



**UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
CARRERA COMPUTACIÓN**

**TRABAJO DE TITULACIÓN COMO REQUISITO PREVIO
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**

**DISPENSADOR AUTOMATIZADO PARA ALIMENTO DE
POLLOS EN LA HACIENDA “CASTILLO” DEL CANTÓN
NARANJAL, PROVINCIA DEL GUAYAS**

AUTOR

ROMERO MENDOZA EDDY ELIUTH

TUTOR

ING. SÁNCHEZ PALACIOS LUIS ENRIQUE, MSC

**NARANJAL, ECUADOR
2026**



**UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
CARRERA COMPUTACIÓN**

APROBACIÓN DEL TUTOR

El suscrito, docente de la Universidad Agraria del Ecuador, en mi calidad de Tutor, certifico que el presente trabajo de titulación: DISPENSADOR AUTOMATIZADO PARA ALIMENTO DE POLLOS EN LA HACIENDA “CASTILLO” DEL CANTÓN NARANJAL, PROVINCIA DEL GUAYAS, realizado por el estudiante ROMERO MENDOZA EDDY ELIUTH; con cédula de identidad N°0929879377 de la carrera COMPUTACIÓN, Unidad Académica Programa Regional de Enseñanza Naranjal, ha sido orientado y revisado durante su ejecución; y cumple con los requisitos técnicos y legales exigidos por la Universidad Agraria del Ecuador; por lo tanto, se aprueba la presentación del mismo.

Atentamente,

Ing. Sánchez Palacios Luis MSc.

Milagro, 19 de mayo del 2026



**UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
CARRERA COMPUTACIÓN**

APROBACIÓN DEL TRIBUNAL DE SUSTENTACIÓN

Los abajo firmantes, docentes designados por el H. Consejo Directivo como miembros del Tribunal de Sustentación, aprobamos la defensa del trabajo de titulación DISPENSADOR AUTOMATIZADO PARA ALIMENTO DE POLLOS EN LA HACIENDA "CASTILLO" DEL CANTÓN NARANJAL, realizado por ROMERO MENDOZA EDDY ELIUTH; el mismo que cumple con los requisitos exigidos por la Universidad Agraria del Ecuador.

Atentamente,

Ing. De La A Salinas Leonela, MSc.

PRESIDENTE

Ing. Lascano Montes Ariana. MSc

EXAMINADOR PRINCIPAL

Ing. Ostaiza Clavijo Ginger. MSc

EXAMINADOR PRINCIPAL

Ing. Sánchez Palacios Luis, MSc.

EXAMINADOR SUPLENTE

Milagro, 19 de mayo del 2026

DEDICATORIA

Dedico este trabajo, ante todo, a Dios, fuente de sabiduría y fortaleza, quien me ha guiado en cada paso y me ha permitido superar los desafíos de este camino.

A mis padres, por su amor incondicional, por creer siempre en mí y por enseñarme que la disciplina y el esfuerzo son las verdaderas claves del éxito. Su ejemplo ha sido mi mayor inspiración para llegar hasta aquí.

A mi familia, que con sus palabras de aliento me ha impulsado a continuar incluso en los momentos más difíciles.

Agradezco también a las personas que confiaron en mí y me acompañaron en este proceso, brindándome su apoyo, comprensión y paciencia. Este logro es el resultado de años de dedicación, perseverancia y fe; por ello, cada página de esta tesis representa un paso más hacia los sueños que juntos hemos construido.

AGRADECIMIENTO

Mi más profundo agradecimiento a Dios por concederme la fortaleza, claridad y constancia necesarias para culminar esta etapa tan importante de mi formación profesional.

A mi tutor, Ing. Luis Enrique Sánchez Palacios, MSc., por su guía académica, sus valiosos consejos y por compartir su experiencia con paciencia y compromiso durante todo el desarrollo de este proyecto.

A la Universidad Agraria del Ecuador, por brindarme los conocimientos, las herramientas y el espacio para crecer tanto personal como profesionalmente.

A mis padres y seres queridos, quienes han sido el motor que me impulsó a no rendirme y a dar lo mejor de mí. Gracias por su amor, apoyo y confianza, sin los cuales este logro no habría sido posible.

Este trabajo no solo representa una meta alcanzada, sino también el reflejo del esfuerzo conjunto de quienes han creído en mi capacidad de cumplir mis objetivos.

Autorización de Autoría Intelectual

Yo, ROMERO MENDOZA EDDY ELIUTH, en calidad de autor del proyecto realizado, sobre DISPENSADOR AUTOMATIZADO PARA ALIMENTO DE POLLOS EN LA HACIENDA "CASTILLO" DEL CANTÓN NARANJAL, realizado por ROMERO MENDOZA EDDY ELIUTH, por la presente autorizo a la UNIVERSIDAD AGRARIA DEL ECUADOR, hacer uso de todos los contenidos que me pertenecen o parte de los que contienen esta obra, con fines estrictamente académicos o de investigación.

Los derechos que como autor me correspondan, con excepción de la presente autorización, seguirán vigentes a mi favor, de conformidad con lo establecido en los artículos 5, 6, 8; 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento.

Milagro, 19 de mayo del 2026

Romero Mendoza Eddy Eliuth
C.I. 0929879377

RESUMEN

En la producción avícola de la Hacienda “Castillo”, en el cantón Naranjal, el método tradicional de alimentación manual genera problemas críticos como el reparto desigual del balanceado, desperdicio constante de alimento. Ante esta realidad, este proyecto de investigación se desarrolló con el objetivo de diseñar e implementar un dispensador automatizado que optimice la distribución del alimento dentro del galpón. Para el desarrollo del sistema, se empleó un enfoque metodológico tecnológico y práctico, construyendo un prototipo electrónico con el uso de sensores para el pesaje exacto de las porciones, medidores de variables ambientales y un detector del nivel de llenado del contenedor. Asimismo, todo el ecosistema se vinculó a una plataforma web que permite almacenar la información en una base de datos. Como principales resultados, el dispositivo demostró la capacidad de suministrar el alimento de forma autónoma, precisa y adaptada a las necesidades del dueño de la hacienda, facilitando además que los encargados supervisen el consumo en tiempo real y descarguen reportes en hojas de cálculo. En conclusión, la implementación de este dispensador automatizado constituye una solución tecnológica viable, eficiente y de fácil adopción que reduce significativamente los costos de operación, mitiga las pérdidas de alimento y eleva la productividad general de la hacienda.

Palabras clave: *avicultura, control de alimento, dispensador automatizado, monitoreo en tiempo real, plataforma web.*

ABSTRACT

In poultry production at the “Castillo” farm, located in the Naranjal canton, the traditional manual feeding method generates critical problems such as uneven feed distribution and constant feed waste. In response to this situation, this research project was developed with the objective of designing and implementing an automated dispenser to optimize feed distribution inside the poultry house.

For the development of the system, a technological and practical methodological approach was employed, involving the construction of an electronic prototype using sensors for precise portion weighing, environmental variable monitoring, and container fill-level detection. In addition, the entire system was integrated with a web platform that enables data storage in a database.

The main results demonstrated that the device is capable of supplying feed autonomously, accurately, and according to the needs of the farm owner. Furthermore, it allows staff members to monitor consumption in real time and download reports in spreadsheet format.

In conclusion, the implementation of this automated dispenser represents a viable, efficient, and easily adaptable technological solution that significantly reduces operating costs, minimizes feed losses, and improves the overall productivity of the farm.

Keywords: *automated dispenser, feed control, poultry farming, real-time monitoring, web platform.*

ÍNDICE GENERAL

1. INTRODUCCIÓN	13
1.1 Antecedentes del problema	13
1.2 Planteamiento y formulación del problema.....	14
1.3 Justificación de la investigación	14
1.4 Delimitación de la investigación	15
1.5 Objetivo general	15
1.6 Objetivos específicos.....	15
2. MARCO TEÓRICO	16
2.1 Estado del arte	16
2.2 Bases científicas y teóricas de la temática	17
2.3 Marco Legal.....	22
3. MATERIALES Y MÉTODOS	23
3.1 Enfoque de investigación	23
3.2 Metodología.....	23
4. RESULTADOS	28
4.1 Diseñó en un simulador el dispensador automatizado de alimento para pollos, integrando sensores y actuadores conectados al microcontrolador.	28
4.2 Codificación del microcontrolador para la activación de los sensores y actuadores de manera eficiente, asegurando un suministro uniforme del alimento.	28
4.3 Implementación el dispensador automatizado en la Hacienda “Castillo” del cantón Naranjal.	29
5.DISCUSIÓN	31
6. CONCLUSIONES Y RECOMENDACIONES	34
6.1 Conclusiones.....	34

6.2 Recomendaciones.....	34
BIBLIOGRAFÍA	35
ANEXOS	38

ÍNDICE DE ANEXOS

Anexo N° 1: Figura 1. Diagrama de Flujo	38
Anexo N° 2: Figura 2. Esquema electrónico del proyecto en fritzing	38
Anexo N° 3: Figura 3. Codificación del ESP32	39
Anexo N° 4: Figura 4. Entrevista Realizada al propietario	39
Anexo N° 5: Figura 5. Encuesta Realizada a los trabajadores	40
Anexo N° 6: Figura 6. Visita del tutor para supervisar el proyecto	40
Anexo N° 7: Tabla 1. Elementos electrónicos para el dispensador automatizado	47
Anexo N° 8: Tabla 2. Elementos para verificación y supervisión del sistema	48
Anexo N° 9: Tabla 3. Consumo semanal alimento para pollos	48
Anexo N° 10: Tabla 4. Registro de dispensación de alimento	49
Anexo N° 11: Tabla 5. Resumen diario del consumo de alimento	48
Anexo N° 12: Acta de entrega y cierre del sistema	50
Anexo N° 13: Tabla 6. Base de datos Firebase	59

ÍNDICE DE APÉNDICE

Apéndice N° 1: MANUAL TÉCNICO.....	50
Apéndice N° 2: MANUAL DE USUARIO.....	82

1. INTRODUCCIÓN

1.1 Antecedentes del problema

En el cantón Naranjal, Ecuador, la avicultura es una actividad económica clave, pero la mayoría de las granjas aún no han implementado comedero automatizado para pollos, lo que limita su productividad. La ausencia de tecnología como los comederos automáticos impide una distribución eficiente del alimento, afectando la conversión alimenticia y el crecimiento uniforme de las aves. Además, el desperdicio de alimento y la falta de monitoreo en tiempo real aumentan los costos operativos y reducen la rentabilidad del sector.

En el Ecuador la Asociación de Médicos Veterinarios Especialistas en Avicultura (AMEVEA), estima el consumo per cápita de carne de ave fluctúa entre 30 y 32 kilogramos al año, donde ésta es la proteína de mayor consumo en este país. En cuanto a la producción, el volumen anual está situado entre 230 y 250 millones de aves de engorde. (Castillo, 2022, p. 16)

En Ecuador han demostrado que los sistemas automatizados pueden reducir el tiempo de distribución del alimento hasta en un 8.6 % en comparación con los métodos manuales, además de garantizar una distribución equitativa con una diferencia mínima de peso entre los comederos. Esto se traduce en un incremento del 10 % en el peso final de los pollos al cierre del ciclo de producción, beneficiando la rentabilidad del productor.

Se llevaron estudios durante un periodo de tiempo donde a los comederos automáticos fueron: consumo 4.53 kg/ave, peso 2.890 kg, conversión alimenticia 1.57, índice de eficiencia productiva europea 413, mortalidad 5.01% 4.36 x Kg. Mientras que, para los comederos manuales se obtuvieron: consumo 4.48 kg/ave, peso 2.736 kg, conversión alimenticia 1.64, índice de eficiencia productiva europea 385, mortalidad 3.46% y S/. 4.67 x Kg en los costos. (Broncano, 2022, p. 11)

En base a los resultados obtenidos, se concluyó que los comederos automáticos son más rentables que los manuales, porque se evidencia una relación directa entre el tipo de equipo utilizado y la performance del ave logrando que en menores días de crianza tenga un mejor peso y por ende menor costo productivo.

El sector avícola representó el 30% del Valor Bruto de Producción Agropecuaria, en el mes de mayo del 2022, representado en un 22% por carne de pollo, 4.1% otras aves y 4.1% huevo de gallina. Durante ese periodo las principales regiones productoras con mayor aporte fueron Lima (53.3 por ciento), La Libertad (18.2 por ciento), Arequipa (10.6 por ciento) e Ica (5.7 por ciento). (Sistema Integrado de Estadísticas Agrarias Peru, 2022, p. 6)

En el Ecuador existe granjas dedicadas tanto a la producción como a la cría de aves de engorde que se toma en cuenta que la carne del pollo es el principal alimento de bajo costo y el más consumido, en la actualidad se producen cerca de 220 millones de aves al año, el cual implica un crecimiento del 400% desde la década de los 90. (Guerra J. , 2019, p. 18)

Se midió el índice de eficiencia productiva europea (IEP), un indicador clave en la avicultura. Con los comederos automáticos, el valor fue 413, mientras que con los manuales fue 385. Este aumento refleja un mejor desempeño general de las aves cuando tienen acceso a un sistema de alimentación más preciso y constante.

En España la sistematización de los procesos de alimentación conlleva la utilización de automatismos y productos cada vez más técnicos y mejor pensados. En el caso de los comederos automáticos para pollos los avicultores están eligiendo comederos que sean asequibles y eficientes que permitan proporcionar una mejor gestión de los piensos aprovechando toda su calidad. (AVINEWS, 2024, p. 15)

1.2 Planteamiento y formulación del problema

1.2.1 Planteamiento del problema

En la actualidad, la producción avícola a enfrentado diversos desafíos relacionados con la eficiencia en la alimentación, el desperdicio de alimento y la uniformidad en el crecimiento de las aves. En muchas granjas avícolas, el proceso de alimentación aún se realiza de manera manual, lo que provoca problemas como la distribución no equitativa del alimento, mayor esfuerzo físico para los trabajadores, riesgo de contaminación y variabilidad en el peso final de los pollos. Además, esta dependencia del trabajo manual implica costos laborales significativos que podrían reducirse mediante la automatización.

1.2.2 Formulación del problema

Dado las dificultades que tiene la Hacienda “Castillo”. A continuación, procedemos a formular la siguiente pregunta:

¿De qué manera la implementación de comederos automatizados para pollos en la Hacienda “Castillo” del Cantón Naranjal Provincia del Guayas influiría en la distribución de alimentos?

1.3 Justificación de la investigación

El presente proyecto se desarrolló con el propósito de evaluar la implementación de un prototipo automatizado de alimentación para la Finca “Castillo” en el cantón Naranjal, Ecuador. Utilizando un microcontrolador, sensores y actuadores, el sistema busca optimizar la distribución del alimento, enfocándose

en la necesidad de modernizar el sector avícola local, mejorar la eficiencia en la alimentación, reducir el desperdicio de insumos y aumentar la rentabilidad de los productores.

1.4 Delimitación de la investigación

- **Espacio:** Este trabajo se llevó a cabo en la Finca “Castillo”, ubicada en el cantón Naranjal (provincia del Guayas), con el objetivo de implementar un dispensador automatizado de alimento para pollos. El espacio destinado es de 4m x 3m.
- **Tiempo:** El presente trabajo de titulación tuvo un tiempo de duración de 7 meses.
- **Población:** la recolección de la información se llevó a cabo dentro de la Hacienda “Castillo”, en la cual nos encontramos con el dueño de la finca y los trabajadores.

1.5 Objetivo general

Implementar un dispensador automatizado para alimento de pollos mediante el uso de sensores y actuadores conectados a un microcontrolador para optimizar la distribución de alimentos en la Hacienda “Castillo” del cantón Naranjal provincia del Guayas.

1.6 Objetivos específicos

- Diseñar en un simulador el dispensador automatizado de alimento para pollos, integrando sensores y actuadores conectados al microcontrolador.
- Codificar el microcontrolador para la activación de los sensores y actuadores de manera eficiente, asegurando un suministro uniforme del alimento.
- Implementar el dispensador automatizado en la Hacienda “Castillo” del cantón Naranjal.

2. MARCO TEÓRICO

2.1 Estado del arte

La automatización del monitoreo ambiental en la avicultura se ha convertido en una estrategia esencial para optimizar la eficiencia productiva, reducir costos y mejorar el bienestar animal. En un estudio reciente, Sortica (2023) demostró que la implementación de un sistema automatizado basado en sensores de temperatura, humedad y luminosidad permitió mantener las condiciones ambientales dentro de rangos ideales, lo que resultó en una reducción del 50% en errores de medición manual y una mejora significativa en la capacidad de respuesta ante variaciones climáticas. Además, se observó que mantener la humedad relativa entre el 50% y el 70%, y ajustar la temperatura semanalmente de 35 °C a 26 °C según la edad de las aves, favoreció su desarrollo saludable. Este tipo de automatización no solo facilita la toma de decisiones en tiempo real mediante IoT industrial (IIoT), sino que también representa una solución escalable para pequeños y medianos productores.

El estudio de Broncano (2022) realizado en Perú comparó comederos automáticos y manuales, revelando que los sistemas automatizados mejoraron la conversión alimenticia (1.57 vs. 1.64), aumentaron el peso promedio de las aves (2.890 kg vs. 2.736 kg) dando como resultado en la eficiencia y productividad al momento de implementar la automatización.

AVINEWS (2024) informó sobre innovaciones en comederos automáticos que priorizan la accesibilidad del pollito al pienso y la higiene en el proceso, mostrando la modernización del manejo avícola. La implementación de sensores y controladores ha permitido una distribución más equitativa del alimento, mejorando su crecimiento y evitando pérdidas.

