



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO  
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

**DESARROLLO DE UN SISTEMA DOMÓTICO PARA EL LABORATORIO DE  
ROBÓTICA DE LA UPS BASADO EN RECONOCIMIENTO DE AUDIO Y VISIÓN  
ARTIFICIAL**

Trabajo de titulación previo a la obtención del  
Título de Ingeniero en Electrónica y Automatización

AUTORES: Luis Eduardo Alvarado Cruz

Joseph Daniel Novoa Ruiz

TUTOR: Gustavo Javier Caiza Guanochanga

Quito-Ecuador

2023

## **CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Nosotros, Luis Eduardo Alvarado Cruz con documento de identificación N° 1718265539 y Joseph Daniel Novoa Ruiz con documento de identificación N° 1751260587; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos 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, 09 de Agosto del año 2023

Atentamente,



---

Luis Eduardo Alvarado Cruz

1718265539



---

Joseph Daniel Novoa Ruiz

1751260587

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

Nosotros, Luis Eduardo Alvarado Cruz con documento de identificación No. 1718265539 y Joseph Daniel Novoa Ruiz con documento de identificación No. 1751260587, expresamos nuestra voluntad y por medio del presente documentocedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Desarrollo de un sistema domótico para el laboratorio de robótica de la UPS basado en reconocimiento de audio y visión artificial”. El cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica y Automatización, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Quito, 09 de Agosto del año 2023

Atentamente,

---

Luis Eduardo Alvarado Cruz  
1718265539

---

Joseph Daniel Novoa Ruiz  
1751260587

## **CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN**

Yo, Gustavo Javier Caiza Guanochanga con documento de identificación N° 1721192191, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación **DESARROLLO DE UN SISTEMA DOMÓTICO PARA EL LABORATORIO DE ROBÓTICA DE LA UPS BASADO EN RECONOCIMIENTO DE AUDIO Y VISIÓN ARTIFICIAL**, realizado por Luis Eduardo Alvarado Cruz con documento de identificación N° 1718265539 y por Joseph Daniel Novoa Ruiz con documento de identificación N° 1751260587, 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, 09 de Agosto del año 2021

Atentamente,



---

Ing. Gustavo Javier Caiza Guanochanga, Mgst.  
1721192191

## RESUMEN

El término "domótica" se refiere a la implementación de tecnología en los hogares para automatizar y controlar una variedad de tareas, tales como seguridad, iluminación, climatización, dispositivos electrónicos, entre otros. Gracias a avances como el Internet de las cosas y tecnologías como la Inteligencia Artificial ha habido grandes cambios en la rama de la domótica, mismos que han facilitado la interconexión de dispositivos y sistemas, proporcionando comodidad, eficiencia energética, seguridad y un mayor control sobre los dispositivos y sistemas que se encuentran en el hogar. En este proyecto se realizará la implementación de dichos sistemas domóticos en el laboratorio de robótica de la Universidad Politécnica Salesiana. Se utilizarán sistemas de reconocimiento facial y reconocimiento de voz para lograr la automatización y el control de diferentes aspectos del entorno.

Se desarrollará un sistema de reconocimiento facial que utilizará la visión artificial para permitir un registro seguro y automático de estudiantes y personal que ingresen al laboratorio. Esto facilitará el proceso de control de acceso y aumentará la seguridad al identificar de manera precisa a cada persona. Sin embargo, el reconocimiento de voz se implementará para el control automatizado de varios dispositivos y parámetros en el laboratorio. Los usuarios podrán controlar la iluminación, los motores y las persianas, optimizar la eficiencia y facilitar la interacción con el entorno mediante comandos de voz.

La combinación de estas tecnologías de visión artificial y reconocimiento de voz hará que el laboratorio de robótica sea un lugar de trabajo más seguro e inteligente. Con la interconexión de los sistemas y dispositivos, se optimizará el uso de recursos y se promoverá un entorno más eficiente en términos de energía.

Palabras Clave: Reconocimiento facial, reconocimiento de voz, ESP32, ESP32-CAM, Raspberry pi

## **ABSTRACT**

The term "home automation" refers to the implementation of technology in homes to automate and control a variety of tasks, such as security, lighting, air conditioning, electronic devices, among others. Thanks to advances such as the Internet of Things and technologies such as Artificial Intelligence, there have been major changes in the field of home automation, which have facilitated the interconnection of devices and systems, providing convenience, energy efficiency, security and greater control over the devices and systems that lie at home. In this project, the implementation of these domotic systems will be carried out in the robotics laboratory of the Universidad Politécnica Salesiana. Facial recognition and voice recognition systems will be used to achieve automation and control of different aspects of the environment.

A facial recognition system using artificial vision will be developed to enable secure and automatic registration of students and staff entering the laboratory. This will facilitate the access control process and increase security by accurately identifying each person. However, voice recognition will be implemented for automated control of various devices and parameters in the lab. Users will be able to control lighting, engines and shutters, optimize efficiency and facilitate interaction with the environment via voice commands.

The combination of these artificial vision and voice recognition technologies will make the robotics lab a safer and smarter workplace. The interconnection of systems and devices will optimize the use of resources and promote a more energy-efficient environment.

**Keywords:** Facial recognition, voice recognition, ESP32, ESP32-CAM, Raspberry pi

## ÍNDICE GENERAL

CAPITULO 1. Introducción .....	11
CAPITULO 2. Problema: .....	12
2.1 Antecedentes: .....	12
2.2 Importancia y alcances .....	12
2.3 Delimitación: .....	13
CAPITULO 3. Objetivos Generales y Específicos .....	14
3.1 Objetivo General: .....	14
3.2 Objetivos específicos: .....	14
CAPITULO 4. Fundamentos teóricos .....	15
4.1 Sistemas domóticos .....	15
4.2 Seguridad Biométrica .....	15
4.2.1 Reconocimiento facial .....	15
4.2.2 Reconocimiento de Voz .....	17
4.3 ESP32 .....	17
4.4 ESP32-CAM .....	18
4.5 Arduino .....	19
4.6 ArduinoWebsocket .....	19
4.7 Python .....	19
4.8 TensorFlow .....	20
4.9 OpenCv .....	20
4.10 Tkinter .....	20
4.11 Matplotlib .....	20
4.12 Alexa .....	21
4.13 Dimmer .....	21
4.14 Relé .....	22
4.15 CONVERTIDORES DC/DC – BUCK .....	22
CAPITULO 5. Implementación .....	23
5.1 Dispositivo Echo Dot (ALEXA) .....	23
5.2 Dimmer NEXT HOME/NHE-D100 .....	24
5.3 Allone Pro – Persianas - Proyector .....	26
5.4 Sistema reconocimiento facial para abrir cerradura .....	28
5.5 Motores para plataforma .....	33
5.6 Visión Artificial para registro de estudiantes .....	36
CAPITULO 6. Resultados .....	38
6.7 Alexa (Luces/Proyector/Persianas/Motores) .....	38
6.8 Reconocimiento Facial para abrir cerradura .....	42

6.9 Visión artificial (Registro Estudiantes) .....	44
CAPITULO 7. Presupuesto .....	48
CAPITULO 8. Conclusiones .....	49
CAPITULO 9. Referencias bibliográficas .....	50
CAPITULO 10. ANEXOS .....	52



## ÍNDICE DE FIGURAS

Figura 4.1: Tarjeta ESP32.....	18
Figura 4.2: Tarjeta ESP32-CAM.....	18
Figura 4.3: Echo DOT de Alexa.....	21
Figura 4.4: Atenuador de luz inteligente.....	21
Figura 4.5: Relé.....	22
Figura 4.6: Conversor Buck.....	22
Figura 5.1: Echo Dot.....	20
Figura 5.2: Funciones de activación de Alexa.....	24
Figura 5.3: Dimmer Next HOME/NHE-D100.....	24
Figura 5.4: Conexión del atenuador inteligente.....	25
Figura 5.5: Ventana Dispositivo Next.....	26
Figura 5.6: Orbivo Allone Pro.....	27
Figura 5.7: Interfaz Orbivo Allone Pro.....	27
Figura 5.8: Instalación de particiones.....	28
Figura 5.9: Líneas para particiones Boards.....	28
Figura 5.10: Configuraciones para ESP32-CAM.....	29
Figura 5.11: Circuito ESP32-CAM.....	29
Figura 5.12: Diagrama de flujo del Código de la ESP32-CAM.....	30
Figura 5.13: Diagrama de flujo del Código de la ESP32-CAM.....	31
Figura 5.14: Diagrama de flujo del Código de la ESP32-CAM.....	32
Figura 5.15: Nodos Para activación de motores.....	33
Figura 5.16: Dispositivos en nodos Alexa.....	33
Figura 5.17: Diagrama de flujo del Código de la ESP32.....	34
Figura 5.18: Nodos en node red.....	35
Figura 5.19: Conexiones Motores.....	35
Figura 6.1: Gráfica de barras de la Tabla 6.1.....	38
Figura 6.2: Gráfica de barras de la Tabla 6.2.....	39
Figura 6.3: Interfaz ESP32-CAM.....	42
Figura 6.4: Interfaz ESP32-CAM.....	42
Figura 6.5: Interfaz ESP32-CAM.....	43
Figura 6.6: Interfaz Visión Artificial.....	44
Figura 6.7: Error de acceso.....	45
Figura 6.8: Registro correcto.....	45
Figura 6.9: Tabla base de datos.....	45
Figura 6.10: Gráfica de barras de la Tabla 6.7.....	46
Figura 6.11: Gráfica de barras de la Tabla 6.8.....	47

## ÍNDICE DE TABLAS

Tabla 6.1: Pruebas de reconocimiento Alexa .....	38
Tabla 6.2: Pruebas de reconocimiento Alexa .....	39
Tabla 6.3: Comandos Alexa (persianas) .....	40
Tabla 6.4: Comandos Alexa (luces).....	41
Tabla 6.5: Comandos Alexa (proyector).....	41
Tabla 6.6: Comandos Alexa (proyector).....	41
Tabla 6.7: Pruebas de reconocimiento Facial con luz.....	46
Tabla 6.8: Pruebas de reconocimiento Facial sin Luz .....	46

## **CAPÍTULO 1.    Introducción**

En la actualidad, renovar tanto los espacios de trabajo como los personales mediante la automatización se ha vuelto algo muy concurrente, pues la tecnología ha ido avanzando con el fin de mejorar nuestra eficiencia y comodidad en diferentes tipos de entornos, y es de esta necesidad de mejorar de donde surge la domótica.

La domótica se refiere al control de mecanismos inteligentes que se encuentran presentes en una vivienda o un lugar de trabajo para automatizarlos. Para realizar el control de dichos mecanismos la domótica cuenta con diversas tecnologías entre ellas el reconocimiento facial y de voz. Estas tecnologías permiten la identificación y autenticación de las personas a través de sus características faciales o vocales. Por un lado, el reconocimiento de voz utiliza modelos de aprendizaje automático y algoritmos para convertir palabras habladas en texto o comandos que los sistemas domóticos pueden entender; y por otro lado la visión artificial interpreta o captura varias imágenes utilizando un alcance de perceptibilidad similar al del ser humano para que pueda comprender y actuar de acuerdo con las necesidades.

El reconocimiento de voz y facial en domótica no solo mejora la interacción y la comodidad con los dispositivos del hogar o el lugar de trabajo, sino que también aumentan la seguridad al permitir que solo personas autorizadas tengan acceso a determinadas áreas o funciones. Dadas las ventajas anteriores se optó por implementar dichas tecnologías para realizar el control domótico del laboratorio implementando sistemas de reconocimiento de voz para lo que viene a ser iluminación, persianas y motores para la subida y bajada de una plataforma; y el visión artificial para el control de cerradura de un armario donde se guardaran robots, controles, entre otras piezas y también un sistema combinado de los dos anteriores para el registro de quienes ingresen al laboratorio.

## **CAPÍTULO 2. Problema:**

### **2.1 Antecedentes:**

Para llevar a cabo experimentos y prácticas en general en los laboratorios, es fundamental contar con un ambiente controlado y seguro. La domótica, aplicada en este caso a un laboratorio de robótica, permite automatizar y controlar una variedad de aspectos del entorno, como la iluminación, la climatización, la seguridad y el suministro de energía. Esto puede mejorar la productividad operativa, la precisión de los resultados, facilitar la interacción entre los investigadores y los equipos y dar por hecho un entorno de trabajo seguro.

### **2.2 Importancia y alcances**

La implementación de domótica en los laboratorios de robótica tiene muchas ventajas. Por un lado, la gestión de la iluminación posibilita la mejora en la eficiencia del consumo energético. Por otro lado, facilita la interacción y colaboración entre las personas en el entorno del laboratorio. Estos sistemas de recursos compartidos y control de acceso permiten una mejor planificación y asignación de espacios y equipos, lo que evita problemas y maximiza la utilización eficiente de los recursos disponibles. Además, mediante la integración de sistemas de comunicación y monitoreo en tiempo real, se posibilita promover el intercambio de conocimientos y la colaboración entre los miembros del laboratorio.

Tomando en cuenta dichos beneficios, el laboratorio de robótica, debido a la presencia en el caso de un instructor, el acto de tomar asistencia o registrar la asistencia de los estudiantes que acuden al laboratorio resulta una ineficiencia en términos de tiempo para el desarrollo de la clase. También se plantea una problemática adicional con relación al uso del mobiliario, como la vitrina donde se almacenan los objetos de laboratorio, como drones, diseños 3D, joysticks, filamentos, entre otros. Si no se cuenta con un inventario adecuado de estos objetos, o si no se les brinda el debido cuidado, puede afectar significativamente al docente a cargo del laboratorio. Por otro lado, tanto las persianas como la iluminación LED del espacio de clases deberían ser controladas de manera automática, evitando la necesidad de ajustarlas manualmente como se hace actualmente.

### **2.3 Delimitación:**

El desarrollo de este proyecto tendrá lugar en el laboratorio de robótica de la UPS. El enfoque se limitará al análisis e implementación de sistemas domóticos en este laboratorio, teniendo en cuenta los aspectos de viabilidad, técnica y operativa. La delimitación temporal abarcará el tiempo necesario para implementar y evaluar los sistemas domóticos propuestos, mientras que la delimitación espacial se centrará en el entorno del laboratorio. Por ello se plantea abordar las siguientes problemáticas:

**Registro de asistencia:** Se implementará un sistema automatizado para registrar la ayuda de los estudiantes en el laboratorio. Esto reducirá el tiempo dedicado a la asistencia manual y mejorará el desarrollo de las clases.

**Gestión de inventario:** Se evaluará la implementación de un sistema automatizado de inventario para mobiliario y objetos de laboratorio como drones, diseños 3D, joysticks y filamento. Esto facilitará el control y seguimiento de los objetos, así como la asignación de responsabilidades adecuadas, evitando pérdidas y descuidos.

**Control de iluminación y persianas:** En el espacio de clases, se investigará si es posible automatizar el control de las persianas y las luces LED. Esto permitirá ajustarlos de manera rápida y precisa, mejorando el ambiente de trabajo y brindando comodidad al personal docente.

## **CAPÍTULO 3. Objetivos Generales y Específicos.**

### **3.1 Objetivo General:**

- Desarrollar un sistema domótico mediante el uso de visión artificial y reconocimiento de voz para la automatización del laboratorio de robótica de la UPS.

### **3.2 Objetivos específicos:**

- Elaborar un plan de trabajo mediante la observación del espacio para la selección de los dispositivos que se usara en el laboratorio de robótica.
- Implementar un registro automático de estudiantes mediante reconocimiento facial y audio para el control de ingreso al laboratorio de robótica.
- Realizar la implementación del sistema de control mediante audio para el control automático de los parámetros seleccionados en el laboratorio de robótica.
- Realizar pruebas de funcionamiento de los sistemas mediante experimentación en el laboratorio para la validación de la implementación propuesta.

## **CAPÍTULO 4. Fundamentos teóricos.**

### **3.3 Sistemas domóticos**

La domótica permite la gestión inteligente y automática de los hogares. Esta se refiere a un conjunto de sistemas y tecnologías que tienen el fin de automatizar una vivienda a través del control inteligente de iluminación, comunicaciones, seguridad y dispositivos de la vivienda u hogar con el fin de aumentar el bienestar, confort y seguridad. (Sarachu, 2023).

Hay una gran variedad de protocolos y sistemas disponibles para construir hogares digitales. Estos protocolos y tipos de sistemas domóticos incluyen Wi-Fi, Bluetooth, cableado, inalámbricos y PLC. Aunque en esencia, todos hablan un lenguaje diferente. Cada sistema conecta los dispositivos y les enseña cómo llevar a cabo una función específica. (Sistemas Domóticos, 2020).

