Diseño e implementación de un sistema de reconocimiento facial como medida de identificación para los sistemas de control de acceso del Metro de Medellín.

Daniel Eduardo Salcedo Yeneris

Ingeniero electrónico

Tutor
Jaime Alberto Vergara Tejada, Ingeniero de telecomunicaciones

Universidad de Antioquia
Facultad de ingeniería
Ingeniería Electrónica
Medellín, Antioquia, Colombia
2023

Tabla de contenido

LISTA DE FIGURAS

Siglas, acrónimos y abreviaturas

SCA: Sistema(s) de control de acceso(s)

ML: Machine Learning

CV: Computer Vision

RSA: Algoritmo de encriptación asimétrico de Rivest-Shamir-Adleman

VM: Virtual Machine

1. Resumen

La seguridad es una de las necesidades básicas de toda empresa, pues para garantizar un buen servicio están obligadas entre otras cosas a proteger los activos, salvaguardar la información, dar continuidad a sus servicios, dar protección a clientes o empleados y entre otros deberes cumplir con las normas de seguridad establecidas por la ley, es aquí donde entran los sistemas de control de acceso (**SCA**) los cuales no son más que un conjunto de dispositivos y software diseñados para regular y gestionar el acceso a un determinado espacio físico, área o recurso, permitiendo o denegando la entrada a personas autorizadas o restringiendo el acceso a personas no autorizadas. Bajo esta premisa, en este proyecto se desarrollará e implementará una aplicación de reconocimiento facial que por medio del uso del lenguaje de programación Python, sus librerías visión por computadora (**CV**), comunicación por sockets, criptografía y machine learning (**ML**) actúe a modo de prototipo para una posible integración de la biometría en los SCA del metro de Medellín.

2. Abstract

Security is one of the basic needs of every company, as they are obligated to protect assets, safeguard information, ensure service continuity, provide protection to clients or employees, and comply with security regulations established by the law, among other responsibilities, in order to guarantee good service. This is where Access Control Systems (ACS) come into play, which are nothing more than a set of devices and software designed to regulate and manage access to a specific physical space, area, or resource, allowing or denying entry to authorized individuals and restricting access to unauthorized individuals. Based on this premise, this project will develop and implement a facial recognition application that, through the use of the Python programming language, computer vision (CV) libraries, socket communication, cryptography, and machine learning (ML), will act as a prototype for the potential integration of biometrics into the ACS of the Medellin Metro.

3. Introducción

Llevar control y registro de las personas que acceden a las instalaciones de una empresa es una de las necesidades básicas de seguridad debido a que al prestar servicios a un gran número de personas es fundamental contar con mecanismos que protejan la operatividad de estos, y un paso fundamental para garantizar la continuidad de los servicios y la protección de los recursos propios de la empresa es la supervisión de los individuos que pretenden acceder a la infraestructura de la empresa. Es por esta necesidad imperante en cada negocio que los avances tecnológicos en cuanto a la identificación de individuos han permitido que las organizaciones opten por incorporar en sus instalaciones mecanismos que permitan reconocer a sus empleados por medio de distintos mecanismos de autentificación.

Para garantizar la correcta identificación de las personas en los **SCA** se han desarrollado soluciones innovadoras que permiten asociar o asignar a cada individuo una característica que lo representara ante una plataforma que gestionara los permisos y llevara registro de los intentos de acceso en aquellas zonas que posea un dispositivo capaz de reconocer esta característica del usuario. Es claro que los **SCA** al requerir mecanismos de autentificación que permitan diferenciar en la medida de lo posible a cada persona de forma inequívoca requieren que la característica que representa al usuario sea única, por lo que el usuario puede tener asociado unas credenciales, un objeto físico como una tarjeta MIFARE, un objeto virtual como un código QR o bien una medida biométrica como la identificación por rostro la cual será el objeto de estudio de este proyecto.

La importancia de las medidas biométricas radica en su capacidad para proporcionar un nivel excepcionalmente alto de precisión y confiabilidad en la identificación de personas. A diferencia de las contraseñas o las tarjetas de acceso, que pueden ser olvidadas, robadas o duplicadas, las medidas biométricas son intrínsecas y únicas para cada individuo. Esto las convierte en una herramienta invaluable en entornos donde se requiere un control de acceso seguro y preciso, como instalaciones gubernamentales, empresas, instituciones financieras y sistemas de seguridad en general. Al utilizar medidas biométricas, se fortalece la protección de la información sensible, se previenen fraudes y se garantiza una identificación más confiable de las personas, lo que resulta esencial en el contexto actual de seguridad y protección de datos.

La elección de la identificación facial sobre otras medidas biométricas se basa en varias ventajas significativas. En primer lugar, la identificación facial ofrece una experiencia de usuario intuitiva y sin contacto, lo que resulta conveniente y cómodo para los usuarios. A diferencia de otras medidas biométricas, no requiere tocar un sensor o colocar un dedo en un escáner, lo que facilita su adopción. Además, la identificación facial es inherentemente única y difícil de falsificar, ya que utiliza características faciales distintivas de cada individuo. Esta singularidad y la complejidad de la estructura facial hacen que la identificación facial sea altamente segura y resistente a fraudes. Otra ventaja clave es su capacidad para adaptarse a diferentes entornos y condiciones, como cambios de iluminación o uso de maquillaje, manteniendo un alto nivel de precisión. Por último, la identificación facial se ha vuelto ampliamente accesible gracias a su integración en dispositivos móviles y otros sistemas, lo que la convierte en una solución práctica y versátil para aplicaciones de autenticación. En general, la identificación facial destaca por su conveniencia, seguridad, adaptabilidad y amplia disponibilidad, lo que la convierte en una opción atractiva sobre otras medidas biométricas.

4. Objetivos

4.1 Objetivo General

Desarrollar e implementar un sistema que permita, mediante procesamiento digital de imágenes, identificar a un usuario a través de su rostro usando medidas y proporciones únicas de cada individuo como método de autentificación en un sistema de control de acceso en una red.

4.2 Objetivos Específicos

- Diseñar el sistema de autenticación priorizando la economía, la simpleza y la seguridad en la protección de los datos.
- Implementar el sistema de autentificación utilizando herramientas de software libre que sirvan para llevar a cabo la identificación, comunicación y la seguridad de la información en el sistema.
- Diseñar y ejecutar pruebas de funcionamiento y rendimiento sobre un entorno de experimentación, con el fin de evaluar diferentes métricas y ajustar la solución para optimizar o corregir su funcionamiento.

5. Metodología

OBJETIVO ESPECIFICO	ACTIVIDADES
Diseñar el sistema de autenticación priorizando la economía, la simpleza y la seguridad en la protección de los datos.	**1.1** Realizar una investigación acerca de las métricas únicas extraíbles de los rostros. **1.2** Indagar sobre los alcances y limitaciones de los mecanismos de reconocimiento faciales en trabajos similares. **1.3** Con base en previas investigaciones y tomando en consideración los requerimientos especificados, diseñar el algoritmo de autentificación. **1.4** Seleccionar una topología de red para el sistema de control de acceso.
Implementar el sistema de autentificación utilizando herramientas de software libre que sirvan para llevar a cabo la identificación, comunicación y la seguridad de la información en el sistema, minimizando de forma simultánea la cantidad de equipos especializados y periféricos a usar.	**2.1** Implementar el algoritmo de reconocimiento mediante un script de Python que use el framework de mediapipe. **2.2** Crear una red de máquinas virtuales donde se definirán los roles de cliente y servidor. **2.3** Implementar la función de comunicación y envío de información. **2.4** Realizar pruebas de funcionamiento.
Diseñar y ejecutar pruebas de funcionamiento y rendimiento sobre un entorno de experimentación, con el fin de evaluar diferentes métricas y ajustar la solución para optimizar o corregir su funcionamiento.	**3.1** Definir las pruebas necesarias para obtener métricas que evalúen el rendimiento del sistema y detectar comportamientos anómalos en el sistema ante diferentes escenarios. **3.2** Efectuar correctivos y optimizaciones según se requiera.

