PROYECTO FINAL DE CICLO

TÉCNICO SUPERIOR EN AUTOMOCIÓN

Sistema Antirrobo para Vehículos Basado en Reconocimiento Facial

Autor: Carlos Fernández García
DNI: 06325902 M
Profesor-coordinador: Ladislao Navarro Carrasco
Modalidad del proyecto: Investigación Experimental
Curso: TSA 2ºB 2024-2025

"El verdadero lujo del automóvil es el tiempo que ahorra a quien lo usa."

Gianni Agnelli

INDICE

1 RESUMEN

El presente proyecto de fin de ciclo aborda el desarrollo de un sistema de seguridad innovador para vehículos, fundamentado en la tecnología de reconocimiento facial y conectividad inalámbrica. El objetivo principal del proyecto reside en diseñar e implementar un sistema que permita el arranque del vehículo únicamente tras la verificación de la identidad del conductor mediante reconocimiento facial.

Se complementa el sistema con una aplicación móvil para el control y gestión del mismo. Se espera que la implementación de este sistema contribuya a un incremento de la seguridad vehicular y a la reducción del riesgo de sustracción de automóviles.

La metodología empleada se basa en la investigación de sistemas antirrobo existentes, el análisis de tecnologías de reconocimiento facial aplicadas a la seguridad automotriz, el desarrollo de un prototipo funcional empleando hardware accesible como Arduino Uno y ESP32-CAM, y la evaluación del rendimiento del sistema en condiciones reales.

Se anticipa que este proyecto aportará una solución viable y adaptable a diversos tipos de vehículos, ofreciendo una alternativa moderna y de mayor seguridad en comparación con los sistemas antirrobo convencionales...

ABSTRACT

An innovative vehicle security system integrating facial recognition and IoT is being developed. The main objective is to design a system that enables vehicle start only after verifying driver identity via facial recognition authorized, enhancing security and mitigating theft risk.

The project encompasses research on conventional anti-theft systems and analysis of facial recognition technologies for vehicle applications. Biometric systems for access control will be implemented, and a functional prototype will be developed using accessible hardware like a Arduino Uno and ESP32-S3-CAM. The system's performance and security will be evaluated under real-world conditions.

This project aims to advance automotive security, offering a modern alternative to traditional anti-theft systems, with a viable, efficient, and adaptable solution for various vehicles. A substantial improvement in theft prevention is proposed through advanced technologies to ensure vehicle security.

Palabras clave: Reconocimiento Facial, *Biométrico*, Seguridad Vehicular, Antirrobo, Arduino Uno, ESP32-*S3*-CAM*, IoT*

2 OBJETIVOS

El proyecto persigue como **objetivo general** el diseño y desarrollo de un sistema antirrobo para vehículos basado en reconocimiento facial que incremente la seguridad y reduzca el riesgo de sustracción.

Para la consecución de este objetivo general, se establecen los siguientes **objetivos específicos**:

Investigar los sistemas antirrobo convencionales existentes en el sector de la automoción, analizando sus funcionalidades, limitaciones y vulnerabilidades.

Analizar las tecnologías de reconocimiento facial disponibles, evaluando su aplicabilidad y eficacia en el contexto de la seguridad vehicular.

Estudiar la viabilidad de implantación del sistema en diferentes tipos de vehículos del parque automovilístico actual.

Desarrollar un prototipo funcional del sistema antirrobo, utilizando hardware accesible y de bajo coste como Arduino Uno y ESP32-CAM.

Evaluar el rendimiento y la seguridad del prototipo en condiciones de funcionamiento reales, identificando posibles mejoras y áreas de optimización.

Documentar el proceso de diseño, desarrollo y evaluación del sistema, elaborando una memoria técnica detallada del proyecto.

Estos objetivos específicos se han definido como pasos lógicos y necesarios para alcanzar el objetivo general del proyecto. Cada uno de ellos se centra en una fase concreta del desarrollo y contribuye de manera directa al resultado final.

3 METODOLOGÍA

La metodología empleada en este proyecto se centra en un enfoque de **investigación aplicada y desarrollo tecnológico**, estructurado en las siguientes fases principales:

Investigación documental y bibliográfica: Se realizará una revisión exhaustiva de la literatura científica y técnica existente en relación con sistemas antirrobo para vehículos, tecnologías de reconocimiento facial y seguridad automotriz. Se analizarán patentes, artículos científicos, publicaciones especializadas y documentación técnica relevante.

Análisis y estudio de tecnologías: Se llevará a cabo un análisis técnico detallado de las tecnologías de reconocimiento facial disponibles, incluyendo diferentes algoritmos, sensores y plataformas de hardware. Se evaluarán sus características, rendimiento, costes y adecuación para la aplicación en el sistema antirrobo.

Diseño del sistema: Se definirá la arquitectura general del sistema antirrobo, especificando los componentes de hardware y software que lo integrarán, así como las interfaces de comunicación entre ellos. Se diseñará la lógica de funcionamiento del sistema, incluyendo los procesos de captura de imagen, reconocimiento facial, verificación de identidad y control del arranque del vehículo. Se diseñará también la aplicación móvil de control del sistema.

Desarrollo del prototipo: Se implementará un prototipo funcional del sistema antirrobo utilizando la plataforma Arduino Uno y la cámara ESP32-CAM, seleccionadas por su accesibilidad y coste reducido. Se desarrollará el software necesario para la gestión del reconocimiento facial, el control del hardware y la comunicación con la aplicación móvil.

Pruebas y evaluación: Se llevarán a cabo pruebas exhaustivas del prototipo en un entorno controlado y, si es posible, en un vehículo real. Se evaluará el rendimiento del sistema en términos de precisión del reconocimiento facial, tiempo de respuesta, fiabilidad, seguridad y usabilidad. Se utilizarán métricas cuantitativas y cualitativas para analizar los resultados de las pruebas.

Documentación y elaboración de la memoria: Se documentará detalladamente todo el proceso de investigación, diseño, desarrollo y evaluación del proyecto. Se elaborará una memoria técnica que recoja todos los aspectos relevantes del trabajo realizado, incluyendo la

descripción del sistema, la metodología empleada, los resultados obtenidos, las conclusiones y las posibles líneas de trabajo futuro.

Esta metodología proporciona un marco de trabajo sistemático y organizado para abordar el desarrollo del proyecto, asegurando la consecución de los objetivos planteados de forma eficiente y rigurosa.

4 CONTENIDO PRINCIPAL

4.1 Sistemas antirrobo en automoción: Estado del arte.

Los sistemas antirrobo para vehículos han evolucionado mucho a lo largo de la historia de la automoción, desde los primeros dispositivos mecánicos hasta las sofisticadas soluciones electrónicas y telemáticas actuales. Tradicionalmente los sistemas antirrobo se han clasificado en función de su nivel de protección y tecnología empleada. Entre los sistemas convencionales más extendidos se encuentran:

Alarmas audibles y visuales: Estos sistemas, ampliamente instalados, se basan en la detección de accesos no autorizados al vehículo, como la apertura de puertas, capó o maletero, o la rotura de cristales. Al detectar una intrusión, activan una señal acústica y luminosa para disuadir al ladrón y alertar al entorno. Si bien son efectivas como elemento disuasorio inicial, las alarmas convencionales pueden ser desactivadas por delincuentes con conocimientos técnicos o ignoradas en entornos urbanos con alta contaminación acústica.

Inmovilizadores de motor: Estos sistemas impiden el arranque del motor en caso de no detectarse la llave o el dispositivo de autorización correcto. Los inmovilizadores electrónicos, integrados en la unidad de control del motor, son más efectivos que los inmovilizadores mecánicos, ya que dificultan la manipulación del sistema de encendido. Sin embargo, algunos inmovilizadores pueden ser vulnerables a técnicas de *bypass* electrónico o al robo de la propia unidad de control (ECU).

Dispositivos antirrobo mecánicos: Este tipo de sistemas incluye elementos como barras de bloqueo para el volante o los pedales, o cepos para las ruedas. Su principal ventaja reside en su simplicidad y robustez física, dificultando el robo por medios expeditivos. No obstante, su

instalación y desinstalación puede resultar engorrosa, y su eficacia depende de la calidad y resistencia del material.

A pesar de la generalización de estos sistemas, el robo de vehículos sigue siendo un problema relevante. Los delincuentes han desarrollado técnicas cada vez más sofisticadas para eludir o neutralizar los sistemas antirrobo convencionales, incluyendo el uso de herramientas electrónicas para la inhibición de alarmas, el *key cloning* para duplicar llaves electrónicas, o el *CAN bus hacking* para manipular la red de comunicaciones del vehículo.

En este contexto, surge la necesidad de desarrollar sistemas antirrobo más avanzados y robustos, que incorporen tecnologías emergentes y ofrezcan una mayor protección frente a las nuevas amenazas. Los sistemas de seguridad biométrica, como el reconocimiento facial, representan una línea de investigación prometedora en este campo, al ofrecer una identificación del usuario más segura y personalizada que los sistemas tradicionales basados en llaves o códigos.

4.2 Tecnología de reconocimiento facial.

El reconocimiento facial es una tecnología biométrica que, impulsada por la **Inteligencia Artificial (IA)**, permite identificar o verificar la identidad de una persona a partir de una imagen o un vídeo de su rostro. Esta tecnología se basa en el análisis y la comparación de patrones faciales únicos, presentes en la estructura y las características del rostro humano. El proceso general de reconocimiento facial se puede dividir en las siguientes etapas principales, donde la IA desempeña un rol crucial:

Detección facial: En esta etapa, el sistema, utilizando algoritmos de IA, localiza y aísla los rostros en la imagen o vídeo. Estos algoritmos, a menudo basados en aprendizaje automático (machine learning), han sido entrenados con grandes conjuntos de datos de imágenes para identificar patrones faciales y distinguirlos de otros elementos visuales.

Extracción de características: Una vez detectado el rostro, se extraen las características faciales distintivas mediante técnicas de IA. Redes neuronales convolucionales (CNNs), un tipo de algoritmo de aprendizaje profundo (deep learning), son frecuentemente empleadas para

mapear y medir rasgos faciales complejos y sutiles. Estas redes neuronales aprenden a identificar las características más relevantes para la identificación facial y las codifican en una plantilla facial compacta y discriminante.

Comparación y reconocimiento: La plantilla facial generada por la IA se compara con una base de datos de plantillas faciales conocidas. Algoritmos de comparación basados en IA evalúan la similitud entre plantillas, teniendo en cuenta la variabilidad natural de los rostros y las posibles distorsiones. Estos algoritmos determinan si existe una coincidencia superando un umbral de similitud predefinido, identificando así a la persona.

La efectividad del reconocimiento facial reside en gran medida en la sofisticación de los algoritmos de IA empleados en cada etapa. El avance en áreas como el aprendizaje profundo ha permitido desarrollar sistemas de reconocimiento facial altamente precisos y robustos, capaces de operar en condiciones variables y con un alto grado de fiabilidad.

Existen diferentes enfoques y algoritmos para el reconocimiento facial, que se pueden clasificar en función de la tecnología de captura de imagen y las técnicas de análisis empleadas. Entre los tipos de sistemas de reconocimiento facial más comunes se encuentran:

Reconocimiento facial 2D: Se basa en el análisis de imágenes faciales bidimensionales, capturadas con cámaras convencionales. Es la tecnología más extendida y económica, pero puede ser vulnerable a variaciones en la iluminación, el ángulo de visión o las expresiones faciales.

Reconocimiento facial 3D: Utiliza sensores tridimensionales para capturar la geometría del rostro, creando un modelo 3D detallado. Es más robusto ante variaciones de iluminación y ángulo, y ofrece mayor precisión y seguridad que el reconocimiento 2D, pero requiere hardware más sofisticado y costoso.

Reconocimiento facial basado en espectro infrarrojo: Utiliza cámaras infrarrojas para capturar imágenes térmicas del rostro, basadas en el patrón de calor emitido por la piel. Es menos sensible a las variaciones de iluminación y puede funcionar en condiciones de baja luminosidad o oscuridad, pero puede verse afectado por cambios en la temperatura corporal.

La tecnología de reconocimiento facial tiene múltiples aplicaciones en diversos ámbitos, incluyendo la seguridad, el control de acceso, la identificación personal, la vigilancia y el marketing. En el sector de la automoción, el reconocimiento facial se está explorando para diversas funcionalidades, como el acceso sin llave al vehículo, la personalización de la configuración del vehículo según el conductor, la monitorización del estado del conductor para prevenir la somnolencia o la distracción, y, como se propone en este proyecto, los sistemas antirrobo.

En el contexto de la seguridad vehicular, el reconocimiento facial ofrece ventajas significativas en comparación con los sistemas tradicionales. Al ser un sistema biométrico, la identificación se basa en una característica única e inherente al individuo, el rostro, lo que dificulta la suplantación de identidad o el uso fraudulento de llaves o códigos. Además, el reconocimiento facial puede integrarse de forma natural en la experiencia del usuario, sin necesidad de interacción física o memorización de contraseñas.

4.3 Hardware y software para el prototipo.

El prototipo funcional del sistema antirrobo se ha implementado mediante la integración de diversos componentes hardware y el desarrollo de software específico. La selección de estos elementos se ha basado en criterios de accesibilidad, eficiencia y adecuación a los requerimientos del proyecto.

4.3.1 Componentes Hardware

La parte hardware del prototipo está compuesta por componentes principales que ejecutan la lógica central y el reconocimiento facial, y componentes secundarios que proporcionan soporte, alimentación, interacción visual y actuación.

4.3.1.1 Componentes Principales

Placa Arduino Uno: (Figura 4.1) Se ha seleccionado esta placa como microcontrolador principal debido a su robustez, facilidad de programación y amplia comunidad de soporte. Arduino Uno proporciona la plataforma para controlar los actuadores clave del sistema antirrobo y gestionar la comunicación con el módulo ESP32-S3-CAM. Sus pines de

entrada/salida digital son fundamentales para la activación del relé de arranque del vehículo simulado, y su interfaz serie facilita la comunicación de datos y comandos con la ESP32-S3-CAM.

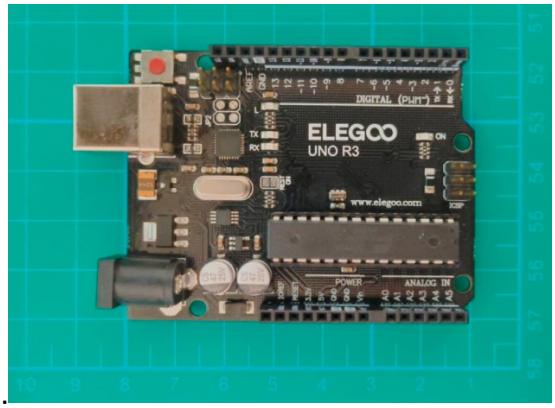


Figura 4.1: Placa Arduino Uno, microcontrolador principal del prototipo. Foto: propia

Módulo ESP32-S3-CAM: (Figura 4.2) Este módulo integra las funcionalidades de captura de imagen, procesamiento inicial para reconocimiento facial y conectividad inalámbrica. La inclusión de un microcontrolador ESP32-S3 le confiere suficiente capacidad de procesamiento para ejecutar algoritmos de visión artificial, y su cámara permite obtener las imágenes del conductor. La conectividad WiFi y Bluetooth integradas son esenciales para la comunicación con la aplicación móvil de control y monitorización. La elección de la versión S3 se justifica por su mejor rendimiento y mayor memoria en comparación con modelos anteriores.