### **3.4 Seguridad Biométrica**

La biometría consiste en un enfoque utilizado para reconocer a las personas a partir de características biológicas o físicas. Entre los ejemplos más conocidos de tecnología biométrica se encuentran el reconocimiento facial, el escaneo de retina y el mapeo de huellas dactilares. Los dispositivos de escaneo biométrico capturan datos para verificar la identidad, los cuales se comparan con una base de datos almacenada para determinar si se debe permitir o denegar el acceso al sistema. En otras palabras, la seguridad biométrica utiliza las características únicas de su cuerpo como "llave" para acceder a determinados lugares o información. (Kaspersky, 2023).

#### **3.4.1 Reconocimiento facial**

El reconocimiento facial consiste en el uso de una técnica para confirmar o autenticar la identidad de una persona mediante el análisis de su rostro. Estos sistemas pueden emplearse para identificar individuos en fotografías, videos o en tiempo real. Frecuentemente, el reconocimiento facial no se basa en una extensa base de datos de imágenes para establecer la identidad de alguien; en cambio, su enfoque principal es detectar y reconocer al propietario legítimo del dispositivo, al mismo tiempo que limita el acceso a otras personas.

Además de su aplicación en el desbloqueo de dispositivos móviles, el reconocimiento facial se extiende a la detección de similitudes entre los rostros de individuos captados por cámaras especiales e imágenes almacenadas en un registro. Los sistemas de tecnología facial pueden variar, pero generalmente funcionan de la siguiente manera: (RecFaces, 2022)

El primer paso es la detección de rostros, donde la cámara localiza y identifica una imagen de un rostro, ya sea individual o en medio de una multitud, incluso en diferentes orientaciones, como mirando de frente o de perfil.

En el siguiente paso, se lleva a cabo el análisis de rostro, que recopila y analiza una fotografía del rostro. La mayoría de las tecnologías de reconocimiento facial se centran en imágenes 2D en lugar de imágenes 3D porque son más prácticas al comparar imágenes 2D de fotografías públicas o de una base de datos. El software analiza y evalúa la geometría del rostro utilizando variables importantes como distancia, profundidad, la forma y contornos faciales en general. Con el fin de determinar puntos de referencia faciales esenciales para diferenciar un rostro particular.

En el tercer paso, los datos se convierten en imágenes de rostros a través del proceso de captura facial. Este proceso implica la conversión de los datos analógicos del rostro en un conjunto de datos digitales que se basan en los rasgos faciales de la persona. Básicamente, el análisis facial se convierte en una fórmula matemática, lo que produce una huella facial única y numérica que representa al individuo.

En el cuarto paso del proceso, se realiza la búsqueda de una coincidencia. Para ello, se hace uso de una base de datos que contiene rostros previamente conocidos, y se compara con la huella facial obtenida. A diferencia de las huellas dactilares y el iris, el reconocimiento facial se considera una de las medidas biométricas más naturales porque intuitivamente reconocemos a nosotros mismos y a los demás al mirar sus caras. (Kaspersky, 2022).



### **3.4.2 Reconocimiento de Voz**

El área interdisciplinaria de la lingüística computacional, denominada reconocimiento de voz, se concentra en el desarrollo de tecnologías que permiten a las computadoras reconocer y transcribir el lenguaje hablado en forma de texto. Este campo es también conocido como reconocimiento de voz a texto, voz automatizada o síntesis de voz en computadora. Engloba el conocimiento e investigación en campos como lingüística, informática e ingeniería eléctrica como resultado de su desarrollo y aplicación (RecFaces, 2021).

El proceso de reconocimiento de audio se divide en tres etapas:

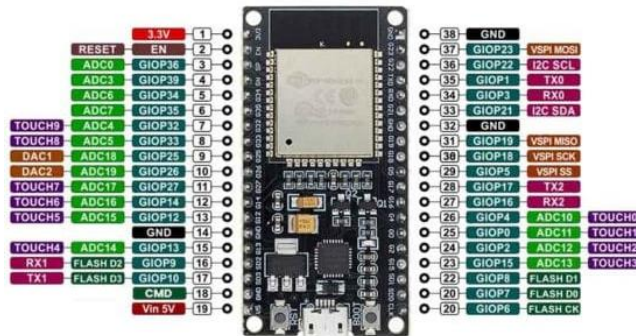
- **Detección de palabras:** En esta fase, se resaltan las palabras presentes en el audio, lo que puede generar varias hipótesis de reconocimiento para una misma palabra.
- **Evaluación mediante un modelo de lenguaje:** Las diferentes hipótesis de reconocimiento se someten a una evaluación con un modelo de lenguaje. Este modelo verifica la coherencia de la nueva palabra en relación con las palabras ya reconocidas anteriormente.
- **Procesamiento del texto reconocido:** En la última etapa, el texto reconocido se procesa para convertirlo a su forma numérica, agregar signos de puntuación como guiones, entre otros. El resultado final de este proceso de reconocimiento se envía como respuesta (RecFaces, 2021).

### **3.5 ESP32**

Los ESP32 son una serie de tarjetas de bajo consumo de energía y costo asequible, desarrollados por la empresa Espressif Systems. Estos módulos son una evolución del ESP8266 bien conocido, y su capacidad para soportar conexiones Wi-Fi y Bluetooth es su característica más notable. Debido a su eficiencia y funcionalidad, es una opción muy valorada para una variedad de aplicaciones y proyectos (Carmenate, 2022). En la figura4.1 se muestra la tarjeta ESP32 de 38 pines.

Figura 4.1: Tarjeta ESP32

**PINOUT**  
**ESP32 38 PINES ESP WROOM 32**

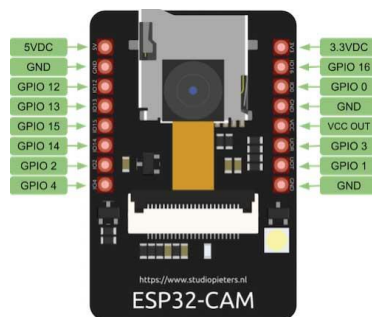


Tarjeta ESP32 de 38 pines PIN/POUT (Morales, 2023)

### 3.6 ESP32-CAM

El módulo ESP32-CAM (Figura 4.2) es una versión compacta y de bajo consumo de energía que integra una cámara, desarrollado sobre la plataforma ESP32. Está equipado con una cámara OV2640 y cuenta con una ranura para tarjeta TF integrada. Su versatilidad le permite trabajar adecuadamente con el Internet de las cosas (IoT), como monitoreo inalámbrico de video, transmisión de imágenes a través de Wi-Fi, reconocimiento de códigos QR, entre otros. Es ampliamente utilizado en diferentes proyectos y aplicaciones relacionadas con el IoT (Netikon.gr, 2023).

Figura 4.2: Tarjeta ESP32-CAM



ESP32-CAM PIN/OUT (Pascual, 2022)

### **3.7 Arduino**

La plataforma Arduino es una aplicación que combina tanto hardware como software de libre acceso, siendo su diseño flexible y de sencilla utilización tanto para creadores como para desarrolladores. Con esta plataforma, es posible construir diversos microordenadores integrados en una sola placa, los cuales pueden ser utilizados de múltiples maneras por la comunidad de creadores, según sus necesidades y proyectos específicos (Fernández, 2022).

### **3.8 ArduinoWebsocket**

La biblioteca proporciona una interfaz sencilla y amigable para trabajar con WebSockets, tanto en su función de cliente como de servidor. WebSockets se fundamenta en el protocolo TCP, facilitando la transmisión de datos entre redes. Su confiabilidad y eficiencia lo convierten en la elección predominante para la mayoría de los usuarios. Por otro lado, TCP permite establecer una comunicación punto a punto, denominada sockets, habilitando así el intercambio bidireccional de datos.

Cuando se establecen conexiones bidireccionales, como las habilitadas por WebSocket, se permite intercambio simultáneo de datos, lo que agiliza el acceso a la información. De manera específica, esta biblioteca permite una comunicación directa y en tiempo real entre su servidor y cualquier aplicación web. De esta manera, la información solicitada se muestra en tiempo real, lo que enriquece la experiencia del usuario al obtener información de manera instantánea. (Desarrollo web, 2022).

### **3.9 Python**

Python como lenguaje de programación goza de gran popularidad y cuenta con una gran gama de aplicaciones para el desarrollo de aplicaciones web, ciencia, software de datos y aprendizaje automático (ML). Los desarrolladores eligen Python debido a su eficiencia y facilidad de aprendizaje, además de su capacidad para funcionar en diversas plataformas. El software de Python se puede adquirir de manera gratuita e integrar en todo tipo de sistemas, lo que permite el desarrollo de proyectos (Lahtela & Kaplan, 2023)

### **3.10 TensorFlow**

TensorFlow es una biblioteca que permite a las personas a realizar cálculos numéricos de alta calidad. Su arquitectura sumamente adaptable permite que esta biblioteca sea utilizada en distintas plataformas entre ellas CPU y GPU que son las más generales y hasta TPU (Tensor Processing Unit), posibilitando su aplicación en equipos de escritorio, dispositivos móviles o periféricos y clústeres de servicios. Numerosos campos científicos utilizan su núcleo flexible de computación numérica. TensorFlow se encuentra bajo la licencia Apache 2.0, lo que significa que se puede usar y compartir libremente (Tensorflow, 2023).

### **3.11 OpenCv**

OpenCV es una biblioteca de código abierto que engloba una amplia gama de implementaciones con más de 2500 algoritmos, especializada en visión artificial y aprendizaje automático. Entre las funciones con las que cuenta se encuentran el reconocimiento facial para identificar objetos o rostros, la búsqueda de imágenes similares, la corrección de ojos rojos en fotografías, el reconocimiento de escenarios, el seguimiento de movimientos oculares, la clasificación de acciones humanas en videos y la extracción de modelos 3D (Rodriguez, 2021).

### **3.12 Tkinter**

El módulo "tkinter", también conocido como "interfaz Tk", es una interfaz gráfica predeterminada de Python para el uso de herramientas de GUI Tk. Estas librerías están disponibles para la mayoría de los sistemas operativos Unix y Windows. Es importante destacar que Tk no es una parte esencial de Python, sino que está mantenido por ActiveState. Tkinter puede ser compilado con o sin soporte de hilos y es compatible con versiones de TCL/TK. (Python, 2023).

### **3.13 Matplotlib**

Matplotlib es una biblioteca útil para Python que permite visualizar de manera estática, animada e interactiva una gran variedad de gráficas. Esta herramienta tiene la capacidad de generar gráficos de alta calidad aptos para publicaciones en diversos formatos impresos y ambientes interactivos. Matplotlib puede usarse en s Python, servidores web y en distintas herramientas de interfaz gráfica de usuario, lo que lo hace muy versátil (Matplotlib, 2023).

### 3.14 Alexa

El asistente virtual Alexa lanzado en noviembre de 2014 junto con los dispositivos inteligentes Echo, opera mediante el control de voz. Este funciona de manera similar a otros asistentes como son Cortana, Siri y Google Assistant. Alexa se activará al pronunciar su nombre, ejecutándose así el altavoz integrado y preparándose para escuchar las instrucciones verbales del usuario. A través del reconocimiento de voz, el asistente responderá adecuadamente a las preguntas o comandos proporcionados (FERNÁNDEZ, 2023).

Figura 4.3: Echo DOT de Alexa



Dispositivo para la ejecución de Alexa (Amazon, 2022)

### 3.15 Dimmer

Un dimmer es un mecanismo que permite que un interruptor de pared, que generalmente tiene forma de rueda o pulsador, regule la intensidad de las luces. En cualquier momento del día, al accionar el interruptor, la intensidad de la luz se puede aumentar o disminuir gradualmente hasta alcanzar el nivel deseado. Este sistema se puede instalar de forma inalámbrica o mediante cableado en hogares, oficinas o cualquier otro lugar que necesite controlar la intensidad de las luces (Canalejo, 2021).

Figura 4.4: Atenuador de luz inteligente

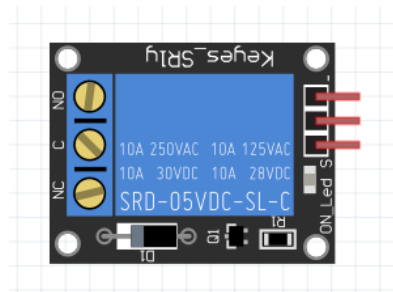


Dimmer Nexxt Home (Next Home,2023)

### 3.16 Relé

El funcionamiento de un relé se basa en una bobina con conexión a una fuente de alimentación. El relé produce un campo electromagnético al activar la bobina provocando el cierre del contacto del relé, el cual está normalmente abierto. A través de esto se habilita el paso de corriente a través del circuito normalmente cerrado, lo que permite el encendido de una lámpara o el arranque de un motor. Sin embargo, el campo electromagnético desaparece cuando se corta la corriente a la bobina y el contacto del relé vuelve a abrirse. Esto interrumpe el flujo de corriente hacia el circuito eléctrico que estaba conectado a la lámpara o motor (SEAS, 2019).

Figura 4.5: Relé

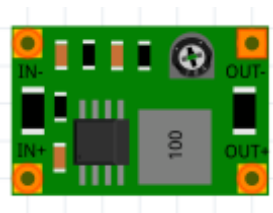


Relé con activación a 5V (Fritzing Demo, 2007-2016)

### 3.17 CONVERTIDORES DC/DC – BUCK

Son dispositivos electrónicos que utilizan componentes como bobinas y capacitores para cambiar los niveles de voltaje de entrada a niveles adicionales deseados. Estos componentes almacenan energía temporalmente y luego la descargan para obtener el voltaje de salida deseado. Los convertidores Buck operan con el voltaje de entrada para entregar un menor voltaje de salida. Esta es una situación común en electrónica, y cuando es necesario, se utiliza el conocido regulador 7805 o LM317 para obtener un voltaje variable (Argos, 2019).

Figura 4.6: Conversor Buck



Convertidor para la regulación de voltaje (Fritzing Demo, 2007-2016)

## CAPÍTULO 5. Implementación

Para llevar a cabo la domotización del laboratorio de robótica, fue necesario seleccionar un método para controlar el laboratorio mediante comandos de voz. Durante esta etapa, se exploraron diversas opciones, incluyendo el desarrollo de código propio y la utilización de servicios de asistentes virtuales existentes. Al comparar ambas alternativas, se llegó a la conclusión de que resulta considerablemente más eficiente implementar todos los sistemas en función de un dispositivo ya existente. De lo contrario, al configurar los dispositivos con un sistema de reconocimiento de voz desarrollado por estudiantes, se encontrarían limitaciones significativas. En contraste, al emplear un dispositivo como Alexa, se abre la posibilidad de que los docentes puedan implementar nuevos sistemas a futuro.

De la misma manera en el proceso de domotización del laboratorio de robótica del laboratorio, se decidió implementar un sistema de reconocimiento facial a través de visión artificial para realizar un registro de estudiantes y otro sistema para abrir una cerradura automáticamente mediante comandos de voz. A continuación, se presentan los dispositivos utilizados en el proceso de domotización del laboratorio de robótica de la UPS, junto con una guía detallada de cómo configurarlos.

### 3.18 Dispositivo Echo Dot (ALEXA)

El dispositivo Echo Dot (Figura 5.1) adquirido fue uno de 5ta Generación, para comenzar a usarlo se debe realizar las configuraciones iniciales dentro de la app de Alexa que se puede adquirir mediante del enlace de Google Play o App Store.

Figura 5.1: Echo Dot

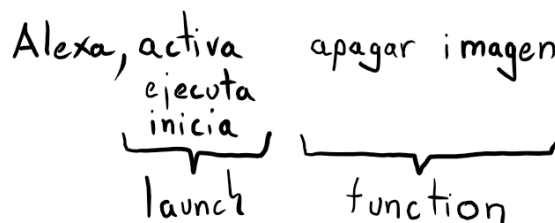


Dispositivo para ejecución de Alexa (Amazon 2022)

Alexa otorga un gran número de usos al usuario como reproducir música, hacer preguntas, controlar dispositivos, entre otros. Dentro de dichos usos en el que se concentro es el control de dispositivos, para ello Alexa utiliza el reconocimiento de voz para realizar los comandos dados por el usuario.

Es importante tomar en cuenta que Alexa necesita seguir una estructura (palabra de invocación, palabra de ejecución, función a realizar) para la activación de ciertos comandos, ejemplo en figura 5.2, de lo contrario puede haber ciertos conflictos con la activación adecuada de los comandos.

