INSTITUTO TECNOLÓGICO SUPERIOR DEL ESTADO DE HIDALGO

ESP32-CAM MODELO 3D

1 A

2025-2029

MTRO. SAÚL ISAÍ SOTO ORTIZ

CERÓN SÁNCHEZ VICTOR
RENDÓN GARCÍA HUGO EMMANUEL
JOSE MANUEL MENDOZA RODRIGUEZ

TIC´S

INTRODUCCIÓN A LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES

FECHA DE ENTREGA: 03-12-25

LABORATORIO-CISCO

ÍNDICE

OCTUBRE 30

INTRODUCCIÓN

Este informe justifica la necesidad de sistemas de reconocimiento visual de bajo costo. El objetivo fue desarrollar un prototipo con ESP32-CAM capaz de detectar y diferenciar dos objetos: una pila y una pieza LEGO. Como principales resultados, el sistema logró identificar ambos elementos exitosamente, mostrando el objeto detectado en una pantalla OLED. Además, activa un LED específico para cada objeto y emite una alerta sonora mediante un buzzer al confirmar la detección. Se concluye que el ESP32-CAM es viable para tareas de visión artificial simples con múltiples salidas.

INTRODUCCIÓN

El Reconocimiento de Objetos es una rama clave de la visión artificial. Su objetivo es identificar y clasificar elementos en una imagen o video, tarea que se realiza cada vez más con modelos de aprendizaje automático (Machine Learning) entrenados para reconocer patrones visuales.

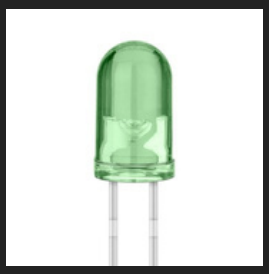
Históricamente, estas tareas requerían gran poder computacional. Sin embargo, la microelectrónica ha democratizado su acceso. El módulo ESP32-CAM es un ejemplo clave: es un microcontrolador de bajo costo que integra un procesador, conectividad Wi-Fi y una interfaz de cámara. Esta combinación lo hace ideal para el "Edge Computing", permitiendo procesar imágenes directamente en el dispositivo.

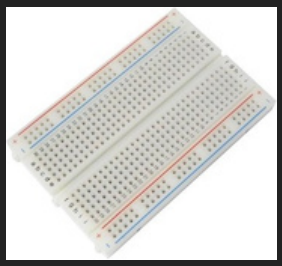
El "qué" de este informe es la unión de estos conceptos. Se presenta un prototipo basado en el ESP32-CAM para demostrar su viabilidad en una tarea específica: la detección y diferenciación de dos objetos, una pila y una pieza LEGO. El sistema identifica el objeto y comunica el resultado al usuario mediante una pantalla OLED, indicadores LED diferenciados y una alerta sonora (buzzer).

Las siguientes secciones servirán como guía del proyecto, detallando la metodología, la configuración del hardware y software, los resultados de las pruebas y las conclusiones finales sobre el rendimiento del sistema y su trabajo futuro.

MATERIALES FISICOS

- Esp32-Cam
- Modulo Esp32-Cam
- Lego
- Pila
- Jumpers
- Leds
- Buzzer
- Protoboard
- Cautín
- Pasta para soldar
- Soldadura
- Pantalla Oled
- Resistencias
- Cable micro-USB



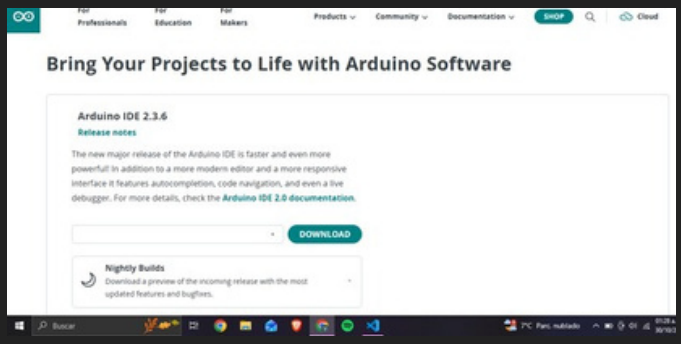


MATERIALES DIGITALES

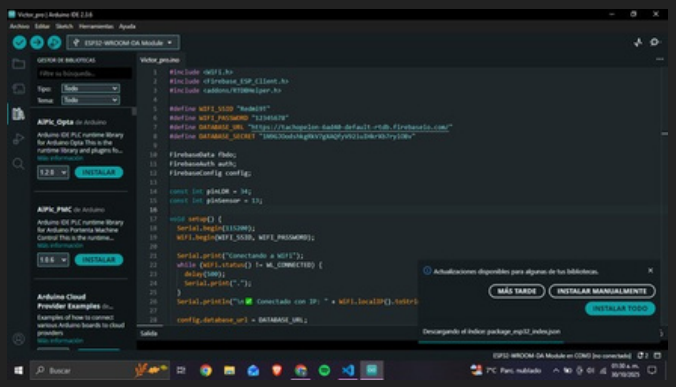
- Arduino IDE
- Librerias
- Edge-Impulse

