

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TELECOMUNICACIONES

Tema:

SISTEMA DE CONTROL DE ACCESO AUTOMATIZADO CON INTELIGENCIA ARTIFICIAL PARA EL MONITOREO DE ESTUDIANTES Y DOCENTES EN LOS TALLERES TECNOLÓGICOS DE LA FISEI

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniera en Telecomunicaciones

ÁREA: Comunicaciones

LÍNEA DE INVESTIGACIÓN: Electrónica, telecomunicaciones y Sistemas de control

AUTOR: Silvia Guadalupe Barba Proaño

TUTOR: Ing. Geovanni Danilo Brito Moncayo, Mg.

Ambato - Ecuador

febrero – 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE CONTROL DE ACCESO AUTOMATIZADO CON INTELIGENCIA ARTIFICIAL PARA EL MONITOREO DE ESTUDIANTES Y DOCENTES EN LOS TALLERES TECNOLÓGICOS DE LA FISEI, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Silvia Guadalupe Barba Proaño, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2025.

Firmado electrónicamente por:
GEOVANNI DANILO
BRITO MONCAYO

Ing. Geovanni Danilo Brito Moncayo, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA DE CONTROL DE ACCESO AUTOMATIZADO CON INTELIGENCIA ARTIFICIAL PARA EL MONITOREO DE ESTUDIANTES Y DOCENTES EN LOS TALLERES TECNOLÓGICOS DE LA FISEI es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2025.



Silvia Guadalupe Barba Proaño
C.C. 1805113899
AUTORA

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2025.

Silvia Guadalupe Barba Proaño
C.C. 1805113899
AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Silvia Guadalupe Barba Proaño estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE CONTROL DE ACCESO AUTOMATIZADO CON INTELIGENCIA ARTIFICIAL PARA EL MONITOREO DE ESTUDIANTES Y DOCENTES EN LOS TALLERES TECNOLÓGICOS DE LA FISEI, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, febrero 2025.

Firmado electrónicamente por:
FRANKLIN OSWALDO MAYORGA MAYORGA

Ing. Franklin Mayorga, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
JUAN PABLO PALLO NORONA

Ing. Juan Pablo Pallo Noroña Mg.
PROFESOR CALIFICADOR

Firmado electrónicamente por:
MARIO GEOVANNY GARCIA CARRILLO

Ing. Mario Geovanny Garcia Carrillo Mg.
PROFESOR CALIFICADOR

DEDICATORIA

"Esta tesis está dedicada, en primer lugar, a Elohim, por ser mi guía constante, otorgándome fortaleza, sabiduría y salud para superar cada desafío en este camino y permitiéndome alcanzar las metas propuestas.

Con todo mi amor y gratitud, a mis padres y hermanos, pilares fundamentales de mi vida. Su apoyo incondicional, sacrificio y dedicación han sido una fuente de inspiración inagotable, y su fe en mí me ha dado la fuerza para culminar esta etapa universitaria.

A mi amado esposo, mi compañero de vida, cuya paciencia, amor y aliento han sido esenciales para mantenerme firme incluso en los momentos más difíciles. Gracias por caminar a mi lado en este viaje.

A mis hijos, mi mayor fuente de inspiración, esperanza y alegría. Cada paso dado y cada meta alcanzada lleva consigo el sueño de construir un futuro lleno de posibilidades y amor para ustedes".

A todos ustedes, dedico este logro con un profundo agradecimiento y todo mi corazón

.

Guadalupe Barba

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a mi alma mater por brindarme la oportunidad de formarme en una profesión y por permitirme ser parte de su prestigiosa comunidad. Es un honor llevar con orgullo el nombre de esta institución que ha sido fundamental en mi crecimiento académico y personal.

Agradezco profundamente a los docentes que, a lo largo de mi formación, compartieron sus conocimientos y experiencias, contribuyendo significativamente a mi desarrollo integral. Su dedicación y esfuerzo dejaron una huella duradera en mi camino.

Deseo extender un agradecimiento especial al Ing. Geovanni Danilo Brito Moncayo, Mg, mi docente tutor, por su invaluable guía, paciencia, disponibilidad y apoyo a lo largo de todo el proceso de investigación. Su vasta experiencia, conocimientos y compromiso fueron fundamentales para orientar mi trabajo y enriquecerlo con sus valiosos aportes.

Finalmente, a mis compañeros de clase y amigos, quienes han sido un gran apoyo durante todo este proceso. Sus palabras de aliento, intercambio de ideas y su espíritu de compañerismo han sido invaluables, haciendo de esta etapa una experiencia más significativa y enriquecedora. Su influencia ha dejado una huella imborrable en mi vida.

Con profundo agradecimiento.

Guadalupe Barba

ÍNDICE GENERAL DE CONTENIDOS

PORTADA i

APROBACIÓN DEL TUTOR ii

AUTORÍA iii

DERECHOS DE AUTOR iv

APROBACIÓN DEL TRIBUNAL DE GRADO v

DEDICATORIA vi

AGRADECIMIENTO vii

ÍNDICE GENERAL DE CONTENIDOS viii

ÍNDICE DE TABLAS xi

ÍNDICE DE FIGURAS xii

ÍNDICE DE ANEXOS xv

RESUMEN EJECUTIVO xvi

ABSTRACT xvii

CAPÍTULO I. MARCO TEÓRICO 1

1.1 Tema de investigación 1

1.1.1 Planteamiento del problema 1

1.2 Antecedentes investigativos 3

1.3 Fundamentación teórica 5

1.3.1 Integración de sistemas de control de acceso y monitoreo 5

1.3.2 Interfaces 7

1.3.3 Reconocimiento facial 8

1.3.4 Algoritmos de detección y análisis de imágenes 9

1.3.5 Herramientas de Desarrollo para Reconocimiento Facial 11

1.3.6 Técnicas clave en análisis de imágenes 15

1.3.7 Control de acceso físico 16

1.3.8 Protocolos y normativas que abordan tanto la seguridad física de las instalaciones como la protección de datos personales. 17

1.3.9 Tipos de seguridades para el control de acceso 18

1.3.10 Electro cerraduras 20

1.4 Objetivos 21

1.4.1 Objetivo general 21

1.4.2 Objetivos específicos 21

CAPÍTULO II. METODOLOGÍA 22

2.1 Materiales 22

2.2 Métodos 23

2.2.1 Modalidad de la investigación 23

2.2.2 Recolección de información 23

2.2.3 Procesamiento y análisis de datos 24

CAPÍTULO III. RESULTADOS Y DISCUSIÓN 25

3.1 Descripción de la finalidad de la propuesta. 25

3.2 Desarrollo de la propuesta 26

3.2.1 Requerimientos del Sistema 26

3.2.2 Esquema General del Sistema 27

3.2.3 Selección de componentes y dispositivos 32

3.2.4 Diseño y configuración de la programación del reconocimiento 39

3.2.5 Tiempos de procesamiento 53

3.2.6 Implementación y verificación de funcionamiento 53

3.2.7 Presupuesto de implementación del proyecto 64

3.2.8 Presupuesto de implementación del proyecto 64

3.2.9 Costo Final 65

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 66

4.1 Conclusiones 66

4.2 Recomendaciones 67

REFERENCIAS BIBLIOGRÁFICA 69

ANEXOS 73

ÍNDICE DE TABLAS

Tabla 1. Tipos de seguridad y sus dispositivos 19

Tabla 2. Ventajas, limitaciones y desafíos de las electroceraaduras. 21

Tabla 3. Características de hardware 22

Tabla 4. Características del software 22

Tabla 5. Criterio de recolección de información 24

Tabla 6. Comparativa de cámara web 32

Tabla 7. Microcontroladores 35

Tabla 8. Comparativa de cerradura eléctrica 36

Tabla 9. Comparativa de Bases de datos. 38

Tabla 10. Beneficios de Firebase para el sistema de acceso po reconocimiento facial. 39

Tabla 11. Pruebas de procesamiento según el número de capturas y tiempo de procesamiento. 53

Tabla 12 . Insumos para el prototipo 64

Tabla 12. Horas tomadas para el desarrollo del prototipo 64

Tabla 13.Salario para un Ingeniero en Telecomunicaciones 65

ÍNDICE DE FIGURAS

Figura 1. Reconocimiento facial. 8

Figura 2. Control de acceso. 16

Figura 3. Partes de la electrocerradura 20

Figura 4. Esquema general 27

Figura 5. Esquema electrónico 29

Figura 6. Diagrama de procesamiento de IA de la imagen 31

Figura 7. Algortimos de reconocmiento facial y su proceso. 34

Figura 8. Registro de usuarios. 40

Figura 9. Flujo de trabajo del registro de usuario. 41

Figura 10. Control de registro cuando el usuario ya existe en el sitema. 42

Figura 11. Validación para un nuevo registro. 42

Figura 12. Interfaz de registro. 43

Figura 13. Captura de imágenes del usuario. 43

Figura 14. Seguridad de aceeso. al sistema de registro. 44

Figura 15. Visualización del número de toma de capturas. 44

Figura 16. Estructura de almacenamiento en dataset. 45

Figura 17. dataset carpeta donde esta almacenada las imágenes 45

Figura 18. Almacenamiento en Firebase. 47

Figura 19. Reconocimiento y detección facial. 48

Figura 20. Flujo de de detección y reconocmiento. 49

Figura 21. Conexión y recuperación de datos en Firebase..... 50

Figura 22. Conexiones electricas ..... **¡Error! Marcador no definido.**

Figura 23. Conexiones del sistema de acceso. 55

Figura 24. Implementación del prototipo. 56

Figura 25. Registro de usuario. 57

Figura 26. Listados de usuarios creados en dataset..... 57

Figura 27. Fotos proporcionadas..... 58

Figura 28. Archivo Txt. 58

Figura 29. Nuevo registro 59

Figura 30. Interfaz de la Jetson Nano, inicio de códigos de reconocimiento..... 59

Figura 31. Proceso de reconocimiento facial. 60

Figura 32. Usuario desconocido..... 60

Figura 33. Usuario reconocido con cédula de identidad 61

Figura 34. Individuo no registrado denominado "desconocido". 61

Figura 35. Alertas notificadas en Telegram. 62

Figura 36. Usuario reconocido por el sistema. 62

Figura 37. Cerradura magnética otorga acceso de ingreso. 63

Figura 38. Reconocimiento de varios rostros a la vez. 63

Figura 39. Esquema general de las herramientas y algoritmos en los procesos del sistema de acceso. 73

Figura 40. Vista Superior de la placa Jetson. 74

Figura 41. Vista inferior de la placa. 74

Figura 42. Vista delantera y tracera de la placa Jetson. 75

Figura 43. Esquema de pines GPIO. 76

Figura 45. Especificaciones técnicas de la cerradura magnética. 77

Figura 46. Genius QCam Full HD 1080P. 78

Figura 47. Datasheet cámara 78

Figura 48. Collage de fotografías de implementación del sistema. 79

Figura 49. Salarios mínimos sectoriales y tarifas 2024 102

ÍNDICE DE ANEXOS

Anexo A. Diagramas 73

Anexo B. Datasheet y descripciones de la placa Jetson Nano modelo P3541 74

Anexo C. Datasheet y descripciones de la cerradura electromagnética 77

Anexo D. Datasheet y descripciones de la cámara web. 78

Anexo E. Implementación del prototipo. 79

Anexo F. Código python de cada fase del sistema. 80

Anexo G. Implementación del prototipo capturas de docentes. 96

Anexo H. Implementación del prototipo capturas de encargados de Talleres tecnológicos. 98

Anexo I. Implementación del prototipo capturas de estudiantes. 99

Anexo I. Salarios mínimos sectoriales y tarifas 2024 102

RESUMEN EJECUTIVO

El presente trabajo de investigación se centra en el desarrollo de un sistema de control de acceso automatizado con inteligencia artificial para el monitoreo de estudiantes y docentes en los talleres tecnológicos de la FISEI de la Universidad Técnica de Ambato.

El proyecto aborda desde el análisis de los requerimientos técnicos y operativos hasta la implementación de un sistema inteligente que combina hardware y software especializado.

Se diseñaron esquemas de integración del sistema que incluyen el uso de dispositivos de captura biométrica y cámaras de reconocimiento facial, conectados a una plataforma de inteligencia artificial. El sistema permite identificar y registrar automáticamente las entradas de los usuarios, garantizando un control eficiente. Además, se establecieron parámetros para gestionar alertas en tiempo real y generar reportes detallados sobre la asistencia y permanencia de los usuarios en los talleres.

La implementación del sistema incluyó el desarrollo de algoritmos de aprendizaje automático para optimizar el reconocimiento facial y la autenticación de usuarios, así como la integración de una interfaz amigable que facilita su uso por parte del personal administrativo. Se realizaron pruebas funcionales en entornos simulados y reales, verificando la precisión del reconocimiento y la robustez del sistema frente a diferentes condiciones de operación.

Finalmente, el sistema se validó mediante pruebas piloto en los talleres tecnológicos, demostrando su eficacia en la gestión del acceso y el monitoreo continuo, aportando a la seguridad y optimización del uso de los recursos disponibles en la FISEI.

Palabras clave: Control de acceso, inteligencia artificial, reconocimiento facial, monitoreo, automatización.

ABSTRACT

The present research work focuses on the development of an automated access control system utilizing artificial intelligence for monitoring students and teachers of the faculty in the technological workshops of the FISEI at the Technical University of Ambato.

The project encompasses everything from analyzing technical and operational requirements to implementing an intelligent system that integrates specialized hardware and software.

Integration schemes for the system were designed, which include the use of biometric capture devices and facial recognition cameras connected to an artificial intelligence platform. This system enables automatic identification and registration of user entries, ensuring efficient control. Additionally, parameters were established to manage real-time alerts and generate detailed reports on user attendance and duration in the workshops.

The implementation of the system included the development of machine learning algorithms to optimize facial recognition and user authentication, as well as the integration of a user-friendly interface that facilitates its use by administrative personnel. Functional tests were conducted in both simulated and real environments, verifying the accuracy of recognition and the robustness of the system under various operational conditions.

Finally, the system was validated through pilot tests in the technological workshops, demonstrating its effectiveness in access management and continuous monitoring, contributing to security, and optimizing the use of available resources at FISEI.

Keywords: Access control, artificial intelligence, facial recognition, monitoring, automation.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA DE CONTROL DE ACCESO AUTOMATIZADO CON INTELIGENCIA ARTIFICIAL PARA EL MONITOREO DE ESTUDIANTES Y DOCENTES EN LOS TALLERES TECNOLÓGICOS DE LA FISEI.

1.1.1 Planteamiento del problema

En la actualidad, la eficacia de los sistemas de control de acceso en las instituciones educativas se ve gravemente comprometida debido a la falta de inversión en tecnología adecuada. Esta insuficiencia presupuestaria limita la capacidad de implementar soluciones innovadoras que garanticen un ingreso seguro y eficiente para estudiantes y docentes. Sin embargo, muchos centros aún utilizan métodos manuales y obsoletos que generan largas filas y retrasos en la entrada. En estudios, el 30% de los estudiantes reporta haber experimentado demoras significativas al ingresar a sus instalaciones debido a sistemas de control de acceso inadecuados.

La ausencia de sistemas automatizados y eficientes para el control de acceso representa un obstáculo significativo en la gestión educativa. La falta de tecnologías avanzadas dificulta el monitoreo efectivo de entradas y salidas, lo que puede llevar a problemas como el absentismo escolar y la dificultad para gestionar la asistencia. En Ecuador, el absentismo escolar se ha convertido en un problema creciente, con un 25% de los estudiantes faltando a clases regularmente. La implementación de un sistema que utilice inteligencia artificial para el monitoreo de entradas podría no solo mejorar la gestión del acceso, sino también proporcionar datos valiosos sobre patrones de asistencia, permitiendo a las instituciones tomar decisiones informadas para abordar este problema.

Además, la falta de protocolos claros para el control de acceso genera descoordinación entre los diferentes departamentos dentro de las instituciones educativas. Esta ausencia de estrategias conjuntas y comunicación efectiva resulta en una gestión fragmentada del acceso a distintos departamentos, dificultando la identificación de patrones de

comportamiento y necesidades específicas. Esta descoordinación no solo afecta la seguridad, sino que también socava la confianza de estudiantes y docentes en las medidas implementadas para proteger su bienestar dentro del entorno educativo. En un contexto más amplio, la seguridad en las instituciones educativas es un tema de creciente preocupación en América Latina; según datos recientes, el 40% de los padres expresa inquietud sobre la seguridad en las instituciones.

La implementación de sistemas avanzados de control de acceso no solo se construyó a mejorar la seguridad física, sino que también se fomentó un ambiente educativo más seguro y propicio para el aprendizaje. La falta de medidas adecuadas puede resultar en una disminución del interés por parte de los estudiantes y sus familias hacia las instituciones educativas, afectando así su reputación y capacidad para atraer a nuevos alumnos. Por lo tanto, es fundamental abordar estos desafíos mediante soluciones tecnológicas que integren inteligencia artificial y automatización para optimizar el control de acceso.

Este enfoque integral permite apreciar la complejidad del problema del control de acceso en talleres tecnológicos y subraya la necesidad urgente de soluciones innovadoras que integren tecnología avanzada. Con ello, se espera no solo mejorar la seguridad y eficiencia en las entradas y salidas, sino también contribuir al desarrollo académico y personal de los estudiantes al crear un entorno más ordenado y accesible. En resumen, es imperativo que las instituciones educativas reconozcan la importancia del control automatizado como una herramienta esencial para garantizar un ambiente seguro y eficiente que favorezca el aprendizaje y desarrollo integral de sus estudiantes.

Finalmente, al adoptar un sistema automatizado que utilice inteligencia artificial para el monitoreo del acceso, los talleres tecnológicos no solo mejoran su infraestructura tecnológica, sino también promueven una cultura institucional centrada en la seguridad y el bienestar. Esto no solo beneficia a los estudiantes y docentes actuales, sino que también establece un precedente positivo para futuras generaciones, convirtiendo a esta institución en referente en innovación educativa y seguridad escolar.

1.2 Antecedentes investigativos

Tras revisar la literatura académica en los archivos de diversas universidades nacionales e internacionales, así como en bases de datos en línea de artículos científicos, se presentan los siguientes antecedentes de investigación que contribuyen al tema de estudio.

En el trabajo titulado “Control de acceso vehicular con visión artificial para urbanizaciones en la ciudad de Piura, 2023” de Castillo y Santamaría, se implementó un sistema basado en visión artificial para mejorar el control vehicular en urbanizaciones. Se utilizaron tecnologías como Python, React, y algoritmos de machine learning para la identificación de vehículos y matrículas. Los resultados mostraron una reducción significativa en el tiempo de registro vehicular, con una disminución de 75.25 segundos en el ingreso y 121.18 segundos en la salida. Asimismo, el sistema logró un 100% de efectividad en el reconocimiento de matrículas y la satisfacción de los usuarios aumentó en un 32.5%, demostrando que la visión artificial mejora notablemente la eficiencia del control vehicular [1].

En el 2022, Darío Javier Llanga Herrera presentó su investigación titulada "Diseño de un prototipo de entorno de video vigilancia integrado Open Source utilizando cámaras integradas en vehículos", en la Escuela Superior Politécnica de Chimborazo. El trabajo se enfocó en la implementación de un sistema de video vigilancia utilizando visón artificial, con algoritmos de SIFT y SURF para la detección y análisis de imágenes. El sistema se integró a cámaras vehiculares, permitiendo un monitoreo eficiente en tiempo real en el parqueadero de la Facultad de Informática y Electrónica. Tras evaluar la efectividad de los algoritmos, el algoritmo SURF demostró ser más eficiente con un 89% de precisión comparado con el 56% de SIFT, mejorando la localización y cobertura visual para la detección de incidentes. Se concluye que el sistema basado en Open Source mejoró significativamente la seguridad, y se recomienda combinar distintos sistemas de extracción de características para construir sistemas más robustos [2].

En 2022, Jorge André Cabrera Savinovich y Dennys Saúl Pérez Sandoval presentaron su proyecto titulado “Diseño y simulación de sistema de videovigilancia para el

monitoreo de cultivos y control de acceso del personal", en la Escuela Superior Politécnica del Litoral (ESPOL), Ecuador. Este trabajo se centró en desarrollar un sistema integral de seguridad mediante el uso de tecnología de videovigilancia basada en cámaras conectadas a microcontroladores como el Raspberry Pi y el módulo ESP32. El sistema implementó reconocimiento facial utilizando el algoritmo Eigen Faces para controlar el acceso del personal autorizado, garantizando seguridad en una empresa agrícola. Lograron una solución eficiente y escalable que no solo garantizó la seguridad del personal, sino que también permitió la prevención del robo de cultivos en tiempo real, con alertas enviadas a través de WhatsApp al detectar movimientos sospechosos. La implementación del sistema resultó en una reducción del 30% en incidentes de seguridad durante el primer año, evidenciando su efectividad en la protección de activos agrícolas y la optimización de recursos operativos [3].

En 2020, Eduardo Arturo Briones Gárate presentó su trabajo titulado "Sistema web de reconocimiento facial para control de acceso biométrico, utilizando inteligencia artificial" en la Escuela Superior Politécnica del Litoral (ESPOL), Ecuador. Este proyecto se enfocó en el desarrollo de FaceProBri, un sistema de control de acceso biométrico basado en reconocimiento facial, diseñado para empresas y organizaciones. El sistema emplea redes neuronales siamesas y algoritmos como Viola-Jones para la detección y comparación de rostros, validando la identidad de los usuarios mediante una base de datos preestablecida. El uso de tecnologías como Python y TensorFlow permitió un procesamiento eficiente de imágenes, integrando además el framework Django para la gestión del backend. FaceProBri fue concebido para optimizar la seguridad y automatización en múltiples entornos, incluyendo hospitales, supermercados y controles migratorios, contribuyendo a la mejora del tiempo y la precisión en los procesos de verificación de identidad se logró una tasa de precisión del 95% en el reconocimiento facial durante las pruebas, lo que indica una alta efectividad en la validación de usuarios autorizados. Además, el sistema redujo el tiempo promedio de verificación a menos de 2 segundos por usuario, mejorando notablemente la eficiencia operativa en entornos como supermercados y hospitales. En un piloto realizado en una cadena de supermercados local, se registró una disminución del 40% en incidentes relacionados con el acceso no autorizado en los primeros seis meses tras la implementación del sistema. Estos datos evidencian cómo FaceProBri no

solo mejora la seguridad, sino que también optimiza procesos administrativos y contribuye a un entorno más seguro y eficiente para las organizaciones que lo adoptan [4].

En 2020, César Nacipucha y Jonathan Frías, en su investigación titulada "Diseño de un prototipo de control de acceso a través de reconocimiento facial mediante la creación de un algoritmo basado en software libre utilizando LattePanda", en la Universidad Politécnica Salesiana del Ecuador sede Guayaquil, desarrollaron un sistema de control de accesos que garantiza un alto nivel de seguridad para espacios restringidos. Utilizando algoritmos inteligentes, el sistema identificó y autenticó a los usuarios mediante reconocimiento facial, implementándose en un dispositivo LattePanda. Como resultado, el prototipo logró una tasa de precisión del 92% en la identificación de usuarios autorizados durante las pruebas iniciales, con un tiempo promedio de autenticación de 1.5 segundos por usuario. Además, se reportó una reducción del 35% en incidentes de acceso no autorizado en las áreas donde se implementó el sistema durante los primeros seis meses. La creación de una base de datos en la nube facilitó el acceso remoto y la gestión continua de los datos generados, permitiendo un monitoreo efectivo que optimizó la seguridad y mejoró la experiencia del usuario al proporcionar un método seguro y eficiente para autenticarse [5].

1.3 Fundamentación teórica

El desarrollo de esta investigación demanda una comprensión profunda del contexto de las tecnologías de seguridad, para así diseñar una solución eficiente y adaptada a las necesidades específicas de talleres.

1.3.1 Integración de sistemas de control de acceso y monitoreo

La integración de sistemas de control de acceso y monitoreo es esencial para mejorar la seguridad en diversas organizaciones. Esta integración permite combinar diferentes tecnologías, como cámaras de vigilancia, sensores de movimiento y sistemas de

control de acceso, para ofrecer una solución de seguridad unificada que optimiza la gestión del acceso y proporciona una vigilancia constante [6].

Un sistema de control de acceso moderno incluye componentes tanto físicos como digitales. Los elementos físicos pueden ser cerraduras electrónicas, lectores de tarjetas y sistemas biométricos, mientras que los componentes digitales implican software que gestiona el acceso y monitorea la actividad en tiempo real. Esta combinación no solo mejora la seguridad al restringir el acceso a personas no autorizadas, sino que también facilita la identificación y el seguimiento de los usuarios dentro de las instalaciones [7] [8].

La integración de sistemas también permite una respuesta más rápida ante incidentes. Por ejemplo, si un sensor de movimiento se activa, el sistema puede alertar automáticamente a los responsables de seguridad y proporcionar imágenes en tiempo real desde las cámaras cercanas. Esto mejora significativamente la capacidad de respuesta ante situaciones potencialmente peligrosas [9].

Además, la automatización juega un papel crucial en la eficiencia operativa. Los sistemas integrados pueden simplificar procesos como el registro de visitantes, permitiendo a las organizaciones gestionar el acceso temporal para proveedores o clientes sin comprometer la seguridad general [7]. También se pueden generar informes detallados sobre las actividades dentro del edificio, lo que ayuda en la toma de decisiones informadas y en la planificación estratégica [8].

Sin embargo, para lograr una integración efectiva, es fundamental considerar la compatibilidad entre los diferentes sistemas y establecer políticas claras sobre el control de acceso. La capacitación del personal en estas políticas es crucial para mantener un entorno seguro. En resumen, la integración de sistemas de control de acceso y monitoreo no solo mejora la seguridad física, sino que también optimiza la gestión operativa y proporciona una respuesta más efectiva ante incidentes [7].

Esta estrategia se ha vuelto indispensable en un contexto donde las amenazas a la seguridad son cada vez más sofisticadas, haciendo necesario que las organizaciones se adapten mediante soluciones tecnológicas avanzadas [10].

1.3.2 Interfaces

Una interfaz es un punto de interacción entre diferentes componentes, dispositivos, o sistemas que permite el intercambio de información o la realización de tareas conjuntas. En tecnología, las interfaces se utilizan para facilitar la comunicación entre hardware, software o usuarios [11].

Tenemos varios tipos de interfaces como son los siguientes:

- ***Interfaces de Usuario (UI)***

Estas interfaces conectan a los usuarios con sistemas o dispositivos. Pueden ser interfaz Gráfica de Usuario (GUI) que utiliza elementos visuales como ventanas, íconos, y menús para facilitar la interacción. Interfaz de Línea de Comandos (CLI) la cual permite la interacción mediante comandos escritos e interfaces de Voz (VUI que facilitan la comunicación a través del reconocimiento de voz [12].

- ***Interfaces de Programación de Aplicaciones (API)***

Las APIs permiten que diferentes aplicaciones o sistemas intercambien información de manera estándar. Proveen funciones o servicios específicos sin exponer la implementación interna.

- ***Interfaces de Hardware***

Conectan dispositivos físicos, como periféricos y sistemas internos.

- ***Interfaces de Red***

Gestionan la comunicación entre sistemas a través de redes como pueden ser protocolos HTTP, TCP/IP o FTP [13].

1.3.3 Reconocimiento facial

El reconocimiento facial es una tecnología que permite identificar o verificar la identidad de una persona a partir de su rostro, utilizando imágenes digitales o vídeos como se muestra en la Figura 1. Este proceso se basa en el análisis de características faciales únicas que son extraídas y comparadas con una base de datos de rostros conocidos. Aunque tradicionalmente se utilizaban métodos como el clasificador en cascada de Viola-Jones, la evolución hacia redes neuronales convolucionales ha mejorado significativamente la precisión del reconocimiento facial, reduciendo las tasas de error a menos del 1% en los sistemas más avanzados [14].

