



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE

TELECOMUNICACIONES

**“DESARROLLO DE UN SISTEMA IOT DE CONTROL Y MONITOREO DE
INVENTARIO EN TIEMPO REAL BASADO EN SISTEMAS EMBEBIDOS DE
CÓDIGO ABIERTO”**

Trabajo de titulación previo a la obtención del

Título de ingeniero en telecomunicaciones

AUTORES: HENRRY ALEJANDRO MURILLO AVECILLAS

JOHN ANTHONY BRAVO MAIGUA

TUTOR: Ing. HOLGER SANTILLÁN C. Msc

Guayaquil – Ecuador

2025

I. **Certificado de responsabilidad y autoría del trabajo de titulación**

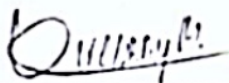
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Henry Alejandro Murillo Avecillas con documento de identificación N° 0923829717, y John Anthony Bravo Maigua con documento de identificación N° 0953306008 manifestamos que:

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

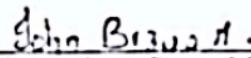
Guayaquil, 20 de agosto del año 2025.

Atentamente,



Henry Alejandro Murillo Avecillas

0923829717



John Anthony Bravo Maigua

0953306008

II. Certificado de cesión de derechos de autor del trabajo de titulación a la Universidad Politécnica Salesiana.

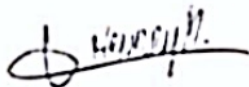
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA.

Nosotros, Henry Alejandro Murillo Avecillas, con C.I. 0923829717 y John Anthony Bravo Maigua, con C.I. 0953306008, expresamos nuestra voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana del Ecuador la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Científico: "DESARROLLO DE UN SISTEMA IOT DE CONTROL Y MONITOREO DE INVENTARIO EN TIEMPO REAL BASADO EN SISTEMAS EMBEBIDOS DE CÓDIGO ABIERTO", el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Institución facultada para ejercer plenamente los derechos concedidos.

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

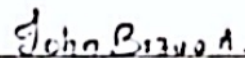
Guayaquil, 20 de agosto del año 2025.

Atentamente,



Henry Alejandro Murillo Avecillas

0923829717



John Anthony Bravo Maigua

0953306008

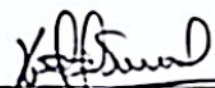
II. Certificado de Dirección del Trabajo de Titulación.

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.

Yo, Holger Jorge Santillán Carranza, con documento de identificación N° 0916303134 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN SISTEMA IOT DE CONTROL Y MONITOREO DE INVENTARIO EN TIEMPO REAL BASADO EN SISTEMAS EMBEBIDOS DE CÓDIGO ABIERTO**, realizado por Henry Alejandro Murillo AVECILLAS con documento de identificación N° 0923829717, y John Anthony Bravo Maigua con documento de identificación N° 0953306008, obteniendo como resultado final el trabajo de titulación bajo la opción de Artículo académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 20 de agosto del año 2025.

Atentamente,



Holger Jorge Santillán Carranza
0916303134

IV. Dedicatoria

Quiero dedicar este artículo, en primer lugar, a mis padres. Ellos han sido mi mayor fuente de fortaleza y motivación a lo largo de todo este camino. Su apoyo constante, su confianza en mí y sus palabras de aliento en los momentos más difíciles me han dado el valor para seguir adelante, incluso cuando todo parecía cuesta arriba. Me han enseñado a creer en mis capacidades, a no rendirme y, sobre todo, a no conformarme nunca con menos de lo que soy capaz de alcanzar. Gracias por ser ejemplo de esfuerzo, paciencia y amor incondicional.

También quiero expresar mi más profundo agradecimiento a mis profesores. Su dedicación, compromiso y pasión por la enseñanza han marcado una gran diferencia en mi vida. Más allá del conocimiento académico, me han brindado herramientas, valores y perspectivas que me acompañarán siempre. Cada clase, consejo y gesto de apoyo ha sido fundamental en mi crecimiento tanto intelectual como personal. Esta dedicatoria es una pequeña forma de reconocer la gran huella que han dejado en mí.

V. Agradecimiento

Quiero expresar mi más sincero agradecimiento a mi familia, quienes me han guiado con sabiduría en los momentos en que he cometido errores y han sido un constante aliento para que no me detenga en el camino hacia mis metas. En especial, agradezco profundamente a mis padres, quienes jamás dejaron de impulsarme y creer en mí, incluso en los momentos más difíciles.

También extendiendo mi gratitud a mi tutor y a todos mis profesores, cuyas enseñanzas y acompañamiento fueron fundamentales para que pudiera desarrollar este proyecto de la mejor manera posible.

A mis seres queridos y a mis padres, gracias por su apoyo incondicional y su paciencia a lo largo de todo este proceso académico. Su confianza en mí ha sido una fuente inagotable de motivación. A mi tutor y docentes, gracias por su compromiso, orientación y dedicación. A todos ustedes, les dedico este logro con profunda gratitud y humildad.

VI. Resumen

El presente proyecto tuvo como finalidad desarrollar un sistema IoT de control y monitoreo de inventario en tiempo real mediante sensores de peso, utilizando un Arduino UNO R4 WiFi y una Raspberry Pi 3, integrados a la plataforma en la nube TagoIO. El objetivo principal fue automatizar el proceso de gestión de inventario en pequeños negocios, transformando el inventario físico en un sistema digital eficiente y de bajo costo. La metodología aplicada fue de tipo descriptiva con enfoque mixto, basada en el desarrollo e integración de hardware y software para la toma, procesamiento y visualización de datos. Se utilizaron cinco sensores de peso tipo celda de carga con módulos HX711, conectados al Arduino, que enviaron las lecturas hacia la Raspberry Pi, y posteriormente al dashboard en TagoIO, donde se obtuvieron resultados de la media aritmética de 1.349 segundos en cuanto a tiempo de respuesta, y una tasa de error porcentual del 3,86% indicando que es baja. Entre las conclusiones se determinó que el sistema permite un monitoreo preciso, visualización remota y reducción de errores humanos. Se recomienda escalar el número de sensores, proteger los componentes y optimizar el código para notificaciones automáticas.

Palabras clave: IoT, Control de inventario, Monitoreo en tiempo real, sensores de peso, TagoIO

VII. Abstract

The present project aimed to develop an IoT system for real-time inventory control and monitoring using weight sensors, an Arduino UNO R4 WiFi, and a Raspberry Pi 3, integrated with the TagoIO cloud platform. The main objective was to automate the inventory management process in small businesses by transforming physical inventory into an efficient and low-cost digital system. The applied methodology was descriptive with a mixed approach, based on the development and integration of hardware and software for data collection, processing, and visualization. Five load cell weight sensors with HX711 modules were used, connected to the Arduino, which sent the readings to the Raspberry Pi and then to the TagoIO dashboard. The results obtained showed an arithmetic mean response time of 1.349 seconds and a percentage error rate of 3.86%, indicating a low margin of error. Among the conclusions, it was determined that the system allows for accurate monitoring, remote visualization, and reduction of human errors. It is recommended to scale the number of sensors, protect the components, and optimize the code for automatic notifications.

Keywords: IoT, Inventory Control, Real-time Monitoring, Weight Sensors, TagoIO

VIII. Índice de Contenido

I.	Certificado de responsabilidad y autoría del trabajo de titulación	1
II.	Certificado de cesión de derechos de autor del trabajo de titulación a la Universidad Politécnica Salesiana.....	2
III.	Certificado de Dirección del Trabajo de Titulación.	3
IV.	Dedicatoria	4
V.	Agradecimiento	4
VI.	Resumen.....	5
VII.	Abstract.....	6
VIII.	Índice de Contenido	7
IX.	Introducción	9
9.1.	Arduino UNO	10
9.2.	Sensores de peso con HX711.....	11
9.3.	Lector RFID RC522	11
9.4.	ESP32	12
9.5.	Raspberry Pi.....	13
9.6.	Arduino IDE.....	13
9.7.	Python.....	14
9.8.	TagoIO.....	14
9.9.	Dashboard web.....	14
9.10.	Ecuaciones a utilizar	14
X.	Artículos Relacionados	16
XI.	Metodología	18
11.1.	Diseño de Estudio	18
11.1.1.	Tipo de Investigación	18
11.1.2.	Justificación del Diseño Elegido	18
11.1.3.	Hipótesis.....	20
11.1.4.	Diagrama, esquema del prototipo	20
11.2.	Materiales y Herramientas	21
11.3.	Procedimiento	24
11.3.1.	Diagrama de flujo: Pasos para realizar el prototipo.....	24
11.4.	Desarrollo del prototipo (Descripción de fases).....	24
11.4.1.	Procedimiento para iniciar Arduino R4 Wi-Fi	24
11.4.2.	Calibración del sensor de peso	26
11.4.3.	Prueba del sensor de peso.....	27

11.4.4.	Prueba de pantalla LCD I2 20x4	28
11.4.5.	Integración del módulo de expansión V5.0 para facilitar conexiones.....	29
11.4.6.	Desarrollo del Programa	30
11.4.7.	Configuración de Raspberry Pi3.....	31
11.4.8.	Creación de Dashboard en Tago IO	35
11.4.9.	Creación de bot de Telegram.....	38
11.4.10.	Proceso de obtención de los datos de inventario	39
11.4.11.	Creación del dashboard e integración Raspberry Pi3 con Tago IO	40
XII.	Interpretación de Resultados	46
12.1.	Media Aritmética	46
12.2.	Desviación estándar	47
12.3.	Tasa de Error Porcentual	48
XIII.	Discusión	49
XIV.	Conclusiones	51
XV.	Recomendaciones	52
XVII.	Anexos.....	58
Anexo 1	58
Anexo 2	60
Anexo 3	61

IX. Introducción

En un inicio, todo era manejado de forma física, es decir, que los procesos se realizaban manualmente, todo lo que era contabilizado se registraba en documentación física, todo escrito a mano. Las organizaciones tenían que contratar personal que les ayudara con la gestión de los productos, lo cual, significaba invertir recursos humanos, económicos y tiempo. Pasado los años, específicamente en la época de 1840, donde las industrias estaban haciéndose presentes, se integraron algunos tipos de sistemas de conteo a través de piezas, sistemas para almacenar productos, a fin de mantener un control más eficiente y organizado [1]. A pesar de ello, no quedaba exento el que hubiera errores humanos en el área de bodega, logística y contabilidad, específicamente problemas para actualizar datos al momento de realizar inventario.

Actualmente ya se cuenta con las herramientas de códigos de barra, etiquetas para identificar productos, lo cual sin duda significó una agilización para realizar los conteos de los productos de una manera mucho más rápida y organizada dentro de los locales. Sin embargo, aunque se logró optimizar el control y seguimiento, lo cierto es que aún se tenía la necesidad de que los empleados realizaran actualizaciones manuales dirigiéndose a los estantes, lo cual, se trataba de una limitante de obtener información precisa y rápida.

Por otro lado, con la llegada de la tecnología, se marcó un antes y un después en cuanto a la forma de realizar inventario, esto debido a que, gracias a las computadoras, se hizo posible el poder llevar registros de digitalmente, de modo que se fue suprimiendo el uso de documentos físicos. Algunos de estas herramientas informáticas tales como CRM y ERP se fueron integrando distintas áreas y procesos empresariales como contabilidad, ordenes de pedidos, entre otros [2]. No obstante, a pesar de ello, aún existían aspectos que podían mejorarse, como la visualización en tiempo real, etc.

Por consiguiente, como otro de los grandes avances con los que se cuenta en la actualidad, se tiene a las tecnologías IoT, las cuales también contribuyen como una opción de avance en la gestión de inventarios. En este caso, al complementarse con otros elementos como hardware, sensores que ayuden a detectar la presencia de productos, y dispositivos inteligentes, ha hecho posible que se pueda realizar un monitoreo en tiempo real. Una vez que los sistemas obtienen los datos capturados, estos son enviados a una aplicación web o móvil, donde se puede llevar un control, permitiendo tener acceso desde cualquier lugar y en cualquier momento [3].

Conforme a estas soluciones de gran avance tecnológico, que logran complementar sistemas de información con IoT, se ha logrado un avance considerable en la gestión de inventarios. Muy aparte de que estos sistemas den la posibilidad de consultar datos en tiempo real, permite que, con base a dicha información, las empresas puedan tomar decisiones eficientes y eficaces, que les ayuden a prevenir problemas futuros [4]. El contar con este tipo de transformación permite dejar de lado los métodos tradicionales, brindando una ventaja con otras organizaciones.

Dentro del mismo contexto, el hecho de que se tiene una necesidad que incrementa considerablemente respecto a la integración de métodos tecnológicos, el tema de mantener el uso tradicional y monótono de registro de inventarios ya no es parte de las decisiones de las empresas. En ese caso, las organizaciones además de contar con procesos internos más eficiente, también aspiran un modo de adaptación frente a los constantes cambios de la sociedad [5].

Adicionalmente, la evolución que ha tenido la economía durante los últimos años ha provocado que exista una sensación de demanda y presión en las empresas; las cuales buscan estar actualizadas constantemente y agilizar su modo de operación [6].

9.1. Arduino UNO

De acuerdo a la figura 1, se trata de una placa empleada de manera amplia en trabajos de electrónica y sistemas embebidos, se basa en el microcontrolador ATmega328P. Esta herramienta permite crear distintas apps enfocadas en la interactividad del usuario, trabajando mediante código abierto en su IDE de Arduino, y de la mano con el lenguaje de programación C, C++ [7]. Gracias a su funcionalidad, este recurso es sumamente utilizado para la elaboración de sistemas automatizados, así como también, el control de sensores a través de ordenadores u otros dispositivos electrónicos. Así mismo, se concibe como una excelente opción en los campos de la tecnología, académicos e investigativos, esto debido a que cuenta con un tipo de comunicación en serie que ofrece una óptima compatibilidad con otras herramientas [8].



Figura 1. Placa Arduino UNO

9.2. Sensores de peso con HX711

Tal como se aprecia en la figura 2, denominados como sensores de peso, se definen como dispositivos que se utilizan para calcular cuanta carga existe en una determinada celda, lo cual luego se convierte en una señal que es emitida con la finalidad de transmitir los datos del peso de los productos. Por otro lado, el “HX711” se trata de un módulo que permite amplificar las señales y convertirlas en bits, esto se utiliza específicamente cuando se calculan pesos de carga; permitiendo así que se lean las distintas variaciones mediante un dispositivo electrónico [9]. Adicionalmente, cuenta con la capacidad de poder comunicarse con protocolos seriales como Raspberry Pi y Arduino, lo cual lo hace un recurso bastante valioso, además, destaca por su bajo costo en el mercado, siendo relevante y muy solicitado en proyectos asociados a IoT [10].

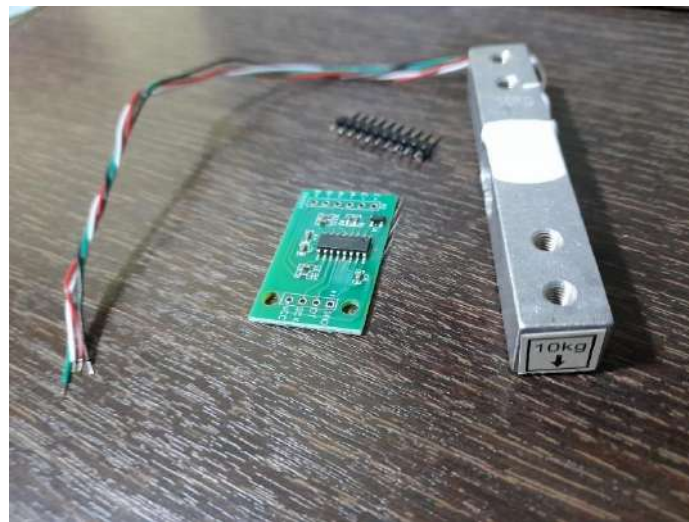


Figura 2. Sensor de peso

9.3. Lector RFID RC522

Tal como se aprecia en la figura 3, este dispositivo se centra en leer y escribir etiquetas identificadoras mediante radiofrecuencia. Gracias a su tecnología de emitir señales sin necesidad de contacto, hace posible la transferencia de datos, lo cual hace más fácil manejar sistemas de información que trabajen con la gestión de procesos, tales como el proceso de inventario [11]. Así mismo, es cien por ciento compatible con las tecnologías de Arduino y Raspberry Pi, de forma conjunta, estos elementos permiten crear una red que captura los datos de los productos, y luego estos viajan de manera fácil y rápida para transmitir la información, optimizando así los procesos de control y monitoreo [12].



Figura 3. Lector RFID RC522

9.4. ESP32

En la figura 4, se muestran los microcontroladores empleados en apps basadas en IoT, destacan mucho dentro del mercado tecnológico gracias a su bajo costo, y su alta capacidad de procesamiento [13]. El ESP32 se trata de un recurso más avanzado, el cual integra conexión Wi-Fi y con bluetooth, así como también cuenta con un alto procesamiento dirigido a diversos protocolos de comunicación. Gracias a que son versátiles, se logran adaptar fácilmente en la implementación de sistemas inteligentes, de control y monitoreo, etc [14]. En este caso el ESP32 del proyecto ya viene incluido en la placa Arduino UNO.