PROCEDIMIENTO

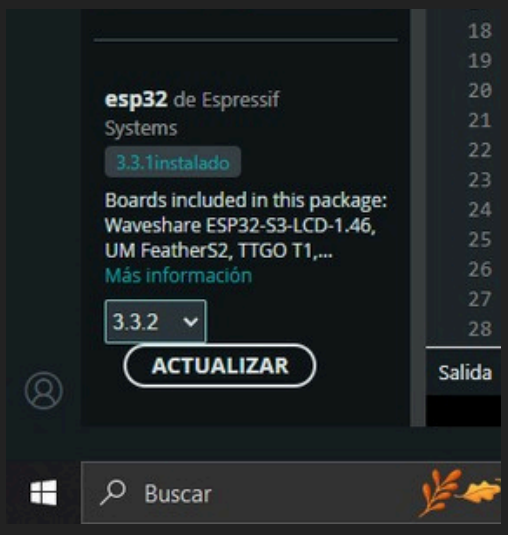
El primer paso es configurar el entorno de desarrollo de Arduino IDE en Windows para que reconozca y pueda programar la placa ESP32-CAM.

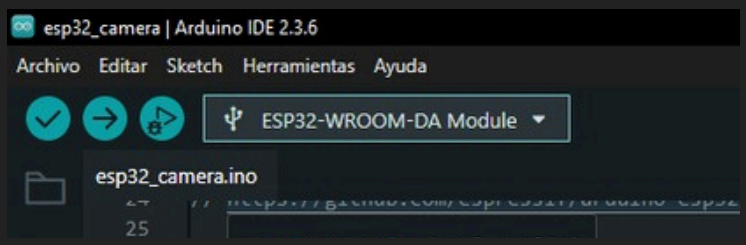


1.Instalación de Arduino IDE: Descargar e instalar la última versión de Arduino IDE (versión 1.8.x o 2.x) desde el sitio web oficial.



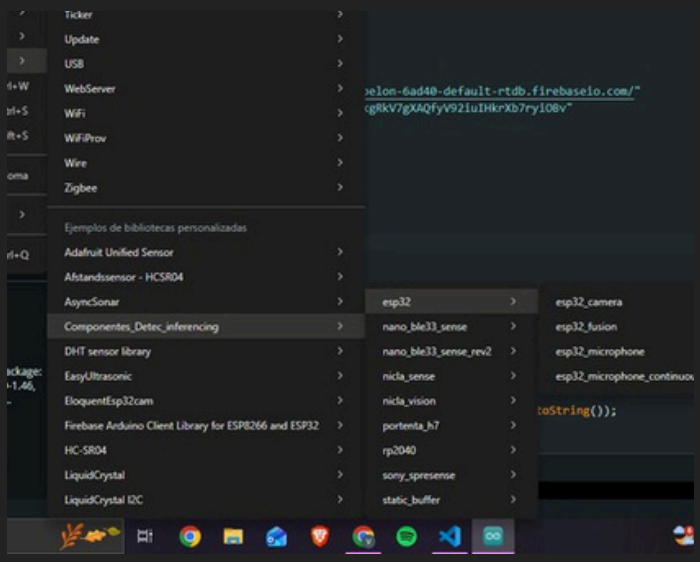
2. Añadir el Gestor de Placas ESP32:
   - Abrir Archivo > Preferencias.
   - En "Gestor de URLs Adicionales de Tarjetas", añadir la siguiente URL: https://raw.githubusercontent.com/espressif/arduino-esp32/master/package_esp32_index.json
3. Instalar las Placas ESP32:
   - Abrir Herramientas > Placa > Gestor de Tarjetas...
   - Buscar "esp32" e instalar el paquete "esp32 by Espressif Systems".
   -





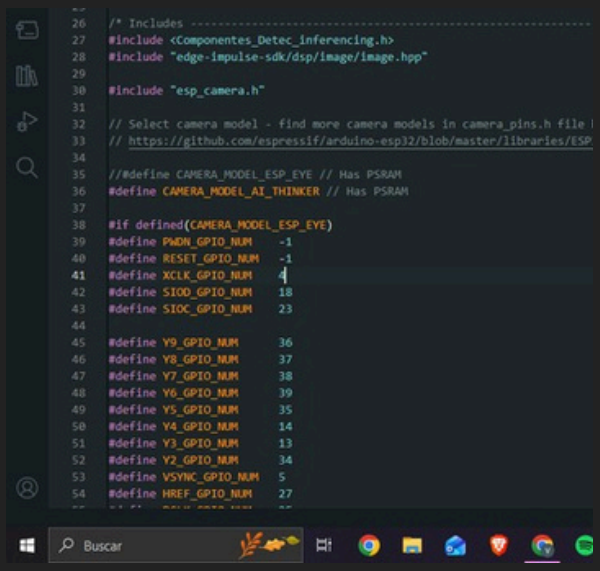
1.Configuración de la Placa en Arduino IDE:

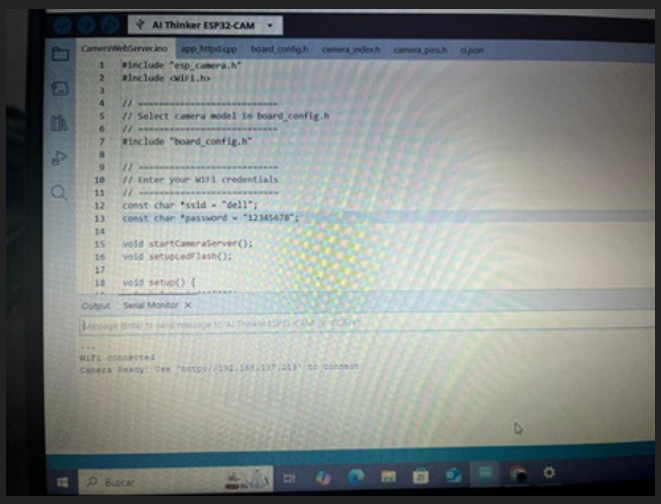
- Conectar el ESP32-CAM a la PC mediante un programador USB-a-TTL (como el FTDI232).
  Importante: Asegurarse de poner el pin GPIO0 a GND antes de subir el código para entrar en modo "Flash".
- En Herramientas > Placa, seleccionar "ESP32 Wrover Module" (o "AI Thinker ESP32-CAM" si aparece).
- Seleccionar el Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)".
- Seleccionar el puerto COM correcto.

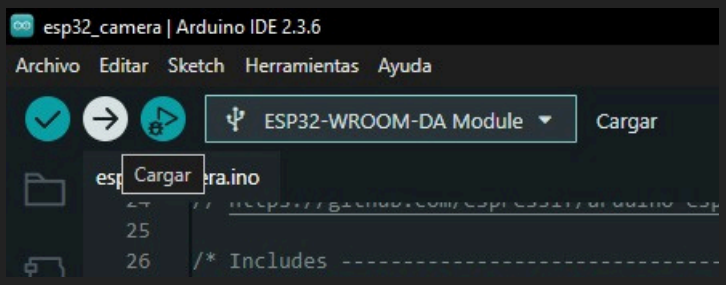


2. Prueba de Verificación (CameraWebServer):

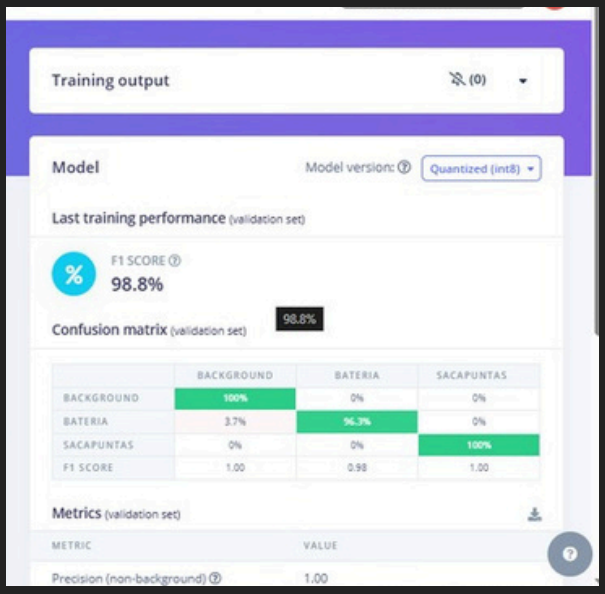
- Abrir Archivo > Ejemplos > ESP32 > Camera > CameraWebServer.
- Descomentar el modelo de placa CAMERA_MODEL_AI_THINKER.
- Introducir las credenciales (SSID y contraseña) de la red Wi-Fi.
- Subir el código (con GPIO0 en GND).
- Quitar el pin GPIO0 de GND y presionar el botón de Reset (RST) en la placa.
- Abrir el Monitor Serial (a 115200 baudios) para ver la dirección IP asignada.
- Ingresar la IP en un navegador web para verificar el streaming de video.