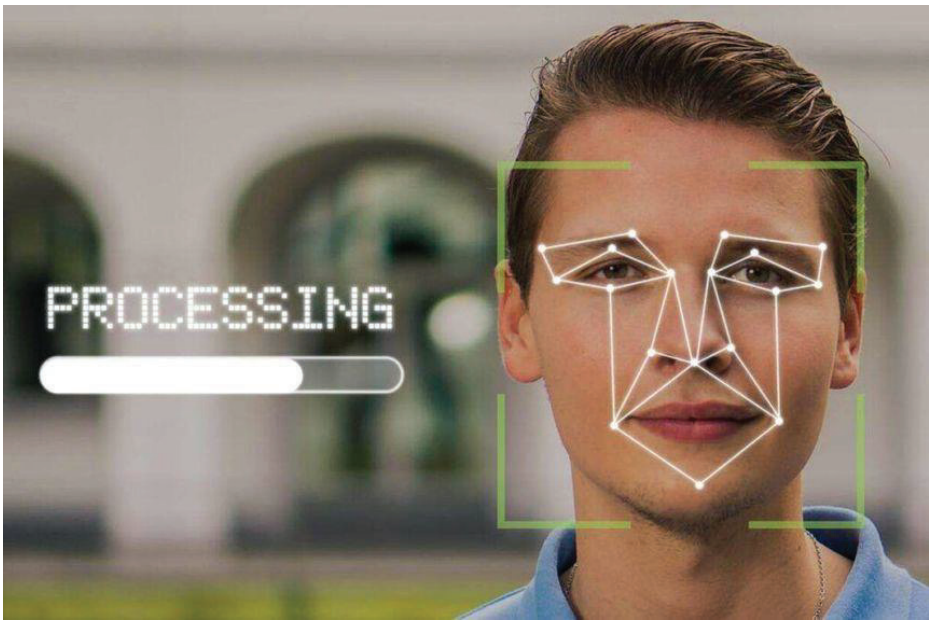


Figura 1. Reconocimiento facial.

Los sistemas de reconocimiento facial operan en dos modos principales: verificación (comparando una imagen con otra) e identificación (comparando una imagen desconocida con múltiples imágenes en una base de datos). La tecnología se aplica ampliamente en seguridad personal, control de accesos, y servicios financieros, entre otros. Sin embargo, enfrenta desafíos como sesgos algorítmicos y variaciones en las condiciones de iluminación que pueden afectar su rendimiento [15].

Además, el reconocimiento facial puede ser complementado con inteligencia artificial para mejorar su eficacia, permitiendo un análisis más rápido y preciso en tiempo real.

Los casos de uso incluyen desde la autenticación en dispositivos móviles hasta la identificación en multitudes por parte de fuerzas del orden. A medida que la tecnología avanza, se espera que su implementación sea aún más común y eficiente en diversas aplicaciones [16].

1.3.4 Algoritmos de detección y análisis de imágenes

El análisis y la detección de imágenes son áreas fundamentales dentro de la visión por computadora, que se han desarrollado significativamente con el avance de la tecnología y los algoritmos de procesamiento. Este campo tiene aplicaciones en diversas industrias, incluyendo la medicina, la seguridad, y el reconocimiento facial, entre otros [17].

El análisis de imágenes se centra en extraer información relevante de las imágenes para comprender su contenido visual. Inicialmente, se utilizaban métodos simples como la transformada de Fourier y el filtrado de imágenes. Sin embargo, con el desarrollo de algoritmos más sofisticados y el aumento en la capacidad computacional, se han implementado técnicas más avanzadas que permiten una mayor precisión en la extracción de datos [18].

Los algoritmos utilizados en el análisis de imágenes incluyen técnicas de aprendizaje automático como las redes neuronales convolucionales (CNN), que son especialmente efectivas para el reconocimiento de objetos. Estas redes aprenden automáticamente a identificar patrones y características relevantes en las imágenes, lo que las hace ideales para tareas como la clasificación y detección de objetos [19].

La detección de objetos es un componente clave del análisis de imágenes. Los algoritmos analizan imágenes o fotogramas de video para identificar y localizar objetos. Esta tecnología se aplica en vehículos autónomos, donde ayuda a identificar peatones y obstáculos, así como en sistemas de seguridad para detectar intrusos. Sin embargo, enfrenta desafíos como la oclusión y la variación del tamaño de los objetos, lo que requiere mejoras continuas en los algoritmos para aumentar su precisión [19].

a. Métodos Basados en Aprendizaje Profundo para Reconocimiento Facial

En los últimos años, el aprendizaje profundo ha revolucionado el campo del reconocimiento facial, proporcionando soluciones más precisas y robustas. Entre las técnicas más destacadas se encuentran los modelos basados en redes neuronales profundas, como FaceNet, DeepFace y ArcFace, que aprovechan las capacidades de las redes neuronales convolucionales (CNN, por sus siglas en inglés) para identificar patrones únicos en los rostros humanos.

- ***FaceNet***

FaceNet, desarrollado por Google, es un modelo innovador que introduce el concepto de **embeddings faciales**. En lugar de simplemente clasificar rostros, este modelo transforma cada imagen facial en un vector numérico (embedding) en un espacio multidimensional, donde la distancia entre dos vectores representa la similitud entre los rostros. Este enfoque permite realizar tanto identificación como verificación facial con alta precisión. FaceNet utiliza la pérdida conocida como **triplet loss**, que optimiza las representaciones faciales al minimizar la distancia entre embeddings de la misma persona y maximizarla para personas diferentes [20].

- ***DeepFace***

DeepFace, desarrollado por Facebook, fue uno de los primeros modelos en alcanzar un nivel de precisión comparable al humano en el reconocimiento facial. Este sistema utiliza una arquitectura basada en redes neuronales profundas y logra alinear rostros de manera precisa antes de extraer características. El modelo entrena con millones de imágenes y, gracias a esta robusta base de datos, puede identificar rostros incluso en condiciones de iluminación y poses desafiantes [21].

- ***ArcFace***

ArcFace, por su parte, introduce una mejora significativa en los sistemas de reconocimiento facial al implementar una función de pérdida basada en márgenes angulares. Este enfoque asegura que los vectores generados para un rostro específico sean más distintivos, incluso en grandes bases de datos. Gracias a su capacidad para

generar embeddings altamente discriminativos, ArcFace se ha convertido en un estándar en aplicaciones modernas de reconocimiento facial [22].

b. Redes Neuronales Convolucionales (CNN) y Su Aplicación

Las CNN son el núcleo de los sistemas modernos de reconocimiento facial. Estas redes están diseñadas para procesar imágenes y extraer características jerárquicas, desde bordes simples en las capas iniciales hasta patrones complejos como la estructura de un rostro en las capas más profundas.

- ***Cómo Funcionan en Reconocimiento Facial***

1. **Preprocesamiento**: El sistema detecta y alinea el rostro para normalizar las poses y escalas.
2. **Extracción de Características**: Las CNN analizan la imagen para identificar patrones únicos, como la distancia entre los ojos, la forma de la nariz y otros rasgos distintivos.
3. **Generación de Embeddings**: El rostro es representado como un vector en un espacio de características.
4. **Comparación**: Los vectores se comparan mediante métricas como la distancia euclidiana o coseno para determinar similitudes.

1.3.5 Herramientas de Desarrollo para Reconocimiento Facial

El desarrollo de sistemas de reconocimiento facial ha sido facilitado por diversas bibliotecas y frameworks que ofrecen implementaciones eficientes de algoritmos y modelos avanzados. Entre las herramientas más utilizadas están OpenCV, Dlib, TensorFlow, PyTorch y face_recognition. Estas tecnologías proporcionan un soporte robusto para la detección, alineación, y análisis de rostros en imágenes y videos.

a. OpenCV

OpenCV (Open Source Computer Vision**)** es una biblioteca de visión por computadora ampliamente utilizada, diseñada para tareas como procesamiento de imágenes, detección de objetos y análisis de video. Su flexibilidad y facilidad de integración la convierten en una herramienta clave en proyectos de reconocimiento facial [23].

- ***Características principales:***

Implementa algoritmos clásicos como Haar Cascades y métodos modernos basados en redes neuronales.

Permite realizar preprocesamiento de imágenes, como normalización de iluminación o escalado, esenciales para la detección facial.

- ***Ventajas:***

Es de código abierto y cuenta con una comunidad activa que desarrolla nuevas funcionalidades.

Compatible con múltiples lenguajes de programación como Python, C++ y Java.

- ***Aplicaciones:***

Reconocimiento de rostros en tiempo real en cámaras de seguridad o sistemas de acceso.

b. Dlib

Dlib es una biblioteca de aprendizaje automático y visión por computadora conocida por su precisión en la detección y análisis facial. Integra algoritmos como Histogram of Oriented Gradients (HOG) y modelos basados en redes neuronales convolucionales (CNN) [24].

- ***Características principales:***

Ofrece modelos preentrenados para la detección facial y la localización de puntos clave (landmarks).

Compatible con frameworks de aprendizaje profundo como TensorFlow y PyTorch.

- ***Ventajas:***

Su precisión en la detección de rostros y puntos clave permite un procesamiento detallado para tareas avanzadas, como la alineación facial.

- ***Aplicaciones:***

Uso en sistemas biométricos y software de monitoreo.

Soporte para el análisis de expresiones faciales y emociones.

c. TensorFlow

TensorFlow, desarrollado por Google, es un framework de aprendizaje profundo ampliamente utilizado para construir, entrenar y desplegar redes neuronales. En el ámbito del reconocimiento facial, se emplea para entrenar modelos sofisticados como FaceNet [25].

- ***Características principales:***

Proporciona herramientas para definir arquitecturas de redes neuronales convolucionales (CNN) desde cero.

Incluye soporte para aceleración mediante GPU, esencial para manejar grandes volúmenes de datos de imágenes.

- ***Ventajas:***

Permite el uso de modelos preentrenados y la personalización de soluciones.

Ideal para aplicaciones de reconocimiento facial a gran escala.

- ***Aplicaciones:***

Entrenamiento de redes neuronales para extracción de embeddings faciales.

Implementación de sistemas en tiempo real con precisión mejorada.

d. PyTorch

PyTorch, desarrollado por Facebook, es una plataforma de aprendizaje profundo que se caracteriza por su enfoque dinámico en la creación de modelos. Su flexibilidad y facilidad de uso lo convierten en una herramienta popular para investigación y desarrollo en inteligencia artificial [26].

- ***Características principales:***

Soporte para redes neuronales dinámicas, lo que facilita la depuración y experimentación.

Compatible con modelos preentrenados en reconocimiento facial, como ArcFace y ResNet.

- ***Ventajas:***

Ofrece una interfaz amigable y una fuerte integración con bibliotecas científicas de Python.

Soporte para GPU, lo que mejora significativamente el rendimiento en el entrenamiento.

- ***Aplicaciones:***

Construcción de redes neuronales personalizadas para reconocimiento facial.

Uso en tareas avanzadas como la mejora de la calidad de imágenes faciales.

e. Face_recognition

Face_recognition es una biblioteca de Python que encapsula las funcionalidades de Dlib para ofrecer una solución simple y eficiente al reconocimiento facial. Es ampliamente conocida por su facilidad de uso [27].

- ***Características principales:***

Basada en embeddings faciales generados por modelos de aprendizaje profundo.

Ofrece funciones para la detección de rostros, reconocimiento y comparación con solo unas líneas de código.

- ***Ventajas:***

Ideal para principiantes o desarrolladores que buscan una solución rápida y funcional.

Compatible con múltiples formatos de imagen y video.

- ***Aplicaciones:***

Implementación de sistemas de acceso biométrico basados en reconocimiento facial.

Identificación automática en bases de datos de imágenes.

1.3.6 Técnicas clave en análisis de imágenes

El reconocimiento de patrones es esencial para la detección efectiva de objetos. Los sistemas pueden utilizar técnicas como la binarización y el filtrado para mejorar la calidad de las imágenes antes del análisis. Un estudio reciente demostró que un sistema basado en MATLAB y OpenCV logró una operatividad del 96.6% en condiciones controladas [28].

- ***Detección y Descripción de Puntos Característicos***

Los algoritmos SIFT (Scale-Invariant Feature Transform) y SURF (Speeded Up Robust Features) son ejemplos clásicos utilizados para detectar y describir puntos

característicos en imágenes. Estos algoritmos son cruciales para establecer correspondencias entre diferentes imágenes, lo que es especialmente útil en aplicaciones que requieren comparación entre imágenes multiespectrales [29].

1.3.7 Control de acceso físico

El control de acceso físico con inteligencia artificial (IA) es una tecnología avanzada que permite gestionar la entrada y salida de personas en espacios restringidos, utilizando sistemas de reconocimiento facial y otros métodos biométricos. Este enfoque no solo mejora la seguridad, sino que también optimiza la administración de accesos en diversas instalaciones, como oficinas, fábricas y talleres tecnológicos [30].

En apertura de puertas, el sistema de control de acceso utiliza cámaras y algoritmos de IA para identificar a los usuarios mediante el reconocimiento facial. Cuando una persona se aproxima a una puerta, el sistema captura su imagen y la compara con una base de datos de rostros autorizados. Si hay una coincidencia, se activa un mecanismo que abre la puerta automáticamente como se muestra en la Figura 2. Este proceso es rápido y eficiente, permitiendo un acceso fluido sin necesidad de llaves o tarjetas [7].

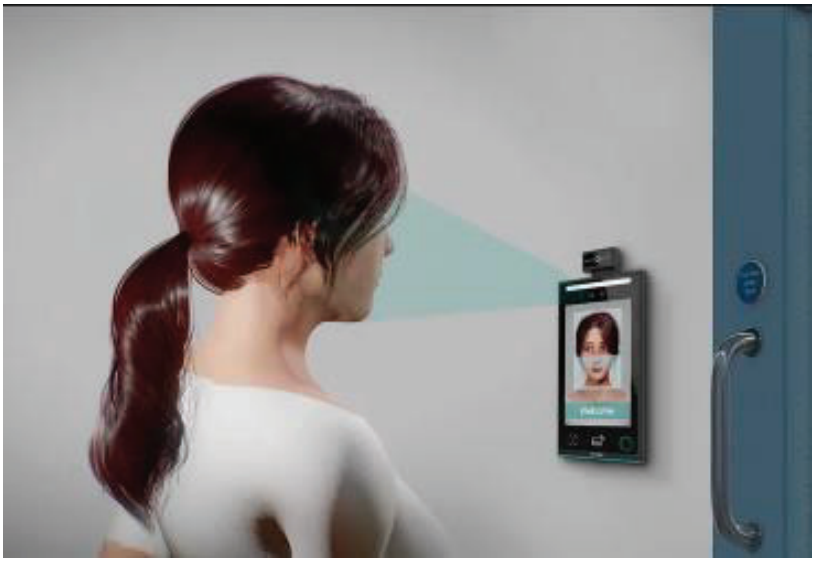


Figura 2. Control de acceso.

Además de la apertura de puertas, estos sistemas permiten establecer diferentes niveles de acceso a zonas restringidas. Por ejemplo, un empleado puede tener acceso a ciertas áreas mientras que otros pueden estar limitados a espacios específicos. La gestión de estos permisos se realiza a través de un software que integra la información del personal y sus derechos de acceso. Esto no solo mejora la seguridad al restringir el acceso no autorizado, sino que también facilita el cumplimiento de normativas internas y externas [14].

El registro de entrada y salida es otro componente clave del control de acceso físico con IA. A medida que las personas ingresan o salen del edificio, el sistema registra automáticamente estos eventos. Esto se puede hacer mediante un software que almacena datos en tiempo real, lo que permite a los administradores monitorear quién está presente en las instalaciones en cualquier momento. Esta funcionalidad es especialmente útil para la gestión del personal, ya que proporciona informes sobre asistencia y puede ayudar a identificar patrones inusuales que podrían indicar problemas de seguridad [26].

La implementación de un sistema de control de acceso físico con IA ofrece múltiples beneficios adicionales. Mejora la eficiencia operativa al reducir el tiempo dedicado al control manual y minimiza errores humanos en el registro. Además, al utilizar tecnologías avanzadas, se aumenta la precisión en la identificación y se disminuyen los riesgos asociados con métodos tradicionales como las tarjetas magnéticas o los códigos PIN [27].

1.3.8 Protocolos y normativas que abordan tanto la seguridad física de las instalaciones como la protección de datos personales.

A continuación, veremos algunas de las normativas clave a considerar:

- ISO/IEC 27001: Esta norma es fundamental para la gestión de la seguridad de la información, y te ayuda a establecer un sistema de gestión de seguridad de la información en tu proyecto. La implementación de reconocimiento facial

requiere medidas estrictas de protección de datos, ya que se trata de información biométrica.

- ISO/IEC 30137-1:2019: Esta norma está diseñada específicamente para la implementación de sistemas de reconocimiento facial en áreas de vigilancia. Incluye recomendaciones sobre el diseño, operación y evaluación de los sistemas de reconocimiento facial, y es relevante en escenarios de control de acceso.

- Normativa GDPR (Reglamento General de Protección de Datos de la UE): Si tu proyecto involucra el tratamiento de datos biométricos de personas dentro de la Unión Europea, o si trabajas con empresas que operan en este ámbito, debes cumplir con el GDPR, que clasifica los datos biométricos como datos sensibles y establece regulaciones estrictas para su uso y almacenamiento.

- Ley de Protección de Datos Personales (dependiendo del país): Es importante revisar la legislación local sobre la protección de datos. Por ejemplo, en Ecuador, aplica la Ley Orgánica de Protección de Datos Personales, que regula el tratamiento de datos biométricos y exige medidas de seguridad apropiadas.

- ANSI/UL 294 - Standard for Access Control System Units: Esta norma establece requisitos de desempeño para los sistemas de control de acceso, incluidos aquellos que utilizan tecnologías biométricas como el reconocimiento facial. Cubre aspectos como la resistencia a manipulaciones y la interoperabilidad con otros sistemas de seguridad.

1.3.9 Tipos de seguridades para el control de acceso a nivel mundial

En la tabla 1 veremos los tipos de seguridad con dispositivos o formas que tenemos para el control de acceso a cada una de ellas y una pequeña descripción.

Tabla 1. Tipos de seguridad y sus dispositivos

Tipo de seguridad	Dispositivos	Descripción
Física	Tarjetas de proximidad o llaves electrónicas	Para autenticar su acceso es necesario tener un dispositivo físico.
	Lectores biométricos	Reconocen características como huellas dactilares, reconocimiento facial o escaneo de iris.
	Teclados y contraseñas	Se debe ingresar un código o contraseña.
	Barreras	Puertas automáticas, cerraduras eléctricas, sensores de movimiento.
Biométrica	Reconocimiento facial	Captura y analiza rasgos faciales.
	Huellas dactilares	Patrones únicos de los dedos.
	Escaneo de iris o retina	Patrones específicos en los ojos.
	Reconocimiento de voz	Verifica a los usuarios por su todo y características vocales.
En la nube	Control de acceso basado en roles (RBAC)	Limita las acciones que un usuario puede realizar según su rol.
	Acceso condicional:	Permite o restringe el acceso según factores como ubicación, dispositivo o hora.
	Auditoría y monitoreo	Rastrea actividades para identificar accesos no autorizados.
Con inteligencia artificial (IA)	Reconocimiento de patrones.	Analiza comportamientos inusuales o intentos de acceso no autorizados.
	Aprendizaje automático	Optimiza la detección de amenazas basándose en datos históricos.
	Integración con cámaras inteligentes	Detecta anomalías en tiempo real.

El análisis de las tecnologías disponibles para sistemas de control de acceso y monitoreo a nivel mundial permite identificar y comparar las soluciones más avanzadas en seguridad, incluyendo métodos biométricos, autenticación multifactor, tarjetas RFID, videovigilancia inteligente y plataformas basadas en la nube. A través de la revisión de normativas internacionales, tendencias emergentes y casos de éxito, se evalúan las ventajas y limitaciones de cada tecnología, considerando su integración con la inteligencia artificial, IoT y Big Data para la optimización de la seguridad. Este estudio proporciona un marco de referencia para la implementación de sistemas más

eficientes, confiables y adaptables a distintos entornos, asegurando un control riguroso y en tiempo real del acceso y la vigilancia en diversos.

1.3.10 Electro cerraduras

Una electro cerradura es un dispositivo de cierre electrónico que utiliza un sistema eléctrico para controlar el acceso a una puerta, caja fuerte u otro compartimento. Este tipo de cerraduras se caracterizan por combinar componentes mecánicos con tecnología electrónica para incrementar la seguridad y facilitar su gestión. Tenemos varios ejemplos como podemos apreciar en la Figura 3. [31].



Figura 3. Partes de la electrocerradura

- ***El funcionamiento general de una electro cerradura consiste en:***

Recibir una señal de acceso a través de su sistema de autenticación, después procesar dicha señal en su unidad de control electrónica. Accionar el actuador eléctrico si la autenticación es exitosa, permitiendo que el mecanismo de cierre se libere, se bloquea nuevamente el acceso una vez cerrada la puerta o desactivado el sistema. [32].

- ***Ventajas, Limitaciones y Desafíos***

En la Tabla 2. Se ver una las características:

Tabla 2. Ventajas, limitaciones y desafíos de las electroceraaduras.

Ventajas	**Limitaciones y desafios**
Mayor seguridad: Sistemas avanzados de autenticación dificultan accesos no autorizados.	Dependencia de energía: En caso de fallos eléctricos, pueden quedar inoperativas sin respaldo.
Facilidad de uso: Permiten apertura sin llaves físicas.	Costo: Suelen ser más costosas que las cerraduras mecánicas tradicionales.
Integración: Compatibilidad con sistemas de automatización del hogar y control de accesos.	Vulnerabilidades tecnológicas: Sistemas mal diseñados pueden ser susceptibles a hackeos

.

1.4 Objetivos

1.4.1 Objetivo general

Implementar un Sistema de Control de Acceso Automatizado con inteligencia artificial para el monitoreo de Estudiantes y Docentes en los Talleres Tecnológicos de la FISEI.

1.4.2 Objetivos específicos

- Analizar las tecnologías disponibles de sistemas de control de acceso y monitoreo utilizados a nivel mundial.
- Aplicar algoritmos de inteligencia artificial para el control y monitoreo del acceso a los Talleres Tecnológicos de la FISEI.
- Validar el funcionamiento del sistema de control de acceso y monitoreo a implementarse.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

El desarrollo del proyecto contó con el respaldo de los talleres tecnológicos de la Facultad de Ingeniería en Sistemas Electrónica e Industrial (FISEI). Para la construcción del prototipo, se tomaron como referencia proyectos previos realizados en el ámbito de control de acceso y monitoreo, adaptando sus características a los requerimientos específicos de este proyecto. Adicionalmente, se consultaron fuentes bibliográficas confiables disponibles en los repositorios digitales de la Universidad Técnica de Ambato, lo que permitió fundamentar de manera sólida los aspectos teóricos y técnicos del sistema. En la Tabla 2 y Tabla 3 se describen las características de hardware y software seleccionados para el sistema.

Tabla 3. Características de hardware

Material	**Características**
NVIDIA Jetson Nano	Memoria RAM de 2GB, en cuanto al almacenamiento utiliza una tarjeta microSD como dispositivo de arranque y almacenamiento principal
Cámara Genius QCam 6000 Webcam	La resolución de videoes Full HD a 30 cuadros por segundo, lo que garantiza una imagen clara y nítida. Su lente tiene de enfoque fijo que ofrece un ángulo de visión de 90° y rotación de 360°.
Cerradura Electromagnética	El electroimán se activa cuando se aplica energía eléctrica, lo que permite la atracción de la contra-placa y el cierre de la puerta.

Tabla 4. Características del software

Material	**Características**
Phyton	Lenguaje flexible que admite programación imperativa, orientada a objetos y funcional.
Redes neuronales	modelo de creación cuyo sistema se basa en el funcionamiento del cerebro humano.

2.2 Métodos

2.2.1 Modalidad de la investigación

El presente proyecto se define como una investigación aplicada, ya que se desarrolló e implementó un sistema de control de acceso y monitoreo inteligente para talleres tecnológicos, utilizando los conocimientos adquiridos en la carrera de Telecomunicaciones, así como la integración de sistemas, la programación y el desarrollo de interfaces para gestionar la seguridad e ingreso al edificio.

La investigación fue bibliográfica, pues se sustentó en la recopilación de información de revistas técnicas, libros, artículos científicos y bases de datos sobre sistemas de seguridad, control de acceso, monitoreo inteligente y sistemas de información.

Además, se llevó a cabo una investigación de campo que permitió validar el funcionamiento del sistema en un entorno real. Esta fase incluyo la implementación del sistema en los talleres tecnológicos de la FISEI, donde se reconoció datos sobre su desempeño en situaciones cotidianas.

Se considera una investigación experimental porque se desarrolló un sistema funcional del sistema de control de acceso y monitoreo. Este sistema fue evaluado y comparado con los estándares de seguridad y los requerimientos específicos de talleres, analizando la eficiencia, la precisión y la usabilidad del sistema.

2.2.2 Recolección de información

Para recolectar la información necesaria para el desarrollo del proyecto, se utilizó la técnica documental, basándose principalmente en tesis, artículos científicos, libros y fuentes bibliográficas que abordan temas relacionados con la seguridad y el control en entornos universitarios, la gestión de estudiantes, la administración de docentes y estudiantes, la implementación de sistemas inteligentes, el control de acceso y el monitoreo en tiempo real, esta información se encuentra respondiendo ciertas preguntas detalladas en la Tabla 4. Que se detalla a continuación.

Tabla 5. Criterio de recolección de información

Criterio	Descripción
¿Para qué?	Se determinó el estado actual del flujo de estudiantes y docentes en talleres tecnológicos.
¿De qué personas u objetos?	De todos los estudiantes y profesores que ingresan a talleres.
¿Cómo?	Realizando un monitoreo en tiempo real del flujo de personas para dar el acceso a través de una puerta automática.
¿Con qué?	Utilizando un sistema inteligente que integra cámaras de vigilancia, reconocimiento facial, sistemas de control de acceso y cerraduras eléctricas, en conjunto con una plataforma centralizada de análisis de datos.
¿Cuándo?	Durante el período académico actual de septiembre 2024 a febrero 2025, con la posibilidad de ampliar el estudio a diferentes períodos del año para analizar patrones de comportamiento y flujo de personas.
¿Dónde?	Talleres tecnológicos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

2.2.3 Procesamiento y análisis de datos

Los datos recopilados durante la investigación fueron sometidos a un proceso de análisis exhaustivo con el fin de obtener resultados claros y precisos, siguiendo las siguientes etapas metodológicas:

- Revisión de la información recopilada sobre tecnologías de monitoreo y control.
- Compendio de información acerca de los dispositivos electrónicos utilizados para el control y monitoreo de estudiantes.
- Estudio del entorno donde se implementa el sistema de control y monitoreo, analizando el área.
- Pruebas del sistema de control y monitoreo en tiempo real para evaluación.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Descripción de la finalidad de la propuesta.

El Sistema de Control de Acceso Automatizado para estudiantes y docentes en los talleres tecnológicos de la FISEI fue diseñado para proporcionar una solución eficiente y segura para gestionar el flujo de personas autorizadas. La implementación de este sistema integra tecnologías de reconocimiento facial basado en inteligencia artificial (IA) y dispositivos de control de acceso físico como cerraduras electrónicas, permitiendo un monitoreo en tiempo real de las entradas y salidas.