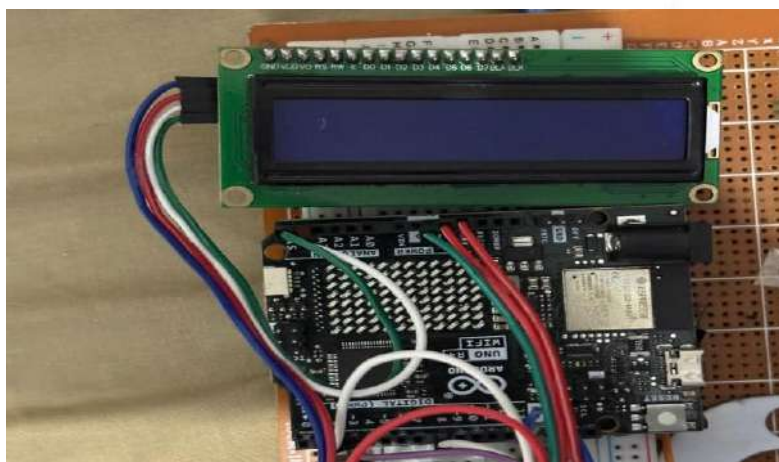


Figura 4.ESP32

9.5. Raspberry Pi

La figura 5 por su lado, es un computador conformado por una única placa, a diferencia de Raspberry Pi tradicional, esta se trata de una versión mejorada y que abarca un procesamiento mucho más avanzando. Dispone de un procesador ARM con cuatro núcleos en total, y una memoria RAM de 8GB, se conecta con Wi-Fi y bluetooth, así mismo, posee puertos USB y una salida HDMI doble, dándole la posibilidad de desplegar diversas apps [15]. Por otra parte, tiene una compatibilidad eficiente con S.O., brindando la facilidad de hacerse cargo de tareas complicadas, además, se utiliza mucho en contextos de educación e industrias, concibiéndose como un recurso importante en sistemas embebidos [16].



Figura 5. Raspberry Pi

9.6. Arduino IDE

Arduino IDE se define como un entorno de desarrollo para código abierto, especialmente para programación microcontroladores y placas Arduino [17]. Gracias a que ofrece una experiencia de usuario clara y accesible, hace mucho más fácil la escritura, compilación y carga de código a través de los dispositivos, y con ello, hace posible la creación de trabajos electrónicos basados en sistemas embebidos [18]. Dado que trabaja de la mano con los lenguajes de programación C y C++, cuenta con una gran cantidad de bibliotecas y esquemas, lo cual, permite optimizar la creación de apps que utilicen sensores de por medio, o que se centran en la comunicación a través de plataformas, adicional a ello, también mantiene compatibilidad sistemas operativos Linux, MacOS y Windows [19].

9.7. Python

Python se trata de un lenguaje de desarrollo de acceso libre, cuenta con una amplitud de uso en la creación de apps, analítica de datos, IA y sistemas embebidos [20]. Este lenguaje resalta mucho gracias a su sintaxis de fácil comprensión e interpretación para los desarrolladores, así mismo, cuenta con una versatilidad asociada a sus múltiples bibliotecas y módulos que hacen posible el desarrollo e implementación de algoritmos complicados [21]. Adicionalmente, Python tiene una compatibilidad con diversos S.O., y su equipo de desarrollo trabaja diariamente para proponer nuevos aspectos que logren mejorar su desempeño [22].

9.8. TagoIO

Se trata de una plataforma en la nube, cuyo diseño se centró en el desarrollo y gestión de soluciones de IoT (Internet de las Cosas) de una manera escalable y eficiente [23]. Así mismo, permite el análisis, recopilación, almacenamiento y visualización de los datos en tiempo real que provienen de los dispositivos conectados, lo cual hace más fácil el tomar decisiones automatizadas a través de reglas y scripts personalizados [24]. Por otra parte, TagoIO se encarga de ofrecer herramientas interactivas como dashboards, integrar diversos servicios y protocolos, además de brindar soportes a los lenguajes de programación, lo cual lo hace ser seleccionado por muchas empresas asociadas con tecnología IoT [25].

9.9. Dashboard web

Un dashboard web se define como una interfaz mediante la cual se puede visualizar, aplicar análisis y sobre todo tratar información mediante gráficos que se actualizan en tiempo real, así como reportes y métricas [26]. Este tipo de herramienta está diseñada para llevar una organización de los datos que se obtienen a partir de múltiples fuentes de información, todo llevado mediante una estructura, de modo que el usuario pueda leer e interpretar la información de forma fácil y rápida [27].

9.10. Ecuaciones a utilizar

Se trabajo de la mano con la media aritmética, la cual permitirá el efectuar el análisis y el procesamiento de los datos que se obtengan a partir de los sensores integrados en el sistema. Y con ello, se espera poder realizar el cálculo del promedio de las variables a ser evaluadas, tales como el peso, la cantidad de productos, comportamientos en general, etc, todo ello con la finalidad de que se tomen decisiones basadas en las mediciones en tiempo real.

Ecuación 1. Formula aritmética para calcular datos obtenidos de los pesos en tiempo real

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + a_3 + \dots + a_n}{n} \quad (1)$$

Donde

\bar{X} : media aritmética

n : número de datos

\sum : sumatoria

Como otra fórmula a aplicar dentro de proyecto se utilizará la desviación estándar, la cual permitirá calcular la variabilidad o dispersión de los datos que sean capturados por los sensores de peso, es decir, cuando se presenten las variaciones en el peso, conteo de unidades, entre otros. Será de mucha utilidad para la detección de errores en el sistema.

Ecuación 2. Formula aritmética para calcular la variabilidad o dispersión de los datos capturados

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2)$$

Donde:

- σ : desviación estándar
- n : número total de datos
- x_i : cada valor individual del conjunto
- \bar{x} : media aritmética del conjunto

Por otra parte, también se utilizó la fórmula de tasa de error porcentual, la cual permitirá comparar los datos esperados en el proceso de inventario, en contraste con los datos reales que el sistema arroja en términos porcentuales.

Ecuación 3. Formula aritmética de tasa de error porcentual esto permite comparar datos del inventario

$$EP = \sqrt{\frac{1}{n} \sum_{i=1}^n \left| \frac{V_{ri} - V_{mi}}{V_{ri}} \right|} \times 100 \quad (3)$$

Donde:

- V_{ri} : valor real o cantidad del producto esperada en la medición i

- V_{m_i} : valor medido por el sistema en la medición i
- n : total de mediciones realizadas
- El resultado se expresa en porcentaje (%)

X. Artículos Relacionados

Conforme a lo expuesto por los autores en el artículo [28], se habla de la importancia de implementar TIC como IoT, sistemas embebidos y computación en la nube en la gestión de inventarios del sector de la construcción. Para ello, se utiliza el sistema ASIMS, el cual integra códigos de barra, Arduino y el módulo ESP8266, a la par de alojamiento de datos en la nube. Gracias a ello, se permitió controlar y brindar seguimiento al encofrado de aluminio, una mejor trazabilidad, reducción de costos, así como prevenir anomalías, incorporando además medidas de seguridad mediante autenticación y cifrado SHA-256.

Dentro del mismo contexto, en el artículo [29] se centra en analizar sistemas automatizados de gestión de inventarios, principalmente enfocados en tecnologías IoT en pymes. Con estas tecnologías se ayuda a mejorar el control de stock, se reducen costes de almacenamiento y se cuenta con inventarios con mucha más precisión. Así mismo, se destaca el uso de sensores IoT e indicadores en tiempo real, lo cual permite optimizar la operatividad del negocio. Y con ello, se ayudó a mejorar entre un 25-35% el proceso de gestión de inventarios.

Por consiguiente, se comparte el artículo [30] el cual se centra en un sistema de inventario inteligente a fin de mejorar la gestión de niveles de inventario, entregas y pedidos. Este proyecto ofrece visualización en tiempo real y automatización, de forma que se ayudase a reducir los riesgos de sobreabastecimiento. Así mismo, se aportó a la mejora de la precisión del control en un 30% y seguimiento un 15%. Adicionalmente, se logró disminuir los tiempos de entrega y espera en un 80%. Con esta implementación se incrementó la rentabilidad en un 55% y se redujeron costos en un 25%.

Por otra parte, se expone el artículo [31] donde se presenta un sistema de monitoreo de inventario en tiempo real haciendo uso de la tecnología RFID. El objetivo de este proyecto se enfocó en reemplazar procesos manuales por uno automatizado y más preciso. El sistema permitió llevar un mejor control de stock y gestión del almacén mediante información detallada. Así mismo, se validó su eficacia a través de una simulación en tiempo real. Y, al termino se propuso su integración con otros modelos a fin de optimizar inventarios y mejorar las políticas de compra.

Así mismo, como otro artículo se comparte el [32] cuyo enfoque se basó en desarrollar una solución inteligente y automatizada para la gestión de inventarios en almacenes pequeños. Para lo cual, se utilizaron sensores IoT, etiquetas RFID y tecnología cloud computing para monitoreo en tiempo real. Además, se integra la inteligencia artificial a fin de generar KPIs de demanda y alerta de reposición.

Por otro lado, su interfaz fue sumamente amigable y ayudó a la reducción de errores humanos y optimización de tiempos.

Seguido de ello, se tiene al artículo [33] cuyo objetivo fue elaborar un sistema ciberfísico basado en IoT para empresa logísticas con almacenes. Para ello, se emplearon ordenadores portátiles, RFID y bluetooth para mejorar la eficiencia operativa y la colaboración. De esta forma, se resalta que, a diferencia de los sistemas tradicionales, este sistema permite la sincronización de información en tiempo real. Los resultados demostraron mejoras en eficiencia, sostenibilidad y control de operaciones.

Como otro artículo relacionado se tiene al [34], el cual se plantea un sistema automatizado de gestión de inventarios donde se emplea tecnología IoT y visión por computadora. A partir de ello, se puso como objetivo solucionar problemas e inconvenientes como errores de conteo, ubicaciones erróneas y discrepancias en los registros. En ese sentido, se utilizaron sensores, etiquetas RFID y cámaras inteligentes para monitorear productos y cantidades. Así mismo, se automatizaron tareas como identificación, conteo y detección de anomalías.

Por su parte, se expone el artículo [35] el cual se basa en el desarrollo de un sistema inteligente de inventario haciendo uso de tecnología IoT como opción de cambio ante métodos tradicionales. En este trabajo se utilizaron sensores, rastreo, análisis en la nube y comunicación entre máquinas. Con su implementación se ayudó a mejorar la visibilidad de inventario mediante la visualización de datos en tiempo real. Así mismo, se ayudó a la reducción de errores humanos y a optimizar el control de stock.

Como otro referente, se tiene al artículo [36] en el presente trabajo se propone un artículo basado en el desarrollo de un sistema automatizado, de bajo costo y confiable, el cual integra tecnología IoT y una base de datos en la nube, lo cual permite contar con un control y monitoreo ágil de inventario en las estanterías de los almacenes. A modo de validación, este estudio se corroboró mediante pruebas realizadas con prototipos, cuyo nivel de confiabilidad fue por encima de 0,8 en escala de Cronbach, lo cual se consideró totalmente viable para ser implementado en entornos reales.

Y, finalmente como un último referente, se presenta el artículo [37] en el cual se propone un sistema que combina drones (UAV) y blockchain a fin de automatizar procesos de inventario. Dentro del proyecto se utilizaron etiquetas RFID y cadena de bloques para asegurar la trazabilidad, integridad y transparencia de los datos. Así mismo, se emplearon contratos inteligentes para trabajar con automatización y sin intervención humana. Las pruebas efectuadas se hicieron en un entorno industrial real, donde se demostró una mayor agilidad frente a los métodos tradicionales.

XI. Metodología

11.1. Diseño de Estudio

11.1.1. Tipo de Investigación

El presente proyecto de titulación se basa en una de investigación aplicada, de tipo cuantitativa y experimental, esto dado que se enfoca en la creación de un sistema IoT dirigido al control y monitoreo de inventario en tiempo real, donde se utilizan sistemas embebidos de código abierto tales como ESP32 y Arduino. En ese sentido, los aspectos cuantitativos de la investigación se aplican en la recolección de datos numéricos a través de los sensores de peso y los dispositivos RFID; mismos que se analizan con fórmulas y estadística, tales como la media aritmética, la desviación estándar y el margen de error expresado mediante porcentajes. Gracias a ello, se pudo contar con una evaluación objetiva y que se pudo ser medida con el rendimiento del prototipo propuesto.

Así mismo, se concibe como aplicada, dado que nace de la investigación de teorías asociadas a la creación de sistemas automatizados enfocados en el control y monitoreo de inventarios en tiempo real, principalmente centrados en entornos comerciales de pequeñas y medianas empresas. Además, se integra un enfoque experimental, debido a que el proyecto ha sido verificado mediante pruebas de funcionamiento y toma de medidas en simulaciones, lo cual logró que se ayude a validar cuán efectivo y fiable era el sistema dentro de un entorno operacional. Adicionalmente, se aplicaron técnicas de modelado a fin de probar a través de la simulación el proceso del sistema.

11.1.2. Justificación del Diseño Elegido



Figura 6. Elementos del diseño del proyecto

- **Análisis del problema**

Tal como se aprecia en la figura 6, como un primer paso se pudo identificar la necesidad de que existía la falta de integración de sistemas modernos y automatizados en pequeñas y medianas empresas específicamente para el proceso de control de inventarios. El mantener métodos tradicionales y registros físicos provocaba que se generaran errores humanos, mal uso del tiempo y escasez de la visibilidad. Debido a este problema se propuso el desarrollo de una solución basada en la automatización, monitoreo y control de inventarios con una mayor eficiencia y precisión.

- **Revisión de la literatura**

En este punto se realizó una revisión de artículos científicos asociados al tema de estudio, con temas clave como la gestión de inventarios, tecnologías RFID, sensores, plataformas web, sistemas embebidos, entre otros. Gracias a este paso se logró identificar prácticas y elementos empleados que fuesen apropiados para el diseño del sistema propuesto, contrastando además soluciones distintas para establecer el enfoque y alcance que mejor se adaptase.

- **Selección de tecnologías y diseño del sistema**

Conforme a los requerimientos del sistema, se eligieron los componentes para su desarrollo, tales como Arduino UNO, ESP32, sensores de peso (HX711), lector RFID RC522, módulos WiFi y la placa Raspberry Pi como unidad central de procesamiento. Así mismo, se definieron el emplear herramientas para el desarrollo como Python, Arduino IDE y Tago IO como base de datos y dashboard web a fin de contar con una efectiva integración y visualización de los datos.

- **Desarrollo del prototipo**

En este paso se procedió con el ensamblaje y programación del sistema en físico. Para lo cual, se realizó la conexión de sensores de peso a las celdas de carga, se integró el módulo RFID para la lectura de productos y el uso de microcontroladores para el procesamiento de la información. Luego de ello, se continuó con la conexión de los dispositivos mediante protocolos inalámbricos, seguido del envío de los datos a la plataforma web visual (dashboard), lo que permite el monitoreo en tiempo real. Así mismo, en esta fase se realizaron las calibraciones y ajustes necesarios para contar con una lectura técnica de los datos.

- **Pruebas y validación del sistema**

Al término del ensamblado del prototipo, se llevaron a cabo pruebas de control para validar su funcionalidad. Para ello, se analizaron aspectos como la precisión de las lecturas de peso, que las etiquetas RFID puedan identificarse correctamente, así como la velocidad de la actualización del

dashboard. Se aplicó la estadística mediante fórmulas como la media aritmética, la desviación estándar y el margen de error con porcentajes, a fin de validar el comportamiento del sistema en varias condiciones.

- **Análisis de resultados**

Una vez que las pruebas fueron realizadas se obtuvieron los datos a fin de identificar posibles mejoras y fortalezas. Así mismo, se pudo observar que el sistema provee un control efectivo y eficiente del inventario, disminuye la propensión de errores humanos y brinda acceso inmediato de datos e información a cualquier dispositivo electrónico que cuenta con conectividad a internet. El poder visualizar los datos en tiempo real fue uno de los aspectos de mayor valor, esto corroborando cuán útil es el sistema propuesto para la toma de decisiones.

11.1.3. Hipótesis

El desarrollo e implementación de un sistema IoT de control y monitoreo en tiempo real basado en sistemas embebidos, permitirá ayudar a los empleados de los negocios pequeños en el procedimiento de gestión de inventario, transformando el inventario físico a uno automatizado y mucho más eficiente. Con este sistema se ayudará a reducir de manera significativa los errores humanos, así como también, se optimizarán los tiempos de registro y actualización de los productos, lo que ayudará a que se cuente con una visualización de la información de los datos del inventario mucho más precisa y efectiva, que podrá ser revisada desde cualquier dispositivo que tenga acceso a internet.

