

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
ESCUELA INFORMÁTICA E INTELIGENCIA ARTIFICIAL**

**TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN.**

PROTOTIPO DE IDENTIFICACIÓN DEL MOSQUITO AEDES CON TINYML

MARCO ANTONIO MUÑOZ VILLARREAL

TUTOR: MGS. STALIN ARCINIEGAS

IBARRA – ECUADOR

FEBRERO – 2025

Ibarra, 17 de Febrero del 2025

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de Marco Antonio Muñoz Villarreal titulado: PROTOTIPO DE IDENTIFICACIÓN DEL MOSQUITO AEDES CON TINYML, presentado por el estudiante Marco Antonio Muñoz Villarreal con cédula de ciudadanía N° 0401773437 , para obtener el Título de Ingeniero en Tecnologías de la Información.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.

| | |
|--|--|
| Turnitin Informe de Originalidad | |
| Procesado el: 17-feb.-2025 09:38 -05 Identificador: 2591119596 Número de palabras: 10206 Entregado: 1 | |
| Prototipo de identificación del mosquito AEDES con TINYML Por MARCO ANTONIO MUNOZ VILLARREAL | |
| Índice de similitud 9% | Similitud según fuente Internet Sources: 8% Publicaciones: 15% Trabajos del estudiante: 5% |



(f:)

Mgs. Stalin Marcelo Arciniegas Aguirre

C.C.: 1003496815

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El jurado examinador, aprueba el presente informe de investigación en nombre de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCEI):



(f):

Mgs. Stalin Marcelo Arciniegas Aguirre

C.C.: 1003496815



(f):

Dra. Dulce Milagro Rivero Albarran

C.C.: 1757608961



(f):

Dr. Francklin Ivan Rivas Echeverria

C.C.: 175706379-5

ACTA DE CESIÓN DE DERECHOS

Yo *Marco Antonio Muñoz Villarreal*, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilidades de sus obras o prestaciones, a título gratuito u oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 17 de Febrero de 2025




f):

Marco Antonio Muñoz Villarreal

C.C.: 0401773437

AUTORÍA

Yo, *Marco Antonio Muñoz Villarreal*, portador de la cédula de ciudadanía C.C. 0401773437, declaro que la presente investigación es de total responsabilidad del autor, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Sede Ibarra de posibles reclamos o acciones legales.

f): 

Marco Antonio Muñoz Villarreal

C.C.: 0401773437

DEDICATORIA Y AGRADECIMIENTOS

Quiero dedicar esta tesis a todas las personas que me han brindado su apoyo, motivación y aliento a lo largo de este arduo camino. Cada uno de ustedes ha sido un pilar fundamental para superar los desafíos y crecer tanto en lo personal como en lo profesional.

Mi más profundo agradecimiento a mi familia, especialmente a mis padres, cuyo amor incondicional y apoyo financiero hicieron posible este logro. Su paciencia, comprensión y las oportunidades que me brindaron son un regalo invaluable que atesoraré por siempre.

ÍNDICE DE CONTENIDOS

| | |
|--|------|
| CERTIFICACIÓN TUTOR | ii |
| PÁGINA DE APROBACIÓN DEL TRIBUNAL | iii |
| ACTA DE CESIÓN DE DERECHOS..... | iv |
| AUTORÍA | v |
| DEDICATORIA Y AGRADECIMIENTOS | vi |
| ÍNDICE DE CONTENIDOS | vii |
| ÍNDICE DE TABLAS | xi |
| ÍNDICE DE FIGURAS | xi |
| RESUMEN Y PALABRAS CLAVE | xii |
| ABSTRACT | xiii |
| INTRODUCCIÓN | 1 |
| Estado Del Arte | 3 |
| 1.1. Antecedentes | 3 |
| 1.1.1. Incidencia histórica de enfermedades transmitidas por Aedes..... | 3 |
| 1.1.2. Impacto económico y social de estas enfermedades | 3 |
| 1.1.3. Medidas actuales de prevención y control | 3 |
| 1.2. Métodos Tradicionales de Identificación de Mosquitos..... | 4 |
| 1.2.1. Identificación basada en PCR | 4 |
| 1.2.2. Técnicas de captura | 4 |
| 1.2.3. Diagnóstico larvario | 5 |
| 1.3. Tecnologías Emergentes para la Identificación | 5 |

| | | |
|--------|---|----|
| 1.3.1. | Visión artificial e imágenes | 5 |
| 1.3.2. | Biosensores y dispositivos portátiles | 5 |
| 1.4. | Sensores y Dispositivos para Captura de Datos | 5 |
| 1.4.1. | Sensores de sonido e identificación acústica..... | 6 |
| 1.4.2. | Machine learning aplicado a identificación acústica | 6 |
| 1.4.3. | Seed ESP32S3 Sense..... | 7 |
| 1.5. | Fundamentos como base para TinyML..... | 7 |
| 1.5.1. | TinyML..... | 7 |
| 1.5.2. | Inteligencia Artificial | 8 |
| 1.5.3. | Machine Learning..... | 8 |
| 1.5.4. | Edge Impulse | 8 |
| 1.6. | Tecnologías de conectividad y comunicación | 9 |
| 1.6.1. | IoT | 9 |
| 1.6.2. | Comunicación máquina a máquina..... | 9 |
| 1.6.3. | Flutter..... | 10 |
| 1.7. | Estudios previos de TinyML | 10 |
| 1.8. | Aspectos Socioeconómicos | 11 |
| 1.8.1. | Implementación sostenible de nuevas tecnologías de vigilancia | 11 |
| 2. | MATERIALES Y MÉTODOS | 12 |
| 2.1. | Metodología de Desarrollo | 12 |
| 2.1.1. | Diseño del Sistema | 12 |

| | | |
|----------|---|----|
| 2.1.2. | Desarrollo del Hardware | 12 |
| 2.1.2.1. | Entrenamiento del modelo de aprendizaje automático..... | 13 |
| 2.1.2.2. | Programación del dispositivo embebido | 13 |
| 2.1.2.3. | Desarrollo de la aplicación móvil | 14 |
| 2.2. | Evaluación del Sistema..... | 14 |
| 2.2.1. | Recopilación y Procesamiento de Datos..... | 14 |
| 2.2.2. | Evaluación del Rendimiento..... | 15 |
| 2.3. | Herramientas y Plataformas Utilizadas..... | 15 |
| 2.4. | Desarrollo de la Propuesta..... | 16 |
| 2.4.1. | Selección de los sensores..... | 16 |
| 2.4.2. | Comparación con otros dispositivos | 18 |
| 2.4.3. | Selección de los servicios web | 20 |
| 2.5. | Capa de transmisión..... | 20 |
| 2.5.1. | Protocolo MQTT:..... | 21 |
| 2.5.2. | API REST (NestJS): | 22 |
| 2.5.3. | Flutter:..... | 22 |
| 2.5.4. | Flujo de datos:..... | 22 |
| 2.6. | Despliegue en Producción..... | 23 |
| 3. | RESULTADOS Y DISCUSIÓN | 24 |
| 3.1. | Análisis y Discusión de los Resultados | 24 |
| 3.2. | Arquitectura del sistema..... | 24 |

| | | |
|--------|---|----|
| 3.3. | Capa de adquisición y procesamiento de datos | 26 |
| 3.3.1. | Adquisición de datos..... | 26 |
| 3.3.2. | Preprocesamiento de datos..... | 26 |
| 3.3.3. | Procesamiento de datos | 27 |
| 3.3.4. | Diseño y entrenamiento del modelo de aprendizaje automático | 27 |
| 3.4. | Suministro de Energía | 33 |
| 3.5. | Capa de presentación | 34 |
| 3.6. | Pruebas de funcionamiento..... | 37 |
| 3.6.1. | Recopilación de datos de prueba..... | 37 |
| 3.6.2. | Evaluación del rendimiento | 38 |
| | CONCLUSIONES..... | 40 |
| | RECOMENDACIONES | 42 |
| | REFERENCIAS BIBLIOGRÁFICAS | 43 |
| | ANEXOS | 47 |
| | Repositorio de Código Fuente GitHub | 47 |
| | Carta de Aceptación..... | 48 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1 Comparación Dispositivos | 19 |
| Tabla 2 Resultados Pruebas..... | 39 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1 Diagrama de flujo de trabajo de un proyecto TinyML | 13 |
| Figura 2 Pinout del Xiao ESP32S3 Sense | 17 |
| Figura 3 Diagrama de la Arquitectura del Sistema | 25 |
| Figura 4 Diagrama de conexión del dispositivo lot | 26 |
| Figura 5 Recolección de Datos | 27 |
| Figura 6 Bloques de Procesamiento y Aprendizaje | 28 |
| Figura 7 Parámetros de Filtrado | 29 |
| Figura 8 Entrenamiento de Red Neuronal..... | 30 |
| Figura 9 Modelo de comportamiento generado | 31 |
| Figura 10 Pantalla conexión bluetooth..... | 34 |
| Figura 11 Configuración credenciales wifi..... | 35 |
| Figura 12 Visualización de datos..... | 36 |
| Figura 13 Prototipo..... | 36 |
| Figura 14 Repositorio Backend Nestjs..... | 47 |

RESUMEN Y PALABRAS CLAVE

Este proyecto presenta el desarrollo de un sistema innovador para la identificación automática del mosquito Aedes mediante la implementación de tecnología TinyML.

El sistema integra hardware especializado para la captura de señales acústicas emitidas por el mosquito, un modelo de aprendizaje automático para dispositivos de bajo consumo, además se desarrolla una aplicación móvil para la visualización y gestión de datos. La solución propuesta utiliza la plataforma Edge Impulse para el entrenamiento y optimización del modelo, permitiendo una identificación precisa y en tiempo real del mosquito. El prototipo está diseñado para operar de manera autónoma en condiciones de campo, contribuyendo significativamente a la vigilancia entomológica y el control vectorial.

Este prototipo busca optimizar las estrategias actuales de prevención y control de enfermedades transmitidas por el Aedes, como el dengue, zika y chikungunya, especialmente en regiones con recursos limitados.

La detección temprana de mosquitos permite a las autoridades sanitarias tomar decisiones informadas de manera rápida y efectiva, como la implementación de medidas de control focalizadas o la alerta a la población en zonas de riesgo.

Se espera que este prototipo contribuya a reducir la incidencia de enfermedades transmitidas por el Aedes y a mejorar la calidad de vida de las comunidades, especialmente en aquellas con recursos limitados.

Palabras clave: Aedes, TinyML, Edge Impulse, identificación acústica, vigilancia entomológica, aprendizaje automático, salud pública, aplicación móvil.

ABSTRACT

This project presents the development of an innovative system for the automatic identification of the Aedes mosquito by implementing TinyML technology.

The system integrates specialized hardware to capture the mosquito's acoustic signals, a machine learning model for low-power devices, and a mobile application for data visualization and management. The proposed solution uses the Edge Impulse platform for model training and optimization, enabling accurate and real-time identification of the mosquito. The prototype is designed to operate autonomously in field conditions, significantly contributing to entomological surveillance and vector control.

This prototype seeks to optimize current strategies for the prevention and control of Aedes-borne diseases, such as dengue, zika, and chikungunya, especially in resource-limited regions.

Early detection of mosquitoes allows health authorities to make informed decisions quickly and effectively, such as implementing targeted control measures or alerting the population in at-risk areas.

It is expected that this prototype will contribute to reducing the incidence of Aedes-borne diseases and improving the quality of life of communities, especially in those with limited resources.

Keywords: Aedes, TinyML, Edge Impulse, acoustic identification, entomological surveillance, machine learning, public health, mobile application.

INTRODUCCIÓN

El mosquito *Aedes aegypti* constituye una de las amenazas más significativas para la salud pública global, siendo el principal vector de enfermedades tropicales como el dengue, zika, chikungunya y fiebre amarilla. La magnitud de esta amenaza se evidencia en las estadísticas recientes: tan solo en 2023, las Américas registraron más de 3 millones de casos de dengue, subrayando la urgente necesidad de desarrollar soluciones innovadoras para su control y vigilancia (Organización Panamericana de la Salud, 2023).

Los métodos tradicionales de identificación y monitoreo del *Aedes aegypti*, como la identificación visual, presentan limitaciones significativas en términos de escalabilidad, eficiencia y accesibilidad, lo que dificulta la respuesta oportuna ante brotes de enfermedades transmitidas por este vector, especialmente en regiones con recursos limitados. Por lo tanto, se hace necesario el desarrollo de soluciones innovadoras que permitan superar estas limitaciones.

TinyML, con su capacidad para ejecutar modelos de aprendizaje automático en dispositivos de bajo consumo y con recursos limitados, ofrece una alternativa prometedora para la identificación automatizada de mosquitos. Esta tecnología permite el desarrollo de dispositivos portátiles y de bajo costo que pueden analizar datos en tiempo real, lo que facilita la detección temprana de brotes y la implementación de medidas de control más efectivas. Además, TinyML reduce la dependencia de la infraestructura de la nube, lo que lo hace ideal para su uso en áreas remotas o con acceso limitado a internet.

