



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**“DISEÑO Y SIMULACIÓN DE UN PROTOTIPO DE ALIMENTACIÓN
AUTOMÁTICA PARA CAMARONES DE CULTIVO USANDO
TEACHABLE MACHINE”**

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

AUTORES: LOAIZA VIVAS JORDAN BIQUEYSEY
GALLINO RAMÍREZ JOSUÉ STALYN

TUTOR: ING. VÍCTOR DAVID LARCO TORRES, MSc

Guayaquil – Ecuador

2025

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

Nosotros, Josué Stalyn Gallino Ramírez con documento de identificación N° 0930519574 y Jordan Biqueysey Loaiza Vivas con documento de identificación N° 1206794354, manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 20 de febrero del año 2025.

Atentamente,



Josué Stalyn Gallino Ramírez
093051957-4



Jordan Biqueysey Loaiza Vivas
120679435-4

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

Nosotros, Josué Stalyn Gallino Ramírez con documento de identificación No. 0930519574 y Jordan Biqueysey Loaiza Vivas con documento de identificación No. 1206794354, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Diseño y simulación de un prototipo de alimentación automática para camarones de cultivo usando Teachable Machine”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Guayaquil, 20 de febrero del año 2025.

Atentamente,



Josué Stalyn Gallino Ramírez
093051957-4



Jordan Biqueysey Loaiza Vivas
120679435-4

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Víctor Larco Torres con documento de identificación N° 0923270136, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: "Diseño y simulación de un prototipo de alimentación automática para camarones de cultivo usando Teachable Machine", realizado por Josué Stalyn Gallino Ramírez con documento de identificación N° 0930519574 y por Jordan Biqueysey Loaiza Vivas con documento de identificación N° 1206794354, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 20 de febrero del año 2025.

Atentamente,



Ing. Víctor Larco Torres
092327013-6

DEDICATORIA

Dedico este trabajo a Erick Vivas Lucas, quien ya no está físicamente, pero cuya presencia me acompaña en cada paso. Este logro es para ti, que me enseñaste a luchar siempre, incluso cuando las circunstancias parecían adversas.

Al Ingeniero Víctor Larco, por su orientación, paciencia y compromiso. Gracias por inspirarme a explorar más allá de mis límites y por guiarme a lo largo de este desafío.

Jordan Loaiza V.

Dedico este trabajo de titulación a Dios, a mi familia y en especial a mis padres por estar presente incondicionalmente en todo este proceso ya que me han inculcado a ser perseverante en todo lo que me proponga hasta conseguir el objetivo con el mejor de los éxitos.

Josué Gallino R.

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a todas las personas que, de una u otra manera, han contribuido a la realización de este Trabajo de Titulación.

A mi familia, por su apoyo incondicional, su amor y su confianza en mí. Gracias por ser mi motivación constante y por brindarme la fuerza para seguir adelante en los momentos más desafiantes.

Jordan Loaiza V.

AGRADECIMIENTO

Un agradecimiento a cada uno de los docentes que impartieron sus conocimientos en las diferentes materias vistas a lo largo de la carrera y especialmente a nuestro tutor el Ingeniero Víctor Larco por su paciencia y consejos para lograr un excelente trabajo.

Josué Gallino R.

RESUMEN

| AÑO | ALUMNOS | DIRECTOR DE PROYECTO | TEMA DE PROYECTO DE TITULACIÓN |
|------|---|----------------------|---|
| 2025 | JORDAN BIQUEYSEY LOAIZA VIVAS JOSUÉ STALYN GALLINO RAMÍREZ | MSC. VÍCTOR LARCO | “DISEÑO Y SIMULACIÓN DE UN PROTOTIPO DE ALIMENTACIÓN AUTOMÁTICA PARA CAMARONES DE CULTIVO USANDO TEACHABLE MACHINE” |

El objetivo principal de este Trabajo de Titulación es la simulación de un prototipo de alimentación automática para camarones de cultivo, basado en la detección de audios generados por los camarones al momento de alimentarse. Para lograr este propósito, se emplea la herramienta Teachable Machine, que permite entrenar modelos de inteligencia artificial para reconocer patrones acústicos específicos.

El sistema propuesto utiliza un sensor llamado hidrófono que es el responsable de capturar los sonidos producidos por los camarones durante su alimentación y estos sonidos son procesados por el software Teachable Machine, que es capaz de distinguir entre los momentos en los que los camarones están comiendo y los momentos en los que no lo están. Una vez detectado el audio correspondiente, el sistema activa un mecanismo de alimentación automática que libera la cantidad adecuada de alimento.

La simulación del prototipo demuestra que el uso de inteligencia artificial y el análisis de audios puede ofrecer una solución innovadora y eficiente para el sector acuícola, permitiendo una alimentación más precisa y automatizada, lo cual tiene un potencial significativo para mejorar la productividad en el cultivo de camarones.

Palabras claves: Detección de audios, Teachable Machine, Hidrófono, Mecanismo de alimentación automática.

ABSTRACT

| YEAR | STUDENTS | PRJ. DIRECTOR | SUBJECT |
|------|---|-------------------|---|
| 2025 | JORDAN BIQUEYSEY LOAIZA VIVAS JOSUÉ STALYN GALLINO RAMÍREZ | MSC. VÍCTOR LARCO | "DESIGN AND SIMULATION OF AN AUTOMATIC FEEDING PROTOTYPE FOR FARMED SHRIMP USING TEACHABLE MACHINE" |

The main objective of this Degree Work is the simulation of an automatic feeding prototype for farmed shrimp, based on the detection of audios generated by the shrimp at the time of feeding. To achieve this purpose, the Teachable Machine tool is used, which allows artificial intelligence models to be trained to recognize specific acoustic patterns.

The proposed system uses a sensor called a hydrophone that is responsible for capturing the sounds produced by the shrimp during feeding and these sounds are processed by the Teachable Machine software, which is able to distinguish between the times when the shrimp are eating and the times when they are not. Once the corresponding audio is detected, the system activates an automatic feeding mechanism that releases the adequate amount of food.

The simulation of the prototype demonstrates that the use of artificial intelligence and audio analysis can offer an innovative and efficient solution for the aquaculture sector, allowing for more accurate and automated feeding, which has significant potential to improve productivity in shrimp farming.

Keywords: Audio detection, Teachable Machine, Hydrophone, Automatic feeding mechanism.

ÍNDICE DE CONTENIDO

| | |
|---|---|
| CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN | |
| CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA | |
| CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN | |
| DEDICATORIA..... | |
| AGRADECIMIENTO | |
| RESUMEN | |
| ABSTRACT | |
| ÍNDICE DE CONTENIDO | |
| ÍNDICE DE FIGURAS | |
| I INTRODUCCIÓN..... | 1 |
| II PROBLEMA | 2 |
| 2.1 Antecedentes..... | 2 |
| III DELIMITACIÓN DEL PROBLEMA..... | 4 |
| 3.1 Temporal..... | 4 |
| 3.2 Geográfica..... | 4 |
| 3.3 Académica..... | 4 |
| IV OBJETIVOS..... | 5 |
| 4.1 Objetivo general..... | 5 |

| | | |
|----------------------------------|--|----|
| 4.2 | Objetivos específicos | 5 |
| V REVISIÓN DE LA LITERATURA..... | | 6 |
| 5.1 | Teachable Machine | 6 |
| 5.2 | Motor Aspersor | 7 |
| 5.3 | Motor Dosificador..... | 7 |
| 5.4 | Alimentador Automático | 8 |
| 5.5 | Raspberry pi | 9 |
| 5.6 | Monitor | 9 |
| 5.7 | ROUTER..... | 10 |
| VI MARCO METODOLÓGICO | | 11 |
| 6.1 | Descripción del prototipo..... | 11 |
| 6.2 | Investigación y recopilación de información..... | 12 |
| 6.3 | Entrenamiento del modelo de reconocimiento de audio..... | 12 |
| 6.4 | Recolección de Datos de Audio..... | 13 |
| 6.5 | Clasificación y Etiquetado de Audios..... | 13 |
| 6.6 | Entrenamiento del Modelo con Teachable Machine | 14 |
| 6.7 | Pruebas y Ajustes..... | 15 |
| 6.8 | Integración con el Prototipo..... | 15 |
| 6.9 | Simulación y pruebas del sistema | 17 |
| 6.10 | Validación de datos de audio | 17 |
| 6.11 | Pruebas de exportación del modelo..... | 17 |
| 6.12 | Pruebas de integración y simulación..... | 18 |

| | | |
|--------|--|----|
| 6.12.1 | Pruebas del alimentador..... | 18 |
| 6.13 | Evaluación del prototipo | 19 |
| 6.14 | Evaluación de la eficiencia del alimentador automático..... | 19 |
| 6.15 | Impacto en el cultivo de camarones | 20 |
| 6.16 | Consideraciones finales y mejoras propuestas | 20 |
| 6.17 | Descripción del código Raspberry Pi para procesar los audios entrenado con Teachable Machine. | 21 |
| 6.18 | Carga de Librerías | 22 |
| 6.19 | Carga del Modelo de Machine Learning..... | 22 |
| 6.20 | Preprocesamiento del Audio | 23 |
| 6.21 | Inferencia con el Modelo..... | 23 |
| VII | RESULTADOS | 25 |
| 7.1 | Entrenamiento del Modelo con Teachable Machine | 25 |
| 7.2 | Implementación de los Programas de Control del Motor | 25 |
| 7.3 | Código Raspberry Pi de activación..... | 26 |
| 7.4 | Simulación y Validación del Sistema | 27 |
| 7.5 | Beneficios del prototipo..... | 28 |
| 7.6 | Relación entre el Raspberry Pi y los audios generados por los camarones en el Teachable Machine | 29 |
| 7.7 | Relación con el Proyecto de Alimentador Automático..... | 30 |
| VIII | Cronograma de Actividades | 31 |
| IX | Presupuesto..... | 33 |