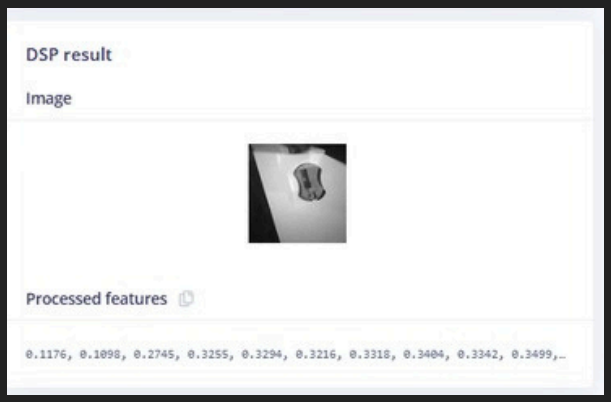


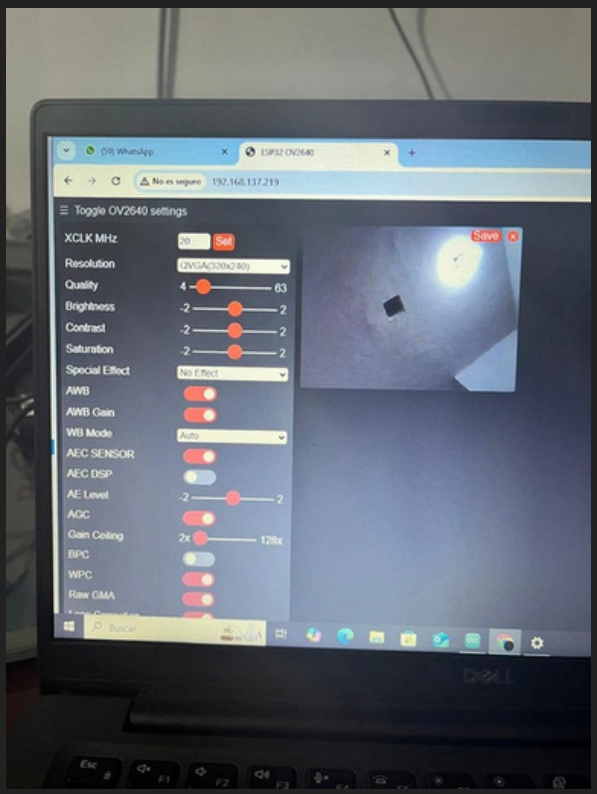




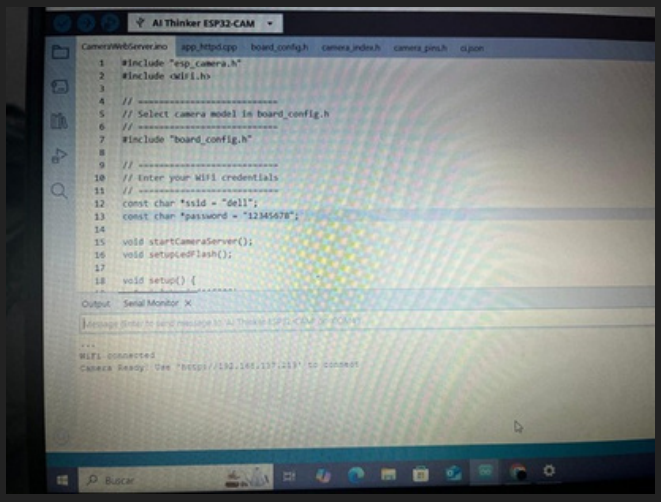
1. l proceso incluyó:
2. Adquisición de Datos: Se recopilaron 100-150 fotos por clase usando un móvil.
3. Diseño del Impulso: Se configuró un pipeline optimizado para baja potencia, estableciendo las imágenes a 96x96 píxeles en escala de grises.
4. Aprendizaje: Se empleó Transfer Learning (usando MobileNetV2 0.35) como bloque de aprendizaje.
5. Entrenamiento: El modelo se entrenó ajustando parámetros hasta lograr una precisión superior al 95%.
6. Despliegue: Finalmente, el modelo completo se exportó como una Librería de Arduino (archivo .zip) lista para ser integrada en un microcontrolador.



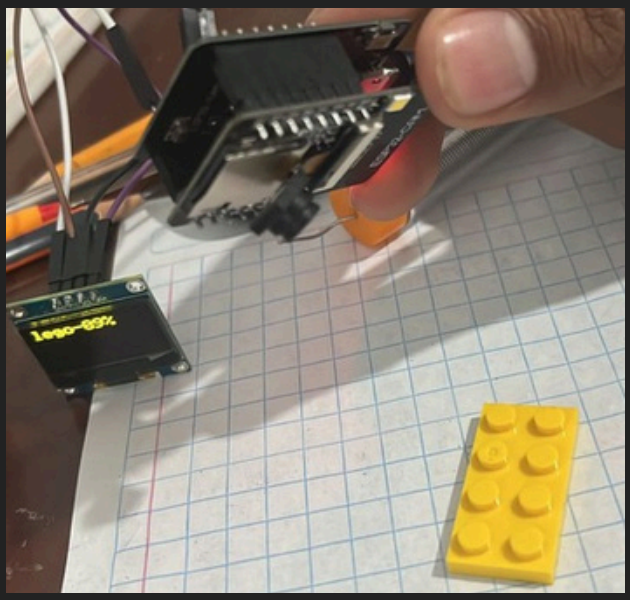




1.Monitor Serial: Sirve para depuración. Confirma que la cámara y el modelo se iniciaron correctamente. En cada ciclo, imprime el tiempo que tardó la inferencia (ej. 250ms) y los puntajes de confianza para todas las clases (ej. pila: 0.85).



2. Pantalla OLED: Una pantalla SSD1306 (I2C conectada a GPIO 21 y 22) muestra la etiqueta del objeto con mayor confianza (ej. "OBJETO: PILA"), pero solo si esa confianza supera un umbral del 80%. Si ningún objeto supera el umbral, muestra "...Buscando...".



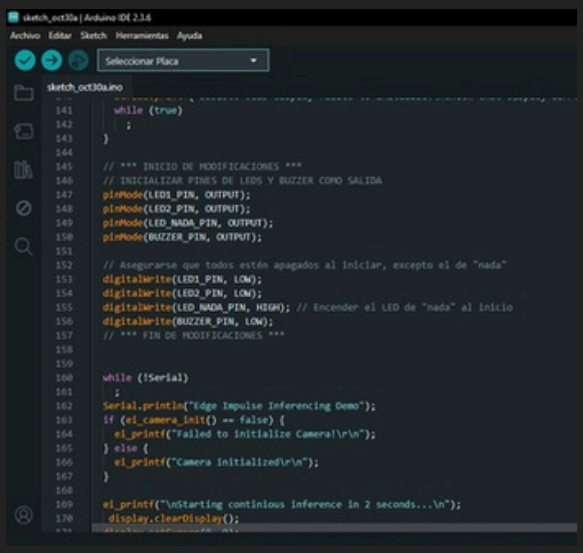
3. Alertas Físicas (LEDs y Buzzer):