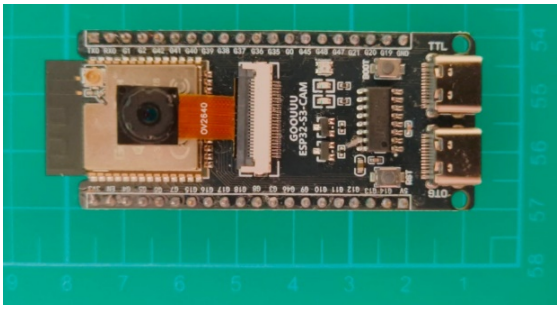


Figura 4.2: Módulo ESP32-S3-CAM, para reconocimiento facial y la comunicación WiFi. Foto propia

4.3.1.2 Componentes Secundarios

Además de las placas principales, el prototipo integra los siguientes componentes para simular un entorno vehicular funcional y facilitar la interacción y pruebas:

Fuente de alimentación ATX: Se utiliza una fuente de alimentación ATX, típicamente empleada en sistemas informáticos, para suministrar las tensiones de alimentación necesarias (principalmente +5V y +12V) a los distintos componentes electrónicos del prototipo. Esta fuente proporciona una alimentación estable y con suficiente capacidad para el conjunto del sistema en fase de desarrollo.

Módulo de relé: Un módulo de relé controlado por la placa Arduino Uno se emplea para simular la interrupción o activación del circuito de arranque del vehículo. Este componente actúa como interfaz entre la lógica de control de bajo voltaje de Arduino y la posible conmutación de corrientes o voltajes mayores, representando el mecanismo que físicamente habilitaría o deshabilitaría el arranque del motor.

Pantallas OLED (0.9" 128x64 y 0.9" 128x32): Se han incorporado dos pequeñas pantallas OLED de diferentes resoluciones para proporcionar *feedback* visual al usuario y al desarrollador durante las pruebas. Estas pantallas permiten mostrar información relevante como el estado del sistema, mensajes de error, resultados del reconocimiento facial o instrucciones. Su reducido tamaño las hace adecuadas para la integración en un prototipo compacto.

Piezas de LEGO: Se han utilizado piezas de construcción LEGO para diseñar y montar la estructura física de soporte del prototipo. Esta solución permite crear un banco de pruebas o una simulación del habitáculo de un vehículo de forma flexible y adaptable, facilitando la disposición de los componentes hardware y la simulación de un motor funcional para demostración.

Cableado diverso: El prototipo requiere una variedad de cables y conectores (cables jumper, cables de alimentación, etc.) para establecer las interconexiones eléctricas y de comunicación entre todos los componentes hardware, asegurando el correcto flujo de energía y datos a través del sistema.

4.3.2 Componentes Software

El desarrollo de software abarca la programación de los microcontroladores y la creación de una aplicación móvil para la interacción con el sistema.

Entorno de Desarrollo Integrado (IDE): Se ha utilizado el entorno de desarrollo de Arduino IDE para la programación tanto de la placa Arduino Uno como del módulo ESP32-S3-CAM, aprovechando su simplicidad y la disponibilidad de múltiples librerías.

Librerías de Reconocimiento Facial: Para la funcionalidad de reconocimiento facial en el ESP32-S3-CAM, se han implementado librerías específicas optimizadas para este hardware, aprovechando su capacidad de procesamiento para tareas de visión artificial.

Protocolo de Comunicación: La interacción inalámbrica entre la aplicación móvil y el módulo ESP32-S3-CAM se ha implementado **exclusivamente** mediante el protocolo **WebSockets**. Este protocolo bidireccional permite una comunicación en tiempo real eficiente entre el dispositivo móvil y el microcontrolador.

Aplicación Móvil: Se ha desarrollado una aplicación móvil para el control y monitorización del sistema. Esta aplicación ha sido creada específicamente para el sistema operativo **Android** utilizando el entorno de desarrollo **Android Studio**. El código fuente de esta aplicación se encuentra alojado en el repositorio de GitHub:

https://github.com/Esmecarbea/TFCCarlosFdezIESBarajas15

4.4 Arquitectura del Sistema.

La arquitectura del prototipo del sistema antirrobo basado en reconocimiento facial se ha diseñado e implementado bajo un enfoque modular y distribuido, cuya lógica general se ilustra en el diagrama de flujo (Figura 4.3). Los componentes clave, ESP32-S3-CAM, Arduino Uno y la aplicación móvil, interactúan de forma coordinada para lograr la funcionalidad deseada. La validación preliminar de la lógica de interconexión y control, excluyendo las funcionalidades directas de la cámara, se llevó a cabo mediante una **simulación detallada en Proteus 8** (Figuras 4.4, 4.5 Y 4.6). En esta fase de simulación, las señales generadas por la cámara en la ESP32-S3-CAM fueron emuladas utilizando pulsadores, permitiendo verificar el correcto funcionamiento del cableado, la lógica de control del Arduino Uno y la interacción simulada entre módulos antes de proceder al montaje físico.

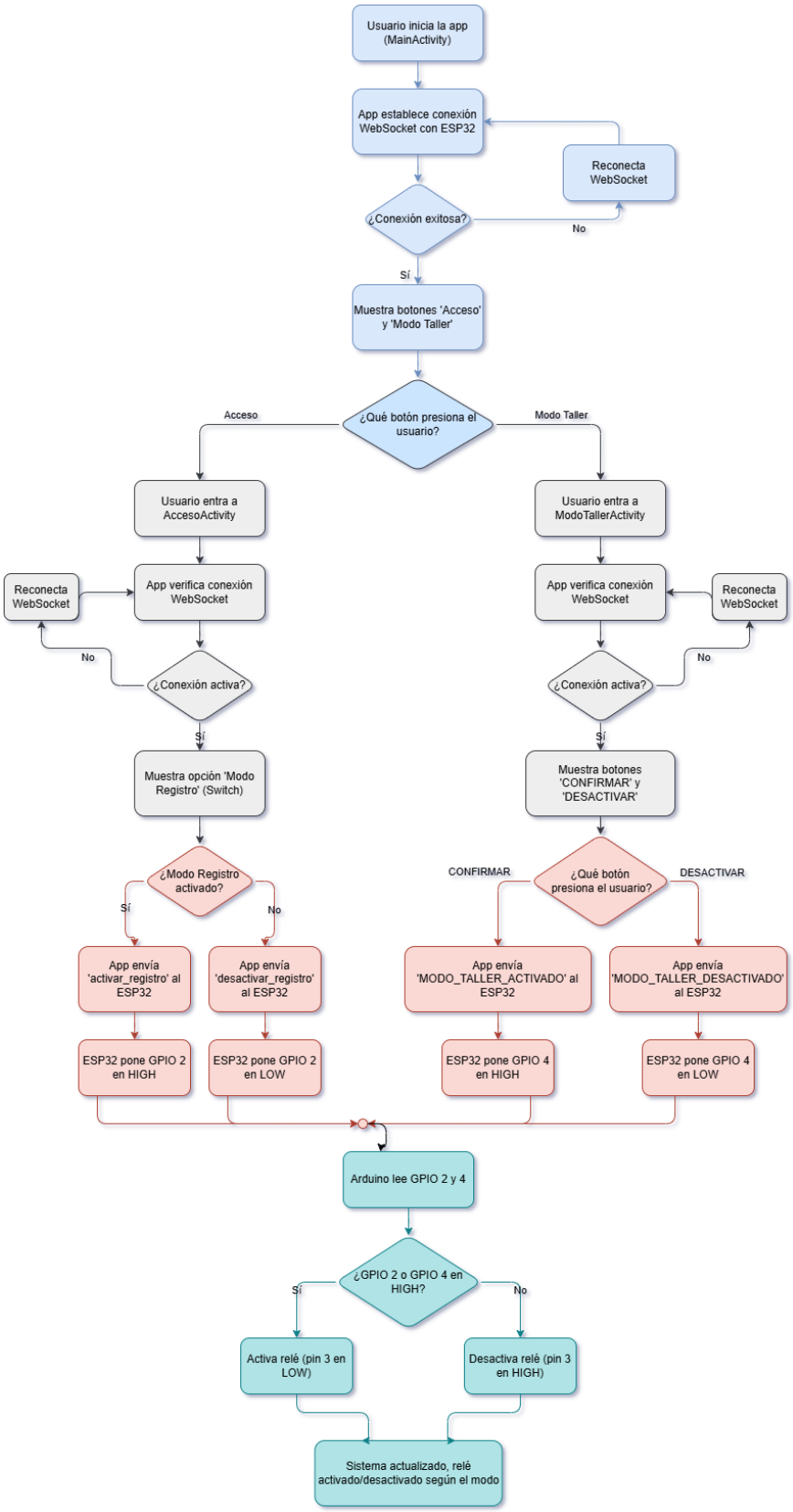


Figura 4.3 Diagrama de flujo de la lógica de funcionamiento de la aplicación móvil.

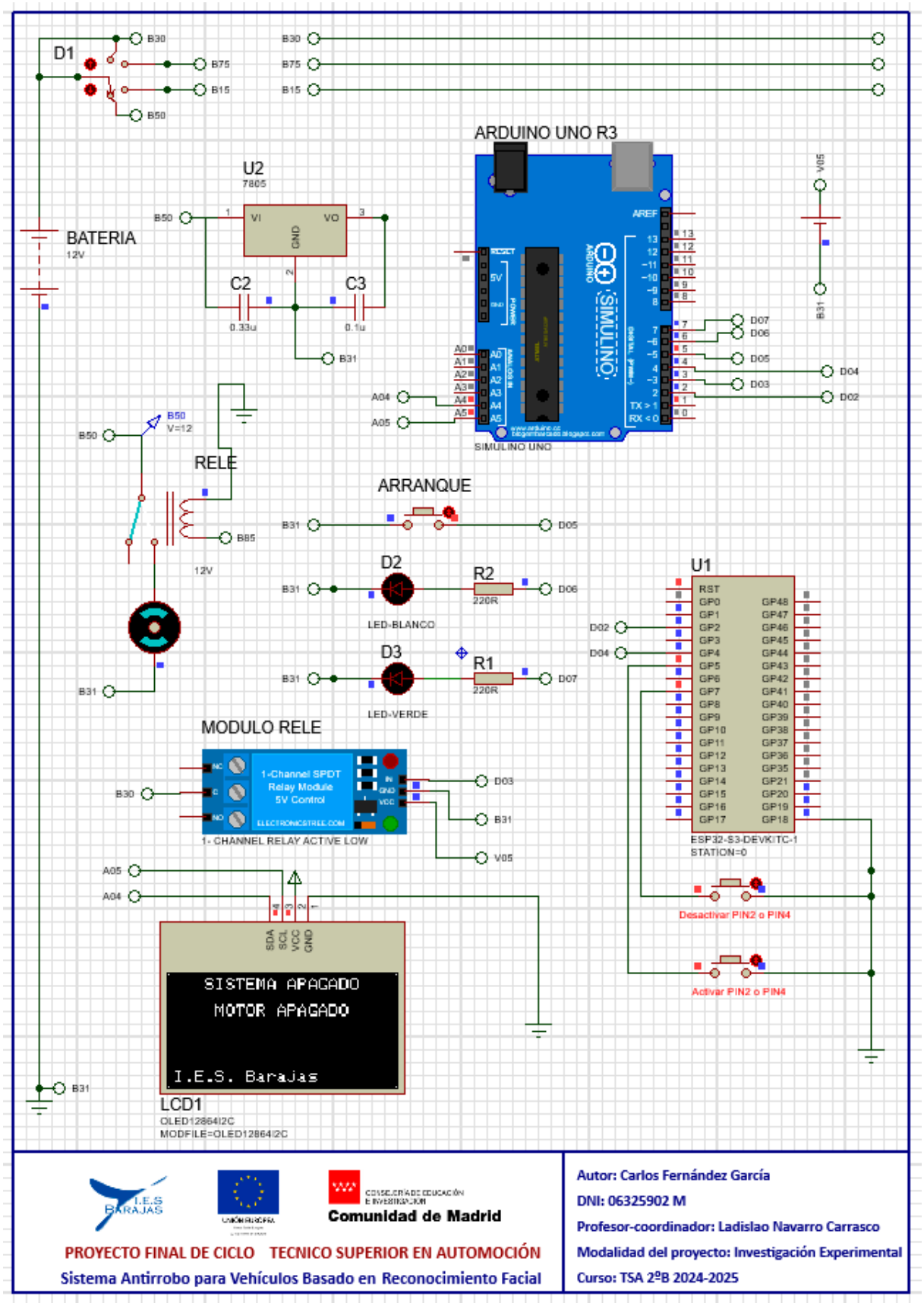


Figura 4.4: Simulación 1 en Proteus 8 SISTEMA APAGADO- MOTOR APAGADO.

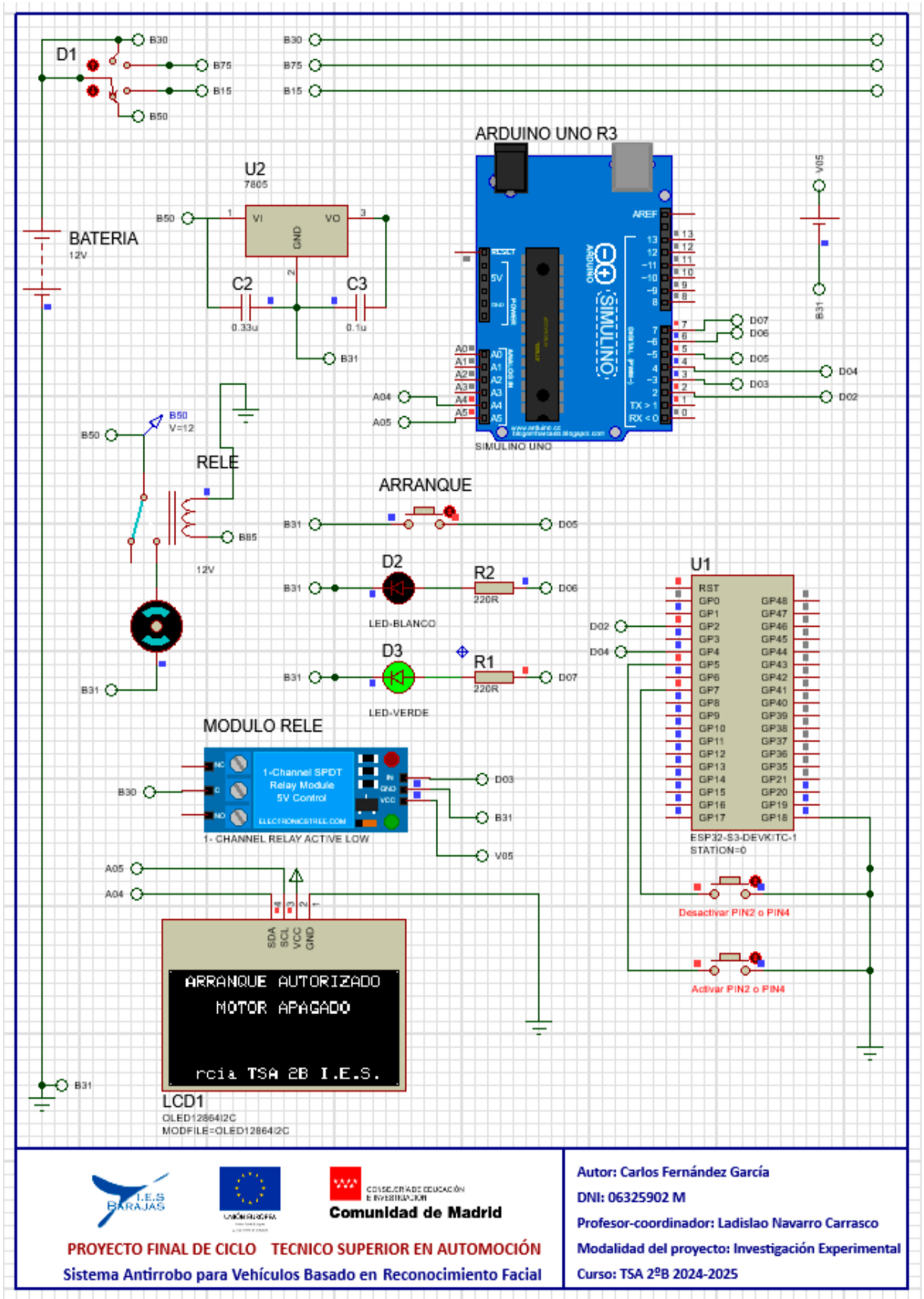


Figura 4.5: Simulación 2 en Proteus 8, ARRANQUE AUTORIZADO-MOTOR APAGADO.