11.1.4. Diagrama, esquema del prototipo

Tal como se observa en la figura 7, se llevó a cabo el desarrollo del prototipo de un sistema IoT de control y monitoreo de inventario en tiempo real, donde se utilizaron sensores de peso HX711 que fueron integrados a las celdas de carga; las cuales identificaron la cantidad de productos en los estantes. En el caso de los sensores de peso, estos se conectaron a través de una placa Arduino UNO y un sensor shield v5.0, de modo que permitiera la recolección y comunicación de los datos. Por consiguiente, se incorporó un lector RFID RC522 con el cual se lograron diferenciar los productos mediante el uso de etiquetas electrónicas, y una pantalla LCD para mostrar la información en tiempo real. En ese sentido, finalmente la información recabada, luego fue enviada de manera inalámbrica al Raspberry Pi, donde se procesaron los datos y se visualizaron a través del dashboard web, el cual puede ser accedido para consultas mediante un ordenador o teléfono celular.

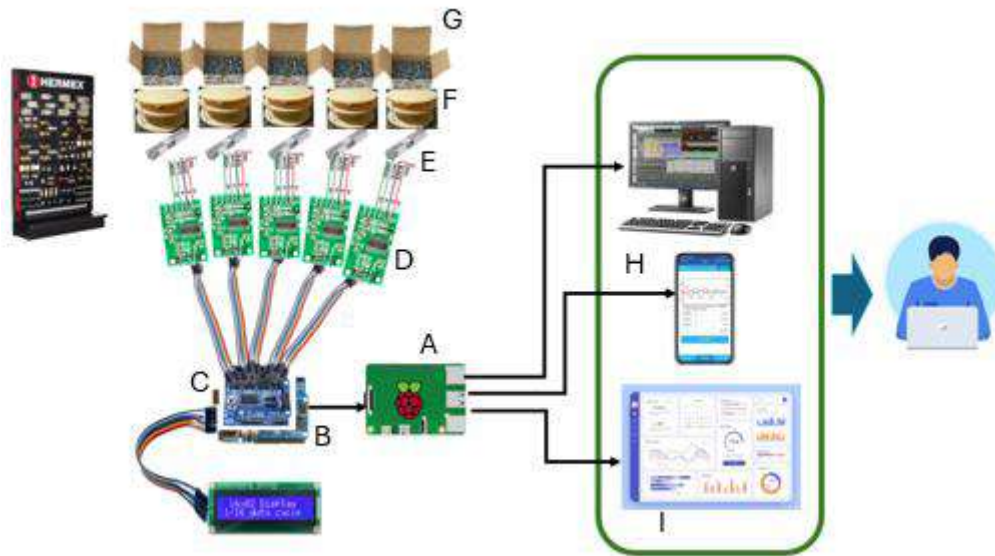


Figura 7. Esquema de prototipo del proyecto

(A) Raspberry Pi: Miniordenador que se encarga de recibir, procesar y enviar los datos recolectados por el sistema hacia plataformas externas.

(B) Placa Arduino: Microcontrolador que recibe señales de los sensores y realiza el procesamiento inicial de los datos de medición.

(C) Módulo Shield v5.0: Tarjeta de expansión que permite conectar la pantalla y otros periféricos al Arduino para visualizar y manejar información.

(D) Módulo HX711: Convertidor analógico-digital especializado en lecturas de celdas de carga, traduce las señales de peso a valores digitales.

(E) Celda de carga: Sensor que mide el peso de los productos transformando la fuerza en señales eléctricas.

(F) Estructura base: Soporte físico donde se instalan las celdas de carga y los productos a medir.

(G) Productos: Objetos (en este caso cajas) cuyo peso se controla y monitorea en el sistema.

(H) Equipo de cómputo: Computadora que recibe los datos procesados para análisis más avanzado o almacenamiento.

(I) Dashboard: Interfaz visual donde se presentan gráficas, estadísticas y reportes del sistema para facilitar la toma de decisiones.

11.2. Materiales y Herramientas

Tal como se describe en el Anexo 1, para el desarrollo del sistema IoT de control y monitoreo de inventarios en tiempo real, se emplearon diversos dispositivos electrónicos, herramientas para su implementación, módulos de comunicación, sensores para medición de peso, microcontroladores,

plataformas digitales que hicieron posible que exista una integración lógica y física del sistema. Cada uno de estos recursos se fueron eligiendo conforme a su compatibilidad, eficiencia, viabilidad, capacidad de trabajo dentro de entornos de sistemas embebidos de código abierto, entre otros aspectos relevantes.

En primera instancia, para el núcleo físico que da origen al proyecto, se utilizó la placa Arduino UNO con ESP32 integrado, lo cual hizo posible la combinación para el desarrollo del sistema, donde se dispuso de capacidades sumamente avanzadas para procesamiento y comunicación del microcontrolador ESP32. En este caso, la placa ya contó con la integración del módulo ESP32, considerada como híbrida, lo cual suprime la necesidad de que se trabaje mediante módulos separados para conexiones Bluetooth o Wi-Fi. Por lo tanto, el ESP32 funciona a través de un procesador dual-core de 32 bits, con una frecuencia que alcanza hasta los 240 MHz, 520 KB de SRAM, así como también, dispone de soporte para los protocolos I2C, UART y SPI, lo cual lo hace un elemento esencial para proporcionar una gestión eficaz y eficiente mediante el uso de sensores y transmisión de datos inalámbrica. Con ello, se tuvo la posibilidad de sintetizar el diseño del hardware y que la comunicación directa sea sencilla.

Por su parte, en cuanto al entorno de desarrollo aplicado, se utilizó Arduino IDE, la cual al ser una herramienta de código abierto permite el manejo de programación de los microcontroladores. En ese sentido, se aplicó para el desarrollo del firmware mediante lenguaje C/C++, donde se definió la lógica de la lectura de los sensores, el manejo de la pantalla LCD, el control de los lectores RFID y la conectividad Wi-Fi. Así mismo, el que sea compatible con distintas bibliotecas hizo más sencillo el proceso de la construcción e implementación del código indispensable para controlar el ESP32 y la interacción entre los sensores.

Por consiguiente, para el proceso de medición de la carga del peso de los productos, se hizo uso de las celdas de carga, las cuales fueron conectadas a los módulos HX711, los cuales hacen de conversores analógico-digitales de 24 bits. A partir de estos módulos se pudo contar con una amplificación de la señal generada por las celdas ante la detección del peso, lo cual, ayudó a que se garantice una precisión en las lecturas, inclusive con variaciones mínimas. De esta manera, cada una de las celdas HX711 se conectó mediante el shield de sensores de la placa Arduino, lo cual permitió que se cuente con una instalación mucho más segura y limpia, con una comunicación digital sincrónica, a fin de reducir las pérdidas de señal.

Respecto a la manera de identificar los productos, se utilizaron lectores RFID RC522, los cuales trabajan a 13.56 MHz, permitiendo así leer sin contacto las etiquetas RFID que van colocadas a los productos del inventario. Con esto, el módulo se comunica a través del protocolo SPI con la placa Arduino /ESP32, esto fue sumamente importante, dado que se promovió la trazabilidad y reconocimiento automático de

cada una de las unidades. De esta forma, se contribuyó a que los procesos de entradas y salidas de los productos fuesen mucho más rápidos sin necesidad de manipular manualmente.

Así mismo, para el procesamiento de la información capturada a través de los sensores de peso, se utilizó la unidad conocida como Raspberry Pi, el cual consta de 8 GB de memoria RAM y un procesador ARM Cortex-A72 de 64 bits (quad-core a 1.5 GHz). Gracias a esta microcomputadora se logró llevar a cabo la ejecución de los scripts que se desarrollaron en Python 3.10, donde se receptaron los datos desde la placa Arduino-ESP32 mediante los sockets o MQTT, así mismo, se efectuaron análisis estadísticos necesarios, tales como la media, desviación estándar y el margen de error, luego se prepararon los datos que son enviados al dashboard.

En cuanto a la visualización del inventario remota en tiempo real, esta fue tratada a través de la plataforma TagoIO (<https://tago.io/>), la cual se trata de una herramienta basada en la nube, la cual hace posible la creación de dashboards personalizados y que se adaptan a los distintos contextos del ámbito empresarial, además de que permite la consignación de datos históricos. En ese sentido, para integrar esta plataforma, se realizó una configuración de buckets de datos, dispositivos online y reglas condicionadas. Así mismo, la conexión efectuada entre el TagoIO y el Raspberry Pi se pudo establecer mediante el protocolo MQTT, en conjunto con la librería (paho-mqtt), permitiendo así realizar un envío efectivo y sumamente ligero de los datos en formato JSON. Adicionalmente, internamente en la plataforma, se procedió con la creación de widgets gráficos, contenido gráfico en línea, KPIs para medición de los pesos y alertas visuales a modo de notificaciones, a fin de mantener prevención frente a valores anómalos o lecturas que mantengan un rango incorrecto.

Para visualizar la salida de los datos localmente, se empleó una pantalla LCD de 20x4 con I2C, la cual va conectada a la placa Arduino/ESP32. Gracias a esta pantalla se pudo mostrar el tiempo real en el que el peso de los productos se detecta o se lee a través del identificador, lo cual permitió brindar una interfaz de usuario sumamente eficiente, que permite contar con una retroalimentación al instante sin necesidad de hacer uso de dispositivos adicionales.

Adicionalmente, se trabajó de la mano con la herramienta Telegram, la cual se utilizó para el envío de notificaciones a través de cualquier dispositivo. En este caso, se realizó la configuración de un bot, el cual se encargará de mostrar los datos que son capturados a través del dashboard como mensaje de texto, de tal forma que cada uno de los pesos de los productos con sus respectivas cantidades puedan ser visualizadas por el usuario de manera eficiente.

De esta forma, se determina que la integración de cada uno de estos elementos y herramientas, permitió la conformación de un sistema altamente robusto y escalable, mediante el cual se puede tener un manejo inteligente en el proceso de gestión de inventarios. Gracias a uso de tecnologías abiertas, protocolos ligeros, herramientas multiplataforma de desarrollo, se pudo contar con un sistema funcional y que se

adapta a las necesidades operacionales de las pequeñas y medias empresas comerciales de la ciudad de Guayaquil.

11.3. Procedimiento

11.3.1. Diagrama de flujo: Pasos para realizar el prototipo

A continuación, se muestra la figura 8, en la cual se desglosan cada uno de los pasos del prototipo:

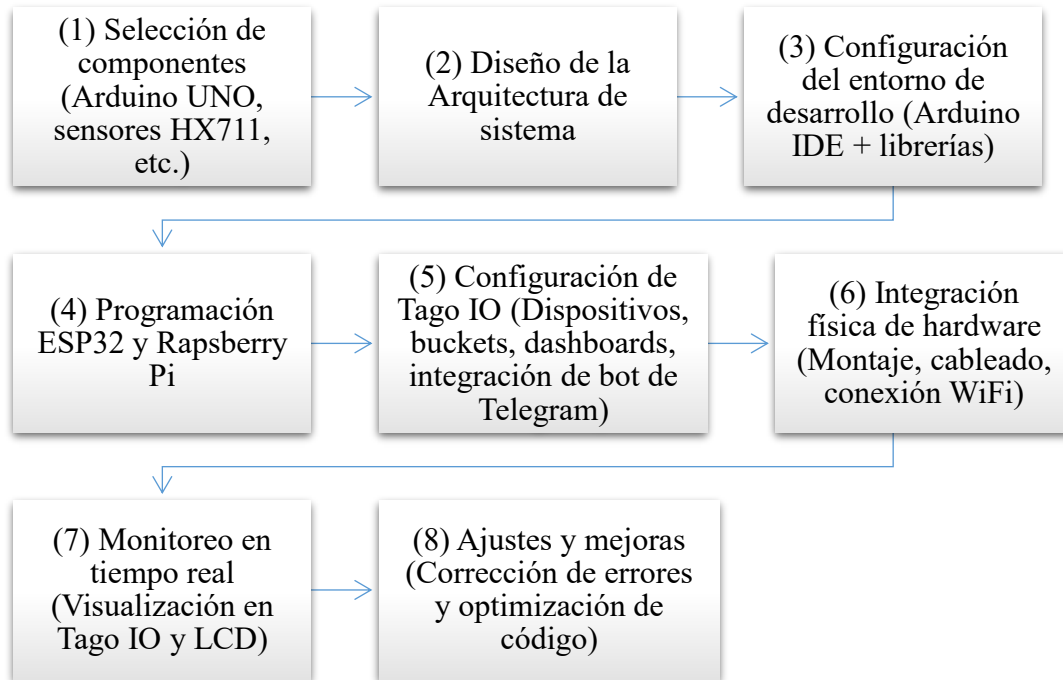


Figura 8. Diagrama de flujo de los pasos para la elaboración de prototipo

11.4. Desarrollo del prototipo (Descripción de fases)

11.4.1. Procedimiento para iniciar Arduino R4 Wi-Fi

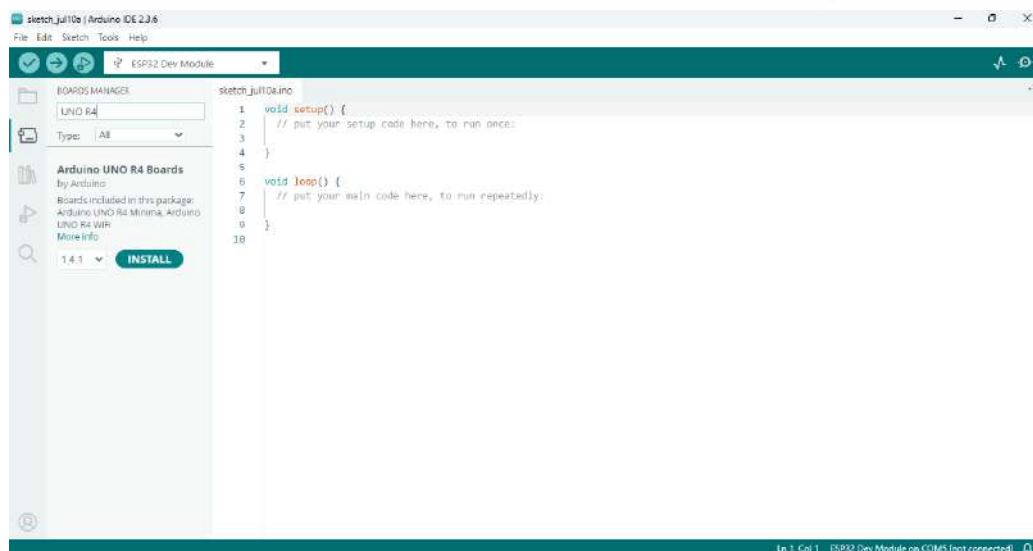


Figura 9. Proceso para iniciar Arduino R4 WiFi.

En la figura 9, se da inicio con la primera fase del desarrollo del prototipo, se preparó el entorno de programación a fin de que este sea compatible con la placa Arduino UNO R4 WiFi, misma que se caracteriza por sus avanzadas capacidades, tales como la matriz LED incluida y conectividad inalámbrica. A partir de este proceso se pudo garantizar que cada uno de los componentes del sistema embebido cuenten con un correcto funcionamiento, y se logre integrar con los datos de manera inalámbrica en tiempo real.

Como un primer paso se procedió a realizar la instalación del dispositivo Arduino UNO R4 WiFi dentro del entorno de desarrollo Arduino IDE. En este caso, se ingresó a la sección “Gestor de placas” que se encuentra alojado en el menú Herramientas > Placa, luego de ello se buscó el paquete perteneciente a las placas UNO R4. Luego de localizarlo, se continuó con la instalación del soporte que proporciona el Arduino para el modelo UNO R4 WiFi, gracias a ello el entorno pudo reconocer y compilar de forma correcta los programas que corresponden a la arquitectura nueva. Dentro de la instalación se incluyeron además definiciones importantes para comunicarse, así como el empleo de periféricos integrados en la placa.

Con el paquete de placas ya instalado, se realizó la conexión de la placa Arduino UNO R4 al ordenador a través de un cable USB-C. Con ello, el sistema operativo lo identificó el dispositivo de manera automática, y una vez estando internamente en el IDE de Arduino se procedió a seleccionar la opción “Arduino UNO R4 WiFi” como placa activa. Para ello, también se requirió seleccionar el puerto correcto (COMx en Windows o /dev/ttyACMx en sistemas UNIX), lo cual se realiza en el menú Herramientas > Puerto. Llevar a cabo esta configuración fue realmente importante a fin de contar con una conexión estable entre la placa y el ordenador, garantizando que el código de programación sea efectuado de manera correcta.

Por consiguiente, se continuó con la instalación de las librerías requeridas para que los módulos integrados en la UNO R4 WiFi funcionen de manera correcta, tales como la matriz LED, el módulo de comunicación WiFi (ESP32-S3) y otras herramientas necesarias para realizar la implementación del prototipo. Este procedimiento fue realizado desde el “Gestor de bibliotecas”, accediendo por el menú Programa > Incluir librería > Gestionar bibliotecas. Dentro de esta etapa se agregaron las librerías sugeridas por Arduino a fin de ganar el máximo provecho de las funciones de conectividad, procesamiento de la placa y visualización.