Figura 5.2: Funciones de activación de Alexa

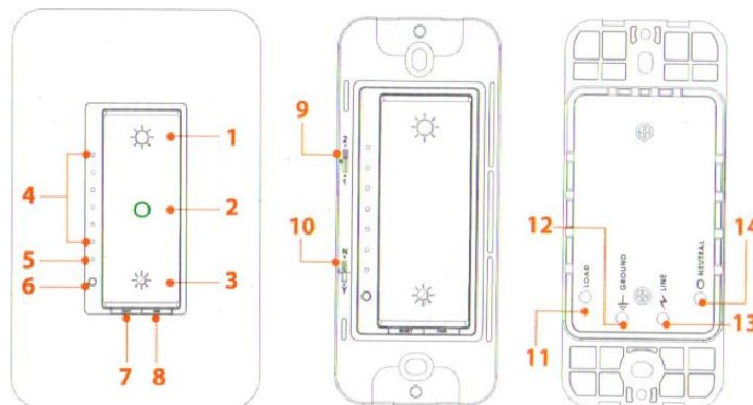


Orden de invocación funciones Alexa

### 3.19 Dimmer NEXT HOME/NHE-D100

Uno de los dispositivos usados fue el Dimmer de marca Next Home (Figura 5.3), este Dimmer como cualquier otro permite la regulación de la luz, pero también se puede usar con focos no regulables. A continuación, se muestran características del dispositivo, así como su configuración para establecer conexión con Alexa.

Figura 5.3: Dimmer Next HOME/NHE-D100



Especificaciones para conexión y manejo (Next Home,2023)



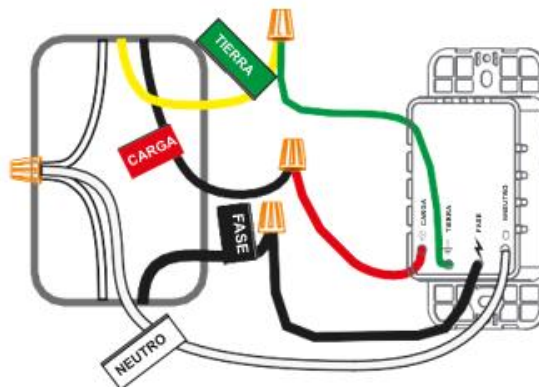
1. Control de brillo de la bombilla
2. Indicador LED de conexión
3. Control de brillo de la bombilla
4. Luces LED indicadoras de brillo
5. Indicador Wi-Fi
6. Botón para los Led de intensidad
7. Botón de reposición
8. Botón de emparejamiento
9. Conmutador DIP 1
10. Conmutador DIP 2
11. Cable del circuito (carga, rojo)
12. Cable a tierra (verde)
13. Cable bajo tensión (fase, Negro)
14. Cable neutro (blanco)

### **Pasos para conexión y configuración del Dimmer:**

#### **Conexión del Dimmer:**

El Dimmer tiene dos conmutadores DIP, que se describen en la tabla del anexo 10.7. Se debe configurar estos conmutadores según las necesidades del usuario, asegurándose de ajustarlos correcta y posteriormente realizar las conexiones mostradas en la figura 5.4.

Figura 5.4: Conexión del atenuador inteligente



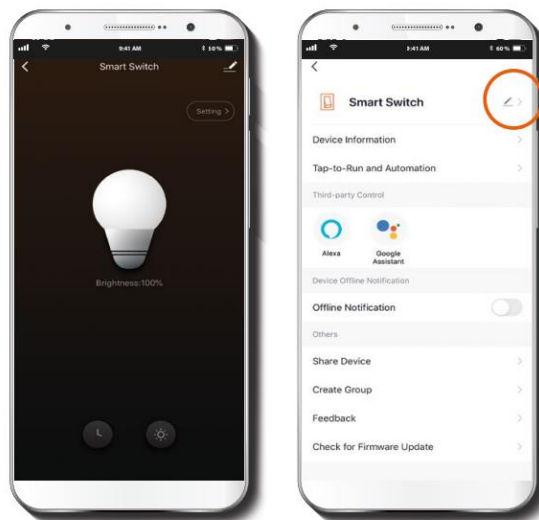
Conexión de cables tierra, carga, fase y neutro (Next Home,2023)

### Configuración para conexión con Alexa

La aplicación Nexxt Solutions Home permite a los usuarios controlar el atenuador inteligente de manera sencilla y práctica. El usuario debe descargar la aplicación desde Google Play o Apple App Store.

Una vez conectado, el dispositivo se verá en la pantalla de la aplicación como se muestra en la figura 5.5, desde donde se puede controlar el brillo y el estado de la luz. Además, ofrece la posibilidad de establecer una conexión con Alexa para controlar el dispositivo mediante comandos de voz, lo que agrega una mayor comodidad en su uso diario.

Figura 5.5: Ventana Dispositivo Next



Ventanas con el dispositivo configurado (Next Home,2023)

### 3.20 Allone Pro – Persianas - Proyector

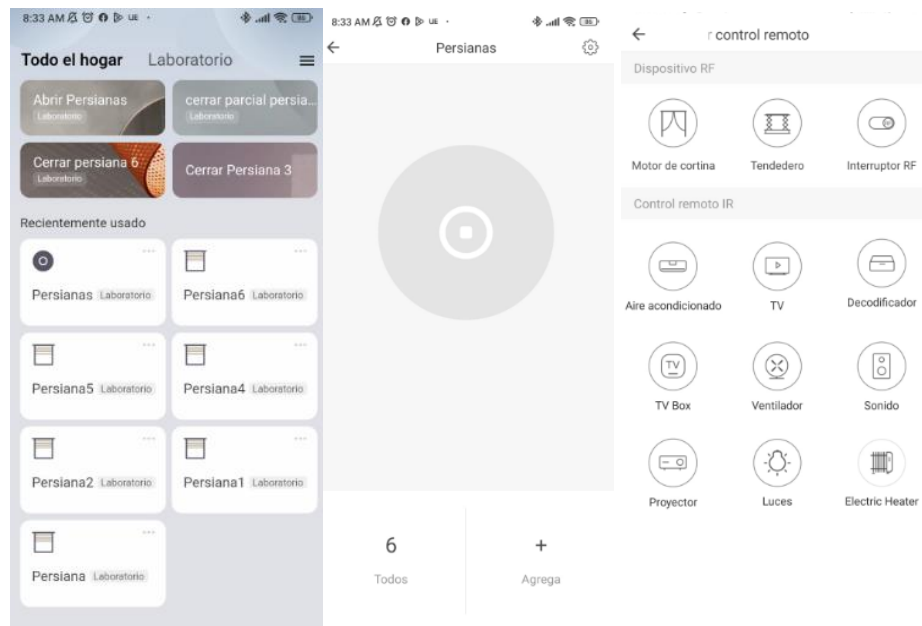
El dispositivo Orbivo Allone Pro (figura 5.6) tiene el funcionamiento de un control remoto, gracias a esto tiene una compatibilidad con una gran gama de dispositivos, permitiendo así el control automático de los dispositivos deseados como en este caso fueron las persianas y el proyector.

Figura 5.6: Orbivo Allone Pro



Dispositivo para domotización (Orbivo, 2023)

Figura 5.7: Interfaz Orbivo Allone Pro



Selección para agregar dispositivos (Orbivo Home, 2023)

Dentro de la aplicación Orbivo (Figura 5.7) se puede vincular cualquier dispositivo deseado se debe seleccionar la marca correcta y asegurarse que el Orbivo se encuentre cerca al dispositivo a conectar. Una vez establecida la conexión se pueden establecer ambientes para los dispositivos por medio de la creación de escenas aquí se puede vincular el dispositivo acorde y hacerle realizar una acción, de lo contrario Alexa solo hará acciones por default como abrir o cerrar.

### 3.21 Sistema reconocimiento facial para abrir cerradura

Para dicho sistema de reconocimiento facial se usó una ESP32-CAM. El programa de Arduino funciona junto con una interfaz Http que permite escoger opciones para implementar el reconocimiento facial y la gestión de autorizaciones. que permite la interacción del usuario a través de una interfaz web y el control remoto del acceso mediante una cámara y un relé. Para poder cargar el código en la tarjeta adecuadamente se deben tener cargadas particiones (anexado en la tabla 10.8 “rzo-partitions”) en el programa de Arduino en las localizaciones mostradas en la figura 5.8 además de una partición en la carpeta boards donde se agregarán las líneas mostradas en la figura 5.9.

Figura 5.8: Instalación de particiones

```
Arduino IDE installed from the Windows Store
C > Users > *your-user-name* > Documents > ArduinoData > packages > esp32 > hardware > esp32 > 1.0.2 > tools > partitions

Arduino IDE installed from the Arduino website:
C > Users > *your-user-name* > AppData > Local > Arduino15 > packages > esp32 > hardware > esp32 > 1.0.2 > tools > partitions
```

#### Localizaciones para encontrar particiones

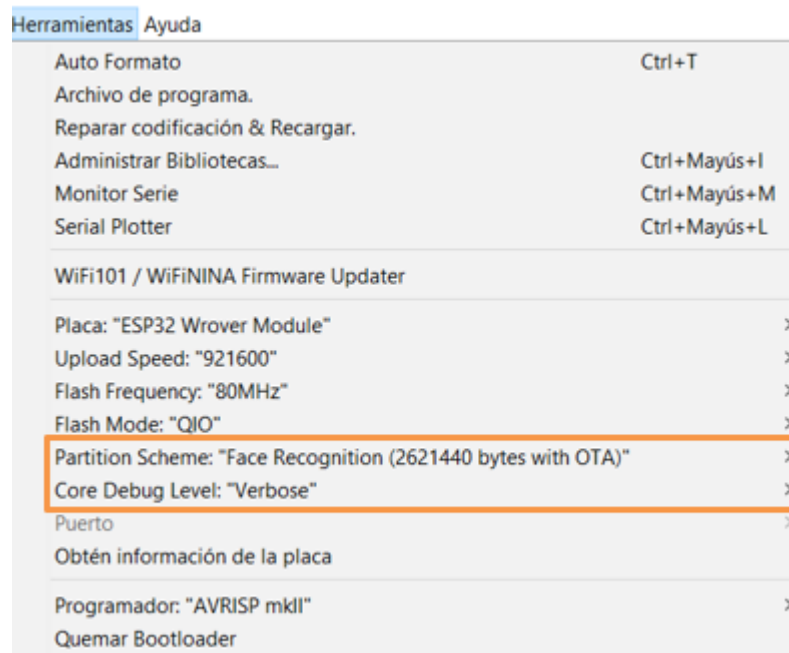
Figura 5.9: Líneas para particiones Boards

```
esp32wrover.menu.PartitionScheme.rzo_partition=Face Recognition (2621440
bytes with OTA)
esp32wrover.menu.PartitionScheme.rzo_partition.build.partitions=rzo_partitio
ns
esp32wrover.menu.PartitionScheme.rzo_partition.upload.maximum_size=2621
440
```

#### Particiones para cargar adecuadamente código

Una vez instaladas las particiones correspondientes se debe considerar las siguientes configuraciones mostradas en la figura 5.10 , y se podrá cargar adecuadamente el código a la ESP32-CAM. Esto se hace debido a que los ESP32 son dispositivos con recursos limitados. Permitiéndole almacenar los datos esenciales del reconocimiento facial, como rostros inscritos, permitiendo su gestión y actualización remota.

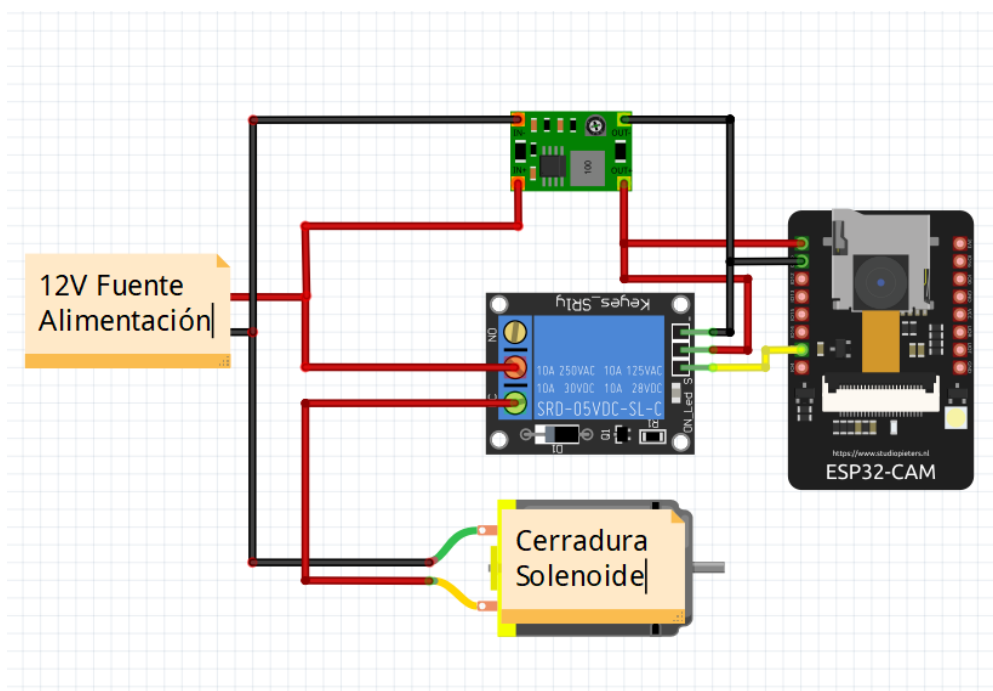
Figura 5.10: Configuraciones para ESP32-CAM



Configuraciones disponibles luego de agregar particiones

A continuación, en las figuras 5.11 y (5.12 - 5.13 - 5.14), se presenta un diagrama del circuito, así como el diagrama de flujo del código de Arduino respectivamente del sistema de reconocimiento facial.

Figura 5.11: Circuito ESP32-CAM



Conexiones para abrir cerradura mediante código (Fritzing Demo, 2007-2016)