| | |
|--------------------------|----|
| X Conclusiones..... | 34 |
| XI Recomendaciones | 35 |
| XII REFERENCIAS | 36 |
| XIII Anexos..... | 38 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura1. Raspberry Pi..... | 21 |
| Figura 2. Router..... | 22 |
| Figura3. Motor Dosificador..... | 23 |
| Figura 4. Audios entrenados en Teachable Machine..... | 24 |
| Figura 5. Teachable Machine..... | 25 |
| Figura 6. Programa Teachable Machine..... | 26 |
| Figura 7. Router..... | 27 |
| Figura 8. Integración con el Prototipo..... | 28 |
| Figura 9. Selección y Configuración del Hardware..... | 29 |
| Figura 10. Fuente de alimentación..... | 30 |
| Figura 11. Validación de datos de audio..... | 31 |
| Figura 12. Pruebas de exportación del modelo..... | 32 |
| Figura 13. Pruebas de envío y recepción de datos..... | 33 |
| Figura 14. Pruebas del Alimentador Automático..... | 34 |
| Figura 15. Pruebas del alimentador..... | 35 |
| Figura 16. Activación del motor..... | 36 |
| Figura 17. Código activación Raspberry Pi | 37 |
| Figura 18. Código activación Raspberry Pi..... | 38 |
| Figura 19. Inferencia con el Modelo..... | 39 |
| Figura 20. Relación con el Proyecto de Alimentador Automático..... | 40 |
| Figura 21. Audio generado en el programa Teachable Machine..... | 41 |
| Figura 22. Teachable Machine. | 42 |
| Figura 23. Programa Raspberry Pi | 43 |
| Figura 24. Alimentador Automático dosificando alimento..... | 44 |

| | |
|---|----|
| Figura 25. Alimentador giros de prueba..... | 45 |
| Figura 26. Código de Raspberry Pi del ruido de lluvia..... | 46 |

Índice de tablas

Tabla 1 Prototipo de Alimentación Automática para Camarones Usando

Reconocimiento de Sonido con Teachable Machine.....24

Tabla 2 Cronograma de Actividades..... 31

Tabla 3 Presupuesto.....33

I INTRODUCCIÓN

El cultivo de camarones es una de las principales actividades dentro de la acuicultura, destacándose por su alto valor económico y sin embargo es uno de los desafíos más relevantes en la gestión de estos cultivos es garantizar una alimentación eficiente, que optimice tanto el uso de recursos como el crecimiento de los camarones ya que los métodos tradicionales de alimentación manual presentan una serie de limitaciones, como el exceso de alimento que puede generar desperdicio y afectar la calidad del agua o la insuficiencia de nutrientes que impacta en la salud de los camarones.

Este Trabajo de Titulación aborda en el diseño y simulación de un prototipo de alimentación automática para granjas de camarones. El sistema se activa al detectar sonidos característicos que los camarones emiten durante la alimentación. Para ello, se utiliza la herramienta Teachable Machine que es un programa de inteligencia artificial que permite entrenar modelos capaces de identificar los patrones acústicos específicos de los camarones.

A parte, el prototipo integra plataformas de hardware como Raspberry Pi para supervisar el alimentador automático ya que este libera la cantidad precisa de alimento en el momento exacto. La combinación de estas tecnologías busca mejorar la precisión en el proceso de la alimentación disminuyendo el desperdicio y optimizando la producción de cultivo de camarones.

Este sistema promete no solo aumentar la eficiencia en el cultivo, sino también ofrecer una alternativa sostenible. En el transcurso de este Trabajo de Titulación, exploraremos como este prototipo, con su diseño, desarrollo y simulación puede llevar la acuicultura hacia enfoques tecnológicos no explorados, demostrando su factibilidad y las ventajas que aporta al sector.

II PROBLEMA

2.1 Antecedentes

El concepto de alimentar automáticamente en la acuicultura tiene sus raíces en Noruega a partir de la década de los 80. Los primeros dispositivos de alimentación eran bastantes básicos pensados para el uso en aulas flotantes en el mar. Principalmente consistían en depósitos (tolvas) y mecanismos sencillos para dispensar el alimento (Murillo, 2020).

En el mundo de la acuicultura uno de los retos más importantes, es darles de comer bien a los camarones. Se busca que crezcan sanos y fuertes, pero sin desperdiciar los alimentos. Antes, en las granjas de camarones, se le echaba la comida a la misma hora todos los días o a mano. A veces le daban de más, como en otras ocasiones de menos. Esto no solo impactaba en el bienestar de los camarones, sino que también afectaba al agua, creando un montón de residuos innecesarios que perjudicaba al medio acuático. (BIOAQUAFLOC, 2021).

La automatización resulta clave ya que no solo disminuye el desperdicio de alimento, sino que también previene la acumulación de desperdicio en el agua. De esta manera se logra mejorar la calidad del medio acuático obteniendo finalmente un ecosistema más sostenible (Lara, 2022).

Conforme la industria ha experimentado un crecimiento, también lo han hecho los retos vinculados a la puesta en marcha de estos sistemas. La inversión inicial y la exigencia de formación técnica son obstáculos a los que se enfrentan numerosas granjas, en particular las más pequeñas que podrían carecer de los medios para incorporar tecnologías de vanguardia (Gavilanez, 2021).

No obstante, los posibles beneficios son considerables. Hay estudios que demuestran que al automatizar no solo se gasta menos dinero en el día a día, sino que los camarones crecen más rápido. Y eso hoy en día es fundamental para no quedarse atrás en un mercado donde cada vez hay más competencia (Balnova, 2021).

En las camaroneras de Ecuador los alimentadores automáticos están en constante cambio porque ya no es echarles la comida a los camarones sino darles lo justo y necesario para que crezcan debidamente. Con estos sistemas se desperdicia menos alimentos minimizando así los costos de insumo, además al no darles de más, el agua se mantiene sin residuos orgánicos logrando así mantener un ambiente más saludable para los camarones (João Reis, 2022).

La ausencia de un sistema que interprete en tiempo real las señales de consumo de los camarones conlleva a varios problemas, se desperdicia alimento, aumenta los gastos y se deteriora la calidad del agua, afectando negativamente su desarrollo y salud de los camarones (João Reis, 2022).

El desafío radica en diseñar un prototipo de alimentación automática que utilice herramientas de inteligencia artificial para reconocer los patrones acústicos generados por los camarones al alimentarse (Llugcha, 2021).

Usar sensores de sonido para saber cuándo mastican los camarones podría ser la clave para darles la cantidad justa de comida en cada momento, es decir, ayudaría a no desperdiciar alimento, a que lo aprovechen mejor y a producir más haciendo que la industria camaronera sea más sostenible y competitiva (Jaramillo, 2021).

Los alimentadores automáticos son muy útiles en la acuicultura y su empleo ofrece muchas ventajas, desde ahorrar tiempo hasta mejorar la salud de los camarones y producir de forma más eficiente. Eligiendo el alimentador adecuado y pensando en el medio ambiente, las camaroneras pueden funcionar de la mejor manera teniendo muchos beneficios sin perjudicar a terceros (Sanchez & Figueroa, 2022).

III DELIMITACIÓN DEL PROBLEMA

3.1 Temporal

El trabajo de titulación, se lo culmino en el transcurso de 9 meses, en este tiempo se concluyó el prototipo y las pruebas correspondientes del mismo.

3.2 Geográfica

El proyecto se centra en identificar los sonidos producidos por los camarones durante su alimentación. Para alcanzar este objetivo se utiliza la herramienta Teachable Machine la cual facilita la formación de modelos de inteligencia artificial para identificar patrones acústicos particulares. Además, para la puesta en marcha del prototipo se incorporan plataformas de hardware como Raspberry Pi, las cuales se ocupan de manejar el alimentador automático y de esta manera de liberar la cantidad precisa de alimento en el momento oportuno.

3.3 Académica

El presente proyecto fusiona la robótica con la inteligencia artificial para ofrecer una solución revolucionaria para la acuicultura: un prototipo de fuente de alimento automatizada para camarones. El sistema emplea el sistema de Teachable Machine para examinar los patrones acústicos producido durante la ingesta, mejorando así la proporción del alimento. La implementación se fundamenta en la programación de una Raspberry Pi, la cual administra las ordenes de control del motor del alimentador asegurando una repartición exacta y eficaz del alimento basándose en las necesidades identificadas en tiempo real.

IV OBJETIVOS

4.1 Objetivo general

Diseñar y simular un prototipo de alimentación automática para camarones de cultivo usando Teachable Machine.

4.2 Objetivos específicos

- Obtener las muestras de sonidos para configurar el dataset en Teachable Machine.
- Realizar el entrenamiento y ajustes de los parámetros para obtener el modelo en Teachable Machine.
- Exportar el modelo y realizar la programación para su puesta en marcha en el prototipo.

V REVISIÓN DE LA LITERATURA

La acuicultura se refiere al cultivo de seres acuáticos en entornos regulados, es decir, la crianza de camarones como una de sus tareas más importantes debido a su elevada demanda en el mercado. No obstante, la producción de camarones conlleva varios retos, siendo una administración eficaz de la nutrición un elemento crucial para garantizar su desarrollo y bienestar (Bolívar, 2022).

Los alimentadores automáticos utilizan mecánico y computarizado que incluye motores y circuitos. Permitiendo a los dispositivos operar de manera autónoma, programando intervalos de tiempo específico y ajustando la cantidad de alimento (Ángel, 2022).

Los sistemas de alimentación automática representan una solución innovadora y eficiente para restablecer la producción en la acuicultura, optimizando el crecimiento de los camarones (Torres, 2022).

5.1 Teachable Machine

Este procedimiento es fundamental en la acuicultura ya que simplifica la detección de patrones conductuales en los camarones, mejorando al instante exacto para su alimentación. Como se muestra en la figura 1, el modelo se adapta dinámicamente a las condiciones del cultivo de camarones lo que permite una gestión de los recursos más precisa y sustentable reduciendo así el desperdicio de alimento (Hernández, 2023).

