314e-02, -4.23384048e-02,  1.48447528e-01,
       -9.20518786e-02, -2.11920366e-01, -1.04987301e-01, -2.34034210e-02,
       -7.10616936e-04,  1.77728832e-02,  3.19845714e-02,  9.52636153e-02,
       -2.63812635e-02, -3.26228619e-01, -7.98772946e-02, -2.41804253e-02,
        9.27887633e-02, -5.52680530e-02, -6.22116216e-03,  1.00496098e-01,
       -1.98179111e-01, -5.81444949e-02,  1.24747835e-01,  2.47852691e-02,
       -2.45951004e-02, -5.99153452e-02,  2.62874454e-01,  1.14570800e-02,
       -2.26565152e-01, -1.45857139e-02,  5.27062826e-02,  2.80763328e-01,
        2.14198321e-01,  4.14833352e-02,  3.18335220e-02, -1.44548774e-01,
        8.95279795e-02, -2.33007789e-01, -8.29123519e-03,  2.15480253e-01,
        1.13069110e-01,  4.32939306e-02,  6.17704652e-02, -1.68474212e-01,
        5.47522120e-02,  8.55521709e-02, -2.30614573e-01,  1.49007859e-02,
        5.11831343e-02, -1.13108687e-01, -9.40743163e-02, -1.35168806e-02,
        2.11052760e-01,  6.79721609e-02, -1.20183825e-01, -1.59150869e-01,
        2.00733930e-01, -2.29777589e-01, -3.82182794e-03,  9.67581347e-02,
       -6.63777217e-02, -1.09712429e-01, -2.40189210e-01, -8.75332728e-02,
        4.16238606e-01,  7.34227747e-02, -1.66992888e-01,  1.72529057e-01,
       -6.44339547e-02, -1.79731809e-02, -1.35930255e-02,  1.03523172e-01,
       -1.68965355e-01,  8.70784279e-05, -6.46956414e-02, -1.42339924e-02,
        1.94968373e-01,  4.13521491e-02, -5.25152637e-03,  1.47293210e-01,
        2.12878305e-02, -1.29222497e-02,  8.87782499e-02,  3.95374745e-02,
       -8.56462941e-02, -1.26313522e-01, -1.58975095e-01, -1.01628721e-01,
       -3.85919623e-02, -2.82063466e-02, -5.59528079e-03,  9.45705026e-02,
       -2.33162507e-01,  1.39463440e-01, -2.33489810e-03, -3.14698443e-02,
        2.51180362e-02,  6.24106824e-02, -1.39730498e-02, -7.58488029e-02,
        8.07846189e-02, -2.88886458e-01,  1.66084394e-01,  1.19534850e-01,
       -5.94032183e-03,  9.46063399e-02, -1.73125253e-03,  8.21291506e-02,
       -5.77010540e-03, -6.34172652e-03, -9.26209763e-02, -1.40618593e-01,
        2.93034390e-02, -1.80744417e-02,  7.59577379e-02, -2.15879790e-0
2]), array([-2.17088200e-02,  1.29429298e-02,  7.70764938e-03, -1.04776703
e-01,
       -1.48255274e-01, -7.44249523e-02, -6.90179020e-02, -1.39892861e-01,
        1.71111077e-01, -1.51655436e-01,  1.54809356e-01, -9.71860811e-02,
       -1.76650584e-01, -2.13890690e-02, -8.34253728e-02,  1.72767296e-01,
       -2.07153335e-01, -1.24072902e-01, -4.90931123e-02, -7.50879645e-02,
        7.08242133e-02, -4.47421856e-02, -1.31509267e-02,  7.18080997e-02,
       -1.72408029e-01, -2.41655409e-01, -7.11192787e-02, -1.50530979e-01,
       -3.82453874e-02, -8.02880749e-02,  1.45031707e-02,  3.68553214e-03,
       -1.90080538e-01, -2.35231146e-02,  3.84742091e-03,  3.09650544e-02,
       -1.15103200e-02, -6.89964667e-02,  1.03152439e-01,  4.72729281e-03,
       -2.80158222e-01, -1.09497327e-02,  5.14366999e-02,  1.70641422e-01,
        1.92148730e-01,  3.78143825e-02, -2.92988811e-02, -9.66501534e-02,
