

MINI PROYECTO 1: Desarrollo de un Sistema Inteligente de Monitoreo y Alertas para la Seguridad de Adultos Mayores

Lopez, J.D. (2191160), Mendez, M.A (2215342),Castrillon J.D.(2226042)

Universidad Autónoma de Occidente

Cali, Colombia

miguel.mendez@uao.edu.co

julian_david.lopez@uao.edu.co

juan.castrillnn@uao.edu.co

Abstract- *This article describes the development of the first mini-project of term 1, focused on the implementation of an intelligent system for real-time detection and response using the Arduino Nano 33 BLE Sense. The process involves data collection using the inertial sensors integrated into the device, training a machine learning model on the Edge Impulse platform, and deploying it on the microcontroller. Additionally, a mobile application developed in App Inventor interacts with the device, receiving real-time data and triggering automatic responses to detected events, such as sudden movements or periods of immobility. This approach combines the power of machine learning for movement pattern detection with a simple and accessible interface to facilitate monitoring and timely intervention..*

Keywords:*Arduino Nano 33 BLE Sense, App Inventor, Edge Impulse, artificial intelligence, inertial sensors, real-time monitoring, machine learning, motion detection.*

Resumen- *Este artículo describe el desarrollo del primer miniproyecto de la materia, enfocado en la implementación de un sistema inteligente para la detección y respuesta en tiempo real mediante el Arduino Nano 33 BLE Sense. El proceso incluye la captura de datos utilizando los sensores inerciales integrados en el dispositivo específicamente el acelerómetro, el entrenamiento de un modelo de inteligencia artificial en la plataforma Edge Impulse, y su posterior despliegue en el microcontrolador. Además, se integra una aplicación móvil desarrollada en App Inventor que*

interactúa con el dispositivo, recibiendo datos en tiempo real y activando respuestas automáticas ante eventos detectados, como movimientos bruscos o periodos de inmovilidad. Este enfoque combina el poder del machine learning en la detección de patrones de movimiento con una interfaz sencilla y accesible para facilitar el monitoreo y la intervención oportuna..

Palabras claves: *Arduino Nano 33 BLE Sense, App Inventor, Edge Impulse, inteligencia artificial, sensores inerciales, monitoreo en tiempo real, machine learning, detección de movimientos.*

I. INTRODUCCIÓN

El envejecimiento de la población a nivel global ha incrementado la demanda de soluciones tecnológicas que mejoren la seguridad y bienestar de los adultos mayores. Entre los principales riesgos para este grupo se encuentran las caídas y los periodos prolongados de inmovilidad, que son responsables de una alta tasa de lesiones graves y complicaciones médicas, incluyendo la muerte. La rápida detección de estos eventos y la intervención oportuna pueden marcar una diferencia significativa en la calidad de vida de las personas mayores, reduciendo el riesgo de complicaciones y mejorando su seguridad en el hogar.

Este proyecto propone el desarrollo de un sistema de monitoreo en tiempo real para la detección de caídas y otros eventos críticos en adultos mayores. A través del uso de sensores inerciales y una plataforma de inteligencia artificial, el sistema puede detectar

patrones de movimiento inusuales y enviar alertas automáticas a cuidadores y familiares mediante una aplicación móvil. La capacidad de proporcionar notificaciones inmediatas permite una respuesta rápida ante emergencias, reduciendo el tiempo de intervención y previniendo posibles complicaciones graves. Esta propuesta busca no solo ofrecer una herramienta innovadora para el cuidado de los adultos mayores, sino también brindar tranquilidad a sus familias y mejorar la efectividad de las medidas de seguridad.

El proyecto tiene un impacto directo en el **Objetivo de Desarrollo Sostenible 3: Salud y Bienestar**, que busca garantizar una vida sana y promover el bienestar en todas las edades. Al proporcionar un sistema de detección temprana de caídas y otros eventos críticos, el proyecto contribuye a prevenir lesiones graves y complicaciones médicas entre los adultos mayores. Esto no solo mejora su calidad de vida, sino que también facilita un envejecimiento saludable, permitiendo una intervención rápida que puede reducir la necesidad de hospitalizaciones y disminuir la sobrecarga en los sistemas de salud. Este enfoque innovador combina tecnología avanzada con el objetivo de mejorar el bienestar de uno de los grupos más vulnerables de la población.