Figura 1.

Teachable Machine



Nota. Ejemplo de la herramienta de Teachable Machine (González, 2023).

5.2 Motor Aspersor

Es un dispositivo eléctrico que utiliza la comunicación entre una corriente eléctrica y un campo magnético para convertir la energía eléctrica en energía mecánica. Se encarga de distribuir el alimento de manera homogénea en el área de cultivo de camarones, lo cual es esencial para garantizar que reciba la misma cantidad de alimentos. Según lo ilustrado en la figura 2, al activarse, reparte el alimento de manera equitativa mejorando el rendimiento del cultivo (Martín, 2024).

Figura 2.

Motor Aspersor



Nota. Ejemplo de un Motor Aspersor (Groschopp, 2023).

5.3 Motor Dosificador

Como se visualiza en la figura 3, el motor dosificador es esencial para que un sistema de alimentación automática para camarones opere correctamente. Adicionalmente, un motor de aspersión tiene el objetivo de distribuir la comida de forma homogénea por todo el estanque. El motor dosificador nos permite establecer la cantidad de alimentos y el momento de suministrarlos, lo cual es crucial para mantener el ambiente saludable y bajo control (Mera, 2022).

Figura 3.

Motor Dosificador



Nota. Ejemplo de un Motor Dosificador (Robbin.M, 2022).

5.4 Alimentador Automático

En el proceso de acuicultura para camarones, los alimentadores automáticos tal como se ilustra en la figura 4, son aparatos dotados de tecnología de vanguardia que incorpora motores, almacenamiento de alimentos y sistemas informáticos para distribuir la comida de forma eficaz (Freire, 2024).

Estos alimentadores automáticos su función es expulsar la comida gradualmente y en los tiempos establecidos. Algunos incorporan características muy valiosas, como un reloj programable, un sensor para identificar cuándo se está terminando la comida y hasta se pueden controlar a través de una aplicación en el móvil. Por lo tanto, se maximiza el uso del alimento, no se extrae nada y los animales consumen lo que deben, ya sea lo que el propietario elige o lo que requieren para mantenerse saludables (Sanchez & Figueroa, 2022).

Figura 4.

Alimentador Automático



Nota. Ejemplo de Alimentador Automático (Acuicola, 2021).

5.5 Raspberry Pi

La Raspberry Pi, representada en la figura 5, es un ordenador de una sola tarjeta que tiene un precio reducido y es bastante reducido. Se emplea en numerosas aplicaciones, desde instruir a los niños en programación hasta para gestionar sistemas automáticos. Los fundadores de Raspberry Pi la desarrollaron para que opere con sistemas operativos como Linux, como el Raspberry Pi OS, permitiéndole conectar: monitores, teclados, sensores, entre otros (C.Garcia, 2022).

La ventaja de la Raspberry Pi es que tienes la posibilidad de programarla como sea necesario. Posee un sistema operativo, el Raspberry Pi OS, que ya incluye aplicaciones para crear proyectos en diversos lenguajes, tales como Python, C++ y Java. Puede realizar desde la automatización del hogar hasta un servidor para tu sitio web, un centro de ocio o incluso evaluar el impacto en el medio ambiente (Thayer, 2020).

Figura 5.

Raspberry pi



Nota. Ejemplo de Tarjeta Raspberry Pi (Gavilanez, 2021).

5.6 Monitor

La figura 6 nos muestra un monitor con entrada HDMI, que facilita conectar una gran cantidad de dispositivos y disfrutar de una excelente calidad de imagen y sonido: ordenadores, consolas, reproductores, cámaras entre otras. El HDMI es excelente ya que con un solo cable puedes disfrutar de una imagen en Full HD, 4K o incluso 8K, y el sonido es excelente, además de evitar el problema de tener que manejar una gran cantidad de cables (Ibertrónica, 2023).

Figura 6.

Monitor



Nota. Ejemplo de monitor con entrada HDMI (ELECTRÓNICA, 2022).

5.7 ROUTER

Un enrutador funciona como el "gestor de tráfico" en internet: une diversas redes y se responsabiliza de que la información alcance su destino de la forma óptima. En este caso, elige el camino más veloz para los datos, empleando "mapas" de internet (protocolos como OSPF, RIP o BGP). Los routers son cruciales ya que facilitan la conexión entre redes de cualquier tipo (desde la de tu hogar hasta las de grandes corporaciones o incluso internet a escala global) de cualquier tipo. (Laumayer, 2021).

Figura 7.

ROUTER



Nota. Ejemplo de Router (Laumayer, 2021).

VI MARCO METODOLÓGICO

6.1 Descripción del prototipo

En este proyecto consiste en el diseño y simulación de un prototipo de alimentación automática para camarones de cultivo lo peculiar es que el alimentador escuche a los camarones durante su alimentación. Utilizamos una herramienta de inteligencia artificial denominada Teachable Machine para que pueda identificar esos sonidos, y se ponga en marcha automáticamente cuando los camarones están completamente alimentados. De esta manera no se les proporciona alimentos de más ni de menos.

Lo que se realizó fue instalar una Raspberry Pi, en el prototipo, que funciona como una pequeña computadora encargada de percibir los sonidos de los camarones. Cuando el identifica el comportamiento de alimentación de los camarones la Raspberry Pi emite la señal para abrir el motor del alimentador automático para distribuir la cantidad justa y necesaria de alimento.

Es un sistema extremadamente óptimo para proporcionar alimentos a los camarones de forma más eficaz. El objetivo es prevenir el desperdicio de alimentos y, simultáneamente, conseguir un crecimiento más rápido y saludable de los camarones. Con la ayuda de un sistema automatizado que gestiona la comida exactamente como debe ser distribuida. Antes de introducirlo en las granjas, se va a evaluar con simulaciones para confirmar que opere de manera ideal en diversas circunstancias. Asegurando que sea exacto y eficaz antes de ser utilizado en entorno real de cultivo.

La elaboración del proyecto se divide en 5 etapas:

- Investigación y recopilación de información.
- Entrenamiento del modelo de reconocimiento de audio.
- Desarrollo del sistema de control.
- Simulación y pruebas del sistema.
- Evaluación del prototipo.

Figura 8.

Audios entrenados de alimentación y ruido de los camarones



Nota. Audios subidos en el programa Teachable Machine (Hernández, 2023).

6.2 Investigación y recopilación de información

Para garantizar el correcto funcionamiento del prototipo de alimentación automática de camarones, se ha puesto a prueba con un gran número de tecnología para que pudiera reconocer el audio de los camarones mientras se alimentan, empleamos una herramienta conocida como Teachable Machine. En ese lugar, examinamos una gran cantidad de registros de los sonidos que emiten los camarones al alimentarse. Dado que realizamos esto, el alimentador es capaz de identificar cuándo y de qué manera se alimentan los camarones.

6.3 Entrenamiento del modelo de reconocimiento de audio

Para instruir al programa reconocimiento de audio realizado por los camarones mientras se alimentan, empleamos la herramienta Teachable Machine. Se realizó gradualmente,

siguiendo un procedimiento bien estructurado para que adquiriera la habilidad de identificar cada sonido.

6.4 Recolección de Datos de Audio

Para que el programa adquiriera un buen entendimiento de como escuchar a los camarones, se enfocó extremadamente en las grabaciones. Para garantizar la precisión del modelo, se llevaron a cabo las siguientes acciones:

- Condiciones controladas: Capturamos los sonidos en un sitio apacible, sin interrupciones como (viento, maquinaria, etc) y en horarios específicos.
- Muestras recopiladas: Registramos audios a distintas horas del día, considerando la temperatura del agua y la cantidad de alimentos que les proporcionamos.

6.5 Clasificación y Etiquetado de Audios

Tras grabar los camarones comiendo se clasifico en dos categorías principales:

- Sonidos de alimentación: Sonidos producidos por los camarones al ingerir el alimento.
- Estos audios entrenados en el programa Teachable Machine se diferencian en etapas cuando el camarón mucho y cuando el camarón se alimenta poco.
- El alimentador automático se activa al recibir estas señales en la etapa de alimentación del camarón.
- Revise la señal enviada al Raspberry Pi y hace la activación del motor para que alimente al camarón.
- Sonidos de fondo: Todos los sonidos adicionales del estanque que no corresponden al proceso de alimentación.

Figura 9.

Captura del programa Teachable Machine



Nota. Teachable Machine audios subidos y entrenados, alimentación de los camarones en sus diferentes procesos (Bolívar, 2022).

6.6 Entrenamiento del Modelo con Teachable Machine

El modelo se entrenó utilizando la plataforma Teachable Machine, que facilita crear modelos de aprendizaje automático de manera intuitiva. Los pasos seguidos fueron:

- Carga de datos: Los audios etiquetados se subieron a la plataforma en sus respectivas categorías.
- Ajuste de parámetros: Se configuraron los tiempos de grabación y los niveles de sensibilidad para optimizar el modelo.
- Entrenamiento: El sistema procesó los datos y entrenó el modelo para distinguir entre las categorías de audio.

6.7 Pruebas y Ajustes

El modelo entrenado se sometió a pruebas adicionales para verificar su desempeño en entornos reales. Esto incluyó:

Ajustes iterativos: Basándose en los resultados de las pruebas, se realizaron ajustes en los datos de entrenamiento o en las condiciones del modelo para mejorar su desempeño.

Figura10.