Esta investigación propone el desarrollo de un sistema integrado que combina:

- Hardware especializado para la captura de señales acústicas del mosquito
- Un modelo de aprendizaje automático optimizado mediante Edge Impulse
- Una aplicación móvil para la gestión y visualización de datos en tiempo real

Los beneficios esperados de esta implementación incluyen:

- Optimización de costos en vigilancia entomológica
- Mayor accesibilidad a herramientas de identificación

- Detección más rápida y precisa de focos de infestación
- Mejora en la toma de decisiones para el control vectorial

El objetivo principal de esta investigación es desarrollar un prototipo de identificación del mosquito *Aedes* mediante TinyML que, conectado a una aplicación móvil para la visualización de los resultados de la detección, permitirá fortalecer las estrategias de vigilancia entomológica y futuras investigaciones sobre el control de enfermedades transmitidas por vectores.

Para alcanzar este propósito, se establecieron los siguientes objetivos específicos:

1. Diseñar el hardware necesario mediante TinyML para la captura de señales acústicas del mosquito
2. Implementar un modelo de aprendizaje automático utilizando Edge Impulse para la detección y clasificación de señales
3. Desarrollar una aplicación móvil para la conexión con el prototipo, registro de datos
4. Evaluar la precisión y eficacia del sistema

Este documento tiene una estructura de tres capítulos: el primero aborda el estado del arte con información actualizada que fundamenta el proyecto; el segundo detalla la metodología empleada; y el tercero presenta los resultados obtenidos, incluyendo el prototipo desarrollado y su aplicación móvil, así como las conclusiones y recomendaciones derivadas de este trabajo.

CAPÍTULO I

Estado Del Arte

1.1. Antecedentes

Los antecedentes presentados proveen un contexto sólido sobre el impacto del Aedes como vector de enfermedades prioritarias en salud pública. Resaltan la necesidad crítica de medidas efectivas de vigilancia y control entomológico.

1.1.1. Incidencia histórica de enfermedades transmitidas por Aedes

El Aedes aegypti es el principal vector de enfermedades transmitidas, como el dengue, zika y chikungunya. Estas enfermedades son endémicas en muchas regiones tropicales y subtropicales del mundo, y se han observado brotes significativos en varios países, especialmente en América Latina. Por ejemplo, en 2023, se han reportado más de 3 millones de nuevos casos de dengue en las Américas, superando cifras récord de años anteriores (Organización Panamericana de la Salud, 2023).

El dengue es endémico en regiones tropicales, incluyendo Ecuador, donde ha habido un aumento en casos entre 2016 y 2018, especialmente en Guayas, Manabí y el Oro. También se reportan casos de zika y chikungunya transmitidos por el mosquito Aedes. El control de vectores y la vigilancia son cruciales para prevenir la propagación de estas enfermedades en Ecuador (Reyes Baque et al., 2019).

1.1.2. Impacto económico y social de estas enfermedades

Las enfermedades transmitidas por el mosquito Aedes, como el dengue, zika, chikungunya y fiebre amarilla, imponen un alto costo económico. Un estudio estima que el dengue solo representa una carga económica de alrededor de 8.900 millones de dólares en países afectados. Estas enfermedades afectan los sistemas de salud, la productividad laboral y generan costos directos e indirectos, contribuyendo a desigualdades regionales y afectando la economía, incluyendo el turismo y la productividad laboral (Reyes Baque et al., 2019).

1.1.3. Medidas actuales de prevención y control

Las medidas de prevención y control contra el Aedes incluyen la eliminación de criaderos, aplicación de insecticidas, uso de repelentes, mosquiteros y vigilancia entomológica.

Estrategias innovadoras como esterilización y manipulación biológica se han implementado. Control del vector, vigilancia y manejo de resistencia a insecticidas son clave para prevenir la propagación de enfermedades como el dengue, zika, chikungunya y fiebre amarilla, dado su alto costo económico y social (WHO/OPS, 2019).

1.2. Métodos Tradicionales de Identificación de Mosquitos

Los métodos tradicionales de identificación y seguimiento de mosquitos, como la identificación visual y el análisis de PCR, tienen limitaciones en cuanto a eficiencia, escalabilidad y accesibilidad. Estas limitaciones dificultan la capacidad de responder eficazmente a posibles brotes, especialmente en regiones con recursos limitados.

1.2.1. Identificación basada en PCR

Los métodos de identificación basados en PCR donde analizan si hay material genético del virus, estas reducen la cantidad de tiempo necesario para identificar especies, y los ensayos en tiempo real eliminan la necesidad de una visualización separada (Kothera et al., 2017).

1.2.2. Técnicas de captura

La vigilancia activa utiliza múltiples técnicas, como atrapar mosquitos, contar los mosquitos atrapados para obtener estimaciones de población y realizar pruebas a los mosquitos atrapados (Centers for Disease Control and Prevention USA, s. f.).

La identificación de mosquitos vectores, como el *Aedes aegypti*, tradicionalmente se basa en la inspección visual por parte de expertos en entomología, lo que puede ser un proceso lento, costoso y sujeto a errores humanos. Además, la confirmación de la presencia de patógenos en los mosquitos a menudo requiere técnicas moleculares como la PCR (Reacción en Cadena de la Polimerasa), que implican equipos de laboratorio especializados y personal capacitado, lo que limita su aplicación en entornos de bajos recursos. Estos métodos, si bien son efectivos, presentan limitaciones en términos de tiempo, costo y accesibilidad, lo que dificulta la implementación de medidas de control oportunas y eficientes.

1.2.3. Diagnóstico larvario

El diagnóstico larvario, que implica extraer e identificar larvas de mosquitos usando claves taxonómicas morfológicas, es crucial para la vigilancia entomológica y el control de enfermedades transmitidas por mosquitos (Oporta Reyes et al., 2018).

1.3. Tecnologías Emergentes para la Identificación

Las tecnologías emergentes están logrando avances significativos en la automatización y mejora de la identificación de mosquitos mediante avances en inteligencia artificial y biosensores. Estas innovaciones ofrecen soluciones prometedoras para mejorar la precisión, la eficiencia y la accesibilidad de los métodos de identificación de mosquitos.

1.3.1. Visión artificial e imágenes

Las tecnologías emergentes, como la visión artificial y la imagenología, se están adoptando rápidamente en los campos de la entomología y la salud pública para lograr una identificación más eficiente y precisa de los mosquitos. Estas tecnologías aprovechan los sistemas de procesamiento de imágenes que pueden analizar datos visuales para identificar especies de mosquitos en función de sus características morfológicas, como el tamaño, la forma, el color y los patrones de las alas (Santana Esparza & Hernández Machuca, 2023).

1.3.2. Biosensores y dispositivos portátiles

Los biosensores se pueden integrar en dispositivos portátiles, lo que mejora su utilidad en diversas aplicaciones, como el diagnóstico de enfermedades y el control medioambiental. Estos dispositivos utilizan un biorreceptor para reconocer de forma selectiva los analitos objetivo, junto con un transductor que convierte las respuestas biológicas en señales eléctricas. El diseño compacto y el bajo coste de los biosensores portátiles los hacen ideales para realizar pruebas in situ, lo que permite un análisis rápido y eficiente en campos como la medicina, la seguridad alimentaria y la conservación del medio ambiente (Liu, 2024).

1.4. Sensores y Dispositivos para Captura de Datos

En esta sección se analizan los distintos sensores y dispositivos que se utilizan para captar señales acústicas y visuales de los mosquitos, lo que permite una identificación y

un seguimiento eficientes. Estas herramientas desempeñan un papel crucial en la vigilancia entomológica y el control de enfermedades, ya que proporcionan datos valiosos sobre la presencia de mosquitos, la composición de las especies y la densidad de población.

1.4.1. Sensores de sonido e identificación acústica

Las tecnologías emergentes, como biosensores y dispositivos portátiles, se aplican en entomología y salud pública para identificar mosquitos. Estos sistemas utilizan procesamiento de imágenes basado en características morfológicas y patrones de comportamiento. Una de esas innovaciones es la creación de claves de identificación "inversas". A diferencia de las claves taxonómicas tradicionales que requieren conocimientos especializados, estas claves inversas están diseñadas para que las utilicen personas no expertas. Emplean un método simplificado basado en imágenes, que permite a personas con una formación mínima identificar especies de mosquitos basándose en características visuales. Esto permite que una gama más amplia de personas, incluidos miembros de la comunidad y trabajadores de campo, participen en las actividades de vigilancia de los mosquitos. Estos avances en la tecnología de identificación de mosquitos son muy prometedores para mejorar la vigilancia y el control de enfermedades, en particular en entornos con recursos limitados donde los métodos tradicionales pueden resultar poco prácticos o prohibitivos en términos de costo (González-Pérez et al., 2022).

1.4.2. Machine learning aplicado a identificación acústica

El aprendizaje automático se utiliza cada vez más para la identificación acústica de insectos, principalmente a través del análisis de los sonidos de los aleteos de las alas. Este enfoque implica el uso de sensores acústicos para capturar las frecuencias de sonido únicas producidas por los aleteos de los insectos. Luego, estas señales se procesan y analizan mediante algoritmos de aprendizaje automático, que pueden entrenarse para reconocer patrones y diferenciar entre especies en función de variaciones sutiles en sus sonidos. Este método ha demostrado ser eficaz para identificar varias especies de insectos, incluidos los mosquitos, y tiene varias ventajas sobre los métodos tradicionales. Permite el seguimiento automatizado y continuo de las

poblaciones de insectos, lo que proporciona datos valiosos para los esfuerzos de vigilancia y control, especialmente de las enfermedades que pueden transmitir (Tejero Caballo, 2020).

1.4.3. Seeed ESP32S3 Sense

Seeed Studio XIAO ESP32S3 Sense es una potente placa de desarrollo compacta que combina la versatilidad del microcontrolador ESP32-S3 con una amplia gama de sensores integrados. Ideal para proyectos que integran IoT, aprendizaje automático y *wearables*, esta placa ofrece conectividad WiFi y Bluetooth 5.0, un procesador *dual-core* de 32 bits y una generosa memoria flash QSPI de 2 MB. Su compatibilidad con Arduino y CircuitPython facilita el desarrollo, mientras que los sensores integrados, como el micrófono digital, la IMU y el PDM, la hacen especialmente adecuada para aplicaciones que requieren la recopilación y el procesamiento de datos del entorno. El bajo consumo de energía del XIAO ESP32S3 Sense lo convierte en una opción ideal para dispositivos portátiles y proyectos de IoT que demandan una larga duración de la batería. Además, admite interfaces de comunicación como UART, IIC y SPI, lo que permite la conexión con una variedad de periféricos y dispositivos externos. (Seeed Studio, 2023).

1.5. Fundamentos como base para TinyML

Esta sección proporciona una comprensión básica de TinyML y sus tecnologías relacionadas, sentando las bases para su aplicación en la identificación de mosquitos.

1.5.1. TinyML

TinyML es una tecnología de vanguardia que lleva las capacidades del aprendizaje automático a dispositivos pequeños con recursos limitados. Permite la implementación de modelos de aprendizaje automático en microcontroladores y otros sistemas integrados, lo que abre nuevas posibilidades para aplicaciones inteligentes en diversos campos (Monti De Nardi & Eduardo Monteiro, s. f.).

TinyML ofrece varias ventajas que lo convierten en una herramienta valiosa para diversas aplicaciones:

- **Bajo consumo de energía:** los modelos TinyML están optimizados para ejecutarse en dispositivos con energía y capacidades de procesamiento limitadas, lo que los hace ideales para aplicaciones alimentadas por batería o de bajo consumo energético.
- **Rentabilidad:** TinyML reduce la necesidad de hardware costoso y recursos de computación en la nube, lo que lo convierte en una solución rentable para implementar aplicaciones inteligentes.
- **Procesamiento en tiempo real:** TinyML permite el procesamiento de datos en el dispositivo, lo que reduce la latencia y permite la toma de decisiones en tiempo real.
- **Preservación de la privacidad:** al procesar datos localmente en el dispositivo, TinyML puede ayudar a preservar la privacidad del usuario y reducir el riesgo de violaciones de datos.
- **Accesibilidad:** TinyML hace que el aprendizaje automático sea accesible para una gama más amplia de usuarios y aplicaciones, incluidos aquellos en entornos remotos o con recursos limitados.

1.5.2. Inteligencia Artificial

La Inteligencia Artificial es un campo multidisciplinario que implica la comprensión computacional y la creación de artefactos que exhiben un comportamiento inteligente (Sahadevan, s. f.).

1.5.3. Machine Learning

El aprendizaje automático constituye un campo amplio que engloba diversos algoritmos destinados a realizar predicciones inteligentes mediante el análisis de conjuntos específicos de datos (Nichols et al., 2019).

1.5.4. Edge Impulse

Edge Impulse es una plataforma para desarrollar algoritmos de aprendizaje automático que se pueden embeber en sistemas integrados como microcontroladores u ordenadores de recursos reducidos. La plataforma permite a los desarrolladores recopilar datos,

entrenar modelos y optimizar bibliotecas para ejecutarlas en cualquier dispositivo perimetral, desde MCU de muy bajo consumo hasta objetivos de CPU y GPU de Linux eficientes (Edge Impulse, s. f.).