De acuerdo con lo planteado por Guerra (2019) documentó la implementación de un comedero automatizado en una finca experimental, donde se evidenció una reducción significativa en el tiempo de alimentación y un aumento en la ganancia de peso. El autor destacó la necesidad de adoptar tecnologías accesibles para pequeños y medianos productores.

Para Diaz (2021) la implementó un sistema automatizado de bajo costo en un galpón avícola de Barranquilla, demostrando que la automatización del suministro de alimento y agua redujo las pérdidas por estrés térmico (del 15% al 5%), mejoró el peso promedio de los pollos (4% más en 4 semanas) y disminuyó el

desperdicio de alimento (de 500g a 100g por pollo semanal), evidenciando que incluso en pequeñas granjas la automatización optimiza la productividad avícola.

En el proyecto de Novoa (2020) el sistema automatizado alimentado por energía solar en granjas avícolas de Gachetá, Colombia, demostró que el control preciso de temperatura, humedad y ventilación mediante sensores DHT11 y actuadores redujo la mortalidad en un 30% y mejoró el peso final de las aves en un 15%. La automatización del suministro de alimento y agua con electroválvulas y sensores ultrasónicos disminuyó el desperdicio de insumos en 40%, mientras que la integración de IoT permitió monitoreo remoto y trazabilidad de datos.

Para Bohorquez (2024) integrar tecnologías de control y monitoreo para garantizar procesos más estables, seguros y confiables, disminuyendo el margen de error humano y mejorando la productividad. Esta transformación tecnológica no solo optimiza los recursos disponibles, sino que también favorece la adaptabilidad de las empresas a las exigencias de un mercado cada vez más dinámico.

De acuerdo a Benavides (2022) la implementación de sistemas automatizados “mejora la calidad de los procesos, garantiza la precisión en la ejecución de tareas y contribuye significativamente al ahorro de recursos operativos”. Esta visión general refuerza la necesidad de adoptar soluciones tecnológicas que no solo modernicen el entorno laboral, sino que también generen un impacto positivo en la productividad y sostenibilidad de las actividades económicas.

2.2 Bases científicas y teóricas de la temática

En palabras de Roa (2018) la incorporación de tecnologías automatizadas en la avicultura representa una solución práctica y eficiente frente a los desafíos que enfrenta este sector, especialmente en contextos rurales. Este tipo de innovación tecnológica no solo mejora la productividad, sino que también democratiza el acceso a soluciones modernas para pequeños y medianos productores que tradicionalmente no contaban con herramientas de este tipo.

Para Guerra (2021) la implementación de sistemas automatizados en la avicultura se ha consolidado como una solución eficiente para optimizar la alimentación de las aves, reducir el desperdicio de insumos y mejorar significativamente los indicadores productivos. Estudios recientes demuestran que el uso de comederos automáticos contribuye al aumento del peso promedio de los pollos, mejora la conversión alimenticia y disminuye la carga operativa en las

granjas. Este tipo de tecnología permite una distribución precisa y constante del alimento, adaptándose a las condiciones del entorno y favoreciendo la rentabilidad del productor. En este contexto, el diseño del presente sistema automatizado se fundamenta en la integración de sensores, actuadores y microcontroladores para garantizar un funcionamiento autónomo y eficiente, el cual estará conformado por:

2.2.1 Microcontrolador ESP32

Según Gomez (2022) incorporar el microcontrolador ESP32 en proyectos agrícolas, especialmente en ambientes como invernaderos. Gracias a sus capacidades de conexión inalámbrica, su bajo consumo energético y la facilidad para integrar sensores, se ha logrado mejorar el monitoreo y control de variables importantes como la temperatura, humedad o riego. Esto no solo optimiza los recursos, sino que también facilita la toma de decisiones y fomenta el uso de tecnologías accesibles para pequeños y medianos productores.

Ortega (2022) señala que una arquitectura IoT de bajo costo para invernaderos, basada en microcontroladores con Wi-Fi, permite a productores pequeños monitorear temperatura y humedad y acceder a los datos desde la nube para tomar decisiones de riego más oportunas.

El microcontrolador ESP32 es el cerebro del dispensador automático. Su capacidad para conectarse por Wi-Fi, junto con su bajo consumo energético, lo convierte en la mejor opción para proyectos agrícolas inteligentes. En la Hacienda "Castillo", este microcontrolador permitirá controlar la salida del alimento, recibir información del sensor de temperatura y registrar los datos recolectados diariamente. Incluso pudiéndose observar desde el navegador del dispositivo móvil para supervisar remotamente el funcionamiento del sistema, ayudando a tomar decisiones más rápidas y reducir pérdidas.

2.2.2 Servomotor

Fernandez (2025) afirma que los servomotores ofrecen una gran ventaja en proyectos donde se necesita movimiento preciso y controlado, como en la automatización industrial, la robótica o sistemas agrícolas automatizados. A diferencia de otros motores, los servomotores permiten controlar con exactitud la posición, velocidad y el ángulo de giro, lo cual es esencial para tareas que requieren movimientos suaves y repetibles.

Park (2023) describen sistemas de alimentación ganadera automatizada en los que el uso de servomotores para accionar compuertas de dosificación garantiza

movimientos precisos y repetibles, asegurando que cada ciclo entregue la misma cantidad de alimento.

El servomotor es el encargado de controlar la compuerta del dispensador, es decir, abrir o cerrar el paso del alimento de forma precisa. A diferencia de otros motores que solo giran sin control, el servo puede detenerse en un ángulo exacto, lo cual es esencial para administrar porciones adecuadas sin desperdiciar alimento, permitiendo repetir el proceso las veces que sea necesario sin fallos mecánicos.

2.2.3 Sensor DHT11

Tal como lo evidencia Martínez (2024) el sensor DHT11 es una solución económica y versátil para el monitoreo agrícola, ya que mide temperatura (0-50°C) y humedad relativa (20-90%) con señal digital precalibrada, facilitando su integración en sistemas IoT como el ESP32. Su bajo costo y fácil adaptabilidad lo hacen ideal para prototipos de agricultura de precisión, permitiendo el envío de datos a dashboards en la nube para optimizar el riego y prevenir riesgos climáticos, aunque su tiempo de respuesta (6-30 segundos) y margen de error ($\pm 5\%$) lo limitan a `es no críticas.

Bhongale (2019) indica que en proyectos de agricultura con IoT el DHT11 se emplea como sensor digital económico para medir temperatura y humedad del aire en invernaderos, enviando los datos a una plataforma remota para apoyar el control automático del riego y la ventilación.

El sensor DHT11 cumple una función preventiva en este sistema. Detecto la temperatura y humedad ambiental dentro del área de crianza de los pollos. Si, por ejemplo, se registran temperaturas elevadas que puedan afectar el bienestar de las aves, el sistema podrá limitar o pausar temporalmente la dispensación de alimento para evitar complicaciones digestivas. Su facilidad de conexión lo hacen ideal para un entorno como el de la finca.

2.2.4 Celda de carga

Pérez (2023) señala que las celdas de carga son componentes esenciales en diversos sectores industriales debido a su capacidad para medir fuerzas y pesos con precisión. En el ámbito médico, se utilizan en equipos como básculas de incubadoras para bebés y camas de hospital, garantizando mediciones exactas para el cuidado de pacientes. En la industria, facilitan el pesaje de materiales a granel y el control de procesos automatizados, mejorando la eficiencia y seguridad.

Además, su aplicación en agricultura permite optimizar cosechas y gestionar el almacenamiento en silos, contribuyendo al rendimiento productivo.

Kumar (2023) propone un sistema de alimentación automática para animales donde una celda de carga bajo el comedero registra continuamente la cantidad de alimento dispensado y el peso restante, lo que permite ajustar las raciones y detectar anomalías en el suministro.

La celda de carga permite medir con exactitud cuánto alimento se ha distribuido cada día. Este sensor actúa como una báscula digital instalada en el compartimento del dispensador. Cada vez que se libera alimento, la celda detecta el peso y lo envía al microcontrolador para su registro. Esto no solo garantiza un control riguroso del consumo diario, sino que también ayuda a ajustar las raciones y detectar posibles fallas si el alimento no se está entregando como se espera.

2.2.5 Sensor TCRT5000

Según Escudero (2023), el sensor TCRT5000 combina un emisor de luz infrarroja y un fototransistor en una carcasa que evita interferencias externas. Está diseñado para detectar variaciones en la luz reflejada, lo que le permite reconocer cambios en la proximidad o presencia de objetos. Este sensor opera con una longitud de onda cercana a los 950 nm y puede captar pequeñas variaciones en corriente logrando una precisión efectiva para recolección de datos.

Vishay (2019) explica que los sensores ópticos reflectivos como el TCRT5000 se utilizan para detectar la presencia de objetos mediante la reflexión de luz infrarroja, siendo adecuados para confirmar si un material ha pasado o se encuentra en una posición determinada dentro de sistemas automatizados.

En el proyecto de comedero automatizado para pollos, el sensor TCRT5000 es implementado como detector óptico para identificar si el alimento ha sido correctamente dispensado al plato. Este sensor se coloca estratégicamente cerca del recipiente, de modo que pueda verificar la presencia del pienso mediante la detección de reflexión de luz infrarroja. Cuando el sensor no detecte alimento en la zona esperada, se activa una señal que indicará la necesidad de dispensar una nueva ración. Lo convierten en una solución ideal para mantener la eficiencia y control en el sistema de alimentación automatizada en la Hacienda "Castillo".

2.2.6 Pantalla LCD 16x2

Ulcuango (2024) da a conocer que la pantalla LCD 16x2 es un dispositivo de visualización alfanumérico compuesto por dos filas y dieciséis columnas,

ampliamente utilizada en proyectos electrónicos debido a su simplicidad y eficiencia para mostrar información básica. Está equipada con retroiluminación LED azul y utiliza el controlador HD44780, lo que permite una programación estándar y compatible con diversos microcontroladores. Su versatilidad y disponibilidad en diferentes combinaciones de tamaños la convierten en un componente esencial en sistemas que requieren mostrar datos como lecturas de sensores o estados operativos de manera clara.

Hernández (2021) reportan en un sistema de monitoreo agrícola que las pantallas LCD 16x2 se emplean como interfaz local para mostrar al operador las lecturas de sensores y el estado de los actuadores, permitiendo verificar en sitio el correcto funcionamiento del sistema sin requerir equipos adicionales.

La pantalla LCD 16x2 es utilizada como una herramienta visual del sistema del dispensador automatizado, ya que permite mostrar en tiempo real la cantidad de alimento restante en el compartimento. Esta función facilita al encargado de la finca conocer de manera inmediata cuándo es necesario recargar el dispensador, evitando así que las aves se queden sin alimento y asegurando una alimentación continua.

2.3 Marco Legal

Art. 281.- El Estado garantizará la soberanía alimentaria como un objetivo estratégico; para ello, será responsabilidad del Estado: fomentar la producción agropecuaria sustentable; conservar y recuperar la agrobiodiversidad y los recursos naturales productivos; promover políticas redistributivas que permitan el acceso del campesinado a los factores de producción; establecer mecanismos preferentes de financiamiento para la producción, comercialización e industrialización de productos agropecuarios; incentivar el desarrollo de la ciencia y la tecnología en la producción agropecuaria; garantizar el desarrollo de organizaciones y redes de productores y consumidores; y proteger la producción agroalimentaria nacional y el ingreso de las personas que se dedican a ella

Art. 385.- El Sistema Nacional de Ciencia, Tecnología, Innovación y Saberes Ancestrales tendrá por objeto generar, adaptar, innovar y difundir conocimientos científicos y tecnológicos, saberes ancestrales, para responder a las necesidades y prioridades del desarrollo del país, al buen vivir. Impulsará el desarrollo de tecnologías y la innovación con pertinencia social, particularmente en áreas estratégicas para el desarrollo nacional y para el mejoramiento de la productividad, la competitividad y la calidad de la vida. (Constitución de la República del Ecuador, 2008, p.108)

Art. 8.- La protección del derecho de autor recae sobre todas las obras del ingenio, en el ámbito literario o artístico, cualquiera que sea su género, forma de expresión, mérito o finalidad. Los derechos reconocidos por el presente Título son independientes de la propiedad del objeto material en el cual está incorporada la obra y su goce o ejercicio no están supeditados al requisito del registro o al cumplimiento de cualquier otra formalidad. Las obras protegidas comprenden, entre otras, las siguientes: a) Libros, folletos, impresos, epistolarios, artículos, novelas, cuentos, poemas, crónicas, críticas, ensayos, misivas, guiones para teatro, cinematografía, televisión, conferencias, discursos, lecciones, sermones, alegatos en derecho, memorias y otras obras de similar naturaleza, expresadas en cualquier forma. (Ley de propiedad intelectual, 2014, p.142)

3. MATERIALES Y MÉTODOS

3.1 Enfoque de investigación

3.1.1 Tipo de investigación

El enfoque que se utilizó es:

Cuantitativo: Este tipo de investigación adoptó un enfoque cuantitativo, centrado en la medición objetiva de variables asociadas al funcionamiento del dispensador automatizado en la Hacienda “Castillo”. Se recopiló datos numéricos a través de sensores de peso, temperatura y nivel de alimento, lo cual permitió analizar con precisión indicadores como la cantidad de alimento distribuido por ciclo y la reducción de desperdicios.

3.1.2 Tipo y alcance de la investigación

Se trató de una investigación aplicada, ya que buscó solucionar un problema práctico mediante el diseño e implementación de un prototipo funcional.

El alcance fue descriptivo y explicativo, porque describió las condiciones actuales de la alimentación avícola en la finca y explicó los beneficios del sistema automatizado.

3.1.3 Diseño de investigación

No experimental: Debido a que no se manipulan deliberadamente las variables independientes en un ambiente controlado. El estudio se realizó en un periodo determinado, sin observación prolongada en el tiempo.

3.2 Metodología

3.2.1 Recolección de datos

3.2.1.1. Recursos

El sistema automatizado requirió de los trabajadores encargados en la Hacienda “Castillo”, quienes fueron responsables de supervisar el funcionamiento general del dispensador y reponer el alimento en la tolva cuando sean notificados por la aplicación móvil. Estas personas recibieron una capacitación básica para el uso, mantenimiento y revisión periódica del prototipo. La simplicidad del sistema permitió que su operación sea eficiente. En paralelo, el autor del proyecto tuvo un rol activo durante todas las fases de implementación: análisis del entorno, diseño en simulador, programación del microcontrolador ESP32, integración de sensores y montaje físico. También fue responsable de realizar pruebas funcionales, recopilar datos operativos y validar el correcto desempeño del sistema dentro de la Hacienda “Castillo”.

Además, se utilizaron fuentes bibliográficas confiables, incluyendo artículos académicos, proyectos similares, manuales técnicos y literatura científica sobre automatización agrícola, sensores aplicados a la avicultura y control mediante microcontroladores. Estos recursos respaldaron el diseño, justificación y funcionamiento del sistema implementado.

El sistema contó con componentes clave como:

- Celda de carga: instalada en la base de la tolva, permitió medir el peso del alimento restante.
- Sensor TCRT5000: ubicado en el plato, detectó la presencia de alimento para verificar que la dispensación se realizó correctamente.
- Sensor DHT22: midió la temperatura y humedad en el ambiente de crianza.
- ESP32: microcontrolador que gestionó todo el sistema y permitió enviar notificaciones a una aplicación móvil.
- Servomotor: abrió o cerró la compuerta del dispensador en función de la lógica programada.
- Pantalla LCD 16x2: mostró en tiempo real el peso de alimento en la tolva.

El prototipo fue diseñado y probado previamente en el simulador Fritzing antes de ser implementado físicamente, lo que redujo errores y facilitó ajustes.

Adicional se elaboró un presupuesto estimado para la construcción del prototipo; adquisición de componentes electrónicos y materiales de ensamblaje. El mismo se presenta en la sección de anexos (**Ver anexo. Tabla 1 y 2**).

3.2.1.2. Métodos y técnicas

3.2.1.2.1. Métodos.

Método inductivo:

El método inductivo parte de la observación y el análisis de casos particulares para llegar a conclusiones generales. Se utilizó comúnmente en investigaciones donde los datos empíricos permiten identificar patrones, tendencias o comportamientos repetitivos que luego son generalizados en principios o conclusiones más amplias.

En este sentido, se analizaron los datos recolectados automáticamente por el sistema del dispensador, como el peso diario del alimento distribuido, la

frecuencia con la que se vacía el plato y el comportamiento de consumo de los pollos en distintos horarios. A partir de esta información empírica, se identificaron patrones de alimentación que permitieron ajustar el sistema para optimizar la distribución y minimizar el desperdicio.

Método deductivo:

El método deductivo parte de conocimientos generales aceptados como válidos para llegar a conclusiones específicas. Se basa en la lógica y la aplicación de principios previamente conocidos para interpretar o predecir situaciones concretas dentro del objeto de estudio.

Desde esta perspectiva, se tomaron en cuenta principios reconocidos sobre eficiencia alimentaria en la crianza de pollos, como la importancia de una distribución uniforme y continua del alimento, o la necesidad de evitar el exceso durante condiciones de calor. Estos conocimientos sirvieron como base para configurar el sistema automatizado, programar el microcontrolador y establecer criterios de operación en función de la temperatura, el peso dispensado y la detección de plato vacío.