II. OBJETIVOS

Objetivo General:

Desarrollar un sistema inteligente capaz de monitorear en tiempo real los movimientos de adultos mayores utilizando los sensores inerciales del Arduino 33 BLE Sense, procesando los datos mediante un modelo de inteligencia artificial entrenado en Edge Impulse para enviar alertas automáticas en caso de detectar eventos críticos, como caídas o inmovilidad prolongada, garantizando una respuesta oportuna.

Objetivos Específicos:

1. Configurar y programar el Arduino 33 BLE Sense para capturar y procesar datos de movimiento a través de sus acelerómetros y giroscopios.

2. Recopilar datos de movimientos típicos y críticos, entrenar un modelo de clasificación de eventos en la plataforma Edge Impulse, y desplegar dicho modelo en el Arduino para permitir la detección en tiempo real.
3. Desarrollar una aplicación móvil que reciba y procese la información proveniente del Arduino, ejecutando acciones automáticas como la activación de alarmas o el envío de notificaciones a cuidadores y familiares.
4. Llevar a cabo pruebas exhaustivas en diversos escenarios que incluyan situaciones de inmovilidad y caídas simuladas, a fin de validar la precisión y confiabilidad del sistema en la detección de emergencias.
5. Documentar detalladamente todo el proceso de desarrollo, desde la configuración inicial del hardware y software, hasta la implementación del modelo de inteligencia artificial y los resultados obtenidos durante las pruebas, garantizando la reproducibilidad del proyecto.

III. METODOLOGÍA

El sistema se compone de varios elementos clave que trabajan en conjunto para garantizar un monitoreo eficaz y una respuesta oportuna a eventos críticos. Los componentes principales incluyen:

1. *Arduino Nano 33 BLE Sense:* El microcontrolador central que captura y procesa datos de movimiento a través de sus sensores integrados.

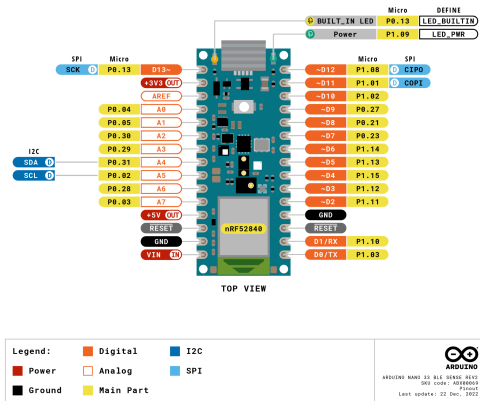


Imagen 1. Arduino nano 33 BLE sense

2. **Sensores Inerciales:** El acelerómetro y el giroscopio del Arduino Nano 33 BLE Sense se utilizan para medir y registrar los movimientos del usuario en tiempo real.
3. **Módulo Bluetooth:** Facilita la transmisión de datos entre el Arduino y la aplicación móvil, permitiendo una comunicación continua y en tiempo real.

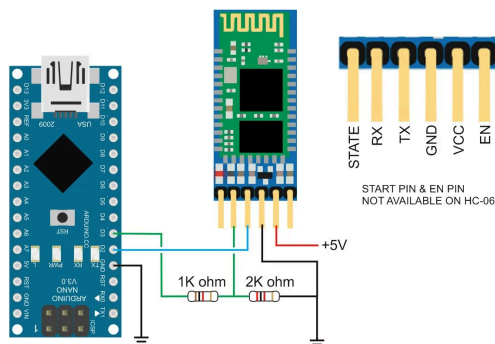


Imagen 2. Módulo Bluetooth

4. **Edge Impulse:** La plataforma de inteligencia artificial utilizada para entrenar un modelo de aprendizaje automático. El modelo entrenado se despliega en el Arduino para la clasificación en tiempo real de los datos de movimiento.
5. **Aplicación Móvil:** Recibe los datos procesados desde el Arduino a través de Bluetooth. La aplicación ejecuta diversas acciones basadas en los resultados del análisis de movimiento.