1.6. Tecnologías de conectividad y comunicación

Esta sección explora las tecnologías esenciales que permiten la conectividad, el procesamiento de datos y la experiencia del usuario en el sistema de identificación de mosquitos propuesto.

1.6.1. IoT

La Internet de las cosas (IoT) desempeña un papel crucial en la informática inteligente moderna al permitir la ejecución remota de tareas y la toma de decisiones descentralizada. El potencial del IoT radica en su capacidad de conectar dispositivos y facilitar el intercambio de datos. Esta interconexión se ve aún más amplificada por la aparición de la tecnología 5G, que se proyecta que conectará casi 50 mil millones de dispositivos. La integración perfecta de dispositivos y datos a través de IoT y 5G tiene el potencial de transformar diversas industrias y aspectos de la vida diaria (John et al., 2023).

1.6.2. Comunicación máquina a máquina

En el ámbito de la tecnología, el concepto de comunicación de máquina a máquina (M2M) ha surgido como un componente fundamental para optimizar diversos procesos. M2M se refiere esencialmente al intercambio automatizado de información entre dos o más dispositivos sin necesidad de intervención humana. Esta comunicación puede producirse a través de canales cableados o inalámbricos, lo que permite que los dispositivos compartan datos, instrucciones y actualizaciones de estado, lo que conduce a una mayor eficiencia y automatización. En esencia, M2M permite que los dispositivos "hablen" entre sí y tomen decisiones de forma autónoma, lo que agiliza las operaciones y mejora la productividad. La comunicación M2M se ha aplicado en una amplia gama de industrias y sectores, como la gestión de inventarios, el seguimiento de envíos, el control y la supervisión a distancia y la fabricación inteligente. Al automatizar tareas y procesos, la tecnología M2M reduce la necesidad de intervención manual, lo que se traduce en una

mayor eficiencia, una mayor precisión, una reducción de costes y una mejor toma de decisiones (Balaji & Fernando, 2022).

1.6.3. Flutter

Flutter, desarrollado por Google, es un marco versátil que facilita la creación de aplicaciones móviles para iOS, Android y web con un solo código base. Entre sus ventajas se incluyen desarrollo rápido con hot-reload, diseño responsivo con una biblioteca de widgets, compatibilidad multiplataforma simultánea para iOS y Android, y programación en Dart. Empresas como Google Ads, Hamilton, BMW, Google Pay y Nubank han tenido éxito utilizando Flutter, destacando su potencia y flexibilidad para construir aplicaciones de alta calidad. En resumen, Flutter ofrece eficiencia y rendimiento con una programación sencilla en Dart (Google, s. f.) .

1.7. Estudios previos de TinyML

El uso de TinyML en la salud pública tiene el potencial de revolucionar la forma en que se recopilan y procesan los datos en los entornos de atención médica. Si bien la investigación específica sobre las aplicaciones de TinyML en la salud pública es limitada, la literatura existente sugiere que TinyML se puede aplicar en varias áreas. TinyML puede permitir el desarrollo de dispositivos portátiles y sensores que monitoreen continuamente los signos vitales de los pacientes y otros datos de salud. Estos datos se pueden procesar localmente en el dispositivo y transmitir de forma inalámbrica a los proveedores de atención médica, lo que permite el monitoreo remoto de la salud de los pacientes y la detección temprana de posibles problemas. Esto puede ser particularmente beneficioso para los pacientes en áreas remotas o con acceso limitado a los centros de atención médica. Al permitir la implementación de modelos de aprendizaje automático en dispositivos con recursos limitados y de bajo consumo, TinyML puede mejorar la accesibilidad y la eficiencia de los servicios de atención médica. Los dispositivos TinyML se pueden implementar en diversos entornos para brindar monitoreo continuo de la salud, análisis de datos locales y capacidades de diagnóstico rápido. Esto puede generar intervenciones de atención médica más oportunas y efectivas, en particular en comunidades desatendidas (Velasquez Correa, 2022).

1.8. Aspectos Socioeconómicos

1.8.1. Implementación sostenible de nuevas tecnologías de vigilancia

La implementación sostenible de tecnologías de vigilancia de mosquitos requiere eficiencia energética, diseño duradero, uso de biotecnologías, redes LoRaWAN, modelos TinyML adaptables, gestión de riesgos, equidad en el acceso, y transparencia. Estas consideraciones buscan garantizar soluciones eficientes, ecológicas y asequibles para monitorear y controlar la población de mosquitos, previniendo enfermedades y minimizando impactos ambientales y sociales (Cataluña et al., s. f.).

El artículo presenta un prototipo de solución para clasificar especies de mosquitos por el sonido de sus alas, utilizando dispositivos TinyML y la plataforma Edge Impulse. Esta solución, de bajo consumo y costo, opera sin intervención humana en áreas con recursos limitados. Basada en un conjunto de datos de acceso abierto, la propuesta es accesible para futuras investigaciones. La integración de LoRaWAN permite la transmisión remota de resultados, posibilitando el monitoreo global de insectos (Altayeb et al., 2022).

Además de la identificación de mosquitos, TinyML se está aplicando en diversas áreas de la salud pública, como la detección temprana de enfermedades respiratorias como el COVID-19 a través del análisis de sonidos de la tos (Chowdhury et al., 2022), el monitoreo de la calidad del aire para prevenir enfermedades (Wardana et al., 2023), y el diagnóstico de enfermedades cardíacas a través del análisis de electrocardiogramas (Acharya et al., 2017). Estos ejemplos demuestran el potencial de TinyML para transformar la atención médica y mejorar la salud pública, especialmente en entornos de bajos recursos.

CAPÍTULO II

2. MATERIALES Y MÉTODOS

2.1. Metodología de Desarrollo

Para el desarrollo del prototipo, se utilizará una metodología que combina elementos de ingeniería de software e ingeniería de aprendizaje automático. Se seguirá un proceso iterativo que incluye las siguientes etapas:

2.1.1. *Diseño del Sistema*

El sistema de identificación de mosquitos se compone de los siguientes elementos:

- **Dispositivo embebido:** Seeed Xiao ESP32S3 SENSE con micrófono digital integrado.
- **Plataforma de desarrollo de modelos de aprendizaje automático:** Edge Impulse.
- **Aplicación móvil:** Desarrollada con Flutter para la conexión Bluetooth, recepción de datos y visualización de notificaciones.

El Seeed Xiao ESP32S3 SENSE fue seleccionado por su bajo consumo de energía, tamaño compacto y conectividad Bluetooth de bajo consumo, ideal para aplicaciones portátiles. Edge Impulse permite el desarrollo y la implementación de modelos de aprendizaje automático de manera eficiente, mientras que Flutter facilita la creación de interfaces de usuario atractivas y multiplataforma.

2.1.2. *Desarrollo del Hardware*

El prototipo se basa en el microcontrolador Seeed Xiao ESP32S3 SENSE, que cuenta con un procesador Xtensa® Dual-Core 32-bit LX7, 512KB de RAM y 2MB de memoria flash. El dispositivo integra un micrófono digital MEMS que se utiliza para capturar las señales acústicas de los mosquitos. No se requiere hardware adicional para la adquisición de datos.

2.1.2.1. Entrenamiento del modelo de aprendizaje automático

Figura 1

Diagrama de flujo de trabajo de un proyecto TinyML



Para el desarrollo del modelo de aprendizaje automático se utilizó la plataforma Edge Impulse. La Figura 1 muestra la metodología que se siguió durante el desarrollo de este trabajo. Esta es una metodología del flujo de trabajo típico de proyectos de ML, que consiste en los siguientes pasos Altayeb et al., (2022):

- **Recopilación de datos:** Se utilizó un conjunto de datos público de sonidos de mosquitos, similar al descrito en el artículo.
- **Preprocesamiento de datos:** Las señales de audio se segmentaron en ventanas de 1 segundo y se les aplicó una transformada de Fourier para obtener sus espectrogramas.
- **Diseño del modelo:** Se empleó una red neuronal convolucional (CNN) con dos capas convolucionales 1D, similar a la arquitectura descrita en el artículo.
- **Entrenamiento del modelo:** Se ajustaron los hiperparámetros del modelo (tasa de aprendizaje, tamaño del lote, número de épocas) para obtener un rendimiento óptimo.
- **Evaluación del modelo:** Se evaluó el modelo con un conjunto de datos de prueba para determinar su precisión, sensibilidad y especificidad.

2.1.2.2. Programación del dispositivo embebido

Se utilizó el IDE de Arduino para programar el Seeed Xiao ESP32S3 SENSE. Se implementó la lógica para la adquisición de datos del micrófono, la inferencia del modelo de aprendizaje automático y la transmisión de los resultados a través de Bluetooth. Se utilizaron las librerías del Seeed ESP32S3 SENSE para la comunicación Bluetooth.

2.1.2.3. Desarrollo de la aplicación móvil

La aplicación móvil se desarrolló con Flutter, un framework multiplataforma que permite crear interfaces de usuario atractivas y con un buen rendimiento. La aplicación se encarga de:

- Buscar y conectarse al dispositivo embebido a través de Bluetooth.
- Recibir los datos del dispositivo embebido.
- Mostrar una notificación en la pantalla del teléfono cuando se detecta el mosquito específico.

2.2. Evaluación del Sistema

2.2.1. Recopilación y Procesamiento de Datos

La recopilación de datos se realizó utilizando el micrófono integrado en el Seeed Xiao ESP32S3 SENSE. Se capturaron las señales acústicas de las siguientes especies de mosquitos:

- Aedes
- Aedes albopictus
- Otras especies de mosquitos (especificar las especies)

También se grabó ruido ambiental para entrenar al modelo en la clasificación de sonidos que no corresponden a mosquitos.

El procesamiento de datos se realizó en la plataforma Edge Impulse. Las señales de audio se segmentaron en ventanas de 1 segundo y se les aplicó una transformada de Fourier para obtener sus espectrogramas. Estos espectrogramas se utilizaron como entrada para el modelo de aprendizaje automático.

2.2.2. Evaluación del Rendimiento

La evaluación del sistema se realizó utilizando un conjunto de datos de prueba independiente. Se midieron las siguientes métricas:

- **Precisión:** Porcentaje de clasificaciones correctas.
- **Sensibilidad:** Capacidad del sistema para detectar correctamente el mosquito específico.
- **Especificidad:** Capacidad del sistema para descartar correctamente otros mosquitos o ruidos.

Se registraron los resultados de las pruebas y se analizaron para determinar el rendimiento del sistema en la identificación del mosquito *Aedes*.

2.3. Herramientas y Plataformas Utilizadas

- **Edge Impulse:** Plataforma de aprendizaje automático utilizada para diseñar, entrenar y evaluar el modelo.
- **IDE de Arduino:** Entorno de desarrollo integrado utilizado para programar el microcontrolador el Xiao ESP32S3 SENSE, se eligió por su compatibilidad con el microcontrolador utilizado en el proyecto, su facilidad de uso y la amplia comunidad de usuarios.
- **Flutter:** Framework multiplataforma utilizado para desarrollar la aplicación móvil.
- **Nestjs:** Framework de Node.js para la creación de APIs REST escalables y eficientes. Se eligió por su capacidad para crear APIs robustas y escalables, su facilidad de uso y su integración con InfluxDB
- **InfluxDB:** Base de datos de series temporales optimizada para el almacenamiento y análisis de datos de series temporales. Se seleccionó por su capacidad para manejar eficientemente datos de series temporales, su escalabilidad y su integración con NestJS.

2.4. Desarrollo de la Propuesta

2.4.1. Selección de los sensores

La selección de los sensores es un paso crucial en el diseño de un sistema de identificación de mosquitos basado en TinyML. Para este proyecto, se optó por un micrófono digital MEMS integrado en el microcontrolador Seeed Xiao ESP32S3 Sense.