Figura 5.12: Diagrama de flujo del Código de la ESP32-CAM

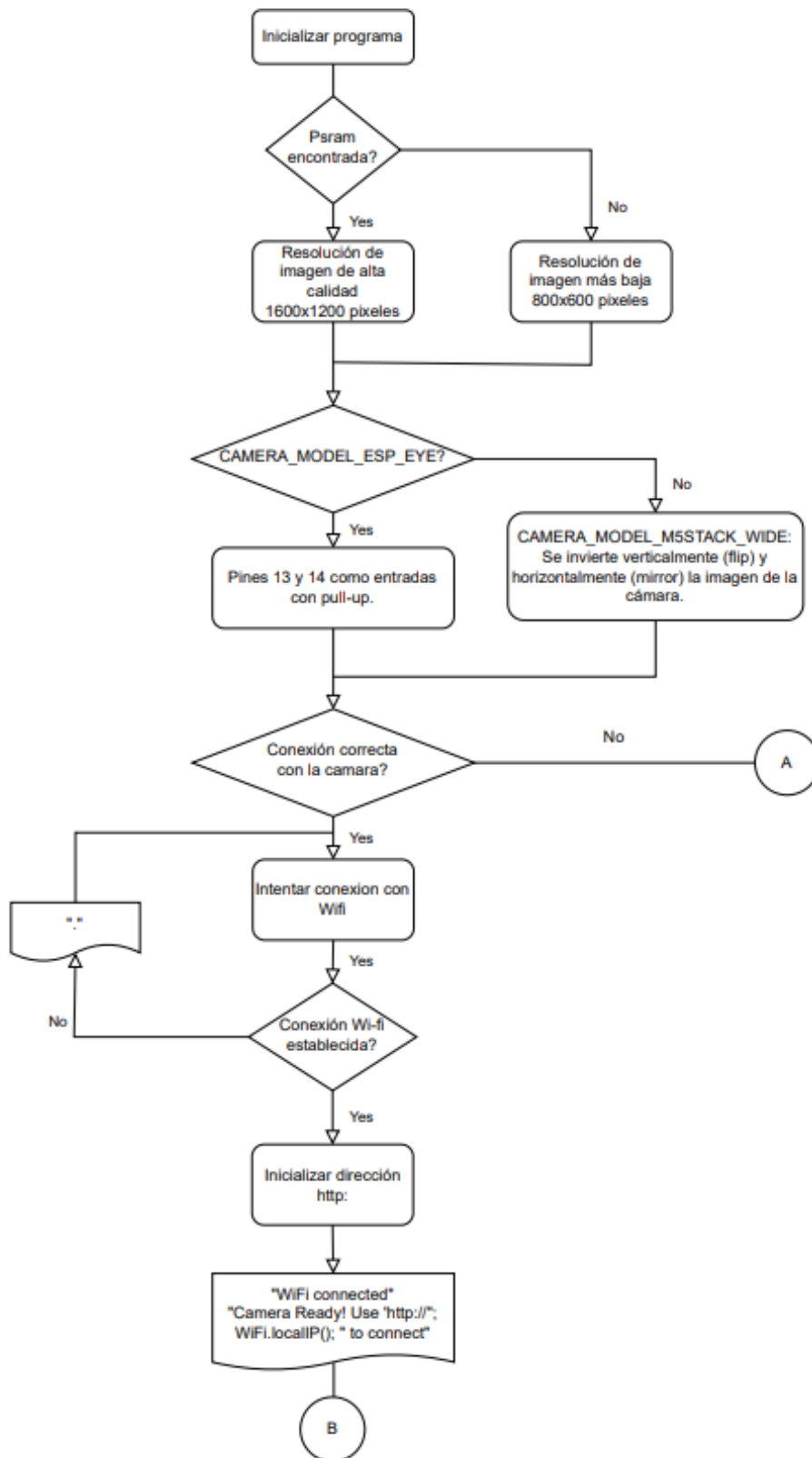


Figura 5.13: Diagrama de flujo del Código de la ESP32-CAM

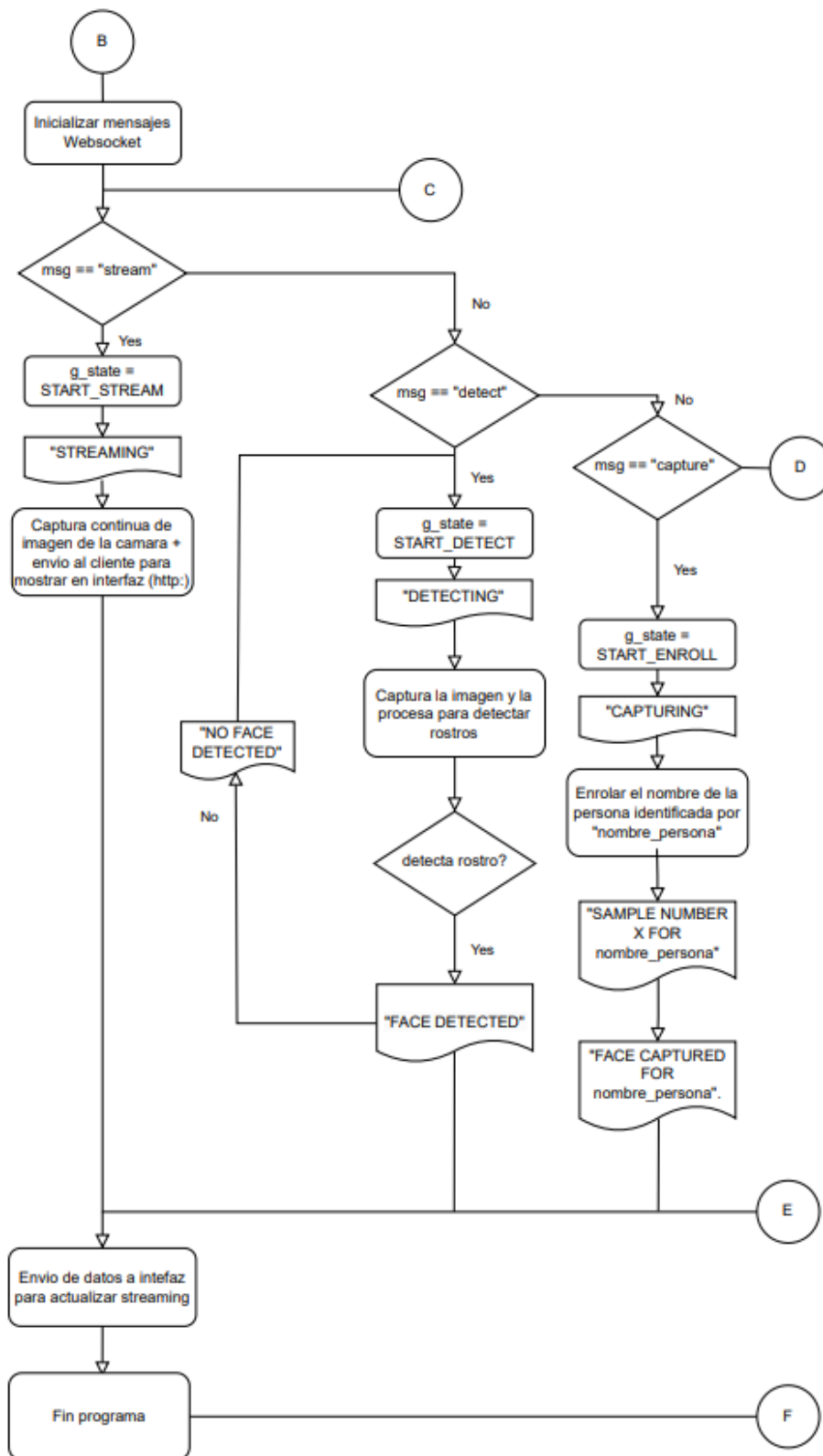
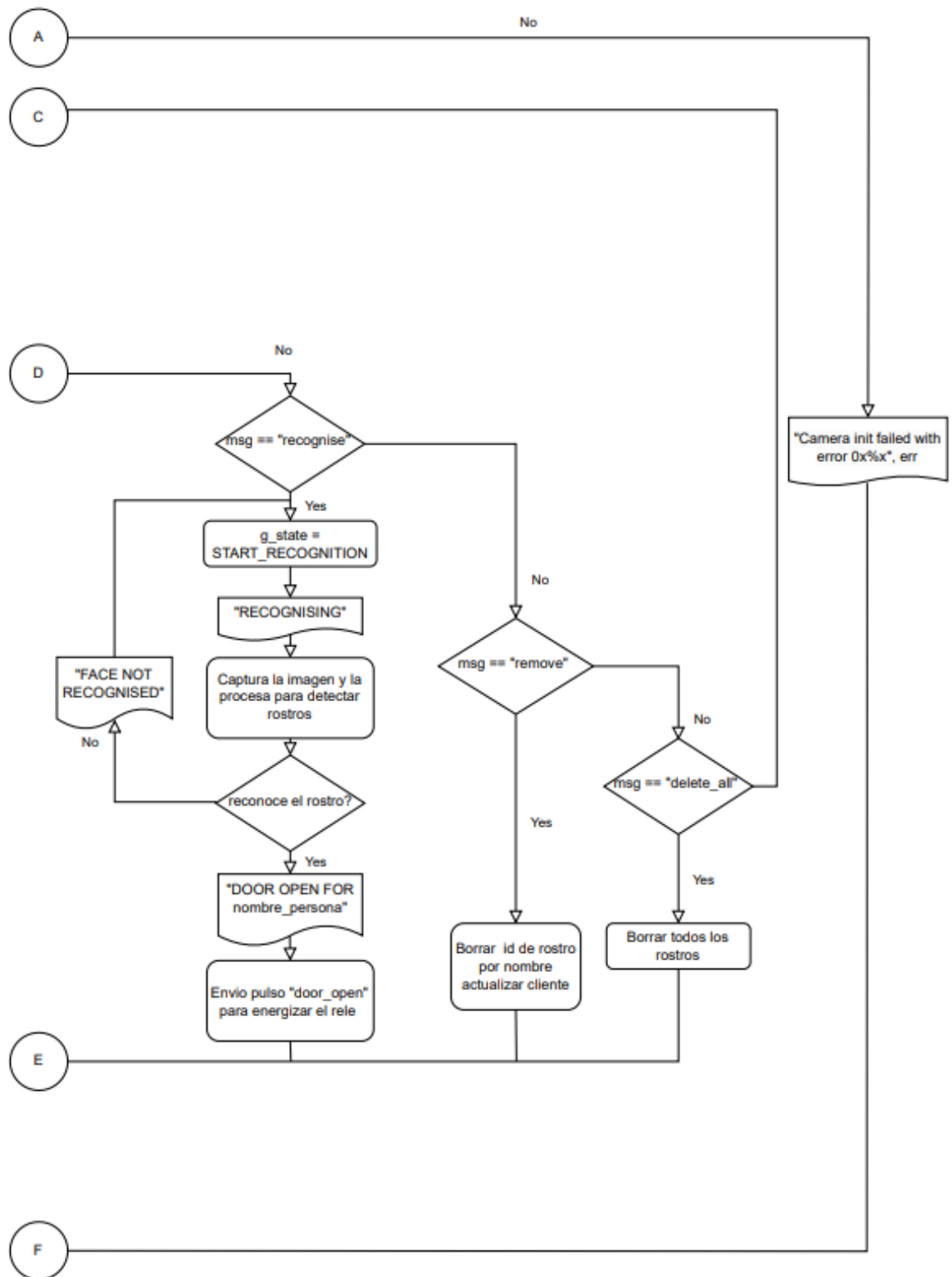


Figura 5.14 Diagrama de flujo del Código de la ESP32-CAM

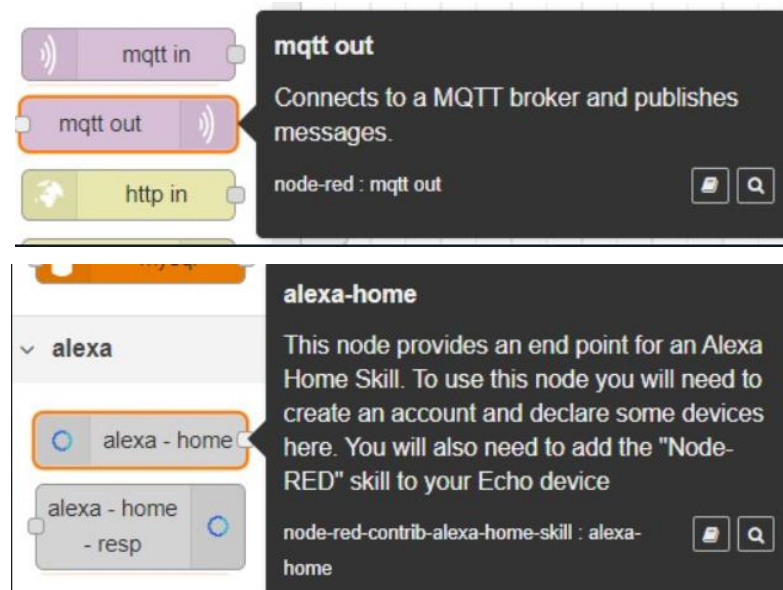




### 3.22 Motores para plataforma

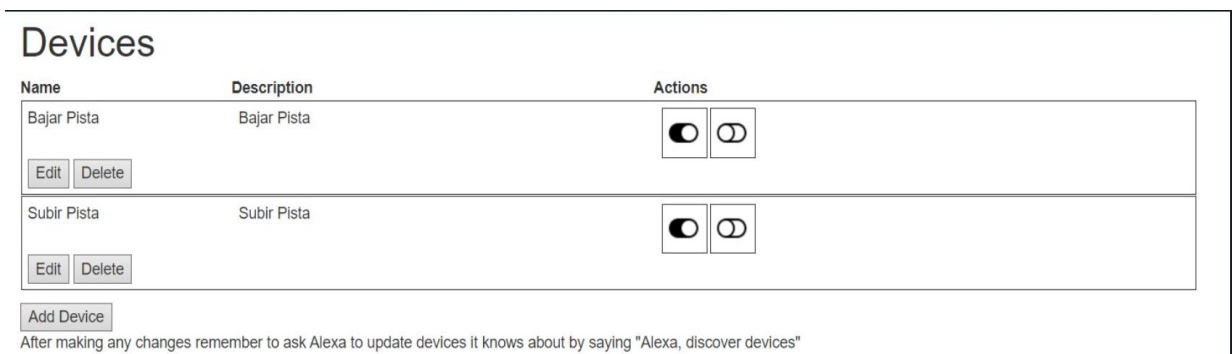
Para los motores se trabaja con un código de Arduino para controlar el movimiento de una pista elevadora mediante un encoder y una conexión MQTT para recibir comandos externos. A continuación, se presenta conexiones con Node RED (Figura 5.15 - 5.16 - 5.18) para establecer los comandos de voz con Alexa, Diagramas de flujo del código (5.17) que controla los motores mediante encoders y las conexiones eléctricas realizadas (5.19).

Figura 5.15: Nodos Para activación de motores



Nodos de Alexa y MQTT (NodeRed, 2023)

Figura 5.16: Dispositivos en nodos Alexa



Dispositivos añadidos para activación de motores (Alexa, 2023)

Figura 5.17: Diagrama de flujo del Código de la ESP32

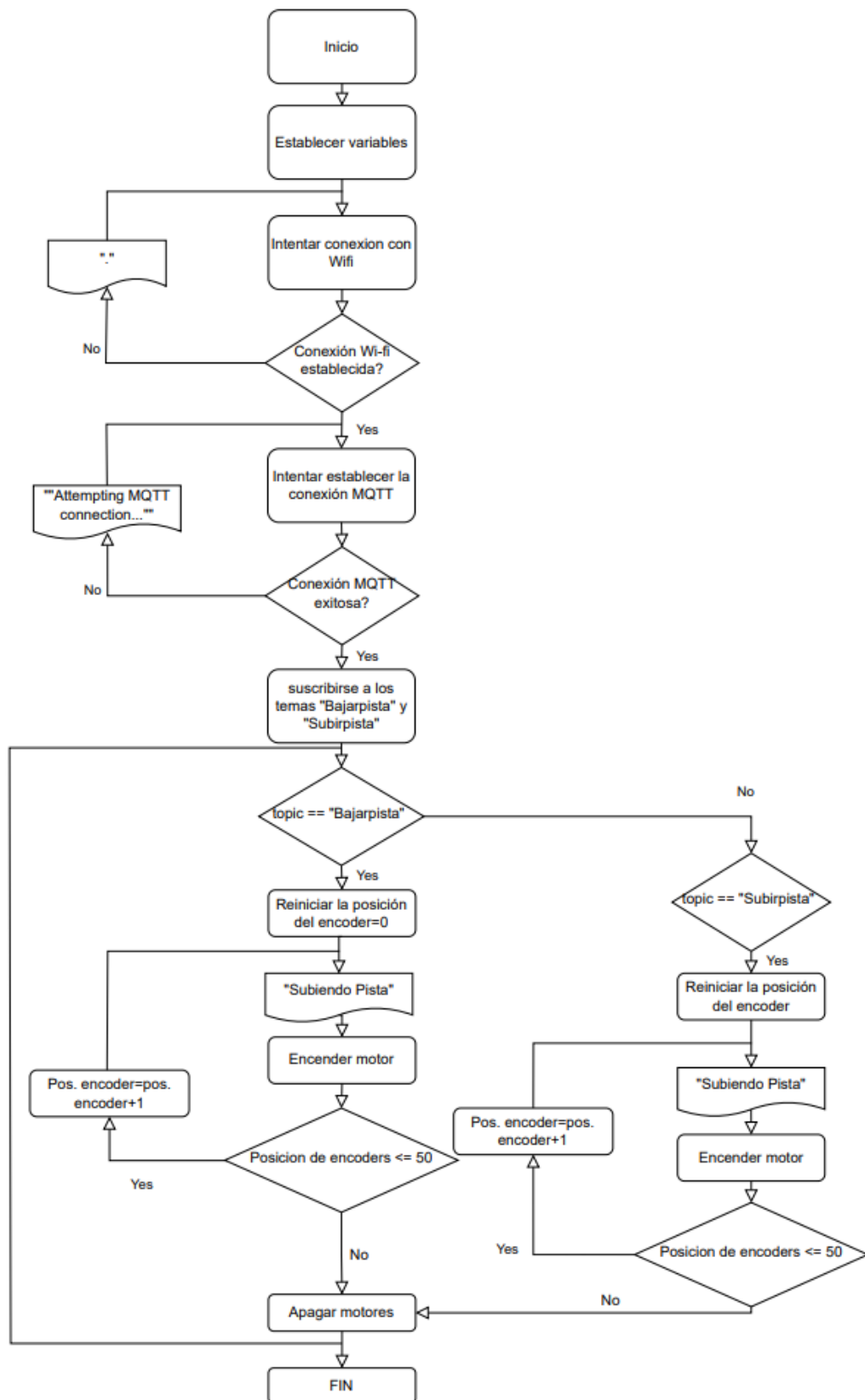
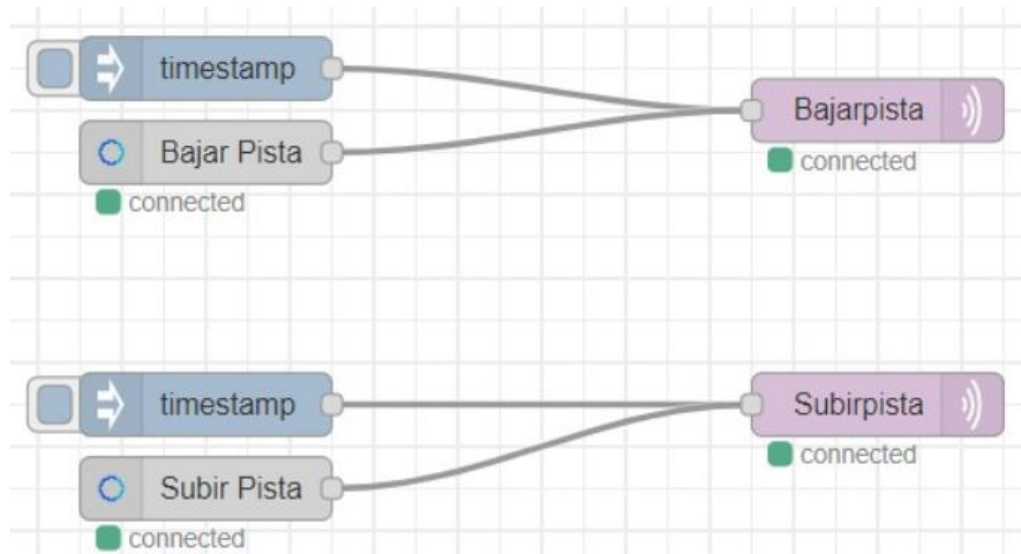