Este proyecto busca resolver problemas tradicionales de seguridad, optimizando la gestión de los accesos mediante el uso de cámaras de alta definición (HD) como es la Genius QCam 6000 con su sensor de alta definición CMOS de 2 megapíxeles, lo que permite una resolución de imagen de hasta 1920 x 1080 píxeles. Además, cuenta con una resolución de video que incluye CIF/VGA hasta 30 fps, 720p HD hasta 30 fps y 1080P hasta 30 fps. y algoritmos avanzados de IA que reconocen a los usuarios registrados en la base de datos con redes neuronales convolucionales (CNN). Cuando una persona se acerca a la puerta de los talleres, el sistema captura su imagen, la procesa en tiempo real y, si se confirma su autorización, activa la cerradura eléctrica para permitir el acceso. Además, este sistema registra la información de cada acceso, almacenando detalles como el nombre, rol (docente o estudiante), hora y fecha del ingreso en una base de datos que se llama Firebase que puede ser consultada para la generación de reportes o auditorías.

El desarrollo de este sistema garantiza que solo las personas autorizadas puedan ingresar a los talleres, previniendo accesos no deseados y facilitando un control detallado de la utilización de las instalaciones.

3.2 Desarrollo de la propuesta

El proyecto desarrollado fue la implementación de un sistema de control de acceso automatizado con inteligencia artificial para monitorear el ingreso tanto de estudiantes como de docentes a los talleres tecnológicos de la FISEI, con este sistema se permite la gestión eficiente y segura de los accesos mediante el uso de una cámara y las notificaciones en tiempo real así también el almacenamiento de los datos. Con la cerradura eléctrica se tiene un mejor control ya que se tiene el control automático de quien está autorizado a ingresar.

3.2.1 Requerimientos del Sistema

Para el desarrollo del proyecto de investigación se identificaron los siguientes requerimientos de funcionalidad para el sistema de control de acceso automatizado con inteligencia artificial, adaptado a las necesidades de los Talleres Tecnológicos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial (FISEI). Estos requerimientos garantizan un control eficiente, seguro y confiable.

A continuación, se describen los aspectos de funcionalidad con los cuales debe constar el sistema como requerimientos:

- Captura y procesamiento de rostros: El sistema debe ser capaz de capturar imágenes faciales en tiempo real. Procesar la imagen y compararla con una base de datos de rostros previamente registrados.

- Gestión de la base de datos: Incluir funcionalidades para registrar, actualizar y eliminar usuarios autorizados, los cuales son docentes y alumnos de la facultad. Además, almacenar datos relevantes como nombre, cédula, entre otros, que permita tener un mejor control y monitoreo de ingreso.

- Autorización de acceso: Validar si el usuario tiene permisos de ingreso, para posterior activar la cerradura eléctrica si el usuario es autorizado.

- Notificaciones y alertas: Enviar alertas al personal encargado cuando un usuario no esté registrado en la base de datos, teniendo así un control de seguridad.

- Indicadores visuales: Se requiere que el sistema tenga indicadores para que el usuario pueda reconocer el acceso permitido o acceso denegado.

3.2.2 Esquema General del Sistema

El sistema sigue una arquitectura distribuida que combina hardware especializado para el control de acceso y procesamiento de imágenes, así como software para la administración y control del sistema. El esquema general del sistema visualiza en la Figura 4 que se describe a continuación:

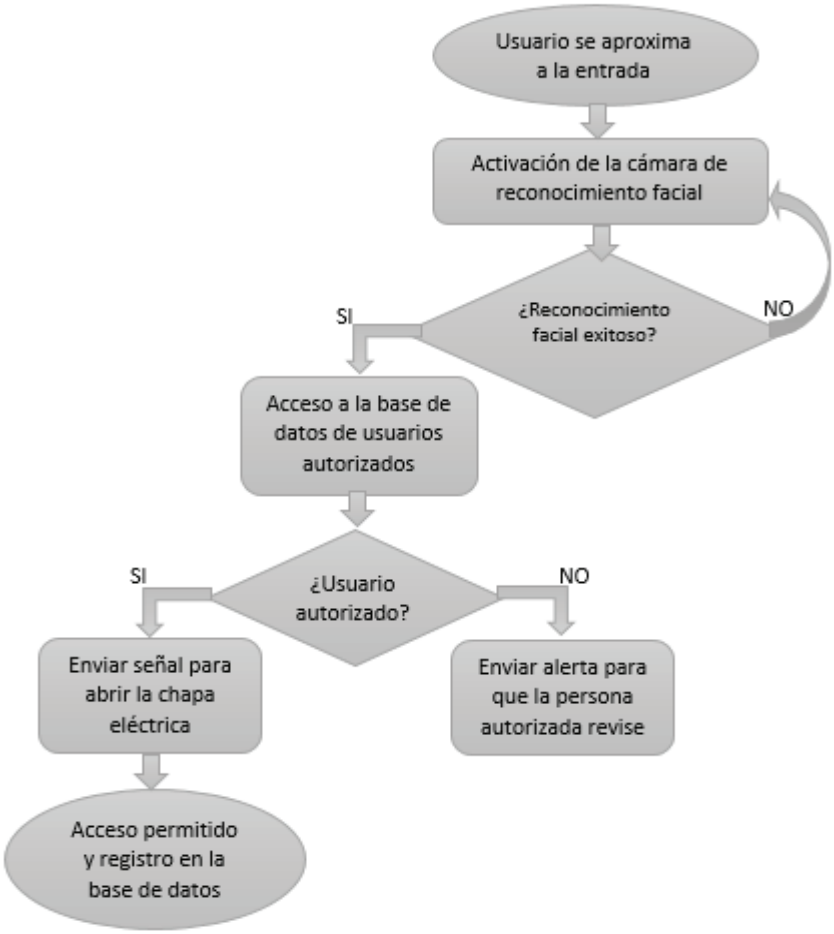


Figura 4. Esquema general

- Detección y Captura de Imagen: El sistema de cámaras está activo en todo momento, monitoreando las zonas de acceso. Cuando se detecta la presencia de una persona en la puerta de entrada, la cámara web captura una imagen del rostro.
- Procesamiento de Imagen y Verificación: La imagen capturada es enviada al servidor donde los algoritmos de IA la procesan. El sistema utiliza redes neuronales entrenadas en reconocimiento facial para comparar el rostro capturado con las imágenes almacenadas en la base de datos.
- Desbloqueo Automático de la Cerradura: Si el rostro coincide con uno de los usuarios autorizados en la base de datos, el sistema envía una señal al controlador de la cerradura electrónica, desbloqueándola y permitiendo el acceso. Si no se detecta coincidencia, el acceso es denegado.
- Registro y Monitoreo: Todos los eventos de acceso (tanto autorizados como denegados) son registrados automáticamente en la base de datos Firebase. La información registrada incluye el nombre del usuario, la hora, fecha y el taller al que se accedió. Estos registros están disponibles para consulta a través de la interfaz de administración.

a. Diagrama Electrónico del reconocimiento facial

A continuación, se presenta en la Figura 5 el esquema eléctrico del reconocimiento facial, donde cada paso se lo detalla más adelante.

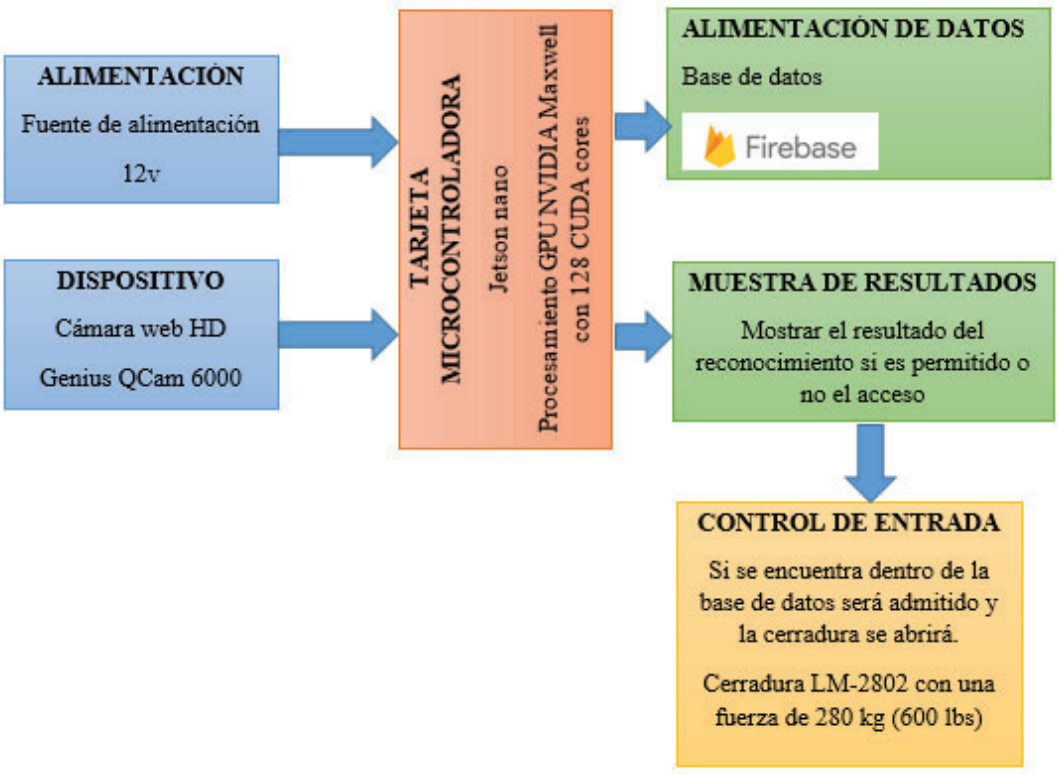


Figura 5. Esquema electrónico

Componentes del sistema:

- Alimentación: Fuente de alimentación de 12v para proporcionar energía a los componentes del sistema.
    - Dispositivo: Una cámara web HD 1080P que captura imágenes para la identificación facial.
- Tarjeta Microcontroladora (NVIDIA Jetson Nano): Es el núcleo del sistema, donde se procesan los datos recibidos del dispositivo y se gestionan las siguientes etapas.
- Alimentación de datos: Conexión a una base de datos que contiene la información necesaria para el reconocimiento facial, como los registros de usuarios autorizados.
- Muestra de resultados: Indica una interfaz para mostrar si el acceso está permitido o denegado, basado en la comparación con la base de datos.
- Control de entrada: Un mecanismo de control, la cerradura eléctrica que permite o niega el acceso según el resultado del reconocimiento facial.

b. Flujo del sistema

La tarjeta microcontroladora recibe alimentación, datos desde la cámara que son las imágenes y la base de datos se abre para comparar la información para la identificación. Cada uno de los datos es procesado para verificar si la persona está registrada en la base de datos, según el resultado del procesamiento se muestran mediante indicadores siendo asi su activación en caso de que este en la base de datos es de color verde y si no es encontrado es rojo, la apertura de la puerta dependerá también de esto abriendo la cerradura eléctrica.

El sistema está diseñado para integrar hardware, cámara y cerradura con inteligencia artificial, reconocimiento facial en la NVIDIA Jetson Nano, garantizando un control de acceso automatizado y seguro.

c. Diagrama del procesamiento de IA para capturar la imagen

En la Figura 6, se detalla el proceso que toma desde la captura de imágenes del rostro, la extracción de las características en matrices almacenada en base de datos para posterior ser comparada con los registros y otorgar o negar el acceso.

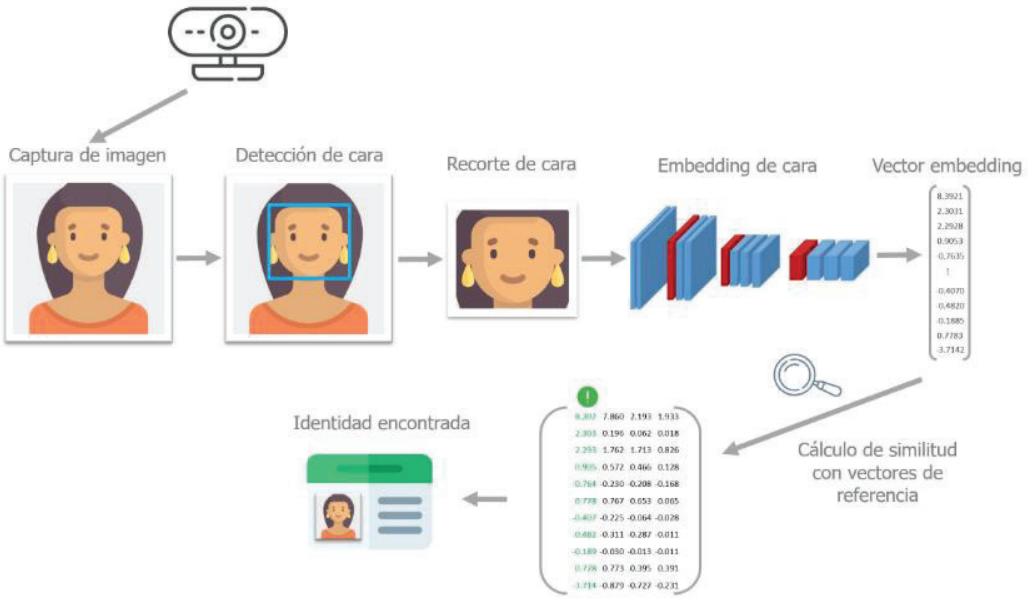


Figura 6. Diagrama de procesamiento de IA de la imagen

- **Captura de imagen**

Se delimita información por nombre, cedula, carrera y facultad.

El sistema procede a tomar capturas de las personas, en este caso se puso 30 imágenes.

Se procede a guardar la información, esta información se guarda en la carpeta del dataset, misma que genera los documentos respectivos, tales como; carpeta con nombre del usuario con las imágenes, y guardar los datos de la persona en el diccionario people_data.

- **Creación del procesamiento de imágenes**

Lo que se hace es tomar de la base de dataset, las imágenes donde también se crea una base de datos con las imágenes y la información de cada persona que hemos tomado, si todo es correcto los datos se guardaran, y quedara ya en la memoria.

Al final se procesa cada imagen dando la extracción de características del rostro en las matrices vectoriales, lo que se denomina como (embedding) y al final la información

adicional, dependiendo del procesamiento de la computadora se toma un determinado tiempo.

- **Prueba del reconocimiento facial**

  Tendrá la comprobación de los datos almacenados en el cual se observa si la persona es identificada o es un desconocido.

3.2.3 Selección de componentes y dispositivos

La selección de componentes se realizó considerando criterios de rendimiento, costo, y compatibilidad, siendo los siguientes:

a. Cámara de alta definición (HD)

En un sistema de control de acceso mediante reconocimiento facial, la cámara es un componente crucial para capturar las imágenes o videos que serán procesados por el microprocesador para realizar la comparación y autenticación. La elección de la cámara depende de factores como la resolución, la calidad de la imagen, la compatibilidad con el sistema y el tipo de iluminación en el entorno. A continuación, se presenta en la Tabla 6 una comparación entre varias cámaras comunes en proyectos de reconocimiento facial.

Tabla 6. Comparativa de cámara web

Características	 **Genius QCam 6000 Webcam**	 **Logitech C920 Pro HD**	 **Microsoft LifeCam Studio**
Resolución	640 x 480 (VGA)	1920 x 1080 (Full HD)	1920 x 1080 (Full HD)
Tipo de lente	Lente de enfoque fijo	Lente de enfoque automático	Lente de enfoque automático

Velocidad de fotogramas	30 fps	30 fps	30 fps
Compatibilidad con software	Compatible con OpenCV y software de terceros	Compatible con OpenCV y software de terceros	Compatible con OpenCV y software de terceros
Iluminación en baja luz	No optimizada para baja luz	Buena para condiciones de poca luz	Buena para condiciones de poca luz
Conexión	USB 2.0	USB 2.0	USB 2.0
Ángulo de visión	60°	78°	75°
Precio	Bajo	Medio	Medio

La Genius QCam 6000 Webcam ha sido seleccionada debido a su relación costo-beneficio en proyectos de reconocimiento facial básicos. Aunque su resolución es de 640 x 480, suficiente para aplicaciones de reconocimiento facial en entornos controlados, su bajo precio la hace atractiva para proyectos de bajo presupuesto. Si bien no tiene características avanzadas de captura en baja luz, su compatibilidad con software de terceros como OpenCV permite la implementación de algoritmos de reconocimiento facial sin necesidad de hardware adicional. Su ángulo de visión de 60° es adecuado para sistemas donde la distancia y el ángulo de captura no son variables críticas.

Esta cámara es una opción válida para prototipos y proyectos donde no se requiere alta resolución ni procesamiento en tiempo real de modelos complejos de inteligencia artificial. Para más detalle de las características técnicas se puede observar en el Anexo D.

b. Algoritmos de reconocimiento facial:

Se implementaron redes neuronales convolucionales (CNN) especializadas en el reconocimiento facial, con un modelo previamente entrenado en grandes conjuntos de datos. Estos algoritmos detectan y reconocen los rostros de los usuarios en tiempo real, comparándolos con los perfiles almacenados en la base de datos como se detalla en la Figura 7.

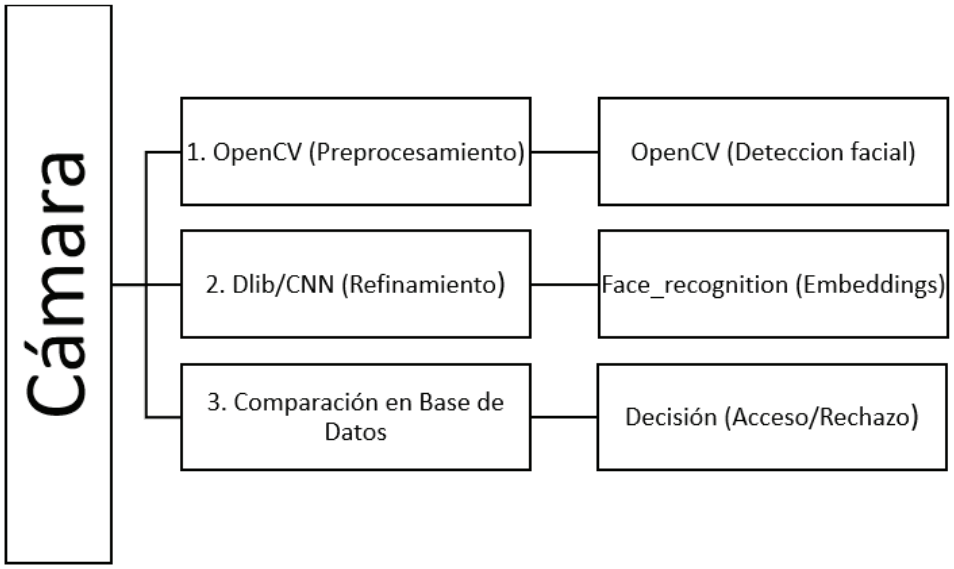


Figura 7. Algortimos de reconocmiento facial y su proceso.

- ***OpenCV (Preprocesamiento):*** Se procesan las imágenes con OpenCV, realizando tareas como la conversión a escala de grises, ajuste de brillo/contraste, eliminación de ruido, etc.

- ***OpenCV (Detección facial):*** Utilizando métodos como los clasificadores en cascada de Haar o los HOG (Histogram of Oriented Gradients), se detectan las regiones faciales dentro de la imagen.

- ***Dlib/CNN (Refinamiento):*** Dlib y la red neuronal convolucional (CNN) se usan para realizar un refinamiento de la detección facial, mejorando la precisión y la localización de las caras.

- ***Face_recognition (Embeddings):*** Una vez que las caras están detectadas y localizadas, se extraen características faciales mediante el modelo Face_recognition, generando "embeddings" (vectores de características) que representan las caras de forma numérica.

- ***Comparación en Base de Datos:*** Los embeddings generados se comparan con una base de datos previamente almacenada que contiene los embeddings de caras conocidas para encontrar coincidencias.

- ***Decisión (Acceso/Rechazo):*** Si se encuentra una coincidencia suficiente en la base de datos, se otorga acceso al usuario, de lo contrario, se rechaza.

c. Microcontroladores

El microcontrolador en un sistema que tiene el control de dar acceso mediante reconocimiento facial es el componente central encargado de coordinar, procesar y gestionar las diferentes tareas que permiten la funcionalidad del sistema implementado. Este dispositivo electrónico integra hardware y software para ejecutar instrucciones y manejar las interacciones entre los sensores, módulos de comunicación, y mecanismos de control. En la Tabla 7 se realiza una comparativa de estos dispositivos de los más utilizados en el desarrollo de proyectos y prototipos.

Tabla 7. Microcontroladores

Características	 **Jetson Nano**	 **Raspberry Pi**	 **Arduino mega 2560**
Procesador	Quad-core ARM Cortex-A57 a 1.43 GHz	Quad-core ARM Cortex-A72 a 1.5 GHz	Microcontrolador ATmega2560 (8 bits, 16 MHz)
Unidad de Procesamiento	GPU NVIDIA Maxwell con 128 CUDA cores	GPU Broadcom VideoCore VI	No tiene GPU
Memoria RAM	4 GB LPDDR4	2 GB, 4 GB o 8 GB LPDDR4	8 KB de SRAM
Compatibilidad con AI	Diseñado para IA, alta compatibilidad con CNN	Compatible, pero sin optimización para IA	No soporta directamente frameworks de IA
Capacidad para Reconocimiento Facial	Excelente rendimiento para modelos complejos y en tiempo real (OpenCV, TensorRT)	Adecuado para proyectos básicos con OpenCV.	Muy limitado, no apto para tareas de IA.
Sistema Operativo	Ubuntu basado en JetPack (Linux optimizado para IA)	Raspberry Pi OS (Linux), otros SO ligeros	Sin sistema operativo (corre firmware).
Interfaces	HDMI, USB 3.0, GPIO, CSI para cámaras	HDMI, USB 3.0, GPIO, CSI para cámaras	GPIO, pero sin soporte para cámaras avanzadas

Facilidad de Implementación	Alta, diseñada para tareas de visión por computadora	Moderada, requiere optimización manual para IA.	Baja, requiere hardware externo para procesamiento.

Se selecciona la Jetson Nano ya que está equipada con una GPU NVIDIA Maxwell de 128 núcleos CUDA, diseñada para acelerar operaciones de inteligencia artificial (IA) y visión de computadora, incluida la inferencia de modelos CNN. Permitiendo así implementar un sistema de reconocimiento facial eficiente. Para más detalles se encuentra en el Anexo C.

d. Cerraduras electrónicas

La fuerza del electroimán es un factor crítico, ya que determina la resistencia de la cerradura frente a intentos de apertura no autorizados. Esta cerradura es comúnmente utilizada en sistemas de control de acceso debido a su confiabilidad, facilidad de integración con sistemas electrónicos y la capacidad de liberarse mediante señales eléctricas. Dependiendo de la fuerza del electroimán, las electrocerraduras pueden ser de diferentes tipos como se tiene en la Tabla 8 y se seleccionan según las necesidades específicas de seguridad de cada instalación, en nuestro caso es en la puerta de vidrio de Talleres Tecnológicos.

Tabla 8. Comparativa de cerradura eléctrica

Modelo de Cerradura	Fuerza del Electroimán	Tipo de Instalación	Características	Aplicaciones Comunes
 **LM-2802**	280 kg (600 lbs)	Superficial o empotrada	Fuerza moderada, tamaño compacto, fácil instalación.	Puertas de acceso estándar, oficinas, hogares

 **LM-2500**	250 kg (550 lbs)	Empotrada	Cerradura con alta resistencia, especialmente para puertas pesadas.	Puertas blindadas, sistemas de alta seguridad.
 **Yale 1200**	120 kg (265 lbs)	Superficial	Cerradura de bajo costo, adecuada para instalaciones ligeras.	Puertas de acceso en entornos de bajo riesgo.
 **SECO-LARM E-9300**	400 kg (880 lbs)	Superficial o empotrada	Alta resistencia, ideal para puertas grandes y de alta seguridad.	Entradas principales de edificios, cajeros automáticos.

Se ha seleccionado la LM-2802 debido a su balance óptimo entre fuerza y tamaño. Con una fuerza de 280 kg (600 lbs), ofrece una resistencia adecuada para puertas de acceso estándar en entornos controlados, como oficinas y hogares, sin ser excesiva para el tipo de puerta en cuestión. Además, su tamaño compacto facilita su instalación en una variedad de entornos sin requerir modificaciones significativas, ayudó también en la puerta de vidrio del edificio. Para conocer más sobre sus características están en el Anexo B.

e. Base de datos:

La base de datos centralizada almacena la información de los usuarios, incluyendo imágenes de referencia de sus rostros y otros detalles como nombre, número de identificación, y rol dentro de la institución (docente o estudiante). También se almacena cada evento de acceso, incluyendo la hora, fecha y la cámara que detectó el ingreso. La base de datos está diseñada utilizando tecnología SQL para permitir búsquedas rápidas y eficientes como se muestra en la Tabla 9, donde se muestra la comparativa de algunas bases de datos.

Tabla 9. Comparativa de Bases de datos.

Característica	Firebase	MySQL	PostgreSQL	MongoDB
Modelo de datos	NoSQL (documentos y JSON)	SQL (relacional)	SQL/JSONB (híbrido relacional)	NoSQL (documentos)
Actualización en tiempo real	Sí, nativo y altamente optimizado	No, requiere soluciones externas	No, requiere soluciones externas	Sí, pero depende del stack usado
Facilidad de integración	Alta, con SDKs para plataformas móviles y web	Media, necesita configuración manual	Media, configuración requerida	Media, requiere configuración extra
Escalabilidad automática	Sí, completamente gestionada	Limitada, requiere ajustes manuales	Limitada, requiere ajustes manuales	Sí, depende del proveedor
Compatibilidad con IA/ML	Alta, se integra con Google ML Kit y TensorFlow	Baja, requiere herramientas externas	Media, puede integrarse con herramientas externas	Media, requiere configuración adicional
Autenticación integrada	Sí, con Firebase Authentication	No, requiere herramientas externas	No, requiere herramientas externas	No, requiere herramientas externas
Soporte para datos offline	Sí, almacenamiento en caché integrado	No, necesita implementación personalizada	No, necesita implementación personalizada	Sí, pero requiere configuración
Seguridad	Reglas de seguridad configurables en tiempo real	Basada en SQL y configuración manual	Basada en SQL y configuración manual	Basada en roles y configuraciones manual
Costos iniciales	Gratis hasta ciertos límites (plan escalable)	Generalmente gratis, depende del servidor	Generalmente gratis, depende del servidor	Generalmente gratis, depende del servidor
Facilidad de uso	Alta, ideal para desarrolladores sin experiencia avanzada	Media, requiere conocimientos de SQL	Media, requiere conocimientos avanzados	Media, conocimientos de NoSQL necesarios

En base a la comparativa, se decide utilizar como base de datos para el sistema de control de acceso mediante reconocimiento facial a Firebase, por sus amplias características, en la Tabla 10 se detalla los beneficios en los diferentes ascpectos que contribuyen al sistema de acceso mediante reconocimiento facial.

Tabla 10. Beneficios de Firebase para el sistema de acceso po reconocimiento facial.

