UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE INGENIERÍA ELECTRÓNICA

DISPOSITIVO DE SEGURIDAD BASADO EN RECONOCIMIENTO FACIAL Y LECTOR DE TARJETAS RFID

Trabajo de titulación previo a la obtención del

Título de Ingeniero Electrónico

AUTOR: Napoleón Enrique Rea Flores

TUTOR: Juan Carlos Domínguez Ayala

Quito – Ecuador

2026

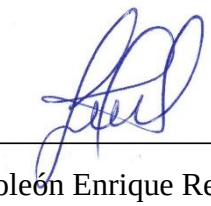
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Napoleón Enrique Rea Flores con documento de identificación N°1715852719 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 23 de febrero de 2026

Atentamente,



Napoleón Enrique Rea Flores
1715852719

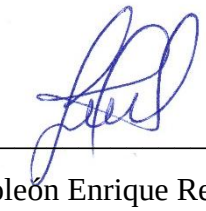
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Napoleón Enrique Rea Flores con documento de identificación No. 1715852719, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor de la proyecto técnico: "Dispositivo de seguridad basado en reconocimiento facial y lector de tarjetas RFID", la cual ha sido desarrollada para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 23 de febrero de 2026

Atentamente,



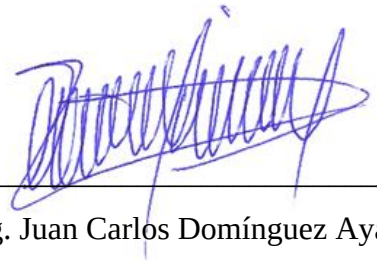
Napoleón Enrique Rea Flores
1715852719

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Carlos Domínguez Ayala con documento de identificación N° 1713195590, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISPOSITIVO DE SEGURIDAD BASADO EN RECONOCIMIENTO FACIAL Y LECTOR DE TARJETAS RFID, realizado por Napoleón Enrique Rea Flores, con documento de identificación N° 1715852719, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 23 de febrero de 2026

Atentamente,



Ing. Juan Carlos Domínguez Ayala, MsC.
1713195590

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Este trabajo desarrolla un sistema inteligente de control de acceso basado en Raspberry Pi 4, enfocado en mejorar de forma efectiva la seguridad y el control de áreas restringidas mediante tecnologías de identificación automática. La solución combina reconocimiento facial, verificación con tarjeta RFID, control físico de acceso mediante un relé y notificaciones en tiempo real a través de Telegram, logrando un sistema integral y fácil de supervisar. Para su implementación, se creó una base de datos facial con 300 imágenes del usuario autorizado, lo que permitió entrenar el modelo y aumentar la precisión del reconocimiento. El acceso se valida en dos niveles de seguridad: primero mediante reconocimiento facial y posteriormente mediante la tarjeta RFID. Cada intento de ingreso, tanto autorizado como fallido, genera una notificación automática al administrador, facilitando un monitoreo remoto e inmediato. Las pruebas realizadas demostraron un desempeño eficiente en distintos escenarios, reduciendo significativamente el riesgo de accesos no autorizados y garantizando un control confiable, lo que confirma que la integración de estas tecnologías ofrece una solución segura, atractiva y de bajo costo para entornos que requieren vigilancia constante y mayor nivel de seguridad.

Palabras clave: Reconocimiento facial, Raspberry Pi, Control de acceso, RFID, Telegram.

ABSTRACT

This work presents the development of an intelligent access control system based on the Raspberry Pi 4, aimed at effectively enhancing the security and management of restricted areas through automatic identification technologies. The proposed solution integrates facial recognition, RFID card verification, physical access control via a relay, and real-time notifications through the Telegram application, resulting in a comprehensive and easily monitored system. For its implementation, a facial database consisting of 300 images of the authorized user was created, enabling model training and improved recognition accuracy. Access validation is performed through a two-level security process, with facial recognition serving as the first authentication layer and RFID card verification as the second. Each access attempt, whether authorized or unsuccessful, automatically triggers a notification to the system administrator, allowing immediate and remote monitoring. Experimental results demonstrated efficient performance under various access scenarios, significantly reducing the risk of unauthorized entry and ensuring reliable control, thus confirming that the integration of these technologies provides a secure, cost-effective, and practical solution for environments requiring continuous surveillance and enhanced security.

Keywords: Facial recognition, Raspberry Pi, Access control, RFID, Telegram.

INTRODUCCIÓN

La seguridad y el control de acceso a áreas restringidas se han convertido en un aspecto fundamental en distintos entornos, como instituciones, empresas y espacios residenciales. Por lo cual, diferentes sistemas tradicionales que dependen solo de llaves o tarjetas carecen de monitoreo y control de acceso, lo que lleva al desarrollo e implementación de soluciones científicas y tecnológicas eficientes y seguras.

Los sistemas de identificación automática han ganado protagonismo por la facilidad para incorporar diversos niveles de seguridad. Al utilizar tecnologías como reconocimiento facial y tarjetas RFID al mismo tiempo, permite una mejor identificación de las personas, disminuyendo ingresos indebidos y a la vez aumentando la confiabilidad y seguridad.

El reconocimiento facial permite identificar en tiempo real a los usuarios por sus rasgos biométricos únicos. Al incorporarse a una comprobación por tarjetas RFID, permite mejorar sustancialmente la seguridad del sistema al aceptar a los usuarios registrados y rechazar a los no registrados.

Este enfoque permite al trabajo que se implemente un sistema inteligente que integra reconocimiento facial, validación por RFID, control físico de acceso mediante un relé y notificaciones en tiempo real a través de Telegram, fortaleciendo la seguridad y el monitoreo del ingreso. Para lo cual una tarjeta Rasberry Pi 4 se perfila como la mejor opción por su amigable lenguaje de programación, su bajo costo y su potente capacidad de procesamiento.

CAPÍTULO 1

1.1. Problema

La inseguridad se ha convertido en una preocupación importante a nivel internacional y nacional, ya que afecta directamente la calidad de vida cotidiana y la sensación de seguridad. El auge del crimen organizado, la expansión de los métodos de robo y el uso de nuevas tecnologías por parte de los grupos criminales ponen de relieve las deficiencias de los sistemas de seguridad tradicionales. Frente a esta situación, la seguridad ya no puede entenderse únicamente como una barrera física, sino como un sistema inteligente capaz de anticipar, identificar y responder ante posibles amenazas (Bravo & Noroño , 2025).

En este contexto internacional, diversos informes especializados señalan un aumento sostenido de los mercados criminales en distintas regiones del mundo, especialmente en países en desarrollo. El Informe Global contra el Crimen Organizado Transnacional (GITOC), presentado en septiembre de 2023, ubica al Ecuador dentro del top 10 países con mayor crecimiento de criminalidad a nivel mundial, compartiendo esta preocupante clasificación con naciones como Myanmar, Colombia, México y Nigeria (Primicias.ec, 2024). Esta posición refleja no solo el impacto del crimen organizado transnacional, sino también la vulnerabilidad de los sistemas de seguridad convencionales (Carpio & Tobar, 2025).

La situación nacional se evidencia claramente en las estadísticas de denuncias por delitos. Entre enero y junio de 2023 se registraron 36.833 denuncias a nivel nacional, de las cuales 12.267 correspondieron a la ciudad de Guayaquil, mientras que 6.660 se reportaron en Quito, representando aproximadamente el 18 % del total nacional. En casos de robo a viviendas, únicamente en Quito contabilizaron 480 denuncias durante el mismo periodo, lo que demuestra que los domicilios siguen siendo objetivos frecuentes de los delincuentes (Primicias.ec, 2024).

1.2. Antecedentes

La situación se platea bajo la siguiente perspectiva ¿cómo proteger de manera más efectiva los hogares de muchas familias y sus espacios de trabajo frente a un entorno cada vez más inseguro? Existen métodos tradicionales de control de acceso, es el caso de cerraduras de acero y madera o tarjetas convencionales, resultan insuficientes ante

técnicas de vulneración sofisticadas, como la suplantación de identidad o el acceso inseguro.

Esta problemática se puede solucionar mediante el uso de nuevas tecnologías como el Internet de las Cosas (IoT) y el aprendizaje automático se presenta como una alternativa innovadora para fortalecer los sistemas de seguridad. El integrar dispositivos inteligentes permite asemejar a los usuarios, registrar eventos y notificar en tiempo real a los propietarios sobre cualquier intento de ingreso. Sin embargo, pese a la existencia de soluciones parciales en el mercado, no se dispone de un sistema integral de control de acceso que combine simultáneamente reconocimiento facial, tarjetas RFID y verificación mediante plataformas de mensajería, lo cual limita el nivel de seguridad alcanzable.

En esta situación nace la necesidad de generar un dispositivo de control de acceso inteligente e integral, capaz de unificar múltiples mecanismos de autenticación y comunicación, con el objeto de elevar el nivel de defensa en residencias y oficinas, contribuyendo una solución tecnológica que reconozca de manera efectiva a las circunstancias reales de inseguridad que se encuentra el Ecuador.

1.3. Importancia y alcances

El valor de este proyecto reside en reforzar la seguridad de hogares, oficinas y espacios comerciales mediante la integración de mecanismos de autenticación multifactor. En situaciones donde los sistemas de control de acceso tradicionales pueden ser vulnerables a la clonación, el robo de identidad o el ingreso no autorizado, la combinación del reconocimiento facial con la validación RFID aumenta significativamente la confianza del sistema, reduciendo el riesgo de intrusión.