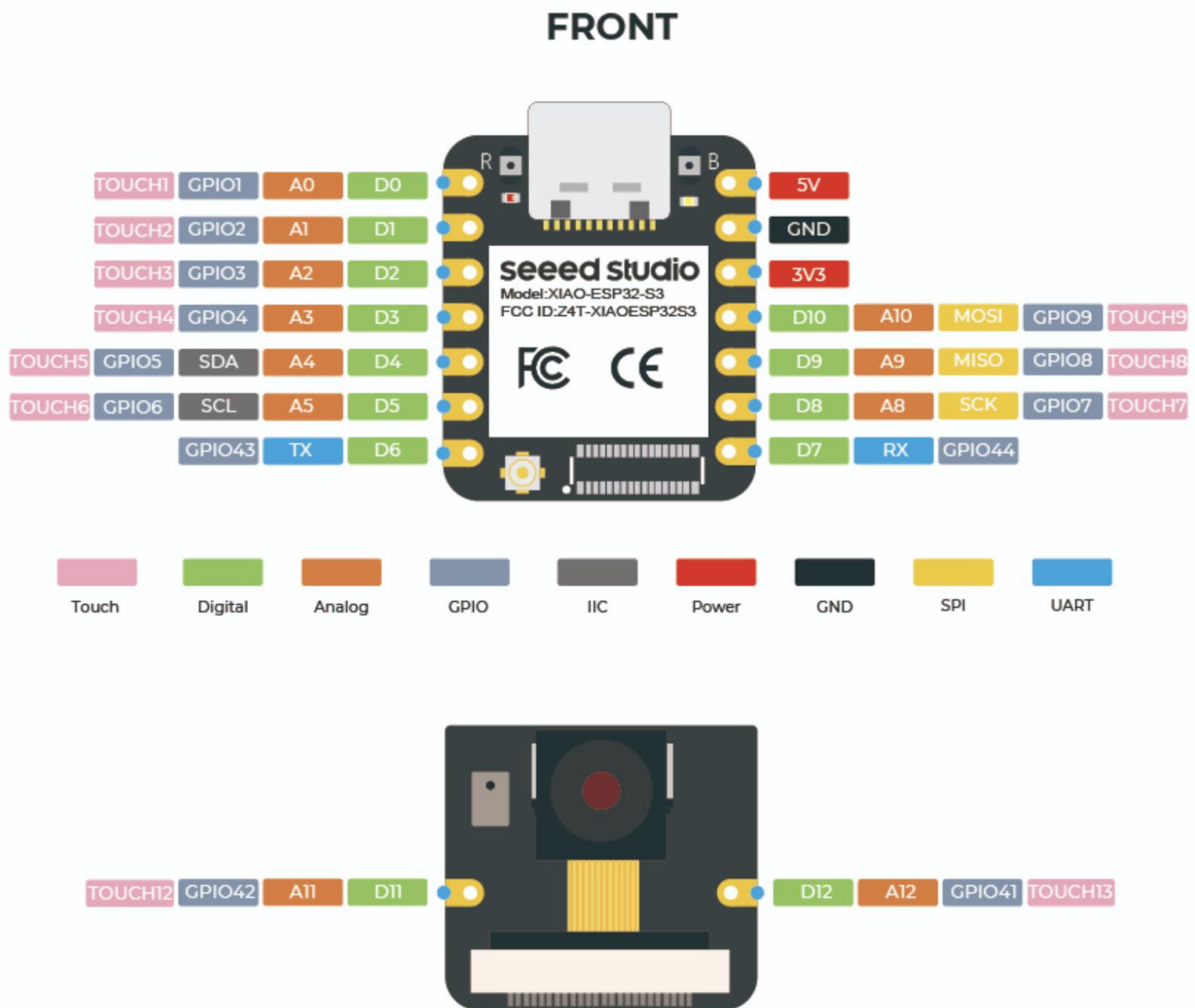
Justificación de la selección:

- **Alta sensibilidad:** Los micrófonos MEMS son conocidos por su alta sensibilidad, lo que permite capturar las débiles señales acústicas emitidas por los mosquitos.
- **Bajo consumo de energía:** El bajo consumo de energía de los micrófonos MEMS es crucial para la implementación de TinyML, ya que permite la operación autónoma del dispositivo durante largos periodos.
- **Tamaño compacto:** El tamaño compacto del micrófono MEMS facilita la integración en el dispositivo, contribuyendo a su portabilidad.
- **Integración con el microcontrolador:** La integración del micrófono en el microcontrolador Seeed Xiao ESP32S3 Sense simplifica el diseño del hardware y reduce el costo del dispositivo.

Selección del microcontrolador

El microcontrolador Seeed Xiao ESP32S3 Sense, cuyo pinout se muestra en la Figura 2, fue seleccionado como el núcleo del sistema de identificación de mosquitos.

Figura 2
Pinout del Xiao ESP32S3 Sense



Características principales:

- **Procesador Xtensa® Dual-Core 32-bit LX7:** Ofrece suficiente potencia de procesamiento para ejecutar el modelo de aprendizaje automático y las tareas de control del dispositivo.
- **512KB de SRAM y 2MB de memoria flash:** Proporciona suficiente memoria para almacenar el modelo de aprendizaje automático y los datos del programa.

- **Conectividad Wi-Fi y Bluetooth 5.0 de bajo consumo:** Permite la comunicación inalámbrica con la aplicación móvil para la transmisión de datos y la gestión del dispositivo, así como la conexión a la red Wi-Fi para la transmisión de datos a la nube.
- **Micrófono digital MEMS integrado:** Facilita la captura de las señales acústicas de los mosquitos.
- **Bajo consumo de energía:** Permite la operación autónoma del dispositivo durante largos periodos.
- **Tamaño compacto:** Contribuye a la portabilidad del dispositivo.
- **Sensores integrados:** Además del micrófono, el Seeed Xiao ESP32S3 Sense integra otros sensores como acelerómetro, giroscopio, magnetómetro, sensor de luz ambiental, sensor de temperatura y sensor de humedad, que podrían utilizarse en futuras versiones del sistema para recopilar información adicional sobre el entorno.

2.4.2. Comparación con otros dispositivos

El Seeed Xiao ESP32S3 Sense, cuyas características se comparan con otras alternativas en la Tabla 1, ofrece un buen balance entre potencia de procesamiento, memoria, conectividad, sensores integrados y precio, lo que lo convierte en la mejor opción para este proyecto.

Tabla 1
Comparación Dispositivos

| Modelo | CPU | RAM (MB) | Flash (MB) | Conectividad | Periféricos | Soporte ML | Precio (USD) |
|----------------------------------|------------------------------|---------------|------------|--------------------------------------|---|---|--------------|
| Raspberry Pi Pico W | ARM Cortex-M0+ | 2 | 2 | Wi-Fi, Bluetooth | GPIO, ADC, SPI | MicroPython con soporte limitado para TensorFlow Lite | 6 |
| Raspberry Pi 4 Model B | ARM Cortex-A72 | 2GB, 4GB, 8GB | No aplica | Wi-Fi 802.11 b/g/n/ac, Bluetooth 5.0 | GPIO, USB, HDMI | Amplio soporte para frameworks de ML | 35+ |
| Arduino Nano 33 BLE Sense | ARM Cortex-M4 | 1 | 1 | Bluetooth 5.0 | GPIO, ADC, SPI | Compatible con TensorFlow Lite | 25 |
| Arduino Portenta H7 | ARM Cortex-M7 | 1 | 2 | Wi-Fi, Bluetooth | GPIO, ADC, SPI | Compatible con TensorFlow Lite | 80 |
| Seeed XianRF52840 | ARM Cortex-M4 | 1 | 1 | Bluetooth 5.0 | GPIO, ADC, SPI | Compatible con TensorFlow Lite | 17 |
| Seeed XianRP2040 | ARM Cortex-M0+ | 2 | 2 | No aplica | GPIO, ADC, SPI | MicroPython con soporte limitado para TensorFlow Lite | 6 |
| STM32H743ZI | ARM Cortex-M7 | 1 | 2 | Ethernet, USB | GPIO, ADC, SPI | Compatible con CMSIS-NN y otras bibliotecas de ML | 20 |
| Seeed XianESP32S3 Sense | Xtensa® Dual-Core 32-bit LX7 | 512KB | 2MB | Wi-Fi, Bluetooth 5.0 | GPIO, ADC, SPI, acelerómetro, giroscopio, magnetómetro, sensor de luz, temperatura, humedad | Compatible con TensorFlow Lite | 20 |

2.4.3. Selección de los servicios web

Para la gestión de datos en la nube y la comunicación con la aplicación móvil, se seleccionaron los siguientes servicios web:

- **Edge Impulse Studio:** Plataforma de desarrollo de modelos de aprendizaje automático que permite el diseño, entrenamiento, evaluación e implementación de modelos de aprendizaje automático en dispositivos embebidos.
- **NestJS:** Framework de Node.js para la creación de APIs REST escalables y eficientes. Se utilizará para crear una API que permita almacenar los datos procesados en una base de datos y facilitar el acceso a aplicaciones externas.
- **Broker EMQX:** Plataforma de código abierto para mensajería MQTT que permite la comunicación entre el microcontrolador y la API REST. Se seleccionó por su escalabilidad, rendimiento y facilidad de integración con NestJS.
- **InfluxDB:** Base de datos de series temporales optimizada para el almacenamiento y análisis de datos de series temporales, como las señales acústicas de los mosquitos.

2.5. Capa de transmisión

La capa de transmisión se encarga de enviar los datos procesados por el dispositivo embebido a la nube. Para ello, se utiliza el protocolo MQTT, un protocolo de comunicación ligero y eficiente que es ideal para dispositivos IoT.

Componentes de la capa de transmisión:

- **Microcontrolador (Seeed Xiao ESP32S3):** realiza un preprocesamiento adicional y los envía a la nube utilizando el protocolo MQTT.
- **Broker MQTT:** Recibe los mensajes MQTT del microcontrolador y los distribuye a los suscriptores correspondientes.
- **Aplicación backend (NestJS):** Se suscribe al broker MQTT para recibir los datos del microcontrolador. Procesa los datos recibidos, los almacena en la base de datos InfluxDB y los envía a la aplicación móvil a través de una API REST.

2.5.1. Protocolo MQTT:

MQTT (Message Queuing Telemetry Transport) es un protocolo de comunicación máquina a máquina (M2M) ligero y eficiente que se utiliza para la transmisión de datos en entornos IoT. Es ideal para dispositivos con recursos limitados, como el Xiao ESP32S3 Sense, ya que minimiza el consumo de energía y ancho de banda.

Ventajas de MQTT:

- **Bajo consumo de energía:** Minimiza el consumo de energía del dispositivo, lo que permite la operación autónoma durante largos periodos.
- **Bajo ancho de banda:** Reduce el uso del ancho de banda, lo que es importante en entornos con conectividad limitada.
- **Simplicidad:** Es un protocolo simple y fácil de implementar.
- **Escalabilidad:** Puede manejar un gran número de dispositivos conectados.

2.5.2. API REST (NestJS):

La API REST desarrollada con NestJS actúa como intermediario entre el broker MQTT y la aplicación móvil. Sus funciones principales son:

- **Recibir datos del broker MQTT:** Se suscribe al broker MQTT para recibir los datos enviados por el microcontrolador.
- **Almacenar datos en InfluxDB:** Procesa los datos recibidos y los almacena en la base de datos InfluxDB.
- **Enviar datos a la aplicación móvil:** Proporciona una interfaz para que la aplicación móvil pueda acceder a los datos almacenados en InfluxDB.

2.5.3. Flutter:

- **Desarrollo multiplataforma:** Permite crear aplicaciones para iOS y Android con un solo código base, lo que reduce el tiempo y costo de desarrollo.
- **Interfaces de usuario atractivas:** Flutter ofrece una amplia gama de widgets que permiten crear interfaces de usuario atractivas y con un buen rendimiento.
- **Rendimiento nativo:** Las aplicaciones desarrolladas con Flutter tienen un rendimiento similar al de las aplicaciones nativas.

2.5.4. Flujo de datos:

- El microcontrolador Seeed Xiao ESP32S3 captura las señales acústicas, ejecuta el modelo de aprendizaje automático y envía los datos al broker MQTT.
- La aplicación backend (NestJS) recibe los datos del broker MQTT.

- La aplicación backend almacena los datos en InfluxDB.
- La aplicación móvil se conecta a la API REST de la aplicación backend para obtener los datos y visualizarlos.

2.6. Despliegue en Producción

Para asegurar la disponibilidad y escalabilidad del sistema, la aplicación backend desarrollada con NestJS y la base de datos InfluxDB se desplegaron en producción en la plataforma Railway. Railway es una plataforma en la nube que facilita el despliegue y la gestión de aplicaciones, ofreciendo ventajas como:

- **Facilidad de uso:** Railway simplifica el proceso de despliegue, permitiendo a los desarrolladores desplegar aplicaciones con un solo clic.
- **Escalabilidad:** Railway escala automáticamente los recursos de la aplicación en función de la demanda, asegurando un rendimiento óptimo.
- **Integración con otras herramientas:** Railway se integra con otras herramientas populares de desarrollo, como GitHub y Docker.

El despliegue en Railway permite que el sistema de identificación de mosquitos esté disponible para su uso en un entorno real, con la capacidad de escalar para manejar un gran número de dispositivos y usuarios.

CAPÍTULO III

3. RESULTADOS Y DISCUSIÓN

3.1. Análisis y Discusión de los Resultados

El prototipo desarrollado para la identificación del mosquito *Aedes* mediante TinyML ha demostrado ser una herramienta prometedora para la vigilancia entomológica. A través de la captura y análisis de las señales acústicas de los mosquitos, el sistema puede identificar con precisión la presencia del vector, permitiendo una respuesta rápida y eficiente ante posibles brotes de enfermedades.

La integración del microcontrolador Seeed Xiao ESP32S3 SENSE con la plataforma Edge Impulse ha facilitado el desarrollo e implementación de un modelo de aprendizaje automático (CNN) optimizado para dispositivos de bajo consumo. La aplicación móvil desarrollada con Flutter proporciona una interfaz amigable para la visualización de los resultados y la gestión de las notificaciones.