Característica	Descripción	Beneficio para el sistema
Firebase Authentication	Gestión segura de usuarios con soporte para correo, redes sociales, y autenticación personalizada.	Permite registrar y autenticar usuarios de manera eficiente y segura.
Firestore Database	Base de datos NoSQL en tiempo real y escalable.	Almacena perfiles de usuarios y registros de acceso.
Firebase Storage	Almacenamiento en la nube para imágenes y archivos.	Ideal para guardar fotos o datos capturados para reconocimiento facial.
Firebase Hosting	Hosting seguro y rápido para aplicaciones web.	Permite implementar la interfaz web del sistema de gestión de acceso.
Firebase Functions	Ejecución de código backend sin servidores ("serverless").	Procesamiento de lógica personalizada, como validación de reconocimiento facial.
Cloud Messaging	Servicio de notificaciones push.	Envía alertas en tiempo real sobre intentos de acceso o anomalías detectadas.
Real-time Database	Base de datos en tiempo real para sincronización inmediata.	Sincronización instantánea entre dispositivos, como actualizaciones de acceso.
Analytics	Seguimiento de eventos de usuarios y métricas de uso.	Proporciona estadísticas sobre el uso del sistema y comportamiento de acceso.
Scalabilidad y alta disponibilidad	Infraestructura diseñada para manejar múltiples usuarios y grandes volúmenes de datos.	Garantiza un rendimiento óptimo incluso en sistemas con muchos usuarios simultáneos.
Seguridad integrada	Reglas de acceso y autenticación sólidas para datos y servicios.	Asegura que solo usuarios autorizados puedan acceder al sistema o sus datos.

3.2.4 Diseño y configuración de la programación del reconocimiento

El sistema propuesto para la gestión de control de acceso mediante reconocimiento facial se ha desarrollado utilizando una serie de tecnologías modernas y herramientas de bajo costo, como la Jetson Nano y diversas bibliotecas de Python para el procesamiento de imágenes y la comunicación en la nube. La metodología descrita a continuación proporciona una visión detallada de los procesos involucrados, desde la adquisición y procesamiento de imágenes hasta el almacenamiento y la actualización periódica de las codificaciones faciales. Se hace énfasis en las técnicas de detección y reconocimiento facial, así como en el uso de Firebase para la gestión de usuarios y en la integración de notificaciones a través de Telegram

a. Primera Fase

Registro de Usuarios

La primera fase del sistema consiste en registrar usuarios en la base de datos a través de una interfaz gráfica interactiva y amigable. Este módulo incluye la captura de imágenes faciales, la generación de codificaciones faciales y el almacenamiento seguro de los datos en una base de datos Firestore (Firebase). En la Figura 8, se detallan los componentes y el flujo de trabajo.

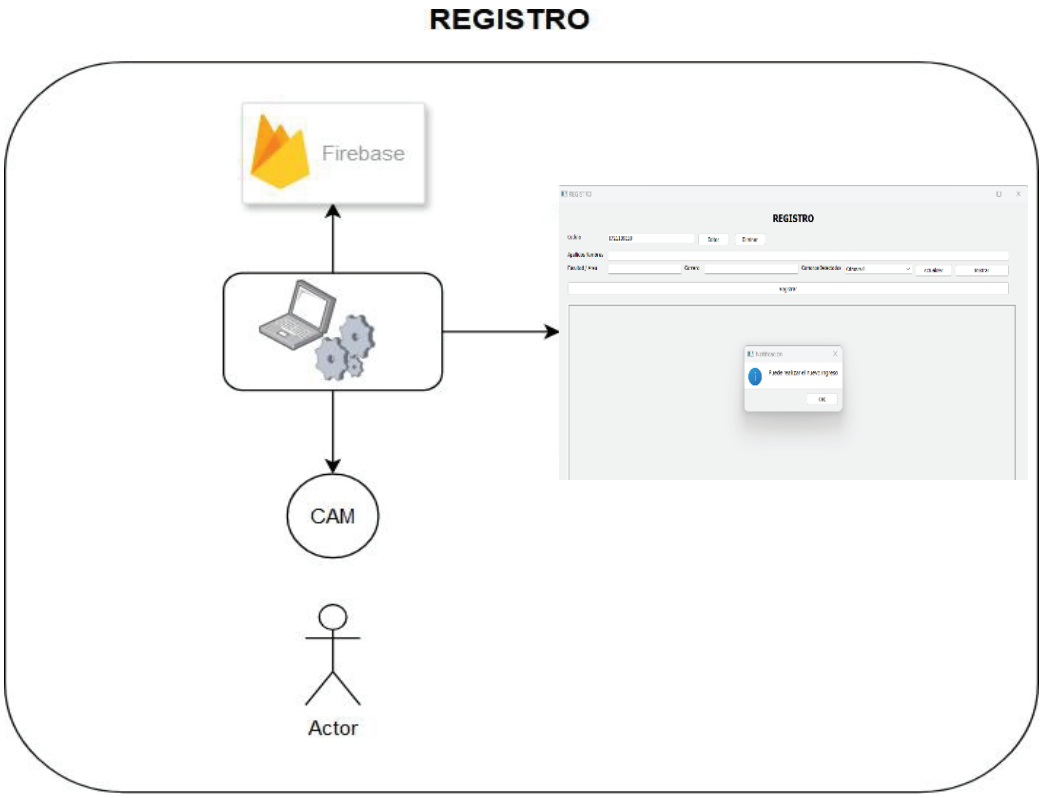


Figura 8. Registro de usuarios.

En la Figura 9 detalla el flujo de trabajo del registro de un nuevo usuario, el sistema realiza una captura de la imagen facial, la cual es procesada mediante una codificación biométrica para generar un perfil único basado en sus características faciales. Esta codificación se guarda en un conjunto de datos y, posteriormente, en el sistema, permitiendo que el usuario pueda ser identificado en futuros accesos mediante reconocimiento facial. El flujo también permite la edición de usuarios existentes, realizando una nueva captura facial para actualizar la información previamente almacenada, o bien, eliminar usuarios y sus datos del sistema, manteniendo el control de acceso a mí.

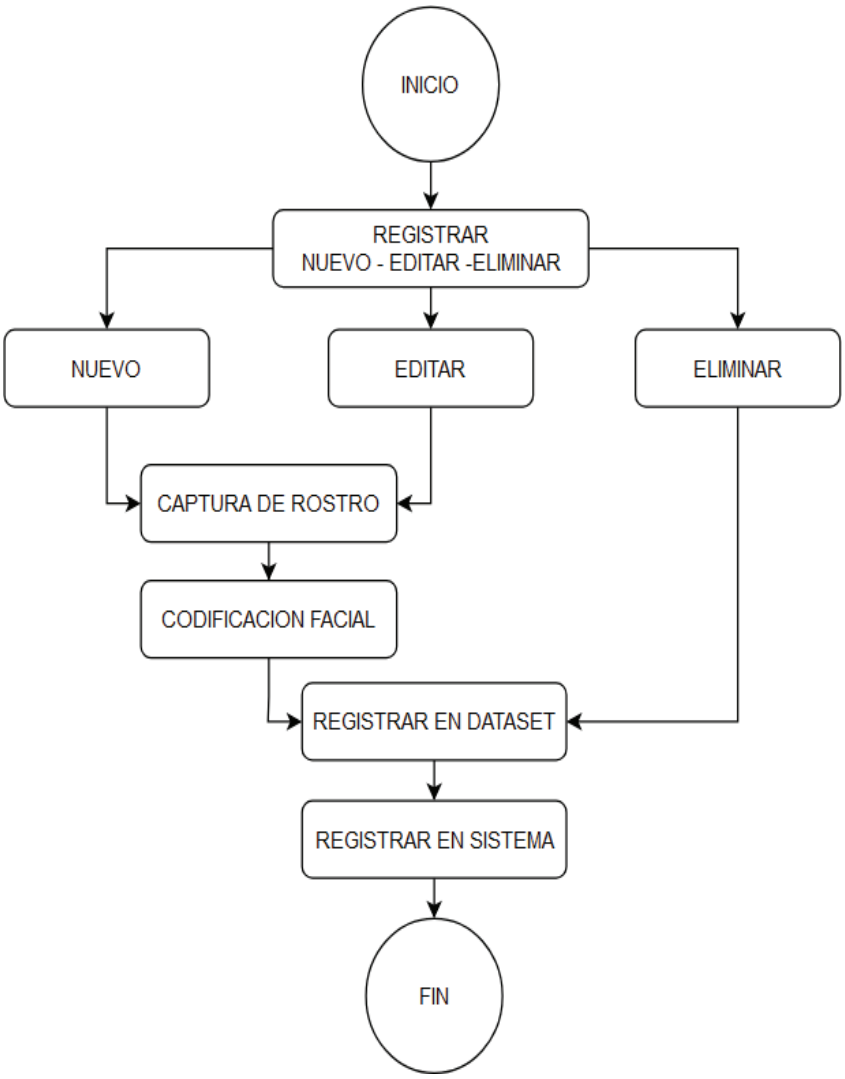


Figura 9. Flujo de trabajo del registro de usuario.

Interfaz Gráfica del Registro

La interfaz gráfica ha sido desarrollada utilizando PyQt5 para facilitar el registro de usuarios. Esta interfaz incluye campos para ingresar los datos personales básicos: nombre completo, cédula, carrera y facultad. Además, incorpora botones funcionales para iniciar la cámara, registrar nuevos usuarios, editar o eliminar datos existentes, y una lista desplegable que permite seleccionar la cámara conectada al sistema.

- Interactividad: Los usuarios pueden ingresar su información a través de campos de texto validados. Además, el sistema valida automáticamente si la cédula ingresada ya existe en la base de datos como se muestra en las Figuras 10 y 11 respectivamente.

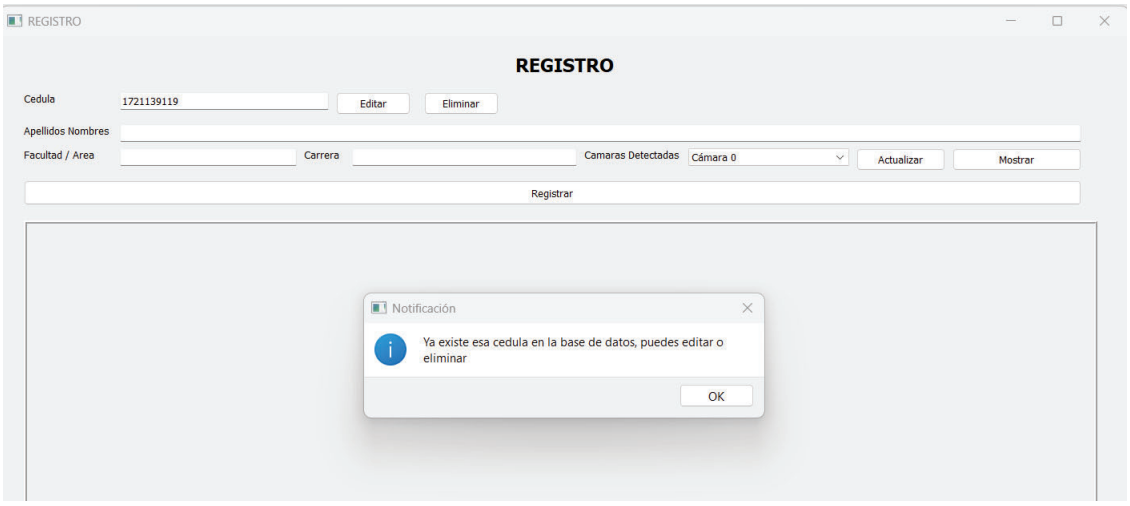


Figura 10. Control de registro cuando el usuario ya existe en el sitema.

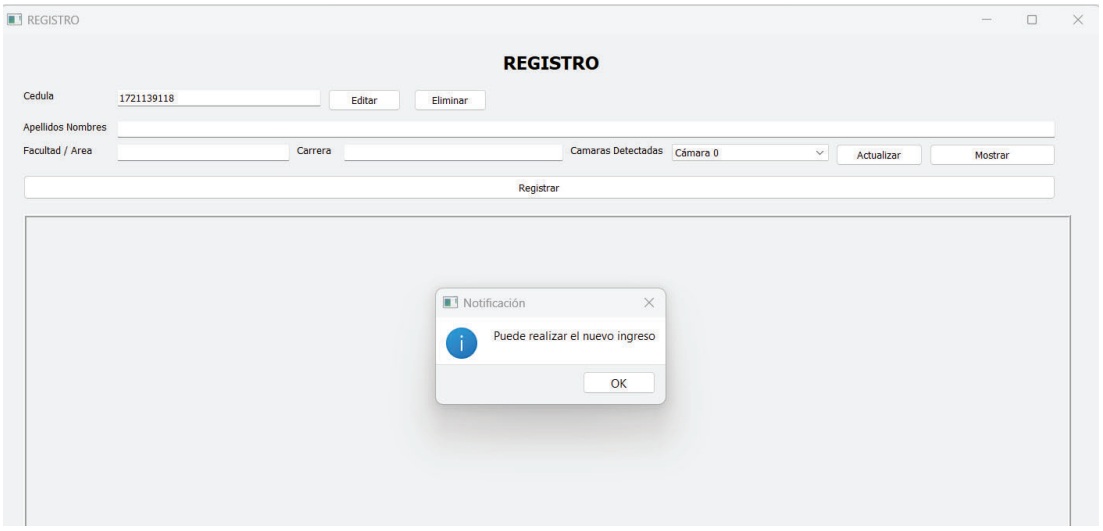


Figura 11. Validación para un nuevo registro.

- **Notificaciones:** Se utilizan cuadros de diálogo (QMessageBox) para informar al usuario sobre el progreso del registro y errores, como credenciales incorrectas o valores inválidos como se observa en la Figura 12.



Figura 12. Interfaz de registro.

Captura de Imágenes y Procesamiento

El módulo de captura utiliza la librería OpenCV para manejar la cámara y face_recognition para la detección de rostros. Este módulo tiene como objetivo capturar imágenes de alta calidad de un usuario registrado y detectar su rostro en cada cuadro para generar las codificaciones faciales.

Inicialización de la Cámara: El sistema permite al usuario seleccionar la cámara mediante un índice (por defecto, se utiliza la primera cámara conectada). Una vez seleccionada, el módulo inicia un hilo separado para capturar continuamente los cuadros desde la cámara como se observa en la Figura 13.

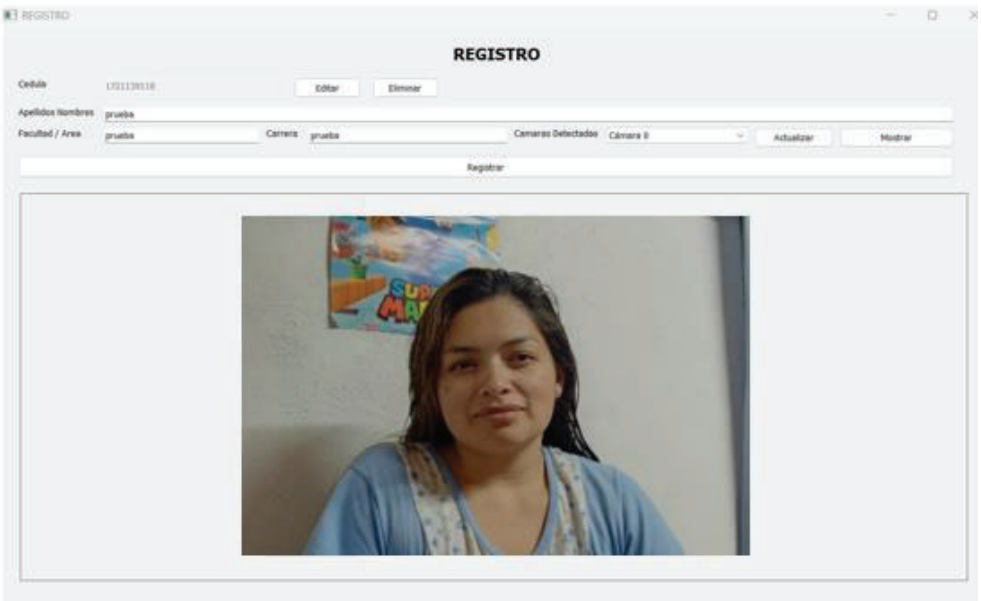


Figura 13. Captura de imágenes del usuario.

- Detección Facial:
    - Se aplica el modelo de detección facial basado en HOG o CNN proporcionado por la biblioteca face_recognition.
    - Solo se permite capturar imágenes si un único rostro es detectado en el cuadro. Esto asegura que las codificaciones faciales no incluyan información incorrecta.

Como se muestra en la Figura 14, donde se tiene usuarios administrativos para el registro y en la Figura 15 el número de capturas de que se delimita para procesar.

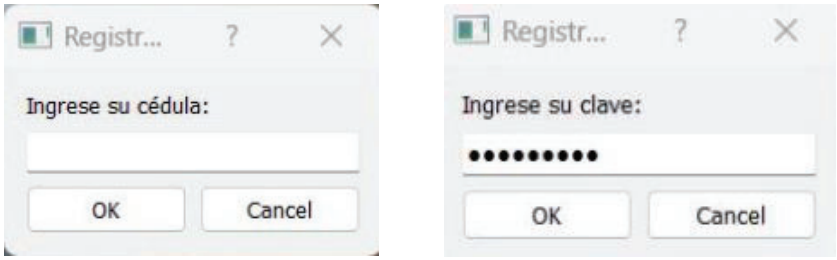


Figura 14. Seguridad de aceeso. al sistema de registro.

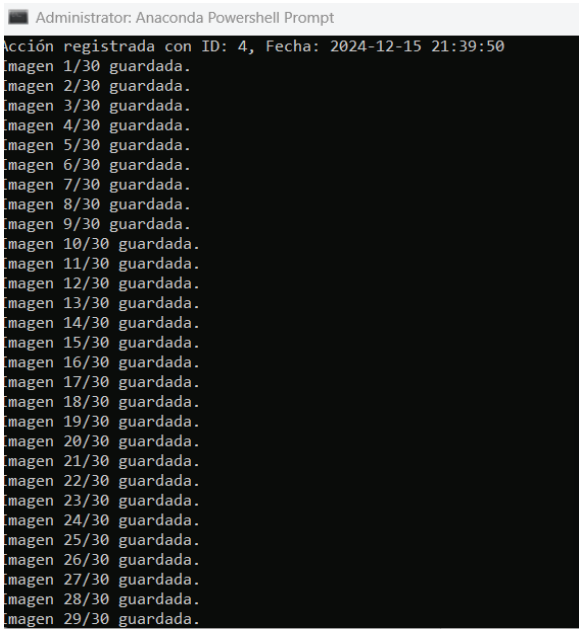


Figura 15. Visualización del número de toma de capturas.

Almacenamiento Temporal: Las imágenes capturadas son almacenadas temporalmente en un directorio nombrado según la cédula del usuario, siguiendo la estructura mostrada en la Figura 16.

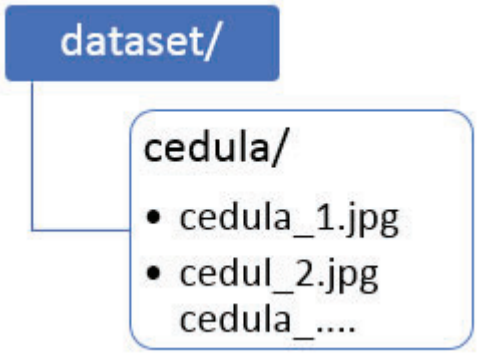


Figura 16. Estructura de almacenamiento en dataset.

En la computadora se crea una carpeta dataset como podemos ver en la Figura 17.

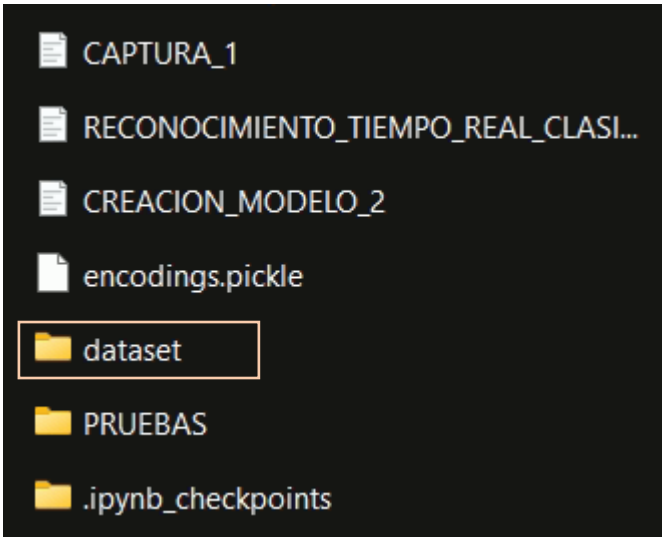


Figura 17. dataset carpeta donde esta almacenada las imágenes

Generación de Codificaciones Faciales

Una vez capturadas las imágenes, se procesan para generar las codificaciones faciales, que se utilizan posteriormente para el reconocimiento. Este proceso incluye:

- Carga de Imágenes: Las imágenes almacenadas en el directorio temporal son cargadas utilizando cv2.imread().
- Detección y Codificación:
    - Se utiliza el modelo CNN de face_recognition para localizar las características del rostro en cada imagen.
    - Las codificaciones faciales (vectores de características) se generan mediante face_recognition.face_en

Almacenamiento en Firebase

Una vez generadas las codificaciones, los datos son estructurados en un formato JSON y almacenados en la base de datos Firestore. Cada usuario tiene un documento único identificado por su cédula. La estructura de los datos almacenados es la siguiente:

```
db.collection('Data_Set').document(self.cedula).set(people_data,
merge=True)
```

En la base de datos se puede observar el embedding de la captura de fotos como es llevado a matrices vectoriales que almacenan la información de las características faciales de manera numérica como muestra la Figura 18.

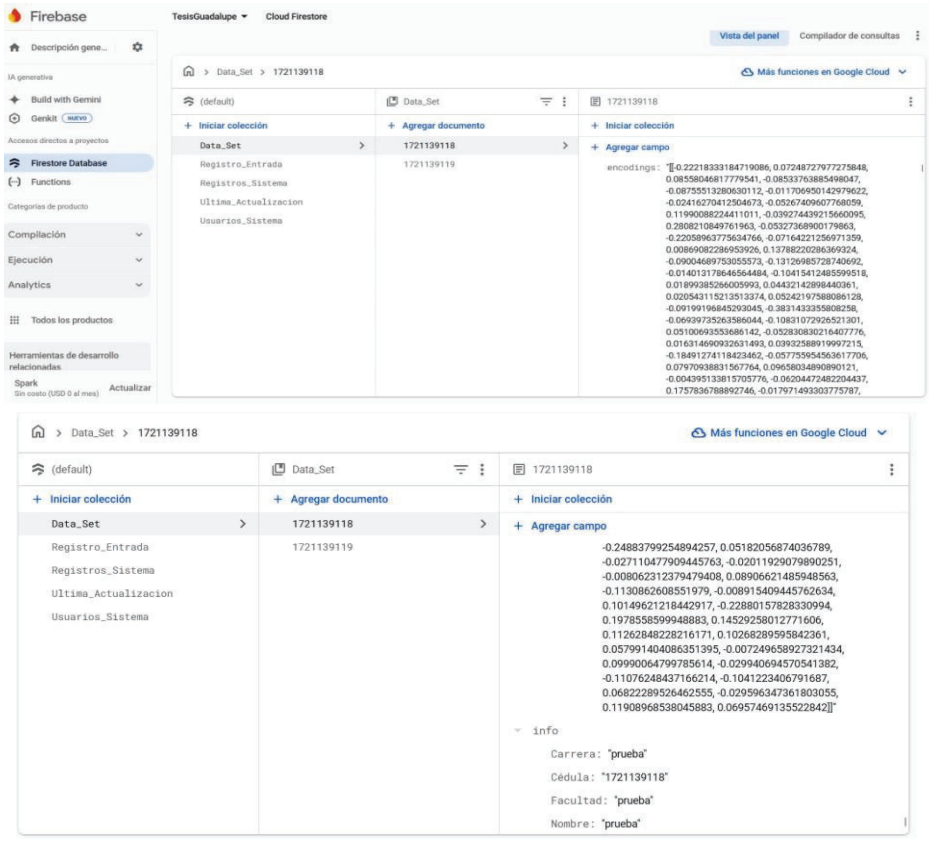


Figura 18. Almacenamiento en Firebase.

Eliminación del Directorio Temporal

Una vez que los datos han sido correctamente almacenados en Firebase, el directorio temporal que contiene las imágenes, es eliminado para garantizar la privacidad de los usuarios y liberar espacio en disco.

```
shutil.rmtree(person_folder)
```

Control de Seguridad

El sistema incorpora autenticación mediante credenciales (cédula y contraseña) para realizar operaciones sensibles, como editar o eliminar registros. Las contraseñas son cifradas utilizando bcrypt y se comparan con los valores almacenados en Firebase para validar su autenticidad.

b. Segunda Fase

La segunda fase del sistema de reconocimiento facial tiene como objetivo integrar la parte de captura y procesamiento de imágenes en tiempo real utilizando una cámara conectada a la Jetson Nano, con la gestión y actualización de datos desde Firebase, así como el envío de notificaciones en Telegram cuando se detectan personas desconocidas. El sistema se basa en una arquitectura de reconocimiento facial en la que se comparan las caras detectadas con un conjunto de codificaciones previamente almacenadas. En la Figura 19 se muestra los elementos involucrados en la fase dos del reconocimiento y detección facial.

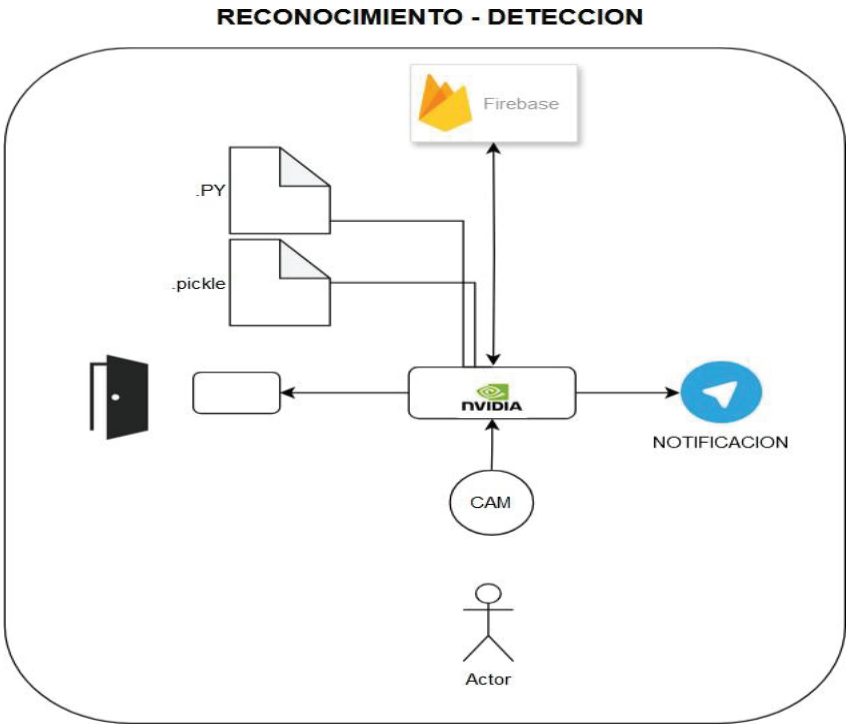


Figura 19. Reconocimiento y detección facial.

En la Figura 20 muestra a detalle el flujo que trabajo que desarrolla al reconocer la imagen y detecatar como un sujeto o individuo registrado para el acceso o caso contrario para el rechazo y posterior notificación de alerta.