El proyecto permite tener un monitoreo constante y utilizando la aplicación Telegram notificaciones en tiempo real. Al ser Raspberry Pi 4 una plataforma amigable permite incorporar visión artificial, machine Learning (aprendizaje automático) e IoT (Internet de las cosas), convirtiéndose en una alternativa fiable, económica y asequible para ser comercializada.

1.4. Delimitación

Delimitación específica: El control de acceso para pequeños entornos, domicilios, oficinas o sitios de comercio; mediante un único punto de acceso físico gestionado por

un relé. Las pruebas del prototipo se realizan en un entorno controlado que imita las condiciones de uso reales.

Delimitación temporal: El proyecto se desarrolla en las fases de diseño, ejecución y unificación de las medidas individuales del sistema como es el reconocimiento facial, autenticación RFID, control de relés y respuestas de mensajes de Telegram, así como pruebas funcionales para validar el rendimiento general del sistema.

Delimitación técnica: El método se efectiviza en el programa Raspberry Pi 4B monopolizando el lenguaje de programación Python y bibliotecas de visión artificial como OpenCV y el clasificador Haar Cascade/LBPH. Se integra un lector RFID como factor de autenticación secundario y se utiliza Telegram para las notificaciones. El alcance del proyecto no incluye la integración con la infraestructura en la nube, el desarrollo de aplicaciones móviles dedicadas, la gestión a gran escala de accesos múltiples ni la implementación de mecanismos avanzados de ciberseguridad empresarial más allá de las medidas básicas de protección de datos descritas en el documento.

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar un dispositivo de acceso inteligente usando el paradigma de solicitud-respuesta mediante internet de las cosas y el aprendizaje automático para la seguridad de casas, oficinas y locales.

1.5.2. Objetivos específicos

- Identificar las necesidades de los usuarios de un dispositivo de acceso doméstico inteligente mediante la realización de un análisis de opciones disponibles en mercado y la información en línea para la determinación de los requerimientos mínimos del dispositivo de acceso inteligente.
- Diseñar un dispositivo de acceso inteligente que cumpla con los requerimientos mínimos para la verificación de su viabilidad técnica.
- Diseñar e implementar un sistema de acceso inteligente para verificar su correcto funcionamiento.
- Evaluar el costo del sistema para determinar si es adecuado para su comercialización en el futuro.

CAPÍTULO 2
FUNDAMENTOS TEÓRICOS

2.1.Seguridad y control de acceso

Se hace necesario contar con un sistema de seguridad para la protección de las personas, por el aumento en las tasas de delincuencia, se hace imprescindible para protegerla vida y los bienes materiales (DMQ, 2023). El sistema de ingreso por reconocimiento facial se convierte en la mejor alternativa para mejorar la seguridad e impedir el ingreso a personas no deseadas. (Correa, 2016). La evolución en estos sistemas de acceso ha pasado de simples cerraduras mecánicas, a dispositivos electrónicas capaces de identificar rostros y brindar notificaciones en tiempo real (Law, 2017).

2.2.Sistemas de seguridad electrónica

El conjunto de dispositivos creados para detectar y responder a una posible intrusión no autorizada se los conoce como sistemas de seguridad. Para su fin se utiliza elementos como sensores biométricos, cámaras, cerraduras eléctricas y aplicaciones de mensajería instantánea. (Sánchez, 2024). Los diferentes dispositivos de seguridad utilizan accesos biométricos y las notificaciones remotas en tiempo real, un claro ejemplo de tecnologías inteligentes (Washington & Dilan , 2023).

2.3.Reconocimiento facial

Esta tecnología analiza el rostro de una persona buscando los rasgos únicos que cada individuo posee. Al utilizar una cámara para recolectar los datos del rostro elimina del proceso el contacto físico y se convierte en un método sencillo y cómodo para los usuarios que utilizan este sistema de seguridad. (Terrazas, 2024). El proceso se enlista en la Figura 1 mostrada a continuación:

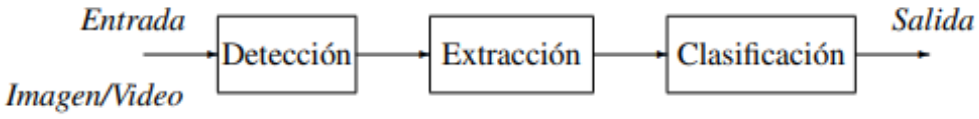


Figura 1:Ciclo del reconocimiento facial.

La detección facial conlleva analizar y procesar imágenes que permitan reconocer solo el rostro de una persona e identificar las características que se van a utilizar para poder

diferenciar con precisión de otros rostros y poder continuar con las siguientes etapas (Venturini & Garay, 2024).

Se extrae los principales rasgos faciales como forma de las cejas, longitud entre los ojos y forma de la nariz, etc. para registrarlos en el sistema y diseñar un único perfil biométrico (Coronel, 2021). Como último se compara los datos obtenidos con la base de datos para comprobar si las características del rostro capturado coinciden con el registrado (W. Zhao, 2003).

Los histogramas de patrones binarios (LBPH) es uno de los procesos más eficientes utilizado para el reconocimiento facial, al contrario de métodos que usan reden neuronales que ofrecen mejor precisión, pero ocupan más recursos computacionales para sus cálculos (Niola & Sanango, 2019).

El uso de mascarillas, gafas, cambios físicos e incluso la iluminación afectan en la precisión del sistema, para lo que se recomienda fortalecer el sistema con otras formas adicionales de autenticación (Bravo & Noroño , 2025).

2.4.Autenticación múltiple en sistemas de seguridad

Utilizar solo un sistema de autentificación puede ser fácilmente vulnerada es aquí donde entra el 2Fa (doble factor), para lo que se utiliza otro método de autentificación con esto reducir el posible ingreso no autorizado y permita aumentar el nivel de seguridad (Areitio, 2008).

2.5.Internet de las Cosas (IoT) aplicado a la seguridad

Permite en la seguridad el intercambio y administración de datos entre dispositivos físicos como control de acceso, sensores, cámaras y plataformas en tiempo real, utilizando internet como medio de comunicación. Dando a los usuarios la facilidad con mayor agilidad a notificaciones de posibles ingresos no autorizados Fúnez (2022).

2.6.Mensajería móvil y notificación de eventos

Una opción eficiente para mejorar los sistemas de seguridad es por medio de notificaciones en tiempo real a través de mensajería por diferentes plataformas, permitiendo una gestión y reacción más eficiente al identificar la identidad de los usuarios que piden acceso a un lugar determinado Sánchez (2024).

2.7.Importancia de los sistemas inteligentes de seguridad en viviendas

El salto evolutivo de los sistemas convencionales a sistemas de seguridad inteligentes a marcado un antes y después debido a la tecnología nueva que incorporan, aumentando ampliamente la seguridad al incorporar reconocimiento facial, tarjetas RFID e IoT y convirtiéndose en un sistema confiable, amigable y escalable para hogares (Coronel, 2021).

Para convertirlo en un sistema de seguridad más robusto y pueda solucionar los problemas actuales, fusionar los sistemas se convierte en el recurso más acertado para disminuir los accesos no autorizados (Washington & Dilan , 2023).

2.8.Raspberry Pi 4B

La Raspberry Pi 4B (RPi 4B) es una computadora de placa única (Single Board Computer, SBC) de alto rendimiento y bajo costo, razones por las cuales se usa en muchos proyectos relacionados con la robótica y la automatización.

Figura 2: Raspberry Pi 4B

La RPi 4B está compuesta por varios elementos:

- Procesador: Broadcom BCM2711, de cuatro núcleos ARM Cortex-A72 capaces de funcionar hasta a 1,5 GHz. El BCM2711 dispone de una GPU 3D VideoCore VI que funciona a 500MHz.
- Memoria RAM: De 1GB, 2GB o 4GB LPDDR4 dependiendo del modelo.

- Conectividad inalámbrica: En cuanto a conectividad inalámbrica, la RPi 4B dispone de WLAN 802.11b/g/n/ac a 2,4 GHz y 5,0 GHz. Además, tiene Bluetooth 5.0.
- Puerto Ethernet: Dispone de un puerto Gigabit Ethernet
- Puertos USB: La RPI tiene 2 puertos de USB 2.0 y otros 2 de USB 3.0.
- Pines GPIO: Se dispone de una cabecera de 40 pines GPIO (GeneralPurpose Input/Output) en los que se podrán conectar diversos componentes electrónicos.
- HDMI: Se tienen dos puertos micro HDMI que soportan hasta una resolución de 4K a 60fps.
- Puerto MIPI DSI: Es un puerto con el que su puede conectar un monitor.
- Puerto MIPI CSI: Es el puerto que se usará para conectar el módulo de cámara a la RPi. 17
- A/V: Puerto Jack hembra de 3.5mm para salida de audio y entrada de voz.
- Almacenamiento: La RPI dispone de un slot para tarjetas microSD para el sistema operativo y almacenamiento de datos.
- Alimentación: Dispone de un puerto USB-C para alimentación a 5V DC. Adicionalmente, se puede alimentar mediante GPIO y mediante PoE (Power over Ethernet). (Fúnez, 2022)

2.9.Módulo de Cámara

El módulo de cámara de Raspberry Pi es una cámara ligera y portátil que se conecta al puerto CSI (Camera Serial Interface) de la Raspberry Pi mediante un cable tipo Ribbon.

Esta conexión usa el protocolo de interfaz serial de cámara MIPI (MIPI CSI).

Figura 3: Módulo de cámara de Raspberry Pi.