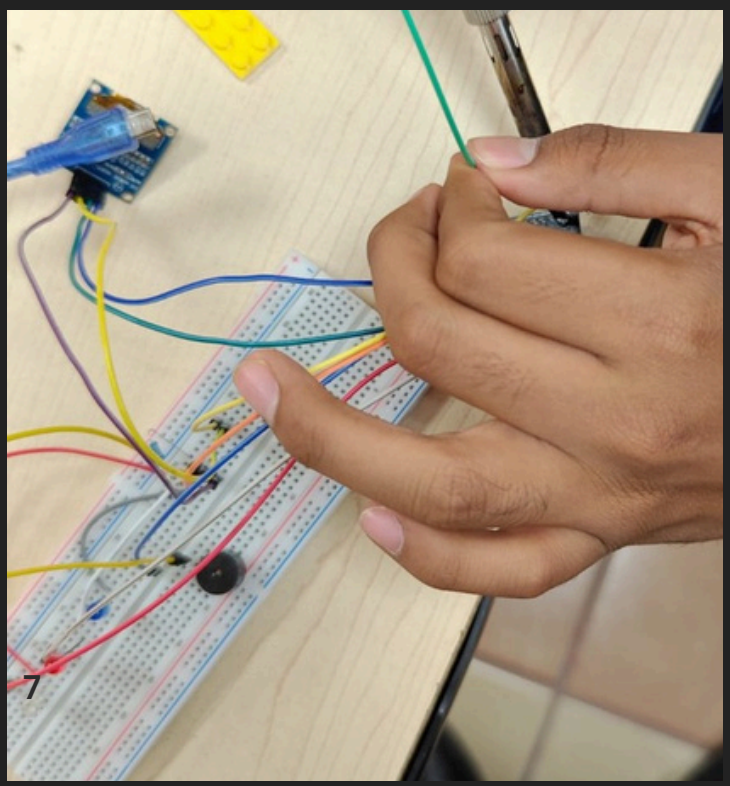
4.Un LED Rojo (GPIO 12) se enciende si detecta "pila".

5.Un LED Verde (GPIO 13) se enciende si detecta "lego".

6. Si detecta "fondo" o la confianza es baja, ambos LEDs se apagan.

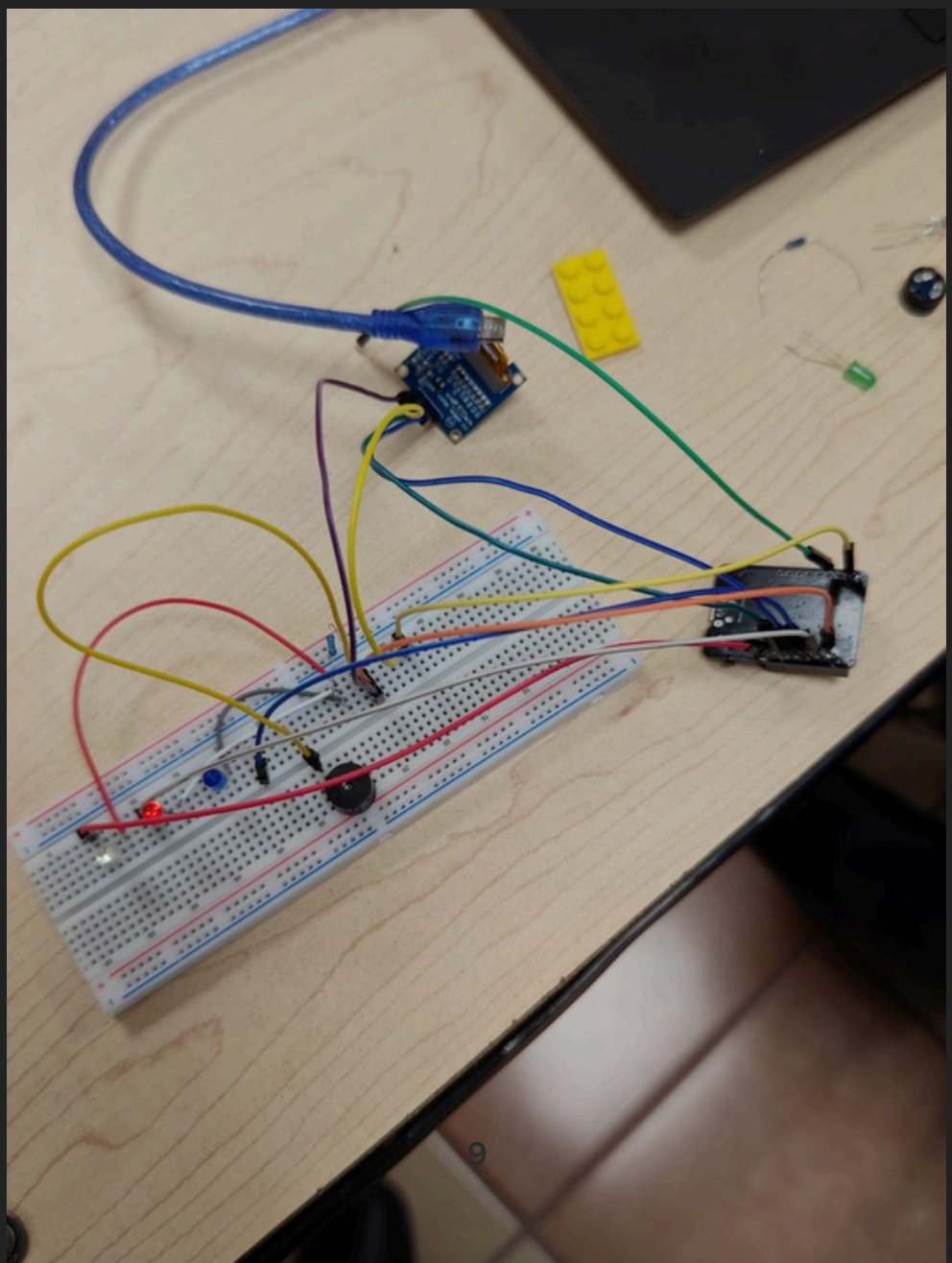
7.Un Buzzer (GPIO 14) emite un sonido corto como alerta auditiva cada vez que se identifica "pila" o "lego".