Luego de haber completado la preparación del entorno, se realizó la carga de algunos sketches a modo de ejemplo que fueron proporcionados por Arduino a fin de corroborar que la placa se encontraba funcionando correctamente. Así mismo se hizo uso del BlinkMatrix, a través del cual se valida de forma visual la funcionalidad de la matriz LEDs 12x8 integrada en la placa. De este modo, una vez terminado de compilar y el programa se cargó, se visualizó el despliegue de los patrones de luz sobre la matriz, lo

cual permitió confirmar que las respuestas de la placa funcionaban exitosamente frente a las instrucciones que era dadas desde el IDE.

Seguido de ello, para validar la conectividad inalámbrica, se realizó la ejecución del sketch de ejemplo WiFiScan, la cual tiene la posibilidad de escanear redes Wi-Fi que se encuentren disponibles en el entorno. De esta forma, al abrir el monitor en serie en el IDE de Arduino, se pudo visualizar la manera en cómo el módulo ESP32 S3 integrado en la UNO R4 WiFi en tiempo real; identificaba diversas redes inalámbricas que se encontraran cerca, dando a conocer sus nombres (SSID) y la señal a través de sus niveles. Gracias a esto se pudo corroborar que el Wi-Fi funcionaba de manera correcta y se encontraba lista y activamente para ser utilizada para las plataformas en la nube o dashboards a través del envío de los datos.

Y, como un último paso, una vez verificada la parte visual como la conexión, se dio término a la preparación e inicialización del entorno de trabajo con la placa Arduino UNO R4 WiFi. A partir de este procedimiento se logró asegurar que todos los recursos y elementos pudieran integrarse sin ningún problema al sistema durante las distintas etapas del proyecto.

11.4.2. Calibración del sensor de peso

En esta fase se desarrolló el procedimiento de calibración del sensor con el módulo HX711, el cual soporta pesos en un rango de hasta 10 kg a fin de asegurar lecturas fiables y precisas. Gracias a este proceso se pudo obtener el factor de conversión, importante para transformar las lecturas del ADC en unidades de peso (Kg).

Para la conexión física y preparación del sistema se conectó la celda de carga de 5 kg al módulo HX711 haciendo uso de los cables codificados: rojo (E+), negro (E-), blanco (A-) y verde (A+). Así mismo, el módulo HX711 se conectó al Arduino: VCC a 5V, GN a tierra, DT al pin digital D2 y SCK al pin D3.

Como primer paso, dentro del entorno Arduino se agregó un sketch de calibración inicial que permite la ejecución del método tare () luego de un retraso breve. Previo a ello, se dio una advertencia que no se agregue ningún objeto sobre la celda, esto con el objetivo de que pueda existir un registro correcto del peso de referencia cero (tara). En esta fase se efectuaron diversas lecturas get_value (n) o read_average (n) de tal forma que se asegure la estabilidad.

Con la tara ya estabilizada, se procedió a colocar sobre la celda de carga un peso patrón conocido. Se utilizó un elemento de 4 kg, lo cual se asemeja al máximo rango de vida útil de la celda (5kg). Donde se observó el valor sin la escala en el monitor en serie a través de las lecturas repetidas con get_value(10).

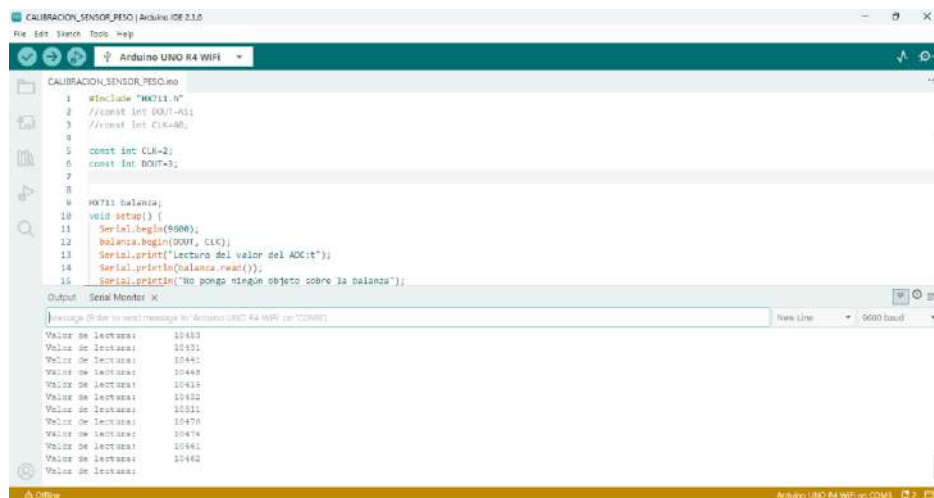
Por consiguiente, con el valor medio obtenido del peso de referencia, se realizó el cálculo del factor de calibración haciendo uso de la fórmula:

Factor de calibración = (peso real) / (lectura promedio del ADC)

Una vez realizado el cálculo, el valor se ingresó en el código mediante `set_scale` (factor)

Una vez se haya definido el factor de calibración, se editó el sketch final agregando la línea `scale.set_scale(0.01817)`; en el `setup()`. Por su parte, también se volvió a ejecutar el `tare()` para poder estar seguros de que cualquier residuo de peso sea eliminado. La codificación fue validada mediante la compilación y subida al Arduino, a partir de lo cual el sistema dio a conocer el mensaje “Listo para pesar” en el monitor en serie.

Tal como se muestra en la figura 10, se ejecuta sketch: `CALIBRACION_SENSOR_PESO.ino`



```

1 #include "HX711.h"
2 //const int DOUT=10;
3 //const int CLK=A0;
4
5 const int CLK=2;
6 const int DOUT=3;
7
8
9 HX711 balanza;
10
11 void setup() {
12   Serial.begin(9600);
13   balanza.begin(DOUT, CLK);
14   Serial.println("Lectura del valor del ADC");
15   Serial.println(balanza.read());
16   Serial.println("No ponga ningun objeto sobre la balanza");
17 }
18
19 void loop() {
20   Serial.println(balanza.read());
21   delay(100);
22 }

```

Serial Monitor X

```

Mensaje (5 de 10) recibi mensaje de Arduino UNO R4 WiFi por COM1
Valor de Lectura: 10453
Valor de Lectura: 10431
Valor de Lectura: 10441
Valor de Lectura: 10448
Valor de Lectura: 10415
Valor de Lectura: 10432
Valor de Lectura: 10311
Valor de Lectura: 10476
Valor de Lectura: 10474
Valor de Lectura: 10541
Valor de Lectura: 10462

```

Figura 10. Calibración del sensor de peso

Se coloca peso conocido: 50 g

Valor de lectura: 10453 (se calcula promedio de las 10 primeras lecturas de la imagen)

Con este valor se calcula la escala:

$$\text{ESCALA} = \text{VALOR DE LECTURA} / \text{PESO REAL}$$

$$\text{ESCALA} = 10453 / 0.05$$

$$\text{ESCALA} = 209060 \text{ (Este valor de escala se Calcula en Kg)}$$

11.4.3. Prueba del sensor de peso

En la figura 11, se ejecuta sketch: `LECTURA_SENSOR_PESO.ino`

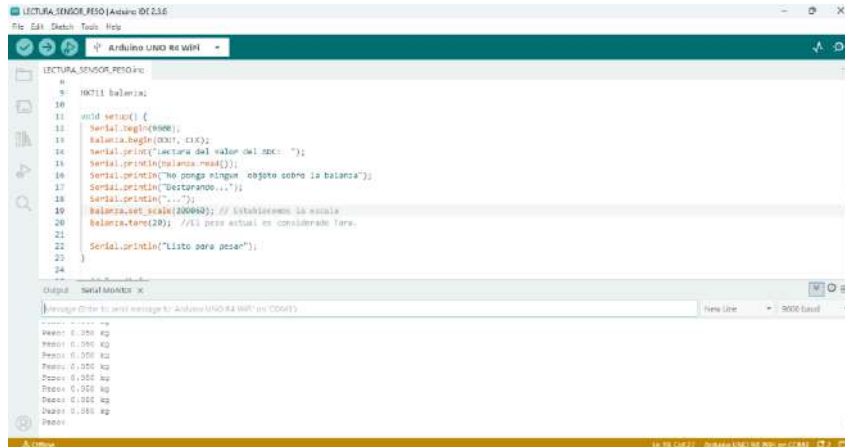


Figura 11. Prueba del sensor de peso

Se coloca peso conocido: 0.050Kg o 50 g

Se obtiene una lectura del sensor de peso de 0.050Kg en el monitor serial, tal como es el peso colocado.

11.4.4. Prueba de pantalla LCD I2 20x4

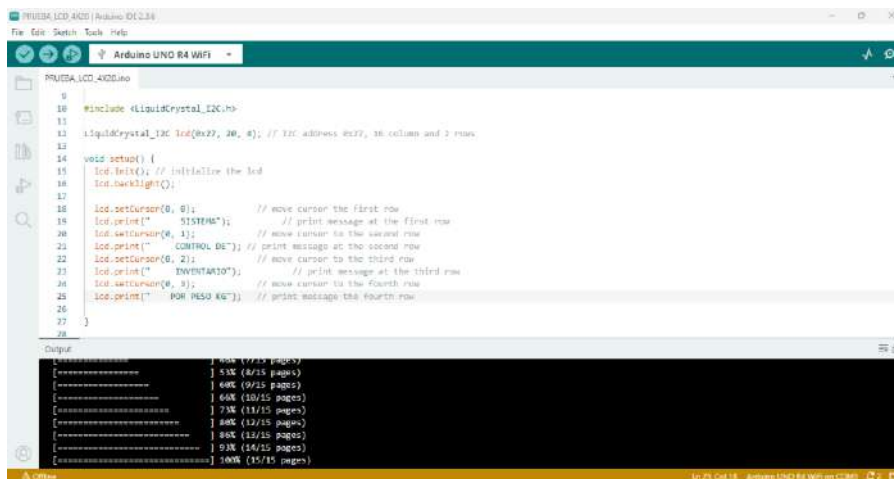


Figura 12. Prueba de la pantalla LCD I2 20x4

Como se observa en la figura 12, se validó el funcionamiento de la pantalla LCDE I2 de 20x4, a través de la cual se muestra la información en tiempo real. En ese sentido, la pantalla se conectó a la placa Arduino UNO R4 WiFi haciendo uso de los pines SDA y SCL, conjuntamente con la alimentación de 5V y GND.

Así mismo, en el entorno de Arduino IDE se empleó la librería LiquidCrystal_I2.h de la mano con la dirección 0x27. Se integró en el programa de prueba la inicialización de display, impresión de mensajes en cada una de las 4 líneas y la activación de la retroiluminación, seguido de posicionar el cursor con setCursor () y dando a conocer el texto con print ().

Al momento de la ejecución se pudo visualizar de manera correcta los siguientes textos:

Sistema:

CONTROL DE
INVENTARIO
POR PESO KG

En este caso, la pantalla dio respuesta sin errores, lo cual hizo posible que se verificara su operatividad y que se confirme como una lista de integración final. Gracias a su correcto funcionamiento se ayuda a garantizar que el usuario cuente con una interfaz clara durante el proceso de monitoreo de inventario.

11.4.5. Integración del módulo de expansión V5.0 para facilitar conexiones

Se monta módulo de expansión sobre el Arduino para facilitar conexiones con el módulo LCD y los 5 sensores de peso.

Con la finalidad de que se optimice la forma en que se organizan las conexiones y dar gestión a los cinco sensores de peso de la mano con la pantalla LCD, se agregó un módulo de expansión Sensor Shield V5.0. al sistema. Para este Shield se realizó un montaje directo en la placa Arduino UNO R4 WiFi, lo que permite conexiones plug-and-play a través de conectores de 3 pines (VCC-GND-Signal), lo que suprimió el que tener que usar una Protoboard con separación en los cables.

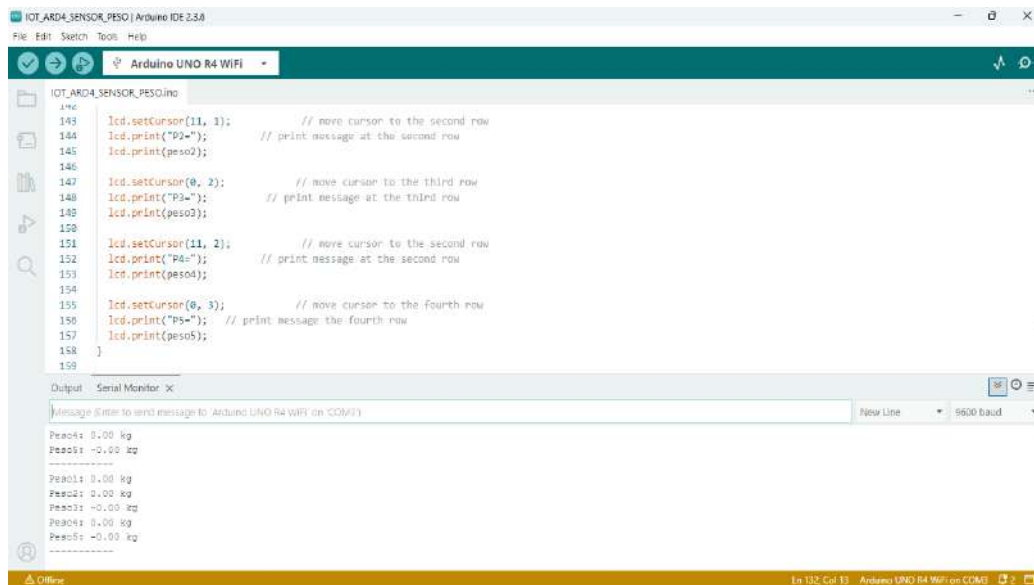
Se lleva a cabo una extensión de todos los pines digitales y analógicos del Arduino, además de las interfaces comunes tales como I²C, Serial, SPI, módulos bluetooth, LCD paralela/serial, ultrasonidos y control de servos. Cada uno de los puertos cuenta con una alimentación de 5V, señal y tierra, haciendo uso de cables tipo “buckled brick” que se insertan de manera fácil y segura, cada pin con etiquetado.

Proceso de integración

- 1. Montaje físico:** Los conectores del Arduino UNO se alinearon con el Shield, y luego se presionó firme a fin de poder contar con una conexión eléctrica correcta.
- 2. Conexión de sensores y módulos:** Se conectaron los sensores de peso con HX711 y la pantalla LCD I²C a los puertos que corresponden (los sensores a los puertos analógicos A0-A4 como las entradas, y la pantalla LCD al puerto I²C). Se garantiza una señal limpia y profesional gracias a los cables empleados.
- 3. Configuración en el código:** En este caso no se requirió modificar la lógica, se realizó la actualización de la referencia de conexión de modo que se ayudase a reflejar la extensión shield, confirmándose además que `Wire.begin()` (para I²C), además de que la lectura de los sensores HX711 tenían un correcto funcionamiento.

Conforme a la integración del Shield V5.0 se demostró un aspecto esencial dentro del desarrollo del prototipo. Por otra parte, aparte de hacer más sencillo el montaje en físico, se ayudó en la mejora de la robustez eléctrica del sistema, permitiendo además una gestión eficiente de sensores múltiples, lo que contribuyó a optimizar la calidad del proyecto final.

11.4.6. Desarrollo del Programa



```

IOT_ARD4_SENSOR_PESO | Arduino IDE 2.3.8
File Edit Sketch Tools Help
Arduino UNO R4 WIFI
IOT_ARD4_SENSOR_PESO.ino
143 lcd.setCursor(11, 1); // move cursor to the second row
144 lcd.print("P2="); // print message at the second row
145 lcd.print(peso2);
146
147 lcd.setCursor(0, 2); // move cursor to the third row
148 lcd.print("P3="); // print message at the third row
149 lcd.print(peso3);
150
151 lcd.setCursor(11, 2); // move cursor to the second row
152 lcd.print("P4="); // print message at the second row
153 lcd.print(peso4);
154
155 lcd.setCursor(0, 3); // move cursor to the fourth row
156 lcd.print("P5="); // print message the fourth row
157 lcd.print(peso5);
158 }
159
Output Serial Monitor x
Message (click to send message to: Arduino UNO R4 WiFi on COM1)
New Line 9600 baud
Peso01: 0.00 kg
Peso02: -0.00 kg
-----
Peso01: 0.00 kg
Peso02: 0.00 kg
Peso03: -0.00 kg
Peso04: 0.00 kg
Peso05: -0.00 kg
-----
En 132, Col 13 Arduino UNO R4 WiFi on COM1

```

Figura 13. Desarrollo del programa

Así mismo, de acuerdo con la figura 13, realizada la integración de los componentes físicos del sistema, se continuó con la construcción del código principal, donde se tuvo como finalidad la lectura de los datos que provenían de los cinco sensores de peso conectados al Arduino UNO R4 WiFi y dar a conocer sus lecturas en una pantalla LCD I2C 20x4. El desarrollo de este programa fue netamente para trabajar en tiempo real, relevando valores individuales en cada punto de peso.