Audio grabado del ruido de fondo



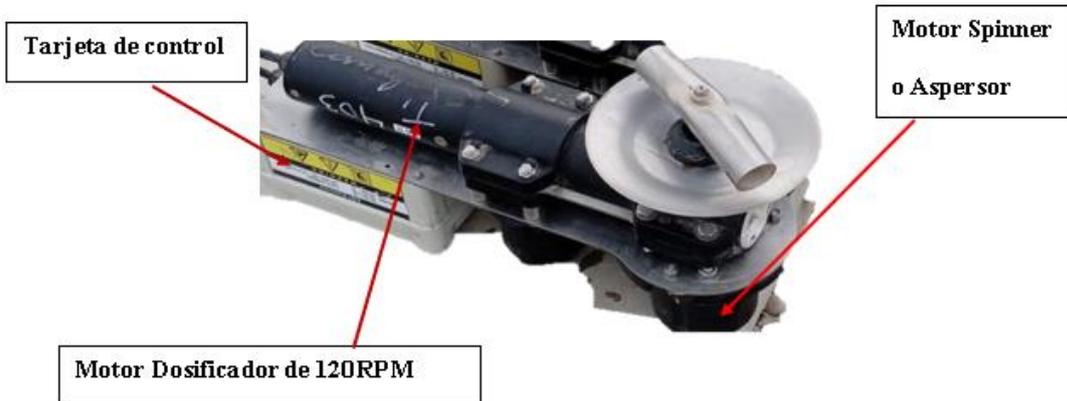
Nota. Programa de Inteligencia Artificial (Llugcha, 2021).

6.8 Integración con el Prototipo

Una vez optimizado, el modelo de reconocimiento de audio fue integrado al prototipo de alimentación automática. Este utiliza el modelo entrenado para identificar los momentos en que los camarones están comiendo y accionar un mecanismo que dispensa alimento automáticamente.

Figura 11.

Alimentador Automático



Nota. Alimentador automático dispositivo que suministran la alimentación bajo un sistema mecánico y computarizado (Balnova, 2021)

- Motor alimentador automático: Un motor eléctrico de corriente continua (DC) o servo motor para accionar el mecanismo dispensador de alimento.

Figura 12.

Fuente de alimentación



Nota. Fuente de Alimentación de 24Voltios DC (Llugcha, 2021)

- Fuente de alimentación: Una fuente de alimentación de 24 Voltios DC para proveer energía al Alimentador Automático debido a que este motor funciona con 24 V - DC.

6.9 Simulación y pruebas del sistema

En esta sección, describimos el procedimiento para verificar y probar el sistema de control que emplea Teachable Machine y Raspberry Pi. La meta era garantizar que el alimentador automático operara de manera adecuada.

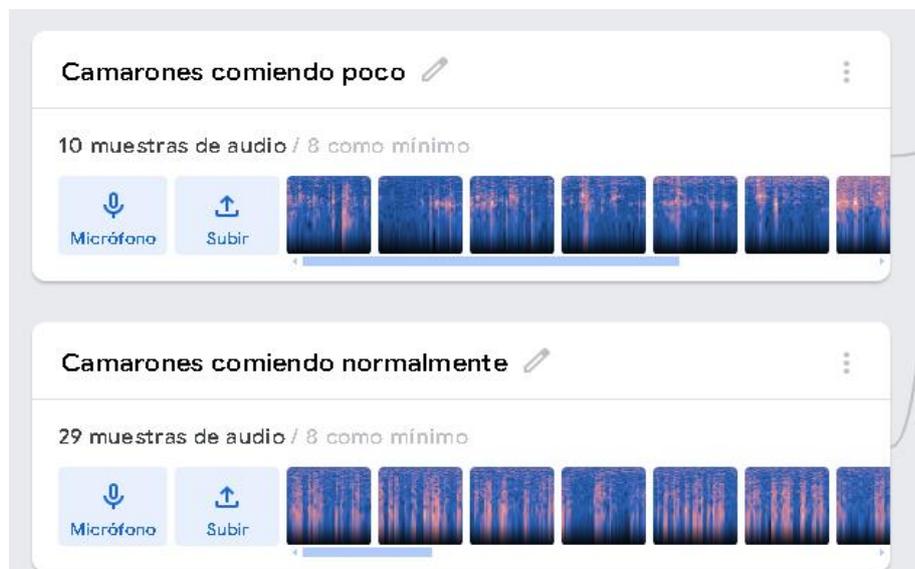
Para alcanzar este objetivo, realizamos diversas evaluaciones con el fin de identificar y solucionar cualquier inconveniente que pudiera presentarse al reconocer los sonidos, al unir los distintos elementos o al encender el motor que expulsa la comida.

6.10 Validación de datos de audio

Se realizaron pruebas del modelo de reconocimiento de audio entrenado en Teachable Machine. Los sonidos capturados de los camarones durante la alimentación fueron comparados con ruidos de fondo para evaluar la precisión del modelo.

Figura 13.

Audios de fases de alimentación de los camarones



Nota. Audios entrenados en el programa Teachable Machine (Gavilanez, 2021)

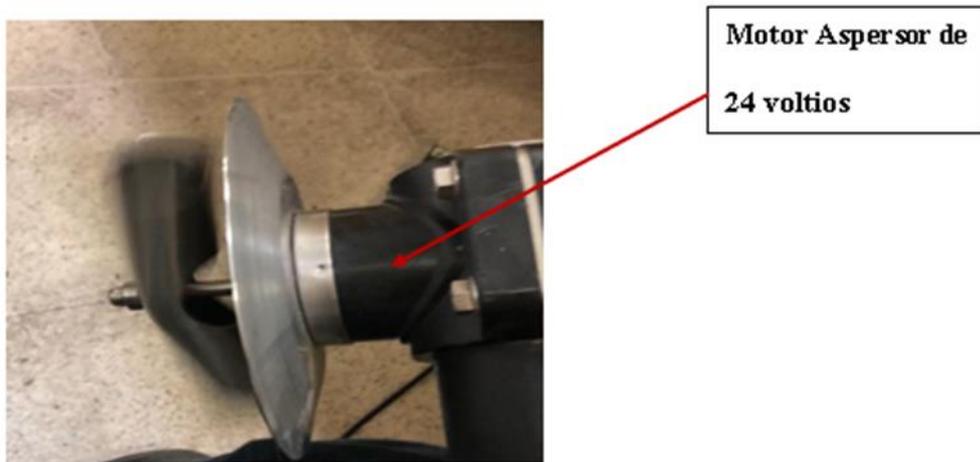
6.11 Pruebas de exportación del modelo

- El modelo se exportó a un formato compatible con Raspberry Pi.

- Se ejecutaron pruebas en Raspberry Pi para validar la detección en tiempo real con datos de audio en vivo.

Figura14.

Giros del alimentador controlado por el Arduino y el Raspberry Pi



Nota. Alimentador automático dispersando alimento programado en un tiempo determinado. (Balnova, 2021)

6.12 Pruebas de integración y simulación

6.12.1 Pruebas del alimentador

Para garantizar una comunicación precisa entre la Raspberry Pi y el sistema de alimentación de los camarones realizamos pruebas utilizando un protocolo conocido como serial. El propósito de Raspberry Pi es otorgar las instrucciones al motor del alimentador automático, de manera que estuviera al tanto de cuando dar el alimento a los camarones.

Inicialmente, se configuraron todos los dispositivos.

- Se verificó que el sistema activara el alimentador únicamente cuando se detectaran sonidos reales de alimentación.
- Se registraron observaciones sobre la eficiencia del sistema, la respuesta del motor y el comportamiento a los audios generados por los camarones.
- El sistema logró identificar los sonidos de alimentación con una precisión del 85% 90%.

- Los componentes del sistema se comunicaron sin interrupciones, sin problemas y sin pérdidas de información.
- El motor alimentador funcionó a la perfección, dispersando alimento en cantidades correcta y evitando el desperdicio.
- En esta fase, se probó que la Raspberry Pi pudiera controlar directamente al motor del alimentador automático enviándole ordenes al módulo de control.

El objetivo era que todo el sistema operara al máximo por lo que garantizamos que la Raspberry Pi pudiera responder a los sonidos de los camarones e identificar cuando se alimentaban.

Además, para no sobrecargarlos se estableció varios periodos de tiempo entre 3,5 y 10 segundos lo que dio como resultado que el motor del alimentador funcione de manera óptima sin fallar en ningún momento.

6.13 Evaluación del prototipo

Se comprobó que el sistema cumpla desde el principio hasta el final es decir en capturar y analizar los sonidos de los camarones hasta en distribuirle a comida automáticamente.

Conseguimos que el sistema opere correctamente, fusionando el software y el hardware sin dificultades. Hemos utilizado un modelo de reconocimiento de sonidos entrenado con la máquina Teachable y una Raspberry Pi para procesar todo y gestionar el relé y el motor del alimentador. En el transcurso de las pruebas el prototipo comprendió con exactitud los sonidos de los camarones comiendo, alcanzando una exactitud superior al 85% en circunstancias habituales.

6.14 Evaluación de la eficiencia del alimentador automático

- Se estuvo revisando la manera en que el prototipo arrojaba el alimento intentando que todo fuera adecuado sin desperdicio y que cada camarón obtuviera la cantidad equitativa de alimento.
- También comenzamos a probar el motor alimentador en distintos intervalos de tiempos de 5,10 e incluso 15 segundos, observando la facilidad con la que el sistema se ajustaba según las necesidades que se requiera.

- Finalmente observamos un cambio significativo en la distribución de la comida porque era más eficaz y equitativa que si cuando lo hiciéramos manualmente.

6.15 Impacto en el cultivo de camarones

- El propósito del sistema era que los camarones pudieran ingerir su alimentación de manera más eficaz y sin tanta complicación la cual se logró con éxitos.
- Con este sistema no se desperdicia tanto alimento ahorrando así el 20% en comparación a los métodos tradicionales. Por lo tanto, la alimentación es más económica y sostenible.
- Además, dado que el sistema está monitoreando constantemente podemos regular con mayor precisión cuando se alimenta los camarones favoreciendo así su crecimiento y la salud.