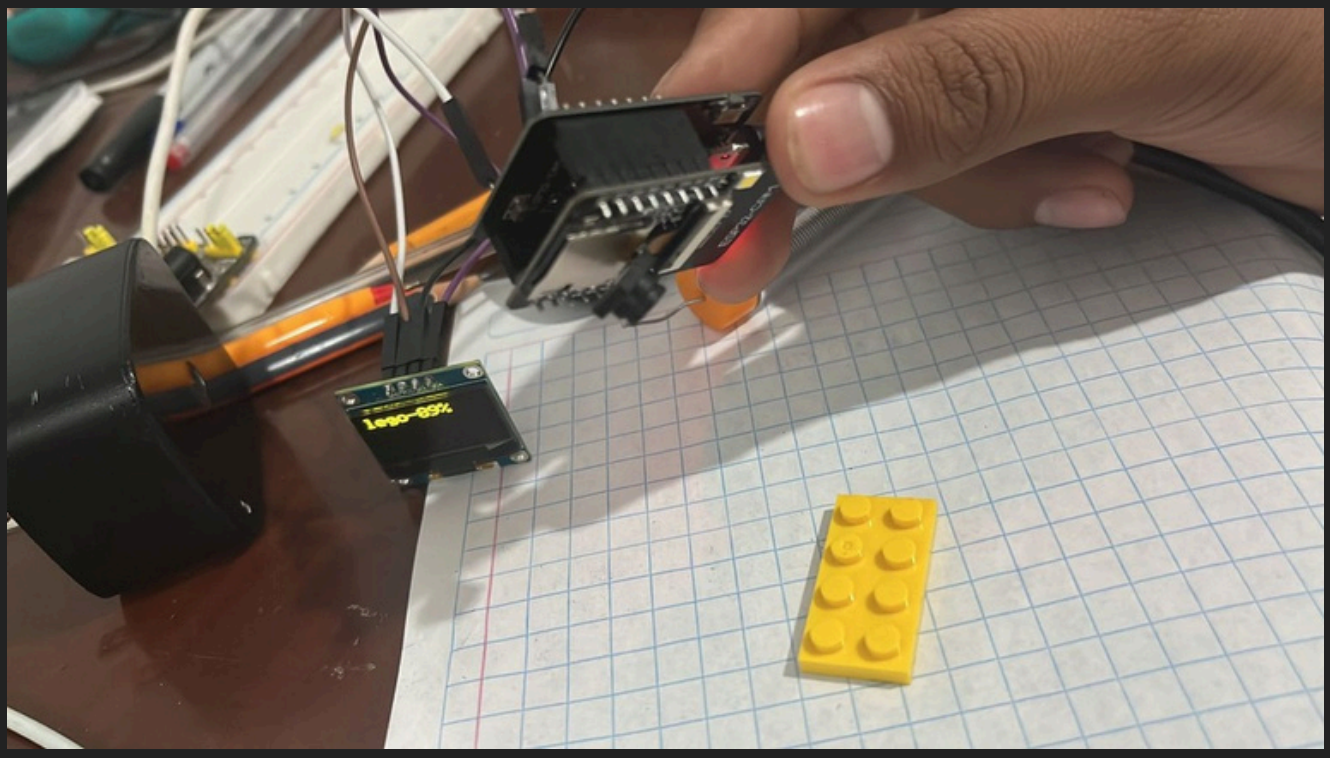


RESULTADOS

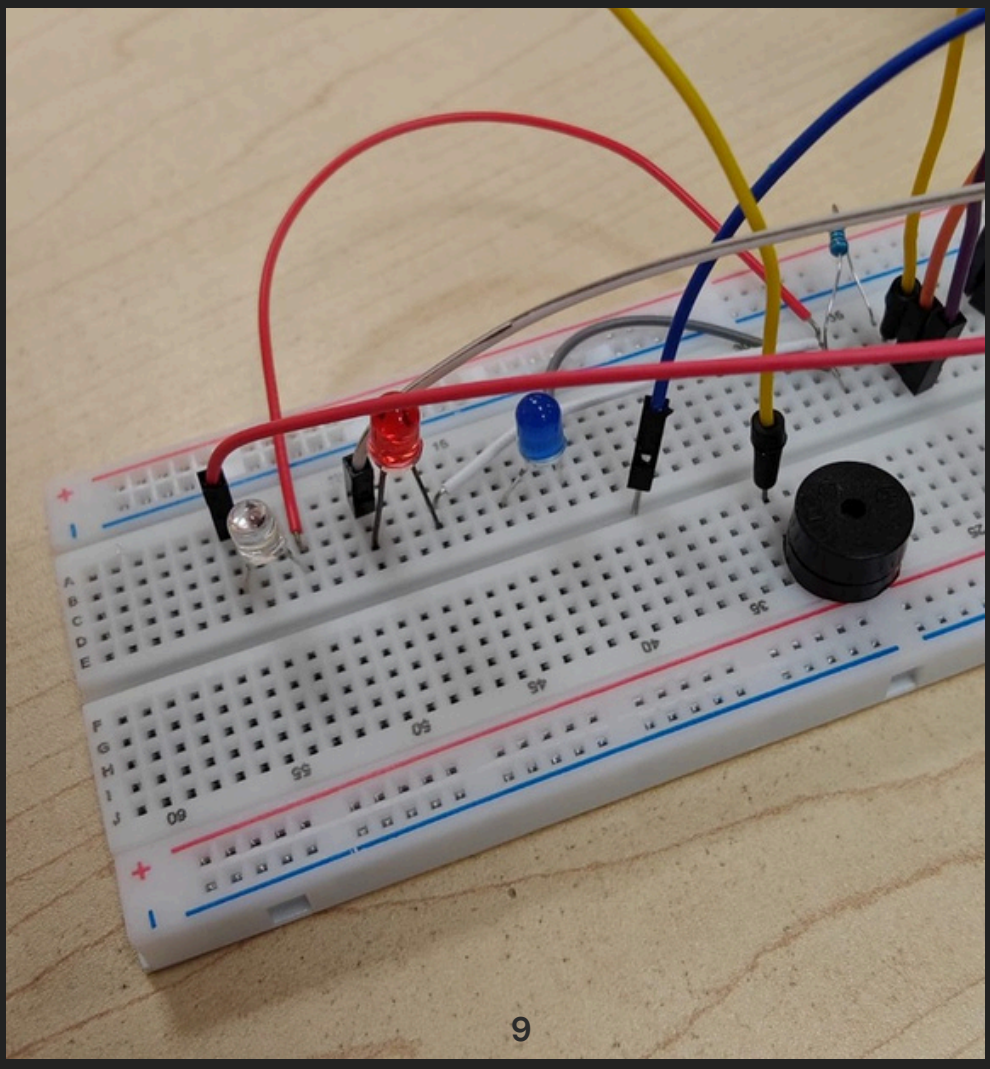
El prototipo funcional del laboratorio utiliza una ESP32-CAM (modelo AI Thinker) para la detección de dos objetos específicos: una pila y un LED. Los resultados de la detección se manifiestan activando simultáneamente un LED indicador (conectado al GPIO 12), un buzzer activo (conectado al GPIO 13) y mostrando un mensaje de confirmación en una pantalla OLED SSD1306 (conectada vía I2C a los pines GPIO 15 y GPIO 14). La configuración del software se realizó en el entorno de Arduino IDE, requiriendo las librerías esp_camera.h para el sensor de imagen y Adafruit_SSD1306.h junto con Adafruit_GFX.h para la pantalla.



El funcionamiento se articula en varios segmentos de código cruciales. En el setup(), es fundamental inicializar la cámara con una configuración que permita el análisis de color; para esto, se seleccionó config.pixel_format = PIXFORMAT_RGB565 y config.frame_size = FRAMESIZE_QVGA (320x240). El formato RGB565 es vital, ya que permite la lectura directa de los componentes de color, a diferencia del formato JPEG que está comprimido. En esta misma sección, se inicializa la comunicación I2C con la OLED y se configuran los pines del LED y el buzzer como OUTPUT.



Los resultados se obtienen cuando, de vuelta en el loop(), se comprueba esta variable. Si objetoDetectado es verdadero, se invoca a la función activarAlerta(), la cual ejecuta digitalWrite(HIGH) en los pines del LED y el buzzer, y utiliza display.println() para mostrar el nombre del objeto en la OLED. Si es falso, desactivarAlerta() apaga los actuadores y limpia la pantalla. Finalmente, es imperativo llamar a esp_camera_fb_return() al final del loop() para liberar la memoria del frame buffer, evitando así una fuga de memoria que bloquearía la ESP32 en segundos. La efectividad del prototipo depende directamente de la calibración precisa de estos rangos de color RGB y los umbrales de píxeles, los cuales se ajustaron empíricamente según la iluminación del laboratorio.