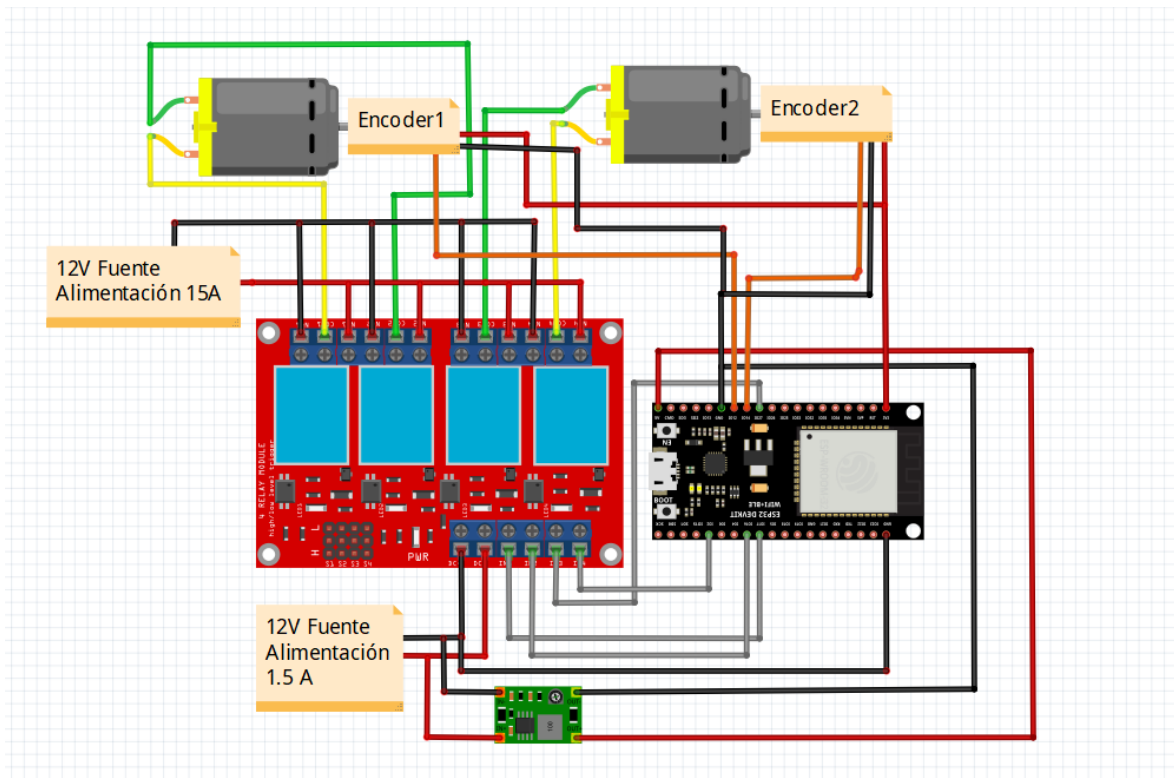
Diagrama de flujo ESP32 para plataforma

Figura 5.18: Nodos en node red



Conexión de nodos para activación de motores

Figura 5.19: Conexiones Motores



Alimentación con relés para activación de motores (Fritzing Demo, 2007-2016)

Para la implementación de cada motor en un contexto específico, como en el caso de operar una plataforma con una carga determinada, es esencial considerar varios factores técnicos críticos. Estos factores incluyen el torque del motor, la capacidad

máxima de carga, el peso propio del motor, su velocidad de operación y, por último, la inercia del sistema. En función de esta información relevante, se llevó a cabo un meticuloso cálculo y selección de los motores más apropiados, teniendo en cuenta datos precisos tanto del motor como de la carga esperada que la plataforma debe soportar, como se documenta en detalle en el anexo 10.3 del presente informe.

### **3.23 Visión Artificial para registro de estudiantes.**

En la implementación de visión artificial se empleó un código en Python la cual se integra con una base de datos para registrar estudiantes mediante el ingreso de su nombre y una aproximación óptima a la cámara, para garantizar un registro preciso. Para la visión artificial, se dio una investigación para identificar las librerías adecuadas a utilizar. Dichas librerías abarcan aspectos como interfaz del programa, comunicación entre ventanas, interacción con la base de datos y la comparación entre un usuario registrado y la imagen analizada almacenada en la base de datos. El objetivo, crear un sistema eficiente y preciso para el registro y reconocimiento de estudiantes utilizando técnicas de visión artificial mediante una base de datos para almacenar información.

En el proceso de implementación del sistema, se efectuaron series de pasos secuenciales que garantizan el buen funcionamiento de la base de datos y la interacción con los usuarios. Dado el planteamiento, inicia ejecutando un código "keys.json", el cual contiene información crítica, incluida la contraseña necesaria para acceder a la base de datos. Esta acción indispensable es para establecer una conexión correcta con la base de datos en cuestión.

Ejecutar el código "database.py", mismo que se encarga de la conexión con la base de datos y manejar los datos relacionados con el registro y "log in" los usuarios. Debido a este código, se logra la interacción fluida entre la base de datos y la aplicación, permitiendo recibir y enviar datos de manera eficiente.

El código del archivo "código.py", el cual contiene toda la interfaz del sistema y hace uso de la conexión importada del código "database.py". Aquí se lleva a cabo la recepción de datos necesarios para su almacenamiento MySQL (base de datos), asegurando así una correcta sincronización

de información relevante.

Para garantizar la óptima funcionalidad de la base de datos en MySQL, se analiza una tabla "user" dentro de la base de datos existente. Esta tabla desempeñará un papel crucial al clasificar los nombres de los estudiantes y los registros de sus rostros capturados, posibilitando así un eficaz manejo de datos para el sistema en general.

## CAPÍTULO 6. Resultados

### 3.24 Alexa (Luces/Proyector/Persianas/Motores)

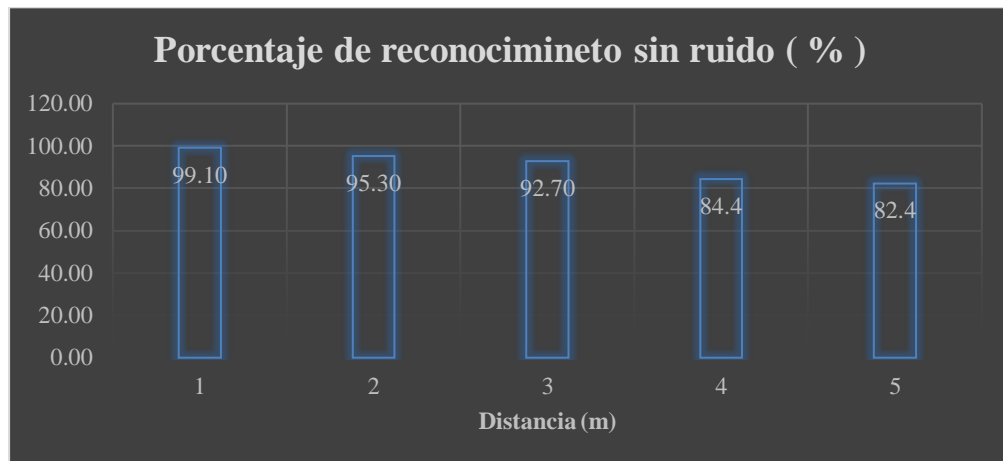
Para evaluar el funcionamiento de cada uno de los sistemas de reconocimiento de voz, se llevaron a cabo pruebas en diversas ocasiones y ambientes, que incluyeron conversaciones cercanas, habla de alumnos y escenarios con música activa. Frente a estas perturbaciones, se observó el comportamiento de Alexa de la siguiente manera:

Tabla 6.1: Pruebas de reconocimiento Alexa

Reconocimiento de voz (Alexa)											
Distancia de detección sin ruido en el entorno (m)	Activación de comando de voz luego de la orden dada, medida en %										Porcentaje de reconocimiento sin ruido (%)
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
1	95	100	99	100	100	99	99	100	99	100	99,10
2	98	95	96	96	94	95	90	99	92	98	95,30
3	99	89	93	94	92	92	94	90	85	99	92,70
4	89	99	86	87	81	85	82	81	79	75	84,4
5	80	75	85	80	95	69	77	86	90	87	82,4

Reconocimiento de voz por distancia sin ruido en entorno

Figura 6.1: Gráfica de barras de la Tabla 6.1



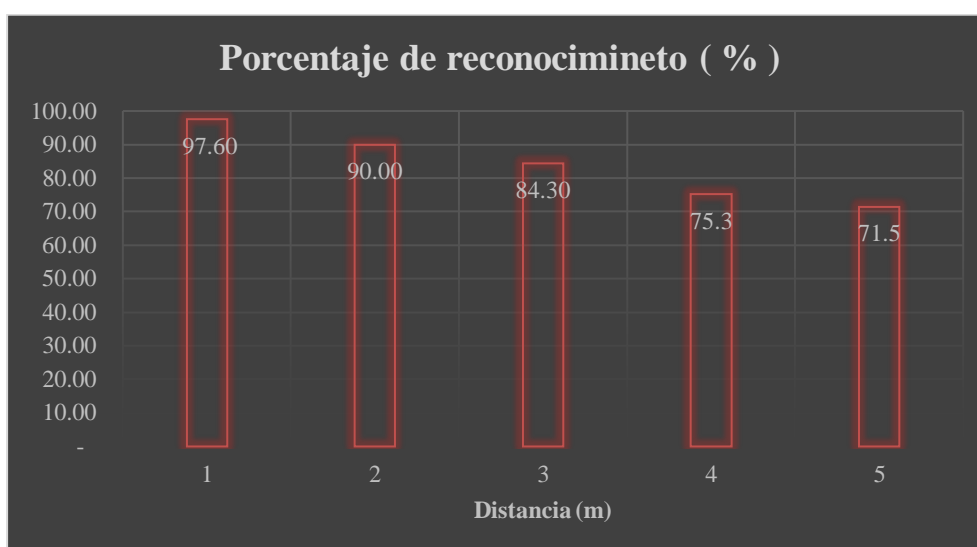
Reconocimiento de voz por distancia sin ruido en entorno

Tabla 6.2: Pruebas de reconocimiento Alexa

Reconocimiento de voz (Alexa)											
Activación de comando de voz luego de la orden dada, medida en %											Porcentaje de reconocimiento (%)
Distancia de detección con ruido en el entorno (m)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
1	95	93	96	99	100	98	98	100	97	100	97,60
2	98	92	92	91	85	90	87	88	89	88	90,00
3	99	86	85	82	70	92	79	80	80	90	84,30
4	66	78	77	80	65	70	79	80	79	79	75,3
5	60	75	70	80	85	50	77	75	70	73	71,5

Reconocimiento de voz por distancia con ruidos en el ambiente

Figura 6.2 Gráfica de barras de la Tabla 6.2



Reconocimiento de voz por distancia con ruido en entorno

Cuando se presentaron situaciones en las que varias personas hablaban al mismo tiempo, Alexa demostró una notable capacidad para distinguir comandos específicos del usuario. No obstante, en ambientes extremadamente ruidosos con múltiples fuentes de perturbación sonora simultánea, el rendimiento disminuyó ligeramente en un promedio de 6.96%, lo que podría afectar la precisión en la comprensión de los comandos. Asimismo, la presencia de música de fondo también impactó el rendimiento de Alexa. A pesar de ello, en general, la asistente fue capaz de adaptarse y reconocer comandos, aunque se notó un aumento leve en la tasa de errores en algunos casos con música a un volumen muy alto.

En cuanto al sistema de control de luces y proyector, no se encontraron mayores problemas en la ejecución de comandos. Al tratarse de un solo dispositivo, el control mediante Alexa se llevó a cabo sin contratiempos. Sin embargo, se identificó un potencial inconveniente con el sistema de persianas, dado que hay seis persianas vinculadas a Alexa. Esto podría llevar a confusiones en los comandos, en la siguiente tabla 6.3, 6.4, y 6.5, se van a proporcionar los comandos correctos para la ejecución de los dispositivos vinculados a Alexa.

Tabla 6.3: Comandos Alexa (persianas)

Acción	Comando
Abrir persianas	Alexa: “Abre persianas” “Abrir persianas” “Open persianas”
Cerrar persianas	Alexa: “Cerrar persianas” “Cierra persianas” “Close persianas”
Persianas a la mitad	Alexa: “Activa, escena 1”
Persiana 1	Alexa: “Cerrar Persiana 1” “Abrir Persiana 1”
Persiana 2	Alexa: “Cerrar Persiana 2” “Abrir Persiana 2”
Persiana 3	Alexa: “Cerrar Persiana 3” “Abrir Persiana 3”
Persiana 4	Alexa: “Cerrar Persiana 4” “Abrir Persiana 4”
Persiana 5	Alexa: “Cerrar Persiana 5” “Abrir Persiana 5”
Persiana 6	Alexa: “Cerrar Persiana 6” “Abrir Persiana 6”

Ejecución de comando, para persianas, mediante Alexa.



Tabla 6.4: Comandos Alexa (luces)

Acción	Comandos
Encender Luces	Alexa: “Enciende luces” “Turn on Lights”
Apagar Luces	Alexa: “Apaga luces” “Turn off lights”

Ejecución de comando, para las luces, mediante Alexa.

Tabla 6.5: Comandos Alexa (proyector)

Acción	Comandos
Encender proyector	Alexa: “Enciende proyector” “Turn on Proyector”
Apagar proyector	Alexa: “Apaga proyector” “Turn off proyector”

Ejecución de comando, para proyector, mediante Alexa.

Otro sistema susceptible de ocasionar fallos es el de los motores, ya que opera a través del internet de la universidad, que es un enlace con variaciones en su estabilidad. En algunas ocasiones, esto puede generar fallas de conexión y, como resultado, los motores no se activan correctamente. De igual manera para la ejecución de los motores son los siguientes comandos de la tabla 6.6.

Tabla 6.6: Comandos Alexa (proyector)

Acción	Comandos
Bajar pista	Alexa: “Activa bajar pista”
Subir pista	Alexa: “Activa subir pista”

EJECUCIÓN DE COMANDO, PARA LAS LUCES, mediante Alexa.

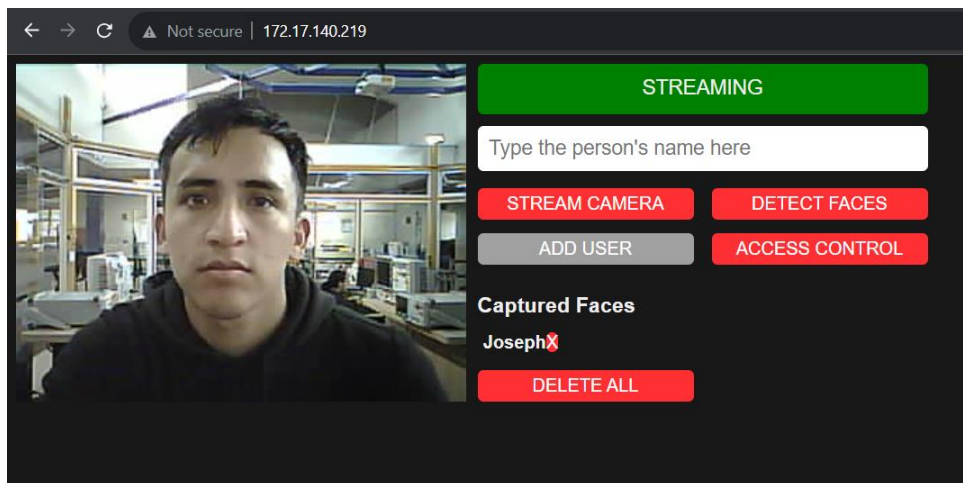
En conclusión, las pruebas realizadas mostraron que Alexa se desempeña de manera óptima en la mayoría de las situaciones, incluso en ambientes con ruido y música de fondo. Sin embargo, es importante considerar ciertas limitaciones en el control de dispositivos múltiples, como las persianas, para evitar confusiones en los comandos. Asimismo, se debe prestar atención a la estabilidad de la conexión a internet para garantizar el funcionamiento adecuado de los dispositivos vinculados al sistema de reconocimiento de voz. Estos hallazgos pueden ser útiles para futuras mejoras y

optimizaciones en la implementación de sistemas similares.