6. Marco Teórico

6.1 Sistemas de control de acceso

Un sistema de control de acceso es una solución tecnológica diseñada para regular y gestionar el acceso a áreas físicas, recursos o información en una organización, permitiendo o denegando el acceso a personas autorizadas y restringiendo el acceso a personas no autorizadas. Este sistema consta típicamente de varios componentes, como lectores de identificación, cerraduras electrónicas para controlar el acceso a puertas o barreras físicas, paneles de control que gestionan la comunicación entre los lectores, las cerraduras y el software del sistema, y el software de gestión que administra y configura parámetros como autorizaciones de acceso, horarios, niveles de seguridad y generación de informes. Además, los sistemas de control de acceso se integran frecuentemente con sistemas de vigilancia, como cámaras de seguridad, para capturar imágenes o videos de eventos de acceso, brindando un mayor nivel de seguridad y monitoreo

6.1.1 Tipos de medición

En los sistemas de control de acceso, existen varios tipos de validación que se utilizan para verificar la identidad de una persona y permitir o denegar su acceso, los métodos más utilizados son [1]

1) **Validación mediante tarjetas o llaves electrónicas:** Este método implica el uso de tarjetas de proximidad, llaves electrónicas u otros dispositivos similares que contienen información de identificación única. Estos dispositivos se presentan o se escanean en un lector de identificación para autenticar al individuo. El sistema compara la información de la tarjeta o llave con los datos almacenados en la base de datos para permitir o denegar el acceso.

2) **Validación biométrica:** La validación biométrica se basa en características físicas o comportamentales únicas de una persona. Algunos métodos biométricos utilizados en los sistemas de control de acceso incluyen:
   a) Reconocimiento facial: El sistema captura la imagen del rostro de la persona y la compara con una base de datos de rostros autorizados.
   b) Reconocimiento de huellas dactilares: Se utilizan sensores para capturar las huellas dactilares y compararlas con las huellas dactilares almacenadas.

c) Reconocimiento de iris: Mediante el uso de cámaras de alta resolución, se captura y compara los patrones únicos del iris del ojo.
d) Reconocimiento de voz: El sistema graba y compara las características únicas del habla de una persona.

3) **Validación de contraseñas o códigos PIN**: Este método implica que los usuarios ingresen una contraseña o un código PIN en un teclado o panel táctil para autenticarse. El sistema compara el valor ingresado con el valor almacenado en la base de datos.

4) **Validación de doble factor o multifactorial:** En este enfoque, se requiere más de un método de autenticación para verificar la identidad de un individuo. Por ejemplo, podría ser necesario presentar una tarjeta de identificación junto con una huella dactilar o proporcionar un código PIN junto con un reconocimiento facial.

Es importante tener en cuenta que la elección del tipo de validación depende de varios factores, como el nivel de seguridad requerido, la conveniencia, el entorno y los recursos disponibles. Cada tipo de validación tiene sus propias ventajas y consideraciones de implementación.

6.1.2 Seguridad y privacidad en los sistemas de control de acceso

La seguridad y la privacidad de los datos son aspectos críticos en los sistemas de control de acceso, ya que estos sistemas manejan información sensible y personal de los usuarios. Los datos personales, como información biométrica o datos de identificación, deben almacenarse de manera segura en las bases de datos de los SCA. Se deben implementar medidas de seguridad adecuadas, como cifrado de datos, para proteger la información contra accesos no autorizados. Es esencial establecer controles de acceso adecuados para garantizar que solo las personas autorizadas puedan acceder y manipular los datos almacenados en los sistemas de control de acceso. Se deben establecer niveles de permisos y privilegios apropiados para limitar el acceso a la información sensible. Cuando los datos se transmiten entre los componentes del sistema de control de acceso, es fundamental utilizar canales seguros de comunicación. Esto implica el uso de protocolos de encriptación y medidas de seguridad para prevenir la interceptación o manipulación de los datos durante la transmisión. Los sistemas de control de acceso deben cumplir con las regulaciones y normativas aplicables en cuanto a la seguridad y privacidad de los datos. Esto puede incluir leyes

de protección de datos y privacidad, como el Reglamento General de Protección de Datos (RGPD) en la Unión Europea, así como otras normativas sectoriales o regionales relevantes. También se deben registrar y auditar los eventos relevantes, como los intentos de acceso, las acciones realizadas por los usuarios autorizados y las violaciones de seguridad. Esto permite la detección de incidentes y ayuda en la investigación forense en caso de violaciones de seguridad [2].

6.2 Visión por computadora

La visión por computadora es una disciplina interdisciplinaria que combina la inteligencia artificial, la ciencia de la computación y la percepción visual. Se centra en desarrollar algoritmos y sistemas que permiten a las máquinas procesar, analizar e interpretar imágenes o videos de la misma manera que los seres humanos lo hacen. El objetivo principal de la visión por computadora es capacitar a las máquinas para comprender y extraer información útil de los datos visuales, lo que incluye tareas como el reconocimiento de objetos, el seguimiento de movimiento, la detección de rostros, la segmentación de imágenes y mucho más. La visión por computadora utiliza una variedad de técnicas y enfoques, como el procesamiento de imágenes, el aprendizaje automático, la geometría computacional y la estadística, para analizar y comprender la estructura, el contenido y el contexto de las imágenes o videos. Estas técnicas permiten a las máquinas identificar patrones, características relevantes y relaciones espaciales en los datos visuales, lo que a su vez permite la toma de decisiones, la clasificación y la interpretación automatizada de la información visual [3].

6.2.1 Procesamiento de imágenes

El procesamiento digital de imágenes es un conjunto de técnicas que se centran en el análisis y la manipulación de imágenes utilizando técnicas y algoritmos computacionales. El objetivo principal del procesamiento digital de imágenes es mejorar, modificar y extraer información útil de las imágenes capturadas por dispositivos electrónicos, como cámaras digitales o escáneres. El proceso de procesamiento digital de imágenes involucra una serie de etapas en las cuales podemos encontrar la *adquisición de imágenes* mediante los dispositivos de adquisición. Posteriormente se hace un *preprocesamiento* donde se aplican técnicas de corrección y mejora a las imágenes para eliminar ruido, corregir la exposición, ajustar el contraste, y demás ajustes que puedan mejorar la calidad general de la imagen. La imagen preprocesada es entonces *filtrada* para resaltar o suavizar características específicas de la imagen. El siguiente paso es la *segmentación* donde se realiza la

partición de la imagen en regiones o segmentos con características similares. Esto puede ser útil para identificar objetos de interés como los rostros humanos y separar el fondo del primer plano en una imagen. De la imagen resultante extraen características relevantes de la imagen, como bordes, contornos, texturas o características específicas de objetos. Estas características se utilizan para el reconocimiento de patrones y demás análisis posteriores. Finalmente se utilizan algoritmos de aprendizaje automático y técnicas de reconocimiento de patrones para identificar objetos, rostros, textos u otras características específicas en una imagen [4].