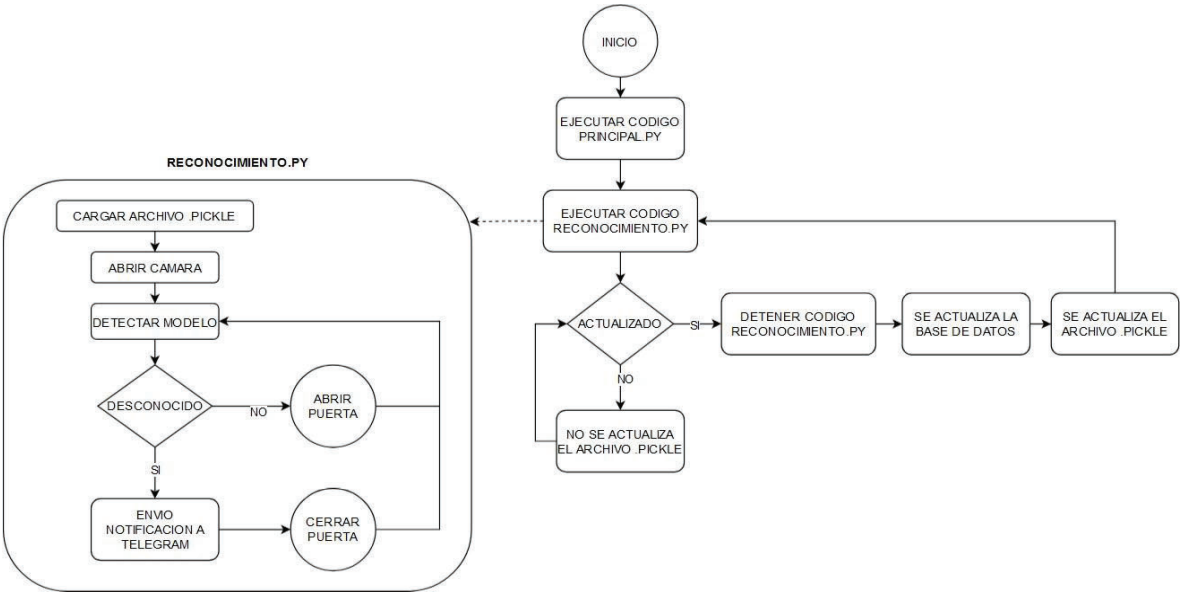


Figura 20. Flujo de de detección y reconocmiento.

Módulo de Gestión de Datos (Firebase) (PRINCIPAL.PY): Este módulo se encarga de la actualización periódica de los datos almacenados en Firebase, específicamente los registros de las personas y los últimos identificadores (ID) detectados, así como la creación de un archivo de datos serializados para el reconocimiento facial.

Módulo de Reconocimiento Facial (RECONOCIMIENTO.PY): Este módulo se encarga de procesar las imágenes de la cámara, realizar la detección de caras y compararlas con las bases de datos de encodings almacenadas en un archivo .pickle. Además, maneja la activación de señales, el envío de notificaciones a Telegram y la gestión de las luces LED según el estado de reconocimiento (conocido o desconocido).

Módulo de Gestión de Datos (Firebase)

Mantener actualizada la base de datos con la información de las personas registradas, actualizar los encodings de las caras, y generar un archivo .pickle con estos datos para su procesamiento en el módulo de reconocimiento facial.

Flujo de Trabajo:

- Conexión con Firebase: Se inicializa la conexión con Firebase utilizando el SDK de Firebase Admin. El proceso comienza con la autenticación mediante

un archivo de credenciales (tesisguadalupe.json) y la conexión a la base de datos Firestore.

- Recuperación de Datos: Se consulta la colección Registros_Sistema para obtener el último ID registrado. Este ID se utiliza para comparar con el campo id del documento Ultima_Actualizacion, lo que permite detectar cambios en los registros. Si el ID registrado en Firestore es distinto al almacenado en el documento de actualización, se procede a actualizar los datos.

En La Figura 21 muestra cómo se visualiza este flujo de trabajo en la base de datos Firebase.

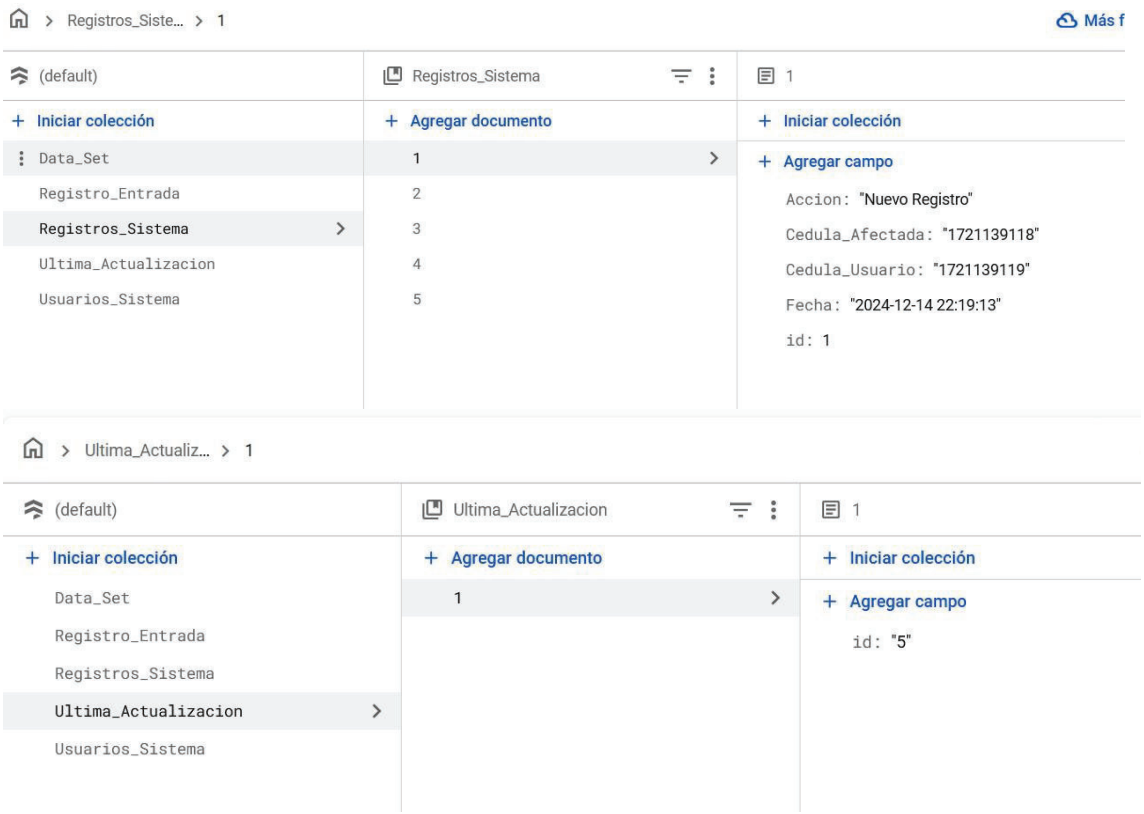


Figura 21. Conexión y recuperación de datos en Firebase.

- Generación del Archivo .pickle: Una vez que se ha actualizado el diccionario con los encodings y datos de las personas, se guarda el archivo .pickle, que contiene toda la información de los registros. Este archivo es utilizado por el módulo de reconocimiento facial para la comparación de las imágenes capturadas con las bases de datos de encodings.

```
# Guardar el diccionario como un archivo .pickle
 with open('encodings.pickle', 'wb') as f:
 pickle.dump(people_data, f)
 print("Datos guardados en encodings.pickle")
```

- Reinicio del Script Secundario: Después de guardar los datos en el archivo .pickle, el script de reconocimiento facial se reinicia para continuar con la detección de personas en tiempo real.

```
# Reiniciar el script secundario
 process = subprocess.Popen(['python3', SCRIPT_TO_MANAGE],
stdout=subprocess.PIPE, stderr=subprocess.PIPE)
 print("Script reiniciado.")
```

- Frecuencia de Actualización: Este proceso se repite cada 15 minutos para garantizar que los datos estén siempre actualizados.

Módulo de Reconocimiento Facial

Detección y reconocimiento de personas a través de una cámara conectada a la Jetson Nano, comparando las imágenes capturadas con los encodings previamente almacenados. El sistema también gestiona las señales LED para indicar el estado del reconocimiento y envía notificaciones a través de Telegram cuando se detecta una persona desconocida.

Flujo de Trabajo:

- Configuración del Hardware: Se configuran los pines GPIO de la Jetson Nano para controlar dos LEDs (uno para indicar "persona conocida" y otro para "desconocida"). Además, se configura el bot de Telegram utilizando un token de acceso y un ID de chat para enviar notificaciones.

- Carga de Encodings: Se carga el archivo .pickle generado en la fase anterior, que contiene los encodings faciales de las personas registradas. Estos encodings se cargan en un BallTree de sklearn, lo que permite realizar

búsquedas eficientes para encontrar el encoding más cercano a una cara detectada.

- Captura de Video: Se inicializa la captura de video utilizando la cámara conectada a la Jetson Nano. Si la cámara no está disponible, se utiliza una cámara USB de respaldo. La resolución de la imagen se reduce para optimizar el rendimiento en la Jetson Nano

- Detección y Reconocimiento Facial: En cada frame de video, se realizan las siguientes operaciones:

  - Se detectan las ubicaciones de las caras utilizando face_recognition. o Se calculan los encodings faciales correspondientes.

  - Se comparan los encodings detectados con los encodings conocidos almacenados en el BallTree. Si la distancia entre un encoding detectado y uno conocido es menor que un umbral predefinido (0.45), se considera que la cara es "conocida". Si la distancia es mayor, se clasifica como "desconocida".

- Control de LEDs: En función del resultado del reconocimiento facial, se activan los LEDs correspondientes: o Si se detecta una persona conocida, se enciende el LED de "conocido" y se apaga el LED de "desconocido". o Si se detecta una persona desconocida, se enciende el LED de "desconocido" y se apaga el LED de "conocido".

- Envío de Notificaciones a Telegram: Si se detecta una persona desconocida, se guarda una imagen de la persona detectada y se envía a través de Telegram con la notificación correspondiente. Esto se maneja de manera asincrónica para evitar bloqueos en el flujo de procesamiento.

- Frecuencia de Reconocimiento: El reconocimiento facial se realiza en tiempo real, con un tiempo de espera de 1 segundo entre cada ciclo de captura y procesamiento.

3.2.5 Tiempos de procesamiento

En la Tabla 11 detalla las pruebas realizadas durante el proceso de registro del usuario. La verificación del funcionamiento se basa en el número de capturas de la cámara o fotografías tomadas del usuario, y se mide el tiempo que le toma al dispositivo realizar estas capturas. Además, se considera el tiempo de procesamiento de las imágenes para que estas sean registradas en la base de datos y almacenadas, lo cual permite su posterior verificación en el sistema de acceso mediante reconocimiento facial.

Tabla 11. Pruebas de procesamiento según el número de capturas y tiempo de procesamiento.

Numero de capturas	Tiempo de captura (segundos)	Tiempo de procesamiento (segundos)	Tiempo total (segundos)
15	2.50	44	46.5
30	5	88	93
50	8.33	156	164.33
75	12.50	225	237.5
100	16.67	295	241.67
150	25	450	475

En base a las pruebas realizadas, se determina que a mayor número de capturas o fotografías obtenidas del usuario, resulta un mejor reconocimiento facial, sin embargo, a mayor número así mismo el tiempo de procesamiento aumenta debido al procesamiento y el hardware que se tiene volviéndolo deficiente, en conclusión, con la captura de 30 a 50 fotogramas en ese rango se considera como aceptable y eficiente el tiempo que le toma al sistema capturar, procesar y posterior verificación fácil del usuario como óptimo para el sistema.

3.2.6 Implementación y verificación de funcionamiento

a. Diagrama de conexión física

Este sistema está compuesto por los siguientes elementos y conexiones:

1. **Jetson Nano**: Es el núcleo central del sistema, encargado de procesar el reconocimiento facial.

- **Cámara**: Conectada directamente a la Jetson Nano a través de un puerto USB, proporciona la entrada de imagen necesaria para realizar el reconocimiento facial.
- **Conexión de red**: El Jetson Nano está conectado a un equipo de red, permitiendo la comunicación con otros dispositivos o sistemas externos para enviar/recibir información.
- **Módulo relé**: La Jetson Nano controla un módulo mediante salidas GPIO. Este relé actúa como un interruptor que gestiona el flujo de corriente hacia la cerradura electromagnética.

2. **Cerradura electromagnética** :
    - Conectada al módulo relé, permite bloquear o desbloquear el acceso a la puerta según las instrucciones de la Jetson Nano.
    - Alimentada por una batería de 12V para garantizar su funcionamiento independiente del Jetson Nano.
3. **Batería de 12V**: Proporciona la energía necesaria para operar la cerradura electromagnética, asegurando un funcionamiento continuo incluso en caso de fallas en el suministro eléctrico principal.

    - Fuente de alimentación: permite mantener cargada la batería, esto ayuda a que en caso de que se vaya la energía eléctrica externa, la seguridad se va a seguir manteniendo.

Las conexiones eléctricas de comunicación se muestran en la Figura 23. El flujo de operación del sistema es el siguiente:

- La cámara captura el rostro de una persona y envía la imagen a la Jetson Nano.
- La Jetson Nano analiza la imagen y determina si la persona tiene acceso autorizado con la comparativa de la información en la base de datos Firebase.
- Si el acceso es autorizado, el Jetson Nano activa el módulo, desbloqueando la cerradura electromagnética y permitiendo el acceso.
- La batería de 12V asegura el funcionamiento estable de la cerradura.

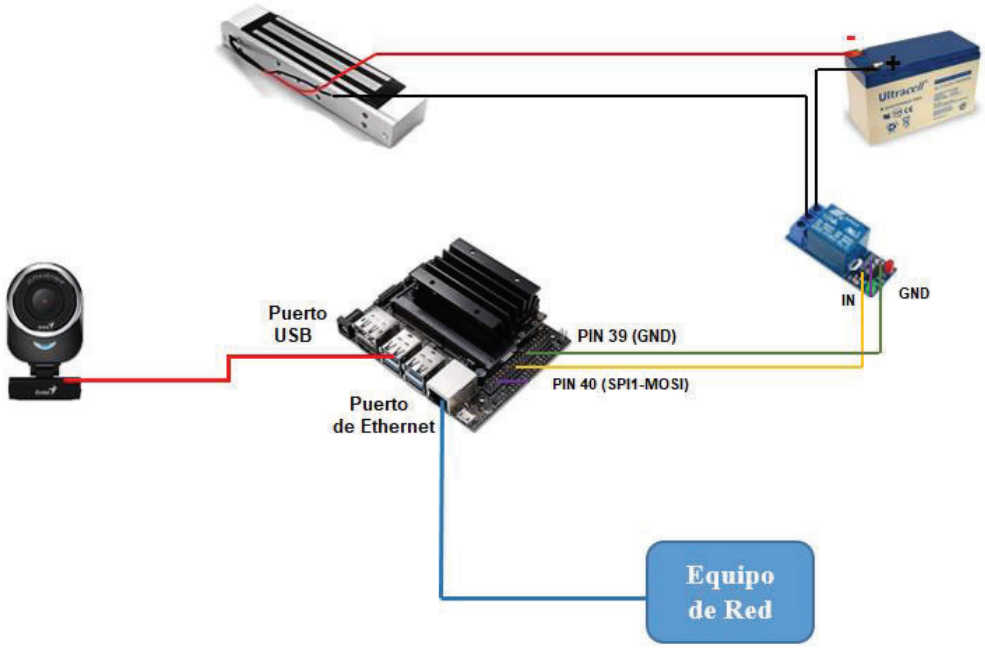


Figura 22. Conexiones del sistema de acceso.

Implementación del prototipo

La implementación se llevó a cabo en los Talleres Tecnológicos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, la cerradura electromagnética se la ubicó en la puerta metálica de ingreso de la casa prototipo a escala que existe en los Talleres de la FISEI, se encuentra instalada de manera vertical por el movimiento de la puerta. La cámara digital HD, está ubicada en la parte superior, una altura estándar que permite la captura o el enfoque del rostro del individuo para el acceso, el dispositivo encargado del procesamiento, reconocimiento y verificación se encuentra ubicado dentro de la casa para evitar daños visuales de la fachada de la casa a escala con el dispositivo y las conexiones, todo esto se muestra en la Figura 24.



Figura 23. Implementación del prototipo.

Verificación de funcionamiento.

El sistema de acceso de control mediante el reconocimiento facial consta de varios procesos o fases, a continuación se detalla el funcionamiento de cada una de ellas y los resultados obtenidos**.**

b. ***Registro del usuario en el sistema:*** En este proceso se toman los datos personales del usuario como se muestra en la Figura 25. Una vez ingresados los datos como nombres, cédula, carrera, entre otros. Al dar click en "registrar" la cámara seleccionada se activa y comienza capturar 30 fotografías del usuario, para procesar y obtener los datos característicos de la persona y transformarlo en matrices numéricas almacenadas en la base de datos para compararlas en el siguiente proceso de detección.

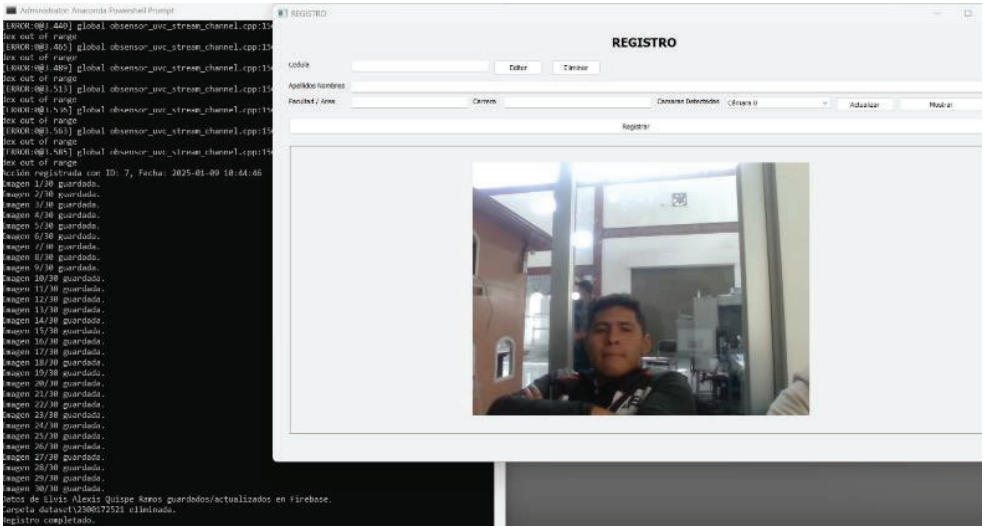


Figura 24. Registro de usuario.

En la Figura 26, se observa la base de datos (Firebase) como almacenada la información dentro del dataset. Se registraron alrededor de 16 usuarios.

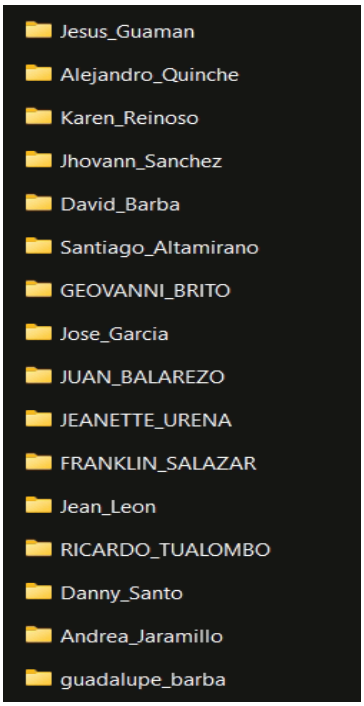


Figura 25. Listados de usuarios creados en dataset

Se almacenan todas las imágenes tomadas, para posterior reconocimiento son 30 fotos en total adicional se crea un txt. Donde está registrado todo en

cuanto a los datos proporcionados como se observa en la Figura 27 y Figura 28.

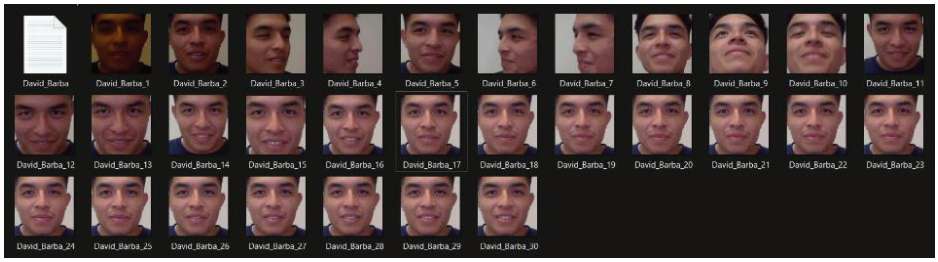


Figura 26. Fotos proporcionadas

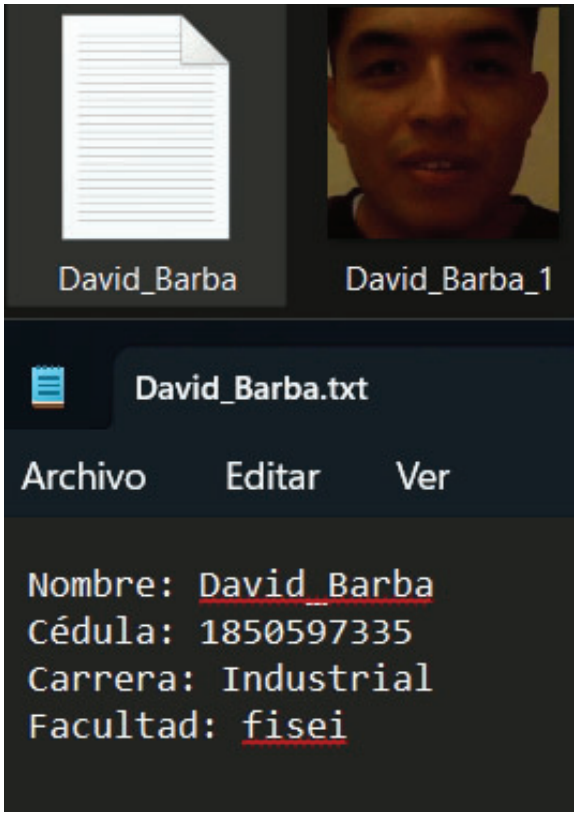


Figura 27. Archivo Txt.

Con cada ingreso en la base de datos se nos tiene un “nuevo registro” igualmente con los datos de cada estudiante y docente, como se observa en la Figura 29.

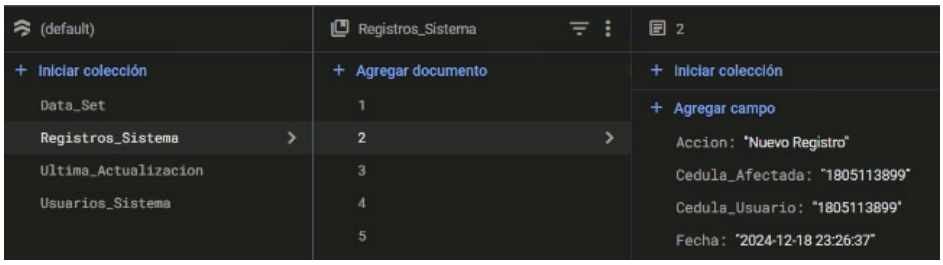


Figura 28. Nuevo registro

En los Anexos G, H e I tenemos las demás capturas recolectas de estudiantes y docentes que más asisten a talleres tecnológicos.

c. ***Detección de la cámara:*** El dispositivo principal en este proceso es la Jetson Nano, aquí se ejecuta el código para realizar el análisis si la persona está registrada y otorgar o negar el acceso, y demás procesos que se mencionan a continuación. En la Figura 30, se muestra la interfaz de la Jetson quien posee un sistema Linux, el código principal se ejecuta y realiza actualizaciones periódicas para identificar nuevos registros o modificaciones en la base de datos y el reconocimiento que carga los encodings, detalla el número de rostros conocidos e inicializa la captura de video.

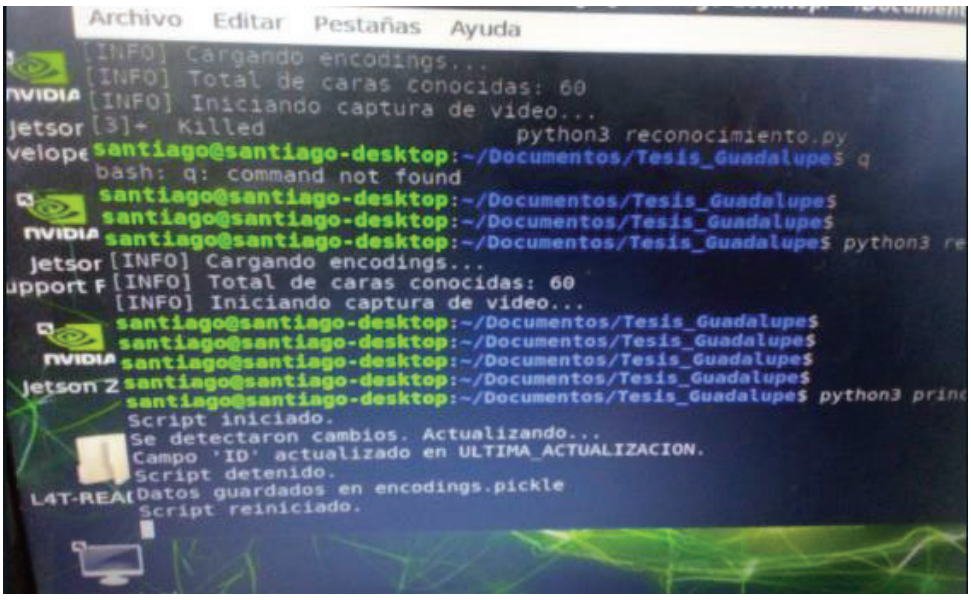


Figura 29. Interfaz de la Jetson Nano, inicio de códigos de reconocimiento.

Se ubica una pantalla cerca del proceso de reconocimiento para visualizar como la cámara identifica al usuario como se muestra en la Figura 31 cuando el usuario es desconocido (marcando su rostro con un recuadro rojo) Como se muestra en la Figura 32 o cuando es reconocido lo identifica y lo asocia con el número de cedula registrado (recuadro verde), tal y como se muestra en la Figura 33.



Figura 30. Proceso de reconocimiento facial.

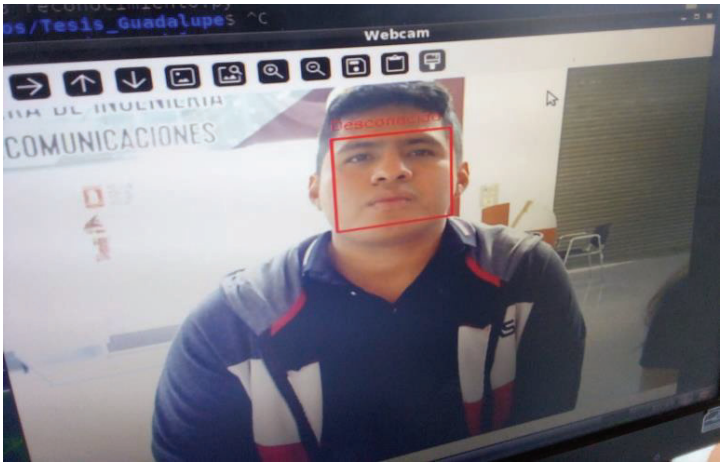


Figura 31. Usuario desconocido

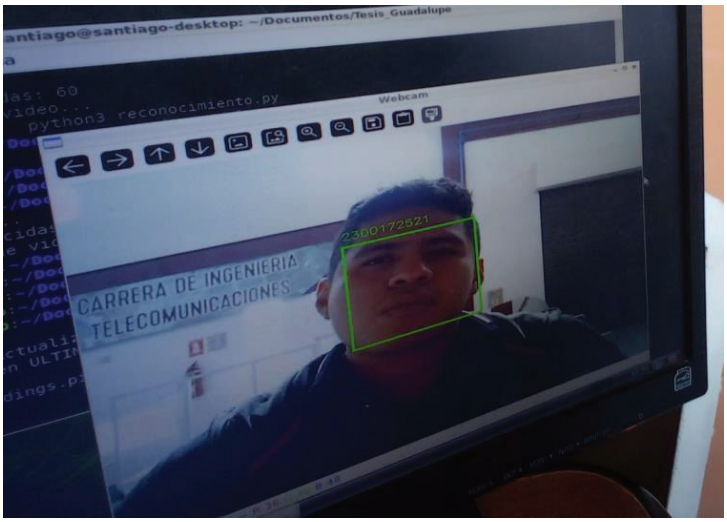


Figura 32. Usuario reconocido con cédula de identidad

A continuación, se detallan los dos eventos, de la otorgación del acceso y el impedimento del acceso a la casa a escala.