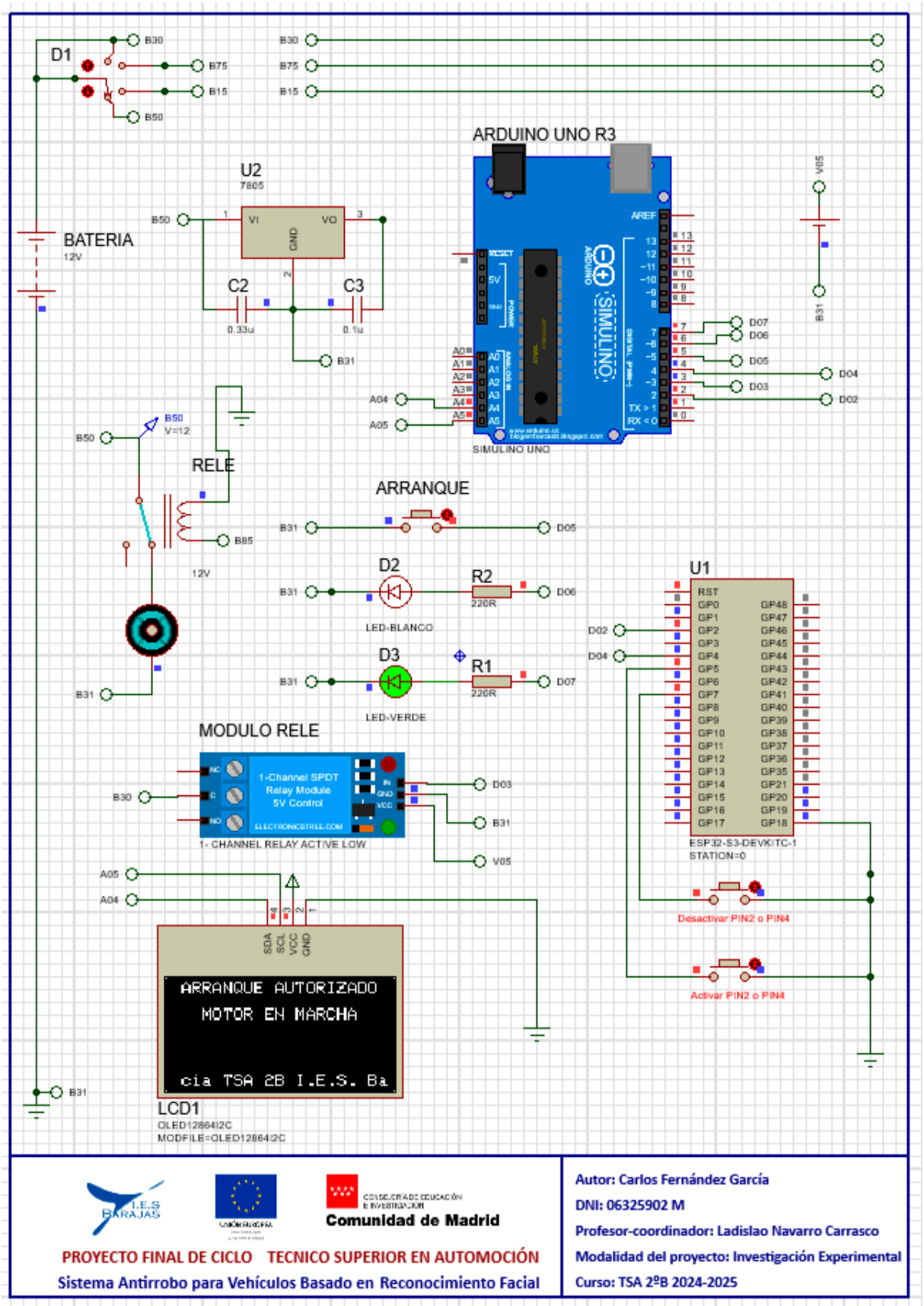


Figura 4.6: Simulación 3 en Proteus 8, ARRANQUE AUTORIZADO-MOTOR EN MARCHA.

La arquitectura final se articula en torno a la siguiente interacción entre los módulos:

- **Módulo ESP32-S3-CAM:** Este módulo constituye la unidad principal de procesamiento de imagen y comunicación inalámbrica. Es responsable de la captura continua de vídeo, la detección facial y la ejecución del algoritmo de reconocimiento facial para verificar la identidad de la persona frente a la cámara. Adicionalmente, gestiona la conectividad WiFi y actúa como servidor para la comunicación basada en el protocolo WebSockets con la aplicación móvil.

- **Comunicación ESP32-S3-CAM ↔ Arduino Uno (Señalización Digital):** A diferencia de una comunicación serie compleja, la interacción entre la ESP32-S3-CAM y el Arduino Uno se simplifica a una **señalización digital directa mediante el cambio de estado de pines específicos**.

    - La ESP32-S3-CAM utiliza el **PIN 2** como salida digital para indicar el **estado de autorización de arranque**. Este pin se pone en estado lógico **HIGH** cuando se ha detectado y reconocido un usuario autorizado, y en estado lógico **LOW** en caso contrario (no se detecta rostro, intruso, error, etc.).

    - Asimismo, la ESP32-S3-CAM utiliza el **PIN 4** como salida digital para señalizar el estado del **"Modo Taller"**. Este pin se pone en estado lógico **HIGH** cuando el Modo Taller está activado desde la aplicación móvil (permitiendo el arranque por cualquier persona), y en estado lógico **LOW** cuando el Modo Taller está desactivado.

- **Placa Arduino Uno:** Esta placa actúa como el **controlador de bajo nivel y la interfaz con los actuadores simulados del vehículo**. Lee continuamente el estado de los pines digitales 2 y 4 provenientes de la ESP32-S3-CAM. Basándose en el estado de estos pines (PIN 2 HIGH o PIN 4 HIGH) y la interacción con el pulsador de arranque simulado, el Arduino Uno determina si se debe activar o desactivar el módulo de relé que simula el circuito de encendido del motor. También gestiona las pantallas OLED para visualización de estados y controla los LEDs indicadores.

- **Comunicación ESP32-S3-CAM ↔ Aplicación Móvil (WebSockets):** La aplicación móvil Android se comunica de forma inalámbrica con la ESP32-S3-CAM utilizando exclusivamente el **protocolo WebSockets** a través de una red WiFi. Esta conexión bidireccional permite:
    - **Desde la App a la ESP32-S3-CAM:** Enviar comandos para controlar funcionalidades como la activación/desactivación del modo de registro facial o la activación/desactivación del "Modo Taller".
    - **Desde la ESP32-S3-CAM a la App:** Enviar información sobre el estado del sistema, mensajes de eventos (por ejemplo, "Intruder", "Enrolling face...", resultado del registro) y el *stream* de vídeo de la cámara para visualización en la interfaz de la aplicación.

La integración de las pantallas OLED y los LEDs indicadores se realiza directamente a la placa Arduino Uno, proporcionando *feedback* visual sobre el estado operativo del sistema (autorización de arranque, estado del motor simulado) sin depender de la aplicación móvil. El módulo de relé, que simula el control del motor, también se conecta y es controlado directamente por la placa Arduino Uno. La fuente de alimentación ATX proporciona la energía necesaria a todos los componentes hardware.

Esta arquitectura permite que las tareas intensivas de procesamiento (reconocimiento facial) recaigan en la ESP32-S3-CAM, mientras que el Arduino Uno se encarga de la lógica de control simple basada en señales digitales y la gestión de periféricos básicos, creando un sistema robusto y eficiente.

4.5 Implementación del Firmware en ESP32-S3-CAM.

El *firmware* desarrollado para el módulo ESP32-S3-CAM constituye el núcleo del sistema, encargado de la adquisición de imágenes, el procesamiento para reconocimiento facial, la gestión de la comunicación inalámbrica y la señalización de estados clave a la placa Arduino Uno. El código se estructura principalmente en dos archivos: 006*.ino, que contiene la lógica principal de configuración e inicialización y el bucle de ejecución (*loop*), y app_httpd.cpp, que

implementa las funcionalidades del servidor web y maneja las operaciones de cámara y reconocimiento facial.

La inicialización del *firmware* comienza con la configuración de la cámara y el módulo ESP32-S3-EYE (*#define CAMERA_MODEL_ESP32S3_EYE*) mediante la estructura camera_config_t y la función esp_camera_init(). Se configuran parámetros como el formato de píxel (JPEG), el tamaño del *frame* (240x240), la calidad de compresión y la ubicación del *framebuffer* en la PSRAM para optimizar el rendimiento del reconocimiento facial.

Simultáneamente, se inicializa la comunicación I2C para interactuar con la pantalla OLED de 128x32 píxeles (*U8G2_SSD1306_128X32_UNIVISION_F_HW_I2C u8g2(...)*), utilizándose para mostrar mensajes de estado inicial, como el proceso de inicialización y la dirección IP asignada.

Tras la inicialización de los periféricos, se establece la conexión a la red WiFi utilizando las credenciales predefinidas (*WiFi.begin(ssid, password)*). Una vez conectado, se inicia el servidor web (*server.begin()*) en el puerto 82, que incluye un *handler* para el protocolo WebSockets (*ws("/ws")*).

El servidor WebSocket (ws) gestiona la comunicación bidireccional con la aplicación móvil Android. La función onWsEvent procesa los mensajes recibidos desde la aplicación, incluyendo comandos para activar o desactivar el modo de registro de nuevos usuarios (*activar_registro*, *desactivar_registro*), y para activar o desactivar el "Modo Taller" (*MODO_TALLER_ACTIVADO*, *MODO_TALLER_DESACTIVADO*). Asimismo, se encarga de enviar mensajes de estado a la aplicación (por ejemplo, informando sobre la activación/desactivación del modo de registro).

Paralelamente, se inicia un servidor HTTP (startCameraServer()) que proporciona acceso al *stream* de vídeo en tiempo real (/stream) y a capturas individuales (/capture) a través del puerto 81. Este servidor también expone *endpoints* para controlar parámetros de la cámara y obtener su estado, aunque la funcionalidad principal para la interacción con la App se basa en WebSockets y la gestión de pines.

La funcionalidad central de **detección y reconocimiento facial** se implementa en el archivo app_httpd.cpp, aprovechando librerías optimizadas para el ESP32-S3

(*human_face_detect_msr01*, *human_face_detect_mnp01*, *face_recognition_tool*, *face_recognition_112_v1_s8*). El proceso implica la captura de un *frame* de vídeo, la detección de rostros dentro de la imagen y, si la detección tiene éxito, la extracción de características faciales. Estas características se comparan con una base de datos de usuarios registrados almacenada en la memoria flash (partición "fr") utilizando algoritmos de reconocimiento. La función run_face_recognition encapsula esta lógica y determina si el rostro detectado corresponde a un usuario autorizado.

Un aspecto crucial de la implementación es la **señalización de estados a la placa Arduino Uno mediante pines digitales**.

- Dentro de la función run_face_recognition, si se detecta un rostro y se verifica como usuario autorizado (recognize.id >= 0), el **PIN 2** de la ESP32-S3-CAM se pone en estado lógico **HIGH** (*digitalWrite(pin2, HIGH)*). Si no se detecta rostro o el usuario no está reconocido, el PIN 2 se pone en estado lógico **LOW** (*digitalWrite(pin2, LOW)*).

- Las funciones activateWorkshopMode() y deactivateWorkshopMode(), llamadas desde el *handler* de WebSockets (onWsEvent), controlan el estado del **PIN 4**. activateWorkshopMode() pone el PIN 4 en **HIGH** (*digitalWrite(pin4, HIGH)*), señalizando la activación del Modo Taller, mientras que deactivateWorkshopMode() lo pone en **LOW** (*digitalWrite(pin4, LOW)*).

El estado del **"Modo Taller"** (estado del PIN 4) se hace persistente utilizando la memoria no volátil (NVS) a través de la librería Preferences. La función setupPin4() lee el último estado guardado al inicio (*preferences.getBool("pin4State", false)*) y lo aplica al PIN 4, asegurando que el sistema recuerde si el Modo Taller estaba activo tras un reinicio del módulo. Las funciones activateWorkshopMode() y deactivateWorkshopMode() actualizan este valor en la NVS (*preferences.putBool("pin4State", pin4State)*).

En resumen, el *firmware* de la ESP32-S3-CAM integra la adquisición y procesamiento de imágenes para reconocimiento facial, la comunicación inalámbrica con la aplicación móvil para control y monitorización, y la señalización digital de estados clave (autorización y modo taller) a la placa Arduino Uno, actuando como el "cerebro" del sistema de seguridad.

4.6 Implementación del Firmware en Arduino Uno.

El *firmware* desarrollado para la placa Arduino Uno (ARDUINOPANTALLAYLEDSFULLOK.ino) desempeña la función de unidad de control secundaria, encargada de la lectura de señales de estado provenientes de la ESP32-S3-CAM y la gestión de los actuadores y dispositivos de visualización del prototipo simulado. Su programación se ha realizado utilizando el entorno de desarrollo de Arduino IDE.

La fase de inicialización (*setup()*) configura los distintos pines digitales de la placa. Se definen los **Pines 2 y 4** como **entradas** (*pinMode(inputPin2, INPUT)*, *pinMode(inputPin4, INPUT)*) para recibir las señales de autorización de arranque y de Modo Taller desde la ESP32-S3-CAM. El **Pin 3** se configura como **salida** (*pinMode(relayPin, OUTPUT)*) para controlar el módulo de relé que simula el circuito de encendido del motor, inicializándose en estado **HIGH** (*digitalWrite(relayPin, HIGH)*) para mantener el relé desactivado inicialmente. Los **Pines 6 y 7** se configuran también como **salidas** (*pinMode(ledWhitePin, OUTPUT)*, *pinMode(ledGreenPin, OUTPUT)*) para controlar los LEDs indicadores blanco y verde, respectivamente, inicializándose en estado **LOW** (*digitalWrite(ledWhitePin, LOW)*, *digitalWrite(ledGreenPin, LOW)*) para permanecer apagados. Adicionalmente, el **Pin 5** se configura como **entrada con resistencia *pull-up*** (*pinMode(buttonPin, INPUT_PULLUP)*) para leer el estado del pulsador que simula el botón de arranque del vehículo.

La comunicación serie se inicializa (*Serial.begin(115200)*) para facilitar la depuración y monitorización del estado del sistema. La pantalla OLED de 128x64 píxeles (*Adafruit_SSD1306 display(...)*), conectada vía I2C, se inicializa con la dirección 0x3C (*display.begin(SSD1306_SWITCHCAPVCC, 0x3C)*). Se configura el tamaño y color del texto para la visualización en la pantalla.