6.2.2 Reconocimiento Facial

El reconocimiento facial hace referencia a una serie de algoritmos que se utiliza para identificar y verificar la identidad de una persona a partir de características faciales únicas y distintivas. Se basa en el análisis y la comparación de patrones y características faciales para determinar si una persona coincide con una imagen de referencia o con los datos almacenados en una base de datos. Como todo buen algoritmo, este tiene etapas bien diferenciadas, siendo la primera la detección facial donde se utiliza un algoritmo para detectar y localizar rostros en una imagen o en un flujo de video. El algoritmo busca características como los ojos, la nariz, la boca y los contornos faciales para identificar posibles regiones faciales en la imagen. Una vez que se ha detectado un rostro, se extraen características distintivas de la imagen facial. Estas características pueden incluir la forma y posición de los ojos, la distancia entre ellos, la forma de la nariz, la boca, entre otros elementos. Las técnicas comunes para la extracción de características incluyen el análisis de puntos clave, el análisis de texturas y el análisis de patrones. A partir de las características extraídas, se crea una representación numérica o un vector que captura las características únicas del rostro. Esta representación se conoce como "plantilla facial" o "descriptores faciales" y se utiliza para realizar comparaciones y buscar similitudes en la etapa que denominaremos de reconocimiento durante la etapa de reconocimiento, se compara la plantilla facial de la persona que se quiere identificar con las plantillas faciales almacenadas en una base de datos. Se utilizan algoritmos de coincidencia para determinar el grado de similitud o distancia entre las características faciales. Si hay una coincidencia cercana o por encima de un umbral establecido, se considera que la persona ha sido reconocida. Según el resultado de la comparación, se toma una decisión sobre la identidad de la persona. Esto puede implicar una coincidencia positiva, una falta de coincidencia o un resultado incierto. Dependiendo del contexto de uso, se pueden tomar acciones específicas, como permitir el

acceso, generar alertas o realizar otras operaciones relacionadas con la identificación de la persona [5]. No obstante es necesario tener en consideración que aunque os algoritmos de reconocimiento facial han mejorado significativamente en los últimos años, todavía presentan ciertas limitaciones y desafíos que es importante tener en cuenta:

1. **Variabilidad en las condiciones de captura:** Los algoritmos de reconocimiento facial pueden ser sensibles a las variaciones en las condiciones de captura, como la iluminación, el ángulo de la cara, la expresión facial y el fondo. Estas variaciones pueden afectar la precisión del reconocimiento.

2. **Cambios en el aspecto:** Los cambios en el aspecto de una persona, como el maquillaje, el peinado, el uso de anteojos o el envejecimiento, pueden dificultar la identificación precisa.

3. **Datos insuficientes:** Para obtener un rendimiento óptimo, los algoritmos de reconocimiento facial requieren un conjunto de datos grande y diverso para el entrenamiento. Si el conjunto de datos es limitado o sesgado, el reconocimiento puede ser menos preciso.

4. **Diferencias étnicas y culturales:** Algunos algoritmos pueden tener dificultades para reconocer rostros de diferentes grupos étnicos o culturales debido a diferencias en la estructura facial.

5. **Privacidad y ética:** El reconocimiento facial plantea preocupaciones sobre la privacidad y el uso no ético de los datos faciales recopilados. La recopilación y el almacenamiento de datos biométricos pueden plantear riesgos para la seguridad y la privacidad de las personas.

6. **Vulnerabilidad a ataques**: Los algoritmos de reconocimiento facial pueden ser vulnerables a ataques de manipulación, como el uso de imágenes falsas o técnicas de camuflaje para engañar al sistema.

7. **Rendimiento en tiempo real:** Algunos algoritmos pueden ser computacionalmente intensivos, lo que puede limitar su rendimiento en tiempo real en sistemas con recursos limitados.

8. **Fallos de identificación y falsas coincidencias:** Aunque los algoritmos han mejorado en la precisión, aún pueden ocurrir errores de identificación o falsas coincidencias, lo que puede tener consecuencias negativas, especialmente en aplicaciones críticas como seguridad y vigilancia.

6.2.3 Mediapipe

En la etapa de detección facial se usará de forma particularmente el framework MediaPipe de código abierto desarrollado por Google que se utiliza para construir aplicaciones de procesamiento de medios en tiempo real. Con su enfoque en el procesamiento de datos multimedia, MediaPipe ofrece una plataforma flexible y eficiente para el desarrollo de aplicaciones de visión por computadora y procesamiento de señales. Una de las características destacadas de MediaPipe es su capacidad para realizar el seguimiento y análisis facial mediante el uso de Face Mesh (malla facial). Face Mesh es un componente clave de MediaPipe que permite detectar y rastrear los puntos clave del rostro en tiempo real, generando una malla 3D que se ajusta a la estructura facial. Esta malla incluye puntos de referencia para los ojos, las cejas, la nariz, los labios y otras áreas faciales. El seguimiento facial con Face Mesh es especialmente útil en aplicaciones de realidad aumentada, animación facial, filtros en tiempo real y muchas otras áreas donde se requiere un reconocimiento y seguimiento preciso de los elementos faciales, en el caso particular de este proyecto será usado para el aislamiento de los rostros en las imágenes capturadas por las cámaras [6].