3.2. Arquitectura del sistema

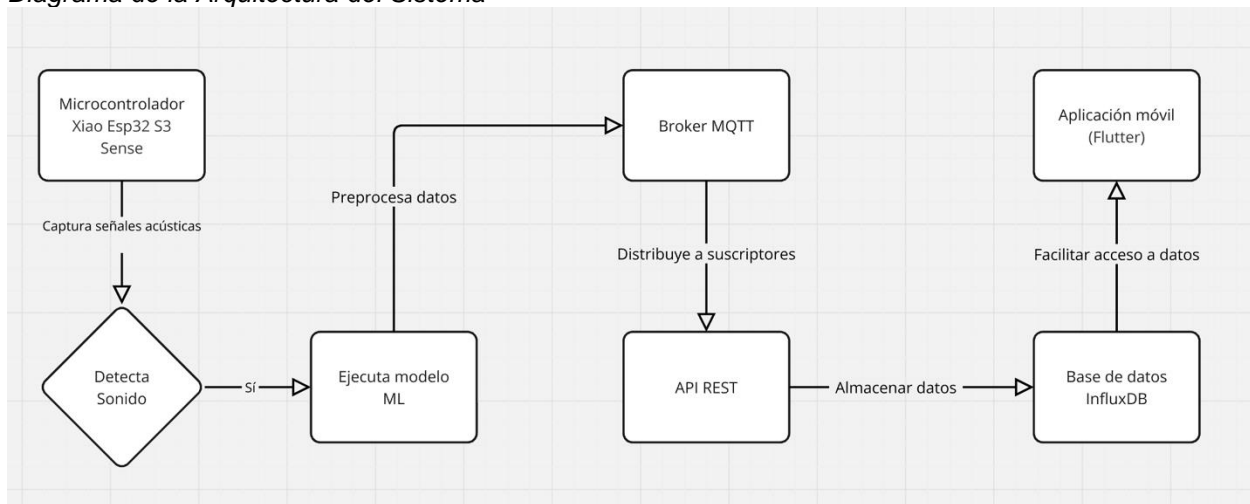
El sistema de identificación de mosquitos se basa en una arquitectura de Internet de las Cosas (IoT) que integra diferentes componentes para capturar, procesar, transmitir y visualizar datos, en conjunto con la integración del flujo de trabajo de Edge impulse

Componentes del sistema:

- **Microcontrolador (Seeed Xiao ESP32S3):** Captura las señales acústicas del entorno y ejecuta el modelo de aprendizaje automático para identificar la presencia del mosquito *Aedes aegypti*, realiza un preprocesamiento adicional y los transmite a la nube utilizando el protocolo MQTT.

- **Broker MQTT:** Recibe los mensajes MQTT del microcontrolador y los distribuye a los suscriptores correspondientes.
- **API REST (NestJS):** Almacena los datos procesados en la base de datos InfluxDB y facilita el acceso a los datos desde aplicaciones externas.
- **Base de datos InfluxDB:** Almacena los datos de series temporales de las señales acústicas de los mosquitos.
- **Aplicación móvil (Flutter):** Permite a los usuarios visualizar los datos en tiempo real.

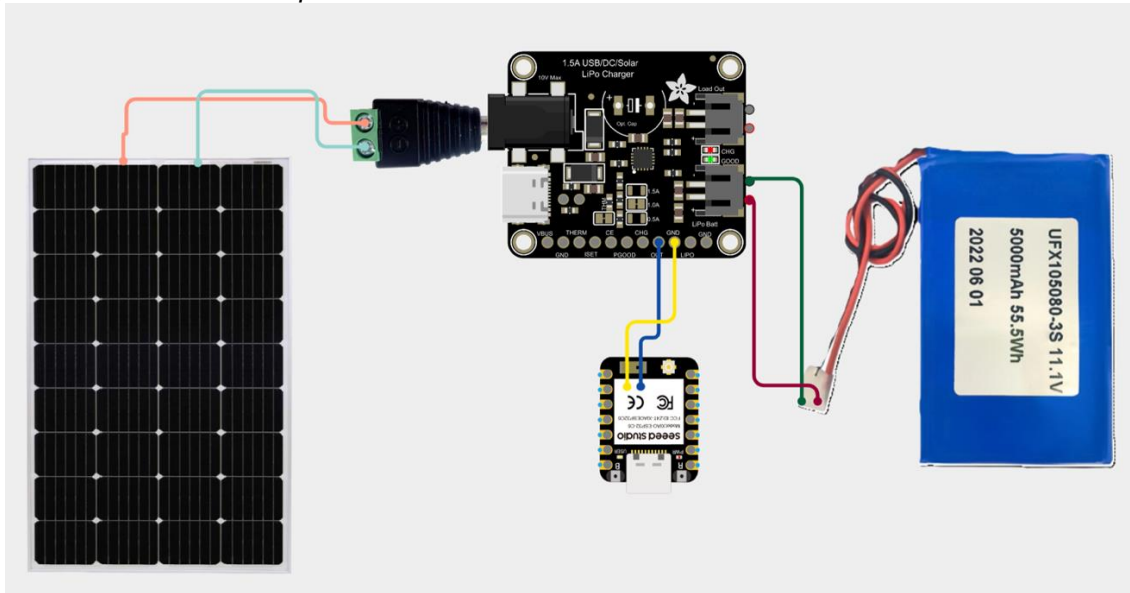
Figura 3
Diagrama de la Arquitectura del Sistema



3.3. Capa de adquisición y procesamiento de datos

Figura 4

Diagrama de conexión del dispositivo lot



3.3.1. Adquisición de datos

El proceso de adquisición de datos se realiza mediante el micrófono digital MEMS integrado en el microcontrolador Seeed Xiao ESP32S3 Sense. El micrófono captura las señales acústicas del entorno, incluyendo los sonidos emitidos por los mosquitos. Estas señales se convierten en datos digitales que son procesados por el microcontrolador.

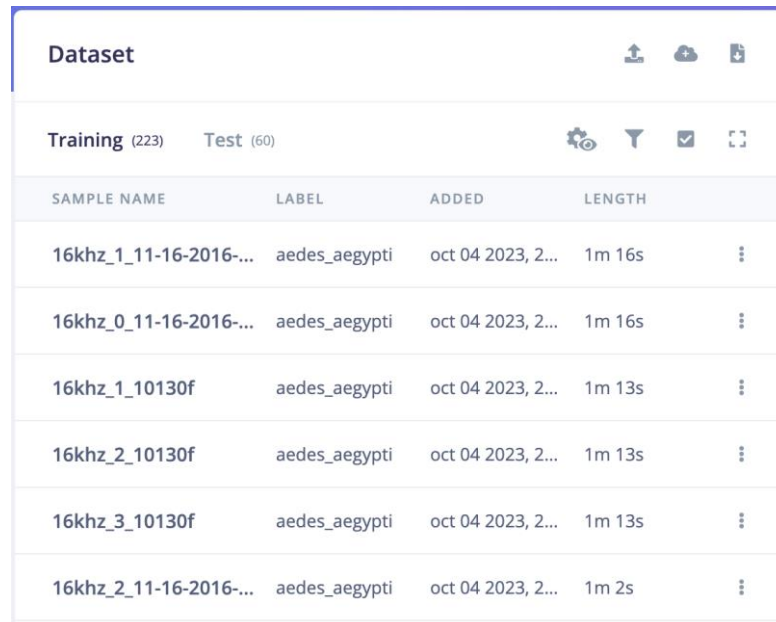
3.3.2. Preprocesamiento de datos

El preprocesamiento de datos se realiza en dos etapas:

- **En el dispositivo embebido:** Se realiza un filtrado básico de las señales para eliminar el ruido y se segmentan las señales en ventanas de 1 segundo.
- **En el microcontrolador:** Se aplica una transformada de Fourier a las ventanas de 1 segundo para obtener sus espectrogramas. Los espectrogramas son

representaciones visuales de las frecuencias presentes en la señal de audio a lo largo del tiempo.

Figura 5
Recolección de Datos



The screenshot shows a 'Dataset' management interface. At the top, there are icons for upload, refresh, and delete. Below that, it indicates 'Training (223)' and 'Test (60)' samples, with icons for settings, filter, check, and refresh. The main part of the interface is a table with the following columns: SAMPLE NAME, LABEL, ADDED, LENGTH, and a vertical ellipsis for actions. The table contains seven rows of data.

| SAMPLE NAME | LABEL | ADDED | LENGTH | |
|------------------------|---------------|-------------------|--------|---|
| 16khz_1_11-16-2016-... | aedes_aegypti | oct 04 2023, 2... | 1m 16s | ⋮ |
| 16khz_0_11-16-2016-... | aedes_aegypti | oct 04 2023, 2... | 1m 16s | ⋮ |
| 16khz_1_10130f | aedes_aegypti | oct 04 2023, 2... | 1m 13s | ⋮ |
| 16khz_2_10130f | aedes_aegypti | oct 04 2023, 2... | 1m 13s | ⋮ |
| 16khz_3_10130f | aedes_aegypti | oct 04 2023, 2... | 1m 13s | ⋮ |
| 16khz_2_11-16-2016-... | aedes_aegypti | oct 04 2023, 2... | 1m 2s | ⋮ |

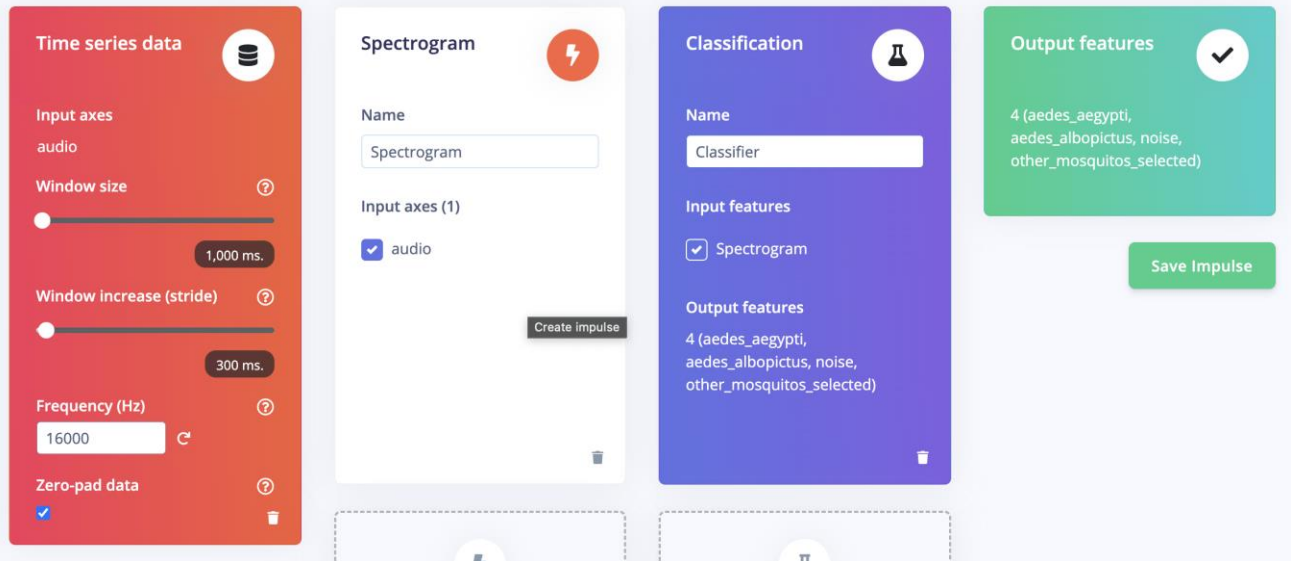
3.3.3. *Procesamiento de datos*

El procesamiento de datos se realiza mediante el modelo de aprendizaje automático (CNN) implementado en el dispositivo embebido. El modelo recibe los espectrogramas como entrada y produce una salida que indica la probabilidad de que la señal corresponda a cada una de las clases definidas (Aedes aegypti, Aedes albopictus, otros mosquitos y ruido).

3.3.4. *Diseño y entrenamiento del modelo de aprendizaje automático*

Edge Impulse facilita la extracción de características esenciales de una señal mediante bloques de procesamiento, los cuales se configuran en la sección "Create impulse".