Este módulo ofrece una buena calidad de imagen y varias ventajas (StackExchange, 2016)

- Menor carga en la CPU: El protocolo USB que usan las cámaras web convencionales, tiene un sobrecoste computacional relativamente alto por su complejidad. El protocolo MIPI CSI, en cambio, tiene un coste computacional reducido. Esto, en un ordenador, no es algo relevante, dado que hoy en día los ordenadores tienen la suficiente potencia como para que este coste computacional no tenga un impacto en el rendimiento. Este hecho cambia cuando se pasa a trabajar con una Raspberry Pi, más limitada en capacidad de procesamiento, comparativamente. Para ahorrar poder computacional en este aspecto, es conveniente usar un módulo de cámara siempre que sea posible.
- Soporte hardware: Si bien es cierto que el módulo de cámara no provee codificación hardware por sí solo, se conecta directamente a la GPU (Graphics Processing Unit) que sí lo hace. Si se considera la alternativa con cámara web USB, se ve que, si se quiere codificación hardware, la cámara ha de hacerlo internamente.

2.10. Arquitectura Escalable

Cuando existe exceso de pedidos en temporadas de alta demanda sea por descuentos o venta de entradas en las diferentes plataformas digitales, es en estas circunstancias donde la escalabilidad se vuelve indispensable (Coronel, 2021).

La escalabilidad en un sistema permite y garantiza las futuras actualizaciones como agregar nuevas funciones sin necesidad de remplazar todo. Esto garantiza que el sistema no quede obsoleto e imposibilite el uso del dispositivo en un futuro, sino que provee sostenibilidad, flexibilidad y capacidad de crecimiento, al incorporar más usuarios y nuevos dispositivos IoT (Venturini & Garay, 2024).

CAPÍTULO 3
MARCO METODOLÓGICO

3.1. Modelo de Investigación

La seguridad y control de accesos a oficinas, casas, departamentos, locales comerciales, etc. se ha convertido en una prioridad por el nivel de inseguridad existente, por lo que esta investigación tiene como propósito diseñar e implementar un dispositivo electrónico para resolver esta problemática.

El trabajo aborda una perspectiva cuantitativa, donde se analizan datos con pruebas operativas, así como el tiempo de respuesta, en el proceso para reconocer un rostro y la seguridad en la fase de acceso.

Esta investigación es descriptiva, ya que define el funcionamiento y los componentes del dispositivo de seguridad. Además, se llevarán a cabo pruebas en un laboratorio controlado para verificar su correcto funcionamiento.

3.2. Población y muestra

Para realizar el análisis del dispositivo se tomó a la población registrada y los acontecimientos de accesos originados por el sistema de acceso inteligente (fase de pruebas). Ya que no es destinada a una población estadística, sino a la evaluación y correcto funcionamiento del dispositivo.

3.3. Recolección de Información

La recolección de información se realiza mediante diversas técnicas e instrumentos, con el objetivo de obtener datos confiables para el desarrollo y validación del dispositivo. Entre las principales técnicas utilizadas se encuentran:

- Revisión bibliográfica, empleada para recopilar información teórica sobre sistemas de seguridad electrónica, reconocimiento facial, autenticación múltiple, Internet de las Cosas y arquitecturas escalables.
- Observación directa, aplicada durante la implementación y las pruebas del sistema, lo que permitió identificar el comportamiento del dispositivo ante distintos escenarios de acceso.

- Pruebas experimentales, realizadas con el propósito de evaluar el desempeño del sistema, considerando aspectos como la detección facial, el tiempo de respuesta, la tasa de aceptación y rechazo.

3.4. Procesamiento

Los datos recopilados se organizan, analizan e interpretan de manera sistemática. En primer lugar, los resultados de las pruebas del sistema se dividen según el tipo de acceso registrado: reconocimiento facial, comprobación RF y fallido. Luego, se elabora un análisis utilizando una matriz de confusión, que describe los verdaderos positivos, los verdaderos negativos, los falsos positivos y los falsos negativos, por lo que se lleva a cabo una evaluación objetiva de la precisión y confiabilidad del sistema.

3.5. Términos Generales

Se definen los términos generales utilizados para el dispositivo de seguridad:

- **Acceso**: Se denomina acceso al proceso mediante el cual una persona intenta ingresar a un espacio físico protegido, utilizando uno o varios mecanismos de autenticación definidos por el sistema de seguridad.
- **Usuario Autorizado o No Autorizados:** Persona preinscrita en el sistema de control de acceso, cuyos datos biométricos faciales o identificadores RFID se almacenan en la base de datos, lo que le otorga acceso a un área segura. Caso contrario será rechazada convirtiéndose en un usuario no autorizado.
- **Evento de Seguridad:** Registro creado por el sistema de cualquier intento de acceso, exitoso o fallido, que se conserva para su posterior monitoreo y análisis.
- **Sistema IoT:** Conjunto de dispositivos interconectados capaces de recopilar, procesar y enviar información a través de una red, lo que permite el control y monitoreo remotos de los sistemas de seguridad.

3.6. Análisis del proceso de control de acceso

El sistema se activa cuando una persona presiona el pulsador del punto de acceso; en ese momento, el dispositivo captura una imagen de su rostro y, opcionalmente, solicita la verificación de la tarjeta RFID.

Los siguientes objetivos se lograrán con su implementación:

- Aumentar la seguridad y automatizar el registro de ingresos y salidas de usuarios autorizados.
- Notificar al administrador del sistema ante intentos de acceso no autorizados.
- Reducir la dependencia de mecanismos físicos tradicionales susceptibles a fallos o manipulaciones.

3.7. Análisis de roles de usuarios del sistema

Aunque el sistema de seguridad puede ser utilizado por múltiples personas, no todos los usuarios interactúan de la misma forma ni poseen el mismo nivel de privilegios. Por esta razón, se definen diferentes roles de usuario, los cuales permiten una correcta administración del sistema y una gestión adecuada de los accesos.

Los roles considerados en el dispositivo de seguridad son los siguientes:

- **Administrador**: Usuario con privilegios totales sobre el sistema. Puede registrar y eliminar usuarios, revisar el historial de accesos y recibir alertas.
- **Usuario autorizado**: Persona registrada que puede acceder al espacio protegido mediante reconocimiento facial y/o tarjeta RFID.
- **Visitante**: Usuario con permisos temporales o limitados.

La Tabla presenta la clasificación de roles y su nivel de prioridad dentro del sistema.

Tabla 1. Roles de usuarios del sistema

Rol de usuario	Nivel de acceso	Prioridad
Administrador	Total	Alta
Usuario autorizado	Parcial	Media
Visitante	Limitado	Baja

Elaborado por: Rea Napoleón

3.8. Análisis de criticidad y clasificación de eventos

No todos los eventos registrados por el sistema poseen el mismo nivel de importancia. Por ello, se realiza una clasificación basada en la criticidad del evento, considerando tanto el tipo de usuario como el tipo de acceso realizado.

Los eventos más comunes identificados en el sistema son:

Eventos de acceso exitoso

- Reconocimiento facial validado
- Tarjeta RFID autorizada
- Acceso combinado facial–RFID

Eventos de acceso fallido

- Rostro no reconocido
- Tarjeta RFID no registrada
- Intentos repetidos de acceso no autorizado

Eventos críticos

- Múltiples intentos fallidos consecutivos
- Acceso fuera de horarios permitidos
- Manipulación del dispositivo

Esta clasificación de eventos permite priorizar alertas, activar notificaciones inmediatas y apoyar la toma de decisiones del administrador del sistema, fortaleciendo la gestión de la seguridad y la respuesta ante incidentes.

3.9. Seguridad de la información

Para proteger la información sensible almacenada en el sistema, se implementó mecanismos de cifrado en una base de datos local alojada en la Raspberry Pi, que contiene los modelos faciales y los códigos UID de las tarjetas RFID. Asimismo, la comunicación entre la ESP32-CAM y la Raspberry Pi se protegió por un protocolo seguro para evitar la interceptación, alteración o acceso no autorizado a la información transmitida.

3.10. Evaluación comparativa del sistema propuesto

Para validar la efectividad del dispositivo desarrollado, se realiza una comparación conceptual entre el sistema propuesto y los métodos tradicionales de control de acceso, tales como cerraduras mecánicas y sistemas basados únicamente en tarjetas.

Tabla 2. Comparativa general de sistemas de control de acceso

Característica	Sistema tradicional	Sistema propuesto
Reconocimiento facial	No	Sí
Tarjeta RFID	Opcional	Sí
Registro automático	No	Sí
Notificaciones en tiempo real	No	Sí
Integración IoT	No	Sí
Escalabilidad	Limitada	Alta

Elaborado por: Rea Napoleón

3.11. Desarrollo

En el presente capítulo se describe de manera detallada el desarrollo e implementación del sistema de control de acceso propuesto, basado en una Raspberry Pi 4, el cual integra reconocimiento facial mediante OpenCV, verificación por tarjeta RFID, activación de un relé y notificaciones en tiempo real a través de la aplicación Telegram. El desarrollo comprende la programación del sistema, la creación de la base de datos facial, el entrenamiento del modelo de reconocimiento, la integración de los módulos de seguridad y la validación del funcionamiento del sistema.

3.12. Entorno de desarrollo y configuración del sistema

El sistema fue desarrollado sobre la plataforma Raspberry Pi 4, utilizando el sistema operativo Raspberry Pi OS. El lenguaje de programación empleado fue Python, debido a su compatibilidad con bibliotecas de visión artificial y control de hardware, utilizando el entorno de desarrollo Thonny IDE.

Para la implementación del sistema se usaron los siguientes recursos:

- OpenCV, para el procesamiento de imágenes y reconocimiento facial.
- Haar Cascade, para la detección de rostros en tiempo real.
- NumPy, para el manejo de datos numéricos.
- RPi.GPIO, para el control del pulsador y el relé.