Método analítico:

El método analítico consiste en descomponer un fenómeno en sus partes o variables fundamentales para entender su funcionamiento y las relaciones entre ellas. Permite estudiar cómo interactúan diferentes factores entre sí y cómo influyen en el resultado final.

De esta manera, se examinaron de forma detallada variables como el peso del alimento entregado (medido por la celda de carga), la temperatura y humedad del ambiente (detectadas por el sensor DHT11), y la presencia o ausencia de alimento en el plato (detectada por el sensor TCRT5000). El análisis de estas variables ayudó a establecer relaciones causa-efecto entre las condiciones del entorno, el funcionamiento del sistema y el comportamiento alimenticio de las aves, aportando insumos clave para mejorar el prototipo.

3.2.1.2.2. Técnicas:

Se aplicaron una entrevista al dueño y una encuesta a los trabajadores de la Hacienda “Castillo” para conocer la dinámica actual de alimentación y validar la funcionalidad del sistema automatizado desde la perspectiva del usuario final. Además, se realizó la observación directa durante las pruebas de campo para

ajustar detalles de funcionamiento y asegurar que el sistema se adapte bien al entorno productivo.

Scrum

Para la gestión del desarrollo del dispensador automatizado en la Hacienda “Castillo”, se aplicó la metodología ágil Scrum, adaptada a las características de un proyecto académico con enfoque en prototipado electrónico. Esta metodología permitió dividir el proyecto en ciclos cortos llamados sprints, facilitando la organización del trabajo, la evaluación constante de avances y la mejora continua del sistema automatizado.

Sprints definidos:

Sprint 1: Recolección de información y análisis del problema

Se realizaron entrevistas al propietario de la finca y encuestas a los trabajadores para conocer las rutinas de alimentación actuales, problemas frecuentes y necesidades específicas. También se observó directamente el entorno físico y se definieron los requerimientos técnicos iniciales.

Sprint 2: Diseño del prototipo

Se elaboró un modelo virtual del dispensador en el software Fritzing, conectando los sensores, actuadores y el microcontrolador ESP32. Esta etapa permitió detectar errores y validar el diseño sin necesidad de construir físicamente el sistema.

Sprint 3: Programación del microcontrolador

Se desarrolló el código del sistema utilizando Arduino IDE. Se integraron los módulos para el control del servomotor, la lectura de la celda de carga (para medir el peso de alimento en la tolva), el sensor DHT11 (para registrar temperatura y humedad), y el sensor TCRT5000, ubicado en el plato para detectar la presencia o ausencia de alimento (**Ver anexo. Figura 4**).

Sprint 4: Construcción del prototipo físico

Se ensamblaron los componentes en una estructura física adecuada para la finca. Se instalaron los sensores, se conectó el microcontrolador y se realizaron pruebas iniciales del dispensador.

Sprint 5: Evaluación y ajustes del sistema

Durante esta fase, se monitoreó el funcionamiento del prototipo en condiciones reales. Se recolectaron datos operativos (peso diario dispensado,

fallos, lecturas de sensores) y se realizaron ajustes en el código y hardware para mejorar la precisión, eficiencia y confiabilidad del sistema.

3.2.2 Población y muestra

3.2.2.1. Población

La población estuvo compuesta por las personas directamente involucradas en el proceso de alimentación avícola dentro de la Hacienda “Castillo”, ubicada en el cantón Naranjal. Esta población estuvo conformada por los trabajadores operativos y el propietario de la finca, quienes desempeñan funciones relacionadas con la supervisión, manejo y ejecución de las actividades de alimentación de los pollos.

3.2.2.2. Muestra

En este caso no se seleccionó una muestra, ya que el tamaño de la población es reducido y manejable. Por tanto, se trabajó con la totalidad de la población, lo que permitió realizar un análisis exhaustivo sin necesidad de aplicar técnicas de muestreo estadístico o no estadístico.

3.2.3 Análisis estadístico

Se aplicaron herramientas estadísticas descriptivas para interpretar los datos recolectados, como la cantidad de alimento distribuido por día, la frecuencia de recargas y la temperatura ambiental. Se calcularon medidas como media, desviación estándar y frecuencia de eventos. Además, se pudo realizar un análisis comparativo con registros anteriores para evaluar la mejora en eficiencia. Estos análisis se realizaron utilizando hojas de cálculo y software básico, garantizando claridad en la interpretación de resultados (**Ver anexo. Tabla 4**).

3.2.4 Diagrama de flujo

El flujo del sistema inicio con la activación del microcontrolador ESP32. El sistema verifica si la cantidad de alimento en la tolva (medido por la celda de carga) es suficiente. Luego, en función del tiempo o una lógica definida, se activa el servomotor para dispensar alimento. El sensor TCRT5000 detecta si hay alimento presente en el plato. Si no lo detecta, se realiza una nueva dispensación. Todos los datos se visualizan en la pantalla LCD y se registran. Si el peso en la tolva está por debajo del mínimo establecido, se envía una alerta a la aplicación móvil de la persona encargada para que reponga alimento (**Ver anexo. Figura**

4. RESULTADOS

4.1 Diseño en un simulador el dispensador automatizado de alimento para pollos, integrando sensores y actuadores conectados al microcontrolador.

Se realizó inicialmente una encuesta a los trabajadores de la Hacienda “Castillo”, con el propósito de identificar el método actual de alimentación, el tiempo empleado y los principales inconvenientes del proceso manual. Los resultados obtenidos evidenciaron una alta dependencia del trabajo manual, así como la presencia de desperdicio de alimento y tiempos prolongados en la rutina diaria (**Ver Anexo. Figura 6**).

A partir de esta información, se procedió al diseño del sistema en un entorno de simulación, utilizando el software Fritzing. En este entorno se integraron los componentes electrónicos principales del proyecto: el microcontrolador ESP32, el sensor DHT11 para temperatura y humedad, el sensor óptico TCRT5000, la celda de carga con el módulo HX711, el servomotor SG90 y la pantalla LCD 16x2.

La simulación permitió validar las conexiones eléctricas y la lógica de control antes de construir el prototipo físico. Se comprobó la correcta comunicación entre los sensores y el microcontrolador, verificándose que el sistema respondía adecuadamente a las condiciones definidas: lectura de peso, temperatura, humedad y detección de alimento en el plato.

Como resultado del proceso de simulación, se obtuvo un modelo funcional virtual del dispensador automatizado, el cual permitió optimizar la disposición de los componentes y reducir posibles errores de cableado o incompatibilidades eléctricas. Este diseño sirvió como base para la posterior implementación física del sistema, disminuyendo riesgos durante el montaje real y facilitando la transición hacia la siguiente etapa del proyecto. (**Ver anexo. Figura 3**).

4.2 Codificación del microcontrolador para la activación de los sensores y actuadores de manera eficiente, asegurando un suministro uniforme del alimento.

En cumplimiento del segundo objetivo específico, se desarrolló la programación del microcontrolador ESP32 utilizando el entorno Arduino IDE, incorporando las bibliotecas oficiales necesarias para el funcionamiento de cada componente del sistema.

El código implementado permitió la lectura continua de los valores de peso a través del módulo HX711, así como la medición de temperatura y humedad mediante el sensor DHT11. De igual manera, se programó la activación del servomotor en función de los datos obtenidos por la celda de carga y la detección del nivel de alimento en el plato a través del sensor óptico TCRT5000.

Durante esta fase, se logró establecer una lógica de control automatizada, en la cual el sistema evaluó constantemente las condiciones de operación y tomó decisiones de forma autónoma, asegurando una dispensación uniforme del alimento. La información relevante del sistema, como el estado de operación y las lecturas de los sensores, fue visualizada en tiempo real mediante la pantalla LCD 16x2, facilitando la supervisión del proceso.

Las pruebas funcionales demostraron que el sistema fue capaz de dispensar alimento en intervalos regulares, manteniendo estabilidad en las lecturas y una correcta respuesta del servomotor. Además, se evidenció una mejora en el control de las porciones entregadas, lo que permitió reducir el desperdicio de alimento en comparación con el método manual previamente utilizado en la hacienda. Estos resultados confirmaron la eficiencia del algoritmo implementado y la correcta integración del software con el hardware del sistema **(Ver anexo. Figura 4)**.

4.3 Implementación el dispensador automatizado en la Hacienda “Castillo” del cantón Naranjal.

El prototipo del dispensador automatizado fue instalado físicamente en el galpón avícola de la Hacienda “Castillo”, en un área aproximada de 4 m x 3 m, previamente seleccionada por su accesibilidad y condiciones ambientales. La estructura del sistema fue construida con materiales livianos y resistentes a la humedad, garantizando su adecuada adaptación al entorno productivo.

Una vez instalado, el sistema fue sometido a un periodo de pruebas de siete días consecutivos, durante los cuales operó de manera continua sin presentar fallos técnicos. Durante este tiempo, se observó que el dispensador realizó la distribución del alimento de forma automática y constante, reduciendo significativamente la intervención manual por parte de los trabajadores **(Ver anexo. Tabla 3)**.

De acuerdo con los reportes del personal de la finca, se evidenció una reducción del tiempo diario dedicado a la alimentación, así como una mejora en la organización del trabajo. Asimismo, el sistema mostró un comportamiento estable bajo condiciones ambientales propias de la zona, con temperaturas de hasta 33 °C y niveles de humedad relativa cercanos al 75 %, lo que confirmó su fiabilidad y adaptabilidad al contexto real de producción avícola. ()

En conjunto, los resultados obtenidos durante la implementación demostraron que el dispensador automatizado cumplió con los objetivos planteados, ofreciendo una solución funcional que optimiza la distribución del alimento, reduce el esfuerzo operativo y mejora la eficiencia del proceso de alimentación en la Hacienda “Castillo” (**Ver anexo. Tabla 5**).

5.DISCUSIÓN

El presente trabajo tuvo como objetivo la implementación de un dispensador automatizado para alimento de pollos mediante el uso de sensores y actuadores que se conectaron a un microcontrolador, con el único propósito de optimizar la distribución del alimento en la Hacienda “Castillo” del cantón Naranjal, provincia del Guayas. Los resultados evidencian que la automatización facilita el control del proceso de alimentación, reduciendo la mano de obra y obteniendo beneficios en el bajo desperdicio del alimento.

El estudio de Bootwong (2025) evalúa un prototipo de comedero automático de una sola línea de alimentación en una granja de pollos de engorde de sistema cerrado, comparándolo con el método tradicional de suministro manual de alimento. El sistema automático mantiene los platos constantemente llenos, lo que permite que las aves accedan al alimento en todo momento y reduce la competencia por el consumo, el amontonamiento y el riesgo de que los pollos se pisen durante la alimentación. Como resultado, la mortalidad disminuye en torno a un 18%, la tasa de supervivencia aumenta y el índice de conversión alimenticia se mejora de 2,6 a 1,7, mostrando un impacto directo sobre la eficiencia productiva. Además, se evidencia que la inversión en este tipo de tecnología puede recuperarse en menos de dos meses, debido al incremento en rendimiento y al mejor aprovechamiento del alimento. Estos hallazgos respaldan la idea de que la automatización del suministro favorece simultáneamente el bienestar de las aves, la productividad y la rentabilidad de la explotación avícola.

En la fase de diseño, la simulación del sistema permitió validar la lógica de control y las conexiones eléctricas antes de la construcción física. Esta verificación previa fue clave para disminuir errores de cableado y ajustar la integración de componentes, de modo que el prototipo responda adecuadamente a la lectura de peso, la detección de alimento en el plato y la visualización de información mediante la pantalla LCD. Durante la codificación y las pruebas iniciales, la integración de la celda de carga con el módulo HX711 y el servomotor dio la posibilidad de una dosificación periódica del alimento, manteniendo un rango error inferior al 6% en las mediciones de peso. Asimismo, se registró una reducción del 8,4% en el desperdicio de alimento y una disminución del 40% en el tiempo diario destinado a

la alimentación, lo que sugiere mejoras en la eficiencia operativa frente a la práctica manual.

Los beneficios económicos del sistema automatizado desarrollado en este proyecto son consistentes con lo reportado por Lumban (2024), quien diseñó un sistema de alimentación programada basado en IoT para pollos de engorde. En su investigación, el autor determinó que el sistema IoT resulta 4,25 veces más eficiente en costos que el método convencional para una población de 5000 pollos. El sistema incluye dispositivos de automatización capaces de regular los volúmenes de alimento e implementar horarios regulares, permitiendo a los productores monitorear y gestionar la alimentación de los pollos de forma remota sin necesidad de estar físicamente presentes en la granja. Además, el análisis de escalabilidad garantiza una distribución uniforme del alimento, reduciendo el desperdicio y maximizando la utilización de este. Los autores concluyen que la implementación de este tipo de sistemas mejora la eficiencia y sostenibilidad en la producción avícola, contribuyendo positivamente a satisfacer la alta demanda futura de carne de pollo.

En el entorno de la Hacienda “Castillo”, el sistema operó de manera continua durante siete días y mostró adaptabilidad a condiciones ambientales de hasta 33 °C y 75% de humedad relativa. La incorporación del monitoreo de temperatura y humedad aporta un valor adicional, ya que estas variables influyen en el consumo de alimento y en el bienestar de las aves.

La reducción en los costos de personal y en el riesgo sanitario observada en el presente proyecto guarda relación con lo señalado por el equipo de trabajo Anoliefo (2021), quienes desarrollaron un sistema automatizado de alimentación para granjas avícolas utilizando un microcontrolador Arduino Uno. Los autores destacan que el alto costo y la elevada demanda de mano de obra asociados al sistema de cama profunda en avicultura representan un gran desafío para los productores. El sistema propuesto simula las funciones de los asistentes de granja en la entrega de alimento y agua a las aves en intervalos de tiempo específicos, detectando el nivel de alimento y agua en los respectivos comederos y bebederos y dispensándolos de manera inteligente en respuesta a los niveles detectados. Los resultados mostraron que el sistema es capaz de detectar y predecir eficazmente el nivel de alimento y agua en la granja con una precisión del 98,79% y dispensar los insumos dentro de un tiempo de respuesta de 50-60 ms. Este sistema también

mejora la gestión energética en la granja y reduce significativamente los costos de personal.

La automatización del suministro de alimento implementada en el presente trabajo es coherente con los objetivos planteados por Subedi (2024) en su sistema de gestión avícola inteligente basado en IoT. Estos autores desarrollaron un sistema que controla parámetros ambientales y automatiza la alimentación, ofreciendo un enfoque rentable, orientado a la calidad y al ahorro de recursos en la producción avícola. Al aprovechar dispositivos inteligentes y tecnologías IoT, el sistema puede reemplazar la mano de obra manual, abordando los desafíos laborales de la industria e introduciendo semiautomatización. El sistema abarca tareas como la alimentación mediante un contenedor, el suministro de agua a través de bebederos de nipple, la regulación de temperatura y humedad con ventiladores y calentadores, iluminación automática y monitoreo ambiental en tiempo real accesible a través de datos en la nube IoT. Esta solución tecnológica es prometedora para mejorar la productividad y la eficiencia en la producción avícola y en aplicaciones agrícolas más amplias.

Finalmente, aunque el prototipo cumplió con los objetivos planteados, se identifican oportunidades de mejora relacionadas con la robustez mecánica de la compuerta, la protección de componentes frente a polvo y humedad, y la calibración periódica de la celda de carga para mantener la precisión. De igual manera, la incorporación de alertas automáticas y un registro histórico de datos a lo largo de las semanas permitiría fortalecer el seguimiento del consumo y facilitar la planificación de reposición de alimento sin llegar a tener pérdidas.

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El diseño del dispensador automatizado de alimento para pollos mediante un simulador permitió integrar de forma precisa los sensores, actuadores y el microcontrolador, facilitando la visualización del funcionamiento del sistema antes de su implementación física, lo que ayudó a reducir errores en el ensamblaje y a optimizar el uso de los componentes electrónicos del proyecto.

La codificación del microcontrolador para la activación eficiente de sensores y actuadores permitió establecer un control automatizado del suministro de alimento. Esto fue fundamental para garantizar una distribución uniforme del alimento, minimizar el desperdicio y mejorar la eficiencia del proceso de alimentación en la hacienda.

La implementación del dispensador automatizado en la Hacienda “Castillo” representó una alternativa tecnológica viable para modernizar el proceso de alimentación avícola. El proyecto planteó una solución que contribuyó a optimizar el tiempo de trabajo, mejorar la gestión del alimento con una reducción del 8% en desperdicio de alimento y con un error inferior al 6% en la medición de peso.

6.2 Recomendaciones

Realizar pruebas adicionales del diseño en el simulador antes del ensamblaje físico definitivo, con el fin de verificar la correcta conexión y compatibilidad de los sensores, actuadores y el microcontrolador, asegurando así un montaje más preciso y eficiente del sistema.

Efectuar validaciones progresivas del código del microcontrolador, probando individualmente el funcionamiento de cada sensor y actuador, para garantizar un control automatizado estable del suministro de alimento y mantener una distribución uniforme con mínimo desperdicio.

Capacitar al personal de la Hacienda “Castillo” en el uso y mantenimiento del dispensador automatizado, así como considerar la incorporación de mejoras futuras al sistema, tales como monitoreo remoto o ajustes automáticos en función de las condiciones ambientales, con el propósito de fortalecer la productividad del sistema avícola.