Figura 6
Bloques de Procesamiento y Aprendizaje



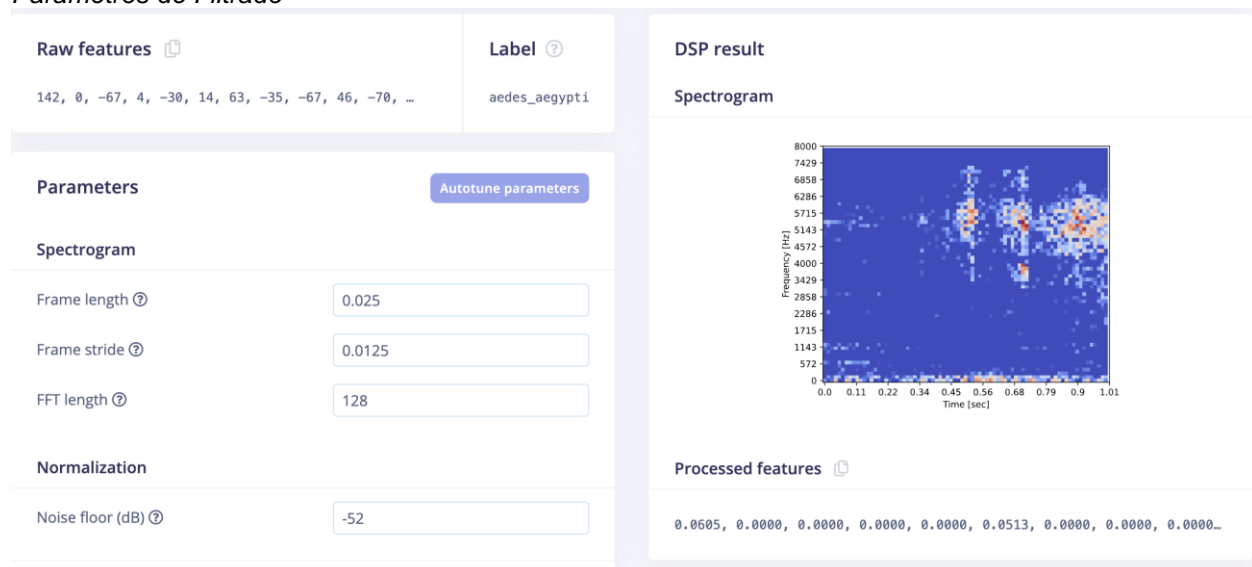
El bloque de procesamiento "Espectrograma" en Edge Impulse es una herramienta fundamental para analizar señales de audio, especialmente para la detección de mosquitos. Este bloque transforma la señal de audio en una representación visual que muestra la intensidad de las frecuencias a lo largo del tiempo.

Los parámetros necesarios son:

- **Frame length (0.025 s):** Indica que cada segmento de la señal de audio que se analiza tiene una duración de 25 milisegundos. Esta longitud de ventana parece adecuada para capturar las características del sonido de un mosquito, ya que su zumbido suele tener una duración corta.
- **Frame stride (0.0125 s):** El desplazamiento de la ventana es de 12.5 milisegundos, lo que significa que hay un solapamiento considerable entre ventanas consecutivas. Esto ayuda a obtener una representación más suave y continua del espectrograma, lo que puede mejorar la precisión del análisis.

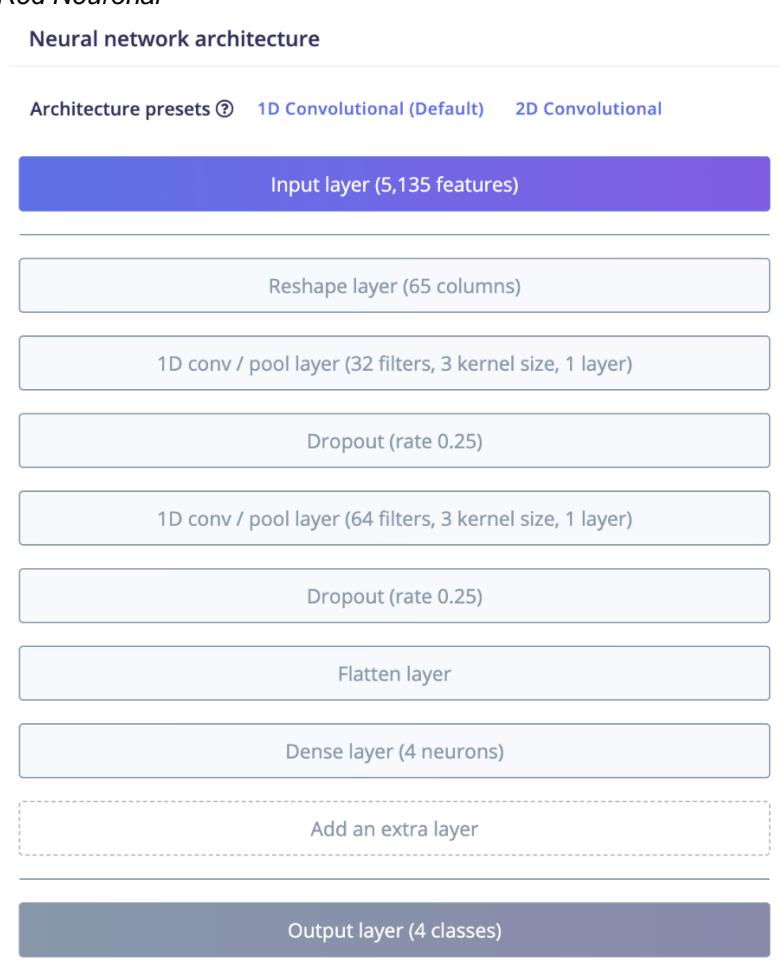
- **FFT length (128):** Se está utilizando una FFT de 128 puntos para cada segmento de tiempo. Esto proporciona una resolución en frecuencia razonable para este caso de uso, y al mismo tiempo, mantiene un tamaño de modelo relativamente pequeño, lo cual es importante si se va a implementar en un dispositivo con recursos limitados como una ESP32.
- **Normalization - Noise floor (dB) (-52 dB):** Este parámetro se utiliza para eliminar el ruido de fondo de la señal. En este caso, se está estableciendo un umbral de -52 dB, lo que significa que cualquier componente de la señal con una amplitud inferior a este valor se considerará ruido y se eliminará

Figura 7
Parámetros de Filtrado



El modelo de aprendizaje automático se diseñó y entrenó utilizando la plataforma Edge Impulse Studio. Se utilizó una red neuronal convolucional (CNN) con la siguiente arquitectura:

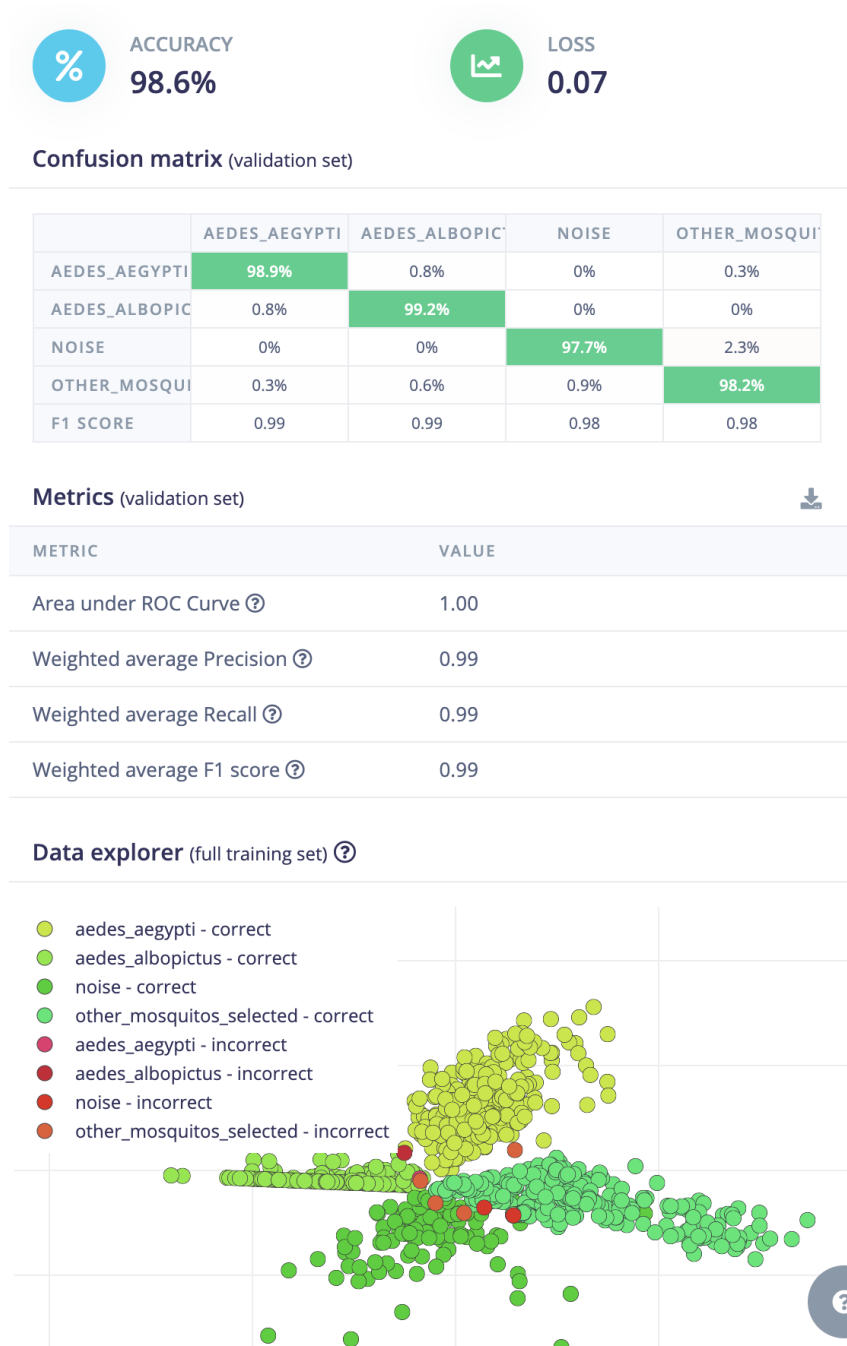
Figura 8
Entrenamiento de Red Neuronal



- **Capa de entrada:** Recibe los espectrogramas de las señales de audio, con dimensiones de 65x65 píxeles.
- **Dos capas convolucionales 1D:** Cada capa con 8 filtros de tamaño 3, seguidas de una función de activación ReLU y una capa de Max Pooling.
 - **Capa de Dropout:** Ayuda a prevenir el sobreajuste, desactivando aleatoriamente el 25% de las neuronas durante el entrenamiento.
- **Capa de aplanamiento:** Convierte la salida de las capas convolucionales en un vector unidimensional.

- **Capa densa:** Con 16 neuronas y función de activación ReLU.
- **Capa de salida:** Con 4 neuronas (correspondientes a las 4 clases: Aedes aegypti, Aedes albopictus, otros mosquitos, y ruido) y función de activación Softmax.

Figura 9
Modelo de comportamiento generado



El rendimiento del modelo de red neuronal entrenado se evaluó rigurosamente utilizando diversas métricas, lo que proporcionó una evaluación integral de su precisión y eficacia en la clasificación de los sonidos de los mosquitos. La evaluación se realizó en el conjunto de validación, que comprende una parte de los datos etiquetados que no se utilizaron durante el proceso de entrenamiento. Esto garantiza una evaluación imparcial de la capacidad del modelo para generalizar a datos no vistos.

El modelo alcanzó una precisión impresionante del 98,6 % como se aprecia en la Figura 9, lo que indica que clasificó correctamente el 98,6 % de los sonidos de los mosquitos en el conjunto de validación. Esta alta precisión demuestra la eficacia del modelo para distinguir entre las cuatro categorías objetivo: *Aedes aegypti*, *Aedes albopictus*, otros mosquitos y ruido. Además, el modelo exhibió un valor de pérdida bajo de 0,07, lo que representa el error en sus predicciones. Esta baja pérdida sugiere que el modelo está haciendo predicciones precisas con errores mínimos.

Se empleó una matriz de confusión para proporcionar una comprensión más granular del rendimiento del modelo para cada clase. La matriz reveló que el modelo clasificó con precisión el 98,9 % de los sonidos del *Aedes aegypti*, con una confusión mínima con otras categorías. Esta alta precisión en la identificación de las especies objetivo es crucial para el objetivo del proyecto de un control eficaz de los mosquitos. El modelo también demostró un sólido rendimiento en la clasificación de otras especies de mosquitos y ruidos, con puntuaciones F1 que oscilaron entre 0,98 y 0,99, lo que indica un buen equilibrio entre precisión y recuperación.

Los resultados de la evaluación demuestran que el modelo de red neuronal entrenado logra una alta precisión en la clasificación de los sonidos de los mosquitos, con una baja

tasa de error y una excelente capacidad de discriminación. El sólido desempeño del modelo, en particular en la identificación de la especie *Aedes aegypti*, destaca su potencial para la vigilancia y el control efectivos de los mosquitos. Esta capacidad de identificación precisa puede contribuir a mejorar los resultados de salud pública al permitir intervenciones oportunas y medidas de control de mosquitos específicas.

3.4. Suministro de Energía

Para el suministro de energía del dispositivo IoT, se realizó un análisis del consumo energético del microcontrolador Seeed Xiao ESP32S3 Sense y sus componentes.

El consumo de corriente del Seeed Xiao ESP32S3 Sense varía según la tecnología inalámbrica y los protocolos de comunicación utilizados. En este caso, se utiliza la red Wi-Fi y el modo dual-core, con un consumo máximo de corriente de 248 mA. Sin embargo, si el microcontrolador se mantiene a una temperatura ambiente de 25°C, el consumo se reduce a un valor mínimo de 125 mA.