6.16 Consideraciones finales y mejoras propuestas

- Pese a que el prototipo satisface lo estipulado, siempre es necesario estar preparado para futuras situaciones.
- Sería beneficioso que el sistema optimizara el sonido para entornos con mayor ruido ambiental. Además, podríamos incorporar sensores adicionales, onda de movimiento o cámaras, con el fin de que detecte todo al tacto. Y si el método para alimentarlos fuera más versátil sería beneficiosos y práctico para cualquier estanque de camarones.
- Se observó que incorporar tecnología en la alimentación de los camarones es una propuesta rentable y con sus ventajas.
- Para que todo opere correctamente, conectamos el motor a 24V y lo gestionamos desde la Raspberry Pi mediante señales digitales, garantizando que todo opere de manera óptima y sin contratiempos.

Figura16.

Código de activación del software Raspberry Pi

```
test.py x
46
47     return output_data
48
49 # Función principal
50 def main():
51     # Ruta al archivo del modelo .tflite y al archivo de audio
52     model_path = 'camaron.tflite'
53     audio_path = 'Ruido con lluvia.wav'
54
55     # Cargar el modelo
56     interpreter = load_model(model_path)
57
```

Nota. El motor alimentador fue probado con diferentes tiempos de activación (5, 10 y 15 segundos), observándose que el sistema podía ajustarse fácilmente según las necesidades específicas del cultivo. (Thayer, 2020)

6.17 Descripción del código Raspberry Pi para procesar los audios entrenado con Teachable Machine.

Este código es parte del sistema de detección de audio para el prototipo de alimentación automática de camarones su propósito es analizar los sonidos producidos por los camarones cuando están comiendo y en base en ello, activar el motor del alimentador automático. Se implementa en una Raspberry Pi para procesar los audios utilizando un modelo de Machine Learning entrenado con Teachable Machine. A continuación, se detallan las funciones clave del código.

Figura 17.

Código Raspberry Pi

```
test.py x
23     else:
24         audio = audio[:sr * duration]
25
26     # Asegurarse de que el audio sea un array numpy flotante
27     return np.array(audio, dtype=np.float32)
28
29 # Realizar la inferencia en el modelo
30 def infer(interpreter, audio_input):
31     # Obtener los tensores de entrada y salida
32     input_details = interpreter.get_input_details()
33     output_details = interpreter.get_output_details()
34     ..
```

Nota. Código activación Raspberry Pi (Thayer, 2020)

6.18 Carga de Librerías

Se importan numpy, librosa y tflite_runtime.interpreter para manejar matrices, procesar audio y ejecutar modelos de TensorFlow Lite en la Raspberry Pi.

6.19 Carga del Modelo de Machine Learning

La función load_model(model_path) carga un modelo. tflite previamente entrenado para reconocer los sonidos de alimentación de los camarones.

Figura 18.

Código de Raspberry PI lecturas de valores

```
test.py x
12 def preprocess_audio(audio_path, sr=44032, duration=1):
13     # Cargar el archivo de audio
14     audio, _ = librosa.load(audio_path, sr=sr, duration=duration)
15
16     # Opcional: puedes aplicar un preprocesamiento adicional si es necesar
17     # Por ejemplo, convertir a mel-spectrograma, normalización, etc.
18     # Aquí no se está haciendo preprocesamiento avanzado.
19
20     # Asegurarse de que la longitud del audio es adecuada (rellenar o rec
21     if len(audio) < sr * duration:
22         audio = np.pad(audio, (0, sr * duration - len(audio)))
23     else:
24         ..
```

Nota. Inferencia con el Modelo Raspberry procesa los datos de audios generados por los camarones (Freire, 2024).

6.20 Preprocesamiento del Audio

A continuación el código `preprocess_audio(audio_path, sr=44032, duration=1)` convierte el archivo de audio en una matriz numpy con una frecuencia de muestreo de 44,032 Hz y una duración de 1 segundo.

Usa Librosa para cargar el audio y ajustar su tamaño para que coincida con la longitud requerida por el modelo.

6.21 Inferencia con el Modelo

`Infer(interpreter, audio_input)` ejecuta la inferencia con el modelo cargado en la Raspberry Pi. Obtiene los tensores de entrada y salida del modelo para procesar los datos de audio y determinar si los camarones están comiendo.

Figura 19.

Funciones de lecturas de valores

```
1 import numpy as np
2 import librosa
3 import tflite_runtime.interpreter as tflite
4
5 # Cargar el modelo .tflite
6 def load_model(model_path):
7     interpreter = tflite.Interpreter(model_path=model_path)
8     interpreter.allocate_tensors()
9     return interpreter
10
11 # Preprocesar el archivo de audio
12 def preprocess_audio(audio_path, sr=44032, duration=1):
```

Nota. El software Raspberry PI procesa los audios en un tiempo programado para dispersar el alimento (Thayer, 2020).

A continuación, se detalla en la tabla 1 el Prototipo de Alimentación Automática para Camarones Usando Reconocimiento de Sonido con Teachable Machine.

Tabla 1

Descripción del proceso del prototipo realizado.

| Aspecto | Descripción |
|--|--|
| Uso del sonido | El sonido se utiliza para activar la señal del alimentador automático. Este sistema es entrenado para la detección ciertos sonidos específicos generados por los camarones como su alimentación y su comportamiento en el agua. Este sonido envía la señal al alimentador para iniciar la dispersión del alimento. |
| Configuración del Dataset | Para crear el dataset, se grabarán diferentes sonidos generados por los camarones. Se incluirán sonidos de interacción con el agua, movimiento y ruidos externos alrededor. Estos sonidos se clasificarán y se etiquetarán para entrenar el modelo en Teachable Machine. |
| Medición de Datos de Camarones | En este prototipo, no realiza una medición directa de parámetros biológicos o de crecimiento de los camarones. El prototipo se enfocará en detectar su comportamiento sonoro como indicativo de que es momento de alimentación. |
| Simulación del Alimentador Automático | El prototipo utiliza un mecanismo automatizado que libera el alimento cuando el modelo de Teachable Machine detecta el sonido adecuado. El alimentador será simulado en un entorno controlado para asegurarse que la respuesta a los estímulos sonoros sea adecuada. |

Nota: Se presenta tabla que explica el proceso del prototipo.

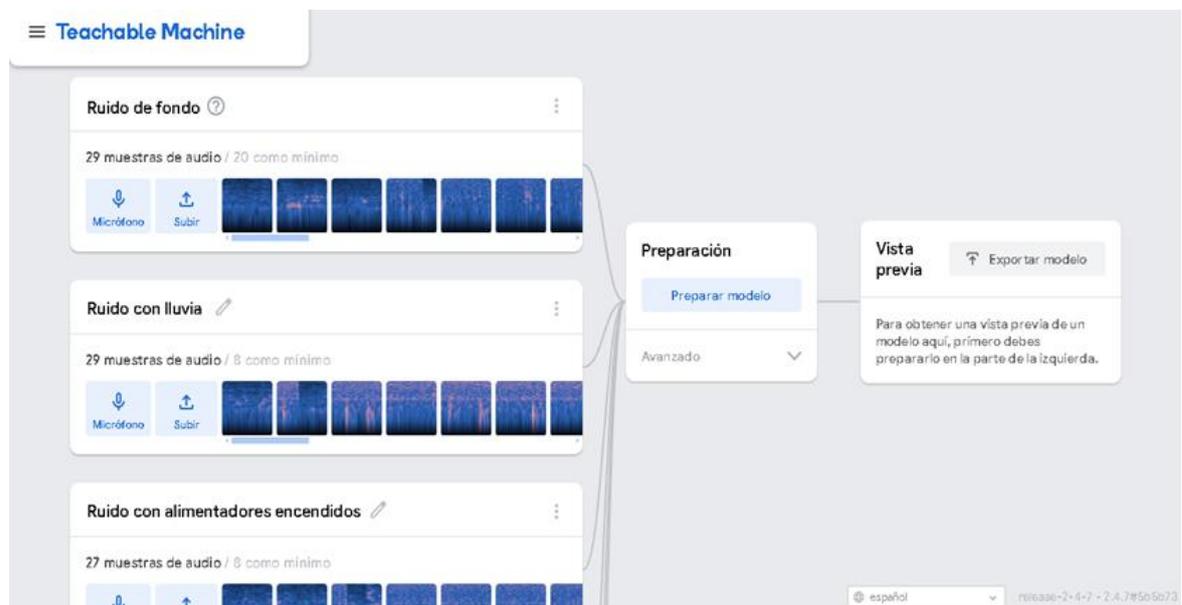
VII RESULTADOS

7.1 Entrenamiento del Modelo con Teachable Machine

- Observamos los sonidos que emiten los camarones al alimentarse y los estudiamos para crear un sistema que pueda identificarlos.
- El sistema Teachable Machine consiguió identificar con gran precisión superior al 90% cuando los camarones se encontraban completamente alimentados.

Figura 21.

Se presenta los audios que han sido entrenados en el programa Teachable Machine



Nota. Audios entregados de diferentes ruidos

7.2 Implementación de los Programas de Control del Motor

Se utilizaron dos programas diferentes para gestionar el funcionamiento del motor alimentador.

El motor alimentador fue probado con diferentes tiempos de activación 5, 10 y 15 segundos, observándose que el sistema podía ajustarse fácilmente según las necesidades específicas del cultivo.

- Códigos de programación para activación del alimentador automático
- Se realizaron varias pruebas para la activación del alimentador automático el equipo respondía correctamente a todos los giros que se le enviaba en el tiempo establecido.
- Se verificó que el sistema libere alimento solo cuando el modelo de IA detecta actividad alimenticia en los camarones.
- Su propósito es analizar los sonidos producidos por los camarones cuando están comiendo y en base en ello, activar el motor del alimentador automático.
- Se implementa en una Raspberry Pi para procesar los audios utilizando un modelo de Machine Learning entrenado con Teachable Machine.