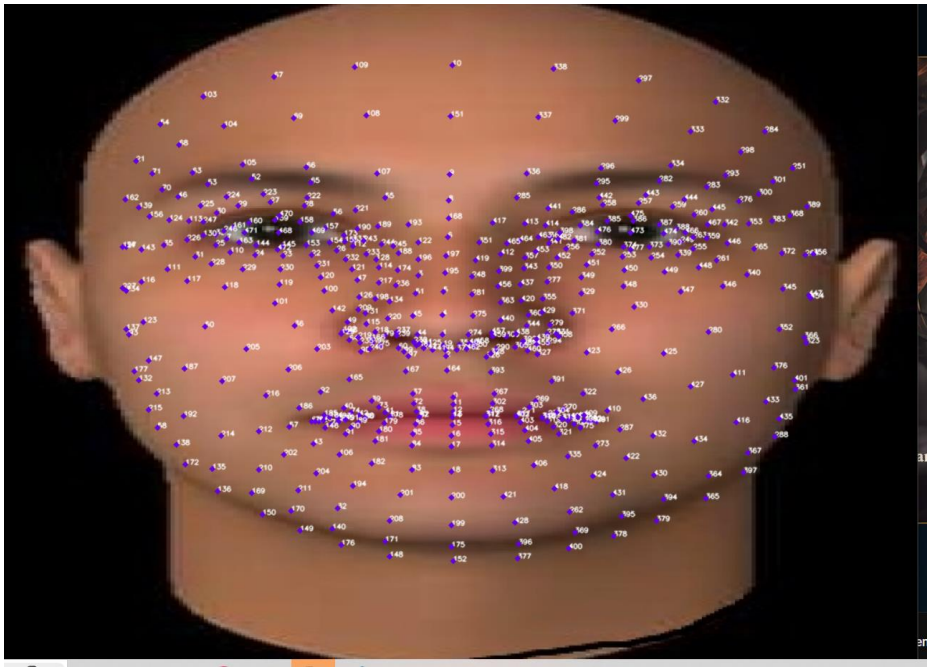

Figura 1: Face mesh de mediapipe

6.3 Comunicación

El proyecto tiene integrado un componente de telecomunicación mediante el uso de unas herramientas conocidas como sockets que no son más que una abstracción que permite la comunicación entre dos programas que se ejecutan en una red. Actúa como un punto final en una conexión de red y se utiliza para enviar y recibir datos entre las aplicaciones que se comunican. Los sockets permiten la comunicación entre programas que se ejecutan en diferentes dispositivos y plataformas, lo que brinda flexibilidad en el intercambio de datos en una red y adicionalmente facilitan la comunicación en tiempo real, lo que es esencial para aplicaciones que requieren una respuesta rápida, como las aplicaciones de mensajería instantánea o los sistemas de transmisión de video en vivo. Sin embargo, también existen algunas desventajas en el uso de sockets debido a que la implementación y gestión de la comunicación entre sockets puede ser compleja, especialmente en aplicaciones que requieren lógica adicional para garantizar una conexión confiable y segura, además requieren un cuidadoso manejo y mantenimiento para garantizar una comunicación eficiente y prevenir problemas como fugas de memoria o conexiones colgadas [7].

7. Implementación

Una vez realizadas las investigaciones expuestas en el marco teórico en donde se mostraron las herramientas y conceptos a utilizar, y de igual forma definidos los alcances del proyecto en los objetivos específicos y general, se expondrá el diseño e implementación del mismo dividiéndolo en 3 partes bien diferenciadas, el diseño del protocolo de comunicación por sockets en el cual se realizara un handshake para compartir las llaves públicas de los hosts con el server y viceversa [8], la etapa de entrenamiento en la cual con base en videos de los sujetos de prueba capturados con las cámaras adquiridas para el trabajo se entrenara el programa por medio de los algoritmos de ML ofrecidos por la librería multiplataforma OpenCV[9][10] donde se obtendrá un archivo de salida producto del entrenamiento que estará disponible para la última de las 3 partes, siendo esta la etapa de reconocimiento facial en donde se usara el archivo generado durante el entrenamiento para estimar la identidad de la persona, cada uno de estos aplicativos serán escritos en Python.

7.1 Comunicación y redes

La comunicación entre cliente y servidor es establecida mediante sockets que se envían mensajes encriptados por el método de cifrado asimétrico RSA. El proceso de generación de claves implica la selección de dos números primos grandes y el cálculo de otros parámetros matemáticos relacionados. La seguridad del cifrado radica en la dificultad de factorizar el producto de estos números primos para obtener la clave privada.

7.1.1 Servidor

Se define un socket que utilice el protocolo TCP/IP para recibir las conexiones de todos los clientes clientes y realizar un intercambio de mensajes encriptados mediante el uso de RSA. Se crea la clase Server y se definen como principales atributos la dirección IP y puerto del servidor, de igual forma se establece el tamaño máximo de cada mensaje. Se crea un método llamado **socketOpen()** que es invocado desde el constructor de la clase y se encarga de abrir el socket y asociarlo a la dirección IP y puerto especificados, además, genera las claves pública y privada utilizando la librería RSA. Luego, el servidor se pone en modo de escucha para aceptar conexiones de clientes entrantes y automáticamente se llama al método **receiveConnections()** el cual ejecuta en un bucle infinito en el que se aceptan conexiones de clientes entrantes. Después de aceptar una conexión, se usa el método **RSAHandshake()** que realiza un intercambio de claves RSA con el cliente, es decir, el

servidor envía su clave pública al cliente y recibe la clave pública del cliente. Este proceso establece una comunicación segura y permite la encriptación y desencriptación de los mensajes. Finalizado el hanshake se inicia un hilo para manejar la comunicación con ese cliente en particular y en paralelo se sigue escuchando y aceptando clientes. El que se encarga propiamente de la comunicación es el método **handlemsg()** que ejecuta bucle infinito dentro de un hilo y se encarga de recibir y manejar los mensajes del cliente. Utiliza los métodos **receive_encrypted_msg()** y **send_encrypted_msg()** para recibir y enviar mensajes encriptados, respectivamente.

7.1.2 Cliente

El código fuente usado por el cliente es el encargado de realizar el reconocimiento facial y enviar mensajes encriptados a un servidor a través de una conexión TCP/IP. Se define una clase llamada application y se crean atributos como la cámara la cual es definida con un numero entero que identifica a cada cámara (en caso de haber más de una), la dirección IP, el puerto para comunicarse con el servidor, la longitud de cada mensaje, el nombre del host (o hostname) y las claves pública y privada RSA. Desde el constructor se llama al método **connectServer()** se encarga de establecer una conexión con el servidor utilizando un socket TCP/IP. Realiza un intento de conexión y, en caso de éxito, realiza un intercambio de claves RSA para establecer una comunicación segura recibiendo primero la clave del servidor y luego enviando la propia. Si no se puede establecer la conexión después de un número máximo de intentos, el programa se cierra. Establecida la conexión y finalizado el handshake, se invoca el método **VideoCapture()** que es el encargado de capturar video de la cámara y realizar el procesamiento de reconocimiento facial por medio de los métodos **validation(), identification(),** y **Blink()** que veremos en secciones posteriores, lo importante aquí es que el resultado de la identificación es enviado al servidor por medio de un mensaje encriptado y este contiene el hostname, el usuario que se identificó y la hora con el intento de acceso, el mensaje se manda cuando la persona que está siendo identificada parpadea.