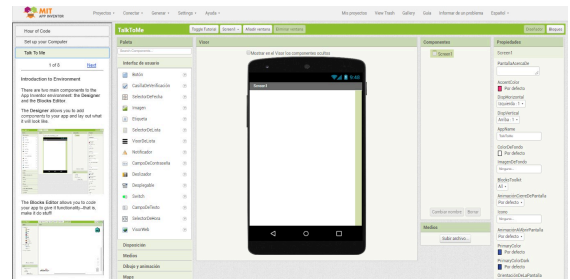


Imagen 3. Diseño en app inventor

Flujo de Datos:

1. **Configuración inicial:** 100Hz por 8000 ms

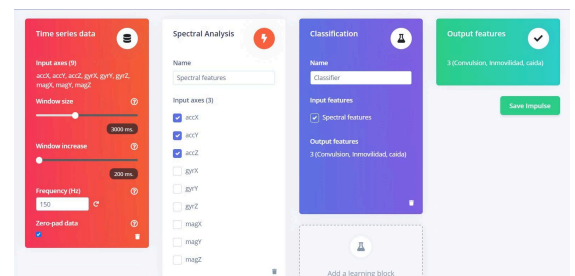


Imagen 3. Configuración de parámetros

2. **Procesamiento:** Los datos capturados se envían a Edge Impulse para entrenar el modelo de aprendizaje automático. Una vez entrenado, el modelo se despliega en el Arduino, que procesa los datos en tiempo real y clasifica los eventos según patrones predefinidos. En este caso se usaron 3 movimientos (convulsión, inmovilidad, caída), 10 tomas de cada dato para entrenamiento, 3 tomas de cada dato para entrenamiento

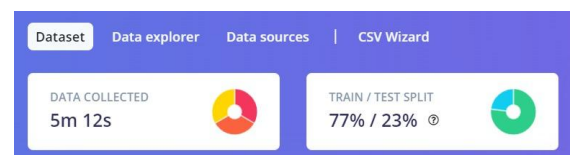


Imagen 4. Resultado de procesamiento de datos

3. **Entrenamiento del modelo:** Se utilizan los datos capturados para entrenar un modelo de clasificación en Edge Impulse. Se utilizaron 2

capas ocultas, una de 30 neuronas y otra de 15 neuronas

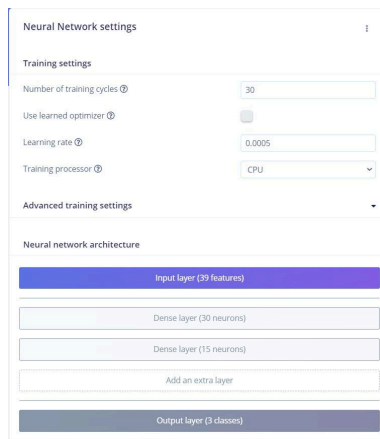


Imagen 5. Modelo a entrenar en edge impulse

4. *Validación del Modelo:* Evalúa el modelo utilizando herramientas como la matriz de confusión para verificar la precisión y fiabilidad del modelo.



Imagen 6. Resultados del modelo.

Test data Classify all

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCO...	LENGTH	ACCURACY	RESULT
caida.570t3ska	caida	8s	100%	26 caida
caida.570t11...	caida	8s	100%	26 caida
caida.570svu...	caida	8s	96%	25 caida, 1 uncertain
Convulsion.S...	Convulsion	8s	100%	26 Convulsion
Convulsion.S...	Convulsion	8s	100%	26 Convulsion
Convulsion.S...	Convulsion	8s	100%	26 Convulsion
Inmovilidad....	Inmovilidad	8s	100%	26 Inmovilidad
Inmovilidad....	Inmovilidad	8s	100%	26 Inmovilidad
Inmovilidad....	Inmovilidad	8s	100%	26 Inmovilidad

Imagen 7. Test con datos de validación



Imagen 8. Matriz de confusión

Los resultados obtenidos del entrenamiento del modelo son precisos y fiables, como se puede analizar en el gráfico de los datos se puede evidenciar que el modelo responde a uno de clasificación, también se puede ver que en temas de precisión, el modelo tiene una precisión del 100% con respecto al análisis de inmovilidad y de convulsión, y obtiene uno de 98.7% en caída, esto quiere decir que el modelo es acorde a lo que se necesita, a pesar de tener un pequeño error en el análisis de caída, no es algo significativo para dañar la correcta medición.

IV. IMPLEMENTACIÓN

Una vez validado, despliega el modelo en el Arduino 33 BLE Sense utilizando el firmware generado por Edge Impulse. El modelo se ejecutará en tiempo real, clasificando los movimientos capturados por los sensores y activando las alertas necesarias.