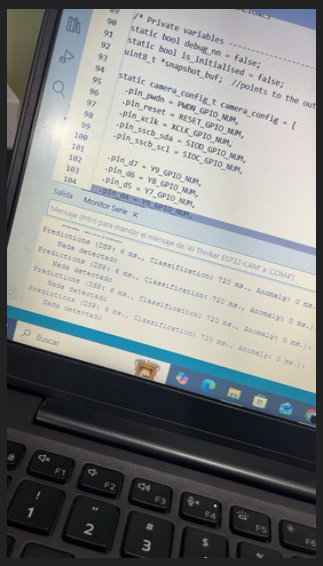
PROCEDIMIENTO DE FABRICACIÓN Y DISEÑO (CARCASA)

Antes de la programación, se procedió al diseño de un contenedor para proteger el hardware y mejorar la estabilidad de la cámara.

Modelado 3D

Medición: Se tomaron medidas del módulo ESP32-CAM utilizando un vernier, considerando el espacio para los pines inferiores.

Diseño CAD: Se modeló una caja rectangular con una tolerancia de 0.8mm. Se realizaron extrusiones de corte para crear la apertura del lente de la cámara en la cara superior y una ranura lateral para la salida de cables.



Impresión y Post-procesado

La pieza se imprimió en PLA color negro para evitar reflexiones de luz internas que pudieran afectar al sensor.
Aislamiento interno: Se aplicó una capa de recubrimiento aislante (silicón negro) en el fondo de la carcasa. Esto cumple dos funciones: fijar el módulo para evitar vibraciones y aislar eléctricamente las soldaduras de la base.

Ensamble: Se insertó el módulo ESP32-CAM, alineando el lente con la apertura frontal, obteniendo un dispositivo compacto y protegido.

CONCLUSIÓN

El desarrollo de este laboratorio permitió al equipo comprender los desafíos prácticos de la visión artificial en sistemas embebidos. Los hallazgos más importantes fueron los siguientes:

- El hallazgo principal fue la dependencia crítica de la alimentación eléctrica. El prototipo era completamente inestable y sufría reinicios por Brownout hasta que se reemplazó la alimentación USB por una fuente externa de 5V/2A capaz de manejar los picos de corriente de la cámara. Además, el segundo hallazgo fue la gestión de la memoria (RAM). Se determinó que la omisión de la función esp_camera_fb_return() era la causa de los bloqueos del sistema, demostrando que la gestión explícita de la memoria es tan crucial como la lógica de detección.
- El descubrimiento clave fue que la calibración del sensor y el formato de los datos determinan el éxito del proyecto. La detección por umbral de color demostró ser funcional, pero extremadamente sensible a las condiciones de iluminación ambiental; un cambio en la luz del laboratorio requería una recalibración completa de los rangos RGB. El segundo hallazgo fue la necesidad de usar PIXFORMAT_RGB565 en lugar de JPEG, ya que solo el formato de datos crudos (RGB565) permite el análisis matemático de los píxeles necesario para el algoritmo de detección.

DEMOSTRACION DEL PROTOTIPO

El prototipo desarrollado con la ESP32-CAM se puede aplicar directamente para resolver un problema real en el entorno educativo: la suplantación de identidad en el pase de lista. En lugar de detectar una pila o un LED, el sistema se reconfigura para la detección de rostros, actuando como un kiosco de verificación en la entrada del salón.

La problemática es que los alumnos pueden registrar la asistencia de compañeros ausentes (suplantación). La solución combina la ESP32-CAM con un lector de ID (como QR o NFC). El alumno escanea su credencial, y el sistema exige que la ESP32-CAM valide la presencia física de una persona (detección de rostro, no reconocimiento).

Aquí, los componentes del laboratorio actúan como la interfaz de usuario: la pantalla OLED guía al alumno ("ESCANEE ID", "MIRE A LA CAMARA"). Si la ESP32-CAM detecta exitosamente un rostro tras el escaneo, el LED indicador se enciende en verde, el buzzer emite un "beep" de confirmación y la OLED muestra "ASISTENCIA REGISTRADA". Si no se detecta un rostro, el LED parpadea en rojo, el buzzer emite un tono de error y la OLED indica "ERROR: ROSTRO NO DETECTADO", previniendo así el fraude.

FUENTES BIBLIOGRÁFICAS

[[1] R. Santos and S. Santos, Build ESP32-CAM Projects. Random Nerd Tutorials, 2021. [Online]. Available: https://randomnerdtutorials.com/

[2] L. Llamas, "Guía definitiva ESP32-CAM: Primeros pasos y streaming de vídeo," LuisLlamas.es, 2021. [En línea]. Disponible: https://www.luisllamas.es/esp32-cam-video-streaming/

[3] J. A. Pérez López, "Prototipo de sistema de control de asistencia mediante detección facial utilizando ESP32-CAM," Tesis de grado, Dept. Ing. Eléctrica, Univ. de El Salvador, San Salvador, El Salvador, 2021

[4] E. M. Gonzalo, IoT: Internet de las Cosas: Desarrolla tus proyectos con ESP32 y ESP8266. Barcelona, España: Marcombo, 2019.

[5] A. P. Quishpe López, "Sistema de detección de objetos en tiempo real mediante procesamiento digital de imágenes con ESP32-CAM para aplicaciones de seguridad," Tesis de grado, Fac. Ing. en Sistemas, Electrónica e Industrial, Univ. Técnica de Ambato, Ambato, Ecuador, 2022.