El bucle principal de ejecución (*loop()*) realiza una lectura continua del estado lógico de los pines de entrada 2 y 4 provenientes de la ESP32-S3-CAM (*digitalRead(inputPin2)*, *digitalRead(inputPin4)*). Esta lectura es fundamental, ya que determina la condición de **"ARRANQUE AUTORIZADO"**. Dicha condición se considera

verdadera si el estado del Pin 2 o el estado del Pin 4 es **HIGH** (*statePin2 == HIGH || statePin4 == HIGH*).

Basándose en la condición de "ARRANQUE AUTORIZADO":

- Se controla el **LED verde** (ledGreenPin): Se enciende si la condición es verdadera y se apaga en caso contrario.

- Se actualiza el **texto en la pantalla OLED**: Muestra "ARRANQUE AUTORIZADO" si la condición se cumple, o "SISTEMA APAGADO" si no se cumple y el motor no está en marcha. También se visualiza el estado actual del motor ("MOTOR EN MARCHA" o "MOTOR APAGADO") y un texto en scroll en la parte inferior.

El control del **motor simulado** (activación/desactivación del relé y LED blanco) se gestiona a través de la lectura del pulsador (buttonPin). Se detecta el **flanco descendente** del pulsador (*buttonState == LOW && lastButtonState == HIGH*), lo que indica una pulsación del botón de arranque. Sin embargo, la activación del motor solo se produce si, simultáneamente a la detección del flanco descendente, la condición de **"ARRANQUE AUTORIZADO"** es verdadera. Si ambas condiciones se cumplen, se alterna el estado del motor (*motorState = !motorState*).

La activación o desactivación física del relé (relayPin) se realiza mediante digitalWrite(relayPin, motorState ? LOW : HIGH). Es importante notar que, según la configuración utilizada, el relé se activa con un estado lógico **LOW** y se desactiva con un estado lógico **HIGH** en el pin de control. El LED blanco (ledWhitePin) refleja visualmente el estado del motor, encendiéndose o apagándose según el valor de motorState.

Finalmente, se implementa una lógica de seguridad adicional: si en cualquier momento la condición de "ARRANQUE AUTORIZADO" deja de cumplirse (es decir, tanto el Pin 2 como el Pin 4 pasan a estar en LOW), el *firmware* de Arduino **apaga forzosamente el motor simulado** y el LED blanco (*motorState = LOW; digitalWrite(relayPin, LOW); digitalWrite(ledWhitePin, LOW);*), independientemente del estado previo o la interacción con el pulsador.

En síntesis, el *firmware* de Arduino Uno actúa como una capa de control y seguridad, interpretando las señales simples de autorización de la ESP32-S3-CAM para gestionar la

activación del motor simulado a través de un relé, proporcionar información visual al usuario y asegurar la desconexión del sistema si la autorización principal se pierde.

4.7 Desarrollo del Software de la Aplicación Móvil Android.

El software de la aplicación móvil constituye la interfaz principal para la interacción del usuario con el sistema antirrobo, permitiendo controlar ciertas funcionalidades y visualizar el estado del sistema y el *stream* de vídeo de la cámara. El desarrollo de esta aplicación se ha realizado de forma nativa para el sistema operativo **Android** utilizando el entorno **Android Studio** y el lenguaje de programación **Kotlin**. La comunicación inalámbrica con el módulo ESP32-S3-CAM se gestiona exclusivamente a través del protocolo **WebSockets**, para lo cual se ha empleado la librería **OkHttp** de Square.

La aplicación se estructura en **tres actividades principales**, cada una responsable de un conjunto específico de funcionalidades:

1. **MainActivity:** Actúa como el punto de entrada de la aplicación (Figura 4.7). Su función principal es establecer la conexión inicial con el servidor WebSocket alojado en la ESP32-S3-CAM a través de una instancia de la clase WebSocketManager. Una vez establecida la conexión (o intentado establecerla), presenta al usuario las opciones de navegación principales: acceder a la funcionalidad de "Acceso" (reconocimiento facial y registro) o al "Modo Taller".

Figura 4.7: Captura de pantalla de la MainActivity de la aplicación.

AccesoActivity: Esta actividad gestiona la interfaz relacionada con el **reconocimiento facial y el registro de usuarios** (Figura 4.8). Al iniciar, verifica la conexión WebSocket existente a través del WebSocketManager. Utiliza un componente **WebView** para cargar y mostrar el *stream* de vídeo en tiempo real proveniente del servidor HTTP de la ESP32-S3-CAM (en la URL http://192.168.46.44:81/stream). Se ha implementado un mecanismo de *timeout* para la carga del *stream* y manejo básico de errores de conexión del WebView. Esta actividad permite al usuario, mediante un *Switch* específico, **activar o desactivar el modo de registro** de nuevos usuarios. Al accionar este *Switch*, se envía el mensaje correspondiente (*activar_registro* o *desactivar_registro*) a la ESP32-S3-CAM a través del WebSocketManager. La actividad también recibe y procesa mensajes provenientes de la ESP32-S3-CAM vía WebSocket (gestionados por un WebSocketListener interno), actualizando la interfaz de usuario para mostrar información relevante como "Intruso detectado", "Registrando cara...", "Usuario registrado con ID: X", etc.

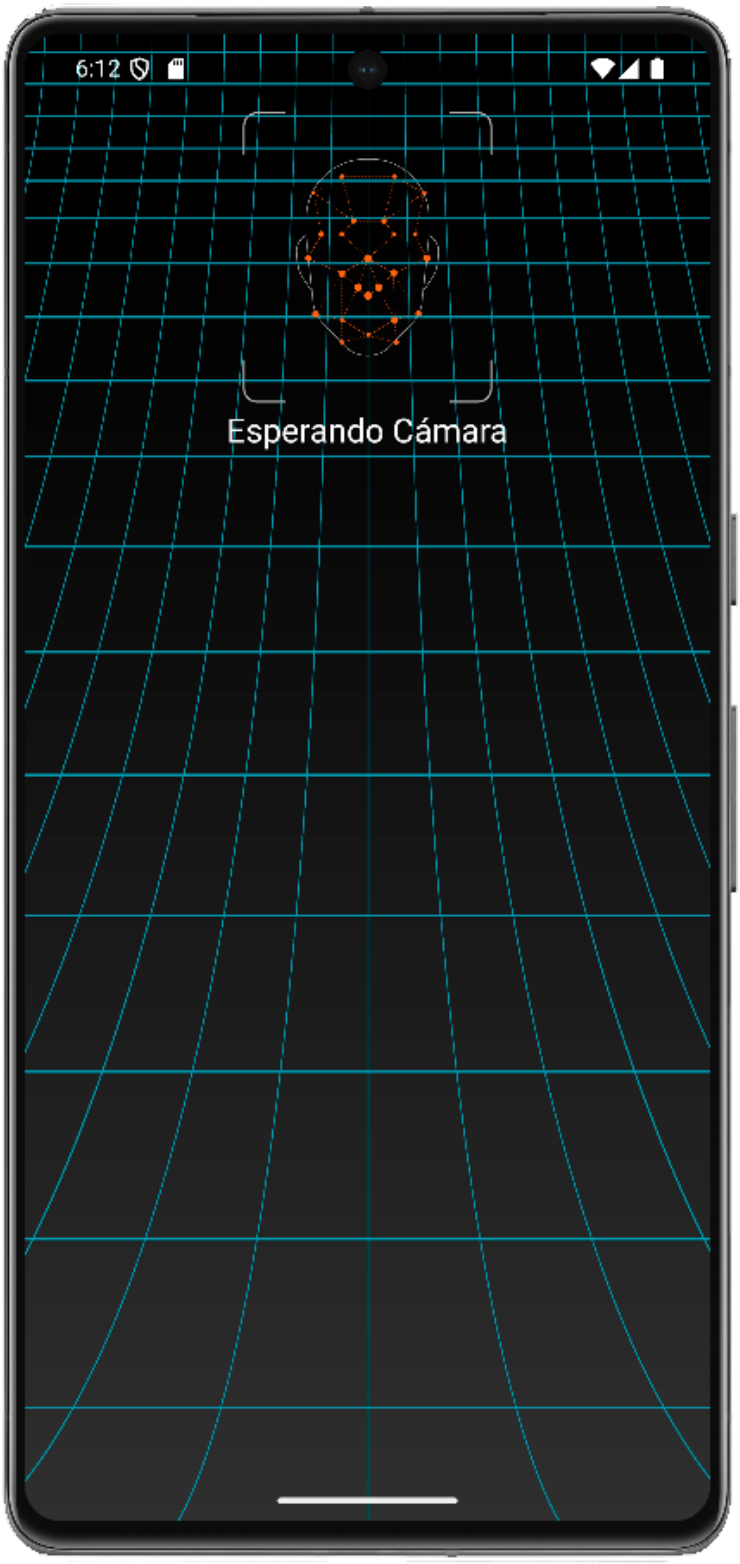


Figura 4.8: Captura de pantalla de la AccesoActivity de la aplicación ESPERANDO CÁMARA

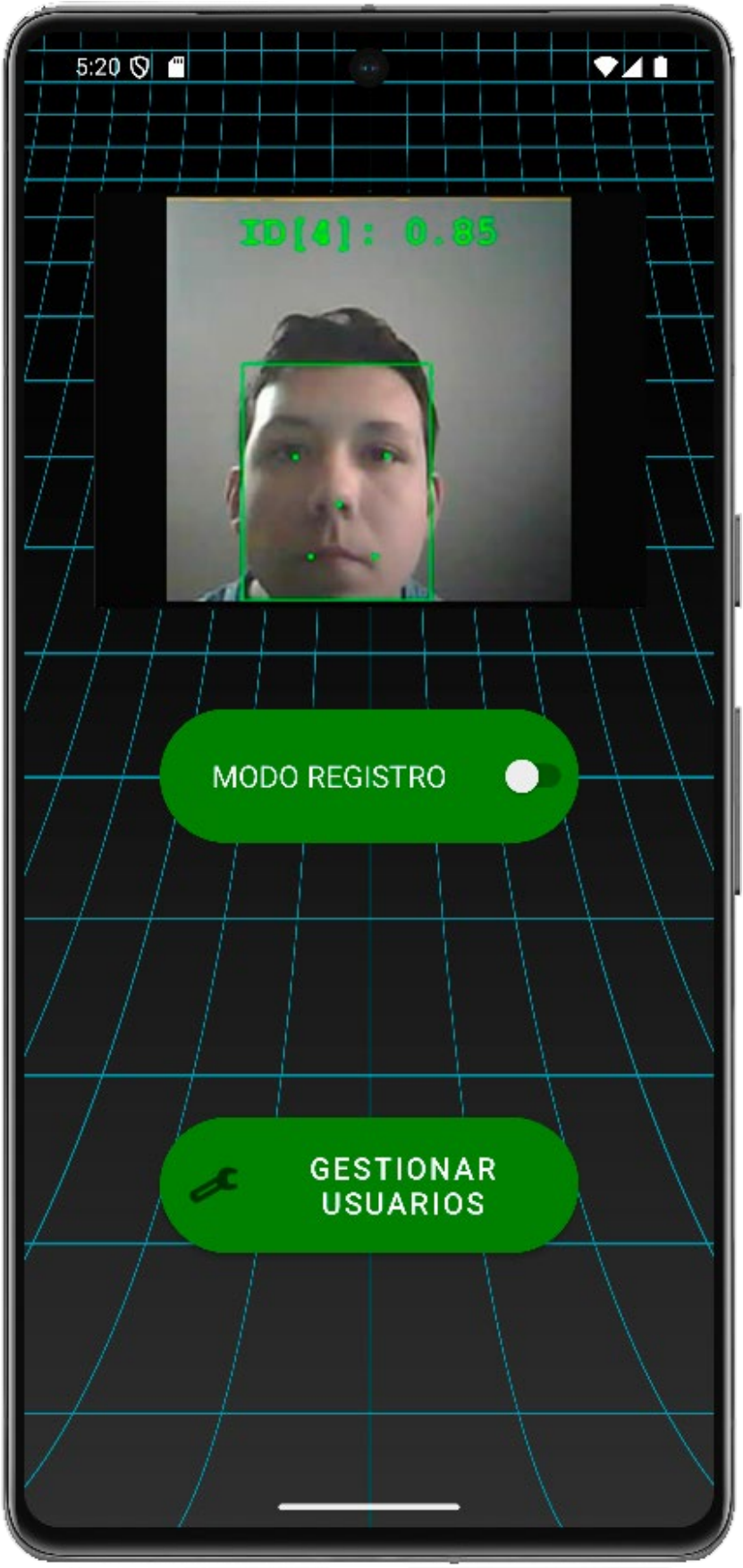


Figura 4.9: Captura de pantalla de la AccesoActivity USUARIO AUTORIZADO

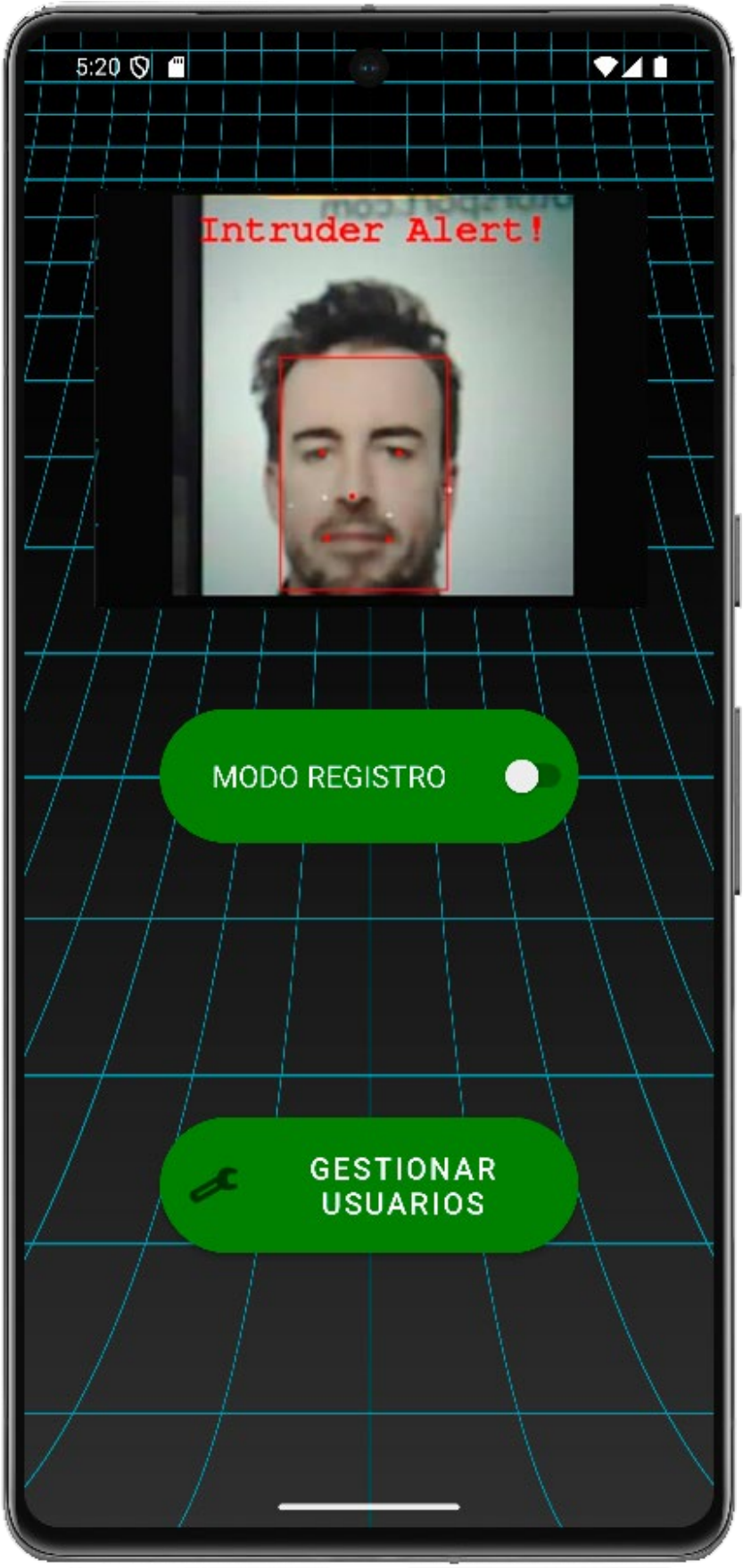


Figura 4.9: Captura de pantalla de la AccesoActivity USUARIO NO AUTORIZADO

2. **ModoTallerActivity:** Esta actividad se encarga de la gestión del **"Modo Taller"** del sistema (Figura 4.W). Permite al usuario activar o desactivar esta funcionalidad mediante un *Switch*. Al cambiar el estado del *Switch*, se envía el mensaje correspondiente (*MODO_TALLER_ACTIVADO* o *MODO_TALLER_DESACTIVADO*) a la ESP32-S3-CAM a través del WebSocketManager. El estado de este *Switch* se **persiste** utilizando SharedPreferences de Android, lo que asegura que el Modo Taller permanezca activo (o desactivado) incluso si la aplicación se cierra y se vuelve a abrir. La interfaz de usuario de esta actividad incluye elementos visuales (texto, fondo circular) y animaciones para indicar claramente el estado del Modo Taller.