BIBLIOGRAFÍA

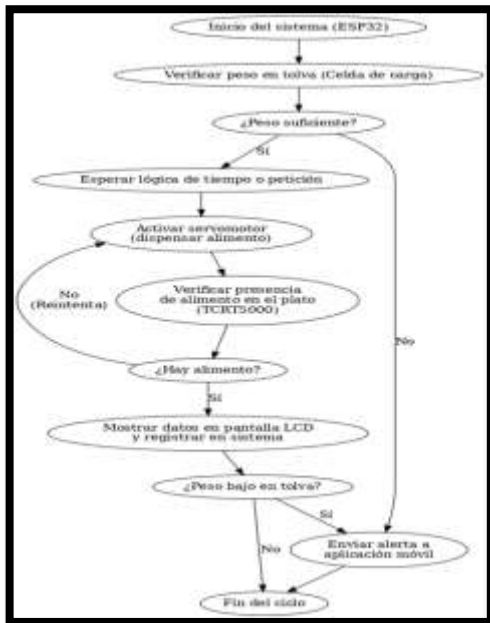
- Anoliefo, E. (2021). Diseño e Implementación de un Sistema de Alimentación Automatizado para Granjas Avícolas. *Proceedings of the International Conference on Industrial Engineering and Operations Management* , 1(1), 539.
- AVINEWS. (2024). Los nuevos diseños de platos para un rápido acceso del pollito al pienso y maximizar los aspectos higiénicos. *AVINEWS*, 1(2), 12-15.
- Benavides, S. (2022). Sectorización de la crianza de pollos de engorde en Ecuador. *Diseño de un sistema automatizado para la crianza y detección de enfermedades en pollos de engorde*, 24-26. Guayaquil, Ecuador: Escuela Superior Politécnica del Litoral.
- Bhongale, H. (2019). Agricultura de invernadero mediante IoT. *International Journal of Computer Trends and Technology (IJCTT)*, 69(9), 6.
- Bohorquez, Y. (2024). Marco teórico. *Diseño de un Avisistema viable para la producción de pollos de engorde con enfoque en la automatización para el mejoramiento del bienestar de las aves en Restrepo Meta*, 22. Villavicencio, Colombia: Universidad Cooperativa de Colombia.
- Bootwong, A. (2025). Evaluación e implementación de un sistema de alimentación automática para la granja de pollos de engorde. *Brazilian Journal of Poultry Science*, 27(3), 10.
- Broncano, K. (2022). Comederos automatizados para pollos productividad. *Evaluación productiva del engorde de pollos utilizando comederos manuales y automáticos*, 7-11. Lima, Perú: Universidad Nacional Agraria la Molina.
- Castillo, J. (2022). Introducción y objetivos. *Identificación y cuantificación de parásitos internos en aves domésticas en el cantón Guaranda*, 12-16. Guaranda, Ecuador: Universidad Estatal de Bolívar.
- Constitución de la República del Ecuador. (2008). Art. 281. *Constitución de la República del Ecuador*. Quito, Ecuador.
- Diaz, J. (2021). Conclusión. *Desarrollo de un sistema automatizado de bajo costo basado en lenguaje micropython para el suministro de alimento y bebida para el galpón de la granja avícola ubicada en calle 110 # 38-700 en la ciudad de Barranquilla.*, 85. Barranquilla, Colombia: Universidad Autónoma del Caribe.

- Escudero, B. (2023). Selección de los sensores y módulos electrónicos. *Diseño y prototipado de sistemas para teleconsulta usando tarjetas de desarrollo de bajo costo con fotoplestimografía*, 31-35. Guayaquil, Ecuador: Escuela Superior Politécnica del Litoral.
- Fernandez, F. (2025). Control de servomotor con el microcontrolador empotrado MicroBlaze MCS. *Impacto Revista de Ciencia y Tecnología*, 5(1), 27.
- Gomez, S. (2022). Sistema IoT basado en ESP32 para el control y monitoreo de cultivos en invernadero con enfoque de Agricultura 4.0. *Facultad de Ingeniería*, 4(1), 2.
- Guerra, A. (2021). Aplicaciones del control automático. "Implementación de un sistema automatizado de un comedero y bebedero de pollos para beneficio de micro-empresas y emprendedores de bajos recursos económicos en la parroquia Santa Rosa", 21-31. Latacunga, Ecuador: Universidad de las Fuerzas Armadas.
- Guerra, J. (2019). Antecedentes. *Implementación de un comedero automatizado para aves productoras de carne en la etapa de cría y engorde en la finca experimental "La María"*, 17-18. Riobamba, Ecuador: Escuela Superior Politécnica de Chimborazo.
- Hernández, J. (2021). Desarrollo de un sistema de monitoreo de invernaderos de agricultura inteligente utilizando dispositivos IoT de bajo costo. *Atlantis Press*, 5(1), 126.
- Kumar, A. (2023). Sistema inteligente de alimentación avícola mediante IoT y monitorización de celdas de carga. *Journal of Information Technology and Digital World*, 13(5), 134.
- Ley de propiedad intelectual. (2014). Art. 8. *Ley de la propiedad intelectual*. Quito, Ecuador.
- Lumban, G. (2024). Desarrollo de un sistema de alimentación programado y eficiente basado en IoT para pollos de engorde. *CEPAT Journal of Computer Engineering: Progress, Application, and Technology*, 3(1), 34.
- Martinez, L. (2023). Internet de las cosas aplicado a la agricultura: estado actual y su aplicación mediante un prototipo. *Revista Ibérica de Sistemas y Tecnologías de Información*, 53(106), 22.
- Novoa, F. (2020). Diseño y simulación de un sistema automatizado para producción avícola en la región del Guavio. *INVENTUM*, 28(1), 12-13.

- Ortega, J. (2022). Diseño de una arquitectura IoT de bajo coste para la monitorización de invernaderos. *Computación y Sistemas UNAM*, 26(1), 221.
- Park, J. (2023). Aplicación del Internet de las Cosas (IoT) para sistemas optimizados de alimentación del ganado. *AgriEngineering*, 3(4), 843.
- Pérez, E. (2023). Introducción. *Diseño de un Sistema de adquisición y procesamiento de señales provenientes de una celda de carga*, 3. La Habana, Cuba: Universidad Tecnológica de La Habana José Antonio Echeverría.
- Roa, E. (2018). Desarrollo de un dispensador automatizado de maíz en avicultura de traspatio. *Revista Tecnoacademia*, 1(2), 7.
- Sistema Integrado de Estadísticas Agrarias Peru. (2022). *Producción y Comercialización de productos avícolas*. Resumen Ejecutivo. <https://www.gob.pe/institucion/midagri/informes-publicaciones/5414945-boletin-estadistico-mensual-del-sector-avicola-2024>
- Sortica, F. (s.f.). Fundamento teórico. *Sistema de monitoreo automatizado de variantes presentes en la alimentación avícola*. Porto Alegre, Brasil: Universidad Estatal de Rio.
- Subedi, K. (2024). Sistema de gestión avícola inteligente basado en IoT. *Journal of Smart Machines and Computing (IRO Journals)*, 6(1), 27.
- Ulcuango, H. (2024). Desarrollo de un dispositivo de medición de la radiación solar con monitoreo IoT basado en Ubidots. *Revista Social Fronteriza*, 4(5), 22.
- Vishay Intertechnology. (2019). Aplicación de sensores ópticos réflex. *Vishay Semiconductors*, 1(1), 6.

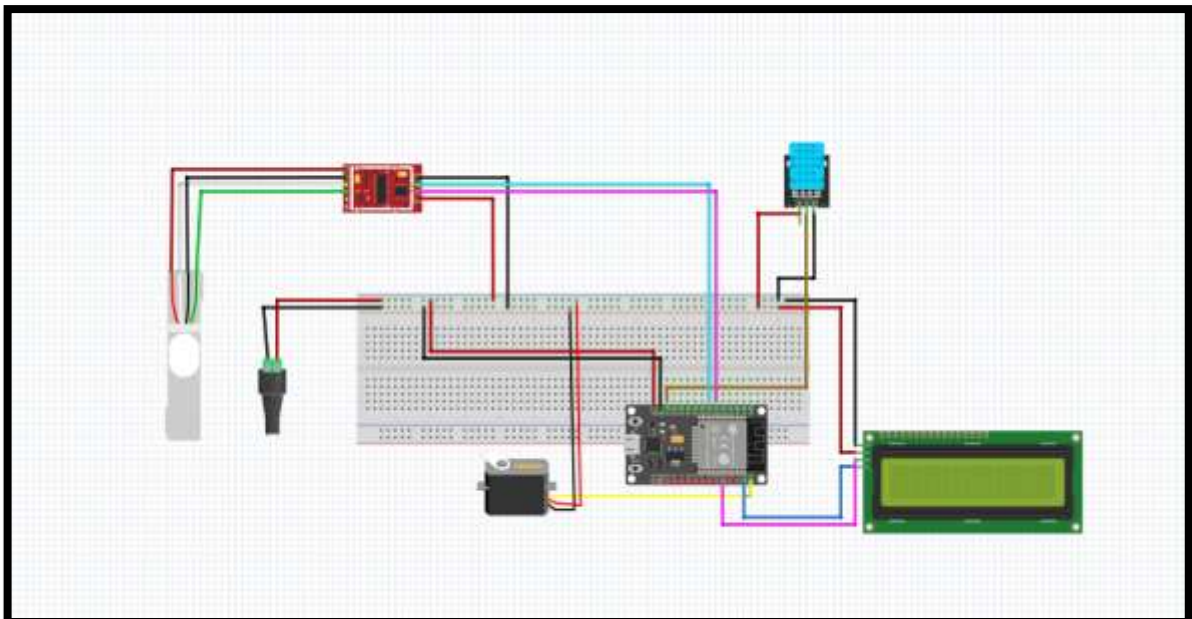
ANEXOS

Figura 1.

Diagrama de Flujo


Elaborado por: El Autor, 2026

Figura 2.

Esquema electrónico del proyecto en fritzing

Elaborado por: El Autor, 2026

Figura 5.
Encuesta Realizada a los trabajadores


UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
"DR. JACOBO BUCARAMORTIZ"
CARRERA COMPUTACIÓN

Objetivo: El objetivo de la encuesta dirigida a los trabajadores de la Hacienda "Castillo" es obtener datos relevantes sobre las prácticas diarias de alimentación de los pollos, el tiempo y esfuerzo requeridos, y la percepción del personal sobre el posible uso de un sistema automatizado. Esta información permitirá identificar necesidades operativas, evaluar la aceptación del proyecto por parte de los usuarios directos y adaptar el diseño del sistema para facilitar su integración en las rutinas laborales.

1. ¿Cuál es su rol principal en la finca?

Alimentar a los pollos
 Limpiar las instalaciones
 Supervisar el galpón
 Otro: _____

2. ¿Cómo realiza actualmente la alimentación de los pollos?

Manual con balde o recipientes
 Por tolvas simples
 Otro: _____

3. ¿Con qué frecuencia alimenta a los pollos por día?

1 vez
 2 veces
 3 veces
 Más de 3 veces

4. ¿Cuánto tiempo aproximadamente le toma completar la alimentación diaria?

Menos de 30 minutos
 30-60 minutos
 Más de 1 hora

5. ¿Ha notado que se desperdicia alimento durante el proceso?

Sí
 No
 A veces

6. ¿Sabe cuánto alimento consume cada grupo de pollos por día?

Sí
 No
 Aproximadamente

7. ¿Le resultaría útil un sistema que arries cuando el alimento se está por terminar?

Sí
 No
 No lo sé

8. ¿Cree que un sistema automático de alimentación facilitaría su trabajo?

Sí
 No
 Depende del sistema

9. ¿Se sentiría cómodo aprendiendo a usar un sistema automatizado con sensores?

Sí
 No

10. ¿Le gustaría participar en pruebas del nuevo sistema cuando esté instalado?

Sí
 No

Elaborado por: El Autor, 2026

Figura 6.
Visita del tutor para supervisar el proyecto

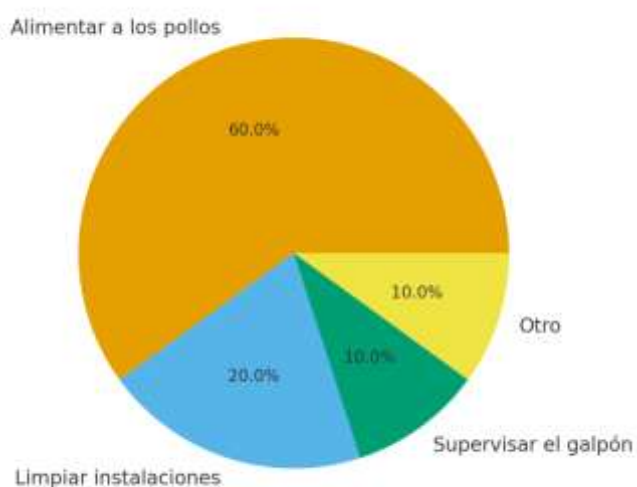


Elaborado por: El Autor, 2026

Pregunta 1. ¿Cuál es su rol principal en la finca?

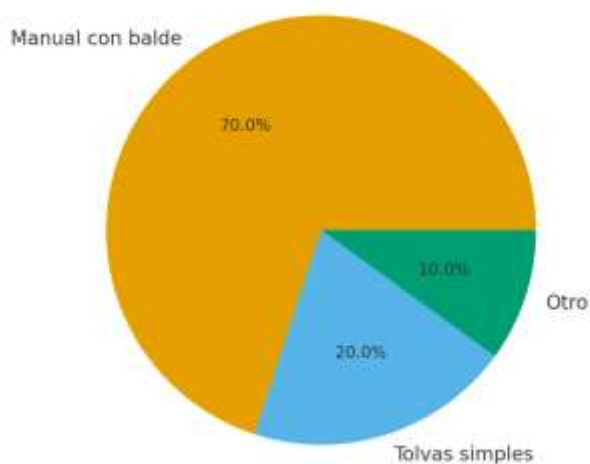
Descripción	Cantidad	Porcentaje
Alimentar a los pollos	6	60%
Limpiar las instalaciones	2	20%
Supervisar el galpón	1	10%
Otro	1	10%
Total	10	100%

Pregunta 1: Rol principal en la finca

**Pregunta 2. ¿Cómo realiza actualmente la alimentación de los pollos?**

Descripción	Cantidad	Porcentaje
Manual con balde o recipiente	7	70%
Por tolvas simples	2	20%
Otro	1	10%
Total	10	100%

Pregunta 2: Método actual de alimentación



Pregunta 3. ¿Con qué frecuencia alimenta a los pollos por día?

Descripción	Cantidad	Porcentaje
1 vez	1	10%
2 veces	5	50%
3 veces	3	30%
Más de 3 veces	1	10%
Total	10	100%

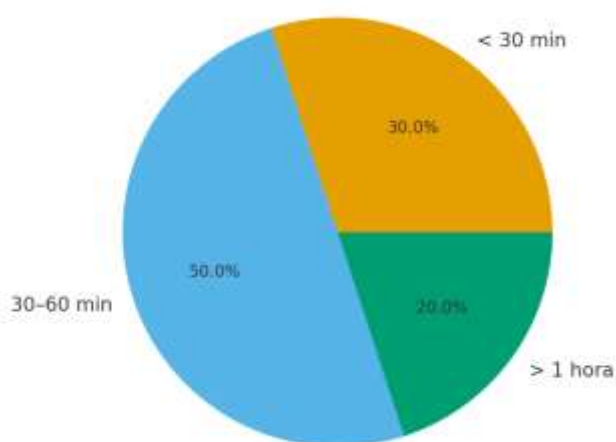
Pregunta 3: Frecuencia de alimentación



Pregunta 4. ¿Cuánto tiempo le toma completar la alimentación diaria?

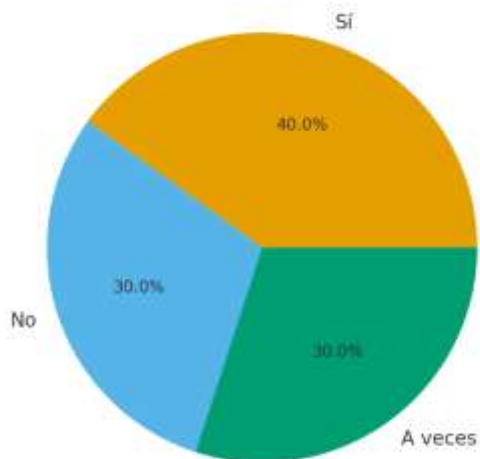
Descripción	Cantidad	Porcentaje
Menos de 30 minutos	3	30%
30–60 minutos	5	50%
Más de 1 hora	2	20%
Total	10	100%

Pregunta 4: Tiempo diario de alimentación

**Pregunta 5. ¿Ha notado desperdicio de alimento durante el proceso?**

Descripción	Cantidad	Porcentaje
Sí	4	40%
No	3	30%
A veces	3	30%
Total	10	100%

Pregunta 5: ¿Hay desperdicio de alimento?



Pregunta 6. ¿Sabe cuánto alimento consume cada grupo por día?

Descripción	Cantidad	Porcentaje
Sí	4	40%
No	3	30%
Aproximadamente 3		30%
Total	10	100%

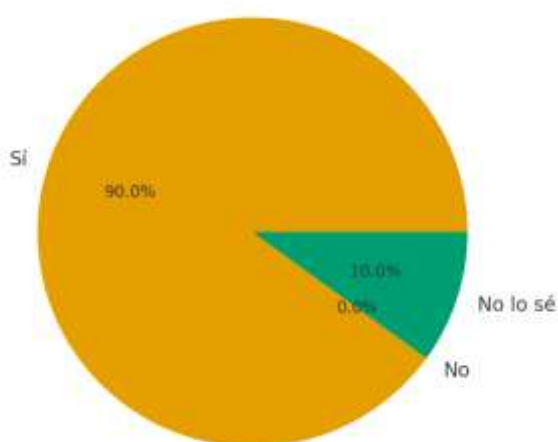
Pregunta 6: ¿Conoce el consumo por grupo?