- Biblioteca RFID, para la lectura y validación de tarjetas.
- API de Telegram, para el envío automático de notificaciones.

3.13. Desarrollo del módulo de reconocimiento facial

3.13.1. Fabricación de la base de datos facial

La primera etapa, es la construcción de la base de datos. Se programó un módulo que captura las fotografías del rostro de la persona que será autorizada en el sistema, con una cámara conectada a la Raspberry Pi, se capturaron 300 fotos que se guardan en una carpeta en la memoria de la Raspberry Pi, con la programación que se observa en el **Anexo 1**. Este proceso se utilizó con la finalidad de tener una base de datos con la mayor cantidad de información y mejorar la exactitud del reconocimiento facial.

3.13.2. Preparación del modelo de reconocimiento facial (entrenamiento)

Con la formación de la base de datos (ver **Anexo 2)**. La función del programa en esta etapa es analizar las imágenes, clasificando las características faciales, la forma y contornos principales, la separación entre los ojos, con el uso de algoritmos de reconocimiento facial (OpenCV).

3.13.3. Reconocimiento facial en tiempo real mediante OpenCV y Haar Cascade

Con el modelo entrenado, el sistema entra en modo de reconocimiento facial en tiempo real. Mediante la librería Haar Cascade, como se ve su código en el **Anexo 3**, el sistema detecta automáticamente el rostro de la persona que intenta acceder y compara dicha información con la base de datos entrenada. Si el rostro no se encuentra registrado, el sistema niega el acceso de forma inmediata.

3.14. Activación del sistema y control de acceso físico

3.14.1. Activación del sistema mediante pulsador

El sistema incorpora un pulsador físico que permite iniciar el proceso de reconocimiento facial. Este mecanismo evita activaciones innecesarias del sistema y garantiza que el proceso de acceso solo se ejecute cuando el usuario lo solicita de forma intencional.

3.14.2. Verificación de acceso mediante tarjeta RFID

Cuando el rostro es reconocido correctamente, el sistema solicita una segunda validación a través de una tarjeta RFID. El lector RFID compara el identificador único (UID) de la tarjeta con los registros almacenados en el sistema. Si la tarjeta es válida, se autoriza el acceso; caso contrario, el sistema niega el ingreso.

3.14.3. Activación del relé y acceso autorizado

Una vez superadas las dos validaciones de seguridad (reconocimiento facial y RFID), el sistema activa un relé que permite el acceso físico al área protegida, representando la apertura de una puerta o la habilitación de un mecanismo de seguridad.

3.15. Integración del sistema con la plataforma Telegram

Con el fin de proporcionar notificaciones en tiempo real sobre los eventos de acceso al sistema, se integró la plataforma de mensajería instantánea Telegram como medio de comunicación entre el dispositivo de seguridad y el usuario administrador. Esta integración permite informar de manera inmediata sobre accesos autorizados e intentos de ingreso fallidos.

3.15.1. Búsqueda e inicio del bot administrador (@BotFather)

El primer paso para la integración consistió en buscar el chatbot oficial de Telegram denominado @BotFather, el cual es la herramienta proporcionada por la plataforma para la creación y administración de bots.

Este bot permite generar nuevos bots, asignarles nombres, obtener tokens de acceso y gestionar su configuración.

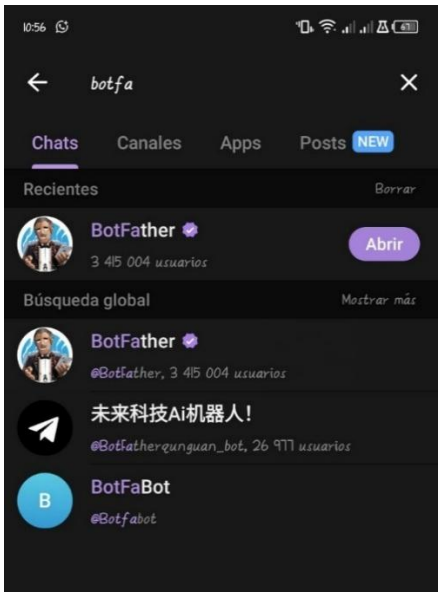


Figura 4: Busqueda del bot administrador.

3.15.2. Inicio de la conversación con @BotFather

Una vez localizado el chatbot @BotFather, se ingresó al chat y se presionó la opción INICIAR, lo que habilita la interacción con el sistema y permite el uso de comandos para la creación de un nuevo bot. Es necesario para activar el menú de comandos y continuar con el proceso de configuración.

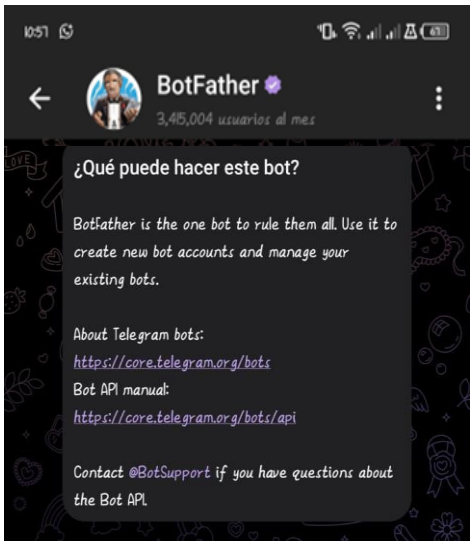


Figura 5: Inicio de conversación con @BotFather.

3.15.3. Ejecución del comando /start

Posteriormente, se digitó el comando **/start**, el cual inicializa formalmente la sesión con @BotFather.

Este comando permite acceder a las funciones disponibles para la gestión de bots y prepara el entorno para la creación de un nuevo bot personalizado.

Figura 6: Comando start.

3.15.4. Creación de un nuevo bot mediante el comando /newbot

A continuación, se ejecutó el comando /newbot, el cual permite crear un nuevo bot dentro de la plataforma Telegram.

Durante este proceso, el sistema solicita la asignación de un nombre identificativo para el bot y un nombre de usuario único que finalizará en "_bot".

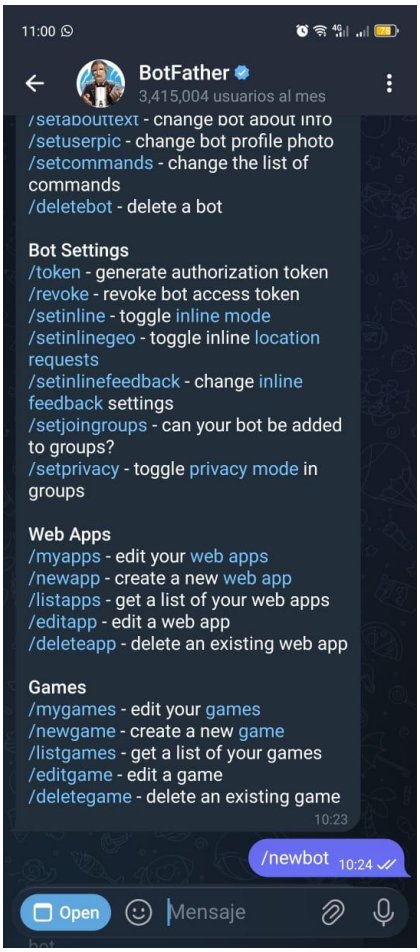


Figura 7: Comando newbot.

3.15.5. Asignación del nombre del bot y obtención del TOKEN

En esta etapa se asignó el nombre Enrique_bot y el nombre de usuario Enrique83_bot. Una vez completada esta configuración, Telegram generó automáticamente un TOKEN de acceso, el cual es indispensable para establecer la comunicación entre el bot y la tarjeta Raspberry Pi 4.

Este token fue posteriormente utilizado dentro del programa del sistema para permitir el envío de notificaciones automáticas hacia el usuario administrador.

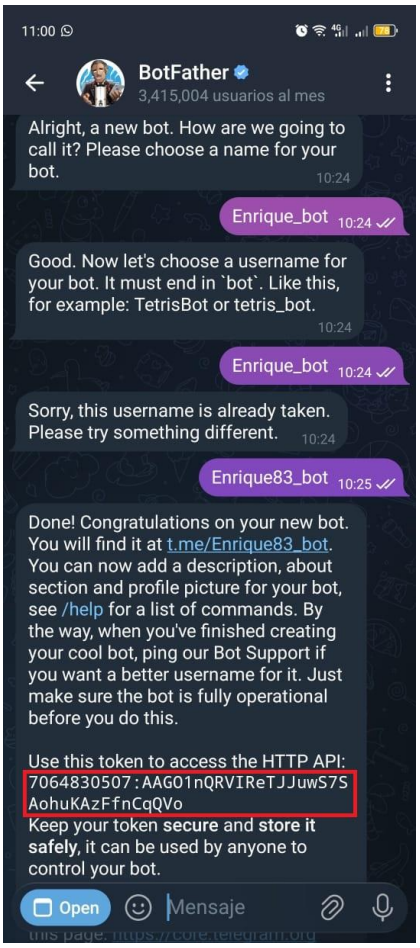


Figura 8: Asignación del bot.

3.15.6. Programación de la comunicación entre Telegram y la Raspberry Pi

Una vez obtenido el token del bot y el ID del usuario administrador, se procedió a la programación de la comunicación entre Telegram y la Raspberry Pi (ver **Anexo 4**). Para ello, se implementó un script en Python que utiliza la API de Telegram, permitiendo el envío automático de mensajes desde el sistema de control de acceso.

Esta programación establece la conexión permanente entre el dispositivo y Telegram, habilitando el envío de notificaciones cada vez que se produce un evento relevante dentro del sistema.