        1.74612880e-01, -1.83876112e-01, -4.64502908e-02,  9.84592214e-02,
        4.75991257e-02,  2.97236759e-02,  1.31985154e-02, -1.00301258e-01,
        7.04521313e-02,  1.23101175e-01, -2.03982741e-01,  3.97678465e-02,
        1.22220106e-01, -1.40164837e-01, -4.79181036e-02, -8.39961246e-02,
        2.13514343e-01,  5.62651381e-02, -1.66312620e-01, -1.62168980e-01,
        9.05229822e-02, -2.14391887e-01, -6.84647858e-02,  8.79901797e-02,
       -9.44527909e-02, -1.78191379e-01, -3.13630700e-01, -6.69833869e-02,
        3.73632252e-01,  1.16421148e-01, -1.40428901e-01, -3.37653235e-03,
       -7.01155066e-02,  2.35233847e-02,  7.26752058e-02,  1.65376782e-01,
        6.24430664e-02,  3.11809294e-02, -7.37623572e-02, -4.71496098e-02,
        1.86266378e-01, -1.12579882e-01,  2.21040174e-02,  2.28817418e-01,
       -8.56278613e-02,  2.69936640e-02, -8.57367218e-02,  4.77332547e-02,
       -1.99459583e-01,  3.63188982e-02, -1.60113554e-02, -1.83848962e-02,
        1.34166405e-02, -1.41829355e-02,  8.68246643e-05,  6.22202940e-02,
       -1.47614121e-01,  1.50707051e-01, -6.51269108e-02,  1.55478753e-02,
       -3.40062380e-02, -3.63181233e-02, -4.88613136e-02, -6.16940968e-02,
        1.47009090e-01, -2.24411219e-01,  1.70494586e-01,  2.02551141e-01,
        8.07212070e-02,  8.49697068e-02,  5.50645553e-02,  1.51591320e-02,
       -5.11594154e-02, -3.67256720e-03, -2.11852193e-01, -4.63667884e-02,
       -1.32769914e-04, -1.20217122e-01,  5.87354191e-02, -4.65129502e-0
2]), array([-0.11690348,  0.09187673, -0.04169759, -0.10796894, -0.1120897
9,
       -0.04063752, -0.01202122, -0.06610031,  0.27239528, -0.19050099,
        0.22541459, -0.02272561, -0.20317207, -0.0030212 , -0.07033441,
        0.16078614, -0.18467796, -0.12819096, -0.02171786, -0.01327374,
        0.06697481, -0.01480292,  0.01899798,  0.09609997, -0.15150948,
       -0.34002733, -0.08512358, -0.07665589,  0.01385281, -0.04574398,
        0.05184037,  0.02764285, -0.22861914, -0.04885298, -0.01229658,
        0.04957796, -0.05510253, -0.07189362,  0.22749464, -0.00418171,
       -0.2461912 , -0.08247934,  0.05880195,  0.24969673,  0.10804304,
        0.06758489,  0.01408855, -0.11971184,  0.12371513, -0.19783893,
        0.04098572,  0.11023659,  0.03312082,  0.05670823,  0.0203638 ,
       -0.11920778,  0.09455434,  0.15318082, -0.19079542, -0.00134257,
        0.03227879, -0.04737316,  0.00167519, -0.00130132,  0.32470316,
        0.09251774, -0.0878139 , -0.11856095,  0.25146112, -0.15938906,
       -0.05632911,  0.07445189, -0.0859834 , -0.20735925, -0.25887653,
       -0.00683991,  0.39974129,  0.14636195, -0.05798582,  0.08771606,
       -0.05832242, -0.04660477,  0.10317848,  0.17171042, -0.03434834,
       -0.02601156, -0.05534097,  0.04360592,  0.25989655, -0.02341759,
       -0.04614941,  0.22155353,  0.02016821,  0.00574607, -0.00317901,
        0.01505867, -0.12315997,  0.05828607, -0.09495041, -0.0025768 ,
        0.09806458,  0.02536745, -0.01408046,  0.16091423, -0.20437561,
        0.11905266, -0.00756987, -0.02977391,  0.04803633,  0.05694926,