Pregunta 7. ¿Le resultaría útil un sistema que avise cuando el alimento se está por terminar?

Descripción	Cantidad	Porcentaje
Sí	9	90%
No	0	0%
No lo sé	1	10%
Total	10	100%

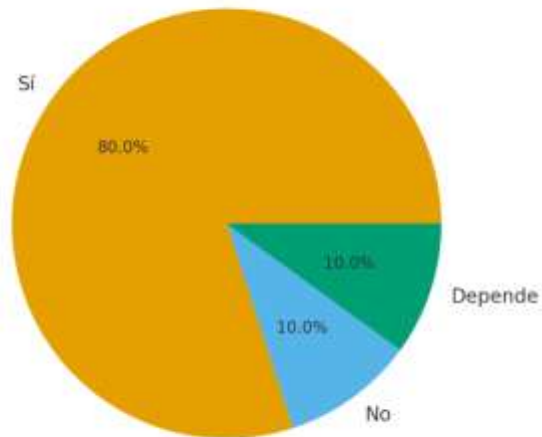
Pregunta 7: ¿Sería útil aviso de poco alimento?



Pregunta 8. ¿Un sistema automático facilitaría su trabajo?

Descripción	Cantidad	Porcentaje
Sí	8	80%
No	1	10%
Depende del sistema	1	10%
Total	10	100%

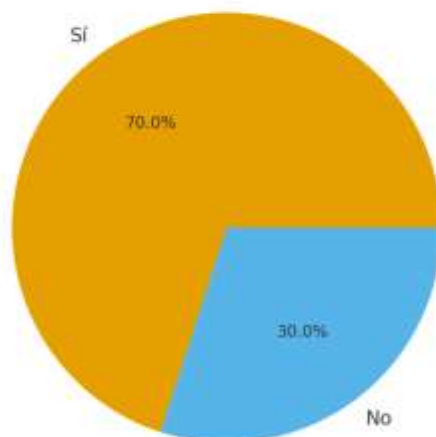
Pregunta 8: ¿Facilitaría su trabajo un sistema automático?



Pregunta 9. ¿Se sentiría cómodo aprendiendo a usar un sistema automatizado?

Descripción	Cantidad	Porcentaje
Sí	7	70%
No	3	30%
Total	10	100%

Pregunta 9: ¿Se sentiría cómodo aprendiendo a usarlo?



Pregunta 10. ¿Le gustaría participar en las pruebas del sistema?

Descripción	Cantidad	Porcentaje
Sí	8	80%
No	2	20%
Total	10	100%

Pregunta 10: ¿Participaría en las pruebas?

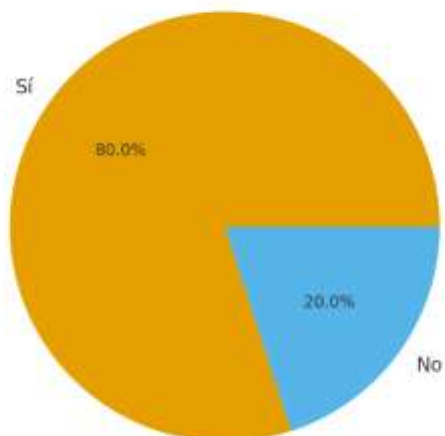


Tabla 1.

Elementos electrónicos para el dispensador automatizado

CONCEPTO	CANTIDAD	P. U	TOTAL
Microcontrolador ESP32	1	\$25,00	\$25,00
Sensor DHT11	1	\$5,00	\$5,00
Sensor TCRT5000	3	\$3,00	\$9,00
Celda de carga + HX711	1	\$15,00	\$15,00
Servomotor SG995	1	\$20,00	\$20,00
Pantalla LCD 16x2	1	\$10,00	\$10,00
Jumpers (paquete)	2	\$2,00	\$4,00
Protoboard	1	\$6,00	\$6,00
Fuente de alimentación	1	\$25,00	\$25,00
Caja para montaje	1	\$30,00	\$30,00
Total			\$149,00

Elaborado por: El Autor, 2026

Tabla 2.

Elementos para verificación y supervisión del sistema

CONCEPTO	CANTIDAD	P. U	TOTAL
Computadora portátil	1	\$600,00	\$600,00
Router WiFi	1	\$20,00	\$20,00
Multímetro digital	1	\$25,00	\$25,00
Cables y conectores	1	\$8,00	\$8,00
Total			\$653,00

Elaborado por: El Autor, 2026

Tabla 3.

Consumo semanal alimento para pollos

Día de la semana	Cantidad de pollos	Consumo diario medido (g)	Consumo promedio por pollo (g/día)
Lunes	15	1 742	116,1
Martes	15	1 815	121,0
Miércoles	15	1 689	112,6
Viernes	15	1 793	119,5
Sábado	15	1 856	123,7
Domingo	15	1 721	114,7
Total semanal	_____	12 480 g	_____
Promedio diario	15	1 783 g	118,8 g

Elaborado por: El Autor, 2026

Tabla 4.

Resumen diario del consumo de alimento

Fecha	Consumo diario (g)
25/01/2026	1744
26/01/2026	1996
27/01/2026	1701

Fecha	Consumo diario (g)
28/01/2026	1960
29/01/2026	1918
30/01/2026	1768
31/01/2026	1998
TOTAL	13085

Elaborado por: El Autor, 2026

Tabla 5.

Registro de dispensación de alimento

Fecha	Hora	Cantidad dispensada (g)
25/01/2026	07:30	690
25/01/2026	12:30	456
25/01/2026	18:30	598
26/01/2026	07:30	720
26/01/2026	12:30	586
26/01/2026	18:30	690
27/01/2026	07:30	624
27/01/2026	12:30	479
27/01/2026	18:30	598
28/01/2026	07:30	699
28/01/2026	12:30	606
28/01/2026	18:30	655
29/01/2026	07:30	639
29/01/2026	12:30	617
29/01/2026	18:30	662
30/01/2026	07:30	621
30/01/2026	12:30	558
30/01/2026	18:30	589

Fecha	Hora	Cantidad dispensada (g)
31/01/2026	07:30	644
31/01/2026	12:30	666
31/01/2026	18:30	688

Elaborado por: El Autor, 2026

Acta de entrega y cierre del sistema

Fecha: 19/05/2026

Lugar: Santa Rosa de Flandes, Cantón Naranjal, Provincia del Guayas, Ecuador.

PARTES INVOLUCRADAS

Cliente/Organización: Carlos Castillo

Desarrollador: Eddy Romero

OBJETIVO DEL ACTA

El presente documento formaliza la entrega y cierre del sistema denominado: "**DISPENSADOR AUTOMATIZADO PARA ALIMENTO DE POLLOS EN LA HACIENDA "CASTILLO" DEL CANTÓN NARANJAL, PROVINCIA DEL GUAYAS**", desarrollado como parte del proyecto de titulación e investigación académica.

DESCRIPCIÓN DEL SISTEMA ENTREGADO

Nombre del sistema: Dispensador automatizado para alimento de pollos "Chicken Feeder".

Objetivo principal: Implementar un prototipo automatizado mediante el uso de sensores y actuadores conectados a un microcontrolador para optimizar la distribución de alimentos, reducir el desperdicio de insumos y mejorar la eficiencia en la alimentación avícola dentro de la Hacienda "Castillo".

Características principales

- **Monitoreo de peso en tiempo real:** Medición continua del alimento restante disponible en la tolva mediante una celda de carga y el módulo de conversión HX711.
- **Monitoreo de variables ambientales:** Medición de la temperatura y humedad relativa dentro del galpón mediante el sensor

DHT11/DHT22 para la prevención de complicaciones digestivas en las aves.

- **Detección de alimento en el plato:** Verificación óptica de la presencia o ausencia de pienso en el recipiente a través del sensor infrarrojo reflectivo TCRT5000.
- **Dosificación y control automático:** Apertura y cierre preciso de la compuerta de la tolva mediante un servomotor para la administración de porciones adecuadas sin fallos mecánicos.
- **Visualización local de datos:** Presentación en sitio y en tiempo real del peso del alimento, la temperatura y la humedad mediante una pantalla LCD 16x2.
- **Sincronización inalámbrica en la nube:** Conexión a redes Wi-Fi mediante el microcontrolador para el envío continuo de telemetría y la recepción de comandos.
- **Mecanismo de alerta temprana:** Activación automática de una bandera de alerta de bajo nivel en la base de datos cuando el peso del alimento en la tolva desciende a un umbral crítico igual o inferior a 1200 gramos.
- **Acceso y gestión remota:** Visualización de KPIs del estado en tiempo real, registro de programaciones horarias y envío de comandos operativos (open, close, dispense) mediante una interfaz o aplicación web (index.html).

MÓDULOS ENTREGADOS

- **Módulo de procesamiento y control central:** Gestión autónoma de la lógica de dosificación y comunicación inalámbrica integrada mediante el microcontrolador ESP32.
- **Módulo de pesaje:** Medición calibrada y filtrado por promedio de la cantidad de alimento en la tolva a través de la celda de carga y el módulo HX711.
- **Módulo de dosificación mecánica:** Control de posicionamiento angular de la compuerta del dispensador a través de un servomotor.

- **Módulo de sensores:** Captura de condiciones del entorno con el sensor DHT y detección de reflexión de luz infrarroja en el plato con el TCRT5000.
- **Módulo de interfaz local:** Despliegue visual de datos operativos en tiempo real para el encargado de la finca mediante pantalla LCD 16x2.
- **Módulo de base de datos:** Almacenamiento estructurado de telemetría, estados de la compuerta y registros de eventos en Firebase Realtime Database.
- **Módulo de aplicación web:** Interfaz de usuario (login.html y dashboard index.html) para supervisión a distancia, control manual de compuertas y descarga de informes de consumo acumulado de los últimos 7 días en formato Excel.

DOCUMENTACIÓN INCLUIDA

Se entrega la siguiente documentación como parte del proyecto:

- Manual de usuario.
- Manual técnico del sistema.
- Diagramas electrónicos y esquemas de conexiones de los componentes en Fritzing .
- Código fuente desarrollado para el microcontrolador ESP32 en Arduino IDE .
- Estructura JSON y configuración de reglas de la base de datos implementada en Firebase .

VALIDACIÓN Y PRUEBAS

- El cliente ha revisado y aprobado el funcionamiento del sistema, incluyendo las siguientes pruebas en el entorno real:
- Pruebas de pesaje continuo de la tolva y filtrado de lecturas mediante algoritmo de promedio simple.
- Verificación de mediciones de temperatura y humedad ambiental dentro de los rangos del galpón.
- Pruebas de detección del flujo y presencia de alimento en el plato mediante el sensor óptico TCRT5000.

- Validación de la apertura y cierre de la compuerta variando los ángulos del servomotor.
- Pruebas de visualización correcta de caracteres y datos en la pantalla LCD 16x2.
- Pruebas de conectividad Wi-Fi, autenticación con Firebase Auth y envío de telemetría en tiempo real a la base de datos.
- Validación del sistema de notificación de nivel bajo al igualar o descender del umbral de 1200g.
- Pruebas de recepción y ejecución de comandos remotos (open, close, dispense) enviados desde la aplicación web.

ENTREGABLES ADICIONALES

- Capacitación básica a los trabajadores y personal operativo de la Hacienda “Castillo” para la supervisión del sistema y reabastecimiento de la tolva.
- Configuración inicial del proyecto en Firebase Console (Authentication y Realtime Database).
- Instalación física y puesta en funcionamiento del prototipo en el galpón avícola en un espacio destinado de 4m x 3m.

DECLARACIÓN DE CONFORMIDAD

Las partes involucradas certifican que el sistema entregado cumple con los objetivos establecidos en el proyecto de investigación y con los requerimientos definidos para la optimización de la distribución de alimento en el galpón avícola de la Hacienda “Castillo”. Asimismo, se deja constancia de que el prototipo fue sometido a un periodo de pruebas de siete días consecutivos operando de forma continua y estable en condiciones ambientales reales de la zona, verificándose una correcta dosificación autónoma, reducción del desperdicio de alimento, almacenamiento de datos en Firebase y despliegue local de la información. Por lo tanto, el cliente recibe a conformidad el sistema descrito en la presente acta.

FIRMAS DE LAS PARTES**Por el Cliente****Nombre:** Carlos Castillo**Cargo:** Propietario / Representante de la Hacienda "Castillo"**Firma:** _____**Por el Desarrollador****Nombre:** Eddy Romero**Cargo:** Investigador y Desarrollador del Sistema**Firma:** _____

APÉNDICE

Apéndice N° 1: MANUAL TÉCNICO

MANUAL TECNICO

**DISPENSADOR AUTOMATIZADO
PARA ALIMENTO DE POLLOS**



Descripción general del sistema

El sistema está compuesto por tres componentes principales: una base de datos en la nube (Firebase Realtime Database) que almacena telemetría y recibe comandos; un firmware en ESP32 que mide sensores (celda de carga HX711, DHT22, TCRT5000), controla la compuerta mediante servomotor y envía/recibe datos desde Firebase; una aplicación web (login.html + index.html) que permite visualizar el estado en tiempo real, registrar programaciones y enviar comandos al dispensador.

Requisitos del sistema

Requisitos de hardware

Microcontrolador ESP32 (con Wi-Fi integrado).

- Voltaje de operación: 3.3 V
- Conectividad: Wi-Fi y Bluetooth
- Frecuencia: hasta 240 MHz

Servomotor (conexión a pin PWM del ESP32).

- Voltaje de operación: 4.8 – 6 V
- Ángulo de giro: 0° – 180°
- Control por señal PWM

Celda de carga + módulo HX711 (DT y SCK).

- Capacidad típica: 5 – 20 kg (según modelo)
- Conversión: 24 bits ADC
- Alimentación: 2.6 – 5.5 V

Sensor de temperatura y humedad DHT22 (o DHT11 si se adapta el código).

- Rango temp.: -40 a 80 °C (DHT22)
- Precisión: ± 0.5 °C
- Alimentación: 3.3 – 5 V

Sensor óptico TCRT5000 (entrada digital/analógica según conexión).

- Tipo: Infrarrojo reflectivo
- Distancia detección: 2 – 15 mm
- Voltaje: 3.3 – 5 V

Pantalla LCD 16x2 con módulo I2C (dirección típica 0x27).

- Visualización: 16 columnas × 2 filas

- Interfaz: I2C
- Voltaje: 5 V

Fuente de alimentación estable.

- Salida: 5 V regulados
- Corriente recomendada: ≥ 2 A
- GND común para todos los módulos

Requisitos para uso en PC (Computadora).

- Computadora de escritorio o portátil
- Procesador mínimo: Intel Core i3 o equivalente
- Memoria RAM: 4 GB mínimo
- Espacio en disco: 500 MB libres
- Puerto USB (para programación y pruebas del ESP32)
- Conexión a Internet (Wi-Fi o Ethernet)
- Sistema operativo:
- Windows 10 o superior
- (Opcional: Linux o macOS para desarrollo)
- Navegador web actualizado:
- Google Chrome, Mozilla Firefox o Microsoft Edge

Requisitos de red

- Router Wi-Fi funcional
- Red local con acceso a Internet
- Permitir comunicación entre el ESP32 y la aplicación web/base de datos

Requisitos para uso en Android (Teléfono o Tablet)

- Smartphone o tablet Android
- Memoria RAM: 2 GB mínimo
- Espacio libre: 100 MB
- Conectividad Wi-Fi o datos móviles

Sistema operativo:

- Android 8.0 (Oreo) o superior
- Navegador web actualizado o aplicación móvil compatible

Requisitos de software

- Arduino IDE (o PlatformIO) con soporte para ESP32.

- Navegador web moderno (Chrome/Edge/Firefox) para ejecutar login.html e index.html.
- Cuenta de Google/Firebase para crear el proyecto en Firebase Console.

Configuración de Firebase

1. Iniciar Sesión firebase
2. Crear un proyecto en Firebase Console.
3. Habilitar Realtime Database y seleccionar ubicación (region).
4. Configurar reglas de seguridad (para pruebas se puede usar modo de prueba; para producción se recomienda autenticación).
5. Habilitar Authentication → método "Email/Password" y crear un usuario (correo y contraseña) que usará el ESP32.
6. En Project settings (Configuración del proyecto) copiar: API Key y Database URL. Estos valores se colocan en el firmware del ESP32 y en el index.html.