### 3.25 Reconocimiento Facial para abrir cerradura

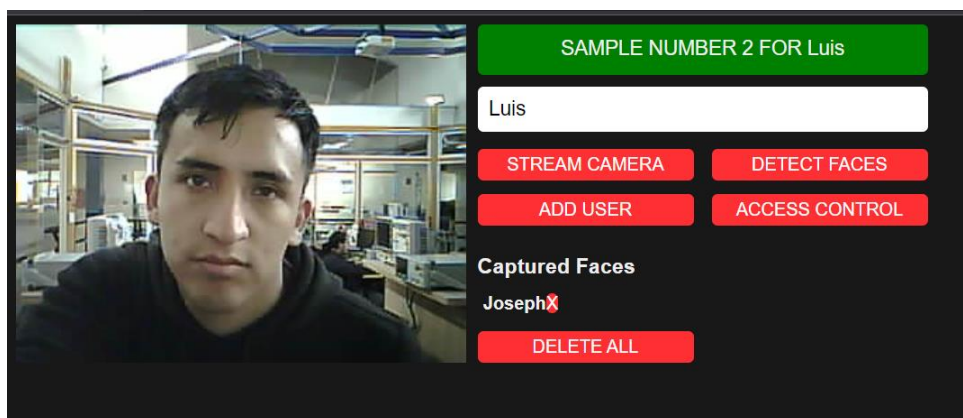
El sistema de reconocimiento facial para el desbloqueo de cerraduras fue sometido a pruebas en diversos momentos del día y en diferentes entornos, incluyendo escenarios con persianas abiertas, luces encendidas, luz solar directa y persianas cerradas, entre otros. En la mayoría de los casos, se puede constatar que el sistema de reconocimiento facial basado en la ESP32-CAM operó adecuadamente, a pesar de las perturbaciones mencionadas anteriormente. Los rostros previamente registrados en la ESP32-CAM fueron identificados correctamente, incluso ante variaciones pequeñas en iluminación y en la hora del día. A continuación, se muestra el funcionamiento general del código, desde reconocimiento hasta guardado de un nuevo usuario en las figuras 6.3, 6.4 y 6.5.

Figura 6.3: Interfaz ESP32-CAM



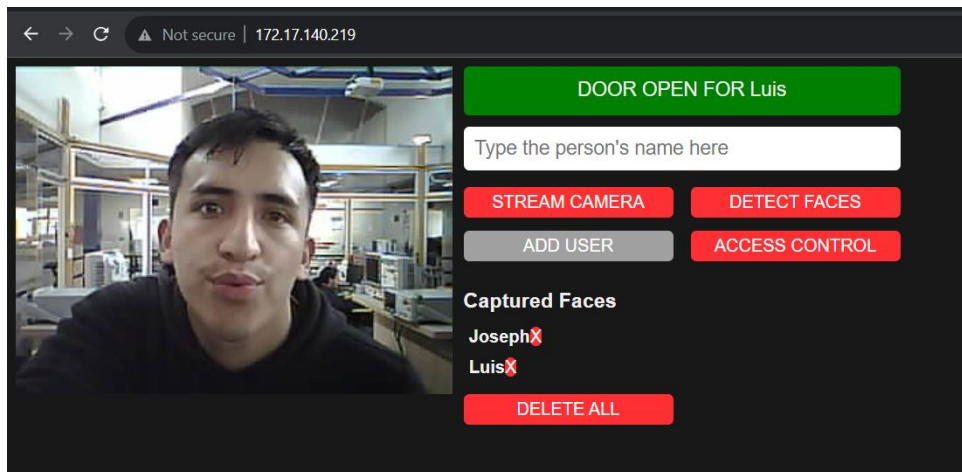
Cámara en vivo (streaming)

Figura 6.4: Interfaz ESP32-CAM



Captura rostro + asociación con nombre

Figura 6.5: Interfaz ESP32-CAM



Control de acceso abriendo cerradura

No obstante, se identificaron ciertas problemáticas asociadas con el sistema de reconocimiento facial. Una de ellas reside en la naturaleza no estática de la dirección IP proporcionada por la red universitaria. Esta característica implica la necesidad de conectar la ESP32-CAM a una computadora para obtener la dirección IP actual, a menos que el sistema se mantenga conectado de manera permanente. No obstante, esta última opción no es conveniente debido a que la ESP32-CAM tiende a calentarse fácilmente, lo que podría afectar su rendimiento y durabilidad.

Además, otra limitación del sistema se manifiesta en la precisión del reconocimiento facial cuando se producen cambios significativos en el rostro del individuo. Por ejemplo, al soltarse el cabello o aplicar maquillaje excesivo o al disminuir demasiado la iluminación del espacio, el sistema enfrenta dificultades para reconocer de manera correcta al usuario debido a limitaciones inherentes a la resolución y definición de la cámara utilizada.

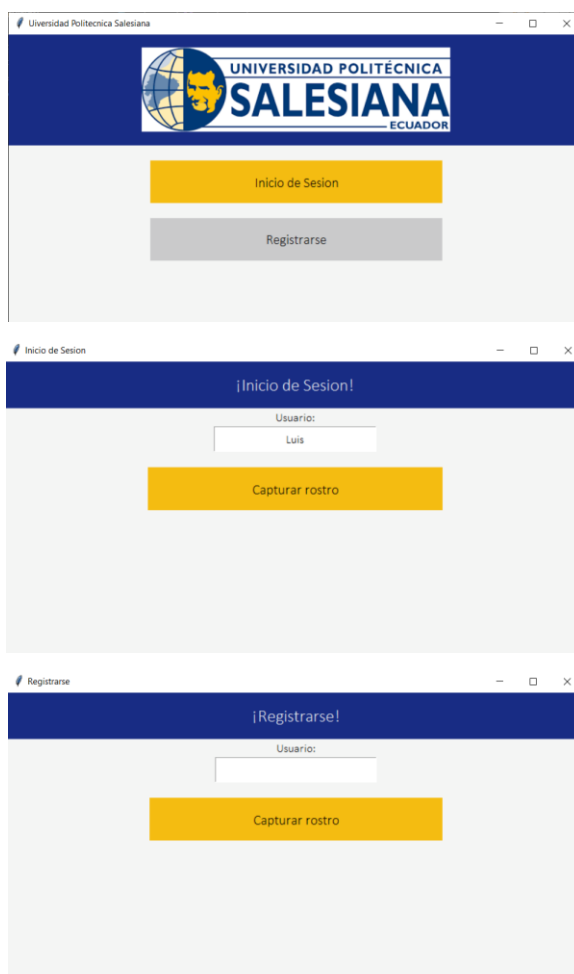
Estas observaciones resaltan la necesidad de considerar cuidadosamente la implementación de mecanismos para la gestión dinámica de la dirección IP y abordar las limitaciones asociadas con la precisión del reconocimiento facial en situaciones que involucren cambios notables en la apariencia facial. Tales consideraciones resultan fundamentales para garantizar el óptimo funcionamiento y confiabilidad del sistema de reconocimiento facial propuesto en contextos reales de uso y podrían constituir áreas de mejora para futuros desarrollos tecnológicos.

### 3.26 Visión artificial (Registro Estudiantes)

Se llevaron a cabo pruebas de visión artificial en el sistema de registro de estudiantes, enfocándose en el análisis de la cámara y sus posibles influencias en el proceso, especialmente en relación con la iluminación. Además, se llevó a cabo una verificación exhaustiva de la integridad y el uso adecuado de la base de datos, la cual permite el almacenamiento de múltiples usuarios. Estas acciones condujeron a un rendimiento óptimo del sistema de registro de estudiantes mediante visión artificial, evitando cualquier inconveniente que pudiera surgir.

Por otro lado, la interfaz realizada se diseñó acorde a la “Universidad Politécnica Salesiana”, con botones animados, para un contacto amigable hacia el usuario, se puede visualizar en la Figura 6.6.

Figura 6.6: Interfaz Visión Artificial



Interfaz de inicio de sesión y registro de estudiantes.

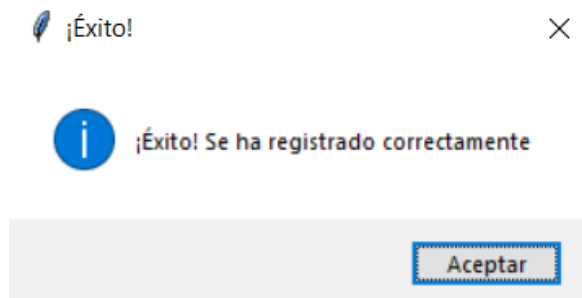
Al momento de ser registrado correctamente o en caso contrario, si no se registra correctamente, un mensaje emergente da el aviso, para que por último se pueda revisar en la base de datos, como se puede visualizar en la Figura 6.7, 6.8 y 6.9.

Figura 6.7: Error de acceso



No se encuentra registrado un usuario con el nombre dado.

Figura 6.8: Registro correcto



El usuario registrado, se encuentra habilitado.

Figura 6.9: Tabla base de datos

	idUser	name	photo
▶	4	Luis	BLOB
•	NULL	NULL	NULL

Comprobación del registro.

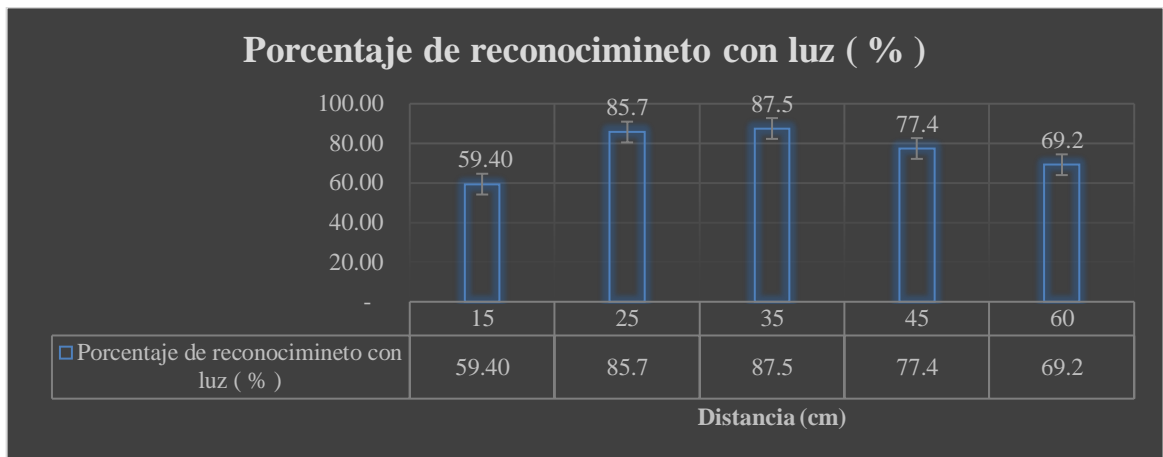
Los sistemas de reconocimiento facial mostraron los siguientes comportamientos (tablas 6.7 y 6.8) acorde a la distancia de lectura de rostro:

Tabla 6.7: Pruebas de reconocimiento Facial con luz

Reconocimiento facial - cerradura eléctrica											
Pruebas Apertura de cerradura en después de reconocer rostro.											Porcentaje de reconocimiento con luz (%)
Distancia de reconocimiento facial (cm)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
15	60	63	58	40	58	61	66	71	62	55	59,40
25	95	80	87	85	89	82	83	85	80	91	85,7
35	83	90	88	88	90	83	89	97	81	86	87,5
45	88	80	76	73	78	70	70	78	80	81	77,4
60	68	70	64	69	63	66	66	67	69	90	69,2

Pruebas Apertura de cerradura en 100 (ms) después de reconocer rostro.

Figura 6.10: Gráfica de barras de la Tabla 6.7



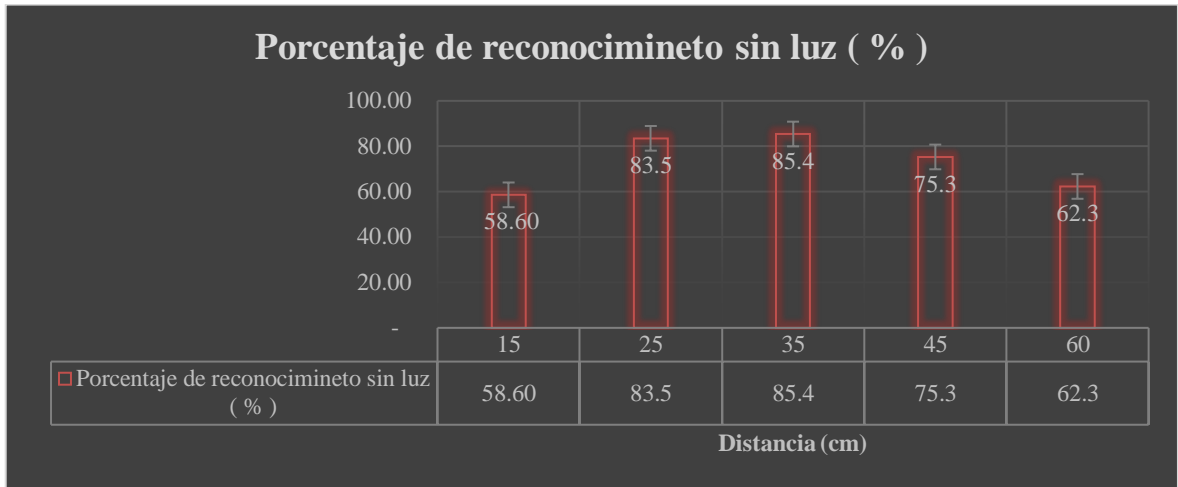
Reconocimiento de rostro acorde a la distancia de imagen con luz

Tabla 6.8: Pruebas de reconocimiento Facial sin Luz

Reconocimiento facial - cerradura eléctrica											
Pruebas Apertura de cerradura en después de reconocer rostro.											Porcentaje de reconocimiento sin luz (%)
Distancia de reconocimiento facial (cm)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
15	60	62	50	43	66	61	60	61	67	56	58,60
25	90	80	85	86	81	88	81	80	80	84	83,5
35	80	91	84	85	89	82	81	92	81	89	85,4
45	78	70	76	73	76	71	70	78	81	80	75,3
60	50	62	64	65	66	61	60	63	62	70	62,3

Pruebas Apertura de cerradura en 100 (ms) después de reconocer rostro.

Figura 6.11: Gráfica de barras de la Tabla 6.8



Reconocimiento de rostro acorde a la distancia de imagen sin luz

Como se puede ver en las gráficas de datos de las figuras 6.11 y 6.12 anteriores el reconocimiento de rostro tiene una mejor captura de imagen cuando hay luz en el ambiente, de lo contrario su eficiencia se ve un tanto reducida.

## CAPÍTULO 7. Presupuesto

Tabla 7.1 Presupuestos

Producto	Cantidad	Valor
ESP32 CAM con modulo	2	\$ 36,00
Reles / 24V	2	\$ 15,00
Cables Macho-Macho	40	\$ 3,00
Cables Macho-Hembra	40	\$ 3,00
Fuente de 30V DC	1	\$ 9,00
Porta pilas AAA	1	\$ 1,50
Diodo 1N4007	2	\$ 2,00
Tip122	2	\$ 1,50
Pulsador	1	\$ 0,65
Resistencias 10k	4	\$ 0,40
Regulador de voltaje	2	\$ 1,50
Condensador	2	\$ 0,50
Cerradura eléctrica con solenoide 12V	1	\$ 8,00
Fuente 5V	1	\$ 5,00
Fuente 12V 1,5A	2	\$ 4,00
Costo de diseño + mano de obra	1	\$ 50,00
Acople Polea Motores	2	\$ 40,00
Polea	2	\$ 16,00
Cable trenzado	8m	\$ 4,00
ESP32	1	\$ 12,00
Cámara	1	\$ 12,00
Encoder	2	\$ 3,00
Micro SD card	1	\$ 8,00
Tornillos, tuercas, cables, canaletas	varios	\$ 10,00
Fuente 12V 14A	1	\$ 20,15
Total:		\$ 250,20

Tabla de gastos generales



## **CAPÍTULO 8. Conclusiones**

- Se creó un plan de trabajo detallado y adecuado que permitió identificar los dispositivos necesarios para implementar el sistema domótico en el laboratorio de robótica. Para asegurar la elección adecuada de los dispositivos y garantizar una implementación efectiva, se necesitó una observación cuidadosa del espacio mediante el análisis de las necesidades específicas como la iluminación, seguridad y eficiencia de tareas del laboratorio de robótica de la UPS para una óptima automatización.
- Mediante el análisis de reconocimiento facial, se llevaron a cabo experimentos en distintos contextos ambientales, evidenciándose que, en un entorno de baja iluminación, la tasa de acierto del reconocimiento facial es del 73.02%, mientras que en un entorno bien iluminado alcanza el 75.84%. Este comportamiento indica una respuesta satisfactoria del sistema, con una mínima variación en los valores registrados independientemente de las condiciones lumínicas. Tal consistencia se atribuye a la elevada resolución proporcionada por la cámara implementada. Este avance mejoró el proceso de control de entrada al laboratorio, facilitó el acceso de los estudiantes y proporcionó mayor seguridad al identificar a cada persona de manera precisa.
- La implementación del sistema de control mediante audio permitió automatizar el control de varios parámetros importantes en el laboratorio de robótica. Los porcentajes de activación de comando de voz han evidenciado una notable capacidad para ajustar aspectos esenciales como la iluminación, los proyectores, los motores y las persianas de manera más rápida y sencilla, mejorando la eficiencia en general y la comodidad del entorno de trabajo. A su vez se pudo observar una muy buena respuesta del sistema de voz incluso en presencia de las perturbaciones de sonido; teniendo una respuesta del 92,42% sin presencia de ruido y del 83,74% en presencia de este.
- Las pruebas de funcionamiento realizadas en el laboratorio fueron Esenciales para validar la implementación propuesta. Durante estas pruebas, el sistema domótico funcionó correctamente y cumplió con los objetivos, lo que demuestra su eficacia y viabilidad en el entorno real del laboratorio de robótica.