3.15.7. Funcionamiento del sistema de notificaciones en Telegram

El sistema de notificaciones opera de forma automática en función del resultado del proceso de validación de acceso, como se observa su programación en el **Anexo 5**. Cuando una persona intenta ingresar, el sistema evalúa el reconocimiento facial y la

validación mediante tarjeta RFID. Dependiendo del resultado, se envía uno de los siguientes mensajes al usuario administrador:

- “Ingreso autorizado”, cuando el rostro es reconocido y la tarjeta RFID es válida.
- “Intento de ingreso fallido”, cuando el rostro no se encuentra registrado o la tarjeta RFID no es válida.

Este mecanismo permite un monitoreo constante del sistema de seguridad, brindando información inmediata sobre cualquier intento de acceso.

3.16. Flujo general de funcionamiento del sistema

El funcionamiento completo del sistema se desarrolla de la siguiente manera:

1. Inicio del dispositivo.

Figura 9: Inicio del dispositivo.

2. Ejecución del programa desde la interfaz Thonny.

Figura 10: Ejecución del programa desde la interfaz Thonny.

3. Creación de la base de datos facial.

Figura 11: Creación de la base de datos facial.

4. Recopilación de 300 imágenes para reconocimiento facial.

Figura 12: Recopilación de 300 imágenes para reconocimiento facial.

5. Entrenamiento del modelo de imágenes.

Figura 13: Entrenamiento del modelo de imágenes.

6. Ejecución final del entrenamiento.

Figura 14: Ejecución final del entrenamiento.

7. Activación del sistema mediante pulsador encendido.

Figura 15: Activación del sistema mediante pulsador encendido.

8. Presión del pulsador para solicitud de acceso.

Figura 16: Presión del pulsador para solicitar acceso.

9. Información del pulsador en acción.

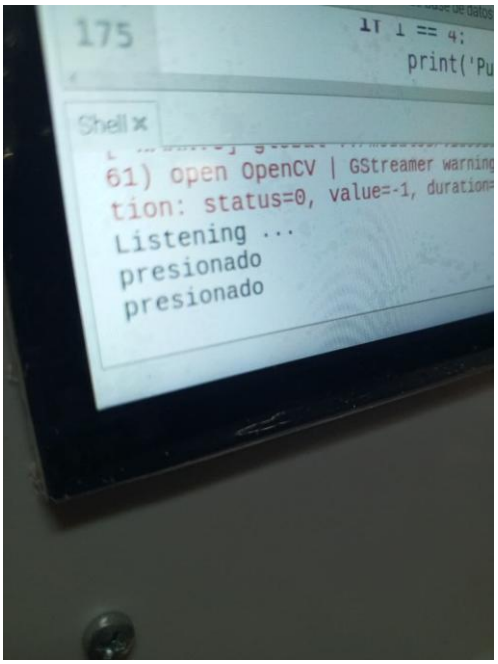


Figura 17: Información del pulsador de acción.

10. Detección de rostro para acceso.

Figura 18: Detección del rostro.

11. Verificación de acceso con tarjeta RFID

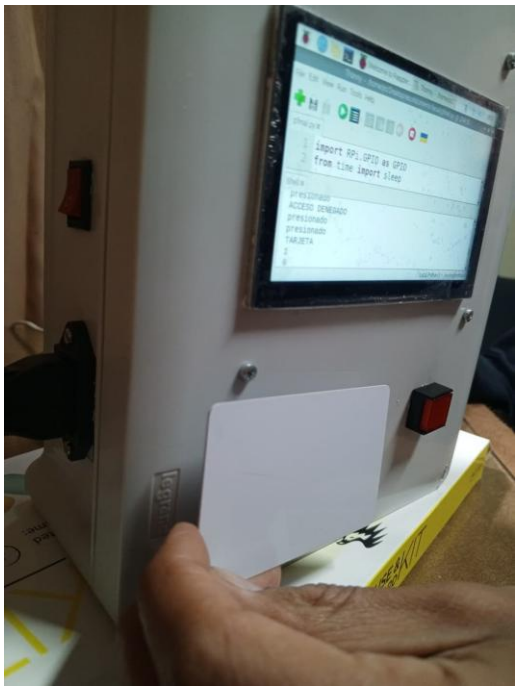


Figura 19: Verificación de acceso con tarjeta RFID

12. Activación del relé en caso de acceso válido.

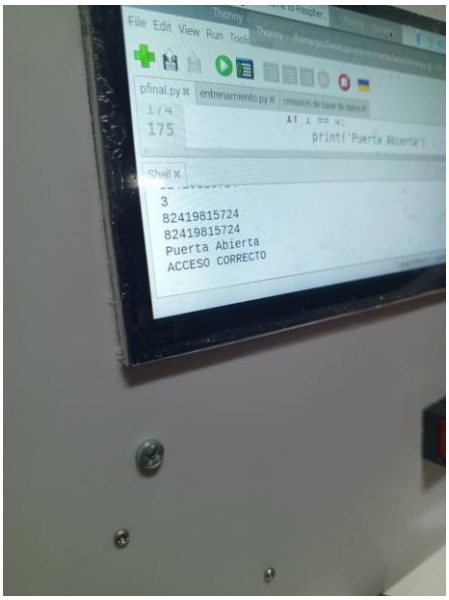


Figura 20: Activación del relé en caso de acceso válido.

13. Envío de notificación a Telegram indicando ingreso autorizado o fallido.

Figura 21: Envío de notificación a Telegram indicando ingreso autorizado o fallido.

3.17. Medidas de seguridad de la información

Para proteger la información confidencial que gestiona el sistema, se ha implementado el cifrado en una base de datos local alojada en la Raspberry Pi. Esta base de datos almacena el modelo facial creado durante el entrenamiento del modelo y el identificador único (UID) de la tarjeta RFID. La seguridad de la información se aborda desde dos perspectivas principales: la seguridad de los datos en reposo y la seguridad de los datos en transmisión. En el primer caso, el cifrado de la base de datos garantiza la confidencialidad de la información almacenada incluso en caso de robo, pérdida o manipulación del dispositivo.

Además, la comunicación entre los diferentes módulos del sistema se realiza mediante mecanismos seguros, lo que reduce el riesgo de interceptación, alteración o suplantación de identidad de los datos transmitidos. Esto es especialmente importante para el envío de notificaciones y alertas de seguridad a los administradores a través de plataformas de mensajería.

CAPÍTULO 4
RESULTADOS

4.1. Funcionamiento del sistema de control de acceso

Durante las pruebas realizadas al sistema de control de acceso implementado, se verificó el correcto funcionamiento de cada uno de los módulos que lo conforman. El sistema respondió de manera adecuada ante los intentos de ingreso, ejecutando de forma secuencial los procesos de reconocimiento facial, validación mediante tarjeta RFID y activación del relé para el control del acceso físico.

Se evidenció que el sistema permite restringir el ingreso únicamente a personas autorizadas, cumpliendo con el objetivo principal de mejorar la seguridad del área donde fue implementado. La integración entre el hardware y el software permitió una operación estable y continua durante las pruebas realizadas.

4.2. Resultados del proceso de reconocimiento facial

El módulo de reconocimiento facial demostró un desempeño adecuado al comparar el rostro capturado por la cámara con los registros almacenados en la base de datos. Cuando el rostro correspondía a un usuario registrado, el sistema continuaba con el siguiente nivel de validación; caso contrario, el acceso era denegado automáticamente.

4.3. Matriz de confusión del sistema de reconocimiento facial

Para evaluar el rendimiento del sistema de reconocimiento facial, se creó una matriz de confusión basada en los resultados obtenidos durante las pruebas experimentales. Esta herramienta permite analizar el comportamiento del sistema al identificar correcta o incorrectamente los intentos de acceso de usuarios autorizados y no autorizados.

Durante las pruebas, se registraron diferentes intentos de acceso, cuyos resultados se clasificaron como verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, como se muestra en la Tabla 3.

Tabla 3. Matriz de confusión.

Clasificación del resultado	Cantidad
Verdaderos positivos (VP)	92
Verdaderos negativos (VN)	95
Falsos positivos (FP)	3
Falsos negativos (FN)	5

Elaborado por: Rea Napoleón

4.4. Tasas de error FAR y FRR

La tasa FAR (Tasa de aceptación falsa) es la posibilidad de que un usuario sea reconocido erróneamente y FRR (Tasa de falso rechazo) cuando el sistema no reconoce a un usuario por error. Para esta parametrización se efectuaron las siguientes ecuaciones para calcular los siguientes indicadores:

FAR = FP / (FP + VN)

FRR = FN / (FN + VP)

Remplazando los datos que se obtuvieron durante las pruebas experimentales:

FAR = 3 / (3 + 95) = 0,0306 (3,06%) FRR = 5 / (5 + 92) = 0,0515 (5,15%)

Estos resultados indican que el sistema presenta una baja tasa de falsas aceptaciones y una tasa de falsos rechazos razonable, lo que lo hace adecuado para aplicaciones de control de acceso en entornos de pequeña escala donde la seguridad del sistema es importante.

4.5. Análisis reconocimiento facial bajo distintas condiciones de iluminación

Para analizar el rendimiento del algoritmo de reconocimiento facial LBPH en diferentes condiciones de iluminación, se realizaron pruebas controladas variando el nivel de luz ambiental, medido en lux. Se evaluó el porcentaje de reconocimientos correctos logrados por el sistema en cada una de estas condiciones.

Tabla 4. Resultados del reconocimiento facial según el nivel de iluminación.