7.3 Código Raspberry Pi de activación

- Este sistema Teachable Machine tiene como objetivo detectar y examinar todos los sonidos producidos por los camarones en su alimentación, empleando un modelo de reconocimiento capacitado anteriormente.
- Una vez que la Raspberry Pi identifica la actividad de alimentación, enciende el alimentador automático, mejorando el proceso sin requerir intervención humana.
- Liberando la cantidad de alimento necesaria en el tiempo exacto, cumpliendo eficazmente con sus requerimientos nutricionales de los camarones.
- Este enfoque además de minimizar el derroche y disminuir los gastos relacionados, ayuda a reservar la calidad del agua, fomentando un ambiente más saludable para el desarrollo de los camarones.

Figura 23.

Software Raspberry Pi



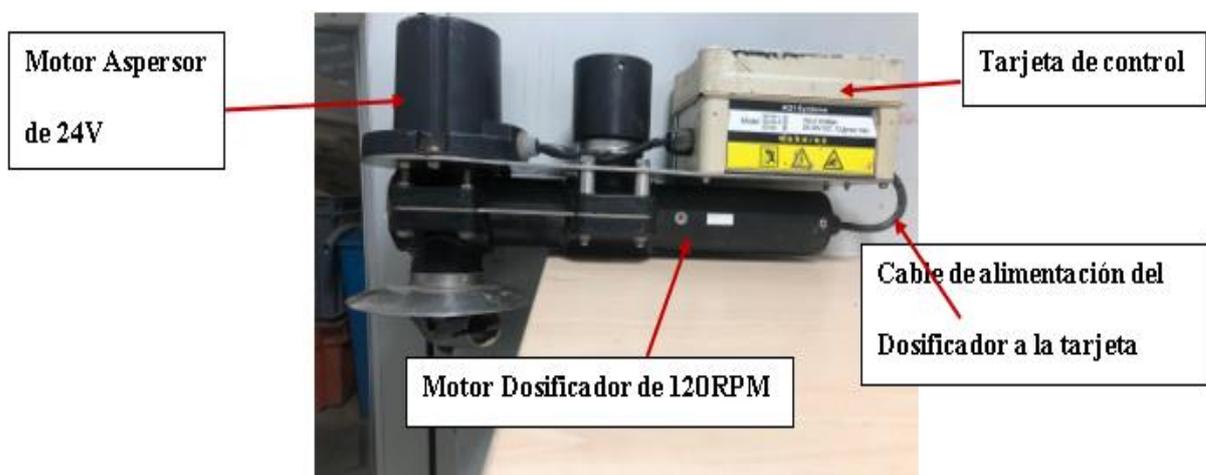
Nota. Software Raspberry desarrollar proyectos de electrónica o programación (C.Garcia, 2022)

7.4 Simulación y Validación del Sistema

- Se realizaron pruebas tanto en un espacio controlado como en condiciones reales.
- Comparamos el funcionamiento del sistema con los métodos tradicionales de alimentación, demostrando una reducción considerable en el desperdicio de alimento.

Figura 24.

Se presenta las partes que conforman al Alimentador Automático



Nota. Alimentador Automático un motor DC que se enciende con 24 V.

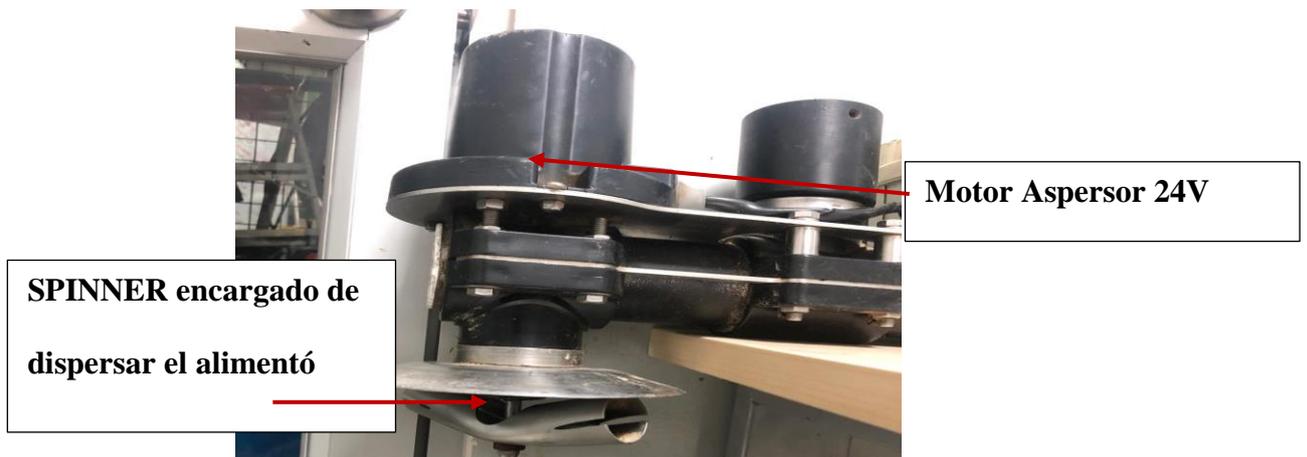
- Los hallazgos revelaron una mejora en la eficiencia alimentaria y la reducción de costos operativos.

7.5 Beneficios del prototipo

- Este sistema automático de alimentación se activa habitualmente cuando se percibe a los camarones comiendo y es proporciona el volumen ideal de alimento.
- De esta manera, se maximiza el aprovechamiento del alimento y el cultivo produce más ya que los camarones obtienen precisamente lo que requieren.
- Se logró un sistema que proporcione alimento de manera eficaz considerando la demanda real de los camarones.
- Se minimizó el impacto ambiental por el exceso de alimento no consumido.
- Se demostró que el uso de IA es una alternativa idónea para hacer que la acuicultura funcione mejor y sea más productiva.
- Disminución del desperdicio de alimento al iniciar la alimentación únicamente cuando los camarones realmente están comiendo.
- Mejora en el crecimiento de los camarones mediante una alimentación más regulada y diseñada en la demanda real.
- Automatización ayuda que el proceso de alimentación sea mucho más eficiente, reduciendo la necesidad de la intervención humana.

Figura 25.

Alimentador Automático



Nota. Se presenta al motor Spinner cuál es su función y con cuantos voltios trabaja.

7.6 Relación entre el Raspberry Pi y los audios generados por los camarones en el Teachable Machine

La integración de Teachable Machine, Raspberry Pi permitió desarrollar un sistema accesible y escalable ya que estas tecnologías facilitó la adopción del sistema en diferentes escalas de producción sin una inversión excesiva.

- Permitiendo cosechas más rápidas y con mayores rendimientos.
- Esto se traduce en una mayor rentabilidad para los productores de camarón.

Figura26.

Se presenta el código del ruido de lluvia

```
47     return output_data
48
49 # Función principal
50 def main():
51     # Ruta al archivo del modelo .tflite y al archivo de audio
52     model_path = 'camaron.tflite'
53     audio_path = 'Ruido con lluvia.wav'
54
55     # Cargar el modelo
56     interpreter = load_model(model_path)
```

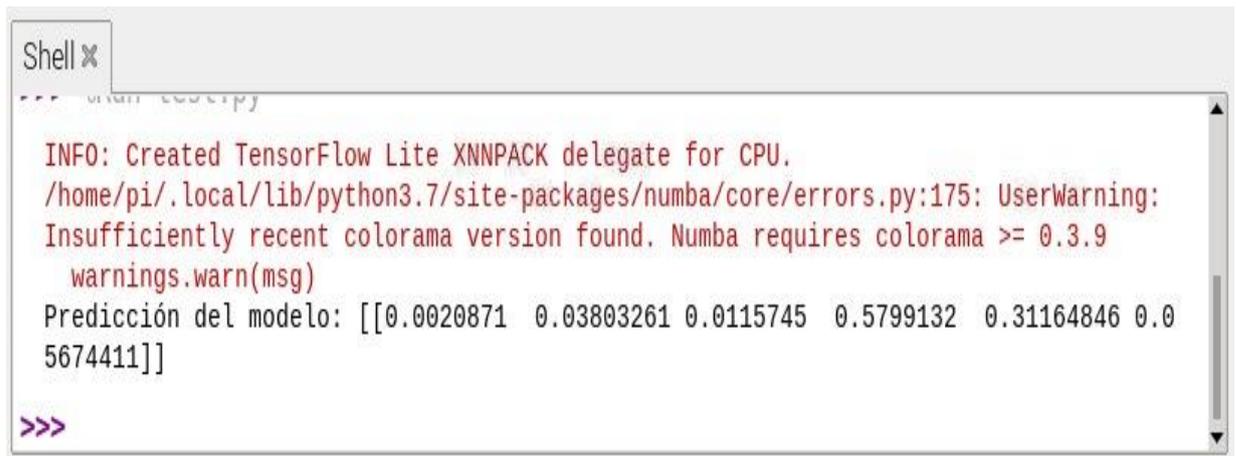
Nota. En este código de Raspberry Pi se presenta el audio entrenado en Teachable machine del ruido de lluvia y es exportado al Raspberry Pi.

7.7 Relación con el Proyecto de Alimentador Automático

- **Captura de Sonidos:** Se usan los audios generados por los camarones un micrófono en la **Raspberry Pi**.
- **Procesamiento en Tiempo Real:** Se analiza el audio con el modelo de Teachable Machine para detectar patrones de alimentación.

Figura 20

Se visualizan 5 formas del encendido del motor



```
Shell x
... main.py

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
/home/pi/.local/lib/python3.7/site-packages/numba/core/errors.py:175: UserWarning:
Insufficiently recent colorama version found. Numba requires colorama >= 0.3.9
  warnings.warn(msg)
Predicción del modelo: [[0.0020871 0.03803261 0.0115745 0.5799132 0.31164846 0.05674411]]
>>>
```

Nota. Resultados para activación del equipo (Thayer, 2020).

VIII Cronograma de Actividades

A continuación, se observa en la Tabla 2, el listado de actividades que se llevarán a cabo en el transcurso del proyecto hasta su finalización.