Dentro del código se integró la librería LiquidCrystal_I2C.h para manejar el display, además de la configuración requerida para dar inicio con la dirección 0x27 y las dimensiones 20x4. En el setup () se estableció la comunicación con la pantalla, configurando el monitor serial y haciendo activa la retroiluminación para validar el funcionamiento a través de la salida de texto.

En el loop(), se realizó la programación de la lectura de los valores por cada uno de los sensores de peso; mismos que se encuentran almacenados en variables individuales (peso01, peso02, etc.). Luego de ello, se dieron a conocer los datos tanto en la pantalla LCD como en el monitor serial, donde se clasificó la información de manera ordenada por filas. Se empleó setCursor () para colocar el cursor en la fila y columna que corresponde, y print () para mostrar el valor de peso en kilogramos, así como el identificador de cada punto (P1-P5).

En la siguiente salida se puede apreciar que el sistema se encontraba en ejecución, a partir de cada una de las líneas del monitor serial se señala el valor de un sensor en el siguiente formato:

Peso01: 0.00 kg

Peso02: 0.00 kg

Peso03: 0.00 kg

Peso04: 0.00 kg

Peso05: 0.00 kg

De manera simultánea, en la pantalla LCD se mostró:

P2->	0.00
P3->	0.00
P4->	0.00
P5->	0.00

Figura 14. Pantalla LCD

Tal como se aprecia en la figura 14, el despliegue fue adaptado a fin de que los datos sean legibles y se encuentren debidamente organizados por cada sensor que vaya conectado al sistema. Para ello, se empleó una estructura de actualización con periodicidad para refrescar la pantalla sin parpadeos, a fin de mantener una continuidad en el funcionamiento del sistema.

11.4.7. Configuración de Raspberry Pi3

Dentro de este apartado se procedió con la integración del procesamiento y visualización del sistema IoT de inventario, se llevó a cabo la configuración de un Raspberry Pi3 y la instalación del sistema operativo en Raspberry Pi4.

Primero se efectuó la descarga e instalación del programa Raspberry Pi Imager desde el sitio oficial. Luego se introdujo una tarjeta microSD de 16 GB en un lector, posteriormente se empleó Imager para elegir la mejor versión para el Raspberry Pi OS, continuando con el montaje de la imagen en la tarjeta. Además, el software permite realizar configuraciones avanzadas (hostname, acceso SSH y conexión WiFi) previo a la grabación de la imagen.

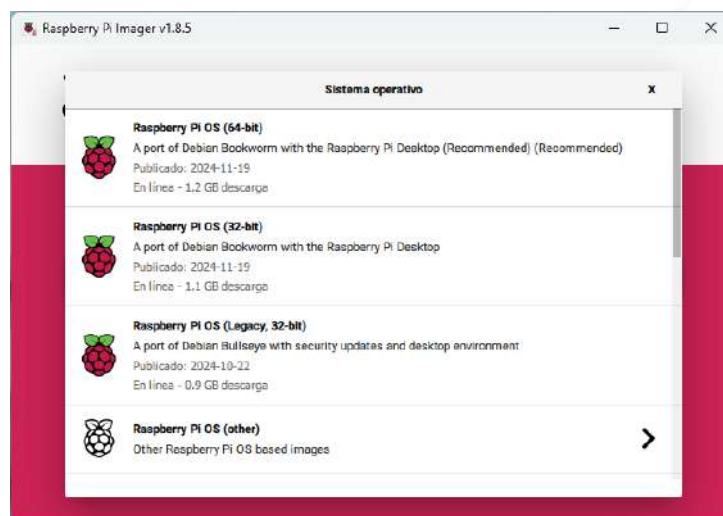


Figura 15. Instalación de Sistema Operativo

Así mismo, conforme a la figura 15, después de haber grabado la microSD, se introdujo en la Raspberry Pi 4 y se prendió el dispositivo. En este caso, la placa dio inicio de manera automática con el sistema operativo ya instalado. Al momento de realizarse el primer arranque se dio por terminada la configuración inicial (zona horaria, credenciales de usuario, idioma y red WiFi).

Por consiguiente, se habilitó el servicio SSH a fin de facilitar el acceso remoto, desde las opciones avanzadas del Raspberry Pi Imager o a través del raspi-config una vez ingresado al sistema. Con ello, se pudo administrar el Raspberry Pi desde otro computador con conexión remota, lo cual hace más fácil el despliegue del sistema IoT sin necesidad de una pantalla o teclado directamente conectados.

Seguido de ello, como se muestra en la figura 16 se abre una sesión SSH (o terminal) en el Raspberry Pi se dio ejecución a los siguientes comandos:

```
bash

sudo apt update
sudo apt upgrade
```

Figura 16. Actualización del sistema

En la figura 17, debido a que el sistema tiene interacción con sensores y módulos (como ADC externos, sensores I²C o SPI), se requirió habilitar los protocolos I²C, SPI y Serial UART. Se ingresó al menú de configuración y luego se ingresó a Interface Options, para proceder a activar los protocolos necesarios. Se reinició la Raspberry a fin de que se hagan los cambios.



Figura 17. Entorno de instalación Raspberry Pi

Credenciales del Raspberry Pi4

HOSTNAME: raspberrypi

CLAVE: admin

Pantalla principal del Raspberry Pi

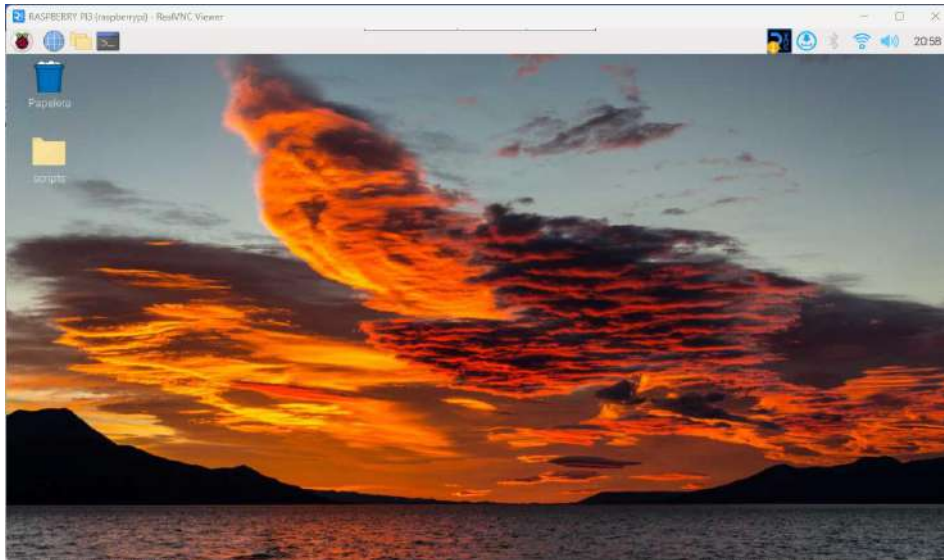


Figura 18. Pantalla principal del Raspberry Pi

Configuración del VNC en Raspberry Pi

En la figura 18, se realizó la configuración donde se habilitó el acceso remoto al entorno gráfico del Raspberry Pi haciendo uso del VNC (Virtual Network Computing), lo cual hace posible que se administre el dispositivo desde otro dispositivo sin una conexión a monitor, ni ningún otro periférico. Con esta función se monitorea y se visualizan los datos del sistema de inventario conectado al Arduino UNO R4 WiFi.

En primera instancia se realizó la actualización del sistema operativo a través de los comandos:

Bash

```
Sudo apt update
```

```
Sudo apt upgrade
```

De esta manera se pudo asegurar versiones finales de los paquetes y se preparó el Raspberry Pi para instalar el servidor VNC, tal como se muestra en la figura 19.

Luego de ello, se instaló el Servidor RealVNC (realnc-vnc-server) y el visor (realvnc-vnc-viewer) de forma directa desde los repositorios oficiales.

Así mismo, se realizó la activación del servidor VNC a través de los siguientes métodos:

- Interfaz gráfica: Desde el menú Preferences → Raspberry Pi Configuration. se habilitó VNC en la ventana Interfaces.
- Línea de comandos: Utilizando sudo raspi-config, se navegó hasta Interfacing Options → VNC → Yes.

En este caso, el servidor se dejó configurado para pueda dar inicio de manera automática al prender la Raspberry Pi.

Una vez que se haya reiniciado el sistema, se mostró el ícono del servidor VNC en la barra de tareas. Se ejecutó hostname -I en el terminal. Se utilizó RealVNC Viewer desde un equipo cliente para mantener la conexión se ingresó la dirección IP-de-la-Pi: 5900. Una vez conectado se accedió con las credenciales de usuario de la Raspberry (por defecto usuario pi y contraseña establecida).

Se dio como recomendación la modificación de la contraseña predeterminada por seguridad y, en una cuenta gratis de RealVNC, se puso como limitante el acceso de hasta cinco dispositivos conectados en la nube de forma opcional. Además, se validaron las configuraciones de cifrado y se pudo asegurar las conexiones privadas a través de SSL/TLS.

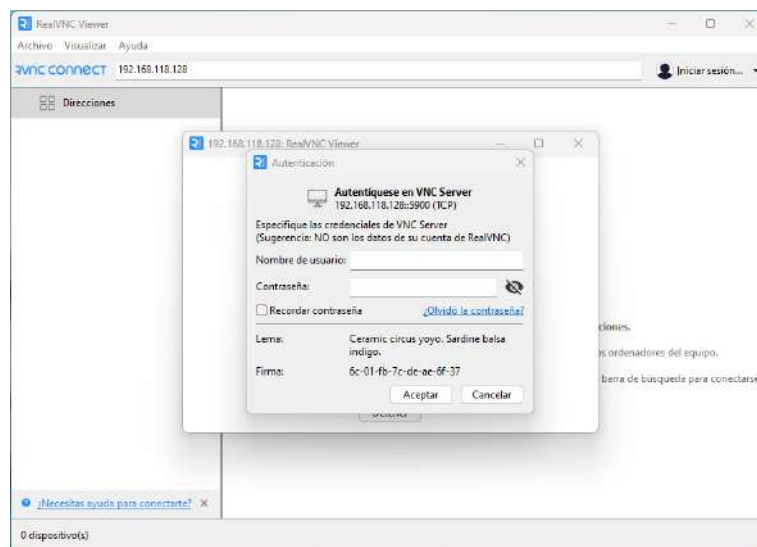


Figura 19. Entorno de configuración del VNC en Raspberry Pi

Credenciales del VNC:

HOSTNAME: raspberrypi

CLAVE: adminrpi

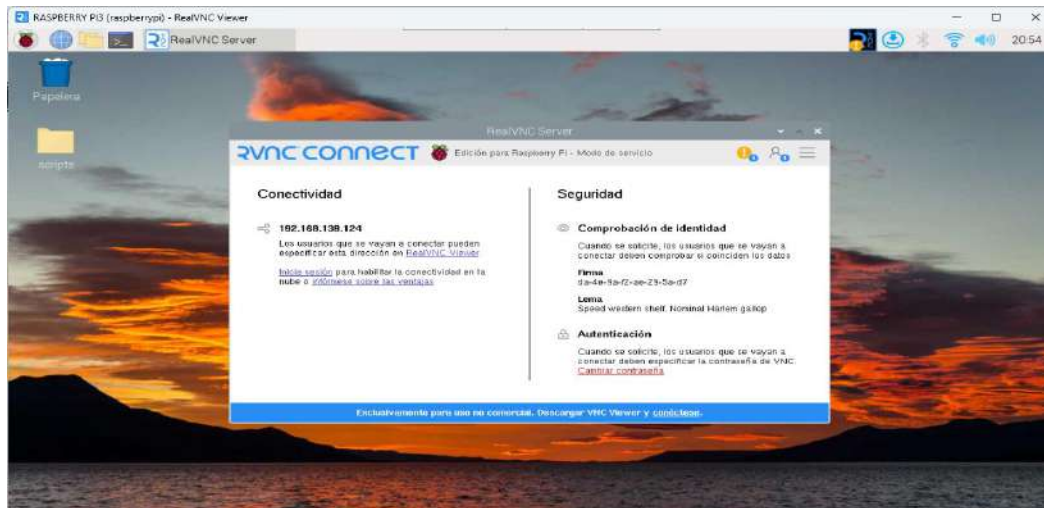


Figura 20. Entorno de configuración del VNC en Raspberry Pi

En la figura 20 se muestra el entorno de configuración de VNC en la Raspberry Pi. Este sistema permite el acceso remoto al escritorio de la Raspberry desde otro dispositivo. La interfaz presenta la información de conectividad, incluyendo la dirección IP utilizada para el enlace. Además, se detallan las opciones de seguridad como autenticación y verificación de identidad. De esta manera, se facilita la administración y control del dispositivo de forma remota mediante RealVNC.

11.4.8. Creación de Dashboard en Tago IO

Como primer paso se ingresó a la plataforma TagoIO y se creó un dispositivo nuevo, donde se seleccionó el tipo de conexión MQTT o Python SDK, esto depende de la disponibilidad de recurso o la preferencia. Una vez creado, se obtuvo un device token, importante para el envío como para la recuperación de los datos desde la Raspberry Pi.

Seguido de esto, se procedió a instalar en la librería Tago en la Raspberry Pi para Python (SDK) y/o paho-mqtt para comunicación MQTT. Gracias a ello se permitió el envío de datos directos al bucket de TagoIO o hacer uso de un cliente MQTT que sea seguro para la publicación de datos desde el Raspberry Pi hacia el servidor.

Se creó un script que incluyó lo siguiente:

```
from tago import device
my_device = device("device_token_here")
data = {
    "variable": "weight01",
    "unit": "kg",
    "value": peso01
}
my_device.insert data
```

Con este código se enviaron los valores de los pesos desde los sensores dirigidos hacia la plataforma TagoIO conforme a las variables que se definieron. Luego de ello se volvió a repetir el proceso para los otros sensores (peso02, peso03, etc.)

Mediante el uso de MQTT se creó un script Python (send_mqtt.py) tal como se aprecia en la figura 21, donde se publican datos en formato JSON al broker de TagoIO:

```
import paho.mqtt.client as mqtt
import json

device_token = "DEVICE_TOKEN_HERE"
broker = "mqtt.tago.io"
topic = "tago/data/post"

client = mqtt.Client()
client.username_pw_set("mqtt_client", device_token)
client.connect(broker, 1883, 60)

msg = {"variable": "weight01", "unit": "kg", "value": peso01}
client.publish(topic, json.dumps(msg))
```

Figura 21. Script de Python

Se repite el proceso para cada peso y luego se desconecta el cliente. Esto hizo posible el registro de los datos en el bucket vinculado al dispositivo en TagoIO.

Generación de credenciales

En la figura 22 se ilustra el proceso de registro y gestión de credenciales en la plataforma **TagoIO**, necesaria para la integración de los dispositivos IoT. Primero, se procede a la creación de una cuenta ingresando los datos personales, correo electrónico y contraseña. Posteriormente, se registran las credenciales de acceso que permitirán la autenticación en el sistema. Estas credenciales son almacenadas para su uso en la conexión de hardware como Raspberry Pi y ESP32. Finalmente, se garantiza el acceso seguro al proyecto denominado Sensor de Peso dentro de la nube de TagoIO.

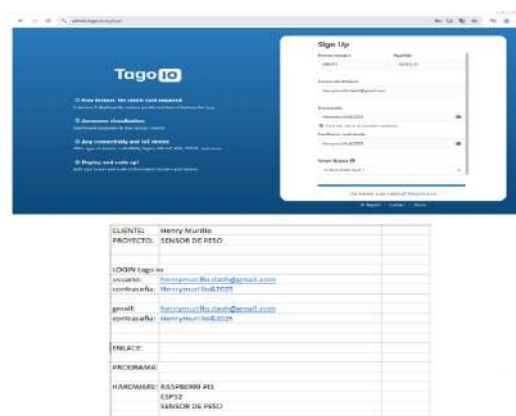


Figura 22. Proceso de generación de credenciales en Tago IO

En la figura 23 se genera una clave o token al momento de crear un dispositivo, en este caso el Raspberry Pi3. Este token es un identificador único que debe incorporarse en el programa del Raspberry para establecer la comunicación segura con la plataforma. Gracias a este proceso, TagoIO puede reconocer y autenticar al dispositivo correctamente. De esta manera se asegura que los datos transmitidos provengan de la fuente configurada. Este mecanismo es esencial para garantizar la conexión y gestión confiable de información dentro del sistema IoT.