Nivel de iluminación (Lux)	Porcentaje de reconocimiento
500	97 %
300	94 %
150	88 %
80	72 %
40	55 %

Elaborado por: Rea Napoleón

El análisis de los resultados muestra que el algoritmo LBPH funciona bien en condiciones de iluminación media y alta, su precisión disminuye significativamente cuando la iluminación es inferior a 100 lux, para lo que se hace indispensable la validación complementaria de una tarjeta RFID.

4.6. Validación mediante tarjeta RFID

La validación por tarjeta RFID entra en funcionamiento una vez el usuario pase el primer filtro que es el reconocimiento facial, una vez validado este, el sistema requiere de la identificación con la ayuda de la tarjeta RFID, una vez generadas las pruebas, se identifica que las tarjetas registradas podrán activar el relé que abre la cerradura de la puerta, certifica y cuando se utilizó una tarjeta RFID no válida, el sistema negó el acceso de forma inmediata, aun cuando el rostro había sido reconocido previamente.

4.7. Análisis del sistema de notificaciones mediante Telegram

El sistema de notificaciones a través de Telegram funcionó de manera automática y en tiempo real durante todos los eventos de acceso. Cada intento de ingreso generó una notificación dirigida al usuario administrador, utilizando el token del bot y el ID del usuario configurados previamente.

Los mensajes enviados permitieron identificar claramente el estado del acceso, notificando “Ingreso autorizado” cuando se cumplían todas las validaciones y “Intento de

ingreso fallido” cuando alguna de ellas no era satisfactoria. Este mecanismo facilitó el monitoreo remoto del sistema, sin necesidad de presencia física en el lugar.

4.8. Eficiencia y confiabilidad del sistema implementado

En las pruebas de funcionamiento continuo, el sistema mostró una respuesta rápida y estable, sin presentar fallos en la comunicación entre la Raspberry Pi, los dispositivos de entrada y la plataforma Telegram. El tiempo de respuesta entre la detección del evento y el envío de la notificación fue prácticamente inmediato.

4.9. Viabilidad técnica y económica del sistema

Al ser Raspberry Pi OS y Python plataformas de software de código abierto, así como la aplicación de Telegram, su implementación no es compleja. Los componentes utilizados en el dispositivo son de calidad y bajo costo, lo que le permite al proyecto ser viable técnica y económicamente.

CRONOGRAMA

Tabla 5: Cronograma

CRONOGRAMA 2025 - 2026												
Inicio: 1/4/2025 - Finalización: 31/1/202		Abril 1/4/2025 a 30/4/2025	Mayo 1/5/2025 a 31/5/2025	Junio 1/6/2025 a 30/6/2025	Julio 1/7/2025 a 31/7/202	Agosto 1/8/2025 a 31/8/2025	Septiembre 1/9/2025 a 30/9/2025	Octubre 1/10/2025 a 31/10/2025	Noviembre 1/11/2025 a 30/11/2025	Diciembre 1/12/2025 a 31/12/202	Enero 1/1/2026 a 31/1/202
Actividades	Duración										
Estudio de las necesidades de los usuarios para un acceso domestico inteligente.	30 Dias										
Estudio de los tipos y caracteristicas de las tarjetas Raspberry.	30 Dias										
Ejecutar un estudio de construcción para la estructura del software (Phyton, ML)	30 Dias										
Revisión y selección de cámara, lector rfid y actuadores.	30 Dias										
Creación del prototipo	30 Dias										
Implementación y programación de la cámara, lector rfid y actuadores.	45Dias										
Programación de servidores y plataforma de mensajes.	45 Dias										
Documentación del Proyecto.	60 Dias										
Tutorias	180 Dias										

Elaborado por: Rea Napoleón

PRESUPUESTO

En la parte económica la cual permitió la viabilidad del proyecto se consideró el costo del hardware, la tarjeta de almacenamiento y los materiales necesarios para la implementación del prototipo de control y acceso inteligente.

Tabla 6. Presupuesto.

Descripción	Costos
Raspberry Pi 4 Modelo B - 4GB	$80
Módulo de cámara Raspberry Pi 2	$50
MicroSD 16GB	$10
Fuente de alimentación micro USB Raspberry Pi de12,5W	$10
Servicio de almacenamiento Render	$20
Modulo RFID	$5
Tarjetas Magnéticas	$2
Mano de Obra	$40
Diseño de Ingeniería	$80
Total	**$297**

Elaborado por: Rea Napoleón

CONCLUSIONES

- El dispositivo para el control de acceso implementado demuestra ser una solución fiable al problema de la seguridad, utilizando reconocimiento facial y confirmando la identidad con tarjetas RFID e incorpora una notificación por medio de Telegram. Todo este sistema ejecutado por una tarjeta Raspberry Pi, permite garantizar que solo las personas registradas puedan ingresar a las áreas protegidas.
- Para identificar correctamente los rasgos faciales de un rostro que desea acceder al sistema, se programó la creación de una base de datos con 300 imágenes de los usuarios lo que permite garantizar minimizar errores y mejorar el reconocimiento facial.
- Al incorporar en la arquitectura del sistema una segunda validación mediante tarjetas RFID el sistema de seguridad se fortalece ya que solo después de pasar los dos filtros el mecanismo permitirá el acceso reduciendo los posibles ingresos a personas no registradas.
- Al utilizar una tarjeta Raspberry Pi y la aplicación Telegram permite al usuario administrador recibir notificaciones en tiempo real, lo que garantiza mantenerlo informado y monitorizando los intentos de ingreso exitosos como los fallidos.
- El dispositivo durante el muestreo de funcionamiento se mostró consistente y estable, con respuestas rápidas y sin errores, lo que evidencia el correcto funcionamiento y capacidad para ser la respuesta al problema de seguridad en ambientes que necesitan monitoreo en el acceso de personas.

RECOMENDACIONES

- Instalar un sistema de iluminación para que la cámara tome las imágenes más nítidas, necesario para aumentar la precisión del reconocimiento facial. Es recomendable la colocación de un sistema externo que se accione con el pulsador, mejorar la cámara con un sistema de visión nocturna (infrarrojo) e implementar una programación más robusta que permita detectar las características de la imagen en estas condiciones.
- Las actualizaciones son uno de los puntos más importantes del sistema ya que garantiza el correcto funcionamiento de la tarjeta Raspberry Pi con sus módulos. Por esta razón se debe dar un constante mantenimiento para garantizar la compatibilidad con nuevas versiones como OpenCV y controladores de los elementos como la cámara que conforman el dispositivo.
- En caso de una falla total del sistema se debe integran otra forma de ingreso como una llave oculta o un dispositivo de emergencia que solo pueda ser activado por el administrador y se pueda garantizar su acceso.
- Para cuando exista un corte de energía eléctrica debe implementarse un sistema de respaldo de baterías para largos periodos ya que es un país propenso a sufrir de este tipo de cortes. Lo que garantizaría el correcto funcionamiento del sistema, así como la conexión a una red de internet.
- Es necesario utilizar el cifrado o técnicas de incognito para tener a salvo la ID y token del bot de la aplicación de Telegram. Así como implementar un sistema de confirmación de doble vía con un mensaje de aceptación del ingreso de una persona,

BIBLIOGRAFÍA

Areitio, B. (2008). *Seguridad de la información. Redes, informática y sistemas de información.* Caprigh.

Bravo, P., & Noroño , F. (2025). *Impacto del Crimen Organizado y la Inseguridad en el Desempeño Económico del Ecuador: Una Perspectiva Costo-Beneficio.* Ciencias Latina.

Carpio, G., & Tobar, A. (2025). *Inseguridad ciudadana en el desarrollo socioecónomico en las familias del cantón Machala Ecuador.* Sapientaie.

Coronel, E. (2021). Sistema Inteligente de identificación facial para registro de asistencia estudiantil en la Universidad Ecotec. *Ecotec, II*(35), 120-136.

Correa, M. (2016). *Impacto de las políticas de seguridad estatal en el Distrito Metropolitano de Quito.* UASB.

DMQ. (2023). *Diagnóstico aproximación a la seguridad y convivencia ciudadana en el DMQ.* DMQ.

Fúnez, E. (8 de Marzo de 2022). Diseño de un sistema de seguridad en el hogar basado en IoT y creación de prototipo. *Colección de Recursos Educativos Abiertos*, pág. 93. Obtenido de www.raspberrypi.org

Historia.ec, L. (22 de Agosto de 2023). Robos a casas: ¿Se puede dormir tranquilos?

Law, T. (2017). Sistema de control de acceso. *Ciencias de la Computación, IV*(2), 26-35.

Niola, C., & Sanango, W. (2019). Desarrollo de un software de seguridad para detección y reeconocimiento facial basados en los algoritmos de Viola - Jones y PCA EIGENFACE. *UPS, I*(2), 93.

Primicias.ec. (12 de Marzo de 2024). ecuador-paises-mayor-criminalidad-mundo. *ecuador-paises-mayor-criminalidad-mundo*, págs. https://www.primicias.ec/noticias/seguridad/ecuador-paises-mayor-criminalidad-mundo/).

Sanchez, J. (2024). *Arquitecturas escalables y distribuidas en la nube para una gestión eficiente de los recursos.* Obtenido de https://www.age2.es/noticias/arquitecturas-escalables-y-distribuidas-en-la-nube-para-una-gestion-eficiente-de-los-

recursos/#:~:text=Una%20arquitectura%20escalable%20nos%20permite,hora%20de%20acceder%20al%20servicio.

Sánchez, L. (2024). *Análisis de ciberseguridad a la infraestructura tecnologica de la empresa Alianza S.A.* UPS.