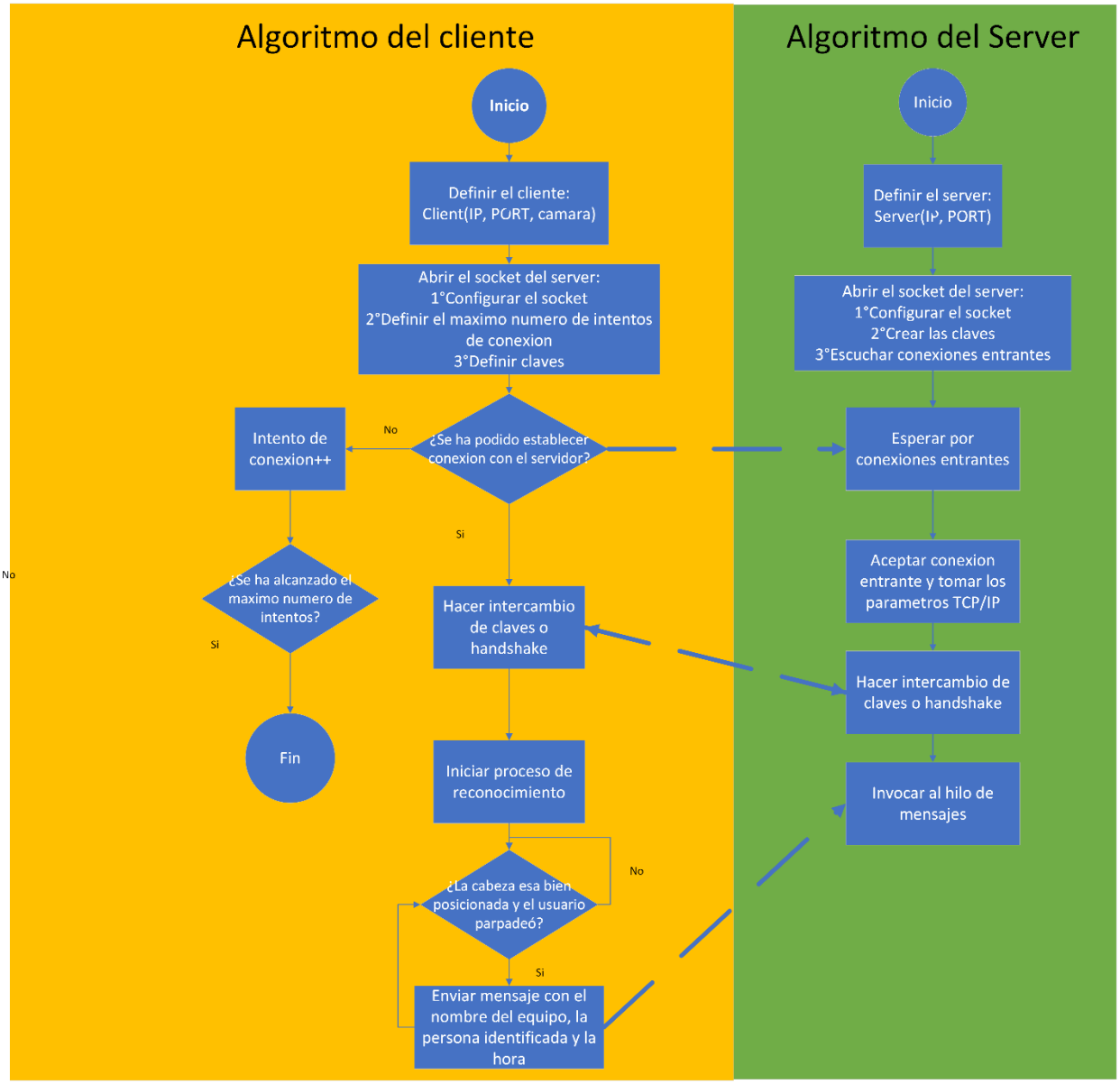

Figura 2: Protocolo de comunicación

7.2 Entrenamiento

El reconocimiento facial al ser un algoritmo de ML requiere de un entrenamiento previo el cual recibirá archivos multimedia de entrada y por medio de una serie de 3 etapas, convertirá la entrada en un archivo XML que contendrá la "Firma Facial" de los sujetos de prueba. Las etapas son descritas a continuación:

- **Generación de videos:** Se captura con las cámaras los videos de los participantes, estableciendo como parámetros el nombre de la persona y la cantidad de frames que se

desea capturar por cada dispositivo, el video de salida será llamado con el nombre del usuario y guardado en formato MP4. Se cuenta con 3 cámaras las cuales se van activando conforme el dispositivo anterior ha capturado los frames solicitados.

- **Generación de datos:** Una vez el video es generado, en el script se ingresa el nombre del sujeto a registrar, de esta forma se declaran las rutas del archivo que contiene el video del usuario en cuestión y la carpeta de datos del sujeto donde se guardaran los datos de salida de esta etapa. Se verifica si la carpeta de datos del sujeto ya existe, si este no es el caso esta es creada y se muestra un mensaje de confirmación, si por el contrario la carpeta de datos ya existe, se pregunta si desea sobrescribir los datos existentes. Definimos una función llamada GeneradorMediapipe() que realizará la generación de muestras utilizando Mediapipe, se solicitara el numero de fotogramas de muestra que se tomarán del video. Se inicializa el proceso de detección de caras y seguimiento utilizando la clase mediante el face mesh de Mediapipe, de esta forma se toma de cada frame el rostro detectado, se dimensiona a una resolución de 640x480px y se convierte en escala de grises.

- **Generación del archivo XML:** Desde la carpeta que actúa como base de datos se seleccionan aquellas subcarpetas que tengan los nombres de usuario, pues estas contienen los rostros de los participantes en escala de grises que son las muestras por usar, Se enlista de nombres de personas asociandoles un numero entero que corresponderá a la etiqueta del usuario (tag) y se almacena el listado en un archivo de texto vacío. Con esta misma lista se inicializan las listas "labels" y "facesData" para almacenar las etiquetas y las imágenes de los rostros respectivamente, todo esto con el objetivo de relacionar todos los rostros de un mismo usuario bajo una misma etiqueta. Se crea una instancia del clasificador de rostros que usa el método de Fisher para el reconocimiento facial que viene incluido en el paquete de OpenCV, este clasificador recibe las listas "labels" y "facesData" y entrega un archivo XML que contiene la información facial de los rostros contenidos en esta última lista.

7.3 Reconocimiento Facial

Como se mencionó en la sección 6.1.2, existen 3 métodos especiales dentro del código del cliente que en su conjunto son los encargados de estimar la identidad de una persona. El identification()

realiza la identificación de la persona en función del reconocimiento facial, para ello extrae la región del rostro, la convierte a escala de grises y la redimensiona todo ello con el objetivo de que la imagen del rostro con la que se hará la estimación sea lo más parecido a los rostros de muestra con los que se entrenó la aplicación. Luego, utiliza un modelo de reconocimiento facial entrenado (face_recognizer) para predecir la identidad de la persona. Con el objetivo de mejorar la identificación del sujeto se creó el validation() con el que se valida la posición y alineación del rostro de forma que el usuario deberá tener el rostro centrado en la cámara, se da una ayuda visual mediante una elipse que cambia de color dependiendo de si la posición y orientación del rostro es correcta. Una vez el rostro está centrado, la función Blink() realiza un seguimiento del parpadeo de los ojos calculando la relación de parpadeo [11] (blinkratio) a través de los puntos clave detectados en el ojo. Si la relación de parpadeo es inferior a un umbral indicara que los ojos están cerrados y comparando este valor con el blinkratio de anteriores fotogramas se determina si ha habido un parpadeo. Todos los métodos mencionados en esta sección utilizan la API face mesh de la biblioteca Mediapipe para estimar la posición de los puntos del rostro en cada fotograma y con estos se toman la posición de los puntos de interés para medir las distancias, proporciones y posiciones claves del rostro.

8. Resultados

Siguiendo la metodología del trabajo por etapas, se ilustrará en un inicio los resultados de las fases del aplicativo por separado con el objetivo de verificar y evaluar la funcionalidad de cada uno de estos y se hará un análisis de las posibles causas de error.

8.1 Comunicaciones

Para simular los protocolos de comunicación se decidió crear 2 entornos diferentes, siendo el primero una red con enlaces redundantes creada en el software de simulación GNS3 el cual entre muchas funciones permite emular equipos de redes (hosts en forma de máquinas virtuales, switches, routers, etc) mediante la imagen de su sistema operativo [12].