El código fuente en el Arduino 33 BLE Sense incluye funciones para capturar y procesar los datos de los sensores, conectarse a la aplicación móvil mediante Bluetooth, y ejecutar el modelo de Edge Impulse en tiempo real. Se realiza una aplicación móvil para dar respuesta a los datos obtenidos del modelo.



Imagen 9. App de respuesta.

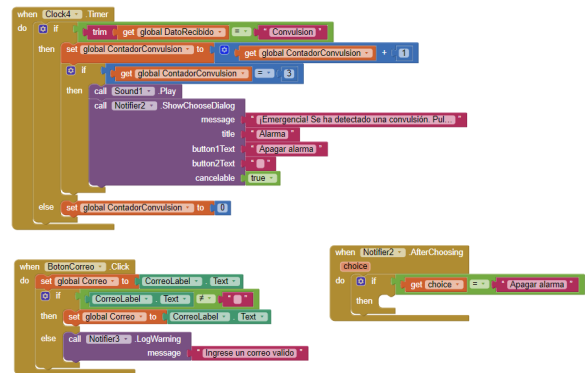
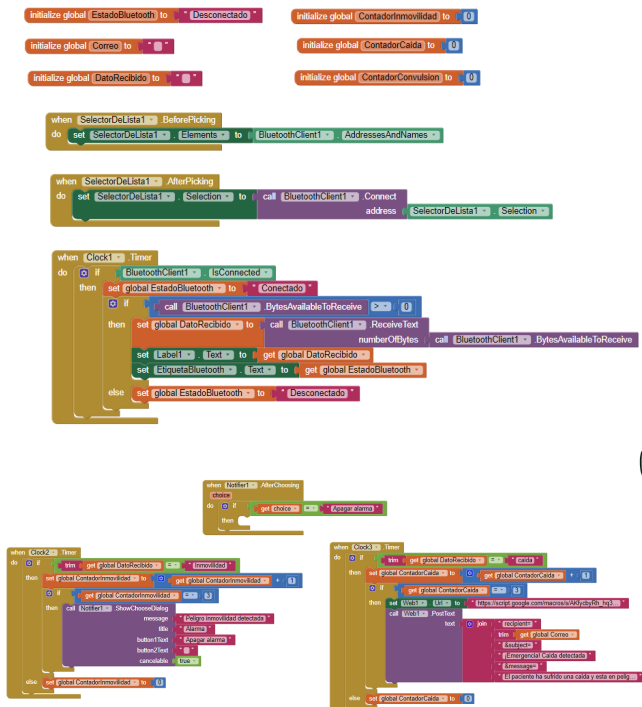


Imagen 10. Código de bloques de la app

La aplicación móvil está configurada para conectarse automáticamente al Arduino 33 BLE Sense cuando está dentro del rango. La interfaz de usuario es intuitiva, permitiendo a los usuarios visualizar el estado del sistema y recibir alertas en tiempo real.

Alerta de inmovilidad: Cuando la aplicación por medio bluetooth recibe la palabra “Inmovilidad” responde con una ventana emergente la cual dice que el paciente está inmóvil y está en peligro, ya que esto puede significar un paro cardíaco.

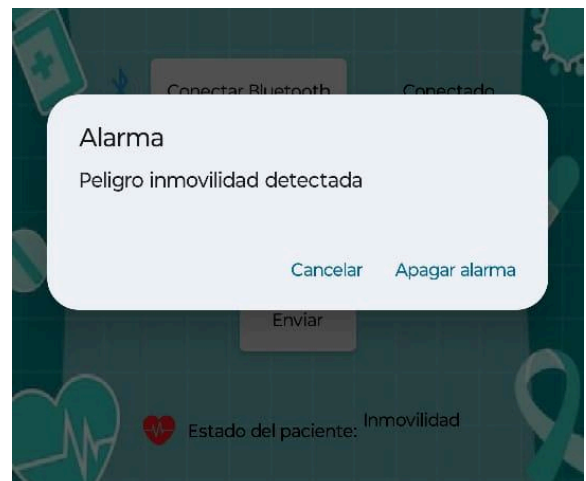


Imagen 11. Alarma de inmovilidad

Alerta de convulsión: Cuando la aplicación por medio bluetooth recibe la palabra “Convulsión” responde con una ventana emergente la cual dice que el paciente está en peligro y tiene una convulsión, además de esto inicia una alarma sonora que para hasta que se apague la alarma.