Tabla 2

Cronograma de Actividades

| Cronograma de Actividades | | | | | | | | | |
|--|-----------|-------|---------|-------|-------|------|-------|-------|--------|
| Actividades | Diciembre | Enero | Febrero | Marzo | Abril | Mayo | Junio | Julio | Agosto |
| Investigación y Planificación | | | | | | | | | |
| Investigación sobre sistemas de alimentación en cultivos de camarones. | X | X | | | | | | | |
| Investigación sobre sistemas de alimentación en cultivos de camarones. | | X | X | | | | | | |
| Investigación sobre sistemas de alimentación en cultivos de camarones. | X | X | X | | | | | | |
| Desarrollo del prototipó | | | | | | | | | |
| Formulación de propuesta y redacción. | X | | | | | | | | |
| Elaboración del Modelo y planificación técnica. | X | | | | | | | | |
| Desarrollo y entrenamiento del Modelo Teachable Machine | | | X | | | | | | |
| Implementación de Hardware | | | | | | | | | |
| Validación y optimización del Modelo. | | | X | | | | | | |
| Integración del programa Raspberry PI | X | X | | | | | | | |
| Integración y pruebas con el motor alimentador. | X | | X | | | | | | |
| Instalación y procesamiento del Raspberry Pi. | X | X | | | | | | | |
| Diseño de la interfaz | | | | | | | | | |

| | | | | |
|--|---|---|---|--|
| Diseño y simulación del prototipo. | | X | | |
| Desarrollo y validación de pruebas con el software. | X | | | |
| Integración de funciones de control y monitoreo. | | | X | |
| Implementación de Software | | | | |
| Desarrollo del Software de control del robot. | X | X | | |
| Integración del software Teachable Machine. | X | | | |
| Programación de algoritmos de simulación. | X | | X | |
| Pruebas y depuración | | | | |
| Pruebas de funcionalidad del alimentador automático en entorno controlado. | X | | | |
| Identificación y solución de errores. | | X | | |
| Documentación y Presentación | | | | |
| Preparación de informes técnicos y manuales de usuario. | X | X | X | |

Nota. Se presentan las fechas para el proceso junto a cada una de sus actividades.

IX Presupuesto

A continuación, se detalla en la Tabla 3, el costo de los materiales utilizados para el desarrollo del prototipo.

Tabla 3
Presupuesto

| DETALLE | CANTIDAD | COSTO UNITARIO | COSTO TOTAL |
|------------------------------------|----------|-------------------|--------------|
| Horas de ingeniería | 120 | \$2.93 | \$ 353 |
| Router | 1 | \$20 | \$20 |
| Protoboard | 1 | \$10 | \$ 10 |
| Teclado USB | 1 | \$10 | \$10 |
| Tarjeta de memoria microSD 32Gb | 1 | \$9 | \$9 |
| Total | | | \$402 |

Nota. Se detallan los gastos que se realizaron para el desarrollo del proyecto técnico.

X Conclusiones

El modelo de alimentación automática para camarones de granja usando Teachable Machine ha demostrado ser una solución nueva y eficaz para la industria acuícola. Esto asiste en la competitividad de todas partes de camarones. Al observar los ruidos que hacen los langostinos durante el acto de comer, el sistema puede saber cuándo están comiendo, cuándo quieren comer y cuándo dejan de comer.

Al activar el mecanismo cuando se necesita, este método no sólo es útil para usar la comida, sino que reduce el desperdicio y ayuda a tener una mejor sostenibilidad y responsabilidad con el medio ambiente en la industria acuática en el cuidado de los camarones.

En definitiva, el diseño y simulación del prototipo de alimentación automática para camarones de cultivo usando sonidos de su actividad de comer, procesados a través de Teachable Machine, muestra un paso grande en la automatización de la acuicultura. Poniendo este sistema ayuda a comer más bien y justo, bajando el desperdicio de comida y mejorando el crecimiento de los camarones. Con el uso de inteligencia artificial para saber qué sonidos son sobre la comida se logra respuesta automática rápido, haciendo mejor el cuidado y bajando costos del trabajo.

XI Recomendaciones

Se espera que mediante la realización de pruebas exhaustivas en diferentes configuraciones del modelo de Teachable Machine se pueda optimizar el prototipo de alimentación automática y rendimiento para los camarones de cultivo. Se debe tener una base de datos sólida y diversa de audios de camarones en diferentes condiciones ambientales para mejorar la precisión del reconocimiento de un sonido asociado con la alimentación. Y, por último, se necesita usar filtros de ruido y algoritmos de preprocesamiento de audio para prevenir activaciones falsas del sistema debido a sonidos no relacionados con la alimentación.

Debe realizarse pruebas y ajustes constantemente para asegurarse de que el modelo funcione adecuadamente. Debe realizarse algún test de pruebas de conectar y desconectar para ver cómo funciona.

Incluir una notificación si el sistema está funcionando mal o está presentando algún error se debería crear una forma de monitorear en tiempo real a través de celular o computadora.

XII REFERENCIAS

- Angel, J. (2022). *Alimentadores automáticos eléctricos y solares para camarón en Ecuador*. Obtenido de <https://bluesensordata.com/feeder>
- Balnova. (2021). *Alimentadores Automáticos Robotilsa: Desempeño y resultados*. Obtenido de <https://www.balnova.com/alimentadores-automaticos-robotilsa-desempeno-y-resultados/>
- BIOAQUAFLOC. (2021). *TECNOLOGÍA SIMBIÓTICA BIOAQUAFLOC*. Mexico.
- Bolivar, A. (2022). *Acuicultura local: tecnologías, desarrollo y genética*.
- C.Garcia. (2022). *¿Qué es Raspberry Pi y para qué sirve?* Mexico: Codeathlon Autumn Edition .
- ELECTRÓNICA, R. (2022). *¿Qué es y para qué sirve el Arduino?* Cartagena: Guellcom.
- Freire, M. (2024). *Maximizando Producción de Camarón con Alimentadores Automáticos*. Obtenido de <https://ventasgp.com/blog/f/maximizando-producci%C3%B3n-de-camar%C3%B3n-con-alimentadores-autom%C3%A1ticos>
- Gavilanez, O. (2021). Obtenido de Repositorio: <https://repositorio.upse.edu.ec/bitstream/46000/6609/1/UPSE-TBM-2021-0010.pdf>
- Hernández. (2023). *Desarrollo e implementación de clasificadores de imágenes mediante el empleo de*. 68-79.
- Ibertrónica. (2023). *Monitor HDMI – ¿Qué son?* España.
- Jaramillo, M. (2021). *Diseño y ensamble de un prototipo de alimentador automático ecológico para piscinas de cría de camarón*. Obtenido de <http://repositorio.ucsg.edu.ec/handle/3317/16238>
- João Reis, P. (2022). *Cómo los alimentadores automáticos de camarones impactan la calidad del agua y el suelo de los estanques de engorde en Ecuador*. Ecuador: Shrimp Innovation LATAM, BioMar – Ecuador.
- Lara, M. (2022). *Alimentacion Automatico*. Obtenido de <https://jetfeeder.com/>
- Laumayer. (2021). *¿QUÉ ES UN ROUTER? BENEFICIOS, CARACTERÍSTICAS Y RECOMENDACIONES PARA SOLUCIONES ADECUADAS*. Medellin.
- Llugcha, L. (2021). *Alimentacion Automatica*. Obtenido de <https://nicovita.com/blog/principios-de-alimentacion-automatica/>
- Martín, J. (2024). *La Industrial*. Obtenido de <https://www.laindustrialeventos.com/motores-paso-a-paso-aplicaciones-industriales/>
- Mera, S. (2022). *MOLINOS S.A.S*. Obtenido de <https://molinoschampion.com/en-que-consisten-los-alimentadores-automaticos/>

Murillo, L. (2020). *Sistemas de alimentaciòn*. Obtenido de <https://www.fishfarmfeeder.com/es/componentes-sistemas-alimentacion-piscifactorias/>

Sanchez, M., & Figueroa, J. (2022). *Diseño de un sistema alimentador automático para una finca camaronera ubicada en la zona Tendales, cantón El Guabo*. Obtenido de <http://www.dspace.espol.edu.ec/handle/123456789/57333>

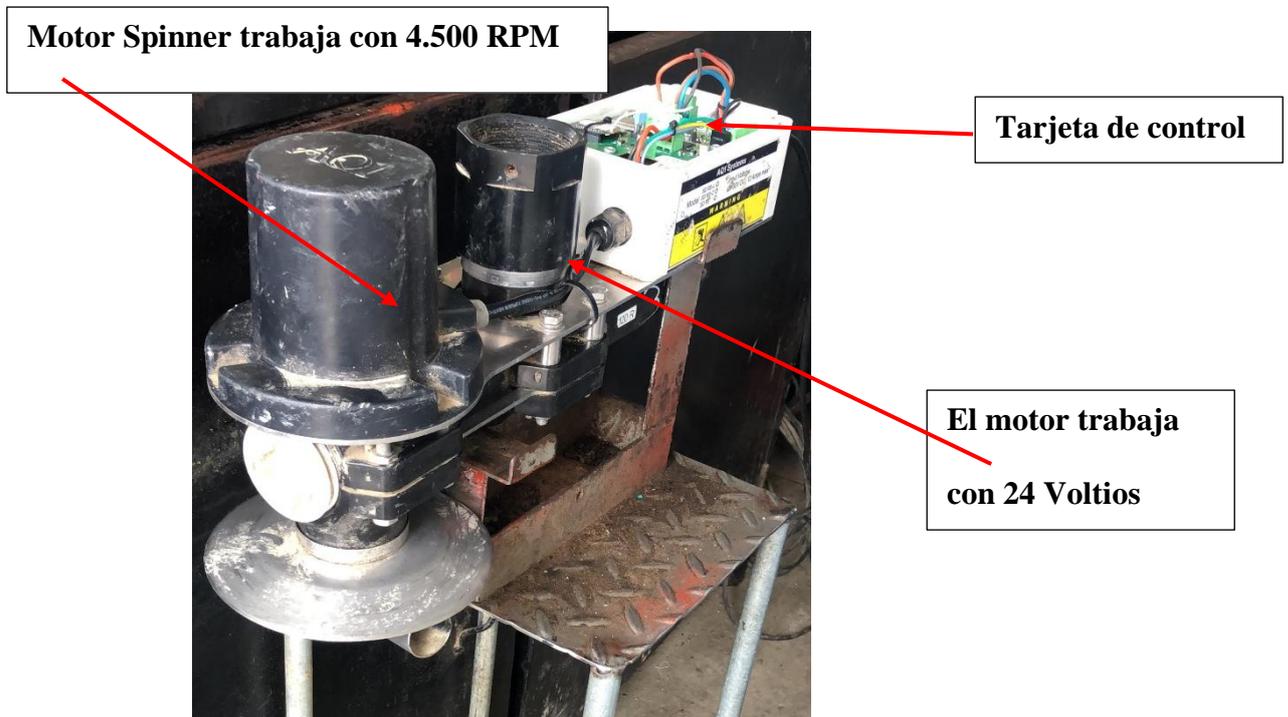
Thayer, L. (2020). *¿Que es Raspberry Pi?* Santiago, Chile: MCI.