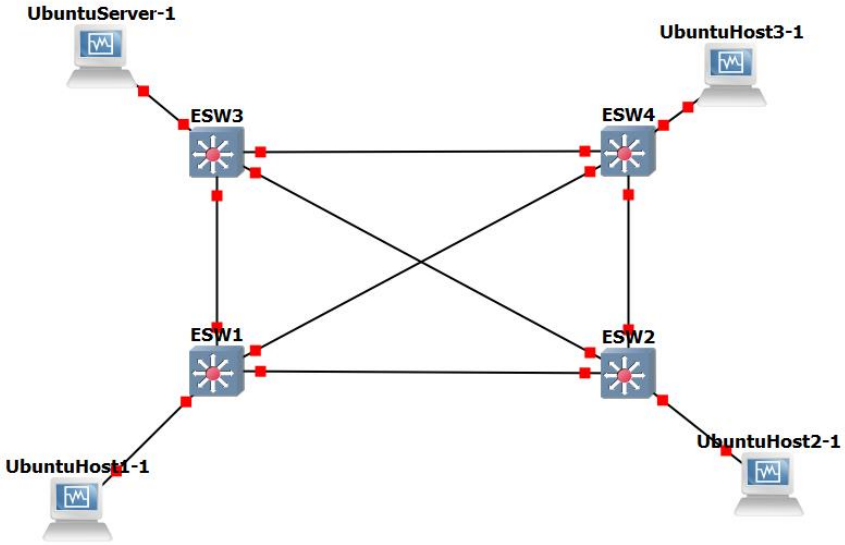

Figura 3: Red de pruebas en GNS3

El segundo entorno consiste en una red de area local domestica mediante el uso de un Macbook y un PC Acer que establecer comunicación por medio de un modem Arris que actúa como router.

En un principio se ilustrarán los resultados en el entorno simulado y se dejara una ilustración meramente demostrativa del entorno real. Se inicia mostrando la configuración de los adaptadores de red para demostrar que están en un mismo segmento de red y de igual forma se ilustra el hostname para identificar fácilmente a cada uno.

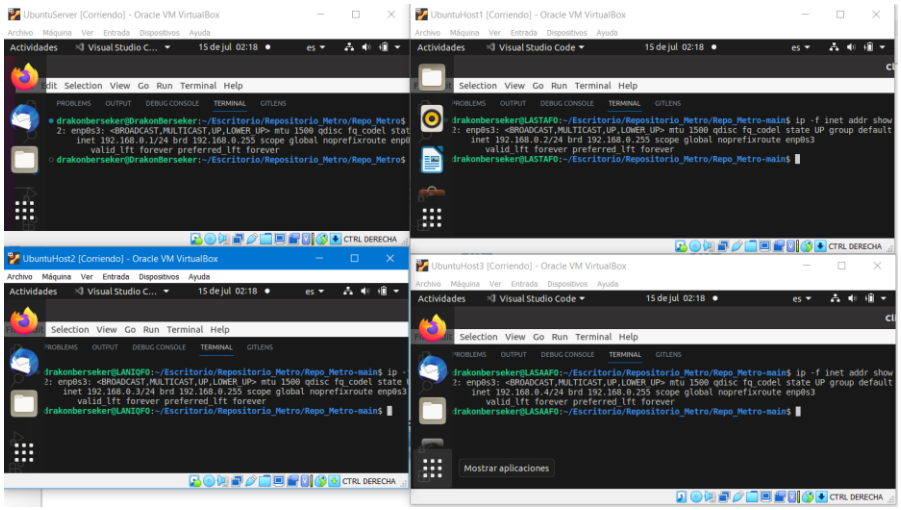

Figura 4: Configuración de los adaptadores de las VM

Se observa que el servidor y los hosts tienen direcciones IP tipo C en el rango [192.168.0.1 – 192.168.0.4] estando la primera dirección reservada para el servidor y con mascara /24[13]. A continuación, establecemos las conexiones con el servidor y enviamos mensajes desde cada cliente al servidor, este mensaje emula lo que enviara el aplicativo completo lo cual no es más una cadena te texto con el nombre del host, un mensaje y la hora.

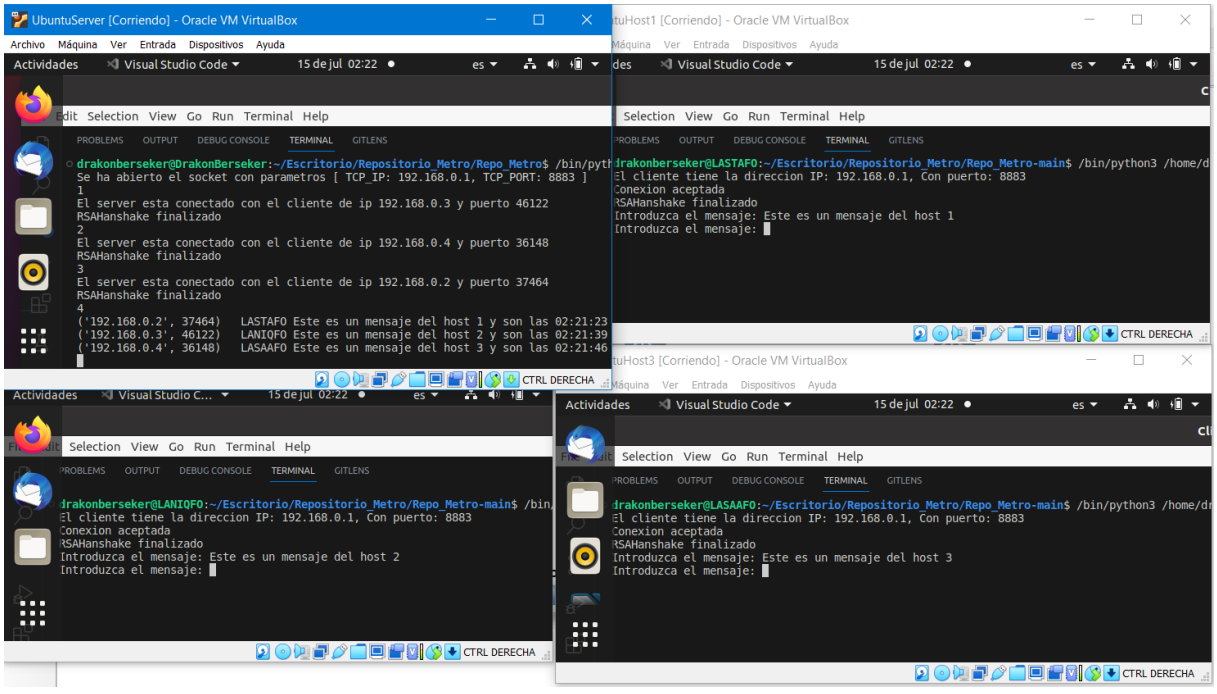

Figura 5: Programa del servidor manejando mensajes de múltiples clientes

En cuanto a la seguridad en la transmisión, se usó Wireshark, un software de código abierto utilizado para el análisis y la captura de paquetes en una red de computadoras en tiempo real, permitiendo examinar los datos que se transmiten a través de una red y proporciona una vista detallada de los paquetes de red, incluyendo sus encabezados, protocolos utilizados, direcciones IP, puertos, tiempos de transmisión, entre otros detalles. En la imagen inferior se observa en la captura de la derecha que cuando en mensaje no es encriptado mediante RSA, si el paquete es capturado durante la transmisión la información queda a disposición del atacante pues esta no está cifrada,