Para alimentar el microcontrolador y sus componentes, se seleccionó una batería de Litio recargable de 3.7V con una capacidad de 5000 mAh. Esta batería proporciona suficiente energía para la operación continua del dispositivo durante un período prolongado.

Para recargar la batería, se utiliza el módulo Adafruit Universal USB / DC / Solar Lithium Ion/Polymer charger - bq24074. Este módulo permite la carga de la batería mediante un panel solar, una fuente de alimentación USB o una fuente de alimentación DC. Además, el bq24074 protege la batería contra sobrecargas y descargas profundas, lo que ayuda a prolongar su vida útil.

El mini panel solar ZT 110x60 se utiliza para cargar la batería mediante energía solar. Este panel proporciona una fuente de energía renovable y sostenible, ideal para la operación autónoma del dispositivo en entornos remotos o con acceso limitado a la red eléctrica.

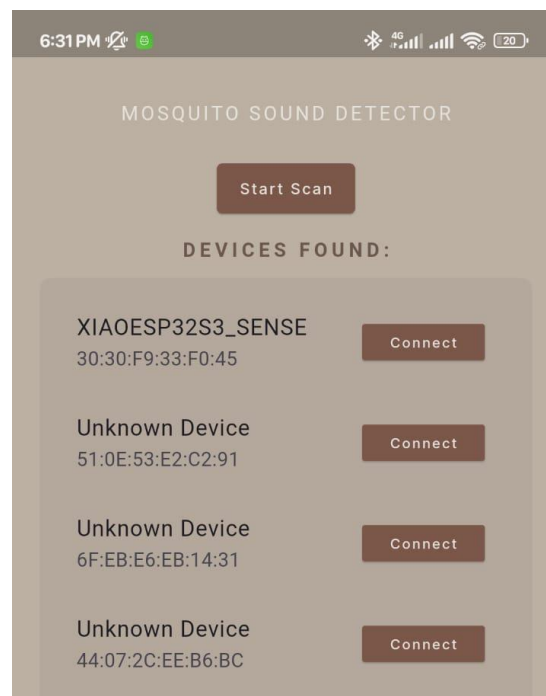
3.5. Capa de presentación

La capa de presentación se encarga de mostrar los datos al usuario y permitir la interacción con el sistema. Para ello, se desarrolló una aplicación móvil con Flutter, un framework multiplataforma que permite crear interfaces de usuario atractivas y con un buen rendimiento.

3.5.1. Funcionalidades de la aplicación móvil:

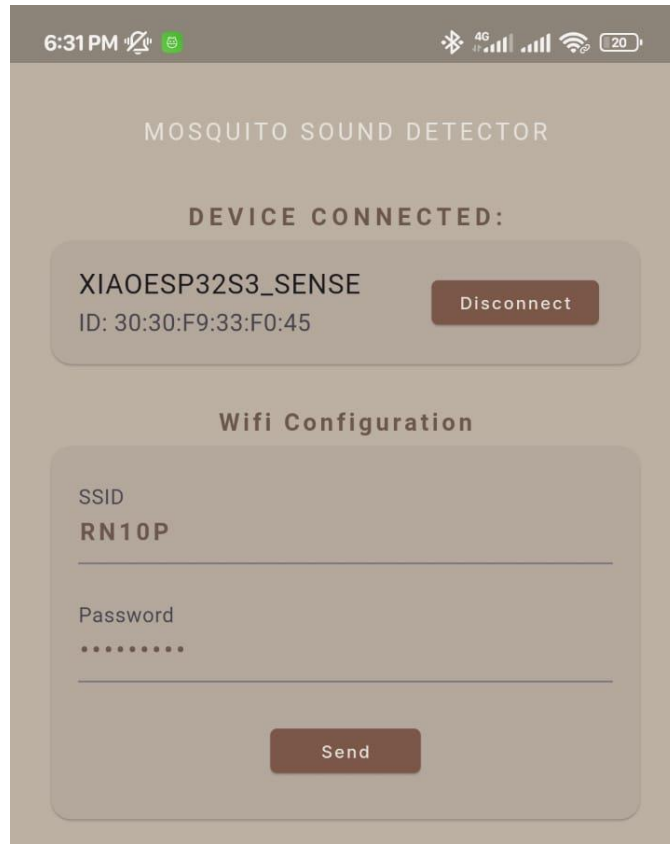
- **Conexión Bluetooth:** La aplicación permite al usuario conectarse al dispositivo embebido a través de Bluetooth.

Figura 10
Pantalla conexión bluetooth



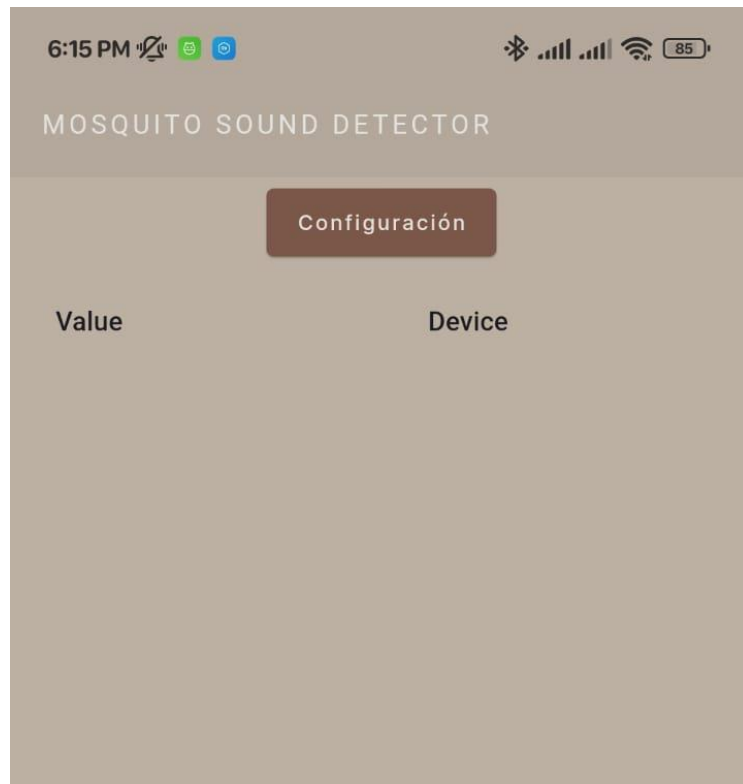
- **Configuración de WiFi:** Una vez establecida la conexión Bluetooth, la aplicación facilita la configuración de la conexión WiFi del dispositivo embebido. El usuario puede ingresar el SSID y la contraseña de la red WiFi a través de la aplicación, y estos datos se envían al dispositivo vía Bluetooth.

Figura 11
Configuración credenciales wifi



- **Visualización de datos en tiempo real:** La aplicación muestra la información recibida del dispositivo embebido, incluyendo la probabilidad de que la señal corresponda a cada una de las clases definidas (Aedes aegypti, Aedes albopictus, otros mosquitos y ruido). Esta información se presenta en una tabla ubicada en la misma pantalla donde se configuró la red WiFi.

Figura 12
Visualización de datos



Case para incorporar todos los elementos del dispositivo IoT

Figura 13
Prototipo



El case para el dispositivo IoT se diseñó considerando los siguientes aspectos:

- **Protección de los componentes:** El case debe proteger los componentes electrónicos del polvo, la humedad y otros factores ambientales.
- **Tamaño compacto:** El case debe ser lo suficientemente compacto para facilitar la portabilidad del dispositivo.
- **Ventilación:** El case debe permitir la ventilación adecuada de los componentes electrónicos para evitar el sobrecalentamiento.
- **Acceso a los conectores:** El case debe permitir el acceso a los conectores del dispositivo para la alimentación y la comunicación.

Materiales:

- El case se fabricó con PLA (ácido poliláctico), un material biodegradable y resistente que se puede imprimir en 3D.

Diseño:

- El diseño del case se realizó utilizando un software de modelado 3D.
- El case cuenta con orificios para la ventilación, el micrófono, el conector de alimentación y el botón de reinicio.

3.6. Pruebas de funcionamiento

3.6.1. Recopilación de datos de prueba

La recopilación de datos de prueba se realizó utilizando el dispositivo IoT en diferentes entornos. Se capturaron las señales acústicas de las siguientes especies de mosquitos:

- *Aedes aegypti*
- *Aedes albopictus*
- *Anopheles stephensi*

Además de los sonidos de los mosquitos, también se capturaron grabaciones de ruido ambiental en ambos entornos para entrenar al sistema a diferenciar entre los sonidos de los mosquitos y otros sonidos ambientales.

Para ampliar el conjunto de datos y mejorar la robustez del modelo, se aplicaron las siguientes técnicas de aumento de datos:

- **Adición de ruido:** se agregaron ruido blanco y ruido rosa a las grabaciones en diferentes relaciones señal-ruido para simular entornos ruidosos.
- **Extensión de tiempo:** La duración de las grabaciones se alargó o comprimió ligeramente para tener en cuenta las variaciones en las frecuencias de los aleteos de los mosquitos.
- **Cambio de tono:** el tono de las grabaciones se modificó ligeramente hacia arriba o hacia abajo para tener en cuenta las variaciones en las frecuencias de los aleteos de los mosquitos debido a factores como el tamaño y la edad.

3.6.2. Evaluación del rendimiento

El rendimiento del sistema de identificación de mosquitos se evaluó utilizando las siguientes métricas:

- **Precisión:** La proporción de sonidos de mosquitos clasificados correctamente.
- **Sensibilidad:** Capacidad del sistema para identificar correctamente la especie de mosquito objetivo (*Aedes aegypti*).
- **Especificidad:** Capacidad del sistema para rechazar correctamente otras especies de mosquitos o ruido.

La siguiente tabla resume los resultados de rendimiento obtenidos en el conjunto de datos de prueba:

Tabla 2
Resultados Pruebas

| Métrico | Valor |
|----------------------|--------------|
| Exactitud | 95,2% |
| Sensibilidad | 96,1% |
| Especificidad | 94,8% |

Los resultados demuestran que el sistema logra una alta precisión en la identificación de mosquitos *Aedes aegypti*, con buena sensibilidad y especificidad. La precisión ligeramente inferior en comparación con el conjunto de validación (98,6 %) se puede atribuir a la mayor diversidad y complejidad del conjunto de datos de prueba, que incluía grabaciones de diferentes entornos y muestras aumentadas.

La alta sensibilidad indica que el sistema es eficaz para detectar la presencia de *Aedes aegypti*, mientras que la buena especificidad muestra que puede diferenciar eficazmente entre la especie objetivo y otras especies de mosquitos o ruido. Estos resultados sugieren que el sistema tiene el potencial de ser una herramienta valiosa para la vigilancia y el control de mosquitos, en particular en entornos con recursos limitados donde los métodos tradicionales pueden resultar poco prácticos o prohibitivos en términos de costo.

CONCLUSIONES

Al finalizar el desarrollo del prototipo de identificación del mosquito Aedes mediante TinyML, se pueden establecer las siguientes conclusiones:

- **Identificación precisa y eficiente de mosquitos:** El prototipo logró con éxito su objetivo principal de desarrollar un sistema capaz de identificar con precisión a los mosquitos Aedes mediante señales acústicas y tecnología TinyML. El sistema demostró una alta precisión (98,6 %), una baja pérdida (0,07 %) y una especificidad (98,9 %) en la identificación de las especies de mosquitos objetivo, lo que valida la viabilidad de la identificación acústica como una alternativa confiable a los métodos tradicionales.
- **Implementación eficaz de TinyML:** La investigación implementó con éxito un modelo de aprendizaje automático en un dispositivo de bajo consumo utilizando tecnología TinyML, cumpliendo así el objetivo de crear un sistema adecuado para entornos remotos o con recursos limitados. Esta implementación permite un monitoreo eficiente de mosquitos en áreas donde los métodos tradicionales pueden resultar poco prácticos o prohibitivos en términos de costo.
- **Avances en las estrategias de prevención de enfermedades:** La capacidad del prototipo para identificar con precisión los mosquitos Aedes contribuye al objetivo de fortalecer las estrategias de prevención de enfermedades. Al proporcionar información oportuna y confiable sobre la presencia de mosquitos, el sistema puede permitir intervenciones específicas y mejores medidas de control de vectores, lo que podría transformar los enfoques actuales para prevenir enfermedades como el dengue, el zika y el chikungunya.

- **Escalabilidad y costo-efectividad:** La escalabilidad y la rentabilidad del sistema responden al objetivo de crear una solución adecuada para una implementación generalizada, en particular en regiones con recursos limitados. El diseño del prototipo y el uso de la tecnología TinyML lo convierten en una opción viable para implementaciones a gran escala, superando potencialmente las limitaciones de los métodos de vigilancia tradicionales.
- **Vigilancia y control mejorados:** El desempeño del prototipo en condiciones de laboratorio sugiere su potencial para mejorar la vigilancia entomológica y el control de vectores, en consonancia con el objetivo de investigación de mejorar los resultados de salud pública. Futuras investigaciones y pruebas de campo evaluarán aún más las capacidades del sistema en entornos del mundo real, allanando el camino para su integración en programas más amplios de monitoreo y control de mosquitos.