## CAPÍTULO 9. Referencias bibliográficas

- Amazon. (2022). What is Alexa?. Amazon. <https://developer.amazon.com/es-ES/alexa>
- Argos. (2019, December 28). Convertidores DC/DC – buck. Nómadas Electrónicos. <https://nomadaselectronicos.wordpress.com/2015/04/12/convertidores-dcdc-buck/>
- Canalejo, L. (2021, August 12). Qué es un dimmer y cómo funciona. Iluminación, ventilación de diseño y bombillas. <https://faro.es/es/blog/que-es-un-dimmer-y-como-funciona/>
- Carmenate, J. G. (2022, March 7). Esp32 WIFI + bluetooth en UN solo lugar. Programar fácil con Arduino. <https://programarfácil.com/esp8266/esp32/>
- CORP., O. M. (2022). Motor sizing tools. Oriental Motor U.S.A. Corp. <https://www.orientalmotor.com/motor-sizing/index.html>
- Desarrollo web. (2022, August 7). ¿Qué es websocket? IONOS Digital Guide. Retrieved May 2, 2023, from <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-websocket/>
- Fernández, Y. (2022, septiembre 23). Qué es Arduino, cómo funciona y qué puedes hacer con uno. Xataka.com; Xataka Basics. <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- Fernández, Y. (2023, January 12). Qué Es Alexa, Qué puedes hacer con él y qué dispositivos son compatibles. Xataka. <https://www.xataka.com/basics/que-alexa-que-puedes-hacer-que-dispositivos-compatibles>
- Inc., Amazon. com. (2022). Dispositivos Amazon Echo y Alexa. Amazon.com. <https://www.amazon.com/-/es/Amazon-Echo/b?node=9818047011>
- Kaspersky. (2022, february 9). What is Facial Recognition – Definition and Explanation. Wwww.kaspersky.com. <https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition>
- Kaspersky. (2023, April 19). What is biometrics? how is it used in security? www.kaspersky.com. <https://www.kaspersky.com/resource-center/definitions/biometrics>
- Lahtela, M., & Kaplan, P. (2023). ES. Amazon. <https://aws.amazon.com/es/what-is/python/>
- Matplotlib. (2023, March 4). PyPI. <https://pypi.org/project/matplotlib/>
- Morales, R. R. (2023, August 1). ESP32 38 pines ESP WROOM 32. UNIT Electronics. <https://uelectronics.com/producto/esp32-38-pines-esp-wroom-32/>
- Netikon.gr. (2023). Camera module based on ESP32. grobotronics.com. <https://grobotronics.com/camera-module-based-on-esp32.html?sl=en>
- Pascual, C. (2022, January 19). ESP32 Cam Introducción y Primeros Pasos. Programar fácil con Arduino. <https://programarfácil.com/esp32/esp32-cam/>
- Python Software Foundation. (abr 17, 2023) Tkinter - interface de python para TCL/TK. Python documentation. <https://docs.python.org/es/3/library/tkinter.html>
- RecFaces. (2022, October 13). ¿Qué es el reconocimiento de voz y cómo funciona? RecFaces. Retrieved May 2, 2023, from <https://recfaces.com/es/articles/reconocimiento-voz>.
- Rodriguez H. (2021) ¿Qué es OpenCV y para qué sirve? . crehana. <https://www.crehana.com/blog/transformacion-digital/que-es-opencv/>
- Sarachu, E. (2023, marzo 8). ¿Qué es la domótica? ¿Cómo funciona una instalación domótica? Eficiencia Energetica; e-Ficiencia: Revista sobre Eficiencia Energética. <https://e-ficiencia.com/domotica-que-es-y-como-funciona/>
- SEAS Estudios Superiores Abiertos. (2019, August 22). El Relé: Para qué es, para qué sirve y qué tipos existen: Blog seas. Blog de SEAS . <https://www.seas.es/blog/automatizacion/el-rele-para-que-es-para-que-sirve-y-que-tipos-existen/>

Sistemas Domóticos (2020). Sistemas domóticos existentes, tipos y estándares.  
Domótica Sistemas.  
[https://domoticasistemas.com/tienda/tutoriales/1\\_sistemas-existentes-tipos-y-estandares.html](https://domoticasistemas.com/tienda/tutoriales/1_sistemas-existentes-tipos-y-estandares.html)

Solutions, N. (2020b). Nexxt Solutions.  
<https://www.nexxtsolutions.com/es/conectividad/interna-productos/NHE-D100-es/>

Tensorflow. (22 mar 2023). tensorflow 2.12. PyPI.  
<https://pypi.org/project/tensorflow/>

## CAPÍTULO 10. ANEXOS

Tabla 10.1 código Arduino reconocimiento facial

```
#include <ArduinoWebsockets.h>
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "camera_index.h"
#include "Arduino.h"
#include "fd_forward.h"
#include "fr_forward.h"
#include "fr_flash.h"

const char* ssid = "UPS_TESIS";
const char* password = "Wi7i@2021";

#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7

#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"

using namespace websockets;
WebsocketsServer socket_server;

camera_fb_t * fb = NULL;

long current_millis;
long last_detected_millis = 0;

#define relay_pin 2 // pin 12 can also be used
```

```
unsigned long door_opened_millis = 0;
long interval = 5000;      // open lock for ... milliseconds
bool face_recognised = false;

void app_facenet_main();
void app_httpserver_init();

typedef struct
{
    uint8_t *image;
    box_array_t *net_boxes;
    dl_matrix3d_t *face_id;
} http_img_process_result;

static inline mtmn_config_t app_mtmn_config()
{
    mtmn_config_t mtmn_config = {0};
    mtmn_config.type = FAST;
    mtmn_config.min_face = 80;
    mtmn_config.pyramid = 0.707;
    mtmn_config.pyramid_times = 4;
    mtmn_config.p_threshold.score = 0.6;
    mtmn_config.p_threshold.nms = 0.7;
    mtmn_config.p_threshold.candidate_number = 20;
    mtmn_config.r_threshold.score = 0.7;
    mtmn_config.r_threshold.nms = 0.7;
    mtmn_config.r_threshold.candidate_number = 10;
    mtmn_config.o_threshold.score = 0.7;
    mtmn_config.o_threshold.nms = 0.7;
    mtmn_config.o_threshold.candidate_number = 1;
    return mtmn_config;
}
```

```
mtmn_config_t mtmn_config = app_mtmn_config();
```

```
face_id_name_list st_face_list;
```

```
static dl_matrix3du_t *aligned_face = NULL;
```

```
httpd_handle_t camera_httpd = NULL;
```

```
typedef enum
```

```
{  
    START_STREAM,  
    START_DETECT,  
    SHOW_FACES,  
    START_RECOGNITION,  
    START_ENROLL,  
    ENROLL_COMPLETE,  
    DELETE_ALL,  
} en_fsm_state;
```

```
en_fsm_state g_state;
```

```
typedef struct
```

```
{  
    char enroll_name[ENROLL_NAME_LEN];  
} httpd_resp_value;
```

```
httpd_resp_value st_name;
```

```
void setup() {
```

```
    Serial.begin(115200);  
    Serial.setDebugOutput(true);  
    Serial.println();
```

```
    digitalWrite(relay_pin, LOW);  
    pinMode(relay_pin, OUTPUT);
```

```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
//init with high specs to pre-allocate larger buffers
if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}
```

```
#if defined(CAMERA_MODEL_ESP_EYE)

  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}

sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

app_httpsver_init();
app_facenet_main();
socket_server.listen(82);

Serial.print("Camera Ready! Use 'http://'");
```



```

Serial.print(WiFi.localIP());
Serial.println(" to connect");
}

static esp_err_t index_handler(httpd_req_t *req) {
    httpd_resp_set_type(req, "text/html");
    httpd_resp_set_hdr(req, "Content-Encoding", "gzip");
    return httpd_resp_send(req, (const char *)index_ov2640_html_gz,
index_ov2640_html_gz_len);
}

httpd_uri_t index_uri = {
    .uri = "/",
    .method = HTTP_GET,
    .handler = index_handler,
    .user_ctx = NULL
};

void app_httpserver_init ()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    if (httpd_start(&camera_httpd, &config) == ESP_OK)
        Serial.println("httpd_start");
    {
        httpd_register_uri_handler(camera_httpd, &index_uri);
    }
}

void app_facenet_main()
{
    face_id_name_init(&st_face_list, FACE_ID_SAVE_NUMBER,
ENROLL_CONFIRM_TIMES);
    aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
}

```

```

    read_face_id_from_flash_with_name(&st_face_list);
}

static inline int do_enrollment(face_id_name_list *face_list, dl_matrix3d_t
*new_id)
{
    ESP_LOGD(TAG, "START ENROLLING");
    int left_sample_face = enroll_face_id_to_flash_with_name(face_list, new_id,
st_name.enroll_name);
    ESP_LOGD(TAG, "Face ID %s Enrollment: Sample %d",
        st_name.enroll_name,
        ENROLL_CONFIRM_TIMES - left_sample_face);
    return left_sample_face;
}

static esp_err_t send_face_list(WebsocketsClient &client)
{
    client.send("delete_faces"); // tell browser to delete all faces
    face_id_node *head = st_face_list.head;
    char add_face[64];
    for (int i = 0; i < st_face_list.count; i++) // loop current faces
    {
        sprintf(add_face, "listface:%s", head->id_name);
        client.send(add_face); //send face to browser
        head = head->next;
    }
}

static esp_err_t delete_all_faces(WebsocketsClient &client)
{
    delete_face_all_in_flash_with_name(&st_face_list);
    client.send("delete_faces");
}

```

```

void handle_message(WebsocketsClient &client, WebsocketsMessage msg)
{
    if (msg.data() == "stream") {
        g_state = START_STREAM;
        client.send("STREAMING");
    }
    if (msg.data() == "detect") {
        g_state = START_DETECT;
        client.send("DETECTING");
    }
    if (msg.data().substring(0, 8) == "capture:") {
        g_state = START_ENROLL;
        char person[FACE_ID_SAVE_NUMBER * ENROLL_NAME_LEN] = {0,};
        msg.data().substring(8).toCharArray(person, sizeof(person));
        memcpy(st_name.enroll_name, person, strlen(person) + 1);
        client.send("CAPTURING");
    }
    if (msg.data() == "recognise") {
        g_state = START_RECOGNITION;
        client.send("RECOGNISING");
    }
    if (msg.data().substring(0, 7) == "remove:") {
        char person[ENROLL_NAME_LEN * FACE_ID_SAVE_NUMBER];
        msg.data().substring(7).toCharArray(person, sizeof(person));
        delete_face_id_in_flash_with_name(&st_face_list, person);
        send_face_list(client); // reset faces in the browser
    }
    if (msg.data() == "delete_all") {
        delete_all_faces(client);
    }
}

void open_door(WebsocketsClient &client) {

```

```

if (digitalRead(relay_pin) == LOW) {
    digitalWrite(relay_pin, HIGH); //close (energise) relay so door unlocks
    Serial.println("Door Unlocked");
    client.send("door_open");
    door_opened_millis = millis(); // time relay closed and door opened
}
}

void loop() {
    auto client = socket_server.accept();
    client.onMessage(handle_message);
    dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, 320, 240, 3);
    http_img_process_result out_res = {0};
    out_res.image = image_matrix->item;

    send_face_list(client);
    client.send("STREAMING");

    while (client.available()) {
        client.poll();

        if (millis() - interval > door_opened_millis) { // current time - face recognised time
        > 5 secs
            digitalWrite(relay_pin, LOW); //open relay
        }

        fb = esp_camera_fb_get();

        if (g_state == START_DETECT || g_state == START_ENROLL || g_state ==
START_RECOGNITION)
        {
            out_res.net_boxes = NULL;
            out_res.face_id = NULL;

```

```

fmt2rgb888(fb->buf, fb->len, fb->format, out_res.image);

out_res.net_boxes = face_detect(image_matrix, &mtmn_config);

if (out_res.net_boxes)
{
    if (align_face(out_res.net_boxes, image_matrix, aligned_face) == ESP_OK)
    {

        out_res.face_id = get_face_id(aligned_face);
        last_detected_millis = millis();
        if (g_state == START_DETECT) {
            client.send("FACE DETECTED");
        }

        if (g_state == START_ENROLL)
        {
            int left_sample_face = do_enrollment(&st_face_list, out_res.face_id);
            char enrolling_message[64];
            sprintf(enrolling_message, "SAMPLE NUMBER %d FOR %s",
ENROLL_CONFIRM_TIMES - left_sample_face, st_name.enroll_name);
            client.send(enrolling_message);
            if (left_sample_face == 0)
            {
                ESP_LOGI(TAG, "Enrolled Face ID: %s", st_face_list.tail->id_name);
                g_state = START_STREAM;
                char captured_message[64];
                sprintf(captured_message, "FACE CAPTURED FOR %s", st_face_list.tail-
>id_name);
                client.send(captured_message);
                send_face_list(client);
            }
        }
    }
}

```

```

    }
}

if (g_state == START_RECOGNITION && (st_face_list.count > 0))
{
    face_id_node *f = recognize_face_with_name(&st_face_list,
out_res.face_id);
    if (f)
    {
        char recognised_message[64];
        sprintf(recognised_message, "DOOR OPEN FOR %s", f->id_name);
        open_door(client);
        client.send(recognised_message);
    }
    else
    {
        client.send("FACE NOT RECOGNISED");
    }
}
dl_matrix3d_free(out_res.face_id);
}
}
else
{
    if (g_state != START_DETECT) {
        client.send("NO FACE DETECTED");
    }
}
if (g_state == START_DETECT && millis() - last_detected_millis > 500) { //
Detecting but no face detected
    client.send("DETECTING");
}
}

```

```

    client.sendBinary((const char *)fb->buf, fb->len);

    esp_camera_fb_return(fb);

    fb = NULL;
}
}

```

Código para abrir cerradura con ESP32-CAM

Tabla 10.2 Código Arduino Motores

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>

#define Subirpista 17
#define Bajarpista 16
#define Spista 4
#define Bpista 2

//int contador = 0;

// Update these with values suitable for your network.

const char* ssid = "UPS_TESIS";
const char* password = "Wi7i@2021";
//const char* ssid = "LUIS_ALVARADO";
//const char* password = "LEAC_2022";
const char* mqtt_server = "broker.hivemq.com";

String _topic;
String _payload;

WiFiClient espClient;
PubSubClient client(espClient);

```

```
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;

const int encoderPin = 14; // Pin del encoder conectado al GPIO 14
const int encoderPin2 = 33;

volatile int encoderPos = 0; // Variable para almacenar la posición del encoder
volatile int lastEncoderState = 0; // Variable para almacenar el estado anterior del encoder
unsigned long lastDebounceTime = 0; // Variable para almacenar el último tiempo de
antirrebote
unsigned long debounceDelay = 8;

volatile int encoderPos1 = 0; // Variable para almacenar la posición del encoder
volatile int lastEncoderState1 = 0; // Variable para almacenar el estado anterior del encoder
unsigned long lastDebounceTime1 = 0; // Variable para almacenar el último tiempo de
antirrebote
unsigned long debounceDelay1 = 8;

void IRAM_ATTR handleEncoderInterrupt() {
  unsigned long currentMillis = millis();
  if (currentMillis - lastDebounceTime > debounceDelay) {
    int currentEncoderState = digitalRead(encoderPin);
    //if (currentEncoderState != lastEncoderState) {
    if (currentEncoderState == HIGH) {
      encoderPos++;
      Serial.println(encoderPos);
    }
    //}
    lastEncoderState = currentEncoderState;
    lastDebounceTime = currentMillis;
  }
}
```