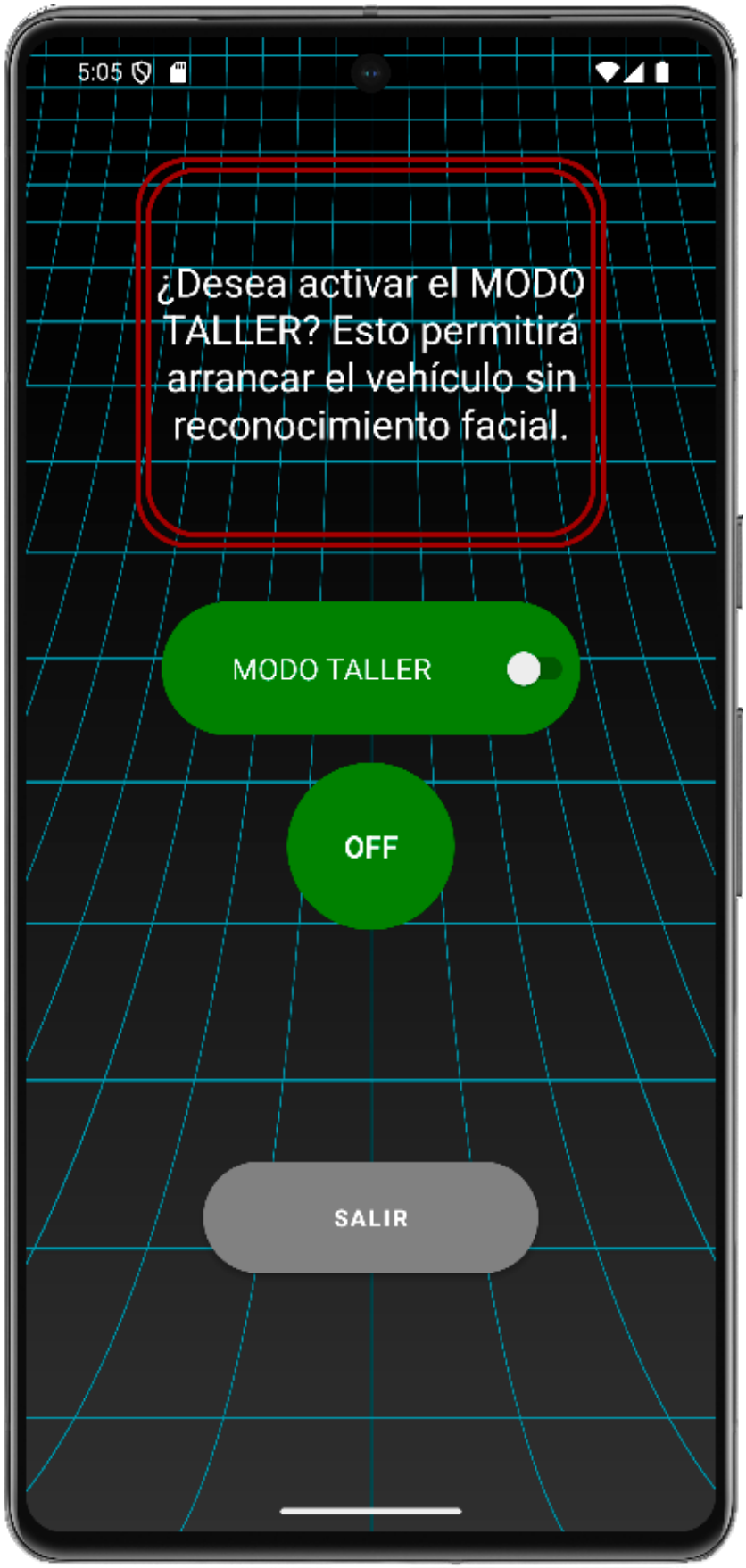


Figura 4.10: Captura de pantalla de la ModoTallerActivity de la aplicación DESACTIVADO

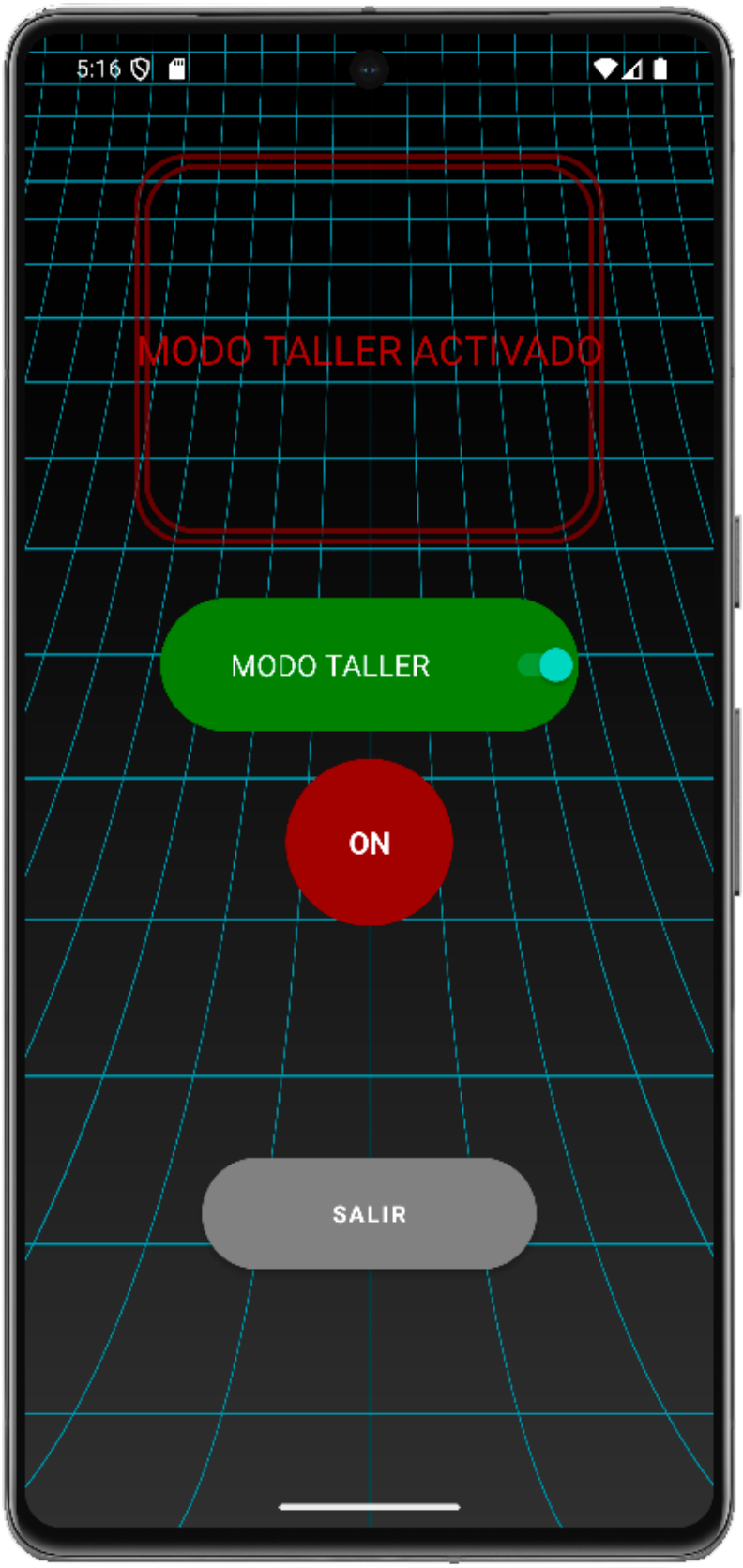


Figura 4.10: Captura de pantalla de la ModoTallerActivity de la aplicación ACTIVADO

La clase **WebSocketManager** encapsula la lógica de conexión y gestión del cliente WebSocket. Se inicializa una instancia en la MainActivity y se comparte (como *companion object*) entre las actividades para mantener una única conexión activa durante la vida útil de la aplicación. Esta clase maneja la creación del cliente OkHttp, la solicitud de conexión a la URL especificada (ws://192.168.46.44:82/ws), y proporciona métodos para enviar mensajes de texto (sendMessage) y gestionar los eventos del ciclo de vida del WebSocket (apertura, recepción de mensajes, cierre, fallo) mediante la delegación a un WebSocketListener proporcionado por la actividad activa. La implementación incluye lógica básica de logging para depuración.

En resumen, el software de la aplicación móvil proporciona la interfaz de usuario necesaria para que el usuario interactúe con el sistema de reconocimiento facial y el Modo Taller, visualice el *stream* de la cámara y reciba notificaciones de estado, actuando como el punto de control y monitorización del sistema a través de la comunicación inalámbrica WebSocket.

5 CONTENIDO ADICIONAL: REALIZACIÓN FÍSICA DEL PROTOTIPO

5.1 Montaje y Configuración Física.

La materialización del prototipo funcional ha implicado el ensamblaje e interconexión de los diversos componentes hardware y electrónicos sobre una estructura de soporte modular. Esta realización física busca simular de forma simplificada el entorno de un vehículo y validar la integración de los distintos módulos del sistema, particularmente la interacción entre el reconocimiento de acceso, la lógica de control y la simulación del sistema de arranque. La configuración general del prototipo se presenta en la Figura 5.1.

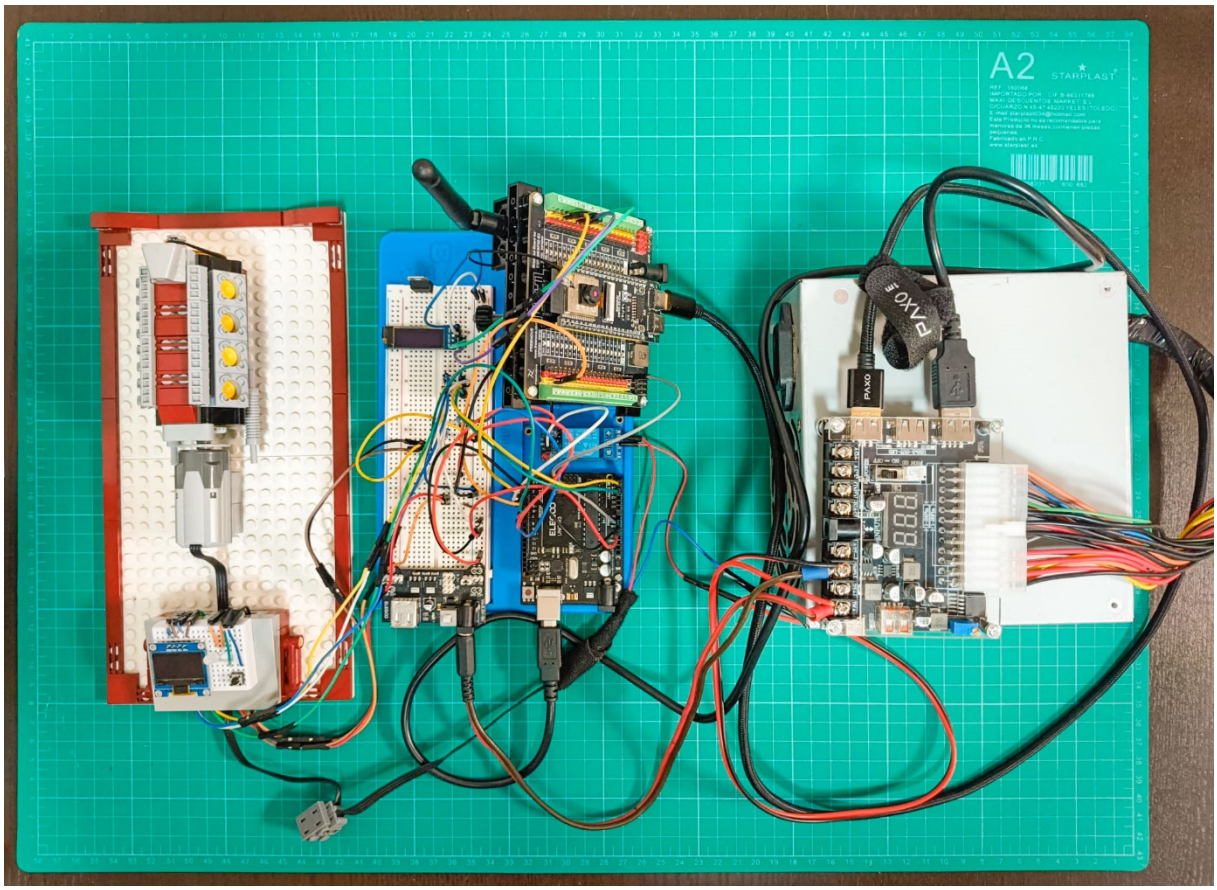


Figura 5.1: Vista general del prototipo físico del sistema antirrobo.

La base de la estructura de soporte se ha construido utilizando **piezas de construcción LEGO**. Esta elección proporciona una plataforma flexible y modular que permite adaptar fácilmente la disposición de los componentes y simular un habitáculo vehicular básico. Las piezas LEGO también se han empleado para crear soportes específicos y para la representación visual y funcional de un motor V8.

El montaje de los componentes electrónicos se realiza principalmente sobre **dos protoboards**. La protoboard de mayor tamaño actúa como un punto central de conexión para la distribución de masas (GND) y tensiones de alimentación (+5V, +3.3V) proporcionadas por una pequeña placa de distribución. En esta protoboard principal se integran resistencias limitadoras de corriente para los LEDs y se conecta una de las pantallas OLED (la de 128x32 píxeles), cuya función específica es mostrar la dirección IP que el módulo ESP32-S3-CAM obtiene al conectarse a la red WiFi. La segunda protoboard, de menor tamaño y montada sobre una base de LEGO (Figura 5.5), simula de forma básica un **tablero de instrumentos vehicular**. En ella se aloja la pantalla OLED de mayor resolución (128x64 píxeles), controlada por la placa Arduino

Uno para visualizar el estado del sistema y el motor; el LED blanco que se enciende cuando el motor simulado está en marcha; y un pulsador que emula el botón de arranque del vehículo.