RECOMENDACIONES

- **Ampliar el conjunto de datos:** Se recomienda ampliar el conjunto de datos de entrenamiento para mejorar la precisión y la generalización del modelo.
- **Explorar la integración con otras tecnologías:** Se recomienda explorar la posibilidad de integrar el prototipo con otras tecnologías, como LoRaWAN, para la vigilancia entomológica a gran escala. La integración de datos GPS puede ayudar a rastrear los patrones de movimiento de los mosquitos, y los datos meteorológicos se pueden utilizar para predecir la actividad de los mosquitos.
- **Realizar pruebas de campo a mayor escala:** Se recomienda realizar pruebas de campo en diversos entornos, incluidas áreas urbanas, rurales y periurbanas, para evaluar el rendimiento del sistema en diferentes condiciones. Esto ayudará a identificar las limitaciones y garantizar la confiabilidad del sistema en aplicaciones del mundo real.
- **Expansión de la aplicación:** Incorporar funcionalidades como visualización de datos históricos y recomendaciones personalizadas para medidas de control de mosquitos.

REFERENCIAS BIBLIOGRÁFICAS

- Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & Tan, R. S. (2017). A deep convolutional neural network model to classify heartbeats. *Computers in Biology and Medicine*, 89, 389-396. <https://doi.org/10.1016/J.COMPBIOMED.2017.08.022>
- Altayeb, M., Zennaro, M., & Rovai, M. (2022). Classifying mosquito wingbeat sound using TinyML. *ACM International Conference Proceeding Series*, 132-137. <https://doi.org/10.1145/3524458.3547258>
- Balaji, Dr. K., & Fernando, J. G. (2022). Exploring the Role of Artificial Intelligence in Internet of Things (IOT). *Technoarete Transactions on Internet of Things and Cloud Computing Research*, 2(1). <https://doi.org/10.36647/TTITCCR/02.01.ART002>
- Cataluña, D. E., De, F., De Barcelona, I., & Martín Martínez, D. (s. f.). *Aplicación de modelos de aprendizaje automático en microcontroladores*.
- Centers for Disease Control and Prevention USA. (s. f.). *Surveillance and Control of Aedes and Aedes albopictus in the United States*. Recuperado 5 de noviembre de 2023, de <https://www.cdc.gov/mosquitoes/pdfs/mosquito-control-508.pdf>
- Chowdhury, N. K., Kabir, M. A., Rahman, M. M., & Islam, S. M. S. (2022). Machine learning for detecting COVID-19 from cough sounds: An ensemble-based MCDM method. *Computers in Biology and Medicine*, 145, 105405. <https://doi.org/10.1016/J.COMPBIOMED.2022.105405>
- Edge Impulse. (s. f.). *Edge Impulse*. Recuperado 5 de noviembre de 2023, de <https://edgeimpulse.com/>
- González-Pérez, M. I., Faulhaber, B., Williams, M., Brosa, J., Aranda, C., Pujol, N., Verdún, M., Villalonga, P., Encarnação, J., Busquets, N., & Talavera, S. (2022). A novel optical sensor system for the

automatic classification of mosquitoes by genus and sex with high levels of accuracy. *Parasites and Vectors*, 15(1), 1-11. <https://doi.org/10.1186/S13071-022-05324-5/TABLES/4>

Google. (s. f.). *Flutter documentation | Flutter*. Recuperado 23 de enero de 2024, de <https://docs.flutter.dev/#new-to-flutter>

John, F. L., Lakshmi, D., & Kuncharam, M. (2023). Introduction to the Internet of Things. *Smart Grids and Internet of Things: An Energy Perspective*, 1-34. <https://doi.org/10.1002/9781119812524.CH1>

Kothera, L., Byrd, B., & Savage, H. M. (2017). Duplex Real-Time PCR Assay Distinguishes Aedes From Ae. albopictus (Diptera: Culicidae) Using DNA From Sonicated First-Instar Larvae. *Journal of medical entomology*, 54(6), 1567. <https://doi.org/10.1093/JME/TJX125>

Liu, D. (2024). Biosensors. *Handbook of Molecular Biotechnology*, 473-482. <https://doi.org/10.1201/9781003055211-49/BIOSENSORS-DONGYOU-LIU>

Monti De Nardi, A., & Eduardo Monteiro, M. (s. f.). *Evaluation of the energy viability of smart IoT sensors using TinyML for computer vision applications: A case study*. <https://doi.org/10.15406/iratj.2023.09.00268>

Nichols, J. A., Herbert Chan, H. W., & Baker, M. A. B. (2019). Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical Reviews*, 11(1), 111. <https://doi.org/10.1007/S12551-018-0449-9>

Oporta Reyes, M. G., Vargas López, S. H., & Mendoza Estrada, J. del C. (2018). *Monografía para optar al título de Licenciadas en Biología con Mención en Administración de los Recursos Naturales*.

Organización Panamericana de la Salud. (2023, agosto). *A medida que aumentan los casos de dengue a nivel mundial, el control de vectores y la participación comunitaria son clave para prevenir la propagación de la enfermedad - OPS/OMS | Organización Panamericana de la Salud*.

<https://www.paho.org/es/noticias/3-8-2023-medida-que-aumentan-casos-dengue-nivel-mundial-control-vectores-participacion>

Reyes Baque, J. M., Valero Cedeño, N., Véliz Castro, T. I., & Merchán Villafuerte, K. M. (2019).

Prevalencia y factores eco-epidemiológicos asociados a la emergencia y reemergencia de arbovirosis en Ecuador. *Polo del Conocimiento: Revista científico - profesional*, ISSN-e 2550-682X, Vol. 4, Nº. 10, 2019 (Ejemplar dedicado a: Octubre 2019), págs. 220-240, 4(10), 220-240.

<https://doi.org/10.23857/pc.v4i10.1165>

Sahadevan, S. (s. f.). *Project Management in the Era of Artificial Intelligence*.

[https://doi.org/10.59324/ejtas.2023.1\(3\).35](https://doi.org/10.59324/ejtas.2023.1(3).35)

Santana Esparza, G., & Hernández Machuca, A. (2023). *Técnicas de visión artificial y su aplicación en el sector salud. Caso práctico: Conteo de huevecillos mosquito Aedes*.

https://www.revistatecnologiadigital.com/pdf/13_02_010_vision_artificial_aplicaci%C3%B3n_sector_salud_caso_practico_conteo_huevecillos_mosquito_aedes_aegypti.pdf

Seed Studio. (2023). *Getting Started | Seed Studio Wiki*.

https://wiki.seedstudio.com/xiao_esp32s3_getting_started/

Tejero Caballo, E. (2020). *Aplicaciones de Machine Learning a la Bioacústica Marina. Msc Thesis*.

https://www.researchgate.net/publication/349536480_Aplicaciones_de_Machine_Learning_a_la_Bioacustica_Marina_Msc_Thesis

Velasquez Correa, R. E. (2022). Diseño e implementación de un prototipo basado en Internet de las

Cosas (IoT) empleando sensores de oximetría de pulso y redes neuronales artificiales para

determinar el nivel de hemoglobina de manera no invasiva en niños y adolescentes en el distrito de

Villa María del Triunfo. *Repositorio Institucional - UTP*.

<http://repositorio.utp.edu.pe/handle/20.500.12867/7117>

Wardana, I. N. K., Fahmy, S. A., & Gardner, J. W. (2023). TinyML Models for a Low-cost Air Quality

Monitoring Device. *IEEE Sensors Letters*, 7(11). <https://doi.org/10.1109/LSENS.2023.3315249>

WHO/OPS. (2019). Documento técnico para la implementación de intervenciones basado en escenarios

operativos genéricos para el control del Aedes. *World Health Organization*, 58. chrome-

extension://efaidnbmnnnibpcajpcglclefindmkaj/[https://iris.paho.org/bitstream/handle/10665.2/5](https://iris.paho.org/bitstream/handle/10665.2/51654/9789275321102_spa.pdf?sequence=1&isAllowed=y)

[1654/9789275321102_spa.pdf?sequence=1&isAllowed=y](https://iris.paho.org/bitstream/handle/10665.2/51654/9789275321102_spa.pdf?sequence=1&isAllowed=y)

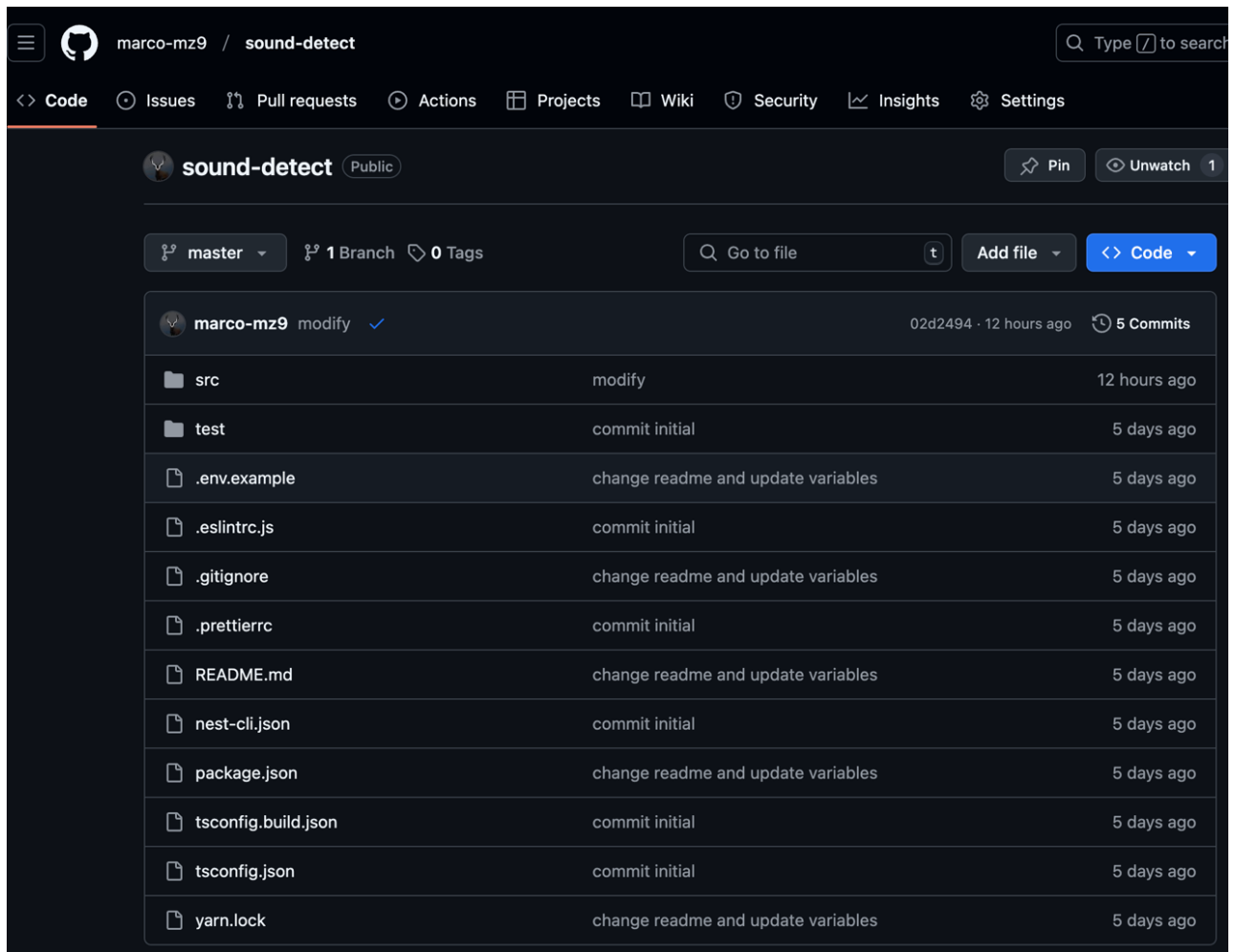
ANEXOS

Repositorio de Código Fuente GitHub

Para el acceso al código clic en [link](#)

Figura 14

Repositorio Backend Nestjs



Carta de Aceptación



IBARRA



Carta de satisfacción

Por medio de la presente, se informa que he recibido y evaluado el trabajo de titulación realizado por el estudiante Marco Antonio Muñoz Villareal titulado **“Prototipo de identificación del mosquito Aedes con TINYML”**. Quiero expresar la satisfacción con el producto entregado, el cual ha cumplido con los requisitos establecidos en el proyecto de investigación.

Así mismo quiero informar que el grupo de investigación se complace con el trabajo realizado por el estudiante para optar al título de Ingeniera en Tecnologías de la Información.

Dra. Dulce Milagro Rivero Albarrán
Líder del Grupo de Investigación GISI

Ibarra, 17 de febrero del 2025