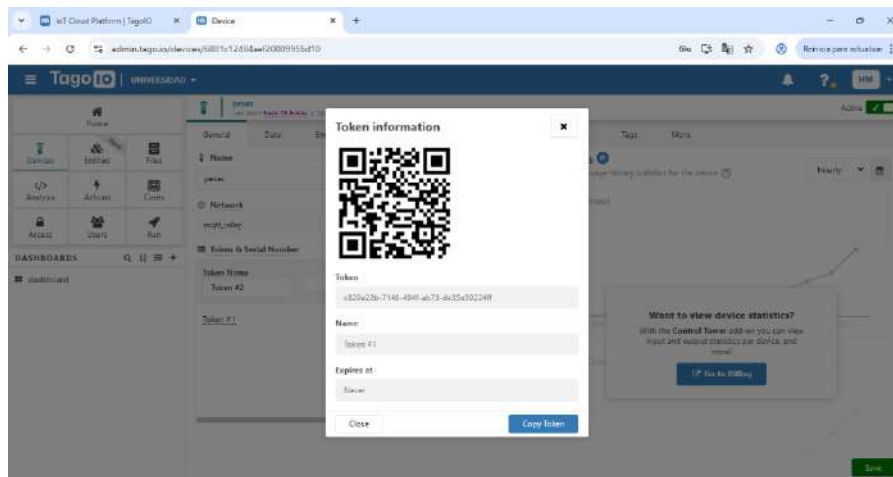


Figura 23. Código de reconocimiento del Raspberry

A continuación, en la figura 24 se crea el método para recibir los datos mediante un bróker MQTT, esta es la herramienta de software que permite enlazar las variables que arriban a la plataforma. Este método lo asigna la plataforma mediante una acción que vincula los datos transmitidos por el broker. Finalmente, la acción generada asegura la correcta integración de la información dentro de la nube de TagoIO.

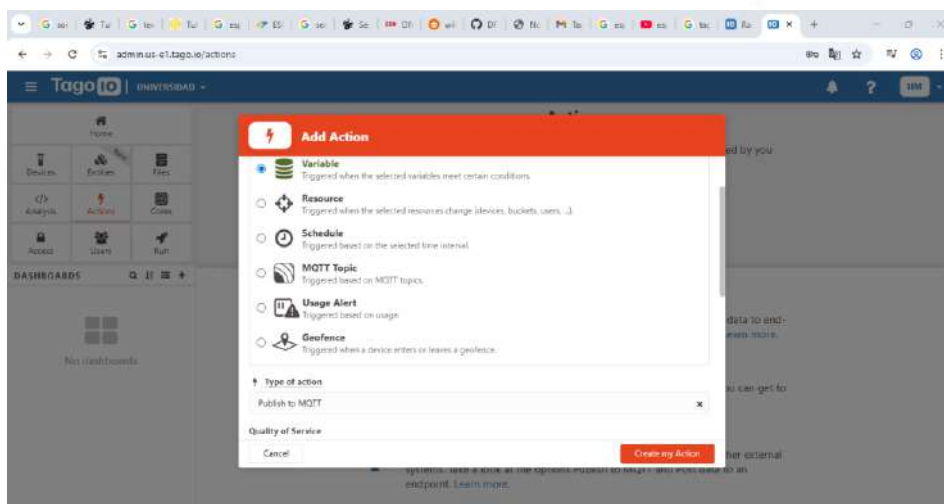


Figura 24. Creación de método para recibir datos

11.4.9. Creación de bot de Telegram

```
# --- CONFIGURACIÓN TELEGRAM ---
TELEGRAM_BOT_TOKEN = "8305680495:AAFwvnB5Pie_ZTelbvUNNfNasqX4kZDN9B0"
CHAT_ID = "1559476765"
telegram_url = f"https://api.telegram.org/bot{TELEGRAM_BOT_TOKEN}/sendMessage"

# Control de tiempo para envío de Telegram
TIEMPO_NOTIFICACION = 5 # segundos
ultimo_envio = 0

# --- NOMBRES PERSONALIZADOS PARA LOS ÍTEMS ---
item_names = {
    "1": "Esparrago de 1/2",
    "2": "Perno de 10mm",
    "3": "Anillo plano 22m",
    "4": "Anillo plano 3mm",
    "5": "Anillo de 4mm"
}
```

Figura 25. Creación del bot de Telegram

Como se aprecia en la figura 25 En primera instancia, se establecieron los elementos y parámetros necesarios para la configuración del bot de Telegram, donde fue necesario contar con un Chat_ID y un Bot_Token. A través del Bot_Token se permite identificar de manera única el bot en Telegram. El bot fue creado mediante BotFather, permitiendo así tener una autenticación del acceso para los mensajes que se envían. Por otro lado, Chat_ID actúa a modo de identificador en el chat donde llegarán las notificaciones. A partir de ambos elementos se crea el enlace o URL de Telegram, la cual se empleará como vía para el envío automático de los mensajes.

Por consiguiente, se definió el tiempo de control para el envío de notificaciones, utilizando las variables TIEMPO_NOTIFICACIÓN = 5, lo cual establece que para cada mensaje se demorará un mínimo de 5 segundos en ser enviado, esto a fin de prevenir saturaciones por demasiados registros al mismo tiempo. Así mismo, la variable ultimo_envío, la cual mantiene una inicialización en 0 con el objetivo de controlar el instante en que se realizó la ejecución de la última notificación. Y, por último, pero no menos importante, se desarrolló un diccionario que llevó el nombre de item_names, el cual hizo posible personalizar los nombres de los ítems, de tal forma que al lado de cada producto se pudiera visualizar la descripción del mismo.

11.4.10. Proceso de obtención de los datos de inventario

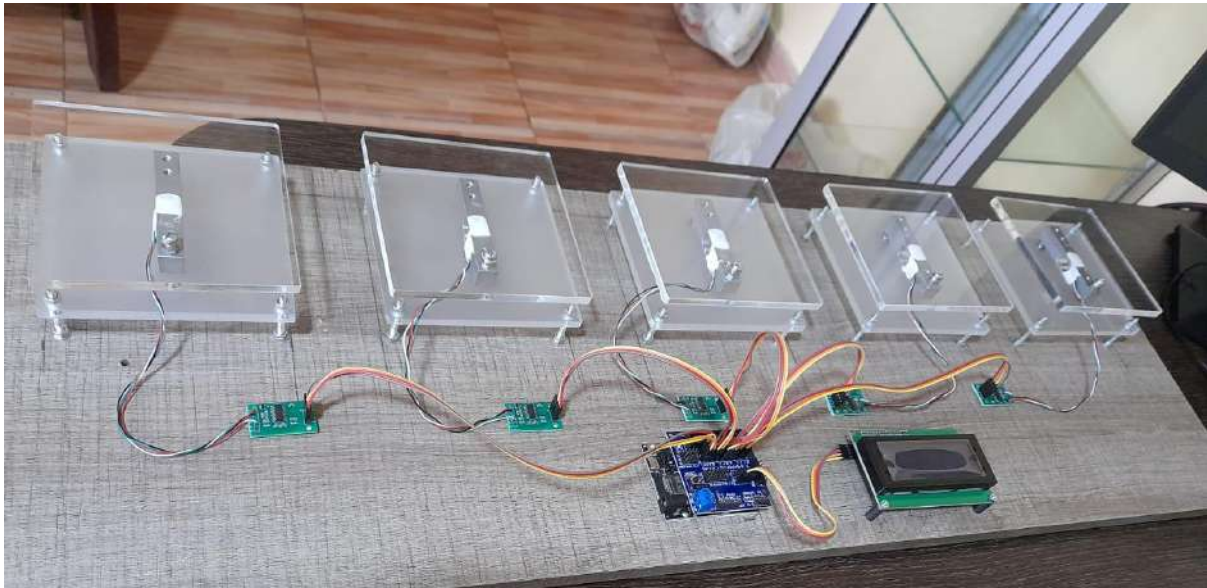


Figura 26. Proceso de obtención de pesos

Tal como se aprecia en la figura 26, el prototipo se encuentra compuesto por cinco plataformas acrílicas cada una con sus celdas de carga y conectadas a un módulo HX711. De esta forma, a través de estos módulos las lecturas son enviadas al Arduino UNO R4 WiFi, que realiza el procesamiento de los datos y los muestra en la pantalla LCD. Con el sensor Shield V5.0



Figura 27. Proceso de obtención de pesos

Seguido de ello, en la figura 27 ya se puede apreciar los recipientes colocados sobre las cinco plataformas de pesaje, lo cual permite representar los productos de inventario. Cada uno de los recipientes se encuentra sobre una celda de carga, esto hace posible que se registre el tiempo real. Una vez generadas las señales estas se procesan a través del Arduino UNO R4 WiFi, para luego ser

mostrados en la pantalla LCD. Así mismo, la información se mantiene lista para luego ser enviada a través del Raspberry Pi, y luego al dashboard donde podrá ser controlada y monitoreada.

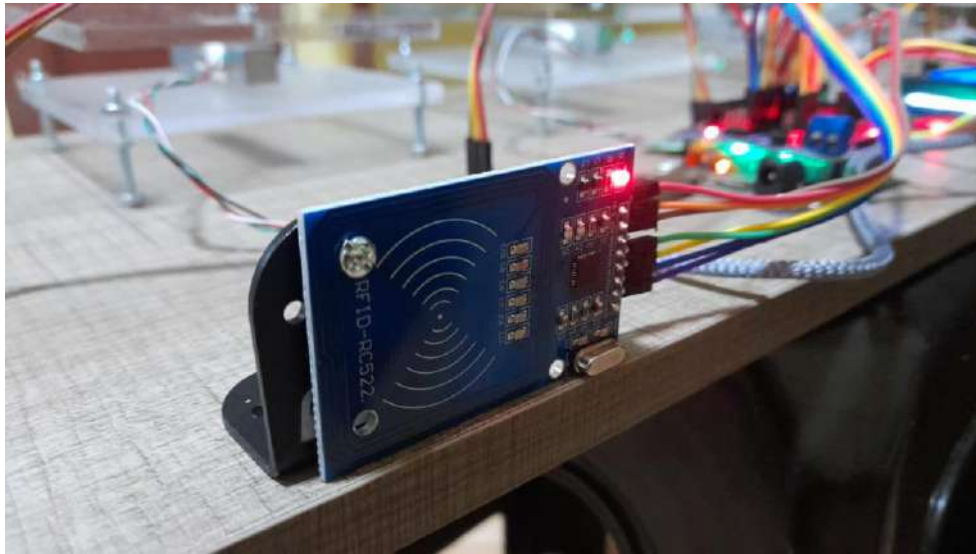


Figura 28. Lector RFID

De acuerdo con la figura 28, en esta fase se integró el lector RFID RC522 al sistema, donde su función se centró en identificar productos a través de las etiquetas electrónicas que cuentan con un código único (UID). Se conectó el módulo al Arduino UNO R4 WiFi haciendo uso del protocolo SPI, esto permitió la detección de cada etiqueta estando cerca del lector. A modo de validar, se desarrolló un programa en Arduino IDE donde se puedan los UIDs en los monitores serial y la pantalla LCD, esto permite que los productos registrados puedan ser visualizados de manera inmediata. Así mismo, dicha integración hizo posible que se automatice la identificación del inventario, esto relacionando cada etiqueta RFID con el peso que corresponde en tiempo real, esto ayudó a que se optimice el control y se ayude a la reducción de errores. Para más detalle de los elementos del prototipo véase el Anexo 2.

11.4.11. Creación del dashboard e integración Raspberry Pi3 con Tago IO

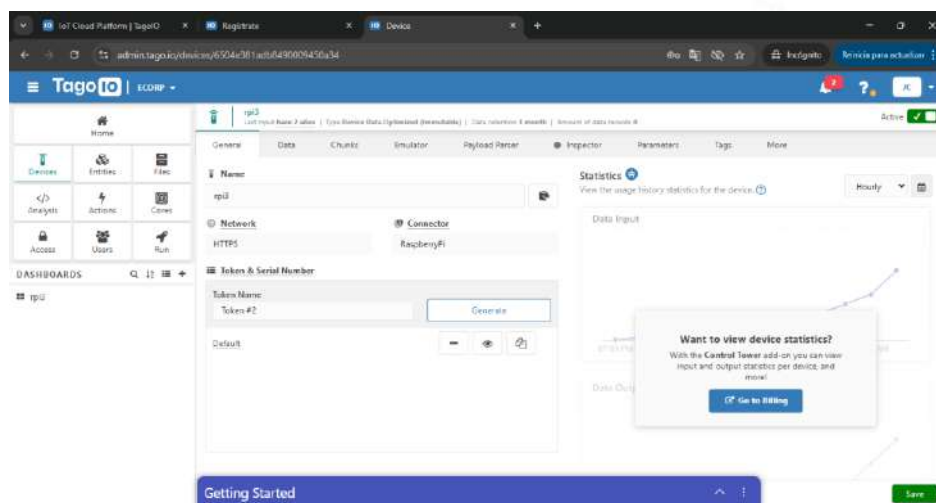


Figura 29. Proceso de integración del Raspberry con Tago IO

Tal como se muestra en la figura 29, dentro de esta interfaz se llevó a cabo la configuración del dispositivo “rp3” en la plataforma Tago IO, donde se realiza la integración con el Raspberry Pi 3. Para ello, se creó un dispositivo nuevo el cual fue seleccionado como “RaspeberryPi” y haciendo uso del protocolo HTTPS para la comunicación de los datos. Se continuó con la generación de un token de autenticación, el cual fue empleado por el Raspberry Pi para enviar datos de los sensores al servidor de TagoIO.

A partir de esta integración se permitió que el Raspberry haga de puente entre el hardware de lectura de inventario (Arduino y sensores de peso) y la plataforma en la nube, donde los datos se pueden visualizar en tiempo real. Gracias a que se cuenta con una configuración de token correcta, se garantiza una transmisión confiable y segura de la información.

Resultado final del Dashboard



Figura 30. Ventar del entorno Dashboard

Como se observa en la figura 30 ya con las fases de diseño, ensamblaje y configuración terminadas del sistema IoT, se continuó con la construcción del dashboard mediante la plataforma TagoIO, cuyo objetivo es poder representar, y visualizar los datos que se han recopilado a partir de los sensores de peso integrados al prototipo. Tal como se aprecia en la figura 28, el dashboard cuenta con un diseño sumamente intuitivo y modular, lo que permite mantener un control y monitoreo de los datos del inventario de manera organizada. Gracias a esta interfaz, los usuarios encargados de manejar la gestión de inventarios pueden hacerlo con mucha más precisión y desde cualquier dispositivo que cuente con acceso a internet.

Tal como se aprecia en el Anexo 3, el dashboard se encuentra compuesto por diversos widgets, donde cada uno de ellos corresponde a un sensor de peso previamente instalado mediante las plataformas de medición. El logo institucional de la Universidad Politécnica Salesiana, continuando con el título del sistema, se encuentran en la parte superior izquierda. Seguido de ello, debajo se encuentran los paneles

catalogados como “peso 1”, “peso 2” hasta el “peso 5”, donde se muestran KPIs o indicadores gráficos con sus correspondientes representaciones lineales. Los diales hacen posible poder observar de forma inmediata el valor actual de los pesos que son capturados por cada celda de carga, por otra parte, las gráficas ayudan a representar el historial de variaciones en el tiempo, haciendo que sea mucho más fácil identificar los patrones o cambios dentro del inventario. Gracias a la doble visualización el usuario puede interpretar datos puntuales, así como el comportamiento históricamente.

Cada uno de los sensores se calibró de manera previa y se fue vinculado con variables independientes (peso1, peso2, etc.) los cuales permite alimentar los gráficos dinámicamente. De esta forma, TagoIO recibe los datos procesados a través del Raspberry Pi desde el Arduino UNO R4 WiFi, para luego representarlos visualmente mediante el protocolo MQTT o mediante scripts de Python. Así mismo, se configuró el sistema a fin de identificar valores que presentaran anomalías, y luego proceder con la emisión de alertas visuales solo de darse el caso de que se superen los límites preestablecidos, de tal forma que se ayude a mejorar la prevención y control de errores en el proceso de gestión de inventario.

Se resalta que gracias a su interfaz se puede escalar funcionalmente a través de la integración de más sensores de peso o nuevas variables. Por consiguiente, los datos que se recopilan tienen la posibilidad de guardarse en la nube, lo que habilita la creación de reportes históricos y estadística mediante análisis. La usabilidad del dashboard no solamente ayuda con la optimización del monitoreo, sino que a su vez permite reforzar la toma de decisiones como estrategias empresariales mediante la generación de información accesible, confiable y en tiempo real.