Estructura de la base de datos (Realtime Database)

El firmware y la web usan rutas (paths) específicas dentro del árbol JSON. Se recomienda mantener la siguiente estructura:

```
{
  "telemetry": {
    "peso_g": 1780,
    "#": "peso actual del alimento medido por la celda de carga",
    "temp_c": 28.5,
    "#": "temperatura ambiente medida por el sensor DHT",
    "hum_pct": 72,
    "#": "humedad relativa del ambiente",
    "tcrt_detecta": true,
    "#": "indica si el sensor TCRT5000 detecta alimento en el plato",
    "compuerta_abierta": false,
    "#": "estado actual del servomotor (true = abierta)",
    "dispensando": false,
    "#": "indica si el sistema está dispensando alimento",
    "objetivo_g": 1800
  }
}
```

```

    "#": "cantidad objetivo en gramos a dispensar"
  },
  "control": {
    "cmd": "idle",
    "#": "comando enviado desde la web: open, close, dispense o idle",
    "target_g": 0
    "#": "gramos objetivo cuando el comando es dispense"
  },
  "programaciones": {
    "-Nv123abc": {
      "hora": "07:30",
      "#": "hora programada de alimentación",
      "pollos": 15,
      "#": "cantidad de pollos registrados",
      "fecha": 1738459200000
      "#": "fecha de creación en formato timestamp"
    }
  }
}
}
}

```

Tabla 6.**Base de datos Firebase**

Ruta	Tipo	Descripción
/telemetry/peso_g	float	Peso (g) leído de la celda de carga (HX711).
/telemetry/temp_c	float	Temperatura en °C (DHT22).
/telemetry/hum_pct	float	Humedad relativa en % (DHT22).
/telemetry/tcrt_detecta	bool	True si el sensor TCRT detecta presencia/flujo.
/telemetry/compuerta_abierta	bool	Estado actual de la compuerta (servo).

/telemetry/dispensando	bool	True mientras el ESP32 dispensa por gramos.
/telemetry/objetivo_g	float	Objetivo en gramos para la orden "dispense".
/control/cmd	string	Comando enviado desde la web: open, close, dispense, idle.
/control/target_g	float	Cantidad objetivo (g) usada cuando cmd = dispense.

Elaborado por: El Autor, 2026

Además, el index.html guardara programaciones (horarios y número de pollos) en una colección en Firebase y genera estadísticas de consumo para los últimos 7 días (gráficas y exportación a Excel). La ruta exacta depende de la sección "Registrar programación" y de las funciones de acumulación de consumo definidas en el script.

Firmware ESP32 (Arduino)

Este firmware realiza: conexión Wi-Fi, autenticación con Firebase, lectura periódica de sensores, control de compuerta por servo, envío de telemetría y recepción de comandos. A continuación, se describen los bloques más importantes y qué parámetros se modifican.

Librerías incluidas

Bloque de inclusión de librerías para Wi-Fi, Firebase (librería Mobizt), servo, I2C/LCD, DHT y HX711. Si se reemplaza algún sensor, aquí es donde normalmente se cambia la librería (por ejemplo, DHT11 vs DHT22).

```
#include <WiFi.h>
// # Permite la conexión del ESP32 a una red WiFi
#include <Firebase_ESP_Client.h>
// # Librería principal para comunicación con Firebase
```

```

#include "addons/TokenHelper.h"
// # Manejo automático del token de autenticación Firebase
#include <HX711.h>
// # Lectura de la celda de carga para medir peso del alimento
#include <DHTesp.h>
// # Lectura del sensor DHT (temperatura y humedad)
#include <Wire.h>
// # Comunicación I2C para la pantalla LCD
#include <LiquidCrystal_I2C.h>
// # Control de la pantalla LCD 16x2
#include <ESP32Servo.h>
// # Control del servomotor que abre y cierra la compuerta

```

Credenciales Wi-Fi y Firebase

En este bloque se definen SSID y contraseña del Wi-Fi, y las credenciales de Firebase: API_KEY, DATABASE_URL, USER_EMAIL y USER_PASSWORD. Si cambia de red o de proyecto Firebase, estos son los valores que se deben actualizar.

```

#define WIFI_SSID "NOMBRE_RED"
// # Nombre de la red WiFi a la que se conecta el ESP32
#define WIFI_PASSWORD "CLAVE_RED"
// # Contraseña de la red WiFi
#define API_KEY "API_KEY_FIREBASE"
// # Clave del proyecto Firebase
#define DATABASE_URL "https://xxxx-default-rtdb.firebaseio.com"
// # URL de la base de datos Realtime Database
#define USER_EMAIL "usuario@email.com"
// # Usuario creado en Firebase Authentication
#define USER_PASSWORD "password"
// # Contraseña del usuario Firebase

```

Asignación de pines

Define los pines del ESP32 donde están conectados servo, DHT, TCRT y HX711. Si se cambia el cableado, se actualizan aquí (PIN_SERVO, PIN_DHT, PIN_TCRT, HX_DT, HX_SCK).

```
#define PIN_SERVO 13
// # Pin PWM que controla el servomotor de la compuerta
#define PIN_DHT 27
// # Pin digital del sensor de temperatura y humedad
#define PIN_TCRT 26
// # Pin del sensor óptico TCRT5000
#define HX_DT 19
// # Pin DT del módulo HX711
#define HX_SCK 23
// # Pin SCK del módulo HX711
```

Parámetros de configuración

Incluye: tipo de sensor DHT (DHTTYPE), ángulos del servo (SERVO_CERRADO y SERVO_ABIERTO) y el factor de calibración del HX711 (HX_FACTOR). • Para ajustar el recorrido del servo: modifique SERVO_CERRADO/SERVO_ABIERTO según el mecanismo.

- Para recalibrar la celda de carga: modifique HX_FACTOR con el valor obtenido en calibración.

- Si la dirección I2C del LCD cambia: modifique lcd(0x27, 16, 2) por 0x3F u otra.

```
#define SERVO_CERRADO 20
// # Ángulo del servomotor cuando la compuerta está cerrada
#define SERVO_ABIERTO 90
// # Ángulo del servomotor cuando la compuerta está abierta
#define LCD_ADDRESS 0x27
// # Dirección I2C de la pantalla LCD
#define LCD_COLUMNS 16
// # Número de columnas de la LCD
#define LCD_ROWS 2
```

```
// # Número de filas de la LCD
```

Objetos globales y variables de estado

Se instancian los objetos: LCD, Servo, DHT, HX711 y los objetos de Firebase (FirebaseData, FirebaseAuth, FirebaseConfig). Luego se declaran variables de telemetría (peso_g, temp_c, hum_pct, tcrt_detecta) y temporizadores (tWeight, tDHT, etc.) que implementan lectura y envío por “periodos” sin bloquear el loop.

```
LiquidCrystal_I2C lcd(LCD_ADDRESS, LCD_COLUMNS, LCD_ROWS);
```

```
// # Inicializa la pantalla LCD
Servo servo;
// # Objeto para controlar el servomotor
HX711 scale;
// # Objeto para la celda de carga
DHTesp dht;
// # Objeto para el sensor DHT
float peso_g = 0;
// # Variable que almacena el peso del alimento
float temp_c = 0;
// # Variable de temperatura
float hum_pct = 0;
// # Variable de humedad
bool tcrt_detecta = false;
// # Estado del sensor TCRT5000
unsigned long tPeso = 0;
unsigned long tDHT = 0;
unsigned long tLCD = 0;
unsigned long tFirebase = 0;
// # Temporizadores para ejecutar tareas sin delay()
#define PERIOD_PESO 2000
// # Intervalo de lectura de peso (ms)
#define PERIOD_DHT 5000
// # Intervalo de lectura del DHT
#define PERIOD_LCD 1000
```

```
// # Intervalo de actualización de la pantalla LCD
#define PERIOD_FIREBASE 3000
// # Intervalo de envío de datos a Firebase
```

Funciones de control de compuerta

abrirCompuerta() y cerrarCompuerta() escriben el ángulo del servo y actualizan compuerta_abierta. Si el servo abre/cierra al revés, invierta los valores SERVO_CERRADO/SERVO_ABIERTO.

```
void abrirCompuerta() {
  servo.write(SERVO_ABIERTO);
  // # Abre la compuerta del dispensador
}
void cerrarCompuerta() {
  servo.write(SERVO_CERRADO);
  // # Cierra la compuerta del dispensador
}
```

Lectura de peso con filtrado

leerPesoFiltrado() implementa un promedio simple: toma 5 lecturas y devuelve el promedio. Para hacer el sistema más rápido: reduzca el número de muestras (por ejemplo 3). Para suavizar más: aumente las muestras, considerando que tardará más.

```
float leerPesoFiltrado() {
  float suma = 0;
  // # Variable para acumular lecturas
  for (int i = 0; i < 5; i++) {
    suma += scale.get_units(1);
    // # Lee el peso desde la celda de carga
    delay(50);
    // # Pequeño retardo para estabilidad
  }
  return suma / 5.0;
```

```
// # Retorna el promedio del peso leído
}
```

Envío de telemetría a Firebase

firebaseWriteAll() envía los valores actuales a rutas dentro de /telemetry usando setFloat y setBool. Si se desea agregar un nuevo sensor, se agrega aquí una nueva escritura con su ruta (por ejemplo /telemetry/luz_lux).

```
void enviarTelemetria() {
  Firebase.RTDB.setFloat(&fbdo, "/telemetry/peso_g", peso_g);
  // # Envía el peso actual a Firebase
  Firebase.RTDB.setFloat(&fbdo, "/telemetry/temp_c", temp_c);
  // # Envía la temperatura
  Firebase.RTDB.setFloat(&fbdo, "/telemetry/hum_pct", hum_pct);
  // # Envía la humedad
  Firebase.RTDB.setBool(&fbdo, "/telemetry/tcrt_detecta", tcrt_detecta);
  // # Envía el estado del sensor óptico
}
```

Recepción de comandos desde Firebase

handleFirebaseCommand() lee /control/cmd y, según el valor, ejecuta:

- open: abre compuerta.
- close: cierra compuerta.
- dispense: lee /control/target_g, inicia dispensado por gramos y registra

/telemetry/base_g.

Al finalizar cada comando, escribe "idle" para evitar re-ejecuciones.

```
void leerComandoFirebase() {
  if (Firebase.RTDB.getString(&fbdo, "/control/cmd")) {
    // # Lee el comando enviado desde la aplicación web
    String cmd = fbdo.stringData();
    if (cmd == "open") {
      abrirCompuerta();
      // # Abre la compuerta desde la web
    }
  }
}
```

```

if (cmd == "close") {
  cerrarCompuerta();
  // # Cierra la compuerta desde la web
}
Firebase.RTDB.setString(&fbdo, "/control/cmd", "idle");
// # Reinicia el comando para evitar repeticiones
}
}

```

setup(): inicialización y conexión

En setup() se inicializan: Serial, LCD (Wire.begin con pines I2C), servo (attach), pines del TCRT, DHT, HX711 (tare), y luego se conecta al Wi-Fi. Si se conecta correctamente, se ejecuta setupFirebase() y se inicializan variables en RTDB (/control/cmd = idle, /control/target_g = 200). El timeout de conexión Wi-Fi es 15 s; puede aumentarse si el entorno tiene señal débil.

```

// # Objetos Firebase (deben existir globalmente)
FirebaseData fbdo;
// # Canal de comunicación con Firebase RTDB
FirebaseAuth auth;
// # Autenticación Firebase (usuario/clave)
FirebaseConfig config;
// # Configuración Firebase (API key + URL)
// # Función para iniciar Firebase
void setupFirebase() {
  config.api_key = API_KEY;
  // # Define la API Key del proyecto Firebase
  config.database_url = DATABASE_URL;
  // # Define la URL de Realtime Database
  auth.user.email = USER_EMAIL;
  // # Usuario de Firebase Authentication
  auth.user.password = USER_PASSWORD;
  // # Contraseña del usuario Firebase
  Firebase.begin(&config, &auth);

```

```
// # Inicia la comunicación con Firebase
Firebase.reconnectWiFi(true);
// # Permite reconectar automáticamente si se cae el WiFi
}
void setup() {
  Serial.begin(115200);
  // # Habilita monitor serial para depuración
  pinMode(PIN_TCRT, INPUT);
  // # Configura el pin del TCRT5000 como entrada
  servo.attach(PIN_SERVO);
  // # Conecta el servomotor al pin configurado
  cerrarCompuerta();
  // # Asegura que el sistema inicia con la compuerta cerrada
  lcd.init();
  // # Inicializa la pantalla LCD
  lcd.backlight();
  // # Enciende la luz de la pantalla LCD
  dht.setup(PIN_DHT, DHTesp::DHT22);
  // # Inicializa el sensor DHT (cambiar a DHT11 si corresponde)
  scale.begin(HX_DT, HX_SCK);
  // # Inicializa el HX711 con sus pines
  scale.tare();
  // # Realiza tara (pone peso inicial a cero)
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  // # Inicia conexión WiFi
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    // # Espera hasta conectar al WiFi
  }
  setupFirebase();
  // # Inicia Firebase con credenciales
  Firebase.RTDB.setString(&fbdo, "/control/cmd", "idle");
  // # Inicializa el comando en estado inactivo
  Firebase.RTDB.setFloat(&fbdo, "/control/target_g", 0);
```

```
// # Inicializa el objetivo de dispensación en 0 g
}
```

loop(): tareas periódicas

El loop() no usa delay (excepto en test serial), sino periodos con millis():

- Lee peso cada PERIOD_WEIGHT.
- Lee DHT cada PERIOD_DHT.
- Actualiza tcrt_detecta continuamente.
- Gestiona dispensado local por gramos (sin depender de internet).
- Actualiza LCD cada PERIOD_LCD.
- Envía telemetría cada PERIOD_FB.
- Lee comandos cada PERIOD_CMD.

Modificar periodos permite equilibrar precisión y consumo de recursos.

```
void loop() {
  // # Lectura del sensor TCRT5000 (presencia de alimento)
  tcrt_detecta = digitalRead(PIN_TCRT);
  // # Guarda si el TCRT detecta alimento en el plato
  // # Leer peso cada cierto periodo
  if (millis() - tPeso >= PERIOD_PESO) {
    peso_g = leerPesoFiltrado();
    // # Actualiza la variable peso_g usando promedio
    tPeso = millis();
    // # Reinicia el temporizador de peso
  }
  // # Leer temperatura y humedad cada cierto periodo
  if (millis() - tDHT >= PERIOD_DHT) {
    TempAndHumidity data = dht.getTempAndHumidity();
    // # Obtiene temperatura y humedad del sensor
    temp_c = data.temperature;
    // # Guarda temperatura
    hum_pct = data.humidity;
    // # Guarda humedad
    tDHT = millis();
  }
}
```

```
// # Reinicia temporizador del DHT
}
// # Actualizar LCD cada cierto periodo
if (millis() - tLCD >= PERIOD_LCD) {
  lcd.clear();
  // # Limpia pantalla
  lcd.setCursor(0, 0);
  lcd.print("Peso:");
  lcd.print((int)peso_g);
  lcd.print("g");
  // # Muestra el peso de alimento
  lcd.setCursor(0, 1);
  lcd.print("T:");
  lcd.print((int)temp_c);
  lcd.print("C H:");
  lcd.print((int)hum_pct);
  lcd.print("%");
  // # Muestra temperatura y humedad
  tLCD = millis();
  // # Reinicia temporizador de LCD
}
// # Enviar telemetría a Firebase cada cierto periodo
if (millis() - tFirebase >= PERIOD_FIREBASE) {
  enviarTelemetria();
  // # Envía valores actuales a /telemetry
  leerComandoFirebase();
  // # Revisa y ejecuta comandos en /control
  tFirebase = millis();
  // # Reinicia temporizador de Firebase
}
}
```

Sistema de notificación por nivel bajo de alimento

El dispensador automatizado incorpora un mecanismo de alerta temprana que permite notificar al responsable de la finca cuando el nivel de alimento en la tolva desciende por debajo de un umbral crítico preestablecido.

Este sistema funciona a partir de la celda de carga, la cual mide de forma continua el peso del alimento disponible en la tolva. El microcontrolador ESP32 procesa esta información y la compara con un valor de referencia configurado en el sistema. Para el presente proyecto, el umbral mínimo de alimento se estableció en 1200 gramos.

Cuando el peso del alimento es igual o inferior a 1200 g, el sistema ejecuta automáticamente las siguientes acciones:

- Genera una alerta interna en el microcontrolador.
- Envía una notificación a la aplicación de mensajería del dispositivo móvil del encargado.

La notificación enviada tiene como finalidad alertar oportunamente al usuario para que proceda a rellenar el dispensador, evitando interrupciones en la alimentación de los pollos y garantizando la continuidad del proceso productivo.

Este mecanismo de notificación contribuye a:

- Reducir el riesgo de desabastecimiento de alimento.
- Optimizar la supervisión del sistema sin necesidad de inspección manual constante.
- Mejorar la eficiencia operativa y la gestión del tiempo del personal.

El valor del umbral puede ser modificado mediante la programación del microcontrolador, permitiendo adaptar el sistema a distintas capacidades de tolva o requerimientos productivos.

```
#define UMBRAL_MINIMO 1200
// # Umbral mínimo de alimento en gramos (modificable)
void verificarNivelBajo() {
  if (peso_g <= UMBRAL_MINIMO) {
    // # Si el alimento está por debajo del umbral
    Firebase.RTDB.setBool(&fbdo, "/telemetry/alerta_bajo_nivel", true);
```

```

// # Activa bandera de alerta en Firebase para notificar al usuario
} else {
  Firebase.RTDB.setBool(&fbdo, "/telemetry/alerta_bajo_nivel", false);
  // # Apaga la alerta si el nivel vuelve a ser suficiente
}
}
}

```

Aplicación web – login.html

login.html es una pantalla sencilla de autenticación local (solo para demostración). No valida contra Firebase: compara contra credenciales fijas (admin / 1234) y guarda un “flag” en sessionStorage.