StackExchange. (20 de Abril de 2016). Raspberry Pi Camera, what's the big deal. págs. https://raspberrypi.stackexchange.com/questions/19379/raspberry-pi-camerawhats-the-big-deal .

Terrazas, E. (2024). Sistema Biométrico con reconocimiento facial. Un procedimiento de registro de casos de Transporte y Seguridad Vial. *Ingeniería y sus Alcances*, *IV*(1), 12-36.

Venturini, J., & Garay, V. (2024). Reconocimiento facial en América Latina: tendencias en la implementación de una tecnología perversa. *A1Sur, I*(12), 1-25.

W. Zhao, R. C. (2003). Face recognition: A literature survey. *ACM computing surveys (CSUR), 35*(4), págs. 399–458.

Washington, M., & Dilan , S. (2023). *Implementación de un sistema domótico de seguridad y control mediante IOT aplicado a un laboratorio.* UTC.

ANEXOS

Anexo 1. Creación Base de Datos

```
import cv2
import imutils
import os
#crear caprpeta
#personName = 'HECTOR'
personName = 'PERSONA'
dataPath= '/home/pc/Desktop/reconocimiento facial'
personPath= dataPath + '/' + personName
############################
if not os.path.exists(personPath):
   print('CARPETA CREADA: ', personPath)
   os.makedirs(personPath)
############################
cap= cv2.VideoCapture(0)
faceClassif = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
count = 0
while True:
   ret,frame=cap.read()
   if ret == False:
      break
   frame = imutils.resize(frame, width=700)
   gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   auxFrame = frame.copy()
   faces = faceClassif.detectMultiScale(gray, 1.3 , 5)
   for(x, y, w, h) in faces:
      #crerar el rectangulo de la persona
      cv2.rectangle(frame,(x, y),(x+w, y+h), (0, 255, 0), 2)
      rostro = auxFrame[y:y + h, x:x + w]
      rostro = cv2.resize(rostro, (720, 720), interpolation=cv2.INTER_CUBIC)
      cv2.imwrite(personPath + '/rostro_{}.jpg'.format(count),rostro)
      count = count +1
   cv2.imshow('frame',frame)
   k = cv2.waitKey(1)
   if k == 27 or count >= 300:
    break
cap.release()
cv2.destroyAllWindows()
```

Anexo 2. Entrenamiento del Programa

```
import cv2
import os
import numpy as np
from os import listdir
dataPath = '/home/pc/Desktop/reconocimiento facial'
peopleList = os.listdir(dataPath)
print('LISTA DE PERSONAS: ', peopleList)

lables = []
facesData = []
label = 0
aux = 0
for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('LEYENDO IMAGENES')
    if aux ==1:
        break
    aux = aux +1
    for fileName in os.listdir(personPath):

        print('ROSTRO: ', nameDir + '/' + fileName)
        lables.append(label)

        facesData.append(cv2.imread(personPath + '/' + fileName, 0))

        image = cv2.imread(personPath + '/' + fileName, 0)

    label = label +1


face_recognizer = cv2.face.LBPHFaceRecognizer_create()
print('entrenado....')
face_recognizer.train(facesData, np.array(lables))
face_recognizer.write('Modelofrotntal.xml')
print('modelo guardado')
```

Anexo 3. Programación Haar Cascade

```
import RPi.GPIO as GPIO
from time import sleep
from mfrc522 import SimpleMFRC522
import sys
import time
import telepot
import cv2
import os
import imutils
from telepot.loop import MessageLoop
from telepot.delegate import per_chat_id, create_open, pave_event_space
import threading

aux = 0
aux1=0
auxp=0
auxo=0
led = 36
i=0
id=0
rele = 37
tarjeta = 82419815724

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
reader = SimpleMFRC522()

dataPath = '/home/pc/Desktop/reconocimiento facial'
imagePaths = os.listdir(dataPath)
print('imagePath= ', imagePaths)

face_recognizer = cv2.face.LBPHFaceRecognizer_create()

face_recognizer.read('Modelofrotntal.xml')
cap = cv2.VideoCapture(0)
auu = 0
faceClassif = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
def Timer_Interrupt():
    global aux
    global auxo
    global i
    global id

    #if aux == 1 :
        #auxo = auxo + 1

    if aux==7:

      i= i +1
      print(i)
      id, text = reader.read()

    threading.Timer(1,Timer_Interrupt).start()


threading.Timer(1,Timer_Interrupt).start()
```

Anexo 4. Código de conexión dispositivo con Telegram (Notificaciones)

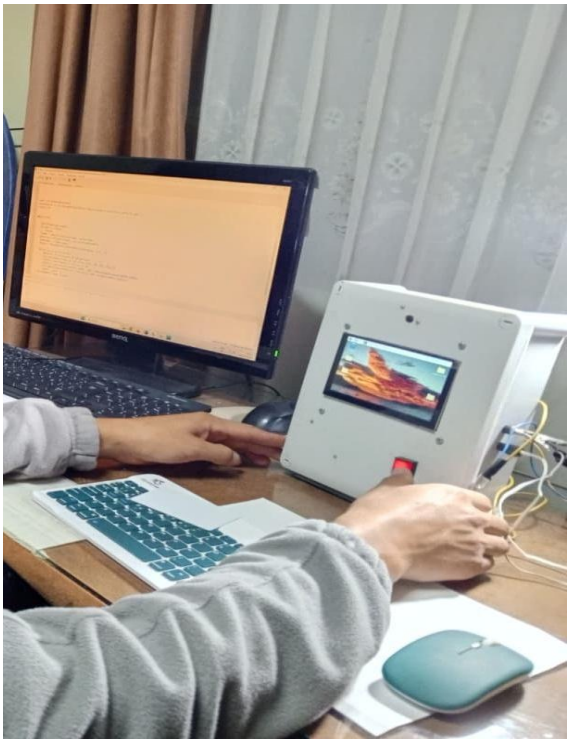
```
71
72  class MessageCounter(telepot.helper.ChatHandler):
73      def __init__(self, *args, **kwargs):
74          super(MessageCounter, self).__init__(*args, **kwargs)
75          self._count = 0
76
77      def on_chat_message(self, msg):
78          self._count += 1
79          self.sender.sendMessage(self._count)
80          self.sender.sendMessage('hola')
81
82
83  #TOKEN = '7349661707:AAEW547K_6RKkaigVoaeSzXCnMpyqJm0NEE'
84  TOKEN = '7064830507:AAGO1nQRVIReTJJuwS7SAohuKAzFfnCqQVo'
85
86  bot = telepot.DelegatorBot(TOKEN, [
87      pave_event_space()(
88          per_chat_id(), create_open, MessageCounter, timeout=10
89      ),
90  ])
91  MessageLoop(bot).run_as_thread()
92  print('Listening ...')
93
94  pulsador = 32
95  GPIO.setup(led,GPIO.OUT, initial=GPIO.LOW)
96  GPIO.setup(rele,GPIO.OUT, initial=GPIO.LOW)
97  GPIO.setup(pulsador,GPIO.IN, pull_up_down=GPIO.PUD_UP)
98  GPIO.add_event_detect(pulsador,GPIO.RISING, callback=apagar, bouncetime=200)
99
100
```

Anexo 5. Programación de funcionamiento

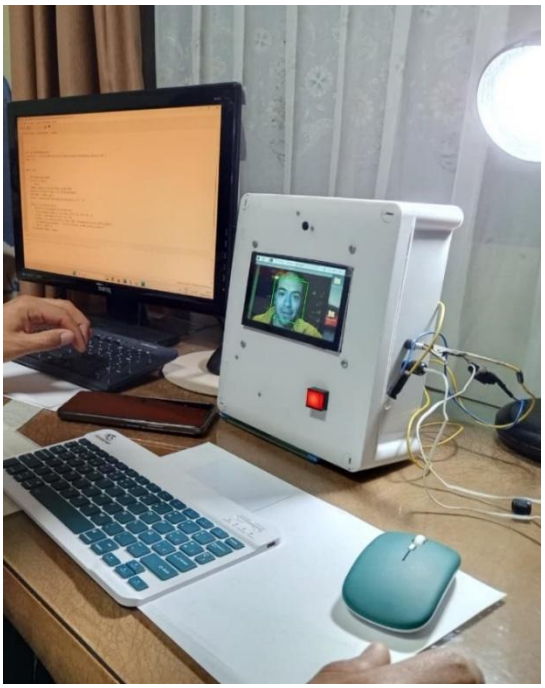
```
while True:

    if aux == 1:
        GPIO.output(led,GPIO.LOW)
        GPIO.output(rele,GPIO.LOW)
        ret,frame = cap.read()
        if ret == False:
            break
        frame = imutils.resize(frame, width=700)
        gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        auxFrame = gray.copy()

        faces = faceClassif.detectMultiScale(gray, 1.3 , 5)

        for(x, y, w, h) in faces:
            rostro = auxFrame[y:y + h, x:x + w]
            rostro = cv2.resize(rostro, (150, 150), interpolation=cv2.INTER_CUBIC)
            result = face_recognizer.predict(rostro)
            cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(25,225,0),1,cv2.LINE_AA)

            if  result[1] > 67 and result[1] < 80 :

               cv2.putText(frame,'CONOCIDO',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
               cv2.rectangle(frame, (x,y), (x+w,y+h),(0,255,0),2)
               au= + 1

               auxp = auxp + 1
            else:
                cv2.putText(frame,'DESCONOCIDO',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
                cv2.rectangle(frame, (x,y), (x+w,y+h),(0,0,255),2)
                auxo = auxo + 1

        cv2.imshow('imagen',frame)


    if auxp == 10:
       cv2.destroyAllWindows()
       aux =7
       auxp = 0
       auxo = 0
       print('primer paso')


    if auxo == 5:
       cv2.destroyAllWindows()
       aux =0
       auxo = 0
       auxp = 0
       GPIO.output(led, GPIO.LOW)
       print('ACCESO DENEGADO')

       bot.sendMessage('7711752496', 'INTENTO DE INGRESO FALLIDO') #coamndo de envio de mensaje
    GPIO.output(led,GPIO.HIGH)
    sleep(0.5)
    GPIO.output(led,GPIO.LOW)
    sleep(0.5)

   #######
    if aux == 7:
        for i in range(0,5):

             print(id)
             if i == 4:
                 print('lazo roto')
                 i=0
                 break
             sleep(1)
        aux = 0
        sleep(1)
        if id==tarjeta:

         GPIO.output(led, GPIO.HIGH)
         print('ACCESO CORRECTO')

         bot.sendMessage('7711752496', 'INGRESO AUTORIZADO') #coamndo de envio de mensakie
         GPIO.output(led,GPIO.HIGH)
         GPIO.output(rele,GPIO.HIGH)
         sleep(2)
         GPIO.output(led,GPIO.LOW)
         GPIO.output(rele,GPIO.LOW)

         id= 0


        else:
         GPIO.output(led, GPIO.LOW)
         print('ACCESO DENEGADO')
         id= 0
         bot.sendMessage('7711752496', 'INTENTO DE INGRESO FALLIDO') #coamndo de envio de mensakie

    k = cv2.waitKey(1)
    if k == 27:
     break
GPIO.output(led, GPIO.LOW)
GPIO.output(rele,GPIO.LOW)
cap.release()
cv2.destroyAllWindows()
```