Imagen 12. Alarma de convulsión.

Alarma de caída: Cuando la aplicación por medio bluetooth recibe la palabra “Caída” responde mandando un correo a la dirección escrita en el espacio de ingrese su correo, el correo que se mande dice que el paciente está en peligro ya que sufrió una caída.



Imagen 13. Alarma de caída

Pruebas y Resultados:

Se realizaron pruebas en varios escenarios simulados para validar la precisión del sistema. Los resultados demostraron que el sistema es capaz de detectar y clasificar correctamente los movimientos críticos, activando las alarmas correspondientes en la aplicación móvil.

V.CONCLUSIONES

El proyecto demuestra que es posible desarrollar un sistema efectivo de monitoreo y alertas para adultos mayores utilizando el Arduino 33 BLE Sense y Edge Impulse. El sistema es capaz de detectar movimientos peligrosos con alta precisión y notificar a los cuidadores en tiempo real, lo que podría reducir

significativamente el tiempo de respuesta en emergencias.

Recomendaciones:

Mejoras Futuras: Se podría integrar un módulo GPS para proporcionar la ubicación exacta del usuario en caso de una caída. Además, se podría explorar la posibilidad de utilizar algoritmos de aprendizaje más avanzados en Edge Impulse para mejorar la precisión en la detección de patrones de movimiento.

Comercialización: Este sistema tiene el potencial de ser desarrollado como un producto comercial que podría beneficiar a una amplia población de adultos mayores y sus cuidadores.

VI. REFERENCIAS

- Arduino. "Arduino Nano 33 BLE Sense." [Online]. Available: <https://store.arduino.cc/nano-33-ble-sense>. [Accessed: 09-Sep-2024].
- Texas Instruments. "Understanding the Sensitivity of MEMS Accelerometers." [Online]. Available: <https://www.ti.com/lit/wp/slyt478/slyt478.pdf>. [Accessed: 09-Sep-2024].
- A. He and M. Wimmer, "Fall Detection Algorithms for Elderly Care: A Review," IEEE Access, vol. 7, pp. 77695-77705, Jun. 2019, doi: 10.1109/ACCESS.2019.2917607.
- Edge Impulse. "Getting started with TinyML on Arduino." [Online]. Available: <https://docs.edgeimpulse.com/docs/tutorials/arduino-nano-33-ble-sense>. [Accessed: 09-Sep-2024].
- A. Cavallaro, R. Gaddam, and S. Mukhopadhyay, "A Survey on Fall Detection Systems With a Focus on Wearable and Unobtrusive Sensing Solutions," IEEE Sensors Journal, vol. 17, no. 21, pp. 7728-7735, Nov. 2017, doi: 10.1109/JSEN.2017.2742905.
- B. Xie, L. Hu, and H. Zhang, "Design and Implementation of a Mobile Elderly Monitoring System," in Proc. of the 2018 IEEE Int. Conf. on Artificial Intelligence and Big Data (ICAIBD),

Chengdu, China, 2018, pp. 125-129, doi: 10.1109/ICAIBD.2018.8396186.

- P. K. Agarwal, M. F. Alhaddad, and A. S. Khamis, "Evaluation of Bluetooth Low Energy for Wearable Fall Detection Systems," in Proc. of the 2019 IEEE 16th Int. Conf. on Wearable and Implantable Body Sensor Networks (BSN), Chicago, IL, USA, 2019, pp. 1-4, doi: 10.1109/BSN.2019.8771027.
- S. Liu and D. Zhou, "A Smart Monitoring System for Senior Citizens Based on IoT Technologies," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4981-4989, Jun. 2019, doi: 10.1109/JIOT.2019.2902816.
- J. Smith and R. Liu, "Fall Detection Systems: A Comprehensive Review," IEEE Rev. Biomed. Eng., vol. 15, pp. 116-130, Mar. 2022, doi: 10.1109/RBME.2022.3143275.
- A. Bagheri, M. Y. Naderi, and S. Khorasani, "Bluetooth Low Energy: A Survey of Modulation Techniques and Protocols," IEEE Commun. Surveys Tuts., vol. 23, no. 4, pp. 2558-2582, Fourthquarter 2021, doi: 10.1109/COMST.2021.3109124.