Estructura HTML y estilos

Define el diseño de la tarjeta (card), logo y campos de entrada. El archivo requiere que chicken.png esté en la misma carpeta para mostrar el icono.

```

<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>AquaMonitor – Iniciar sesión</title>
  <!-- # Ícono del sitio (opcional) -->
  <link rel="icon" href="./AquaMonitor_logo.ico">
  <style>
    /* # Estilos generales de la pantalla de login */
    body{
      margin:0;
      background:#0f172a;
      color:#e5e7eb;
      font-family:system-ui;
      display:flex;
      align-items:center;
      justify-content:center;
      height:100vh;
    }
  </style>

```

```
padding:18px;
}
/* # Recuadro principal (contenedor del formulario) */
.card{
width:min(420px, 100%);
background:#111827;
border-radius:16px;
padding:40px 56px;
text-align:center;
box-shadow:0 10px 24px rgba(0,0,0,.25);
}
/* # Logo superior */
.logo{
width:72px;
height:auto;
display:block;
margin:0 auto 16px;
}
/* # Campos de entrada */
input{
width:100%;
padding:12px;
margin:8px 0;
border-radius:10px;
border:1px solid #1f2937;
background:#0f172a;
color:#e5e7eb;
outline:none;
}
/* # Botón de acceso */
button{
width:100%;
padding:12px;
margin-top:12px;
```

```

border-radius:10px;
border:none;
background:#22c55e;
font-weight:700;
cursor:pointer;
}
/* # Mensaje de error/estado */
#msg{
margin-top:10px;
font-size:13px;
color:#ef4444;
min-height:18px;
}
</style>
</head>
<body>
<div class="card">
<!-- # Logo de la aplicación (debe estar en la misma carpeta) -->

<h2>Login</h2>
<!-- # Campo usuario -->
<input type="text" id="user" placeholder="Usuario" autocomplete="username">
<!-- # Campo contraseña -->
<input type="password" id="pass" placeholder="Contraseña"
autocomplete="current-password">
<!-- # Botón para validar credenciales -->
<button id="btnLogin">Entrar</button>
<!-- # Zona para mostrar mensajes -->
<div id="msg"></div>
</div>
<!-- # Script de validación (se completa en la Figura 27) -->
<script>

```

Bloque de autenticación (JavaScript)

Variables principales:

- USER_OK y PASS_OK: credenciales permitidas.
- KEY: clave usada en sessionStorage ("demo_auth_ok").

Flujo:

- 1) Si ya existe sessionStorage[KEY] = "1", redirige a index.html.
- 2) Al presionar "Entrar", compara usuario y contraseña.
- 3) Si son correctos, guarda sessionStorage[KEY] = "1" y redirige.
- 4) Si son incorrectos, muestra mensaje de error.

Qué modificar:

- Cambiar usuario/clave: edite USER_OK y PASS_OK.
- Hacer login "real" con Firebase Auth: se debe reemplazar este bloque por signInWithEmailAndPassword() y reglas seguras en RTDB.

```
const USER_OK = "admin";
// # Usuario permitido (credencial fija)
const PASS_OK = "1234";
// # Contraseña permitida (credencial fija)
const KEY = "demo_auth_ok";
// # Clave usada en sessionStorage para indicar sesión iniciada
// # Si ya existe sesión, redirige al panel principal
if (sessionStorage.getItem(KEY) === "1") {
  window.location.href = "index.html";
}
// # Evento click del botón "Entrar"
document.getElementById("btnLogin").addEventListener("click", () => {
  const u = document.getElementById("user").value.trim();
  // # Captura el usuario ingresado
  const p = document.getElementById("pass").value.trim();
  // # Captura la contraseña ingresada
  // # Verifica credenciales
  if (u === USER_OK && p === PASS_OK) {
    sessionStorage.setItem(KEY, "1");
    // # Guarda bandera de sesión iniciada
```

```

    window.location.href = "index.html";
    // # Redirige al dashboard
  } else {
    document.getElementById("msg").textContent = "Usuario o contraseña
incorrectos";
    // # Muestra error si las credenciales no coinciden
  }
});
// # Permite iniciar sesión presionando Enter
document.addEventListener("keydown", (e) => {
  if (e.key === "Enter") {
    document.getElementById("btnLogin").click();
  }
});
</script>
</body>
</html>

```

Aplicación web – index.html

index.html es el dashboard principal. Cumple tres funciones: validar que el usuario inició sesión (guardia con sessionStorage), leer datos en tiempo real desde Firebase y mostrarlos, escribir datos/órdenes hacia Firebase (programaciones y comandos del dispensador).

Guardia de acceso (guard)

Al inicio se verifica sessionStorage[KEY]. Si no está en "1", redirige a login.html. Esto evita acceso accidental pero NO es seguridad real (cualquiera puede manipular sessionStorage).

```

<script>
  const KEY = "demo_auth_ok";
  // # Debe coincidir con la clave usada en login.html

  if (sessionStorage.getItem(KEY) !== "1") {
    // # Si no hay sesión iniciada, vuelve al login

```

```

    window.location.href = "login.html";
  }
</script>

```

Estructura visual: KPIs, programación y gráficas

El HTML define una sección “Estado en tiempo real” con 4 indicadores (peso, temperatura, humedad, nivel de tolva) y una barra de porcentaje. Luego incluye un formulario para registrar programación (cantidad de pollos y hora) y una tabla para mostrar programaciones guardadas. Finalmente se muestran gráficas de consumo (pastel y tendencia) y un botón para descargar un informe Excel.

```

<!-- # Sección: Estado en tiempo real -->
<section class="card">
  <h2>Estado en tiempo real</h2>
  <!-- # 4 recuadros KPI -->
  <div class="kpi-grid">
    <div class="kpi">
      <div class="kpi-title">Peso (g)</div>
      <div class="kpi-value" id="kpiPeso">0</div>
    </div>
    <div class="kpi">
      <div class="kpi-title">Temperatura (°C)</div>
      <div class="kpi-value" id="kpiTemp">0</div>
    </div>
    <div class="kpi">
      <div class="kpi-title">Humedad (%)</div>
      <div class="kpi-value" id="kpiHum">0</div>
    </div>
    <div class="kpi">
      <div class="kpi-title">Nivel tolva (%)</div>
      <div class="kpi-value" id="kpiNivel">0</div>
    </div>
  </div>
  <!-- # Barra visual del nivel -->

```

```

<div class="bar-wrap">
  <div class="bar" id="nivelBar"></div>
</div>
<!-- # Botones de control -->
<div class="row">
  <button id="btnOpen">Abrir compuerta</button>
  <button id="btnClose">Cerrar compuerta</button>
  <button id="btnDispense">Dispensar</button>
  <input id="targetG" type="number" value="200" min="1" style="width:120px">
</div>
</section>
<!-- # Sección: Registrar programación -->
<section class="card">
  <h2>Registrar programación</h2>
  <div class="row">
    <input id="inpPollos" type="number" placeholder="Cantidad de pollos" min="1">
    <input id="inpHora" type="time">
    <button id="btnGuardarProg">Guardar</button>
  </div>
</section>
<!-- # Sección: Programaciones guardadas -->
<section class="card">
  <h2>Programaciones guardadas</h2>
  <table id="tablaProg" border="1" cellpadding="8" cellspacing="0"
style="width:100%">
    <thead>
      <tr>
        <th>Horario</th>
        <th>Pollos</th>
        <th>Creado</th>
        <th>Acciones</th>
      </tr>
    </thead>
    <tbody id="tbodyProg"></tbody>

```

```

</table>
</section>
<!-- # Sección: Consumo últimos 7 días -->
<section class="card">
  <h2>Consumo – últimos 7 días</h2>
  <!-- # Gráfico pastel -->
  <canvas id="chartPie"></canvas>
  <!-- # Gráfico barras -->
  <canvas id="chartBars"></canvas>
  <!-- # Botón para descargar el informe -->
  <button id="btnExcel">Descargar informe (Excel .xlsx)</button>
</section>

```

Librerías externas (Chart.js y ExcelJS)

Se cargan librerías por CDN:

- Chart.js para gráficas.
- ExcelJS + FileSaver para exportar el informe en .xlsx.

Si la página se usa sin internet, estas librerías deben descargarse y referenciarse localmente.

```

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<!-- # Librería para generar gráficas -->
<script src="https://cdn.jsdelivr.net/npm/exceljs/dist/exceljs.min.js"></script>
<!-- # Librería para crear el archivo Excel -->
<script src="https://cdn.jsdelivr.net/npm/file-saver@2.0.5/dist/FileSaver.min.js"></script>
<!-- # Librería para descargar el archivo en el navegador -->

```

Conexión con Firebase (SDK Web)

En el `<script type="module">` se importan módulos de Firebase (`initializeApp`, `getDatabase`, `ref`, `onValue`, `set`, `push`, `remove`, `update`, etc.). Luego se define `firebaseConfig` (`apiKey`, `databaseURL`, `projectId`, `appId`, etc.) y se inicializa la conexión:

```
const app = initializeApp(firebaseConfig);
```

```
const db = getDatabase(app);
```

Qué modificar:

- Si se cambia de proyecto Firebase: actualice el objeto firebaseConfig con los datos del nuevo proyecto.

```
<script type="module">
```

```
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.22.0/firebase-app.js";
```

```
// # Inicializa Firebase App
```

```
import {
```

```
  getDatabase, ref, onValue, set, push, remove, update, get, child
```

```
} from "https://www.gstatic.com/firebasejs/9.22.0/firebase-database.js";
```

```
// # Funciones para leer/escribir en Realtime Database
```

```
const firebaseConfig = {
```

```
  apiKey: "API_KEY",
```

```
  // # Clave del proyecto Firebase (reemplazar)
```

```
  databaseURL: "https://xxxx-default-rtdb.firebaseio.com"
```

```
  // # URL de Realtime Database (reemplazar)
```

```
};
```

```
const app = initializeApp(firebaseConfig);
```

```
// # Crea la app Firebase
```

```
const db = getDatabase(app);
```

```
// # Conexión a Realtime Database
```

Lectura de telemetría en tiempo real

Se crea una referencia a /telemetry y se suscribe con onValue(). Cada vez que el ESP32 actualiza Firebase, la web recibe el snapshot y actualiza los KPIs (peso, temperatura, humedad, estado de compuerta) y la barra de nivel de tolva.

Qué modificar:

- Si cambian los nombres de variables en el ESP32 (por ejemplo /telemetry/peso), hay que actualizar los ids y las rutas aquí.

- El cálculo del porcentaje de tolva depende del "máximo" (ej. 2400 g). Para cambiar la capacidad, cambie tolvaMax en HTML/JS.

```

const telRef = ref(db, "telemetry");

// # Referencia al nodo /telemetry
const tolvaMax = 2400;
// # Capacidad máxima estimada de la tolva (g) para calcular porcentaje
onValue(telRef, (snapshot) => {
  const d = snapshot.val();
  // # Obtiene los datos actuales del nodo telemetry
  if (!d) return;
  document.getElementById("kpiPeso").textContent = d.peso_g ?? 0;
  // # Muestra el peso en el KPI
  document.getElementById("kpiTemp").textContent = d.temp_c ?? 0;
  // # Muestra la temperatura en el KPI
  document.getElementById("kpiHum").textContent = d.hum_pct ?? 0;
  // # Muestra la humedad en el KPI
  const nivel = Math.max(0, Math.min(100, ((d.peso_g ?? 0) / tolvaMax) * 100));
  // # Calcula el porcentaje de nivel de tolva
  document.getElementById("kpiNivel").textContent = nivel.toFixed(0);
  // # Muestra el porcentaje
  document.getElementById("nivelBar").style.width = nivel + "%";
  // # Ajusta visualmente la barra del nivel
});

```

Escritura de comandos (open/close/dispense)

Los botones del dashboard escriben en /control/cmd los valores "open", "close" o "dispense". Para dispense también se escribe /control/target_g con el objetivo en gramos. El ESP32 lee estos valores periódicamente (handleFirebaseCommand) y ejecuta la acción.

```

const cmdRef = ref(db, "control/cmd");
// # Ruta donde se envían los comandos al ESP32
const targetRef = ref(db, "control/target_g");
// # Ruta donde se envía la cantidad objetivo en gramos
document.getElementById("btnOpen").addEventListener("click", async () => {
  await set(cmdRef, "open");

```

```

    // # Envía el comando open para abrir la compuerta
  });
document.getElementById("btnClose").addEventListener("click", async () => {
  await set(cmdRef, "close");
  // # Envía el comando close para cerrar la compuerta
});
document.getElementById("btnDispense").addEventListener("click", async () => {
  const g = Number(document.getElementById("targetG").value || 0);
  // # Lee el objetivo en gramos desde el input
  await set(targetRef, g);
  // # Guarda los gramos objetivo
  await set(cmdRef, "dispense");
  // # Envía comando dispense para iniciar la dispensación
});

```

Programaciones guardadas (horarios)

La sección “Registrar programación” toma la hora y el número de pollos y guarda el registro en Firebase usando push(). Luego la tabla se actualiza leyendo esa colección (onValue + renderizado en HTML). También existe la opción de eliminar un registro (remove()).

Qué modificar:

- Si se desea guardar más campos (ej. ración por pollo): agregar un nuevo input en HTML y añadirlo al objeto que se guarda con push().
- Si se quiere impedir datos vacíos: agregar validaciones antes de guardar.

```

const progRef = ref(db, "programaciones");
// # Nodo donde se guardan las programaciones
document.getElementById("btnGuardarProg").addEventListener("click", async ()
=> {
  const pollos = Number(document.getElementById("inpPollos").value || 0);
  // # Obtiene cantidad de pollos
  const hora = document.getElementById("inpHora").value;
  // # Obtiene hora (formato HH:MM)
  if (!pollos || !hora) return;

```

```

// # Evita guardar si falta información
await push(progRef, {
  hora: hora,
  pollos: pollos,
  fecha: Date.now()
  // # Guarda fecha de creación como timestamp
});
});
onValue(progRef, (snapshot) => {
  const data = snapshot.val() || {};
  // # Obtiene todas las programaciones
  const tbody = document.getElementById("tbodyProg");
  tbody.innerHTML = "";
  Object.entries(data).forEach(([id, item]) => {
    const tr = document.createElement("tr");
    const tdHora = document.createElement("td");
    tdHora.textContent = item.hora ?? "";
    tr.appendChild(tdHora);
    const tdPollos = document.createElement("td");
    tdPollos.textContent = item.pollos ?? "";
    tr.appendChild(tdPollos);
    const tdFecha = document.createElement("td");
    tdFecha.textContent = new Date(item.fecha || Date.now()).toLocaleString();
    tr.appendChild(tdFecha);
    const tdAcc = document.createElement("td");
    const btnDel = document.createElement("button");
    btnDel.textContent = "Eliminar";
    btnDel.addEventListener("click", async () => {
      await remove(ref(db, "programaciones/" + id));
      // # Elimina la programación seleccionada
    });
    tdAcc.appendChild(btnDel);
    tr.appendChild(tdAcc);
    tbody.appendChild(tr);
  });
});

```

```
});
});
```

Módulo de consumo y exportación a Excel

El dashboard genera gráficas de consumo por día (últimos 7 días) y tendencia (barras + promedio móvil). Luego “Descargar informe” crea un archivo .xlsx con tablas y las gráficas incrustadas como imágenes base64, usando ExcelJS y FileSaver.

```
// # Datos de ejemplo (últimos 7 días)
const labels = ["Lun","Mar","Mié","Jue","Vie","Sáb","Dom"];
const consumo = [1742, 1815, 1689, 1793, 1856, 1721, 1783];
// # Gráfico pastel
const pie = new Chart(document.getElementById("chartPie"), {
  type: "pie",
  data: {
    labels: labels,
    datasets: [{ data: consumo }]
  }
});
// # Gráfico de barras
const bars = new Chart(document.getElementById("chartBars"), {
  type: "bar",
  data: {
    labels: labels,
    datasets: [{ label: "Consumo (g)", data: consumo }]
  }
});
// # Exportación a Excel
document.getElementById("btnExcel").addEventListener("click", async () => {
  const wb = new ExcelJS.Workbook();
  // # Crea un libro de Excel
  const ws = wb.addWorksheet("Consumo");
  // # Crea una hoja llamada Consumo
  ws.addRow(["Día", "Consumo (g)"]);
```

```
// # Encabezados de la tabla
labels.forEach((d, i) => ws.addRow([d, consumo[i]]));
// # Inserta los datos
const buffer = await wb.xlsx.writeBuffer();
// # Genera el archivo Excel en memoria
saveAs(new Blob([buffer]), "informe_consumo.xlsx");
// # Descarga el archivo al equipo del usuario
});
</script>
```

Publicación y hosteo de la aplicación web (Netlify)

La aplicación web del dispensador automatizado de alimento para pollos se encuentra alojada en un servicio de hosteo gratuito que permite la publicación de sitios web estáticos de forma segura y accesible desde cualquier navegador con conexión a Internet. Para el presente proyecto se utilizó la plataforma Netlify, debido a su facilidad de uso, estabilidad y compatibilidad con aplicaciones desarrolladas en HTML, CSS y JavaScript.

La aplicación web está compuesta por archivos estáticos (login.html, index.html, hojas de estilo y scripts JavaScript), los cuales son interpretados directamente por el navegador del usuario. La comunicación con el microcontrolador ESP32 y la base de datos se realiza mediante Firebase Realtime Database.

Creación del proyecto en Netlify

Para la publicación de la aplicación web se realizó el siguiente procedimiento:

- Creación de una cuenta en la plataforma Netlify mediante correo electrónico o cuenta GitHub.
- Inicio de sesión en el panel de administración de Netlify.
- Selección de la opción Add new site y posteriormente Deploy manually.
- Preparación de la carpeta del proyecto que contiene todos los archivos de la aplicación web.
- Carga de la carpeta completa en el área de despliegue de Netlify.