- Mientras el usuario no esté registrado, la cámara lo identifica como “Desconocido” como esta en la Figura 34, y la Jetson no envía la orden a la cerradura para que esta sea activada y mantiene la fuerza magnética con la puerta metálica de seguridad.



Figura 33. Individuo no registrado denominado "desconocido".

Al mantenerse como usuario desconocido la Jetson envía en tiempo real un mensaje a la plataforma de Telegram la alerta de una persona desconocida intentado acceder, la alerta llega como se muestra en la Figura 35.



Figura 34. Alertas notificadas en Telegram.

- Cuando el usuario ya está registrado, la cámara lo detecta rápidamente y lo asocia al número de cédula registrado en la base de datos como se muestra en la Figura 36.

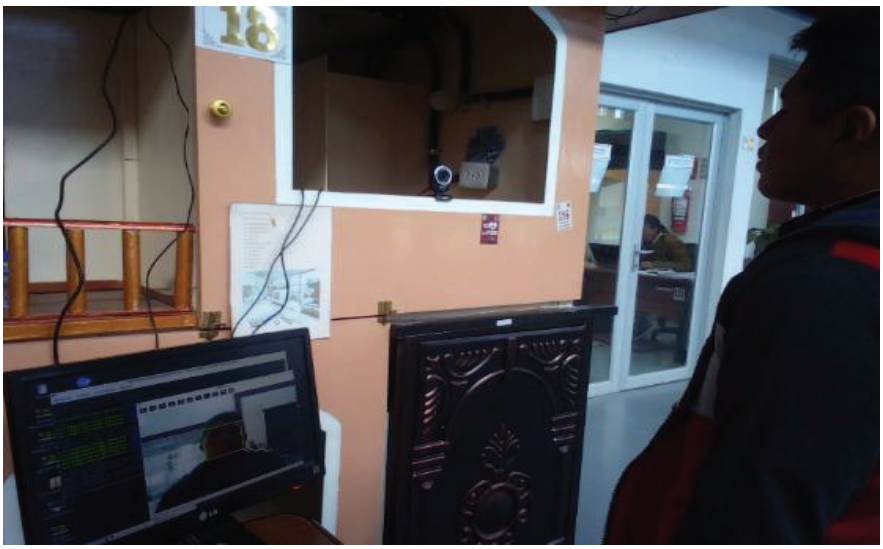


Figura 35. Usuario reconocido por el sistema.

La puerta recibe la orden emitida por la Jetson para que se desactive la fuerza magnética ejercida sobre la puerta metálica, en este proceso la misma cerradura emite un sonido dando a conocer al usuario que ya puede abrir la puerta manualmente con facilidad como se muestra en la Figura 37.



Figura 36. Cerradura magnética otorga acceso de ingreso.

Cabe mencionar, que en el caso de que dos personas a la vez se pongan frente a la cámara en el proceso de reconocimiento, los dos rostros son identificados ya sea como desconocido o identificado en la base de datos su registro, mientras esté una persona registrada, la Jetson envía la solicitud a la cerradura para que permita el acceso, como se muestra la Figura 38.

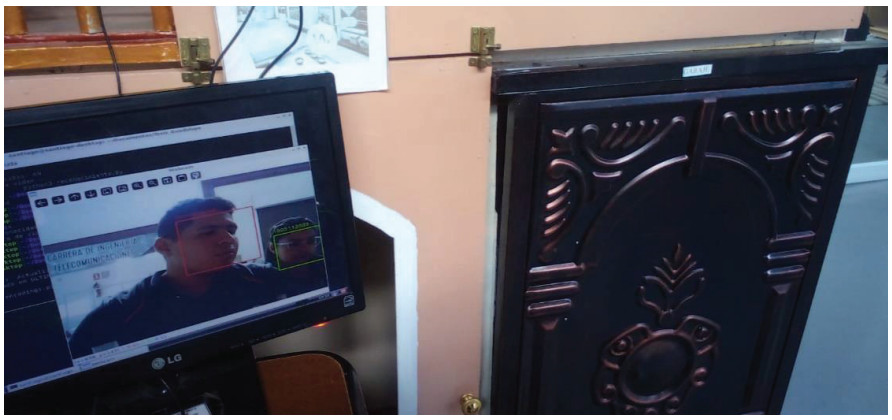


Figura 37. Reconocimiento de varios rostros a la vez.

3.2.7 Presupuesto de implementación del proyecto

En la Tabla 12 se presenta los costos para la realización de prototipo.

Tabla 12 . Insumos para el prototipo

CANTIDAD	DESCRIPCIÓN	TOTAL
1	Jetson nano	$ 400,00
1	Cámara web	$ 54,00
1	Modulo rele	$ 2,50
1	Micro SD de 4GB	$ 5,00
1	Cable UTP	$ 6,00
1	Batería	$ 12,20
1	Funda de tornillos	$ 1,09
1	Broca	$ 0,40
1	Cinta doble fas	$ 1,64
1	Fuente de poder	$ 40,00
1	Electrocerradura	$ 40,00
1	Soporte de la electrocerradura	$ 25,00
TOTAL		$ 587,83

3.2.8 Presupuesto de implementación del proyecto

Para determinar el costo técnico del proyecto, primero se estableció el número de horas laborales empleadas en la programación y configuración del sistema basado en la inteligencia artificial y el control de acceso. En la tabla 12 se detalla este cálculo.

Tabla 13. Horas tomadas para el desarrollo del prototipo

Descripción	**Cantidad**
Tiempo de desarrollo del sistema	30 días
Horas laborales por día	8 horas
Horas totales	240 horas

De acuerdo con los datos del Ministerio de Trabajo, en el Anexo E: Salarios mínimos sectoriales y tarifas 2024, un especialista en sistemas electrónicos tiene una remuneración mínima mensual de $496.57 [20]. Considerando que un mes estándar

tiene 30 días, con 20 días laborables de 8 horas diarias, se obtiene el valor por hora en la Tabla 13.

Tabla 14.Salario para un Ingeniero en Telecomunicaciones

Descripción	**Cantidad (USD)**
Valor mensual	496.57
Valor diario	24.8285
Valor por hora	3.10

El costo del desarrollo técnico viene dado por la Ecuación 3.

$$\boldsymbol{Costo\ desarrollo\ técnico\ =\ Valor\ por\ hora\ *\ Horas\ Totales} \quad (3)$$

Con los valores obtenidos en las Tablas Tabla 9 y Tabla 10 se obtiene:

$$Costo\ desarrollo\ técnico\ =\ 3{,}10\$\ *\ 240\ horas$$

$$Costo\ desarrollo\ técnico\ =\ 744\$$$

3.2.9 Costo Final

El costo total de la investigación incluye tanto el desarrollo técnico como los insumos utilizados durante el proyecto, según la Ecuación 4:

$$Costo\ total\ =\ Costo\ desarrollo\ técnico\ +\ Costo\ de\ insumos \quad (4)$$

Reemplazando los valores obtenidos:

$$Costo\ total\ =\ 602{,}83\$\ +\ \$587{,}83$$

$$Costo\ total\ =\ 1190{,}66\$$$

Por tanto, la totalidad del estudio está valorizado en 1190,66$.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La implementación del sistema de control de acceso autorizado con inteligencia artificial ha demostrado ser muy importante para mejorar la seguridad tanto de las instalaciones como de los estudiantes y docentes, la combinación, de una fuente de alimentación regulada (P1205-B3) junto con una batería de respaldo de 12V garantiza una operación ininterrumpida del sistema de control de acceso. El regulador de voltaje empleado permite la estabilidad del suministro energético, evitando fluctuaciones que puedan comprometer la operación de los dispositivos de seguridad, La integración del sistema de control de acceso con reconocimiento facial requiere una banda ancha mínima de 10 Mbps para transmisión de datos en tiempo real, considerando protocolos de cifrado como AES-256 para la seguridad de la información. Se recomienda el uso de una red LAN Gigabit Ethernet para reducir la latencia y mejorar la capacidad de procesamiento de imágenes biométricas, con la cámara QCam 6000 para el reconocimiento facial de 2MP con un ángulo de 90° hacia arriba y abajo, y rotación de 360°, lente de foco fijo con interfaz USB con su sensor de alta definición CMOS de 2 megapíxeles, lo que permite una resolución de imagen de hasta 1920 x 1080 píxeles, con el software inteligente permite una gestión más efectiva del acceso a talleres tecnológicos, reduciendo tiempo de espera y aumentando la seguridad en las instalaciones.

- El análisis de tecnologías disponibles a nivel mundial ha permitido identificar soluciones innovadoras que se adaptan a las necesidades específicas de la FISEI. La aplicación de algoritmos de inteligencia artificial para el control y monitoreo del acceso no solo optimiza el reconocimiento facial, sino que también proporciona un sistema robusto que puede aprender y adaptarse a nuevas condiciones, mejorando continuamente su desempeño. Se ha utilizado algoritmos como OpenCV un algoritmo que tanto bibliográfica como práctico, se comprueba el alto nivel de confiabilidad y efectividad de aproximadamente

un 96%, además, que permite optimizar los recursos a nivel de hardware en comparación con algoritmos más robustos.

- La validación del funcionamiento del sistema a través de pruebas piloto ha demostrado su eficacia en situaciones reales, garantizando un control efectivo del acceso. La activación de la cerradura eléctrica mediante un relé de estado sólido (SSR) de 12V/5A ha demostrado mayor eficiencia en comparación con relés electromecánicos convencionales, reduciendo el tiempo de respuesta del mecanismo en un 25% y mejorando la durabilidad del sistema, la necesidad de implementar supresores de picos de voltaje y diodos de protección (1N4007 o equivalente) en paralelo con la cerradura para evitar daños en los circuitos por descargas transitorias. Además, la puesta a tierra adecuada con un valor de resistencia inferior a 5 Ω reduce el riesgo de interferencias electromagnéticas (EMI). Este proceso no solo asegura el correcto funcionamiento del sistema, sino que también genera confianza entre los usuarios, lo que es fundamental para la aceptación y éxito del mismo.

4.2 Recomendaciones

- Se recomienda establecer un plan de mantenimiento regular para el sistema de control de acceso automatizado. Incluyendo actualizaciones periódicas del software y hardware, así como revisiones sistemáticas para asegurar su correcto funcionamiento, para mejorar la velocidad de reconocimiento facial y la respuesta del sistema, se recomienda implementar VLANs y QoS (Quality of Service) en la red de telecomunicaciones, priorizando paquetes de datos biométricos sobre otros servicios. El mantenimiento ayudará a prevenir fallos y garantizará que el sistema siga siendo eficaz frente a nuevas amenazas.

- Es esencial proporcionar capacitación continua al personal encargado y de seguridad sobre el uso y gestión del sistema, es recomendable implementar un sistema de monitoreo remoto basado en SNMP (Simple Network Management Protocol) para supervisar el estado de la fuente de alimentación, batería y conexiones de red, con alertas automatizadas ante fallas o pérdida de señal. Esto no solo asegura que todos los usuarios estén familiarizados con las

funcionalidades del sistema, sino que también fomenta una cultura de seguridad dentro de talleres tecnológicos, aumentando la efectividad general del control de acceso.

- Se sugiere implementar un proceso sistemático para el monitoreo y evaluación constante del rendimiento del sistema, Como medida de contingencia, se propone añadir un módulo de autenticación dual (facial + PIN o RFID) para reforzar la seguridad y evitar accesos no autorizados en caso de fallos en el reconocimiento biométrico.

REFERENCIAS BIBLIOGRÁFICA

[1] E. Castillano Mogollón y A. M. Santamaría Montero , «Control de acceso vehicular con visión artificial para urbanizaciones en la ciudad de Piura,» Universidad César Vallejo, Piura - Perú, 2023.

[2] D. J. Llanga Herrera, « Diseño de un prototipo de entorno de video vigilancia integrado Open Source utilizando cámaras integradas en vehículos.,» Escuela Superior Politécnica de Chimborazo, Riobamba, 2022.

[3] J. Cabrera y D. Pérez , «Diseño y Simulación de Sistema de Videovigilancia para el Monitoreo de Cultivos y Control de Acceso del Personal.,» Escuela Superior Politécnica del Litoral, 2022.

[4] E. A. Briones Gárate, «Sistema web de reconocimiento facial para control de acceso biométrico, utilizando inteligencia artificial,» ESPOL, Guayaquil, 2020.

[5] C. A. Nacipucha Nacipucha y J. J. Frías Pérez, «Diseño de un prototipo de control de acceso a traves de reconocimiento facial mediante la creacion de un algoritmo basado en software libre utilizando LattePanda,» Guayaquil, 2020.

[6] A. De Vita, «Funciones y características de los sistemas de control de acceso,» *Altamira,* 2022.

[7] O. Gómez Baryolo, «Modelo de control de acceso para sistemas de información basados en tecnologías web.,» *Journal of Technology,* vol. 12, nº 2, 2023.

[8] «Securitas Technology,» 2024. [En línea]. Available: https://www.securitastechnology.com/us-es/soluciones/integracion-de-sistemas-de-seguridad.

[9] O. S. S.L., «omnitecsystems,» 28 marzo 2022. [En línea]. Available: https://www.omnitecsystems.es/omni/blog/tipos-de-control-de-acceso-mejoran-seguridad.

[10] T. Guillén, «Signicat,» 10 julio 2024. [En línea]. Available: https://www.signicat.com/es/blog/como-funciona-el-reconocimiento-facial.

[11] J. Nielsen, Usability Engineering, Academic Press, 1994.

[12] B. Shneiderman y C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Strategies for Effective Human-Computer Interaction, 2010.

[13] IEEE, Standards for Interface Design, 2021.

[14] S. Kellett, «¿Qué es un reconocimiento facial?,» 20 enero 2021. [En línea]. Available: https://www.avast.com/es-es/c-facial-recognition.

[15] V. S. Hendry. [En línea]. Available: https://veridas.com/es/reconocimiento-facial/..

[16] M. Diaz Martinez, S. R. Roman, G. Santana Esparza y M. Morales Rodriguez, «La industria 4.0 y las redes neuronales artificiales en la ingeniería industrial: Una revisión sistemática de la literatura,» Mexico, 2023.

[17] M. D. González Mosqueda y R. K. & Rodríguez Martínez, Desarrollo de una interfaz gráfica para el reconocimiento de rostros., 2024.

[18] A. Marco, W. Muñoz-Robles y M. A. & Trinidad-Malpartida, Análisis bibliométrico de la aplicación de redes neuronales convolucionales en el área de la visión artificial Bibliometric analysis of the application of convolutional neural networks in the area of computer vision., vol. 18, Investigación Valdizana, pp. e2105-e2105.

[19] G. López , F. Robalino, P. Torres y X. Lopez, «Análisis de algoritmos de procesamiento de imágenes y reconocimiento de patrones con Visión Artificial,» *Medwave,* 2023.

[20] F. Schroff, D. Kalenichenko y J. & Philbin, FaceNet: A unified embedding for face recognition and clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[21] Y. Taigman, M. Yang, M. Ranzato y L. & Wolf, DeepFace: Closing the gap to human-level performance in face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)., 2014.

[22] J. Deng, J. Guo, N. Xue y S. & Zafeiriou, Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering., Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)., 2015, pp. 815-823.

[23] G. Bradski y A. & Kaehler, Learning OpenCV: Computer vision with the OpenCV library., O'Reilly Media, Inc., 2008.

[24] D. E. King, Dlib-ml: A machine learning toolkit., vol. 10, he Journal of Machine Learning Research, 2009, pp. 1755-1758.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean y X. ... & Zheng, {TensorFlow}: a system for {Large-Scale} machine learning., In 12th USENIX symposium on operating systems design and implementation (OSDI 16), 2016, pp. 265-283.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan y S. ... & Chintala, Pytorch: An imperative style, high-performance deep learning library., vol. 32, Advances in neural information processing systems, 2019.

[27] A. Geitgey, face_recognition: The world's simplest facial recognition API for Python and the command line., 2017.

[28] W. R. Guerrero Albán, «Detección y descripción de puntos característicos en imágenes multiespectrales,» *Tecnológica ESPO,* 2014.

[29] Hikvision, «Innovación, seguridad y conectividad en el control de accesos,» *innovacion* .

[30] J. E. M. Reyes Campos, C. S. Castañeda Rodríguez y L. D. Alva Luján, «Sistema de reconocimiento facial para el control de accesos mediante Inteligencia Artificial,» vol. 4, 2023.

[31] ANSI/BHMA, Standards for Electronic Locks, American National Standards Institute., 2021.

[32] A. H. Inc, Smart Lock Overview, 2022.

[33] Ministerio del Trabajo, 3 Enero 2024. [En línea]. Available: https://www.trabajo.gob.ec/wp-content/uploads/downloads/2024/01/3.-Salarios-minimos-sectoriales-2024.pdf.

ANEXOS

Anexo A. Diagramas

La Figura 37 muestra un diagrama de flujo de un sistema de reconocimiento facial. Incluye pasos como la captura de imágenes con una cámara, el preprocesamiento con OpenCV, la detección facial con DLib, la extracción de características y el reconocimiento facial. Finalmente, los datos faciales se comparan con una base de datos para otorgar acceso.

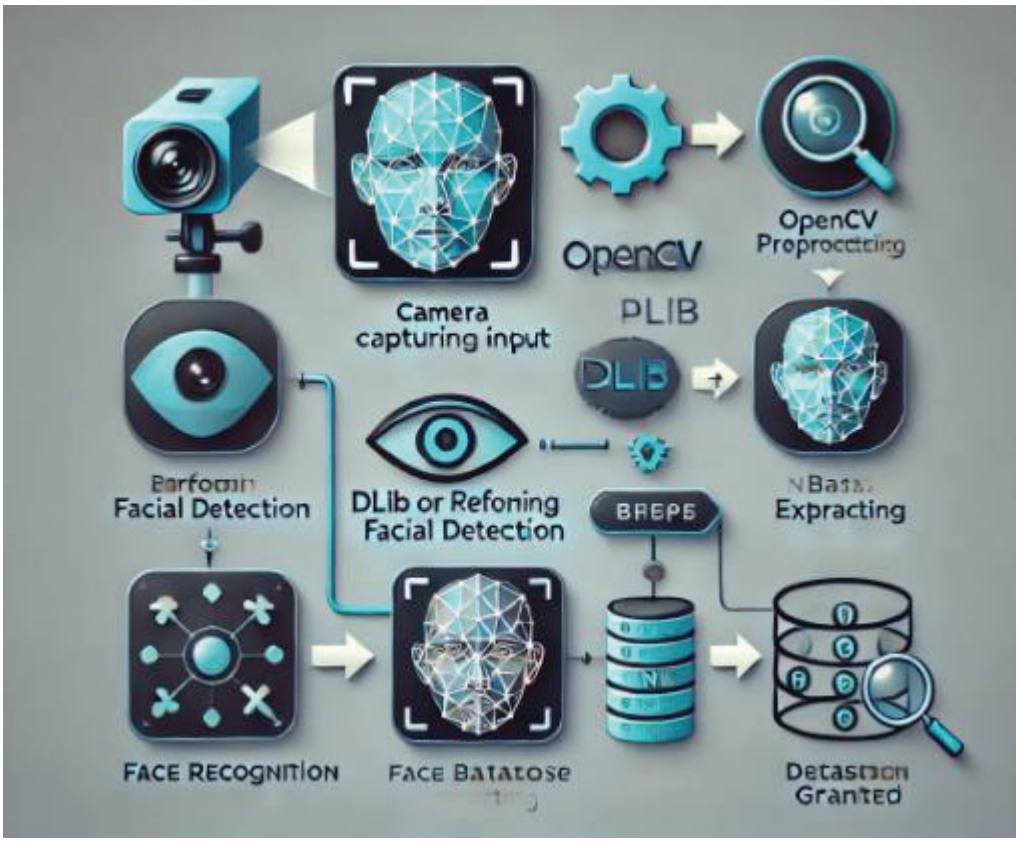


Figura 38. Esquema general de las herramientas y algoritmos en los procesos del sistema de acceso.

Anexo B. Datasheet y descripciones de la placa Jetson Nano modelo P3541

La placa Jetson Nano se lo puede observar con mejor detalle en la Figura 38 a la 41, se representan las diferentes vistas, permitiendo conocer los periféricos y pines para el uso en el sistema de acceso, ya que es el dispositivo principal o el cerebro del sistema.

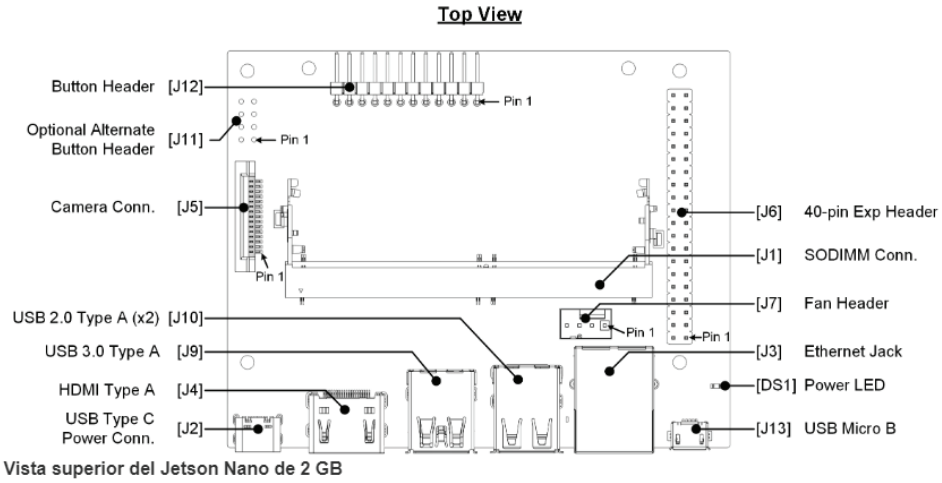


Figura 39. Vista Superior de la placa Jetson.

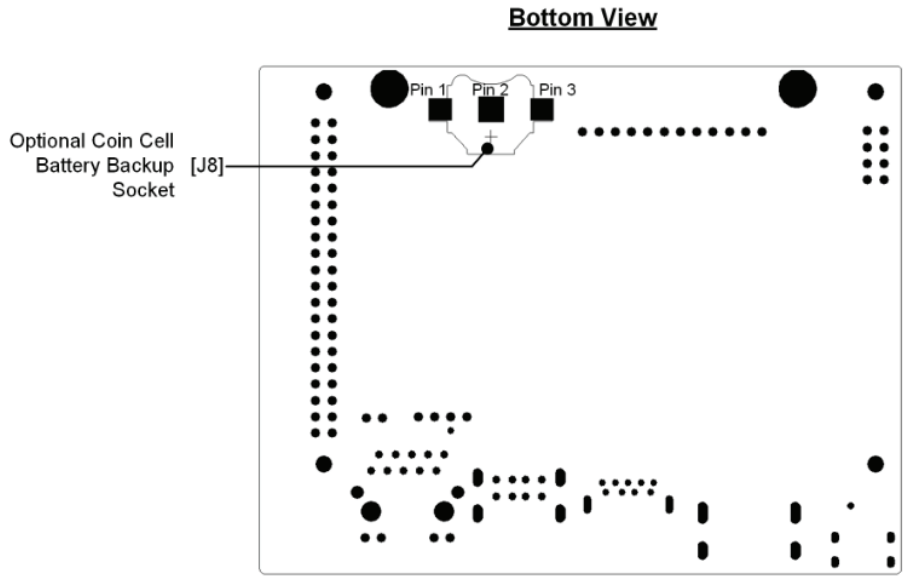


Figura 40. Vista inferior de la placa.

En la Figura 40 se observa de mejor manera los distintos periféricos con los que consta la placa Jetson Nano.

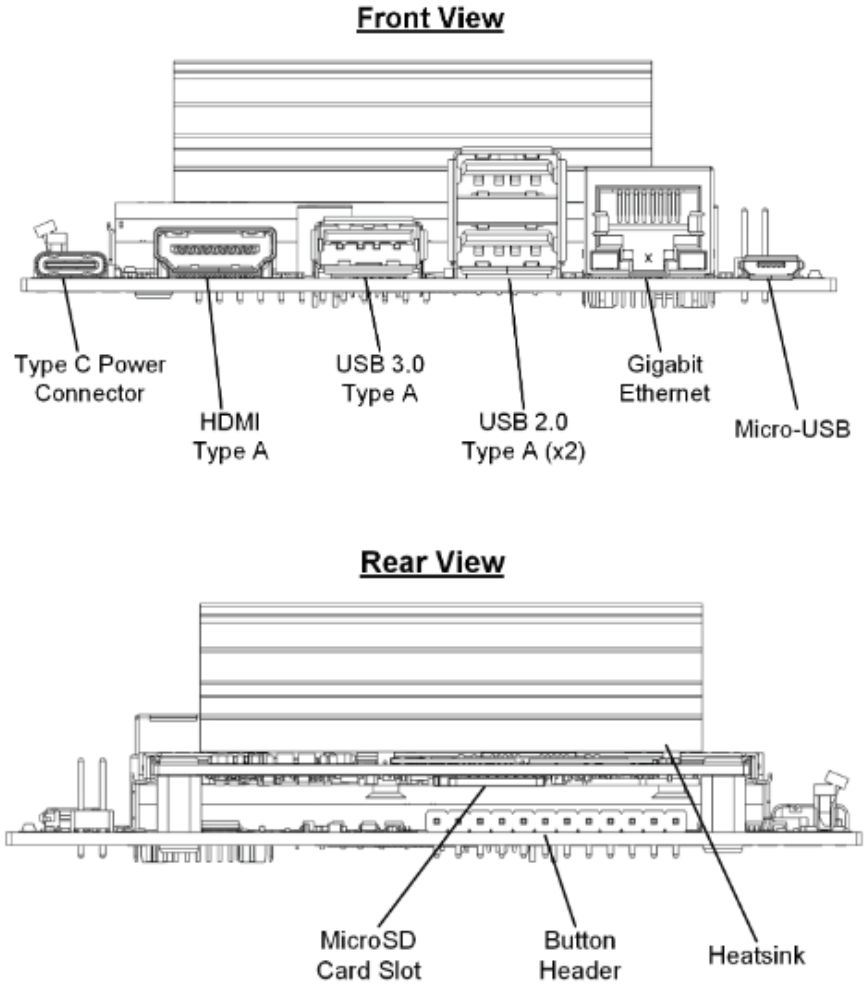


Figura 41. Vista delantera y tracera de la placa Jetson.