Anexo 6. Evidencia Fotográfica

Timbre para acceder (Inicio del Sistema)

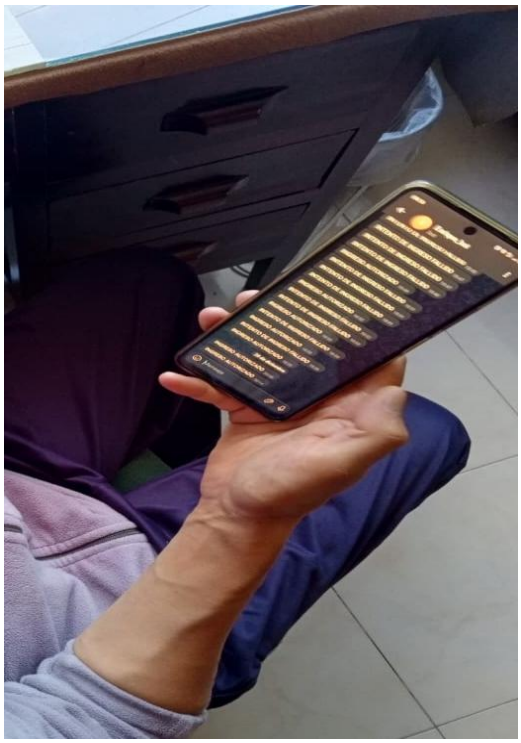


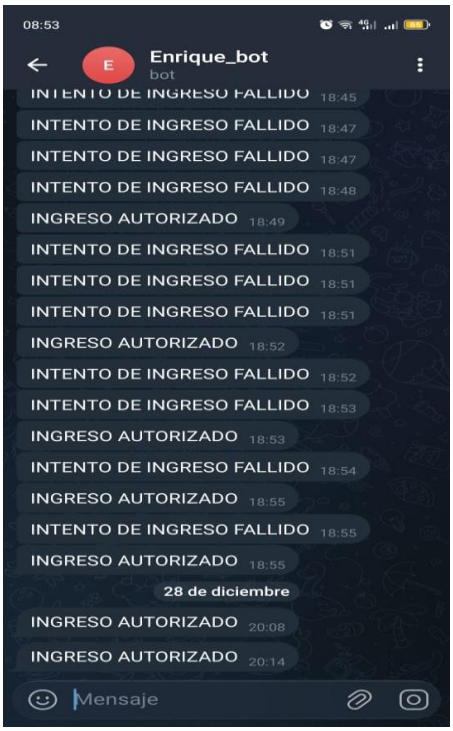
Reconocimiento facial para el acceso



Acceso facial exitoso y verificación con tarjeta RIF

Recepción de notificación en Telegram





Anexo 7. Programación Final

```
import RPi.GPIO as GPIO
from time import sleep
from mfrc522 import SimpleMFRC522
import sys
import time
import telepot
import cv2
import os
import imutils
from telepot.loop import MessageLoop
from telepot.delegate import per_chat_id, create_open, pave_event_space
import threading

aux = 0
aux1=0
auxp=0
auxo=0
led = 36
i=0
id=0
rele = 37
tarjeta = 82419815724

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
reader = SimpleMFRC522()

dataPath = '/home/pc/Desktop/reconocimiento facial'
imagePaths = os.listdir(dataPath)
print('imagePath= ', imagePaths)

face_recognizer = cv2.face.LBPHFaceRecognizer_create()

face_recognizer.read('Modelofrotntal.xml')
cap = cv2.VideoCapture(0)
auu = 0
faceClassif = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
def Timer_Interrupt():
    global aux
    global auxo
    global i
    global id

    #if aux == 1 :
        #auxo = auxo + 1

    if aux==7:
```

```
    id, text = reader.read()

    threading.Timer(1,Timer_Interrupt).start()

threading.Timer(1,Timer_Interrupt).start()

def apagar(pin):
    global aux
    global aux1
    aux1=1
    if aux==0:
        aux=1
    if aux==1:
        print('presionado')

    class MessageCounter(telepot.helper.ChatHandler):
    def __init__(self, *args, **kwargs):
        super(MessageCounter, self).__init__(*args, **kwargs)
        self._count = 0

    def on_chat_message(self, msg):
        self._count += 1
        self.sender.sendMessage(self._count)
        self.sender.sendMessage('hola')

#TOKEN = '7349661707:AAEW547K_6RKkaigVoaeSzXCnMpyqJm0NEE'
TOKEN = '7064830507:AAGO1nQRVIReTJJuwS7SAohuKAzFfnCqQVo'

bot = telepot.DelegatorBot(TOKEN, [
    pave_event_space()(
        per_chat_id(), create_open, MessageCounter, timeout=10
    ),
])
MessageLoop(bot).run_as_thread()
print('Listening ...')

pulsador = 32
GPIO.setup(led,GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(rele,GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(pulsador,GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.add_event_detect(pulsador,GPIO.RISING, callback=apagar, bouncetime=200)

while True:

    if aux == 1:
        GPIO.output(led,GPIO.LOW)
        GPIO.output(rele,GPIO.LOW)
        ret,frame = cap.read()
        if ret == False:
            break
```

```
        frame = imutils.resize(frame, width=700)
        gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        auxFrame = gray.copy()

        faces = faceClassif.detectMultiScale(gray, 1.3 , 5)

        for(x, y, w, h) in faces:
            rostro = auxFrame[y:y + h, x:x + w]
            rostro = cv2.resize(rostro, (150, 150), interpolation=cv2.INTER_CUBIC)
            result = face_recognizer.predict(rostro)
            cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(25,225,0),1,cv2.LINE_AA)

            if result[1] > 67 and result[1] < 80 :

              cv2.putText(frame,'CONOCIDO',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
              cv2.rectangle(frame, (x,y), (x+w,y+h),(0,255,0),2)
              au= + 1

              auxp = auxp + 1
            else:
               cv2.putText(frame,'DESCONOCIDO',(x,y-
20),2,0.8,(0,0,255),1,cv2.LINE_AA)
               cv2.rectangle(frame, (x,y), (x+w,y+h),(0,0,255),2)
               auxo = auxo + 1

        cv2.imshow('imagen',frame)


    if auxp == 10:
      cv2.destroyAllWindows()
      aux =7
      auxp = 0
      auxo = 0
      print('primer paso')


    if auxo == 5:
      cv2.destroyAllWindows()
      aux =0
      auxo = 0
      auxp = 0
      GPIO.output(led, GPIO.LOW)
      print('ACCESO DENEGADO')

      bot.sendMessage('7711752496', 'INTENTO DE INGRESO FALLIDO') #coamndo
de envio de mensakie

    GPIO.output(led,GPIO.HIGH)
    sleep(0.5)
    GPIO.output(led,GPIO.LOW)
```

```
    sleep(0.5)

   #######
    if aux == 7:
       for i in range(0,5):

           print(id)
           if i == 4:
              print('lazo roto')
              i=0
              break
           sleep(1)
       aux = 0
       sleep(1)
       if id==tarjeta:

        GPIO.output(led, GPIO.HIGH)
        print('ACCESO CORRECTO')

        bot.sendMessage('7711752496', 'INGRESO AUTORIZADO') #coamndo de envio
de mensakie
        GPIO.output(led,GPIO.HIGH)
        GPIO.output(rele,GPIO.HIGH)
        sleep(2)
        GPIO.output(led,GPIO.LOW)
        GPIO.output(rele,GPIO.LOW)

        id= 0

       else:
        GPIO.output(led, GPIO.LOW)
        print('ACCESO DENEGADO')
        id= 0
        bot.sendMessage('7711752496', 'INTENTO DE INGRESO FALLIDO') #coamndo
de envio de mensakie

    k = cv2.waitKey(1)
    if k == 27:
     break
GPIO.output(led, GPIO.LOW)
GPIO.output(rele,GPIO.LOW)
cap.release()
cv2.destroyAllWindows()


#GPIO.cleanup()
```