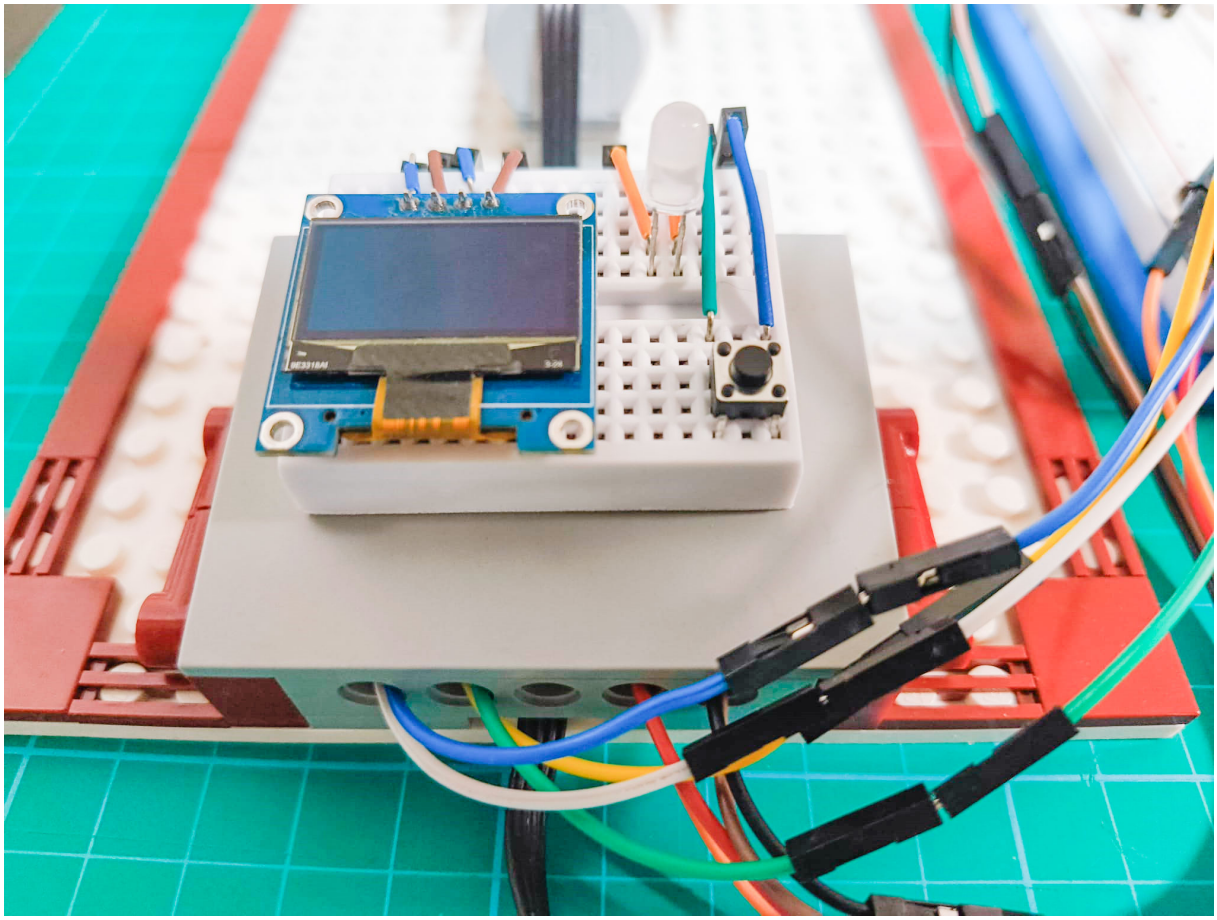


Figura 5.5: Simulación del tablero de instrumentos con pantalla OLED, LED de estado y pulsador de arranque

La **fuente de alimentación** del prototipo se gestiona a través de una **placa de ruptura (breakout board) para fuentes ATX** (Figura 5.2). Esta solución se adoptó debido a la necesidad de contar con múltiples tensiones de alimentación (3.3V, 5V, 12V y una salida variable configurada a 9V para el motor de LEGO) y la conveniencia de los puertos USB integrados para alimentar directamente las placas Arduino Uno y ESP32-S3-CAM. La elección de esta placa ATX se fundamentó también en las dificultades encontradas durante la fase de simulación en Proteus 8 al intentar obtener estas tensiones de forma estable a partir de una única fuente de 12V utilizando reguladores de tensión lineales como el LM7805, demostrando la robustez de la solución implementada.

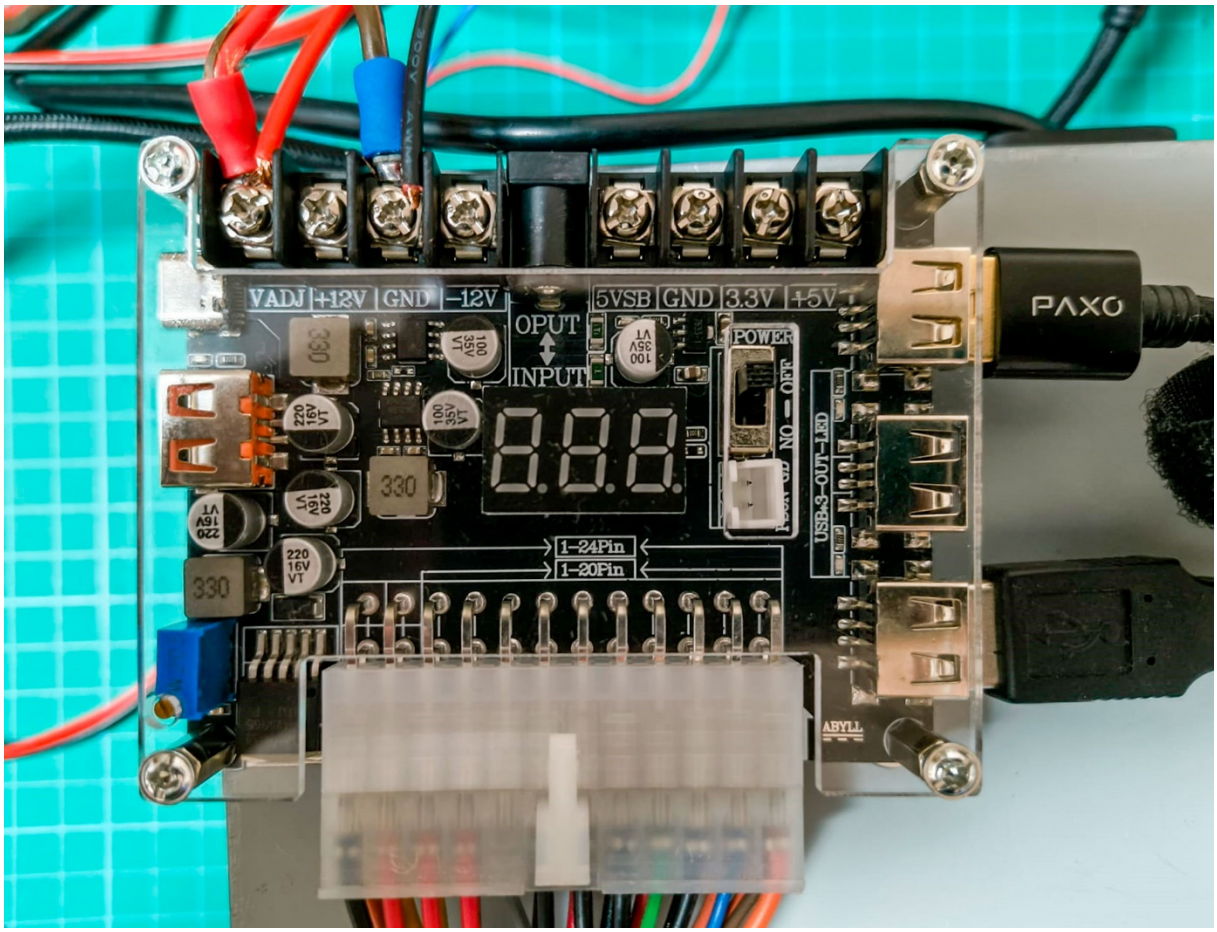


Figura 5.2: Placa de ruptura para fuente ATX, proporcionando múltiples tensiones de alimentación.

La interconexión eléctrica entre los componentes se realiza principalmente mediante **cables tipo *jumper* o latiguillos** (Figura 5.3), tanto macho-macho como macho-hembra. El uso de protoboards y la placa de ruptura específica para la ESP32-S3-CAM (Figura 5.4), que facilita el acceso a sus pines de conexión mediante bornas de tornillo, simplifican considerablemente el cableado y reducen la posibilidad de errores de conexión en comparación con el cableado directo o sobre protoboards sin adaptadores.

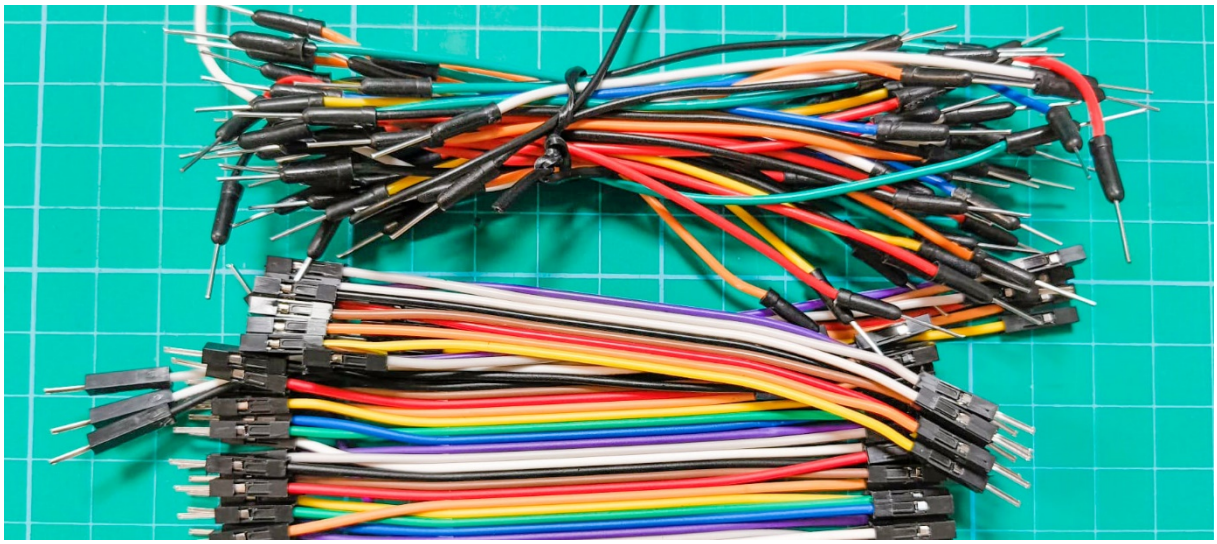


Figura 5.3: Cables tipo jumper utilizados para las interconexiones.

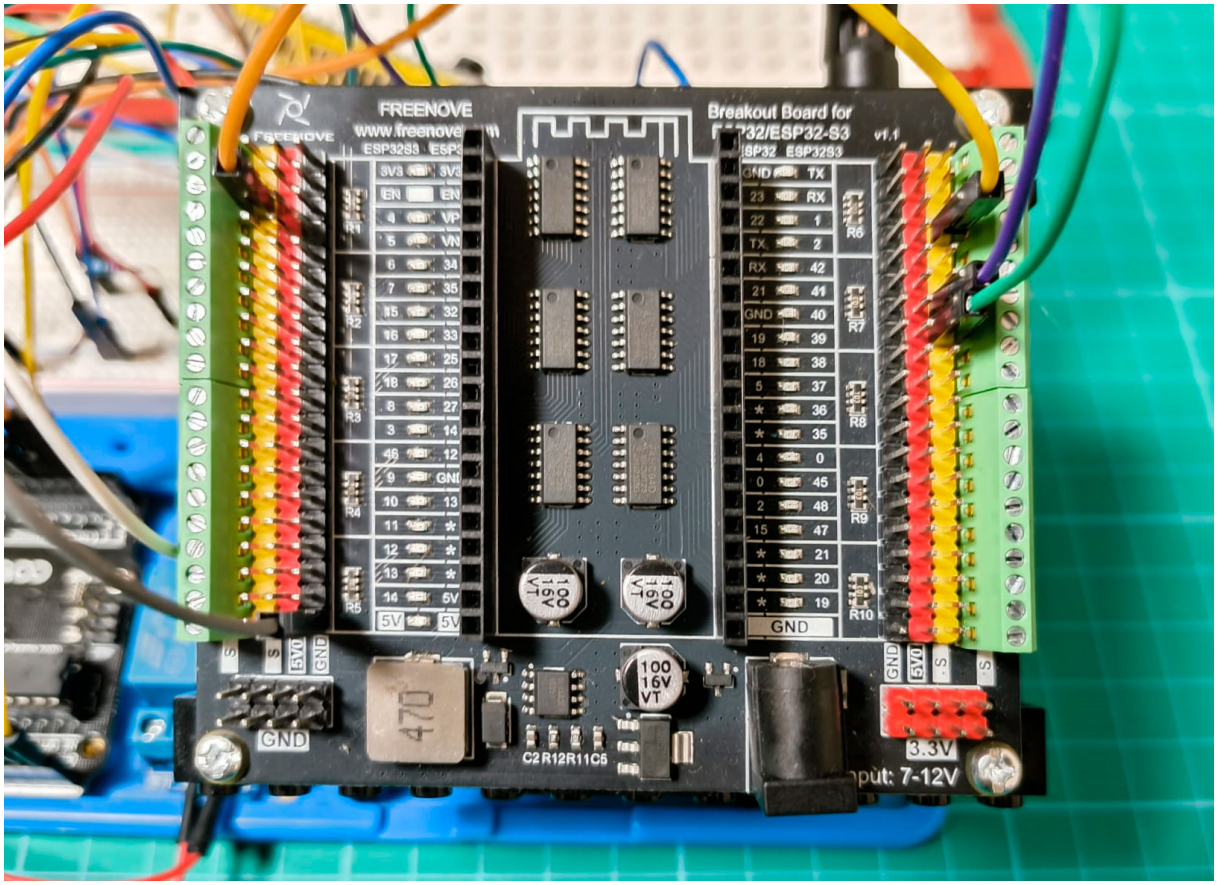
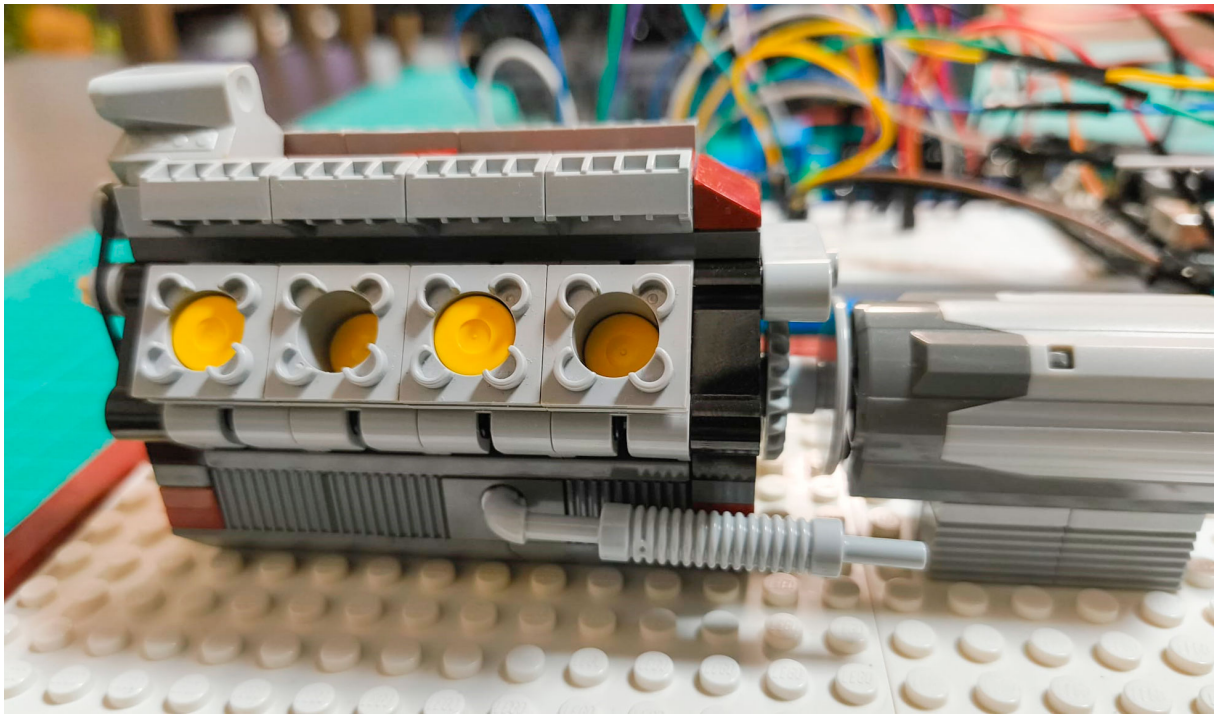


Figura 5.4: Placa de ruptura específica para la ESP32-S3-CAM, facilitando las conexiones.