Datos Obtenidos

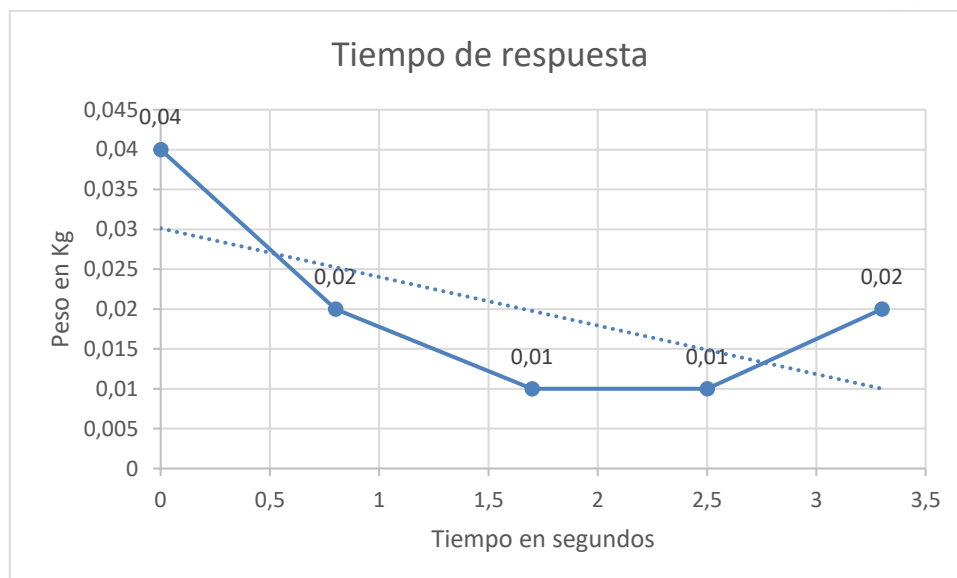


Figura 31. Tiempo de respuesta con un primer peso de los productos

Como se aprecia en la figura 31, se registró un tiempo de 0 segundos hasta los 3,3 segundos, cuyo peso estuvo entre 0,01kg; perteneciente a una unidad de esparrago de $\frac{1}{2}$. Luego de ello, a los 0,5 segundos se incrementa a un peso de 0,11 kg, esto dado que se colocó una unidad de perono de 10 mm. Por consiguiente, a los 1,1 segundos se registra un peso de 0,07 kg, el cual corresponde a la unidad de anillo plano de 22 mm. Así mismo, a los 2,0 segundos se ve una disminución del peso a los 0,05 kg, lo cual se asocia a la colocación de una unidad de anillo plano de 3 mm, y finalmente, a los 3,5 segundos se observa un peso que se incrementó a los 0,07 kg, esto debido a que se colocó un anillo plano de 4 mm.

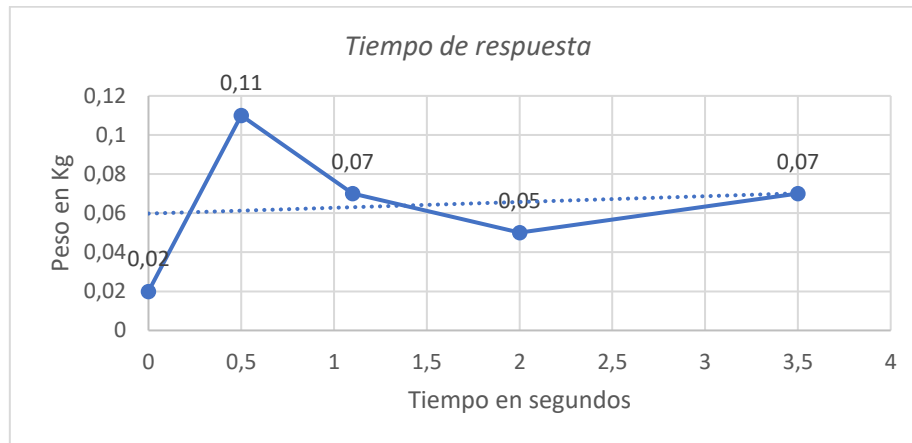


Figura 32. Tiempo de respuesta con un segundo peso de los productos

Por consiguiente, en la figura 32 se registró un tiempo de medición de 0 segundos a 3,5 segundos, en este caso con una detección inicial de peso de 0,02 kg, lo cual corresponde a 5 unidades de Esparrago $\frac{1}{2}$. Seguido de ello, el peso cambia a 0,11 kg, dado que se hizo un cambio de producto por 5 unidades de perno de 10 mm. Por consiguiente, a los 1,1 segundos se ve una disminución del peso a 0,07 kg, lo que indica que se colocó 5 unidades de anillos planos de 22 mm. Además, a los 2,0 segundos el peso cambia a 0,05 kg, lo cual se asocia 5 unidades de anillos planos de 3 mm, y por último, se detecta un último valor de peso a los 3,5 segundos de 0,07 kg, lo que hace referencia a la colocación de 5 unidades de anillo de 4 mm.

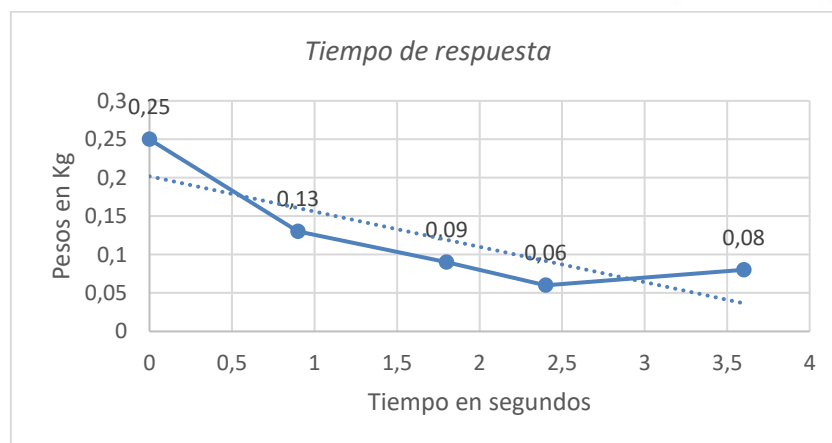


Figura 33. Tiempo de respuesta con un tercer peso de los productos

En la figura 33 se pudo observar el registro de un peso inicial de 0,25 kg el cual se detectó en el segundo 0 correspondiente a 6 espárragos de 1/2. Luego de ello, el sistema dio a conocer una disminución de carga entre los 0,9 y 1,8 segundos, para luego alcanzar los 0,09 kg, esto debido a que se colocaron un total de 6 anillos planos de 22mm. Por su parte, en los 2,4 segundos el peso disminuye a 0,06 kg, porque hubo un cambio de producto de 6 anillos planos de 3mm. Por último, en el segundo 3,6 se puede observar una estabilidad de 0,8 kg, dado que se ingresaron 6 anillos planos de 4mm, adicional, se destaca la precisión del sistema para detectar las entradas y salidas de los pesos.

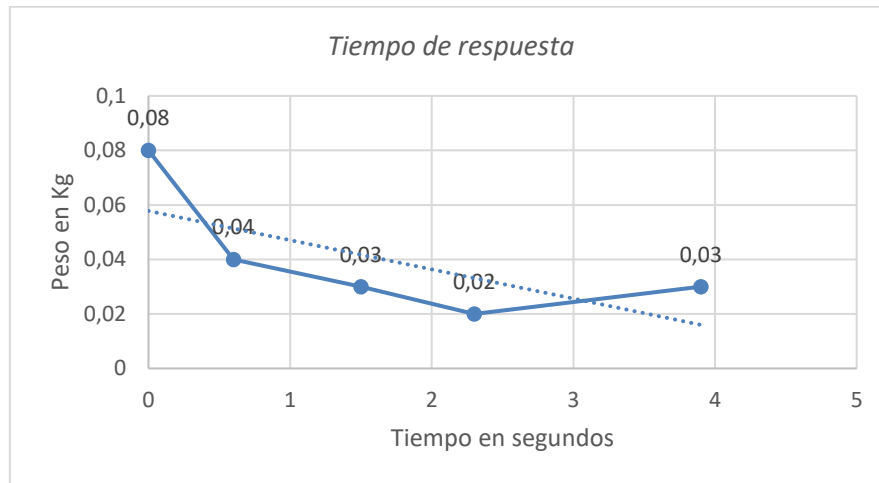


Figura 34. Tiempo de respuesta con un tercer peso de los productos

Seguido de ello, en la figura 34, se logra observar un peso inicial de 0,08 kg durante la primera medición dado que se colocó 2 espárragos de 1/2. En ese sentido, a los 0,6 segundos se ve una disminución del valor a 0,04 kg, esto dado que se cambió el producto por 2 pernos de 3mm. Por consiguiente, entre 1,5 y 2,3 segundos se puede observar como el peso mantiene una disminución hasta llegar a los 0,02 kg, en este punto se cambió el producto por dos anillos planos de 22mm. Así mismo, en el segundo 3,9 se observó un aumento leve a 0,03 kg, esto debido a que se ingresaron 2 anillos de 4mm.

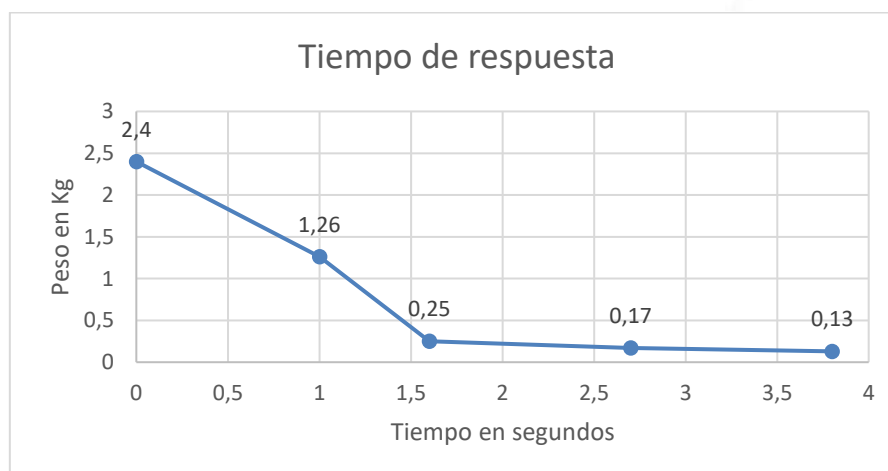


Figura 35. Tiempo de respuesta con un tercer peso de producto

Finalmente, en la figura 35 se pudo visualizar un peso preliminar de 2,40 kg lo que hace referencia al primer peso colocado en la plataforma. Seguido de ello, a los 1,0 segundos, se observa una disminución del peso a los 1,26 kg, dado que hubo un cambio de producto, donde se colocó un total de 14 pernos. Por otro lado, en los 1,6 segundos se ve una disminución a los 0,25 kg dado que se hizo un cambio de producto de un total de 17 anillos planos. Posterior a ello entre los 2,7 y 3,8 segundos se mostró un peso de 0,17 y 0,13 kg, esto dado que se colocó primero 16 anillos planos y luego 10 anillos planos.

Envío de notificaciones vía Telegram



Figura 36. Peso 1 – Espárrago de 1/2

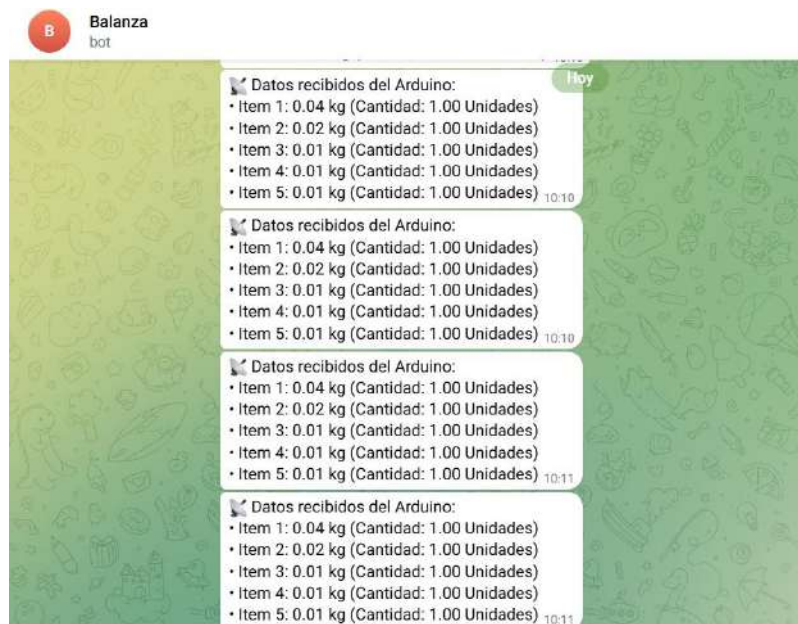


Figura 37. Notificación del peso 1 vía mensaje de texto (telegram)

Como se observa en las figuras 36 y 37, se tuvo la integración de un bot de Telegram en el sistema. En este caso, en el momento en que se capturan los pesos a través de los sensores y posteriormente se muestran en el dashboard, estos datos se envían en tiempo real mediante mensaje de texto, lo cual, permite que los usuarios puedan recibir notificaciones de forma inmediata en el dispositivo de control. Tal como se pudo observar en las imágenes previas, se transmiten los 5 elementos, donde se detalla el peso exacto de cada uno y la cantidad.

Gracias a este proceso, se ayuda a que la información no se limite al entorno del dashboard, sino por el contrario, se logre extender a una vía de comunicación mucha más práctica y de fácil acceso para los usuarios. En ese sentido, el combinar la lectura de los datos, la visualización del sistema y el que se envía de forma automática por Telegram, contribuye al fortalecimiento de la confiabilidad del prototipo, así como también, permite expandir la integración práctica en casos donde el mantener un control y supervisión de remotamente es primordial.

XII. Interpretación de Resultados

12.1. Media Aritmética

Ecuación 4. Formula aritmética permite obtener valores representativos de un conjunto de datos

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x^i \quad (4)$$

Donde

\bar{X} : media aritmética

n : número de datos

Σ : sumatoria

$$\bar{x} = \frac{0,9 + 1,8 + 2,4 + 3,6 + 0,6 + 1,5 + 2,3 + 3,9 + 1 + 1,6 + 2,7 + 3,8 + 0,5 + 1,1 + 2 + 3,5 + 0,5 + 1,1 + 2 + 3,5}{25} =$$

$$\bar{x} = \frac{8,7 + 8,3 + 9,1 + 7,1 + 7,1}{25} = 40,3$$

$$\bar{x} = \frac{40,3}{25} = 1.349 \text{ segundos}$$

Tal como muestran los resultados, la media aritmética de los valores de tiempo muestran un valor de 1,612 segundos, lo cual se encuentra alejado del límite propuesto para medir la eficiencia del tiempo de respuesta del sistema para mostrar los valores del peso de los productos. Con ello, se puede inferir en

que el sistema cuenta con un registro rápido por cada producto, lo que hace que la operatividad sea mucho más ágil incluso frente al cambio de productos, confirmando así el cumplimiento del sistema conforme a los parámetros establecidos en cuanto a rendimiento, agilidad y confiabilidad.

12.2. Desviación estándar

Para el cálculo de la desviación estándar se realizará la sumatoria de todos los pesos obtenidos por las cinco sesiones:

$$\bar{x} = \frac{0,61 + 0,20 + 4,21 + 0,32 + 0,32}{25} = 5,66$$

$$\bar{x} = \frac{5,66}{25} = 0,2264 \text{ kg}$$

Luego se aplica la fórmula de la desviación estándar:

Ecuación 5. Fórmula de la desviación estándar mide que tan dispersos están los datos respecto al promedio

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (5)$$

- σ : desviación estándar
- n: número total de datos
- x_i : cada valor individual del conjunto
- \bar{x} : media aritmética del conjunto
- \sum : sumatoria

$$(x_i - \bar{x})^2$$

Se realiza la sumatoria de cada uno de los cuadrados con lo que se obtiene:

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$
0.25	$0.25 - 0.2264 = 0.0236$	$(0.0236)^2 = 0.000557$
0.13	$0.13 - 0.2264 = -0.0964$	$(-0.0964)^2 = 0.00929$
0.09	$0.09 - 0.2264 = -0.1364$	$(-0.1364)^2 = 0.01860$
0.06	$0.06 - 0.2264 = -0.1664$	$(0.1664)^2 = 0.02770$
0.08	$0.08 - 0.2264 = -0.1464$	$(-0.1464)^2 = 0.02143$

Tabla 1. Valores obtenidos de las mediciones en tiempo real

Y así se continuó sucesivamente con el resto de los datos, se realizó la sumatoria de todas las cantidades, dando como resultado:

Ecuación 6. Permite calcular la lectura de los distintos valores.

$$S = \sum (x_i - \bar{x})^2 = \quad (6)$$

$$\sigma^2 = \frac{S}{n} =$$

$$\frac{6,33}{25} = 0,2532$$

$$\sigma = \sqrt{0.2532} = 0.5033$$

$$\sigma = 0,5033$$

En cuanto a la desviación estándar cuyo resultado fue de 0,5033, se puede observar que existe una variabilidad en la medición de los pesos de los productos que fueron colocados durante las pruebas. A partir de la dispersión observada se pudo determinar que el sistema pudo dar lectura a los distintos valores de manera correcta.