       -0.0606285 , -0.1174508 ,  0.06997258, -0.22684662,  0.2280179 ,
        0.12160999,  0.03808291,  0.16755332,  0.06604396,  0.05382774,
        0.06705527, -0.04881477, -0.23283121, -0.05172123,  0.09082715,
       -0.0565662 ,  0.11665539,  0.01161861]), array([-0.05426005,  0.106
78256,  0.02063346, -0.11917212, -0.13256668,
        0.05831425, -0.10344229, -0.08446901,  0.14126045, -0.07442055,
        0.21396554, -0.09731326, -0.25660917, -0.04938764,  0.01173724,
        0.26187333, -0.20819299, -0.14985217, -0.06309751,  0.00839847,
       -0.01288243,  0.1199915 ,  0.02450119,  0.05498807, -0.14071327,
       -0.4251571 , -0.09662339, -0.05765447, -0.02798934, -0.04221995,
       -0.04336207,  0.14512515, -0.13083078, -0.02932988,  0.13802421,
        0.10451256, -0.06872983, -0.15445282,  0.17734422,  0.07966038,
       -0.22710854, -0.09749389,  0.17911237,  0.22447169,  0.1266486 ,
        0.06194288,  0.05368143, -0.02835995,  0.06238182, -0.25992548,
        0.03001953,  0.13492873,  0.11526047,  0.12237163,  0.0306108 ,
       -0.15164872,  0.06667896,  0.1564161 , -0.25659722, -0.00731734,
        0.06353814, -0.08546682, -0.01284989, -0.06001982,  0.29302663,
        0.13761312, -0.13451812, -0.16307452,  0.16460884, -0.10432013,
       -0.1213999 ,  0.06634714, -0.12653777, -0.20408328, -0.27868965,
       -0.05823856,  0.32714957,  0.15005343, -0.16700241, -0.02337635,
       -0.01344153, -0.08424065, -0.0061795 ,  0.13569261, -0.01067101,
       -0.02844547, -0.0629623 , -0.04299993,  0.26871663, -0.04163023,
        0.00509709,  0.28210622,  0.038203  ,  0.01854514, -0.00321506,
        0.06040353, -0.06465635, -0.01341461, -0.09516247, -0.04112712,
       -0.03168505, -0.02108265, -0.03126177,  0.14877811, -0.20460223,
        0.18701294,  0.00182167, -0.04676448, -0.00655039, -0.00764081,
       -0.05191151, -0.09643606,  0.15202107, -0.27060387,  0.07340267,
        0.20131594,  0.06580024,  0.1583727 ,  0.04683874,  0.09778719,
        0.01631355, -0.10464253, -0.21429202, -0.12679215,  0.09549496,
       -0.06251249, -0.00524055,  0.05454562]), array([-1.95111856e-01,
2.65742503e-02, -3.46516352e-03, -6.53069466e-02,
       -2.23662645e-01, -8.37138519e-02,  2.33088993e-02, -1.42362058e-01,
        1.16009578e-01, -1.48002774e-01,  1.40665501e-01, -8.97523537e-02,
       -2.38441840e-01,  2.44151130e-02,  4.96030077e-02,  2.06192330e-01,
       -1.86837882e-01, -1.16576761e-01, -1.10451199e-01, -6.24286793e-02,
        3.94062661e-02,  6.62324652e-02, -5.16172498e-02,  8.61309171e-02,
       -7.12114871e-02, -3.20227981e-01, -9.21289399e-02, -7.94217512e-02,
       -4.72819060e-02,  5.10259019e-03,  1.31434379e-02,  1.08846419e-01,
       -1.63278431e-01, -1.00804321e-01,  9.66487899e-02,  1.98882446e-02,
       -6.55031279e-02, -1.02815129e-01,  1.88713059e-01, -2.43066195e-02,
       -2.63752908e-01, -1.26173019e-01,  1.49498641e-01,  2.05995157e-01,
        1.02507658e-01,  2.56131570e-05, -1.61733513e-03, -3.21802571e-02,