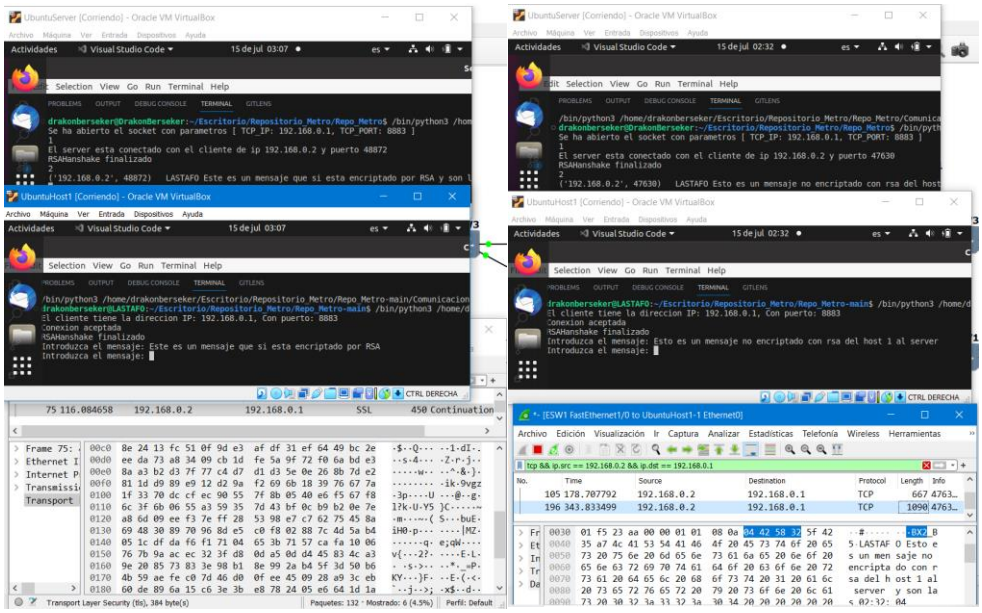

Figura 6: Comparación de paquete encriptado contra no encriptado

Otro aspecto por resaltar es que en la encriptación es necesario definir el tamaño de la clave con un numero de bits y esto obliga a que el tamaño del mensaje a encriptar no sea superior al tamaño de la clave. Así las cosas, el tamaño de la clave definida es de 3072 bits y dado que 8 bits forman un byte, esto prohíbe mensajes con más de 384 bytes. A efectos prácticos esto es una limitación por el uso de la encriptación, pero no representa un problema para el aplicativo debido a que los mensajes que se transmitirán no son tan extensos. Se deja adjunta la imagen que ilustra el programa corriendo en 2 dispositivos reales.

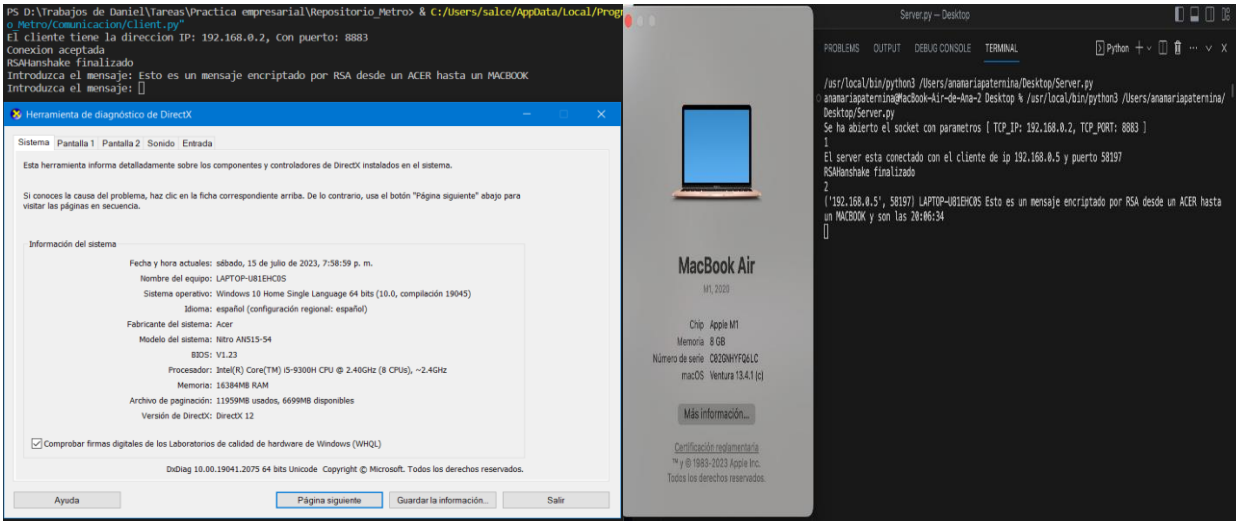

Figura 7: Comunicación en red domestica

8.2 Entrenamiento

En lo referente a la etapa de entrenamiento observamos que el video generado sufre de cambios en la calidad y contraste al cambiar de cámara debido a que se usan 2 modelos diferentes siendo una la cámara integrada del PC usado en el proyecto y la otra una cámara externa genérica, ello produce un cambio de luminosidad perceptible a simple vista que ocasiona ruido al momento de generar las imágenes que actuaran como base del entrenamiento. Pese a que la posición del rostro en cada transición cambia debido a la posición del nuevo dispositivo, es necesario aclarar que el ángulo de cada uno de cada uno respecto al emisor de luz no variaba lo suficiente para producir cambios bruscos de contraste debido a la perspectiva.

Figura 8: Captura de la cámara y rostros en escala de grises

Este cambio es más notable visto desde los rostros en escala de grises donde se aprecia que dependiendo de la cámara que se use algunas muestras son más oscuras que otras. Esta situación se hace más evidente al comparar los histogramas de 2 muestras con una orientación de la cabeza similar pero capturada por cámaras diferentes.

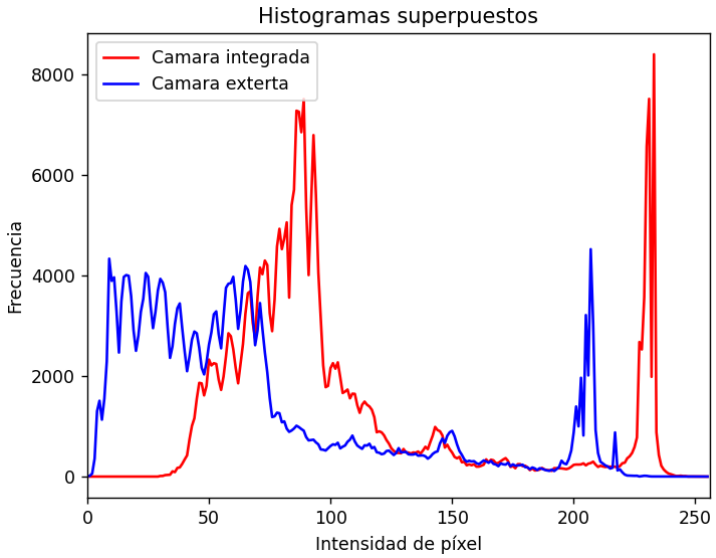

Figura 9: Comparación de histogramas de rostros ante cada modelo de cámara

Aun con esta dependencia de las muestras con el dispositivo de entrada se procedió a generar el archivo XML con estas muestras usando el método de Fisher, para ello se cargaron 890 imágenes de resolución 640x480px por cada uno de los 3 participantes, por lo que se tiene que el vector "facesData" mencionado en la sección 7.2 contiene 2670 imágenes, generando así un archivo de salida de 24MB. Con el objetivo de establecer métricas en el entrenamiento se asumirá que el espacio total ocupado por los folders con las imágenes de muestra es de 120MB, y además que el archivo XML contiene la misma cantidad de información por persona, bajo esta premisa se tiene que cada usuario está ocupando 8MB y que el entrenamiento optimiza el espacio ocupado por las imágenes al reducir los 120MB a solo 24MB, es decir un 20% del tamaño total de los datos de entrada.