```

}
}

void IRAM_ATTR handleEncoderInterrupt1() {
  unsigned long currentMillis1 = millis();
  if (currentMillis1 - lastDebounceTime1 > debounceDelay1) {
    int currentEncoderState1 = digitalRead(encoderPin2);
    //if (currentEncoderState != lastEncoderState) {
    if (currentEncoderState1 == HIGH) {
      encoderPos1++;
      Serial.println(encoderPos1);
    }
    //}
    lastEncoderState1 = currentEncoderState1;
    lastDebounceTime1 = currentMillis1;
  }
}
}

```

```

void setup_wifi() {

  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);

```

```

Serial.print(".");
}

randomSeed(micros());

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

}

void callback(char* topic, byte* payload, unsigned int length) {
  String message;
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    message +=(char)payload[i];
  }

  _topic = topic;

  // Activacion y desactivacion .....
  if (_topic=="Subirpista"){
    Serial.println("Subiendo Pista");
    encoderPos=0;
    if (encoderPos <= 50){
      digitalWrite(Subirpista, HIGH);
    }
  }else{ digitalWrite(Subirpista, LOW); }
}

```

```

if (_topic=="Subirpista"){
Serial.println("Subiendo Pista");
encoderPos1=0;
  if (encoderPos1 <= 50){
    digitalWrite(Spista, HIGH);
  }
}else{digitalWrite(Spista, LOW); }

////////////////////////////////////

  if (_topic=="Bajarpista"){
    encoderPos=0;
Serial.println("Bajando Pista");
  if (encoderPos <= 50){
    digitalWrite(Bajarpista, HIGH);
  }
}else{digitalWrite(Bajarpista, LOW); }

if (_topic=="Bajarpista"){
  encoderPos1=0;
Serial.println("Bajando Pista");
  if (encoderPos1 <= 50){
    digitalWrite(Bpista, HIGH);
  }
}else{digitalWrite(Bpista, LOW); }
////////////////////////////////////
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID

```

```

String clientId = "ESP8266Client-";
clientId += String(random(0xffff), HEX);
// Attempt to connect
if (client.connect(clientId.c_str())) {
    Serial.println("connected");
    // Conexion con los nodos nombre exactos.....
    client.subscribe("Bajarpista");
    client.subscribe("Subirpista");

} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
}
}

void setup() {
    pinMode(Bajarpista, OUTPUT);
    pinMode(Subirpista, OUTPUT);
    pinMode(Bpista, OUTPUT);
    pinMode(Spista, OUTPUT);

    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);

    pinMode(encoderPin, INPUT_PULLUP); // Habilitar resistencia pull-up para el pin del
encoder
    pinMode(encoderPin2, INPUT_PULLUP);

```

```

attachInterrupt(digitalPinToInterrupt(encoderPin), handleEncoderInterrupt, CHANGE);
attachInterrupt(digitalPinToInterrupt(encoderPin2), handleEncoderInterrupt1,
CHANGE);
//
//  pinMode(encoderPin, INPUT_PULLUP); // Habilitar resistencia pull-up para el pin del
encoder
//
// attachInterrupt(digitalPinToInterrupt(encoderPin), handleEncoderInterrupt1,
CHANGE);
}

void loop() {

if (!client.connected()) {
  reconnect();
}
client.loop();
//cont();
if (encoderPos >= 50){
  digitalWrite(Subirpista, LOW);
  digitalWrite(Bajarpista, LOW);
  encoderPos = 0;
}

if (encoderPos1 >= 50){
  digitalWrite(Spista, LOW);
  digitalWrite(Bpista, LOW);
  encoderPos1 = 0;
}
}

```

Código para control de motores mediante encoders

Tabla 10.3 Cálculos del motor.

<b>Table shape and dimensions</b>	
Table type	= <u>Rectangular Table</u>
Diameter	D = _____ [mm]
Width	A = <u>2000</u> [mm]
Depth	B = <u>2000</u> [mm]
Mass	m = <u>26</u> [kg]
Thickness	t = _____ [mm]
Material	_____ ρ = _____ [kg/m <sup>3</sup> ]

<b>Drive shaft dimension</b>	
Shaft diameter	D <sub>2</sub> = <u>3</u> [mm]
Shaft mass	m <sub>2</sub> = _____ [kg]
Shaft length	L = <u>6</u> [mm]
Shaft material	<u>Aluminum</u> ρ <sub>2</sub> = <u>2800</u> [kg/m <sup>3</sup> ]

<b>Table support</b>	
Friction coefficient between the table and the supporting mechanism	μ = <u>0.6</u>
Distance from the table center to the supporting mechanism	l = <u>50</u> [mm]
System efficiency	η = <u>80</u> %

## Transmission belt and pulleys or gears

Primary pulley (gear)

pitch circle diameter (PCD)  $D_{p1} = \underline{25}$  [mm]

mass  $m_{p1} = \underline{\hspace{2cm}}$  [kg]

thickness  $L_{p1} = \underline{3}$  [mm]

material  $\rho_{p1} = \underline{2800}$  [kg/m<sup>3</sup>]

Aluminum

## Mechanism Condition

Mechanism Condition Vertical operation

## Other requirement(s)

Is it necessary to hold the load even after the power supply is turned off? → NO

Is it necessary to hold the load after the motor is stopped, but not necessary to hold after the power supply is turned off? → NO

## Operating conditions

Fixed speed operation      Operating speed       $V_1 = \underline{100}$  [r/min]

Acceleration / deceleration time       $t_1 = \underline{60}$  [s]

## Stopping accuracy

Stopping accuracy       $\pm \Delta\theta = \underline{200}$  °

Tabla 10.4 database.py

```

import mysql.connector as db
import json

with open('keys.json') as json_file:
    keys = json.load(json_file)

def convertToBinaryData(filename):

    try:
        with open(filename, 'rb') as file:
            binaryData = file.read()
        return binaryData
    except:
        return 0

def write_file(data, path):

    with open(path, 'wb') as file:
        file.write(data)

def registerUser(name, photo):
    id = 0
    inserted = 0

    try:
        con = db.connect(host=keys["host"], user=keys["user"],
password=keys["password"], database=keys["database"])
        cursor = con.cursor()
        sql = "INSERT INTO `user`(name, photo) VALUES (%s,%s)"
        pic = convertToBinaryData(photo)

        if pic:
            cursor.execute(sql, (name, pic))
            con.commit()
            inserted = cursor.rowcount
            id = cursor.lastrowid
    except db.Error as e:
        print(f"Failed inserting image: {e}")
    finally:
        if con.is_connected():
            cursor.close()
            con.close()
    return {"id": id, "affected":inserted}

def getUser(name, path):
    id = 0
    rows = 0

```



```

try:
    con = db.connect(host=keys["host"], user=keys["user"],
password=keys["password"], database=keys["database"])
    cursor = con.cursor()
    sql = "SELECT * FROM `user` WHERE name = %s"

    cursor.execute(sql, (name,))
    records = cursor.fetchall()

    for row in records:
        id = row[0]
        write_file(row[2], path)
    rows = len(records)
except db.Error as e:
    print(f"Failed to read image: {e}")
finally:
    if con.is_connected():
        cursor.close()
        con.close()
return {"id": id, "affected": rows}

```

Base de datos Código 1

Tabla 10.5 código.py

```

import database as db
from matplotlib import pyplot as plt
from tkinter import *
from tkinter import messagebox as msg
import os
import cv2
from mtcnn.mtcnn import MTCNN

path = "C:/Users/Luis Alvarado/OneDrive/Escritorio/TESIS/BDDvision/"
txt_login = "Inicio de Sesion"
txt_register = "Registrarse"

## colores disen0

color_white = "#f4f5f4"    ## fondo encabezado
color_gris = "#cacacb"    ## Fondo
color_black = "#101010"
color_blue = "#182c84"
color_black_btn = "#202020"
color_btn = "#e9ebf0"      ## para botones
color_background = "#151515"
color_yellow = "#f4bc11"

## Ajuste pantalla y texto

```

```

font_label = "Calibri Light"
size_screen = "800x400"
color_success = "\033[1;32;40m"
color_error = "\033[1;31;40m"
color_normal = "\033[0;37;40m"
res_bd = {"id": 0, "affected": 0}

# Inicio Codigo

def getEnter(screen):

    Label(screen, text="", bg=color_white).pack() ## Espaciado

def printAndShow(screen, text, flag):

    if flag:
        print(color_success + text + color_normal)
        screen.destroy()
        msg.showinfo(message=text, title="¡Éxito!")
    else:
        print(color_error + text + color_normal)
        Label(screen, text=text, fg="red", bg=color_background,
font=(font_label, 12)).pack()

def configure_screen(screen, text):

    screen.title(text)
    screen.geometry(size_screen)
    screen.configure(bg=color_white) ##fondo segundas pantallas
    Label(screen, text=f"¡{text}!", fg=color_btn, bg=color_blue,
font=(font_label, 18), width="500", height="2").pack() ## fonde titulos
segundas pantallas

def credentials(screen, var, flag):

    Label(screen, text="Usuario:", fg=color_black, bg=color_white,
font=(font_label, 12)).pack()
    entry = Entry(screen, textvariable=var, justify=CENTER,
font=(font_label, 12))
    entry.focus_force()
    entry.pack(side=TOP, ipadx=30, ipady=6)

    getEnter(screen)
    if flag:
        Button(screen, text="Capturar rostro", fg=color_black,
bg=color_gris, activebackground=color_yellow, borderwidth=0,
font=(font_label, 14), height="2", width="40", command=login_capture).pack()
    else:

```

```

        Button(screen, text="Capturar rostro", fg=color_black,
bg=color_gris, activebackground=color_yellow, borderwidth=0,
font=(font_label, 14), height="2", width="40",
command=register_capture).pack()
        return entry

def face(img, faces):
    data = plt.imread(img)
    for i in range(len(faces)):
        x1, y1, ancho, alto = faces[i]["box"]
        x2, y2 = x1 + ancho, y1 + alto
        plt.subplot(1,len(faces), i + 1)
        plt.axis("off")
        face = cv2.resize(data[y1:y2, x1:x2],(150,200),
interpolation=cv2.INTER_CUBIC)
        cv2.imwrite(img, face)
        plt.imshow(data[y1:y2, x1:x2])

# Registro

def register_face_db(img):
    name_user = img.replace(".jpg","").replace(".png","")
    res_bd = db.registerUser(name_user, path + img)

    getEnter(screen1)
    if(res_bd["affected"]):
        printAndShow(screen1, "¡Éxito! Se ha registrado correctamente", 1)
    else:
        printAndShow(screen1, "¡Error! No se ha registrado correctamente",
0)
    os.remove(img)

def register_capture():
    cap = cv2.VideoCapture(0)
    user_reg_img = user1.get()
    img = f"{user_reg_img}.jpg"

    while True:
        ret, frame = cap.read()
        cv2.imshow("Registro Facial", frame)
        if cv2.waitKey(1) == 27:
            break

    cv2.imwrite(img, frame)
    cap.release()
    cv2.destroyAllWindows()

    user_entry1.delete(0, END)

```

```

pixels = plt.imread(img)
faces = MTCNN().detect_faces(pixels)
face(img, faces)
register_face_db(img)

def register():
    global user1
    global user_entry1
    global screen1

    screen1 = Toplevel(root)
    user1 = StringVar()

    configure_screen(screen1, txt_register)
    user_entry1 = credentials(screen1, user1, 0)

# Login

def compatibility(img1, img2):
    orb = cv2.ORB_create()

    kpa, dac1 = orb.detectAndCompute(img1, None)
    kpa, dac2 = orb.detectAndCompute(img2, None)

    comp = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)

    matches = comp.match(dac1, dac2)

    similar = [x for x in matches if x.distance < 70]
    if len(matches) == 0:
        return 0
    return len(similar)/len(matches)

def login_capture():
    cap = cv2.VideoCapture(0)
    user_login = user2.get()
    img = f"{user_login}_login.jpg"
    img_user = f"{user_login}.jpg"

    while True:
        ret, frame = cap.read()
        cv2.imshow("Login Facial", frame)
        if cv2.waitKey(1) == 27:
            break

    cv2.imwrite(img, frame)
    cap.release()
    cv2.destroyAllWindows()

```

```

user_entry2.delete(0, END)

pixels = plt.imread(img)
faces = MTCNN().detect_faces(pixels)

face(img, faces)
getEnter(screen2)

res_db = db.getUser(user_login, path + img_user)
if(res_db["affected"]):
    my_files = os.listdir()
    if img_user in my_files:
        face_reg = cv2.imread(img_user, 0)
        face_log = cv2.imread(img, 0)

        comp = compatibility(face_reg, face_log)

        if comp >= 0.94:
            print("{}Compatibilidad del {:.1%}{}".format(color_success,
float(comp), color_normal))
            printAndShow(screen2, f"Bienvenido, {user_login}", 1)
        else:
            print("{}Compatibilidad del {:.1%}{}".format(color_error,
float(comp), color_normal))
            printAndShow(screen2, "¡Error! Incompatibilidad de datos", 0)
            os.remove(img_user)

    else:
        printAndShow(screen2, "¡Error! Usuario no encontrado", 0)
else:
    printAndShow(screen2, "¡Error! Usuario no encontrado", 0)
    os.remove(img)

def login():
    global screen2
    global user2
    global user_entry2

    screen2 = Toplevel(root)
    user2 = StringVar()

    configure_screen(screen2, txt_login)
    user_entry2 = credentials(screen2, user2, 1)

root = Tk()
root.geometry(size_screen)
root.title("Universidad Politecnica Salesiana")
imagen = PhotoImage(file = "logo.png")
root.configure(bg=color_white)

```

```

Label(text="¡Bienvenido(a) Aula de Robotica!", fg=color_white,
bg=color_blue, font=(font_label, 18), width="800", height="150",
image=imagen).pack()

getEnter(root)
Button(text=txt_login, fg=color_black, bg=color_gris,
activebackground=color_yellow, borderwidth=0, font=(font_label, 14),
height="2", width="40", command=login).pack()

getEnter(root)
Button(text=txt_register, fg=color_black, bg=color_gris,
activebackground=color_yellow, borderwidth=0, font=(font_label, 14),
height="2", width="40", command=register).pack()

root.mainloop()

```

Código de la interfaz

Tabla 10.6 Creación de tabla en la BDD

```

CREATE DATABASE lab;

USE lab;

CREATE TABLE user(
idUser INT AUTO_INCREMENT,
name VARCHAR(75) NOT NULL,
photo LONGBLOB,

CONSTRAINT pk_user_idUser PRIMARY KEY(idUser)
);

SELECT * FROM `user`;

```

Creación tabla de registro

Tabla 10.7 Especificaciones Dimmer Nexxt NHE-D100 (Next Home,2023)

Control	Luz LED	Estado	Descripción
LED de conexión	Azul	Encendida	La luz se ilumina en forma permanente cuando el dispositivo está conectado
	Verde	Apagada	Dispositivo desconectado
Indicador LED de	Rojo	Parpadea rápidamente	Listo para iniciar el emparejamiento (modo predeterminado solamente)

conexión Wi-Fi	Rojo	Parpadea lentamente (una vez cada 3 seg)	Listo para establecer la conexión (modo AP solamente)
	Rojo	Parpadea lentamente	Sin conexión a internet
	Rojo	Permanente	Indica que el emparejamiento se ha realizado exitosamente
Luces de nivel de brillo	Verde	Indicadores de iluminación	Exhiben el nivel de intensidad utilizado, del mín. a máx.
Conmutad or DIP 1		Y	Configure este parámetro para bombillas con intensidad regulable
		N	Configure este parámetro si su bombilla no es regulable (luz convencional)
Conmutad or DIP 2		1	Conmutador 1: Para 110V Aplica a bombillas con los siguientes valores de tensión nominal: a 110VAC, 20mA-100mAh; 2w -10w
			Para 220V Aplica a bombillas con los siguientes valores de tensión nominal: 220VCA, 20mA-50mAh; 4w -10w
		2	*Si la bombilla comienza a parpadear, seleccione la opción 2  Conmutador 2: Para 110V Aplica a bombillas con los siguientes valores de tensión nominal; a 110VCA, 40mA-1A; 2w -10w

			<p>Para 220V</p> <p>Aplica a bombillas con los siguientes valores de tensión nominal: a 220VCA, 20mA-50mAh; 4w -10w</p>
			<p>*Si la bombilla comienza a parpadear, seleccione la opción 1</p>

Descripción de botones y actuadores del dimmer

Tabla 10.8 Rzo Partitions

#	Name,	Type,	SubType,	Offset,	Size,	Flags
	nvs,	data,	nvs,	0x9000,	0x5000,	
	otadata,	data,	ota,	0xe000,	0x2000,	
	app0,	app,	ota_0,	0x10000,	0x280000,	
	fr,	32,	32,	0x290000,	0xEF000,	
	eeprom,	data,	0x99,	0x37f000,	0x1000,	
	spiffs,	data,	spiffs,	0x380000,	0x2F000,	

Particiones para el reconocimiento facial