Torres, L. (2022). *Sistema de Alimentacion*. Obtenido de <https://www.fishfarmfeeder.com/es/componentes-sistemas-alimentacion-piscifactorias/>

XIII Anexos

Figura 27.

Se presenta al motor Alimentador Automático.



Nota. Se presenta alimentador automático el nombre comercial del motor es SD100 y pertenece AQ1 que significa tecnología inteligente y sostenible encargado de dispersar el alimento en un tiempo programado y en el momento exacto para reducir el desperdicio de alimento. esta empresa es la que desarrollo estos equipos que son utilizados en la acuicultura para la alimentación de los camarones.

motor recibe esta señal y empieza a dosificar según la necesidad de alimentación del camarón.

Figura 29.

Las siguientes capturas del programa Teachable Machine presenta los audios generados por los camarones.



Nota. Se visualizan los audios entrenados del ruido y el proceso de alimentación de los camarones, estos audios son extraídos de un sensor llamado hidrófono el cual su proceso es escuchar las señales acústicas en todo momento, pero en la acuicultura su principal función es escuchar la masticación del camarón ya que luego estas señales son receptadas al hidrófono donde luego con la plataforma AQ1 Systems se observa la respuesta de alimentación en graficas.

Figura 30.

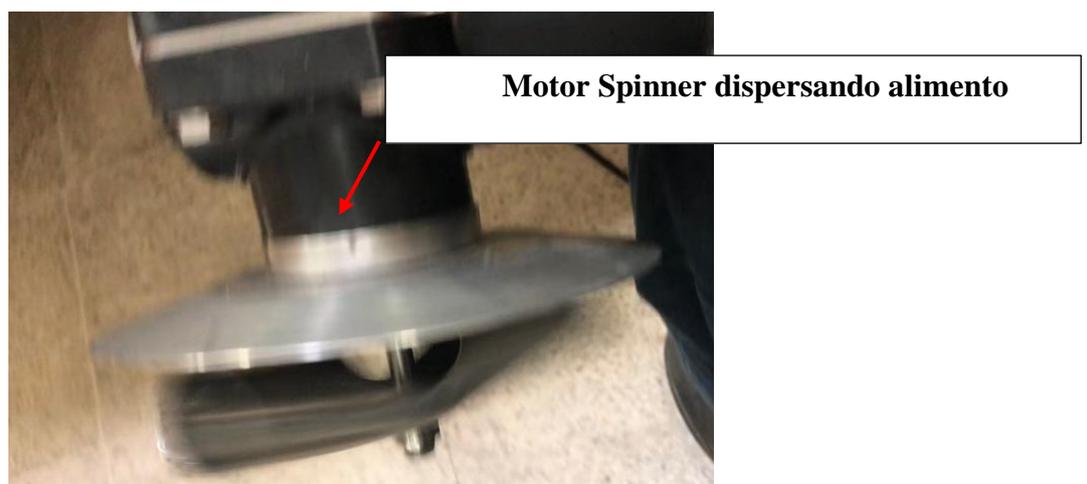
Se presenta capturas del programa Raspberry Pi el encargado de revivir la información para la activación del equipo.

```
f.py x
1 import numpy as np
2 import librosa
3 import tflite_runtime.interpreter as tflite
4 import RPi.GPIO as GPIO
5 import time
6
7 # Definir los pines de control
8 PIN_PWM = 18 # GPIO para el motor (PWM)
9 RELE_PIN = 17 # GPIO para el dosificador
10
11 # Configurar los pines
12 GPIO.setmode(GPIO.BCM)
13 GPIO.setup(PIN_PWM, GPIO.OUT)
```

Nota. Esto indica que el script ejecuta los modelos de Machine Learning generados de la masticación de los camarones y las interferencias generando una salida con varias probabilidades o valores numéricos para hacer que el alimentador dosifique alimento ya que luego se realizó pruebas de giro de dosificación del alimentador, activándose al recibir o detectar que los camarones quieren comer.

Figura 31.

Como se observa en la figura el alimentador automático está dispersando alimento al recibir la señal.



Nota. El motor se activa y procesa la señal enviada apenas recibe la información de los audios entrenados por el programa Teachable Machine y el Raspberry Pi que hace la activación directamente.

Códigos

El siguiente código implementa el software en el Raspberry Pi que cuenta con un modelo de aprendizaje automático que clasifica los audios generados por los camarones y dependiendo de la clasificación, el motor y el motor dosificador son controlados a través de los pines GPIO de la Raspberry Pi. La ejecución comienza con la configuración de los pines GPIO que son necesarios para el control del motor y el dosificador, donde se utilizan señales PWM (modulación de ancho de pulso) en el pin del motor para controlar su velocidad y un relé para activar el dosificador. El código controla las librerías RPi y GPIO para poder realizar las acciones anteriormente mencionadas.

```
import numpy as np
import librosa
import tflite_runtime.interpreter as tflite
import RPi.GPIO as GPIO
import time

# Definir los pines de control
PIN_PWM = 18 # GPIO para el motor (PWM)
RELE_PIN = 17 # GPIO para el dosificador

# Configurar los pines
GPIO.setmode(GPIO.BCM)
GPIO.setup(PIN_PWM, GPIO.OUT)
GPIO.setup(RELE_PIN, GPIO.OUT)

# Configurar PWM en el pin del motor
pwm_motor = GPIO.PWM(PIN_PWM, 1000) # Frecuencia de 1kHz
pwm_motor.start(0) # Iniciar con el motor apagado

# Funciones para controlar el motor
def set_motor_speed(duty_cycle, duration):
    pwm_motor.ChangeDutyCycle(duty_cycle)
```

La clasificación del audio se ejecuta cargando el modelo de TensorFlow Lite que fue entrenado previamente y que está dotado de un reconocedor con la capacidad de identificar sonidos en particular como el de camarones comiendo normalmente este se carga y se preprocesa utilizando la librería librosa.

```
time.sleep(duration)
pwm_motor.ChangeDutyCycle(0) # Apagar motor

def activar_dosificador():
    GPIO.output(RELE_PIN, GPIO.HIGH)
    time.sleep(2)
    GPIO.output(RELE_PIN, GPIO.LOW)

# Cargar etiquetas desde archivo
def load_labels(file_path):
    labels = {}
    with open(file_path, 'r') as file:
        for line in file:
            index, label = line.strip().split(maxsplit=1)
            labels[int(index)] = label
    return labels

# Cargar el modelo .tflite
def load_model(model_path):
    interpreter = tf.lite.Interpreter(model_path=model_path)
    interpreter.allocate_tensors()
    return interpreter

# Preprocesar el archivo de audio
def preprocess_audio(audio_path, sr=44032, duration=1):
    audio, _ = librosa.load(audio_path, sr=sr, duration=duration)
    if len(audio) < sr * duration:
        audio = np.pad(audio, (0, sr * duration - len(audio)))
    else:
        audio = audio[:sr * duration]
    return np.array(audio, dtype=np.float32)

# Realizar la inferencia en el modelo
def infer(interpreter, audio_input):
    input_details = interpreter.get_input_details()
    output_details = interpreter.get_output_details()
    input_data = np.expand_dims(audio_input,
axis=0).astype(np.float32)
    interpreter.set_tensor(input_details[0]['index'], input_data)
```

El código después de realizar la predicción pasa a realizar una comparación entre la etiqueta que se ingresó y una que se propuso que es camarones comiendo normalmente, a continuación, el código enciende el motor y el dosificador, pero primero se enciende el relé y el motor se enciende en tres velocidades: baja, media y alta.

```
interpreter.invoke()
return interpreter.get_tensor(output_details[0]['index'])

# Función principal
def main():
    model_path = 'camaron.tflite'
    audio_path = 'Comiendo normalmente (1).wav'
    labels_path = 'labels.txt'

    labels = load_labels(labels_path)
    interpreter = load_model(model_path)
    audio_input = preprocess_audio(audio_path)
    output = infer(interpreter, audio_input)

    predicted_index = np.argmax(output)
    predicted_label = labels.get(predicted_index, "Desconocido")
    print("Etiqueta predicha:", predicted_label)

    if predicted_label == "Camarones comiendo normalmente":
        print("Iniciando motor y dosificador...")
        set_motor_speed(15, 5)
        activar_dosificador()
        time.sleep(3)

        set_motor_speed(20, 5)
        activar_dosificador()
        time.sleep(3)

        set_motor_speed(30, 5)
        activar_dosificador()
    else:
        print("Etiqueta no coincide, motor no activado.")

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        print("Interrupción detectada. Apagando...")
    finally:
        pwm_motor.stop()
        GPIO.cleanup()
```