- Generación automática de una URL pública para el acceso al sistema.

Estructura de archivos publicada

La carpeta cargada en Netlify contiene los siguientes elementos principales:

- Archivo login.html para el acceso al sistema.
- Archivo index.html correspondiente al panel principal de control.
- Archivos CSS para el diseño visual del sistema.
- Archivos JavaScript para la comunicación con Firebase y la gestión de datos.

Acceso al sistema hospedado

Una vez finalizado el despliegue, el sistema puede ser accedido ingresando la URL proporcionada por Netlify en cualquier navegador web compatible. Desde esta dirección el usuario puede visualizar la información del dispensador en tiempo real, registrar programaciones de alimentación y generar reportes de consumo.

Actualización del sistema hospedado

En caso de realizar modificaciones en los archivos de la aplicación web, es necesario volver a cargar la carpeta actualizada en la plataforma Netlify para que los cambios se reflejen en el sitio publicado. Este procedimiento no afecta la configuración del microcontrolador ni la base de datos Firebase.

Parámetros que se cambian con más frecuencia

Elemento	Dónde	Motivo
Wi-Fi	WIFI_SSID, WIFI_PASSWORD	Cambiar red donde trabaja el ESP32.
Firebase firmware	API_KEY, DATABASE_URL, USER_EMAIL, USER_PASSWORD	Migrar a otro proyecto/usuario en Firebase.
Pines	PIN_SERVO, PIN_DHT, PIN_TCRT, HX_DT, HX_SCK	Adaptar a cableado real.
Servo	SERVO_CERRADO, SERVO_ABIERTO	Ajustar ángulos al mecanismo.
Balanza	HX_FACTOR	Calibración de la celda de carga.
Capacidad tolva (web)	tolvaMax (en HTML/JS)	Porcentaje de nivel y barra.
Login demo	USER_OK, PASS_OK	Cambiar credenciales de acceso local.

Solución de problemas (checklist rápido)

- No conecta a Wi-Fi: verificar SSID/contraseña, señal, y aumentar timeout de 15 s en setup().
- Firebase no recibe datos: revisar API_KEY y DATABASE_URL; confirmar que el usuario/contraseña de Firebase Auth son válidos y que Email/Password está habilitado.
- Lecturas de peso inestables: revisar cableado del HX711, GND común, y ajustar el filtrado o el factor HX_FACTOR.
- Servo no se mueve o vibra: usar fuente 5 V independiente para servo y compartir GND con ESP32; ajustar límites de attach (600–2300 μ s).
- Dashboard no muestra datos: confirmar que firebaseConfig es correcto y que las rutas /telemetry existen; revisar consola del navegador (F12).
- Login no funciona: verificar USER_OK y PASS_OK; borrar sessionStorage o usar “Nueva ventana”.

Apéndice N° 2: MANUAL DE USUARIO

MANUAL DE USUARIO

DISPENSADOR AUTOMATIZADO PARA ALIMENTO DE POLLOS



El presente Manual de Usuario describe el funcionamiento del sistema web desarrollado para el monitoreo y control del dispensador automatizado de alimento para pollos implementado en la Hacienda “Castillo”, ubicada en el cantón Naranjal, provincia del Guayas. El sistema fue diseñado con el objetivo de facilitar la supervisión del proceso de alimentación mediante una interfaz clara e intuitiva, permitiendo al usuario visualizar información en tiempo real, registrar programaciones de alimentación y consultar el consumo semanal del alimento.

Este manual está dirigido a los usuarios responsables del manejo y control del sistema, quienes no requieren conocimientos técnicos avanzados para su utilización.

Acceso al sistema (Pantalla de Login)

Al ingresar a la dirección web del sistema, el usuario visualizará la pantalla de inicio de sesión, la cual corresponde a la pantalla inicial del sistema.

En el centro de la pantalla se encuentra un recuadro de acceso, que contiene los siguientes elementos:

En la parte superior del recuadro:

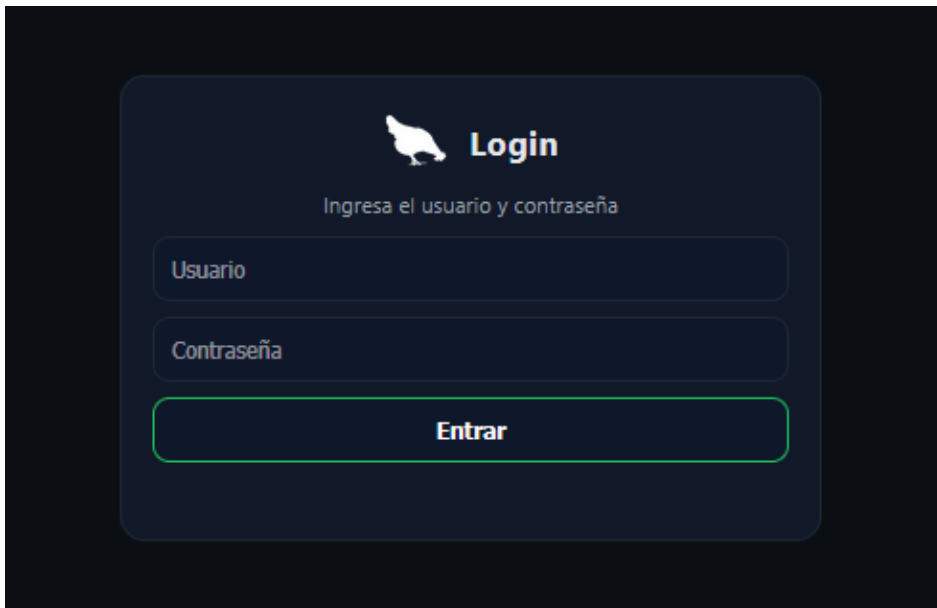
- Logotipo del sistema.
- Título “Login”.
- En la parte central del recuadro:
 - Campo Usuario, donde se ingresa el nombre del usuario.
 - Campo Contraseña, ubicado inmediatamente debajo del campo usuario.

En la parte inferior del recuadro:

Botón “Entrar”, utilizado para validar las credenciales.

Área de mensajes informativos, donde se muestran alertas en caso de error.

Pantalla de Login



Proceso de inicio de sesión

Para acceder al sistema, el usuario debe seguir los siguientes pasos:

- Ingresar el nombre de usuario en el campo Usuario, ubicado en la parte central del recuadro.
- Ingresar la contraseña en el campo Contraseña, situado debajo del campo usuario.
- Presionar el botón “Entrar”, ubicado en la parte inferior del recuadro.
- Si las credenciales son correctas, el sistema redirige automáticamente a la pantalla principal (Index).
- En caso de error, se mostrará un mensaje de advertencia debajo del botón de acceso.

Pantalla principal del sistema (Index)

Una vez iniciada la sesión correctamente, el usuario accede a la pantalla principal del sistema, donde se concentran todas las funciones y datos del dispensador automatizado.

El sistema fue diseñado bajo el criterio de pantalla única, lo que significa que toda la información relevante se encuentra distribuida verticalmente en una sola vista, evitando la navegación entre múltiples páginas y mejorando la comodidad del usuario.

Pantalla principal del sistema (Index).



Sección de estado en tiempo real

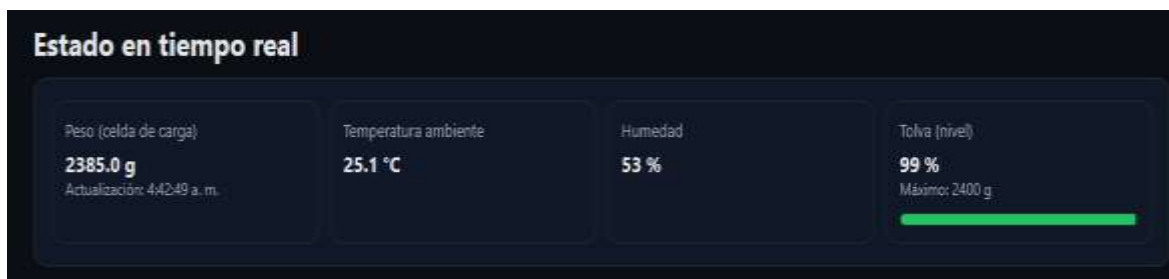
La sección de Estado en tiempo real se encuentra ubicada en la parte superior de la pantalla principal.

En esta sección se presentan cuatro recuadros informativos, alineados horizontalmente, los cuales muestran información clave del sistema:

- Primer recuadro (lado izquierdo):
Muestra el peso del alimento, medido por la celda de carga.
- Segundo recuadro (parte superior central izquierda):
Indica la temperatura ambiente registrada por el sensor.
- Tercer recuadro (parte superior central derecha):
Presenta el valor de humedad ambiental, expresado en porcentaje.
- Cuarto recuadro (lado derecho):
Muestra el nivel de la tolva, expresado en porcentaje, acompañado de una barra visual que indica la cantidad de alimento disponible.

Estos recuadros se actualizan automáticamente, permitiendo al usuario conocer el estado del sistema en tiempo real.

Sección de estado en tiempo real.



Registro de programación de alimentación

La sección “Registrar programación” se encuentra ubicada debajo de los recuadros de estado en tiempo real, en la parte media superior de la pantalla.

Ubicación y elementos

De izquierda a derecha, el usuario encontrará:

- Campo Cantidad de pollos, ubicado a la izquierda del recuadro.
- Campo Hora, ubicado en el centro.
- Botón “Guardar”, ubicado a la derecha.

Registro de programación de alimentación

Al presionar el botón “**Guardar**”, la programación se almacena automáticamente en el sistema.

Programaciones guardadas

La sección “Programaciones guardadas” se encuentra ubicada debajo del recuadro de registro de programación, ocupando la parte media de la pantalla principal.

Esta sección presenta un recuadro con una tabla, donde se muestran todas las programaciones registradas.

Descripción de la tabla

La tabla contiene las siguientes columnas:

- Horario: muestra la hora programada.

- Pollos: indica la cantidad registrada.
- Creado: muestra la fecha y hora de creación.
- Acciones: ubicada en el extremo derecho, contiene el botón Eliminar.

El botón Eliminar, ubicado en la parte derecha de cada fila, permite borrar una programación específica.

Programaciones guardadas

Programaciones guardadas			
Puedes eliminar una fila si ya no la quieres.			
Horario	Pollos	Creado	Acciones
07:30	15	2/2/2026, 5:19:54 a. m.	Eliminar
12:30	15	2/2/2026, 5:20:00 a. m.	Eliminar
18:30	15	2/2/2026, 5:20:17 a. m.	Eliminar

Sección de consumo de alimento

La sección “Consumo – últimos 7 días” se encuentra ubicada en la parte inferior de la pantalla principal.

Gráfico Pastel, Gráfico de barras y tendencia

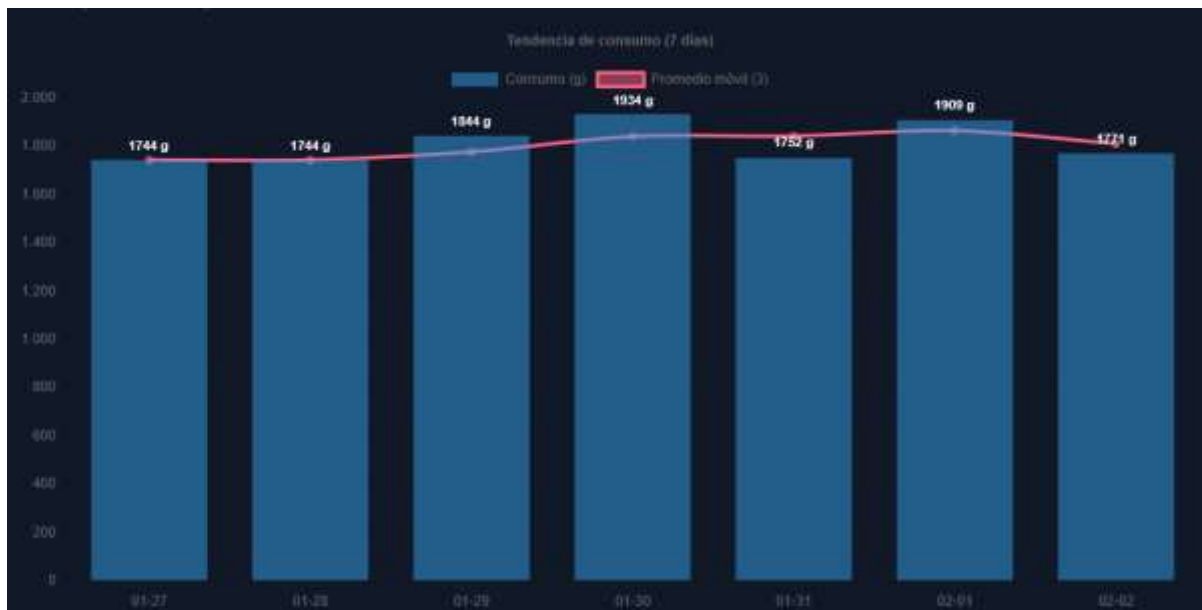
Ubicado en la parte inferior del apartado de programación de consumo guardado.

Muestra el consumo total de alimento distribuido por día de la semana.

Gráfico Pastel de consumo semanal



Gráfico barras de consumo semanal



Botón de descarga del informe

En la parte inferior del gráfico de barras, dentro del mismo recuadro de consumo, se encuentra el botón:

“Descargar informe (Excel .xlsx)”

Este botón permite generar y descargar un informe completo del consumo de alimento.

Generación de informes

Al presionar el botón “Descargar informe”, ubicado en la parte inferior del recuadro de consumo, el sistema genera automáticamente un archivo en formato Excel que contiene:

- Registro detallado de eventos de consumo.
- Resumen diario del consumo de alimento.
- Gráficos estadísticos generados automáticamente.
- El archivo se descarga directamente en el dispositivo del usuario.

Ventajas del diseño de pantalla única

El diseño del sistema, basado en una sola pantalla, ofrece las siguientes ventajas:

- Facilita la identificación de cada recuadro informativo.
- Reduce errores por navegación innecesaria.

- Permite visualizar la información de arriba hacia abajo de forma lógica.
- Mejora la experiencia del usuario final.

Registro detallado de eventos de consumo

	A	B	C	D	E
1	FECHA	HORA	GRAMOS		
2	25/01/2026	07:30	690,00		
3	25/01/2026	12:30	456,00		
4	25/01/2026	18:30	598,00		
5	26/01/2026	07:30	720,00		
6	26/01/2026	12:30	586,00		
7	26/01/2026	18:30	690,00		
8	27/01/2026	07:30	624,00		
9	27/01/2026	12:30	479,00		
10	27/01/2026	18:30	588,00		
11	28/01/2026	07:30	699,00		
12	28/01/2026	12:30	606,00		
13	28/01/2026	18:30	655,00		
14	29/01/2026	07:30	630,00		
15	29/01/2026	12:30	617,00		
16	29/01/2026	18:30	662,00		
17	30/01/2026	07:30	621,00		
18	30/01/2026	12:30	558,00		
19	30/01/2026	18:30	589,00		
20	31/01/2026	07:30	644,00		
21	31/01/2026	12:30	666,00		
22	31/01/2026	18:30	688,00		
23					
24					
25					
26					
27					

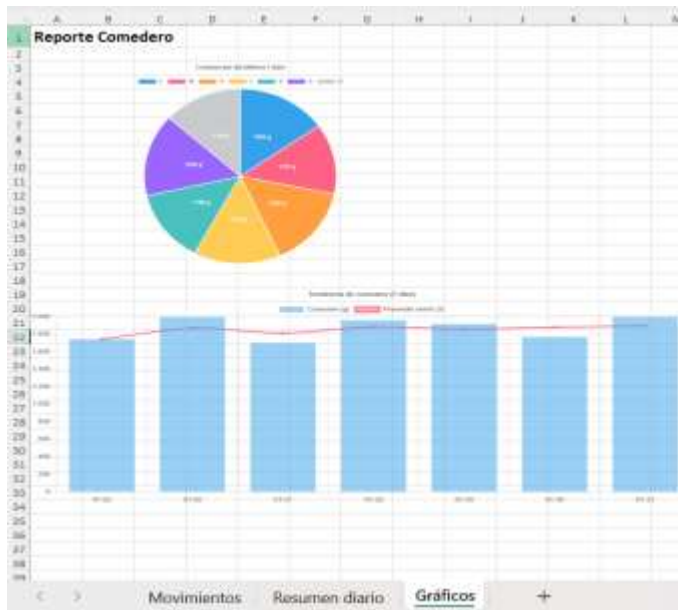
< > **Movimientos** Resumen diario Gráficos

Resumen diario del consumo de alimento

	A	B	C	D	E
1	FECHA	CONSUMO (g)			
2	25/01/2026	1744			
3	26/01/2026	1996			
4	27/01/2026	1701			
5	28/01/2026	1960			
6	29/01/2026	1918			
7	30/01/2026	1768			
8	31/01/2026	1998			
9					
10	TOTAL	13085			
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					

< > **Movimientos** **Resumen diario** Gráficos

Gráficos del consumo semanal en Excel



Consideraciones finales

El sistema web fue desarrollado para proporcionar una herramienta eficiente, clara y accesible para el control del dispensador automatizado de alimento para pollos.

La organización por recuadros y la centralización de funciones en una sola pantalla permiten al usuario identificar rápidamente cada opción, comprender su función y operar el sistema de manera eficiente.