8.3 Reconocimiento

En cuanto al reconocimiento en tiempo real se contó con 3 sujetos de prueba siendo uno de género masculino y 2 de género femenino

Figura 10: Sujetos de prueba para el reconocimiento

Donde en la imagen anterior observamos de izquierda a derecha a Luisa, Ana María y Daniel, con estos sujetos se entrenó la aplicación y se procedió a realizar pruebas sobre el aplicativo cambiando la iluminación de fondo y agregando accesorios al rostro como gafas y tapabocas, todo esto con el objetivo de medir la precisión del reconocimiento ante diferentes escenarios. A continuación, se anexa una imagen a modo de mosaico donde se observa cada prueba realizada

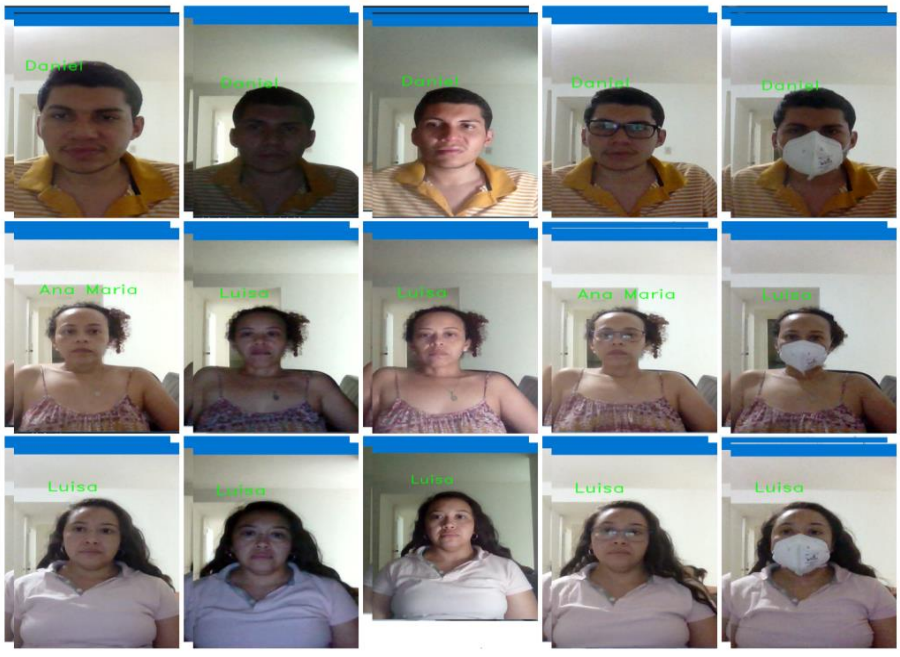

Figura 11: identificación de los sujetos de prueba en diferentes escenarios

Observamos que en términos generales la identificación de los usuarios suele ser acertada ante cambios de luminosidad y con el uso de accesorios, sin embargo es necesario mencionar que las participantes Ana María y Luisa comparten similitudes faciales al ser hermanas de sangre, ello conduce a que la identificación de Ana María suela dar falsos positivos con Luisa lo cual es una falencia que puede ser subsanada definiendo un umbral de decisión que será comparado con un valor numérico que el algoritmo de Fisher arroja en su función de predicción y que puede ser interpretado como un grado de verosimilitud de la predicción.

9. Conclusiones

En este trabajo se implementó una aplicación que integra la comunicación por redes y el reconocimiento facial, con el propósito de ofrecer una alternativa de identificación en los sistemas de control de acceso del metro de Medellín. El trabajo fue realizado únicamente con software de código libre tanto para la implementación, las mediciones y las pruebas de desempeño. En la elaboración de este informe, se ha considerado fundamental dividir los temas tratados en secciones con el objetivo de facilitar la comprensión a lo largo de la lectura. En esta sección final, se presentan las conclusiones generales del proyecto en su conjunto.

Durante el desarrollo de este proyecto, se ha logrado obtener un conocimiento profundo sobre el reconocimiento facial y las ventajas y desventajas de este tipo de algoritmos. Se aprecia que los algoritmos usados tienen problemas al tratar con usuarios parecidos debido a que la firma facial de estos tiene muchas coincidencias, se propone por lo tanto definir un valor de umbral que al ser comparado con la estimación propia del método de Fisher permita medir la verosimilitud de la predicción realizada.

Es notable la necesidad de encriptar los mensajes que se envían entre los clientes y el servidor en los sistemas de control de acceso, pues como se observó en los resultados cualquier ciberataque que intercepte los paquetes de la capa de transporte puede acceder a la información que estos portan lo cual es información sensible al estar dando la ubicación en tiempo real de una persona, lo cual puede representar un riesgo para la seguridad e integridad de estos.

Se aprecia la necesidad de implementar a futuro una interfaz gráfica (frontend) y una base de datos (backend) que permita realizar enrolamientos y generación de reportes desde el servidor, con el objetivo de crear un prototipo de plataforma de control de acceso que permita ver de forma más controlada y ordenada la información proveniente de los clientes

10. Referencias

[1] Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology, 14(1), 4-20.

[2] *EUR-Lex - 32016R0679 - EN - EUR-Lex*. (s/f). Europa.Eu. Recuperado el 12 de julio de 2023, de https://eur-lex.europa.eu/eli/reg/2016/679/oj

[3] Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springe

[4] Castleman, K. R. (2009). Digital Image Processing. Pearson Prentice Hall.

[5] Li, S. Z., & Jain, A. K. (2011). Handbook of face recognition. Springer Science & Business Media.

[6] Google LLC. (2021). MediaPipe: Cross-platform framework for multimodal applied ML. Recuperado de https://google.github.io/mediapipe/

[7] Stevens, W. R., Fenner, B., & Rudoff, A. M. (2004). UNIX network programming (Vol. 1). Addison-Wesley Professional.

[8] Stallings, W. (2013). Cryptography and network security: Principles and practice. Pearson Education.

[9] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media.

[10] *OpenCV: Face Recognition with OpenCV*. (2023). Opencv.org. Recuperado el 12 de julio de 2023, de https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html

[11] Christine Dewi[1,2], Rung-Ching Chen[1], Xiaoyi Jiang[3], Hui Yu.(2022). Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks

[12] GNS3. (2021). About GNS3. Recuperado de https://www.gns3.com/about

[13] Tanenbaum, A. S., Wetherall, D. J. (2011). Computer Networks. Pearson Education.