VII. ANEXOS

Código completo arduino:

```
/* Includes */
#include <Project01_inferencing.h>
#include <Arduino_IMUv051.h> // Biblioteca para el IMU

/* Constantes */
#define CONVERT_G_TO_MS2 9.80665f
#define MAX_ACCEPTED_RANGE 2.0f

/* Variables privadas */
static bool debug_nn = false;

/* Funciones */
void setup() {
  // Iniciar comunicación serial con el PC (si está disponible)
  Serial.begin(115200);

  // Iniciar comunicación serial con el módulo HC-06 (usualmente a 9600 bps)
  Serial1.begin(9600); // HC-06 se conecta a Serial1 (en el Nano BLE Sense, es la UART secundaria)

  // Esperar 5 segundos para ver si hay una conexión serial (PC)
  unsigned long startMillis = millis();
  while ((Serial && millis() - startMillis < 5000) {
    // Esperar por conexión serial durante 5 segundos
  }

  if (!Serial) {
    // Si no está conectado a una PC, imprimir un mensaje en caso de estar conectado más tarde
    Serial.println("Iniciando sin conexión a PC...");
  }

  // Inicializar IMU
  if (IMU.begin()) {
    Serial.println("Fallo al inicializar IMU!");
  } else {
    Serial.println("IMU inicializada");
  }
}
```

```
if (EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME != 3) {
  Serial.println("ERR: EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME debe ser igual a 3 (los 3 ejes del sensor)");
  return;
}

Serial.println("Sistema listo para clasificar y enviar resultados por HC-06.");

void loop() {
  // Realiza la inferencia y obtiene el resultado
  String result = realizarInferencia();

  // Enviar el resultado por el módulo HC-06 (Serial1)
  Serial1.println(result); // Enviar la cadena clasificada al dispositivo Bluetooth conectado al HC-06
  Serial.println("Resultado enviado por HC-06: " + result); // Mostrar en el monitor serial del PC (si está disponible)

  delay(2000); // Controlar la frecuencia de envío
}

/** Realiza la inferencia y devuelve el resultado con la mayor probabilidad */
String realizarInferencia() {
  float buffer[EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE] = {0};

  // Capturar datos del acelerómetro
  for (size_t ix = 0; ix < EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE; ix += 3) {
    IMU.readAcceleration(buffer[ix], buffer[ix + 1], buffer[ix + 2]);

    // Limitar los valores del buffer a un rango aceptable
    for (int i = 0; i < 3; i++) {
      if (fabs(buffer[ix + i]) > MAX_ACCEPTED_RANGE) {
        buffer[ix + i] = (buffer[ix + i] > 0) ? MAX_ACCEPTED_RANGE : -MAX_ACCEPTED_RANGE;
      }
    }
  }
}
```

```
    // Convertir g a m/s^2
    buffer[ix + 0] *= CONVERT_G_TO_MS2;
    buffer[ix + 1] *= CONVERT_G_TO_MS2;
    buffer[ix + 2] *= CONVERT_G_TO_MS2;
  }

  // Crear una señal a partir del buffer
  signal_t signal;
  int err = numpy::signal_from_buffer(buffer, EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE, &signal);
  if (err != 0) {
    Serial.println("Error al crear la señal desde el buffer");
    return "Error";
  }

  // Ejecutar la clasificación
  ei_impulse_result_t result = {0};
  err = run_classifier(&signal, &result, debug_nn);
  if (err != EI_IMPULSE_OK) {
    Serial.println("Error al ejecutar el clasificador");
    return "Error";
  }

  // Encontrar la clase con mayor probabilidad
  float max_probabilidad = 0.0;
  String mejor_clase = "";

  for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
    if (result.classification[ix].value > max_probabilidad) {
      max_probabilidad = result.classification[ix].value;
      mejor_clase = String(result.classification[ix].label);
    }
  }

  #if EI_CLASSIFIER_HAS_ANOMALY == 1
  if (result.anomaly > max_probabilidad) {
    mejor_clase = "Anomalia";
  }
}

#endif

Serial.println("Clasificación más probable: " + mejor_clase);
return mejor_clase;
}
```

Enlace Edge impulse:

<https://studio.edgeimpulse.com/public/507816/live>