        1.43017560e-01, -2.66798198e-01, -4.47918437e-02,  1.06296644e-01,
        8.63021314e-02,  2.80003529e-02,  9.27529484e-02, -1.35005385e-01,
        6.12340532e-02,  1.75000846e-01, -2.52709895e-01,  1.00555532e-02,
        9.90729332e-02, -1.18502907e-01, -5.93015887e-02, -6.91952631e-02,
        2.46662244e-01,  1.80947378e-01, -7.90378153e-02, -1.46271497e-01,
        2.10788026e-01, -1.28517956e-01, -1.17099769e-01,  2.28601657e-02,
       -1.32464498e-01, -2.26204976e-01, -2.65679985e-01, -4.12115753e-02,
        3.74529928e-01,  1.32292241e-01, -1.41472086e-01, -2.85079964e-02,
       -1.40227913e-03, -7.00802123e-03,  1.15140878e-01,  8.10081214e-02,
       -5.17362803e-02, -2.58786567e-02, -3.83528210e-02,  2.00473573e-02,
        2.18961403e-01, -3.15147527e-02, -1.20619116e-02,  2.35208139e-01,
        3.01844850e-02, -8.16656351e-02,  4.98923063e-02,  3.40430066e-02,
       -7.59461746e-02, -5.99592179e-02, -1.67274728e-01, -3.63009423e-02,
        1.04113668e-01, -7.74216950e-02,  2.86470540e-02,  1.80369660e-01,
       -1.86254904e-01,  1.04270473e-01, -6.52914196e-02, -3.33036184e-02,
       -8.39748979e-02, -4.39615920e-02, -9.54784155e-02, -2.06624321e-03,
        8.98485482e-02, -1.35184035e-01,  1.65064633e-01,  1.36515006e-01,
       -8.11323300e-02,  1.31614357e-01, -7.85576273e-03,  8.18297267e-03,
       -1.72940195e-02, -6.11211918e-02, -1.94894269e-01, -1.36000901e-01,
        1.85372621e-01, -5.84506663e-03,  4.36126441e-02, -1.74183361e-0
2]), array([-0.09914504,  0.14152785,  0.06872886, -0.0578348 , -0.1247478
1,
        0.04676101,  0.00254835, -0.0530166 ,  0.13702284, -0.06640923,
        0.16463536, -0.01062522, -0.22839938, -0.11308219,  0.09066703,
        0.14200798, -0.12725231, -0.19701949, -0.19973543, -0.03895202,
       -0.0327974 ,  0.08813433,  0.00112843,  0.03992466, -0.18664148,
       -0.35271701,  0.00121303, -0.18493433,  0.01039234, -0.10157056,
       -0.06812304,  0.01561447, -0.12431837, -0.06215981,  0.05609175,
        0.08260363, -0.10860679, -0.08188166,  0.23893674,  0.01693941,
       -0.26284635, -0.00695946,  0.05254797,  0.26495796,  0.27019259,
       -0.01181126,  0.00968751, -0.06694124,  0.1594312 , -0.24913606,
        0.03933183,  0.14557932,  0.08513058,  0.14169271,  0.04402585,
       -0.15907192,  0.07039659,  0.11770327, -0.26028109, -0.01175402,
        0.02485627, -0.2074288 , -0.05277804,  0.00059059,  0.20074677,
        0.06314625, -0.08723499, -0.12134474,  0.07126664, -0.18201052,
       -0.04993847,  0.08899198, -0.12769115, -0.16728811, -0.27602762,
        0.04347649,  0.36437011,  0.17607521, -0.23711395, -0.03321085,
       -0.02566593, -0.05632351,  0.07799637,  0.13193642, -0.09183672,
       -0.03270352, -0.08726429,  0.04112474,  0.20320483, -0.0508736 ,
        0.03666805,  0.2226439 ,  0.06199019, -0.0022101 ,  0.06438543,
        0.01939118, -0.12082317, -0.02942417, -0.1042324 , -0.04889057,
        0.07248502, -0.06283741,  0.03377333,  0.07924648, -0.19984125,
        0.1939868 ,  0.00130636, -0.04606599, -0.01707852, -0.12825425,