La simulación del **sistema de propulsión** se logra mediante la combinación de un **motor eléctrico de LEGO** que mueve un **motor V8 funcional construido con piezas de LEGO** (Figura 5.6). El arranque de este conjunto motor se controla a través de un **módulo de relé** (identificado en la base azul entre las placas principales). Este módulo de relé, alimentado

con +5V desde la protoboard grande y controlado por el PIN 3 digital de la placa Arduino Uno, conmuta el suministro de 9V (proveniente de la salida variable de la placa ATX) hacia el motor eléctrico de LEGO. Así, el relé actúa como el interruptor principal que habilita o deshabilita la alimentación del motor eléctrico, simulando la función de un relé de arranque vehicular en un coche real. La capacidad de este módulo de relé para manejar tensiones de hasta 30V subraya su potencial aplicabilidad en sistemas eléctricos de automoción (típicamente 12V).



"Figura 5.6: Conjunto motor de LEGO: motor eléctrico y simulación funcional de un motor V8.

La realización física del prototipo, combinando componentes electrónicos estándar con una estructura de soporte adaptable y elementos de simulación, ha permitido validar la integración hardware-software del sistema en un entorno tangible y demostrar sus funcionalidades clave antes de considerar una implementación en un vehículo real.

5.2 Pruebas y Evaluación del Sistema.

La fase de pruebas y evaluación del prototipo se ha llevado a cabo para verificar la correcta integración y funcionamiento de los componentes hardware y software, así como para valorar el rendimiento del sistema desarrollado en un entorno controlado. Esta evaluación se ha enfocado en determinar si el prototipo cumple con los objetivos funcionales planteados,

principalmente en lo referente al reconocimiento facial y el control del sistema de arranque simulado.

Los principales **criterios de evaluación** aplicados durante las pruebas han sido:

- **Precisión del reconocimiento facial:** Se refiere a la capacidad del sistema (módulo ESP32-S3-CAM) para detectar y verificar correctamente la identidad de los usuarios registrados, así como para rechazar a usuarios no autorizados.
- **Tiempo de respuesta:** Medición del tiempo transcurrido desde que un rostro es presentado a la cámara hasta que el Arduino Uno activa la señal de "ARRANQUE AUTORIZADO" (PIN 2 en HIGH) o de "INTRUSO" (PIN 2 en LOW).
- **Fiabilidad de la comunicación:** Verificación de la estabilidad y correcta transmisión de mensajes entre la ESP32-S3-CAM y la aplicación móvil vía WebSockets, así como la correcta señalización digital entre la ESP32-S3-CAM y el Arduino Uno (Pines 2 y 4).
- **Funcionalidad del control de arranque:** Comprobación de que el Arduino Uno activa y desactiva correctamente el relé (simulando el motor) en función de las señales de la ESP32-S3-CAM (autorización/modo taller) y la pulsación del botón de arranque simulado.
- **Funcionalidad del "Modo Taller":** Verificación de que la activación de este modo desde la aplicación móvil permite el arranque simulado independientemente del reconocimiento facial, y que el estado del modo se persiste tras un reinicio de la ESP32-S3-CAM.
- **Usabilidad básica de la aplicación móvil:** Evaluación de la facilidad de uso de las funciones de control y visualización de la aplicación móvil en un nivel básico.

La **metodología de pruebas** ha consistido en la realización de una serie de ensayos en los que se ha interactuado con el prototipo físico y la aplicación móvil bajo diferentes condiciones:

- **Pruebas de reconocimiento:** Presentación de rostros de usuarios registrados al módulo ESP32-S3-CAM en diversas condiciones de iluminación y ángulos (dentro de

las capacidades del sensor y algoritmo). También se presentaron rostros de personas no registradas para evaluar la tasa de falsos positivos/negativos.

- **Pruebas de tiempo de respuesta:** Mediciones aproximadas del tiempo desde que el rostro aparece claramente en el *stream* de vídeo hasta que el LED verde en el tablero simulado se enciende (indicando "ARRANQUE AUTORIZADO").

- **Pruebas de comunicación:** Envío de comandos desde la aplicación móvil (activar/desactivar registro, activar/desactivar modo taller) y verificación de la respuesta correcta del sistema (cambio de estado de pines, recepción de mensajes de confirmación en la App).

- **Pruebas de control de arranque:** Pulsación del botón de arranque simulado en diferentes escenarios (con usuario autorizado presente, con intruso presente, con modo taller activado/desactivado) para verificar la activación/desactivación correcta del relé y el motor simulado.

- **Pruebas de persistencia:** Reinicio del módulo ESP32-S3-CAM con el Modo Taller activado para comprobar que este estado se mantiene.

Resultados y análisis de las pruebas:

Los resultados de las pruebas han permitido verificar la **funcionalidad principal** del sistema. La detección y el reconocimiento facial básico operan correctamente en condiciones de iluminación controlada y ángulos frontales, permitiendo la distinción entre usuarios registrados e intrusos. La tasa de reconocimiento satisfactoria se obtiene cuando las condiciones de captura son óptimas. El tiempo de respuesta para la detección y señalización de la autorización de arranque es aceptable para una primera versión del prototipo.

La **comunicación WebSockets** entre la aplicación móvil y la ESP32-S3-CAM ha demostrado ser fiable, permitiendo el control remoto del modo de registro y del Modo Taller, así como la recepción de mensajes de estado. La señalización digital entre la ESP32-S3-CAM y el Arduino Uno mediante los Pines 2 y 4 funciona según lo diseñado, activando el LED verde de "ARRANQUE AUTORIZADO" en el tablero simulado cuando corresponde (usuario reconocido o modo taller activo).

El **control del sistema de arranque simulado** por parte del Arduino Uno, basado en la lectura de los Pines 2 y 4 y la interacción con el pulsador, opera correctamente. El relé se activa para simular el encendido del motor de LEGO únicamente cuando se cumplen las condiciones de autorización o modo taller activo y se presiona el botón de arranque. El Modo Taller se activa y desactiva correctamente desde la App, y su estado se persiste en la memoria de la ESP32-S3-CAM. La simulación del tablero de instrumentos con las pantallas OLED y los LEDs proporciona *feedback* visual útil sobre el estado del sistema.

A pesar de estos resultados positivos, se identificaron áreas sensibles inherentes a la tecnología de reconocimiento facial, como la degradación del rendimiento ante variaciones significativas en la iluminación o ángulos extremos. Estos aspectos confirman la necesidad de optimizar el entorno de captura y los algoritmos en futuras iteraciones.

En conclusión, las pruebas realizadas confirman que el prototipo desarrollado implementa de manera funcional las características clave de un sistema antirrobo basado en reconocimiento facial, validando la arquitectura hardware-software y los mecanismos de comunicación y control diseñados.

5.3 Aspectos Éticos, Legales y de Seguridad (Consideraciones para una Implementación Comercial).

Si bien el prototipo desarrollado en este proyecto se concibe para fines académicos y de demostración, una eventual implementación comercial de un sistema antirrobo basado en reconocimiento facial en vehículos requeriría una atención rigurosa a los aspectos éticos, legales y de seguridad, particularmente en lo referente a la protección de datos biométricos.

Desde una perspectiva legal y ética, la captura y el procesamiento de imágenes faciales y la creación de plantillas biométricas de los usuarios implica el tratamiento de datos personales sensibles. En el contexto europeo, esto sitúa al sistema bajo el ámbito de aplicación del **Reglamento General de Protección de Datos (RGPD)**. Para cumplir con esta normativa, una implementación comercial debería adherirse a los siguientes principios y requisitos:

- **Obtención de Consentimiento Explícito:** Sería fundamental obtener el consentimiento libre, específico, informado e inequívoco de cada usuario antes de

recopilar y procesar sus datos faciales para fines de reconocimiento. El consentimiento debe ser demostrable.

- **Finalidad Específica y Limitación de Datos:** Los datos biométricos solo podrían ser utilizados para la finalidad específica de autenticación del usuario para el arranque del vehículo y la gestión de la seguridad. Se debería aplicar el principio de minimización de datos, recopilando únicamente la información estrictamente necesaria.

- **Información Transparente al Usuario:** Los usuarios deberían ser plenamente informados sobre qué datos se recopilan, cómo se utilizan, quién tiene acceso a ellos, durante cuánto tiempo se almacenan y cuáles son sus derechos (acceso, rectificación, supresión, limitación del tratamiento, portabilidad).

- **Medidas de Seguridad Robustas:** Dada la sensibilidad de los datos biométricos, sería imprescindible implementar medidas de seguridad técnicas y organizativas avanzadas para protegerlos contra el acceso no autorizado, la pérdida, la alteración o la divulgación. Esto incluiría el cifrado de las plantillas faciales, el almacenamiento seguro de la base de datos, la limitación estricta del acceso al personal autorizado y auditorías de seguridad periódicas.

- **Derechos de los Interesados:** Se deberían habilitar mecanismos claros y sencillos para que los usuarios puedan ejercer sus derechos bajo el RGPD, como solicitar acceso a sus datos, rectificarlos si son inexactos, solicitar su supresión o revocar su consentimiento.

- **Evaluación de Impacto sobre la Protección de Datos (EIPD):** Dada la naturaleza del tratamiento de datos biométricos a gran escala (en caso de comercialización), sería obligatorio realizar una EIPD para identificar y mitigar los riesgos potenciales para los derechos y libertades de los usuarios.

Adicionalmente, se deben considerar los **riesgos de sesgo algorítmico** inherentes a las tecnologías de reconocimiento facial. Los algoritmos pueden mostrar diferencias en su precisión al reconocer rostros de diferentes grupos demográficos (por raza, género, edad, etc.), lo que podría derivar en situaciones discriminatorias (por ejemplo, una menor tasa de

reconocimiento para ciertos usuarios). Para mitigar este riesgo en una implementación comercial, se requeriría:

- El entrenamiento de los modelos de reconocimiento con **conjuntos de datos amplios y diversos** que representen la variedad de usuarios potenciales.
- La selección y, si es posible, la adaptación de **algoritmos que hayan demostrado ser menos propensos a sesgos**.
- La realización de **pruebas de equidad** para evaluar el rendimiento del sistema en diferentes grupos demográficos.

Desde la perspectiva de la **seguridad del sistema en sí**, una implementación comercial debería contemplar medidas adicionales para proteger el sistema contra ataques, como:

- **Protección contra ataques de presentación (*presentation attacks*):** Implementación de técnicas (*liveness detection*) para detectar si el rostro presentado es real o una falsificación (foto, vídeo, máscara).
- **Seguridad de la comunicación:** Asegurar que la comunicación entre la ESP32-S3-CAM y la aplicación móvil, así como cualquier otra comunicación externa, esté debidamente cifrada.
- **Robustez del hardware:** Proteger físicamente el módulo de cámara y procesamiento para evitar manipulaciones o accesos no autorizados.

En conclusión, una implementación comercial de un sistema antirrobo vehicular basado en reconocimiento facial, si bien técnicamente viable, conlleva responsabilidades significativas en materia de protección de datos y seguridad. El cumplimiento normativo y la atención a los posibles sesgos algorítmicos serían pasos indispensables para garantizar un despliegue ético, legal y seguro.

6 CONCLUSIONES Y EVALUACIÓN PROPIA

6.1 Conclusiones.

Tras la finalización del desarrollo y las pruebas del prototipo de sistema antirrobo basado en reconocimiento facial, se han alcanzado las siguientes conclusiones fundamentales en relación con los objetivos definidos al inicio del proyecto:

Se ha logrado **diseñar e implementar un sistema funcional** que valida la viabilidad técnica de integrar reconocimiento facial y control vehicular. El prototipo demuestra la interacción efectiva entre el módulo ESP32-S3-CAM, la placa Arduino Uno y la aplicación móvil Android.

Se ha **investigado el estado del arte** de los sistemas antirrobo convencionales y se ha analizado la tecnología de reconocimiento facial. Esto ha permitido comprender las limitaciones de las soluciones existentes y justificar la propuesta de un sistema biométrico como alternativa más segura.

El prototipo desarrollado, utilizando hardware accesible como el ESP32-S3-CAM y Arduino Uno, verifica la posibilidad de **implementar una solución de reconocimiento facial a un coste razonable**. La comunicación por pines digitales y WebSockets demuestra una arquitectura de control y comunicación viable para este tipo de aplicaciones.

Se ha validado la **funcionalidad del sistema de control de arranque simulado**, confirmándose que el relé se activa correctamente en función del reconocimiento facial autorizado o la activación del modo taller, tal como se gestiona desde el *firmware* de Arduino Uno y la señalización digital de la ESP32-S3-CAM.

La **aplicación móvil** desarrollada proporciona una interfaz de usuario efectiva para **gestionar el sistema de forma remota**, permitiendo el control del modo de registro de usuarios y la activación del modo taller, además de visualizar el estado y el *stream* de vídeo.

Se ha demostrado la **aplicabilidad del "Modo Taller"** con persistencia de estado, asegurando que el vehículo pueda ser manipulado por personal no registrado cuando sea necesario (ITV, taller) de forma controlada desde la aplicación.

En términos generales, el proyecto demuestra que un sistema antirrobo vehicular basado en reconocimiento facial, utilizando tecnologías y plataformas de desarrollo accesibles, es **técnicamente realizable** y presenta un **potencial significativo** para mejorar la seguridad de los vehículos frente a las amenazas actuales. Los resultados de las pruebas confirman la operatividad de los componentes clave y los mecanismos de interacción diseñados.

6.2 Evaluación Propia.

La realización de este Proyecto Final de Ciclo ha supuesto una experiencia de aprendizaje sumamente valiosa y enriquecedora desde el punto de vista técnico y profesional. La integración de conocimientos de electrónica, programación y automoción en un sistema funcional ha permitido consolidar y aplicar los conceptos adquiridos durante el ciclo formativo.