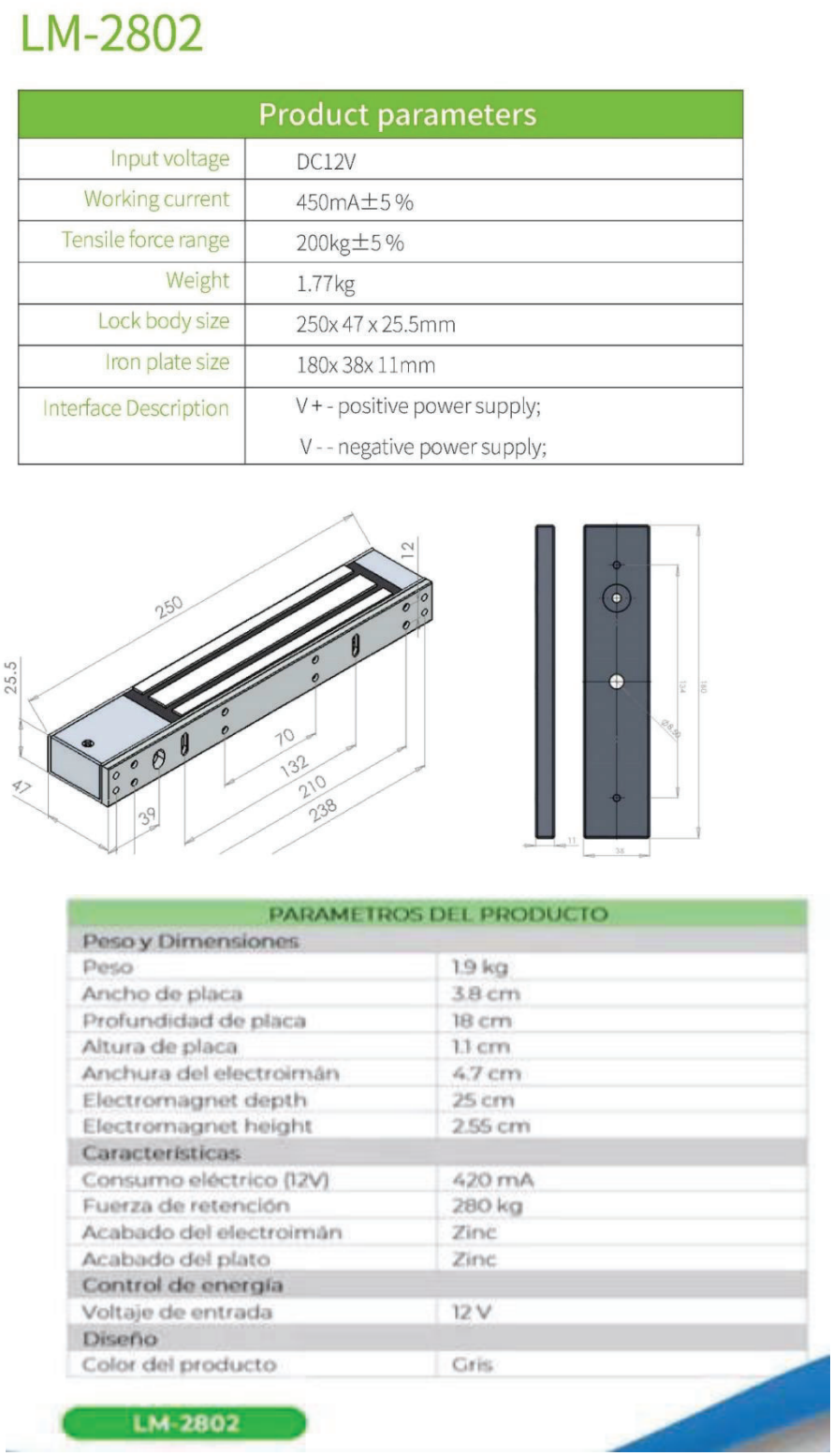
En la Figura 41, esquema de pines GPIO de un sistema SoC, indicando las asignaciones de funciones predeterminadas y alternas. Incluye etiquetas como pines de alimentación, UART, SPI, I2C y GPIO, junto con números de pines físicos, Linux GPIO y SoC GPIO.

SoC GPIO	Linux GPIO #	Alternate Function	Default Function			Default Function	Alternate Function	Linux GPIO #	SoC GPIO
			3.3 VDC	1	2	5 VDC			
PJ.03	75	GPIO	I2C1_SDA	3	4	5 VDC			
PJ.02	74	GPIO	I2C1_SCL	5	6	GND			
PBB.00	216	AUD_CLK	GPIO	7	8	UART1_TXD	GPIO	48	PG.00
			GND	9	10	UART1_RXD	GPIO	49	PG.01
PG.02	50	UART1_RTS	GPIO	11	12	GPIO	I2S0_SCLK	79	PJ.07
PB.06	14	SPI1_SCK	GPIO	13	14	GND			
PY.02	194		GPIO	15	16	GPIO	SPI1_CS1	232	PDD.00
			3.3 VDC	17	18	GPIO	SPI1_CS0	15	PB.07
PC.00	16	SPI0_MOSI	GPIO	19	20	GND			
PC.01	17	SPI0_MISO	GPIO	21	22	GPIO	SPI1_MISO	13	PB.05
PC.02	18	SPI0_SCK	GPIO	23	24	GPIO	SPI0_CS0	19	PC.03
			GND	25	26	GPIO	SPI0_CS1	20	PC.04
PB.05	13	GPIO	I2C0_SDA	27	28	I2C0_CLK	GPIO	18	PC.02
PS.05	149	CAM_MCLK	GPIO	29	30	GND			
PZ.00	200	CAM_MCLK	GPIO	31	32	GPIO	PWM	168	PV.00
PE.06	38	PWM	GPIO	33	34	GND			
PJ.04	76	I2S0_FS	GPIO	35	36	GPIO	UART1_CTS	51	PG.03
PB.04	12	SPI1_MOSI	GPIO	37	38	GPIO	I2S0_DIN	77	PJ.05
			GND	39	40	GPIO	I2S0_DOUT	78	PJ.06

Figura 42. Esquema de pines GPIO.

Anexo C. Datasheet y descripciones de la cerradura electromagnética

La Figura 42 muestra las especificaciones del modelo LM-2802, un electroimán de cerradura.



LM-2802

Product parameters	
Input voltage	DC12V
Working current	450mA±5 %
Tensile force range	200kg±5 %
Weight	1.77kg
Lock body size	250x 47 x 25.5mm
Iron plate size	180x 38x 11mm
Interface Description	V + - positive power supply; V - - negative power supply;

PARAMETROS DEL PRODUCTO	
Peso y Dimensiones	
Peso	1.9 kg
Ancho de placa	3.8 cm
Profundidad de placa	18 cm
Altura de placa	1.1 cm
Anchura del electroimán	4.7 cm
Electromagnet depth	25 cm
Electromagnet height	2.55 cm
Características	
Consumo eléctrico (12V)	420 mA
Fuerza de retención	280 kg
Acabado del electroimán	Zinc
Acabado del plato	Zinc
Control de energía	
Voltaje de entrada	12 V
Diseño	
Color del producto	Gris

LM-2802

Figura 43. Especificaciones técnicas de la cerradura magnética.

Anexo D. Datasheet y descripciones de la cámara web.

La Figura 46 muestra las especificaciones de la cámara Webcam Genius QCam 6000 Full HD 1080P.



Figura 44. Genius QCam Full HD 1080P.

Especificaciones

Interfaz	USB
Resolución (DPI)	2 MP / 1920 x 1080 / 1280 x 720 / 640 x 480 píxeles
Resolución de video	• CIF / VGA: Hasta 30 fps • 720P HD: Hasta 30 fps • 1080P: Hasta 30 fps
Sensor de imagen	CMOS de 2MP
Tipo de lente	lente de foco fijo
Inclinación y giro	Arriba y abajo 90° / Rotación 360°
Micrófono incorporado	SI
UVC (Plug & Play)	SI

Dimensiones y Peso

Dimensiones (A x A x P) mm	54.5 x 42 x 75.4 mm
Dimensiones (A x A x P) pulgadas	2.15 x 1.65 x 2.97 pulgadas
Peso	Aprox. 78 g (Incluye clip y cable)

Compatibilidad / Requerimientos

Compatibilidad	• Windows 7, 8, 8.1, 10 o posterior • Mac OS X 10.10 o posterior • Linux 2.6.21 o posterior
Requerimientos	Un Puerto USB-A

Contenido de la caja

Contenido de la caja	• 1x QCam 6000 • 1x Cable USB • Manual del usuario en varios idiomas

Garantía

Garantía de hardware limitada*	12 Meses de garantía de hardware limitada * La garantía queda anulada si se abre o repara sin autorización.

Figura 45. Datasheet cámara

Anexo E. Implementación del prototipo.

En la Figura 44, se muestra la instalación e implementación de los elementos del sistema de control mediante reconocimiento facial, la ubicación de los mismos en la casa y las conexiones eléctricas desarrolladas.



Figura 46. Collage de fotografías de implementación del sistema.

Anexo F. Código python de cada fase del sistema.

A continuación se detalla el código de la primera fase del sistema, el mismo que realiza el registro de los usuarios y el almacenamiento en la base de datos.

```
import os
import cv2
import face_recognition
import firebase_admin
from firebase_admin import credentials, firestore
import shutil
import json
from PyQt5 import QtWidgets, uic
from PyQt5.QtCore import QThread, pyqtSignal, Qt
from PyQt5.QtGui import QImage, QPixmap
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout,
QLineEdit, QMessageBox, QInputDialog, QInputDialog, QMessageBox,
QLineEdit
from datetime import datetime
import sys
import bcrypt

# Inicializa Firebase Admin SDK
cred = credentials.Certificate("tesisguadalupe.json")  # Reemplaza con
tu archivo JSON de Firebase
firebase_admin.initialize_app(cred)

# Obtén la referencia a Firestore
db = firestore.client()

qtCreatorFile = "registro.ui"  # Nombre del archivo .ui
Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)

class CameraThread(QThread):
    new_frame_signal = pyqtSignal(object)

    def __init__(self, parent=None):
        super().__init__(parent)
        self.cap = None
        self.running = False

    def run(self):
        while self.running:
            ret, frame = self.cap.read()
            if ret:
                self.new_frame_signal.emit(frame)
```

```
    def start_capture(self, camera_index=0):
        self.cap = cv2.VideoCapture(camera_index)
        if not self.cap.isOpened():
            print("No se pudo abrir la cámara.")
            return
        self.running = True
        self.start()

    def stop_capture(self):
        self.running = False
        self.wait()
        if self.cap is not None:
            self.cap.release()

class RegistrationThread(QThread):
    registration_complete_signal = pyqtSignal(str)
    update_progress_signal = pyqtSignal(str)
    new_frame_signal = pyqtSignal(object)

    def __init__(self, cedula, person_name, carrera, facultad,
camera_index, parent=None):
        super().__init__(parent)
        self.cedula = cedula
        self.person_name = person_name
        self.carrera = carrera
        self.facultad = facultad
        self.camera_index = camera_index

    def run(self):
        dataset_folder = 'dataset'
        if not os.path.exists(dataset_folder):
            os.makedirs(dataset_folder)
        person_folder = os.path.join(dataset_folder, self.cedula)
        if os.path.exists(person_folder):
            for file in os.listdir(person_folder):
                os.remove(os.path.join(person_folder, file))
        else:
            os.makedirs(person_folder)
        info_file_path = os.path.join(person_folder,
f"{self.cedula}.txt")
        with open(info_file_path, 'w') as f:
            f.write(f"Nombre: {self.person_name}\n")
            f.write(f"Cédula: {self.cedula}\n")
            f.write(f"Carrera: {self.carrera}\n")
            f.write(f"Facultad: {self.facultad}\n")
        cap = cv2.VideoCapture(self.camera_index)
        num_images = 30
        captured_images = 0
```

```
        while captured_images < num_images:
            ret, frame = cap.read()
            if not ret:
                self.update_progress_signal.emit("No se pudo capturar
el video.")
                break
            rgb_frame = frame[:, :, ::-1]
            face_locations = face_recognition.face_locations(rgb_frame)
            if len(face_locations) == 1:
                top, right, bottom, left = face_locations[0]
                face_image = frame[top:bottom, left:right]
                image_path = os.path.join(person_folder,
f"{self.cedula}_{captured_images + 1}.jpg")
                cv2.imwrite(image_path, face_image)
                captured_images += 1
                self.update_progress_signal.emit(f"Imagen
{captured_images}/{num_images} guardada.")
                self.new_frame_signal.emit(frame)
        cap.release()

        encodings = []
        for img_name in os.listdir(person_folder):
            img_path = os.path.join(person_folder, img_name)
            if img_name.lower().endswith(('.png', '.jpg', '.jpeg')):
                try:
                    image = cv2.imread(img_path)
                    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    boxes = face_recognition.face_locations(rgb,
model='cnn')
                    encodings_in_image =
face_recognition.face_encodings(rgb, boxes)
                    encodings.extend(encodings_in_image)
                except Exception as e:
                    self.update_progress_signal.emit(f"Error al
procesar la imagen {img_name}: {e}")

        embeddings_json = json.dumps([encoding.tolist() for encoding in
encodings])
        people_data = {
            'info': {
                'Nombre': self.person_name,
                'Cédula': self.cedula,
                'Carrera': self.carrera,
                'Facultad': self.facultad,
            },
            'encodings': embeddings_json
        }
```

```
        db.collection('Data_Set').document(self.cedula).set(people_data
, merge=True)
        self.update_progress_signal.emit(f"Datos de {self.person_name}
guardados/actualizados en Firebase.")

        shutil.rmtree(person_folder)
        self.update_progress_signal.emit(f"Carpeta {person_folder}
eliminada.")

        self.registration_complete_signal.emit("Registro completado.")


class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
    def __init__(self):
        QtWidgets.QMainWindow.__init__(self)
        Ui_MainWindow.__init__(self)
        self.setupUi(self)
        self.setWindowFlags(Qt.WindowStaysOnTopHint)
        self.botonActualizar.clicked.connect(self.actualizar_camaras)
        self.actualizar_camaras()
        self.botonMostrar.clicked.connect(self.iniciar_camara)
        self.botonRegistrar.clicked.connect(self.iniciar_registro)
        self.botonEliminar.clicked.connect(self.eliminar_registro)
        self.lineEditCedula.returnPressed.connect(self.on_enter)
        self.botonEditar.clicked.connect(self.editar_registro)
        self.camera_thread = CameraThread()
        self.camera_thread.new_frame_signal.connect(self.displayImage)
        self.accion = "Nuevo Registro"
        self.registration_thread = None


    def editar_registro(self):
        self.accion = "Editado"
        bandera = False
        text = self.lineEditCedula.text()
        # Intentar convertir el texto a un número flotante
        try:
            float_value = int(text)
            # Obtén la referencia al documento usando su ID
            doc_ref =
db.collection('Data_Set').document(str(float_value))
            doc = doc_ref.get()
            if doc.exists:
                data = doc.to_dict()
                self.lineEditApellidosNombres.setText(data['info']['Nom
bre'])
                self.lineEditFacultadArea.setText(data['info']['Faculta
d'])
```

```
                self.lineEditCarrera.setText(data['info']['Carrera'])
                self.lineEditCedula.setDisabled(True)
            else:
                self.show_message("No existe esa cedula, debes
registrar una nueva para poder editar")



        except ValueError:
            self.show_message("El valor ingresado no es un número
valido")



    def eliminar_registro(self):
        bandera = False
        text = self.lineEditCedula.text()
        # Intentar convertir el texto a un número flotante
        try:
            float_value = int(text)

            # Mostrar ventana emergente para ingresar la cédula
(usuario)
            cedula_us, ok_cedula = QInputDialog.getText(self, "Registro
de Usuario", "Ingrese su cédula:")
            if not ok_cedula or not cedula_us:  # Verifica si el
usuario canceló o no ingresó la cédula
                QMessageBox.warning(self, "Error", "Debe ingresar una
cédula.")
                return
            # Mostrar ventana emergente para ingresar la contraseña
            password, ok_password = QInputDialog.getText(self,
"Registro de Usuario", "Ingrese su clave:", QLineEdit.Password)
            if not ok_password or not password:  # Verifica si el
usuario canceló o no ingresó la clave
                QMessageBox.warning(self, "Error", "Debe ingresar una
clave.")
                return

            # Recuperar todos los documentos de la colección
'people_data'
            docs = db.collection('Usuarios_Sistema').stream()
            for doc in docs:
                # Obtener el ID del documento (que corresponde a la
cédula)
                cedula_var = doc.id
                if cedula_var == cedula_us:
                    # Obtener los datos del documento
```

```
                    data = doc.to_dict()
                    decrypted_password =
bcrypt.checkpw(password.encode('utf-8'), data['password'].encode('utf-
8'))
                    if decrypted_password:
                        bandera = True


            if bandera == True:
                doc_ref = db.collection('Data_Set').document(str(text))
                doc_ref.delete()
                self.registrar_accion("Eliminado", str(float_value),
cedula_us)
                #reset
                self.lineEditCedula.setDisabled(False)
                self.lineEditCedula.setText("")
                self.lineEditApellidosNombres.setText("")
                self.lineEditFacultadArea.setText("")
                self.lineEditCarrera.setText("")
                self.show_message(f"Documento eliminado de forma
correcta")


        except ValueError:
            self.show_message("El valor ingresado no es un número
valido")

    def show_message(self, message):
        # Crear un QMessageBox para mostrar el mensaje emergente
        msg = QMessageBox(self)
        msg.setIcon(QMessageBox.Information)
        msg.setWindowTitle("Notificación")
        msg.setText(message)
        msg.exec_()

    # Función para obtener el último ID en la colección y generar el
siguiente ID
    def get_next_id(self):
        # Obtén todos los documentos de la colección
'Registros_Sistema'
        docs = db.collection('Registros_Sistema').order_by('id',
direction=firestore.Query.DESCENDING).limit(1).stream()

        last_id = 0  # Valor por defecto si no hay documentos
        for doc in docs:
            last_id = doc.to_dict().get('id', 0)  # Obtiene el ID del
último documento
```

```
        return last_id + 1  # Sumar 1 al último ID encontrado

    # Función para registrar los datos en Firestore
    def registrar_accion(self, accion, cedula_afectada,
cedula_usuario):
        # Obtener el siguiente ID
        next_id = self.get_next_id()

        # Obtener la fecha actual
        fecha_actual = datetime.now().strftime('%Y-%m-%d %H:%M:%S')  #
Formato 'YYYY-MM-DD HH:MM:SS'

        # Crear un diccionario con los datos que se van a guardar
        data = {
            'id': next_id,
            'Accion': accion,
            'Cedula_Afectada': cedula_afectada,
            'Cedula_Usuario': cedula_usuario,
            'Fecha': fecha_actual
        }

        # Guardar los datos en Firestore en la colección
'Registros_Sistema'
        db.collection('Registros_Sistema').document(str(next_id)).set(d
ata)

        print(f"Acción registrada con ID: {next_id}, Fecha:
{fecha_actual}")

    def on_enter(self):
        bandera = True
        # Obtener el texto del QLineEdit y ejecutar el comando
        text = self.lineEditCedula.text()
        # Intentar convertir el texto a un número flotante
        try:
            float_value = int(text)
            # Recuperar todos los documentos de la colección
'people_data'
            docs = db.collection('Data_Set').stream()
            for doc in docs:
                # Obtener el ID del documento (que corresponde a la
cédula)
                cedula = doc.id
                if cedula == text:
                    self.show_message(f"Ya existe esa cedula en la base
de datos, puedes editar o eliminar")
                    bandera = False
            if bandera == True:
```

```
                self.show_message("Puede realizar el nuevo ingreso")
                self.lineEditCedula.setDisabled(True)

        except ValueError:
            self.show_message("El valor ingresado no es un número
valido")




    def iniciar_camara(self):
        self.camera_thread.start_capture(camera_index=int(self.comboBox
Camaras.currentIndex()))

    def detener_camara(self):
        self.camera_thread.stop_capture()

    def displayImage(self, img):
        qformat = QImage.Format_Indexed8
        if len(img.shape) == 3:
            if img.shape[2] == 4:
                qformat = QImage.Format_RGBA8888
            else:
                qformat = QImage.Format_RGB888
        img = QImage(img.data, img.shape[1], img.shape[0],
img.strides[0], qformat)
        img = img.rgbSwapped()
        self.imgLabel.setPixmap(QPixmap.fromImage(img))
        self.imgLabel.setAlignment(Qt.AlignHCenter | Qt.AlignVCenter)

    def listar_camaras(self):
        max_dispositivos = 10
        camaras_disponibles = []

        for i in range(max_dispositivos):
            cap = cv2.VideoCapture(i)
            if cap.isOpened():
                camaras_disponibles.append(f"Cámara {i}")
                cap.release()

        return camaras_disponibles

    def actualizar_camaras(self):
        camaras = self.listar_camaras()
        self.comboBoxCamaras.clear()
        if camaras:
            self.comboBoxCamaras.addItems(camaras)
```

```
        else:
            self.comboBoxCamaras.addItem("No se encontraron cámaras")

    def iniciar_registro(self):
        bandera = False
        cedula = self.lineEditCedula.text()
        person_name = self.lineEditApellidosNombres.text()
        carrera = self.lineEditCarrera.text()
        facultad = self.lineEditFacultadArea.text()
        if not all([cedula, person_name, carrera, facultad]):
            print("Por favor, completa todos los campos.")
            return

        self.detener_camara()
        camera_index = int(self.comboBoxCamaras.currentIndex())
        # Mostrar ventana emergente para ingresar la cédula (usuario)
        cedula_us, ok_cedula = QInputDialog.getText(self, "Registro de
Usuario", "Ingrese su cédula:")
        if not ok_cedula or not cedula_us:  # Verifica si el usuario
canceló o no ingresó la cédula
            QMessageBox.warning(self, "Error", "Debe ingresar una
cédula.")
            return
        # Mostrar ventana emergente para ingresar la contraseña
        password, ok_password = QInputDialog.getText(self, "Registro de
Usuario", "Ingrese su clave:", QLineEdit.Password)
        if not ok_password or not password:  # Verifica si el usuario
canceló o no ingresó la clave
            QMessageBox.warning(self, "Error", "Debe ingresar una
clave.")
            return
        # Recuperar todos los documentos de la colección 'people_data'
        docs = db.collection('Usuarios_Sistema').stream()
        for doc in docs:
            # Obtener el ID del documento (que corresponde a la cédula)
            cedula_var = doc.id
            if cedula_var == cedula_us:
                # Obtener los datos del documento
                data = doc.to_dict()
                decrypted_password =
bcrypt.checkpw(password.encode('utf-8'), data['password'].encode('utf-
8'))
                if decrypted_password:
                    bandera = True
                    self.registrar_accion(self.accion, cedula,
cedula_var)

        if bandera == True:
```

```
            self.registration_thread = RegistrationThread(cedula,
person_name, carrera, facultad, camera_index)
            self.registration_thread.update_progress_signal.connect(sel
f.update_progress)
            self.registration_thread.registration_complete_signal.conne
ct(self.on_registration_complete)
            self.registration_thread.new_frame_signal.connect(self.disp
layImage)
            self.registration_thread.start()
            #reset
            self.lineEditCedula.setDisabled(False)
            self.lineEditCedula.setText("")
            self.lineEditApellidosNombres.setText("")
            self.lineEditFacultadArea.setText("")
            self.lineEditCarrera.setText("")
            self.detener_camara()
        else:
            QMessageBox.warning(self, "Error", "Credenciales
Incorrectas")


    def update_progress(self, message):
        print(message)

    def on_registration_complete(self, message):
        print(message)
        self.iniciar_camara()

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MyApp()
    window.show()
    sys.exit(app.exec_())
```

A continuación se observa el código Python, este código es el principal de ejecutar el código del reconocimiento facial y almacenar la información correspondiente del registro de las personas, además, que actualiza para ver modificaciones o actualizaciones en la base de datos Firebase.

```
import time
import subprocess
import firebase_admin
from firebase_admin import credentials, firestore
import json
import numpy as np
```

```
import pickle
import os
import shutil
# Inicializar Firebase
cred = credentials.Certificate("tesisguadalupe.json")
firebase_admin.initialize_app(cred)
# Conexión a Firestore
db = firestore.client()
# Nombre de la colección y documento
COL_REGISTROS_SISTEMA = 'Registros_Sistema'
COL_ULTIMA_ACTUALIZACION = 'Ultima_Actualizacion'
DOC_ULTIMA_ACTUALIZACION = '1'  # Documento donde está el campo `ID`
# Ruta al script que se va a manejar
SCRIPT_TO_MANAGE = 'reconocimiento.py'
# Iniciar el script secundario
process = subprocess.Popen(['python3', SCRIPT_TO_MANAGE],
stdout=subprocess.PIPE, stderr=subprocess.PIPE)
print("Script iniciado.")
try:
    while True:
        # Obtener el último documento de REGISTROS_SISTEMA ordenado por
timestamp o ID
        lista = []
        registros_ref = db.collection(COL_REGISTROS_SISTEMA)
        all_documents = registros_ref.get()
        for doc in all_documents:
            lista.append(doc.id)
        last_id = lista[-1]  # Cambia 'ID' al campo correcto en tu base
de datos
        # Obtener el campo `ID` del documento en ULTIMA_ACTUALIZACION
        ultima_actualizacion_ref =
db.collection(COL_ULTIMA_ACTUALIZACION).document(DOC_ULTIMA_ACTUALIZACI
ON)
        ultima_actualizacion_doc = ultima_actualizacion_ref.get()
        current_id = ultima_actualizacion_doc.to_dict().get('id') if
ultima_actualizacion_doc.exists else None
        if last_id != current_id:
            print("Se detectaron cambios. Actualizando...")
            # Actualizar el documento en ULTIMA_ACTUALIZACION
            ultima_actualizacion_ref.update({'id': last_id})
            print("Campo 'ID' actualizado en ULTIMA_ACTUALIZACION.")
            # Detener el script secundario
            process.terminate()
            process.wait()
            print("Script detenido.")
            # Crear un diccionario para almacenar toda la información
de las personas
            people_data = {}
```

```
            # Recuperar todos los documentos de la colección
'people_data'
            docs = db.collection('Data_Set').stream()
            for doc in docs:
                # Obtener el ID del documento (que corresponde a la
cédula)
                cedula = doc.id
                # Obtener los datos del documento
                data = doc.to_dict()
                # Deserializar los embeddings desde JSON si están
presentes
                if 'encodings' in data:
                    encodings = json.loads(data['encodings'])
                    # Convertir la lista de listas en arrays de numpy
                    encodings = [np.array(encoding) for encoding in
encodings]
                else:
                    encodings = []
                # Agregar los datos al diccionario
                people_data[cedula] = {
                    'encodings': encodings,
                    'info': {
                        'Nombre': data['info']['Nombre'],
                        'Cédula': data['info']['Cédula'],
                        'Carrera': data['info']['Carrera'],
                        'Facultad': data['info']['Facultad'],}}
            # Guardar el diccionario como un archivo .pickle
            with open('encodings.pickle', 'wb') as f:
                pickle.dump(people_data, f)
            print("Datos guardados en encodings.pickle")
            # Reiniciar el script secundario
            process = subprocess.Popen(['python3', SCRIPT_TO_MANAGE],
stdout=subprocess.PIPE, stderr=subprocess.PIPE)
            print("Script reiniciado.")
        else:
            print("No se detectaron cambios. Continuando...")
        # Esperar 15 minutos antes de verificar nuevamente
        time.sleep(15 * 60)
except KeyboardInterrupt:
    print("Interrupción manual. Deteniendo...")
    process.terminate()
    process.wait()
print("Programa terminado.")
```