       -0.00413011, -0.05369716,  0.19062576, -0.15143567,  0.14657129,
        0.19894019,  0.03796221,  0.17011052,  0.08667197,  0.01216671,
       -0.06461643, -0.00500946, -0.18748239, -0.0168051 ,  0.13981082,
        0.0049276 ,  0.11953988, -0.00365005]), array([-1.07144892e-01,
5.49039580e-02, -3.71862086e-03, -7.28387386e-02,
       -2.01776445e-01, -4.18144725e-02, -1.78558845e-02, -1.66385949e-01,
        2.01527610e-01, -1.77551359e-01,  2.01475680e-01, -4.97519597e-02,
       -1.91396713e-01,  5.59937842e-02, -6.57159612e-02,  1.87097847e-01,
       -8.74120370e-02, -1.79424912e-01, -1.00069463e-01, -4.31390926e-02,
       -1.34309866e-02, -2.20822245e-02, -7.59432614e-02,  1.04118608e-01,
       -1.16850816e-01, -3.40071142e-01, -1.34287268e-01, -1.48396850e-01,
        2.74298135e-02, -8.79502203e-03, -3.46142054e-02,  2.17224564e-02,
       -1.33903712e-01, -5.76818101e-02,  2.78396271e-02,  2.45228838e-02,
       -5.16085699e-02, -7.74762183e-02,  1.82704642e-01,  3.92617062e-02,
       -2.76663870e-01,  7.30204880e-02,  4.42271531e-02,  2.32417762e-01,
        1.27022177e-01,  3.45661491e-02,  6.84864968e-02, -1.14365093e-01,
        1.66477382e-01, -1.95896402e-01,  5.79251302e-03,  9.47938859e-02,
        8.36722776e-02,  4.56349328e-02,  4.78007048e-02, -1.50591865e-01,
        5.60257360e-02,  8.27151313e-02, -2.81170249e-01,  8.74569044e-02,
        7.73463473e-02, -2.11579390e-02, -1.32768909e-02, -3.54081839e-02,
        2.61911154e-01,  1.63085684e-01, -1.25788391e-01, -1.21517375e-01,
        1.97110817e-01, -2.37211943e-01, -3.52628827e-02,  2.11000200e-02,
       -1.36500493e-01, -2.11942792e-01, -2.59727716e-01,  3.00021302e-02,
        5.15534878e-01,  1.78798750e-01, -1.46740466e-01,  1.70392748e-02,
       -5.43950684e-02, -6.10036254e-02,  8.25695992e-02,  9.99650583e-02,
       -3.73376124e-02, -3.72419134e-02, -6.74724355e-02, -3.17322426e-02,
        2.91965902e-01, -2.57471912e-02, -3.74013139e-03,  2.62179673e-01,
       -8.09216872e-05,  3.94490398e-02,  7.08360784e-03,  9.74052176e-02,
       -1.88167930e-01, -6.56552762e-02, -7.81633556e-02, -3.77691463e-02,
       -2.34639961e-02, -3.24697122e-02,  5.18429233e-03,  8.36328119e-02,
       -1.66924879e-01,  1.89943135e-01, -5.55180348e-02,  4.55125561e-03,
       -9.49651971e-02,  8.50982219e-02, -9.54139829e-02, -5.30844070e-02,
        1.59199253e-01, -1.37922481e-01,  2.03241050e-01,  1.41562179e-01,
        7.15027526e-02,  1.01837650e-01,  1.00362852e-01, -4.46636975e-02,
       -5.09131793e-03, -2.16830410e-02, -1.77768022e-01, -1.04026020e-01,
        5.85329533e-02, -1.12571582e-01,  8.93083811e-02,  3.63789424e-0
2])]
Nombres de rostros a buscar
['Irene', 'Wilfrido', 'Iveth', 'Mayte', 'Micaela', 'Briana', 'Anahi', 'Adr
ian', 'Leandro']

Registros de asistencia:
Nombre del Estudiante    Hora de Registro
-----------------------  -------------------
Irene                    2024-06-23 09:32:23
Wilfrido                 2024-06-23 09:32:27
Adrian                   2024-06-23 09:33:49
```