Se considera que el proyecto ha resultado **útil e interesante** por abordar una problemática real en el sector automotriz, como es la seguridad vehicular frente al robo, y por explorar la aplicación de tecnologías innovadoras como el reconocimiento facial en este ámbito. El proceso de diseño y desarrollo del prototipo ha proporcionado una visión práctica de las distintas fases que componen un proyecto tecnológico, desde la concepción inicial hasta la implementación física y el desarrollo de software.

A nivel de aprendizaje, se ha profundizado significativamente en el manejo de microcontroladores como el ESP32-S3-CAM y Arduino Uno, así como en el desarrollo de *firmware* para la gestión de periféricos, comunicación y procesamiento. La implementación del reconocimiento facial, aunque basada en librerías existentes, ha requerido comprender los principios subyacentes y la optimización para hardware embebido. El desarrollo de la aplicación móvil en Android utilizando Kotlin y WebSockets ha supuesto una introducción práctica al desarrollo de software para dispositivos móviles y la comunicación en tiempo real. La fase de simulación en Proteus 8, a pesar de las dificultades iniciales con ciertos componentes, ha resaltado la importancia de la validación temprana del diseño hardware y software.

Durante el desarrollo del proyecto, se han enfrentado diversos **desafíos técnicos**, como la correcta configuración y comunicación entre la ESP32-S3-CAM y el Arduino Uno (resuelta

mediante señalización digital directa), la optimización del rendimiento del reconocimiento facial en el hardware seleccionado, y la gestión de la conexión WebSocket en la aplicación móvil. La superación de estas dificultades ha fomentado la capacidad de resolución de problemas y la perseverancia.

Si se tuviera la oportunidad de realizar el proyecto nuevamente o continuarlo, se **considerarían ciertos cambios o mejoras**. Se intentaría optimizar la robustez del reconocimiento facial en condiciones de iluminación y ángulos más variados. Se exploraría la posibilidad de configurar una dirección IP estática o implementar mDNS en la ESP32-S3-CAM para facilitar la conexión de la aplicación móvil en diferentes entornos de red. Además, se refinaría la interfaz de usuario de la aplicación móvil y se considerarían mecanismos de seguridad más avanzados para la gestión de la base de datos de usuarios faciales y la comunicación, pensando en una posible aplicación comercial.

En definitiva, este proyecto ha sido una experiencia integral que ha permitido aplicar los conocimientos adquiridos, desarrollar nuevas habilidades técnicas y enfrentarse a los retos propios del desarrollo de sistemas electrónicos y software, consolidando la formación como futuro profesional en el ámbito de la automoción.

6.3 Mejoras Futuras.

El prototipo desarrollado en este proyecto sienta las bases para un sistema antirrobo vehicular basado en reconocimiento facial. No obstante, se han identificado diversas áreas que podrían ser objeto de futuras mejoras y ampliaciones para incrementar su robustez, seguridad, funcionalidad y aplicabilidad en un entorno real.

Una línea de mejora prioritaria se centra en la **optimización de la funcionalidad de reconocimiento facial**. Esto implicaría investigar y aplicar algoritmos de detección y reconocimiento más avanzados y robustos, capaces de operar con mayor precisión bajo condiciones de iluminación variables, diferentes ángulos faciales, oclusiones parciales del rostro y distintas expresiones faciales. La implementación de técnicas de **detección de vida (*liveness detection*)** sería crucial para mitigar el riesgo de ataques de presentación utilizando fotografías o vídeos.

En relación con la **conectividad y usabilidad**, se podría abordar la **variabilidad de la dirección IP** asignada al módulo ESP32-S3-CAM en redes con DHCP. Una solución consistiría en implementar la configuración de una **dirección IP estática** en el *firmware* del ESP32 para redes predefinidas, o explorar el uso del protocolo **mDNS (Multicast DNS)** para permitir el acceso al módulo mediante un nombre de host (hostname) en lugar de la IP, facilitando la conexión desde la aplicación móvil en diferentes entornos WiFi.

La **aplicación móvil** podría ser objeto de un mayor **refinamiento de la interfaz de usuario (UI) y la experiencia de usuario (UX)** para hacerla más intuitiva y funcional. Se podrían añadir características como la visualización de un historial de eventos (intentos de acceso, activaciones/desactivaciones) o la gestión de múltiples usuarios registrados desde la propia aplicación, si bien la gestión de usuarios requeriría una consideración cuidadosa de la seguridad y privacidad de los datos.

La **seguridad** del sistema podría reforzarse mediante la implementación de **mecanismos de cifrado** para la comunicación entre la aplicación móvil y la ESP32-S3-CAM (por ejemplo, utilizando WebSockets Seguros - WSS) y para el almacenamiento de las plantillas faciales en la memoria del módulo. Se deberían considerar protocolos de autenticación robustos para el acceso a la aplicación y al sistema.

De cara a una potencial **implementación en un vehículo real**, se exploraría la integración directa del relé con el sistema eléctrico de arranque del vehículo, tal como se ha discutido, controlando la alimentación de la ECU o del relé de arranque principal. Esto requeriría un análisis detallado de los diagramas eléctricos del vehículo objetivo.

Finalmente, se podrían contemplar **funcionalidades adicionales** para aumentar el valor del sistema de seguridad. Esto podría incluir la integración con sistemas de geolocalización para rastrear la ubicación del vehículo en caso de robo, la adición de sensores para detectar intentos de intrusión (apertura de puertas, vibraciones) o la monitorización del estado del conductor (somnolencia, distracción) utilizando la cámara y algoritmos de visión artificial.

Estas mejoras futuras potenciarían el sistema desarrollado, acercándolo a una solución de seguridad vehicular más completa, robusta y preparada para un despliegue comercial.

7 GLOSARIO

- **Aprendizaje Automático (Machine Learning):** Subcampo de la inteligencia artificial que dota a las máquinas de la capacidad de aprender a partir de datos, identificar patrones y tomar decisiones con mínima intervención humana explícita.

- **Aprendizaje Profundo (Deep Learning):** Subcampo del aprendizaje automático que utiliza redes neuronales artificiales con múltiples capas (profundas) para modelar abstracciones de alto nivel en los datos. Es fundamental en el procesamiento de imágenes y el reconocimiento facial.

- **Arduino Uno:** Placa de microcontrolador de código abierto basada en el microchip ATmega328P. Ampliamente utilizada para prototipado y control de sistemas electrónicos básicos.

- **ATX:** Estándar de diseño de fuentes de alimentación utilizado comúnmente en ordenadores personales. Proporciona múltiples tensiones de salida (+3.3V, +5V, +12V, etc.).

- **Biometría:** Técnica de identificación o verificación de la identidad de una persona basada en sus características físicas o de comportamiento únicas e intrínsecas, como el rostro, la huella dactilar o la voz.

- **Breakout Board:** Placa adaptadora que facilita la conexión de un chip o módulo electrónico (como la ESP32-S3-CAM) a una protoboard o a otros componentes mediante pines o bornas accesibles.

- **Cable Jumper (Latiguillo):** Cable conductor con conectores en sus extremos utilizado para realizar interconexiones temporales o de prototipado en protoboards o entre diferentes módulos electrónicos.

- **Datos Biométricos:** Datos personales obtenidos a partir de un tratamiento técnico específico, relativos a las características físicas, fisiológicas o de comportamiento de una persona física que permitan o confirmen la identificación única de dicha persona (ej. imagen facial, plantilla facial).

- **DHCP (Dynamic Host Configuration Protocol):** Protocolo de red que permite a un servidor asignar automáticamente direcciones IP y otros parámetros de configuración de red a los dispositivos conectados.

- **ECU (Engine Control Unit):** Unidad de Control del Motor. Ordenador integrado en el vehículo que gestiona y controla diversos aspectos del funcionamiento del motor.

- **ESP32-S3-CAM:** Módulo basado en el microcontrolador ESP32-S3 de Espressif, que integra una cámara, conectividad WiFi y Bluetooth, y memoria, diseñado para aplicaciones de visión artificial y IoT.

- **Firmware:** Software de bajo nivel incrustado en un dispositivo electrónico (como un microcontrolador) que proporciona las instrucciones esenciales para su funcionamiento y control del hardware.

- **GPIO (General Purpose Input/Output):** Pines en un microcontrolador o circuito integrado que pueden ser configurados dinámicamente como entradas o salidas digitales, o a veces analógicas.

- **IA (Inteligencia Artificial):** Campo de la informática dedicado al desarrollo de sistemas que pueden realizar tareas que normalmente requieren inteligencia humana, como el aprendizaje, la percepción, la toma de decisiones y el reconocimiento de patrones.

- **Modo Taller:** Funcionalidad específica del sistema antirrobo que, al ser activada, permite el arranque del vehículo independientemente del reconocimiento facial, facilitando su manipulación por personal no registrado (ej. en un taller o ITV).

- **mDNS (Multicast DNS):** Protocolo de red que permite a los dispositivos anunciarse y descubrir otros servicios en una red local sin necesidad de un servidor DNS centralizado, utilizando nombres de host legibles en lugar de direcciones IP.

- **NVS (Non-Volatile Storage):** Almacenamiento no volátil. Tipo de memoria en un dispositivo que conserva su contenido incluso cuando se interrumpe la alimentación, utilizada para guardar configuraciones o datos persistentes.

- **OLED (Organic Light-Emitting Diode):** Diodo orgánico emisor de luz. Tecnología de visualización que utiliza una capa de material orgánico que emite luz cuando se le aplica una corriente eléctrica, permitiendo pantallas delgadas y de bajo consumo.

- **Proteus 8:** Software de simulación de circuitos electrónicos y microcontroladores, utilizado para diseñar y verificar el funcionamiento de sistemas antes de su implementación física.

- **Protoboard:** Placa de pruebas o *breadboard*. Dispositivo con orificios interconectados eléctricamente que permite montar y prototipar circuitos electrónicos de forma temporal sin necesidad de soldar.

- **Reconocimiento Facial:** Tecnología biométrica que identifica o verifica a una persona comparando patrones faciales extraídos de una imagen con una base de datos.

- **Redes Neuronales Convolucionales (CNNs):** Tipo de red neuronal artificial, muy eficaz para procesar datos con una estructura de rejilla, como imágenes. Se utilizan extensivamente en visión artificial para detección y reconocimiento.

- **Relé:** Interruptor electromagnético que permite controlar un circuito de alta potencia o voltaje mediante una señal de control de baja potencia o voltaje.

- **RGPD (Reglamento General de Protección de Datos):** Normativa europea relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.

- **UI (User Interface):** Interfaz de Usuario. Medio a través del cual un usuario interactúa con un sistema informático o dispositivo electrónico.

- **UX (User Experience):** Experiencia de Usuario. Conjunto de percepciones y respuestas de una persona resultantes del uso y/o uso anticipado de un producto, sistema o servicio, incluyendo sus emociones, creencias, preferencias y comportamientos.

- **WebSocket:** Protocolo de comunicación basado en TCP que proporciona canales de comunicación bidireccionales (full-duplex) sobre una única conexión, permitiendo la

comunicación en tiempo real entre un cliente (ej. navegador web, app móvil) y un servidor.

8 BIBLIOGRAFÍA

La elaboración del presente Proyecto Final de Ciclo ha requerido la consulta y aplicación de conocimientos provenientes de diversas fuentes técnicas y de referencia en los campos de la electrónica, la programación, la automoción y la inteligencia artificial. Si bien gran parte de la implementación práctica se ha guiado por recursos de carácter tutorial y demostrativo, la base conceptual y técnica se fundamenta en la documentación oficial y guías especializadas. A continuación, se listan las referencias consultadas o que representan las fuentes de información técnica utilizadas:

Documentación Técnica del Hardware:

Espressif Systems. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *ESP32-S3 Datasheet*. [Enlace o descripción de dónde encontrarlo]. [Descripción: Documentación oficial con las especificaciones técnicas del microcontrolador ESP32-S3].

Arduino. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Arduino Uno Documentation*. [Enlace o descripción de dónde encontrarlo]. [Descripción: Referencia técnica sobre la placa Arduino Uno y su microcontrolador ATmega328P].

[Fabricante del Módulo de Relé]. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Especificaciones Técnicas del Módulo de Relé de 5V*. [Enlace o descripción de dónde encontrarlo]. [Descripción: Hoja de datos con las características de operación y conexión del módulo de relé].

[Fabricante de las Pantallas OLED]. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Datasheet del Display OLED SSD1306*. [Enlace o descripción de dónde encontrarlo]. [Descripción: Documentación técnica del controlador de las pantallas OLED utilizadas].

Documentación de Software y Librerías:

Arduino IDE. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Arduino Reference*. [Enlace a reference.arduino.cc o descripción]. [Descripción: Referencia oficial del lenguaje de programación de Arduino y sus funciones principales].

Espressif Systems. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *ESP-IDF Programming Guide*. [Enlace o descripción]. [Descripción: Guía oficial de programación del framework subyacente para ESP32, relevante para configuraciones avanzadas o librerías específicas].

Kotlin. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Kotlin Documentation*. [Enlace a kotlinlang.org/docs/ o descripción]. [Descripción: Referencia del lenguaje de programación Kotlin utilizado en la aplicación Android].

Android Developers. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Android Studio Documentation*. [Enlace a developer.android.com/studio/ o descripción]. [Descripción: Documentación oficial del entorno de desarrollo y las APIs de Android].

Square, Inc. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *OkHttp Documentation*. [Enlace a square.github.io/okhttp/ o descripción]. [Descripción: Referencia de la librería OkHttp utilizada para la gestión de conexiones HTTP y WebSockets en Android].

[Librería de Reconocimiento Facial utilizada en ESP32]. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Documentación de la librería X para reconocimiento facial en ESP32*. [Enlace o descripción]. [Descripción: Información técnica sobre la librería o framework (ej. TensorFlow Lite Micro, librerías de terceros) utilizada para la implementación del reconocimiento facial en el ESP32-S3].

Librerías para pantallas OLED: Adafruit. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Adafruit SSD1306 Library / Adafruit GFX Library Documentation*. [Enlace a learn.adafruit.com o descripción]. [Descripción: Documentación sobre las librerías de Arduino utilizadas para el control de las pantallas OLED].

Protocolos y Estándares:

IETF. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *RFC 6455 - The WebSocket Protocol*. [Enlace a tools.ietf.org/html/rfc6455 o descripción]. [Descripción: Especificación técnica del protocolo WebSocket].

Parlamento Europeo y Consejo. (2016). *Reglamento (UE) 2016/679 ... relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (RGPD)*. [Enlace a eur-lex.europa.eu o descripción]. [Descripción: Texto oficial del Reglamento General de Protección de Datos].

Recursos Generales y Conceptuales:

[Plataforma/Tutorial General sobre ESP32-CAM]. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Guía de inicio o tutorial avanzado sobre el uso de la ESP32-CAM*. [Enlace a un sitio web plausible como randomnerdtutorials.com, learn.sparkfun.com, etc. o descripción general]. [Descripción: Información general sobre el uso y configuración de la ESP32-CAM para aplicaciones de visión o web server].

[Recurso General sobre Reconocimiento Facial o IA]. (Fecha de consulta: [Indicar fecha aproximada de consulta]). *Artículo o introducción sobre los fundamentos del reconocimiento facial o el aprendizaje automático*. [Enlace a una fuente educativa o técnica plausible]. [Descripción: Información conceptual sobre las tecnologías de reconocimiento facial y su aplicación].

Repositorio del Proyecto (Código Fuente):

[Su Nombre de Usuario en GitHub]. (Fecha de consulta: [Indicar fecha de última modificación o consulta relevante]). *FaT14*. [Enlace a https://github.com/Esmecarbea/FaT14.git]. [Descripción: Repositorio que contiene el código fuente del firmware y la aplicación móvil desarrollados para el proyecto].

9 ANEXOS