A continuación se tiene el código que realiza todo el reconocimiento facial mediante la cámara que enfoca al usuario al momento del ingreso, por lo cual, el código consta

de la solicitud a la cerradura magnética para permitir el acceso y el envió de notificaciones al aplicativo telegram al momento de tener usuarios desconocidos queriendo acceder al lugar. Se observa las librerías, algoritmos y herramientas Python que se utilizaron para el procesamiento y reconocimiento facial.

```
import asyncio  # Necesario para manejar funciones asincrónicas
import cv2
import face_recognition
import pickle
import time
import numpy as np
from sklearn.neighbors import BallTree
import Jetson.GPIO as GPIO
from telegram import Bot, Update
from telegram.ext import Application, CommandHandler, MessageHandler,
filters

# Configuración de los pines GPIO
LED_CONOCIDO = 17  # Cambia al pin que usas
LED_DESCONOCIDO = 27  # Cambia al pin que usas

GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_CONOCIDO, GPIO.OUT)
GPIO.setup(LED_DESCONOCIDO, GPIO.OUT)

# Configuración de Telegram
TELEGRAM_TOKEN = "7560726518:AAFXaKhvXFl8Ga0UkJ1lBPwWRSTBSO_oQgU"  #
Reemplázalo con tu token
CHAT_ID = "-1002451119605"  # Reemplázalo con tu ID de usuario o chat

# Crear el bot de Telegram con un pool de conexiones ajustado
bot = Bot(token=TELEGRAM_TOKEN)
application = Application.builder().token(TELEGRAM_TOKEN).build()

# Cargar los encodings de caras conocidas
print('[INFO] Cargando encodings...')
with open('encodings.pickle', 'rb') as f:
    people_data = pickle.load(f)

# Obtener los encodings conocidos y sus nombres
known_encodings = np.array([enc for data in people_data.values() for
enc in data['encodings']])
known_names = [name for name, data in people_data.items() for _ in
data['encodings']]
print(f"[INFO] Total de caras conocidas: {len(known_names)}")
```

```
# Crear el árbol de búsqueda (BallTree)
tree = BallTree(known_encodings)

# Inicializar captura de video con GStreamer para Jetson Nano (si está
disponible)
print('[INFO] Iniciando captura de video...')
pipeline = (
    "nvarguscamerasrc ! video/x-raw(memory:NVMM), width=1280,
height=720, format=(string)NV12, framerate=30/1 ! "
    "nvvidconv flip-method=0 ! video/x-raw, format=(string)BGRx !
videoconvert ! video/x-raw, format=(string)BGR ! appsink"
)
vs = cv2.VideoCapture(pipeline, cv2.CAP_GSTREAMER) if
cv2.VideoCapture(pipeline).isOpened() else cv2.VideoCapture(0)
time.sleep(2.0)

# Umbral de distancia para reconocer caras
THRESHOLD = 0.45  # Ajustar según sea necesario

# Variables de control de tiempo
last_recognized_time = 0
current_status = None  # "conocido" o "desconocido"

# Función asincrónica para enviar la foto
async def enviar_foto_async(filename, mensaje):
    try:
        with open(filename, 'rb') as photo:
            await bot.send_photo(chat_id=CHAT_ID, photo=photo,
caption=mensaje)
        print(f"[INFO] Foto enviada a Telegram: {filename}")
    except Exception as e:
        print(f"[ERROR] No se pudo enviar la foto: {e}")

# Función principal
async def procesar_video():
    global last_recognized_time, current_status
    try:
        while True:
            ret, frame = vs.read()
            if not ret:
                print("[ERROR] No se pudo capturar el frame.")
                break

            # Reducir resolución para acelerar el procesamiento
            small_frame = cv2.resize(frame, (320, 240))  # Tamaño
reducido para Jetson Nano
            rgb_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
```

```
            # Detección y codificación de caras utilizando el modelo
'cnn' para GPU
            boxes = face_recognition.face_locations(rgb_frame,
model='cnn')
            encodings = face_recognition.face_encodings(rgb_frame,
boxes)

            names = []
            for encoding in encodings:
                # Buscar el vecino más cercano
                dist, idx = tree.query([encoding], k=1)
                name = known_names[idx[0][0]] if dist[0][0] < THRESHOLD
else 'Desconocido'
                names.append(name)

            # Verificar el estado actual (conocido o desconocido)
            if 'Desconocido' in names or not names:  # Si hay un
desconocido o no se detecta ninguna cara
                status = "desconocido"
            else:
                status = "conocido"

            # Cambiar LEDs según el estado
            current_time = time.time()
            if status == "conocido":
                if current_status != "conocido" or current_time -
last_recognized_time > 10:
                    GPIO.output(LED_CONOCIDO, GPIO.HIGH)
                    GPIO.output(LED_DESCONOCIDO, GPIO.LOW)
                    last_recognized_time = current_time
                    current_status = "conocido"
            elif status == "desconocido":
                GPIO.output(LED_DESCONOCIDO, GPIO.HIGH)
                GPIO.output(LED_CONOCIDO, GPIO.LOW)
                if current_status != "desconocido":
                    # Guardar la imagen como archivo temporal
                    filename = f"desconocido.jpg"
                    cv2.imwrite(filename, frame)
                    await enviar_foto_async(filename, "Persona
desconocida detectada")
                current_status = "desconocido"

            # Dibujar resultados en el frame original (no reducido)
            scale_factor = frame.shape[1] / small_frame.shape[1]  #
Para ajustar coordenadas
            for (top, right, bottom, left), name in zip(boxes, names):
                top, right, bottom, left = [int(coord * scale_factor)
for coord in (top, right, bottom, left)]
```

```
                color = (0, 255, 0) if name != 'Desconocido' else (0,
0, 255)
                cv2.rectangle(frame, (left, top), (right, bottom),
color, 2)
                cv2.putText(frame, name, (left, top - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 1)

            # Mostrar el frame
            cv2.imshow('Webcam', frame)

            # Salir al presionar 'q'
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    finally:
        # Liberar recursos
        vs.release()
        cv2.destroyAllWindows()
        GPIO.cleanup()

if __name__ == '__main__':
    asyncio.run(procesar_video())
```

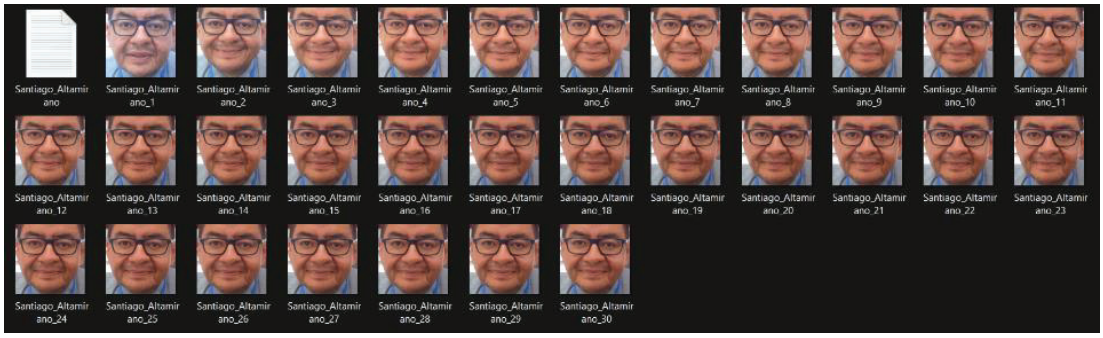
Anexo G. Implementación del prototipo capturas de docentes.

Registro de cada docente en las siguientes imágenes para agregar a la base de datos del sistema, ya que son personas que pasan con frecuencia en el área de Talleres Tecnológicos.

Fotos Ingeniero Jesús Guamán



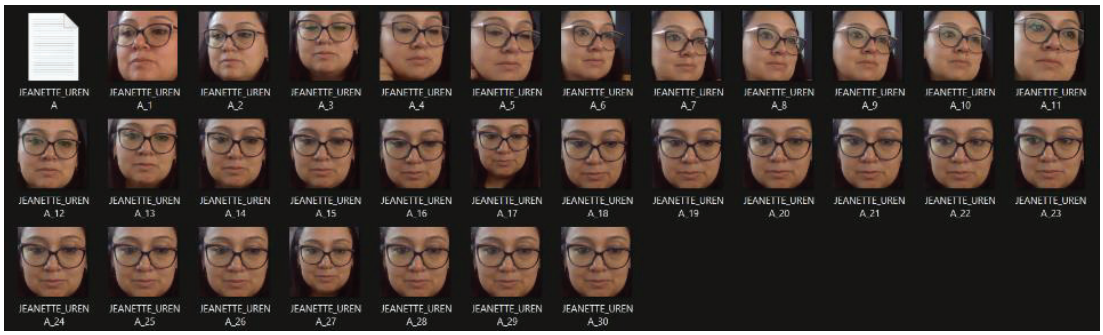
Fotos Ingeniero Santiago Altamirano



Fotos Ingeniero Geovanni Brito



Fotos Ingeniera Jeanette Ureña



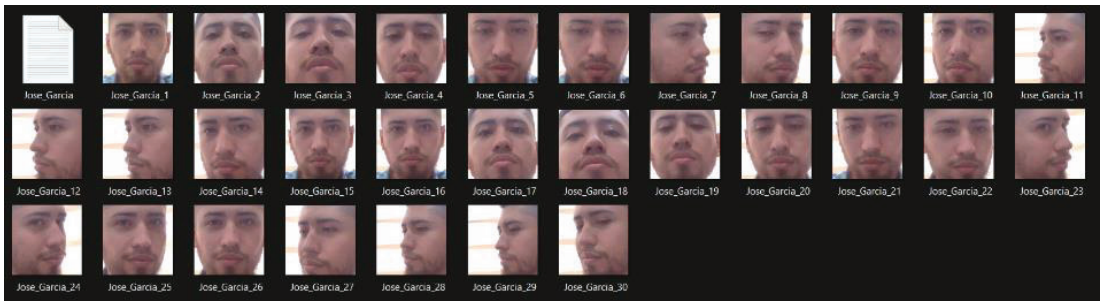
Fotos Ingeniero Franklin Salazar



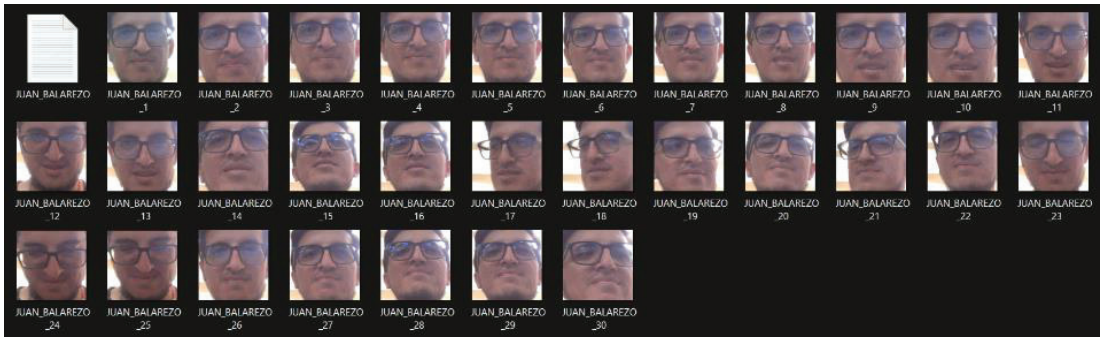
Anexo H. Implementación del prototipo capturas de encargados de Talleres tecnológicos.

Registro de cada ingeniero encargado de Talleres tecnológicos en las siguientes figuras para agregar a la base de datos del sistema, ya que son personas que pasan con frecuencia en el área de Talleres Tecnológicos.

Ingeniero José García



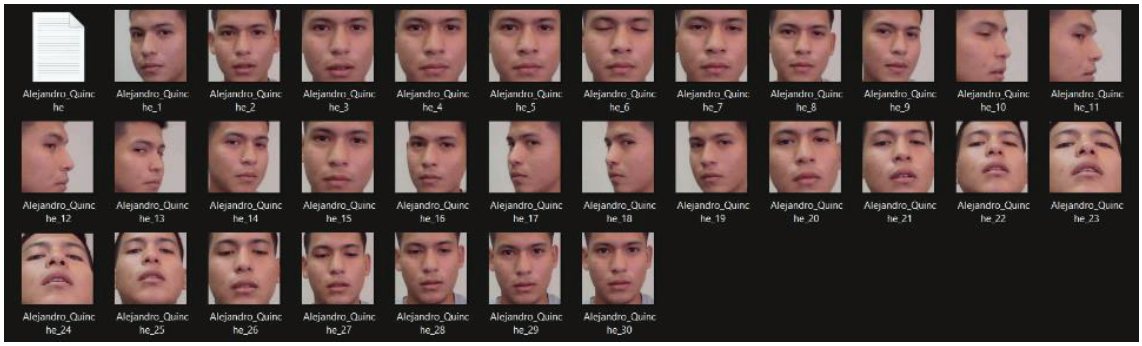
Ingeniero Juan Balarezo



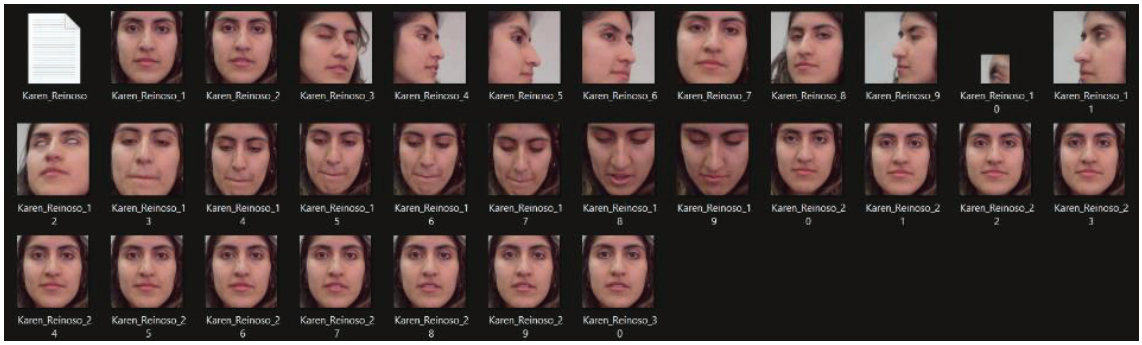
Anexo I. Implementación del prototipo capturas de estudiantes.

Registro de cada estudiante en las siguientes imagenes para agregar a la base de datos del sistema, ya que son personas que pasan con frecuencia en el área de Talleres Tecnológicos.

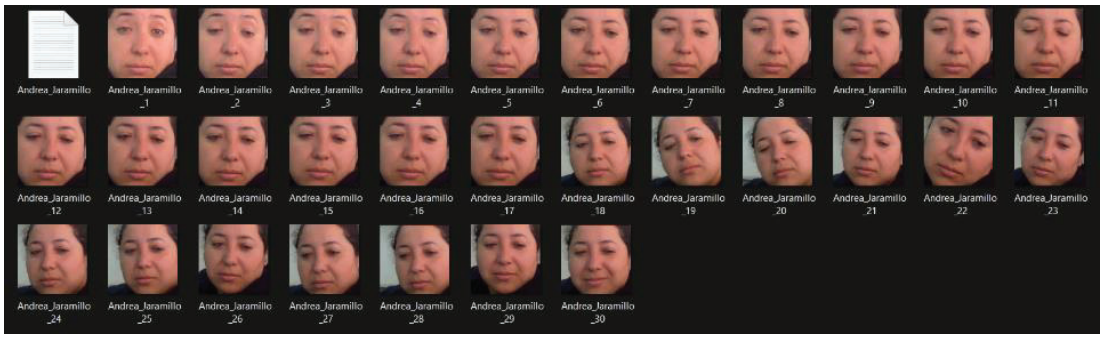
Estudiante Alejandro Quinche



Estudiante Karen Reinoso



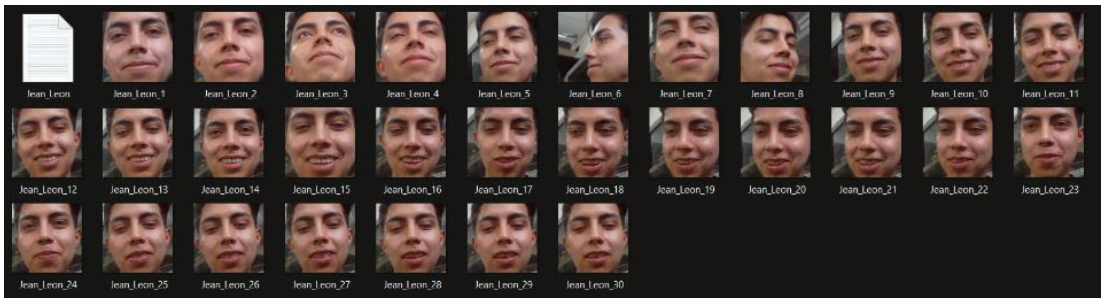
Estudiante Andrea Jaramillo



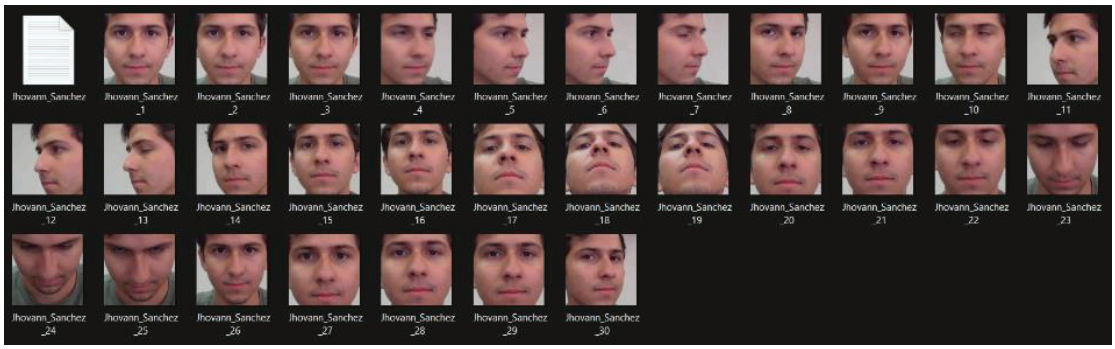
Estudiante Guadalupe Barba



Estudiante Jean Leon



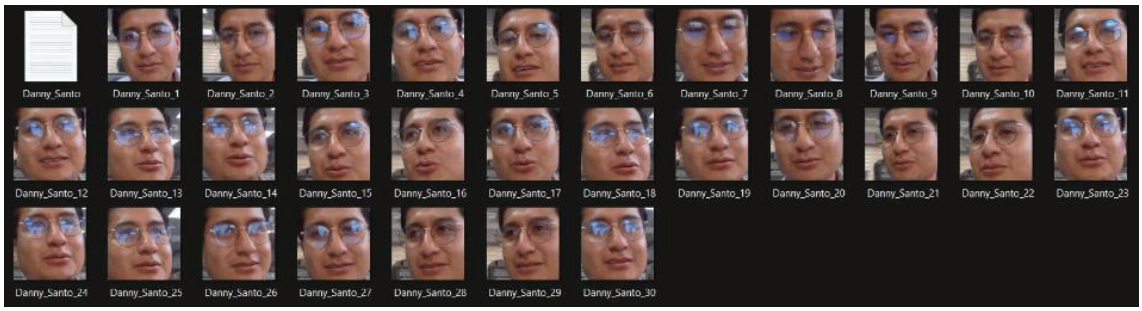
Estudiante Jhovann Sanchez



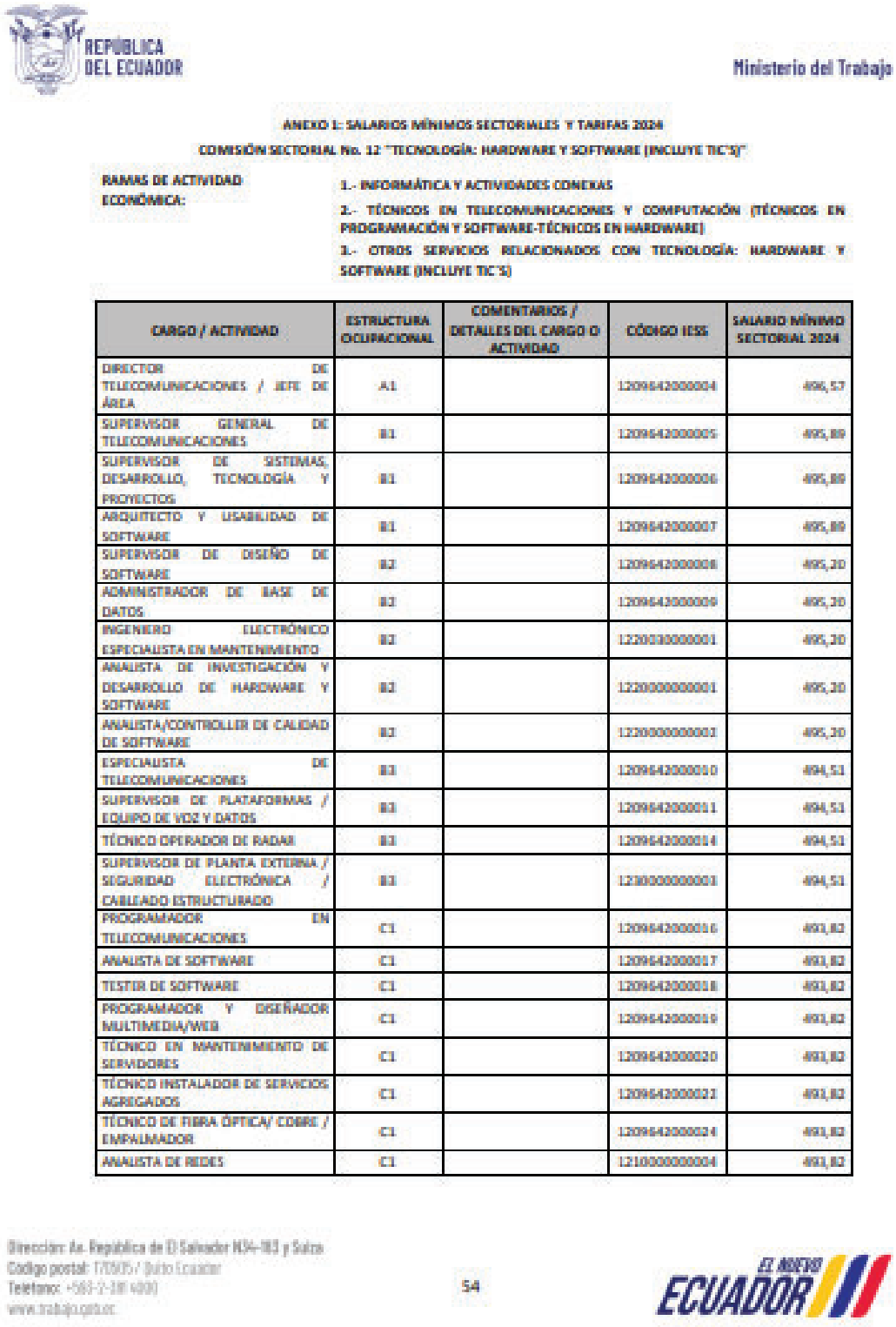
Estudiante Ricardo Tualombo



Estudiante Danny Santo


Danny_Santo
Danny_Santo_1
Danny_Santo_2
Danny_Santo_3
Danny_Santo_4
Danny_Santo_5
Danny_Santo_6
Danny_Santo_7
Danny_Santo_8
Danny_Santo_9
Danny_Santo_10
Danny_Santo_11
Danny_Santo_12
Danny_Santo_13
Danny_Santo_14
Danny_Santo_15
Danny_Santo_16
Danny_Santo_17
Danny_Santo_18
Danny_Santo_19
Danny_Santo_20
Danny_Santo_21
Danny_Santo_22
Danny_Santo_23
Danny_Santo_24
Danny_Santo_25
Danny_Santo_26
Danny_Santo_27
Danny_Santo_28
Danny_Santo_29
Danny_Santo_30

Anexo J. Salarios mínimos sectoriales y tarifas 2024



REPÚBLICA DEL ECUADOR

Ministerio del Trabajo

ANEXO 1: SALARIOS MÍNIMOS SECTORIALES Y TARIFAS 2024

COMISIÓN SECTORIAL No. 12 "TECNOLOGÍA: HARDWARE Y SOFTWARE (INCLUYE TIC'S)"

RAMAS DE ACTIVIDAD ECONÓMICA:

1.- INFORMÁTICA Y ACTIVIDADES CONEXAS

2.- TÉCNICOS EN TELECOMUNICACIONES Y COMPUTACIÓN (TÉCNICOS EN PROGRAMACIÓN Y SOFTWARE-TÉCNICOS EN HARDWARE)

3.- OTROS SERVICIOS RELACIONADOS CON TECNOLOGÍA: HARDWARE Y SOFTWARE (INCLUYE TIC´S)

CARGO / ACTIVIDAD	ESTRUCTURA OCUPACIONAL	COMENTARIOS / DETALLES DEL CARGO O ACTIVIDAD	CÓDIGO IESS	SALARIO MÍNIMO SECTORIAL 2024
DIRECTOR DE TELECOMUNICACIONES / JEFE DE ÁREA	A1		1209642000004	496,57
SUPERVISOR GENERAL DE TELECOMUNICACIONES	B1		1209642000005	495,89
SUPERVISOR DE SISTEMAS, DESARROLLO, TECNOLOGÍA Y PROYECTOS	B1		1209642000006	495,89
ARQUITECTO Y USABILIDAD DE SOFTWARE	B1		1209642000007	495,89
SUPERVISOR DE DISEÑO DE SOFTWARE	B2		1209642000008	495,20
ADMINISTRADOR DE BASE DE DATOS	B2		1209642000009	495,20
INGENIERO ELECTRÓNICO ESPECIALISTA EN MANTENIMIENTO	B2		1220030000001	495,20
ANALISTA DE INVESTIGACIÓN Y DESARROLLO DE HARDWARE Y SOFTWARE	B2		1220000000001	495,20
ANALISTA/CONTROLLER DE CALIDAD DE SOFTWARE	B2		1220000000002	495,20
ESPECIALISTA DE TELECOMUNICACIONES	B3		1209642000010	494,51
SUPERVISOR DE PLATAFORMAS / EQUIPO DE VOZ Y DATOS	B3		1209642000011	494,51
TÉCNICO OPERADOR DE RADAR	B3		1209642000014	494,51
SUPERVISOR DE PLANTA EXTERNA / SEGURIDAD ELECTRÓNICA / CABLEADO ESTRUCTURADO	B3		1230000000003	494,51
PROGRAMADOR EN TELECOMUNICACIONES	C1		1209642000016	493,82
ANALISTA DE SOFTWARE	C1		1209642000017	493,82
TESTER DE SOFTWARE	C1		1209642000018	493,82
PROGRAMADOR Y DISEÑADOR MULTIMEDIA/WEB	C1		1209642000019	493,82
TÉCNICO EN MANTENIMIENTO DE SERVIDORES	C1		1209642000020	493,82
TÉCNICO INSTALADOR DE SERVICIOS AGREGADOS	C1		1209642000022	493,82
TÉCNICO DE FIBRA ÓPTICA/ COBRE / EMPALMADOR	C1		1209642000024	493,82
ANALISTA DE REDES	C1		1210000000004	493,82

Dirección: Av. República de El Salvador N34-183 y Suiza
Código postal: 170505 / Quito-Ecuador
Teletrabajo: +593-2-381 4000
www.trabajo.gob.ec

54

EL NUEVO ECUADOR

Figura 47. Salarios mínimos sectoriales y tarifas 2024