12.3. Tasa de Error Porcentual

x_1	x_2	x_3	x_4	x_5
0,27	0,14	0,10	0,07	0,09
0,09	0,05	0,04	0,03	0,04
2.50	1,30	0,28	0,19	0,15
0,03	0,12	0,08	0,06	0,08
0,03	0,12	0,08	0,06	0,08

Tabla 2. Tabla de valores erróneos

x_1	x_2	x_3	x_4	x_5
0,25	0,13	0,09	0,06	0,08
0,08	0,04	0,03	0,02	0,03
2.40	1,26	0,25	0,17	0,13
0,02	0,11	0,07	0,05	0,07
0,02	0,11	0,07	0,05	0,07

Tabla 3. Tabla de valores medidos

Ecuación 7. Formula aritmética de errores porcentuales

$$TEP = \sqrt{\frac{1}{n} \sum_{i=1}^n \left| \frac{V_{ri} - V_{mi}}{V_{ri}} \right|} \times 100 \quad (7)$$

- V_{ri} : valor real o cantidad del producto esperada en la medición i
- V_{mi} : valor medido por el sistema en la medición i

- n: total de mediciones realizadas
- El resultado se expresa en porcentaje (%)
- \sum : *sumatoria*

Primero se va a calcular el error porcentual para cada uno de los datos:

$$ei = \left| \frac{V_{ri} - V_{mi}}{V_{ri}} \right| \times 100$$

$$ei = \left| \frac{0,27 - 0,25}{0,27} \right| \times 100 = \left| \frac{0,02}{0,27} \right| \times 100 = 7,41\%$$

Así sucesivamente se realiza con cada uno de los datos, luego se procede a realizar la suma de cada uno de los errores porcentuales y se los divide entre el total:

$$\frac{1}{n} \sum_{i=1}^n ei = \frac{96,08}{25} = 3,843$$

Y, finalmente se obtiene la raíz cuadrada del promedio:

$$TEP = \sqrt{3,843} = 3,86\%$$

Finalmente, la tasa de error porcentual obtenida fue de 3.86%, lo que demuestra que existe un error porcentual bajo en relación con los valores reales, esto demuestra que existe una buena precisión a la hora de tomar los pesos.

XIII. Discusión

Conforme al desarrollo del trabajo realizado, el cual se basó en la creación de un sistema IoT para el control y monitoreo de inventario en tiempo real, a continuación, se procede a contrastar cada uno de los estudios consultados durante la investigación, esto a fin de destacar puntos clave y comparar aspectos relevantes que complementen los resultados obtenidos a partir del presente proyecto.

En primera instancia se tiene al artículo [28] en el cual se habla acerca de la integración de tecnologías IoT y computación en la nube en el proceso de gestión de inventarios, centrado en el sector de construcción. En ese sentido, como uno de los aspectos a destacar es que se ayudó a mejorar la trazabilidad de los materiales, lo cual es un punto esencial dentro de este tipo de sector, dado que se afectan tiempos y costos al no mantener una correcta gestión de los mismos para cada obra. Como un punto en común es que ambos se centran en automatizar procesos para potenciar la eficiencia, no obstante, son sectores totalmente distintos dado que el proyecto es para construcción, mientras que el que se aborda en este presente trabajo es para sectores comerciales pequeños. Entre los resultados a destacar del presente estudio, en el primer peso se obtuvo 11 kg, 0,05kg, y 0,07 kg, los cuales tardaron menos de 4 segundos en detectarse, demostrando la eficiencia y efectividad de sistema.

Por otro lado, en el artículo [29] se discute acerca de sistemas automatizados para gestión de inventarios enfocado en empresas medias y pequeñas (PYMES). En este caso, tanto el proyecto [32] como el que se está trabajando coinciden en que su objetivo es reducir los costos y optimizar los niveles de stock, así como también, en el uso de tecnologías IoT para el manejo de inventarios. Sin embargo, el presente trabajo de titulación discrepa en que se han empleado sistemas embebidos de código abierto, lo cual contribuye a la personalización y adaptabilidad de las necesidades concretas de cada entidad, lo cual puede ser considerado una limitación en sistemas más generales como el propuesto en el artículo.

Así mismo en el trabajo [30] cuyo tema se centró en el desarrollo de un sistema inteligente de gestión de inventarios haciendo uso de tecnologías IoT para controlar el stock, se destaca la capacidad que con la cual contaba el sistema para la toma de decisiones en datos actualizados en tiempo real. De esta forma, como un punto asociado al presente trabajo se destaca que ambos proyectos buscan automatizar procesos y mejorar la eficiencia de las operaciones, no obstante, se contrasta que el trabajo de analizado se centra más en la visualización de datos y analítica, mientras que el presente proyecto integra hardware en especial, de la mano de sensores y módulos para comunicación, de tal forma que se logre tener un control y monitoreo de los datos en tiempo real mucho más efectivo.

Por consiguiente, el artículo [31] se presenta un sistema de monitoreo e inventario que hace uso de tecnología RFID, donde la finalidad fue reemplazar los procesos manuales tradicionales por procesos totalmente automatizados. Como punto a destacar de este trabajo es su mejora en la precisión del inventario, de tal forma que se ayudó a reducir pérdidas y a mejorar la satisfacción del cliente. Como aspectos asociados, ambos trabajos comparten el objetivo de automatizar el proceso de gestión de inventarios, sin embargo, se discrepa en que el presente trabajo de titulación integra un conjunto de tecnologías como sensores, microcontroladores, módulos de comunicación, etc, lo cual contribuye a una flexibilidad mayor y mejor funcionalidad. A partir de ello, además de mejorar la precisión, se tiene la posibilidad de integrar diversos datos, enriqueciendo así el análisis y la toma de decisiones.

En el trabajo [32] se expone una propuesta basada en la aplicación de tecnología IoT y algoritmos de IA para brindar una óptima gestión de inventarios. Es decir, que en este trabajo se integró la inteligencia artificial a modo de una herramienta que ayudara a predecir tendencias basadas en el proceso de gestión de inventario. En esto claramente existe un contraste, y es que el artículo se centra en algoritmos predictivos, mientras que en el presente proyecto se busca mejorar un mejor control y manejo de los inventarios a través del uso de sistemas embebidos, no obstante, gracias el artículo investigado en el presente trabajo podría considerarse a futuro la integración de métodos predictivos para mejorar la gestión de inventarios en las pequeñas y medianas empresas.

Seguido de ello, el artículo [33] se basó en el desarrollo de un sistema ciberfísico para logística, donde se utilizaron tecnologías IoT como RFID y bluetooth a modo de sincronizar la información en tiempo real. En contraste con el presente trabajo de titulación, se puede decir que este artículo aborda temas de

entradas y salidas, pero en tema de logística en empresas grandes, lo cual discrepa con este estudio, dado que únicamente se enfoca en monitoreo y gestión de inventarios, sin embargo, si se trata de empresas más grandes, se podría integrar lo mencionado en el artículo [36], lo cual ayudaría a una mayor sostenibilidad y enriquecimiento de proyecto.

Por su parte, el artículo [34] propuso un sistema de gestión de inventarios centrado en almacenes a través del uso de tecnología IoT y visión por computadora. Su enfoque fue la solvencia de problemas frecuentes como errores humanos al contar, ubicaciones erróneas de productos, todo ello mediante el uso de etiquetas RFID, sensores y cámaras inteligentes a fin de monitorizar los movimientos. Contrastando con el presente proyecto de titulación, el cual se basa en la automatización mediante sensores y RFID, el complemento mencionado en artículo [37] es la visión por computadora, lo cual también ayudaría a mejorar la precisión en el seguimiento y la identificación de productos. Es decir, que el integrarla a este presente proyecto permitiría optimizar aún más la gestión de inventarios y reducir errores.

Además, el artículo [35] cuyo tema se centró en la creación de un sistema inteligente de inventarios integrando dispositivos de localización, análisis en la nube y comunicación entre máquinas a fin de optimizar operaciones. En este caso, dentro del artículo se resalta la reducción de errores humanos y un mejor control y seguimiento de stock. Comparado con el proyecto de titulación que hace uso de sistemas embebidos para monitoreo, en este artículo recomienda contar con una visión mucho más holística, puesto que integra análisis de los datos en tiempo real. Con ello, se contribuiría con un elemento más significativo dentro del proyecto, lo cual además de permitir un control de inventario con mayor precisión, permitiría predecir y analizar datos, comportamientos y tendencias a mayor profundidad, aumentando la eficiencia operativa.

En otra parte, el artículo [38] cuyo estudio presenta el diseño de una estación meteorológica basada en FPGA (Field-Programmable Gate Array), implementada en la plataforma Elvis II+ con un chip Xilinx Spartan-3E. Se integran sensores ADM1001 para medir temperatura, humedad, lluvia, dirección y velocidad del viento. El sistema emplea LabVIEW para control e interfaz, y transmite los datos a través de una Raspberry Pi para visualización en tiempo real usando una solución tipo Node-Network. Bajo condiciones normales, se obtuvo una precisión del 85 %, con un error cuadrático medio (RMSE) de 0,2 °C en temperatura y 1,5 % en humedad

XIV. Conclusiones

En primera instancia, la implementación del sistema IoT para el control y monitoreo de inventario en tiempo real, permitió la obtención de datos precisos conforme a la cantidad de productos almacenados, lo cual, fue de mucha ayuda para optimizar de manera significativa el control de stock y a su vez reduciendo el margen de error humano al cual se está propenso al manejar inventarios.

Por su parte, la integración del Arduino UNO R4 WiFi en conjunto con los sensores de peso y el módulo LCD, de la mano con el procesamiento de los datos a través del Raspberry Pi 3, logró demostrar que su eficiencia y correcta funcionalidad. Así mismo, la transmisión de información hacia la plataforma TagoIO conforme al protocolo MQTT o SDK en Python, ayudó a facilitar la visualización de manera remota del inventario, el cual puede ser accedido desde cualquier dispositivo electrónico que cuente con acceso a internet.

Finalmente, a partir de la interfaz diseñada en el dashboard TagoIO, se pudo representar de forma precisa y clara cada uno de los datos recopilados por cada una de las balanzas digitales. Conforme a ello, entre los resultados que destacan, se pudo identificar variaciones de peso que hicieron posible la detección de cambios en el inventario con precisión. Cada uno de los datos se recopilaron el tiempo real y se representaron gráficamente mediante el dashboard, lo que permite que los usuarios puedan monitorear el estado del inventario de forma constante y altamente confiable, lo que demuestra que el sistema desarrollado es eficaz. De manera complementaria, se implementó un sistema de notificaciones en Telegram, el cual transmite los resultados en tiempo real cada 5 segundos, brindando a los usuarios un canal adicional de supervisión inmediata del inventario.

Entre los datos a resaltar, el tiempo promedio de los pesos obtenidos fue de 1.349 segundos, seguido de una desviación estándar de 0,503, demostrando una viabilidad en los pesos obtenidos, y una tasa de error de 3.86%, demostrando que existe un error sumamente bajo, y por ende se comprueba la validez y correcto funcionamiento del sistema.

XV. Recomendaciones

Se sugiere colocar más sensores de peso una vez que el sistema sea implementado en los distintos establecimientos comerciales a futuro, esto con la finalidad de que se cuente con una excelente adaptación con una variedad de productos y mercadería, de modo que se ayude a mantener la precisión al momento de capturar los datos.

Se aconseja además que se protejan los componentes electrónicos que se traten de estructuras o cajas con el objetivo de resguardarlos del polvo, golpes o humedad, específicamente si se tiene planificado realizar instalaciones del sistema en entornos de trabajo mucho más exigentes o que se encuentren bajo condiciones de variación constante.

Así mismo, a fin de mejorar este proyecto a futuro, se recomienda implementar un reconocimiento de productos que son colocados en una balanza distinta a través del uso de inteligencia artificial; donde se haga uso de una cámara para identificar que el producto que se está colocando es erróneo, y que está mal ubicado, donde además de agregará un tipo de alarma que ayude alertar al usuario.

Por último, pero no menos importante, se recomienda optimizar el código del programa que fue implementado en Arduino, donde se puedan integrar rutinas de mantenimiento preventivo y de control de errores, además de la agregación de notificaciones automáticas (por mensaje de texto o e-mail) desde la plataforma TagoIO, esto con la finalidad de advertir acerca de niveles bajos del inventario en tiempo real.

XVI. Referencias Bibliográficas

- [1] C. Morocho, «Gestión de inventarios a través de la clasificación ABC a empresas dedicadas a la venta de materiales de construcción,» *Observatorio de la Economía Latinoamericana*, vol. 7, n° 8, pp. 1-8, 2020, doi:10.5281/zenodo.1234567/inventarios-abc-pulla-2020.
- [2] C. Holguín, *Fundamentos de control y gestión de inventarios*, Universidad del Valle, 2010, <https://doi.org/10.25100/PEU.48>.
- [3] K. Becerra, V. Pedroza, J. Pinilla y M. Vargas, «Implementación de las TIC' S en la gestión de inventario dentro de la cadena de suministro,» *Revista de iniciación científica*, vol. 3, n° 1, pp. 36-49, 2017, doi:10.1234/ric.2017.01.004.
- [4] D. Ramírez, «Transformación Tecnológica en el Modelo de Gestión de Inventarios en las Mipymes, Revisión Bibliográfica,» *Ciencia Latina Revista Científica Multidisciplinar*, vol. 8, n° 1, pp. 3551-3566, 2024, https://doi.org/10.37811/cl_rcm.v8i1.9701.
- [5] J. Hernández, D. Pérez y A. Velazquez, «Medición de parámetros de inventario forestal en bosques plantados, mediante tecnología LiDAR: Comparación de métodos,» *Revista mexicana de ciencias forestales*, vol. 16, n° 87, pp. 72-99, 2025, <https://doi.org/10.29298/rmcf.v16i87.1488>.
- [6] J. Riascos, E. Galván y J. Pincay, «Estrategias basadas en inteligencia artificial para la gestión de inventarios en la cadena de suministro,» *Revista Tecnología En Marcha*, vol. 37, n° 6, pp. 88-97, 2024, <https://doi.org/10.18845/tm.v37i6.7271>.
- [7] Á. Lozada, R. Casallas, J. Bedoya, J. Castellanos y A. Rey, «Dispensador de cajetillas de cigarrillos para Mipyme como experiencia de relación academia-sector productivo, Tuluá,» *Revista Científica*, vol. 43, n° 1, pp. 109-123, 2022, <https://doi.org/10.14483/23448350.17642>.
- [8] M. Espinosa y R. González, «Prototipo de sistema de geolocalización para personas vulnerables usando el módulo A9G y Arduino UNO,» *Ciencia Latina Revista Científica Multidisciplinar*, vol. 7, n° 1, pp. 11359-11388, 2023, https://doi.org/10.37811/cl_rcm.v7i1.5467.
- [9] J. Gómez, J. Suárez, V. López y A. Franco, «Sistema IoT y validación estadística para monitoreo de salud de estudiantes,» *Pädi Boletín Científico De Ciencias Básicas E Ingenierías Del ICBI*, vol. 10, n° (Especial 3), pp. 103-111, 2022, <https://doi.org/10.29057/icbi.v10iEspecial3.9004>.
- [10] F. Iguasnia y J. Rivera, «Sistema Internet of Things (IoT) para la crianza de pollos en granjas avícolas,» *Mikarimin. Revista Científica Multidisciplinaria*, vol. 8, n° 3, pp. 51-64, 2022, doi:10.5281/zenodo.2025.0005678/sistema-iot-crianza-pollos-2022.
- [11] D. Veloz, P. Guerrero, E. Peñafiel y E. Salinas, «Análisis de factibilidad para implementar parqueaderos inteligentes en la Ciudad de Riobamba que promuevan una movilidad sostenible,» *Polo del Conocimiento*, vol. 9, n° 8, pp. 403-423, 2024, <https://doi.org/10.23857/pc.v9i8.7701>.
- [12] J. Vega, M. Contreras y J. Vega, «Trazabilidad de los bultos de cacao mediante el sistema RFID, soportado en el uso de series de tiempo bajo la arquitectura del IoT,» *Cienc. Tecnol. Agropecuaria*, vol. 24, n° 1, pp. 1-28, 2023, https://doi.org/10.21930/rcta.vol24_num1_art:2911.

- [13] D. Hercog, T. Lerher, M. Truntic y O. Težak, «Design and Implementation of ESP32-Based IoT Devices,» *Sensors*, vol. 23, pp. 1-20, 2023, <https://doi.org/10.3390/s23156739>.
- [14] R. Salvador y C. Prieto, «Sistema de seguridad con reconocimiento facial en módulo ESP32,» *Mare Ingenii*, vol. 4, n° 1, pp. 43-52, 2022, <https://doi.org/10.52948/mare.v4i1.684>.
- [15] L. Landázuri, K. Gallo y D. Estrella, «Sistema de Monitoreo/Control de Consumo de energía eléctrica en el hogar mediante Raspberry Pi y Python.: Home energy Consumption monitoring/control System using Raspberry Pi and Python,» *Revista Científica Multidisciplinar G-Nerand*, vol. 4, n° 2, pp. 137-152, 2023, <https://doi.org/10.60100/rcmg.v4i2.126>.
- [16] M. Miranda, J. Hallon y J. Suriaga, «Sistema de cámaras para la detección de mascarillas con Python y Raspberry PI con comunicación inalámbrica a través del sistema global de comunicación móvil (GSM),» *Información Tecnológica*, vol. 33, n° 3, pp. 23-30, 2022, <http://dx.doi.org/10.4067/S0718-07642022000300023>.
- [17] J. Rivera y P. Salinas, «Sistema de Control de Maceración Corporal a Escala Accesible y de Bajo Costo: Integración de Internet de las Cosas-NodeMCU con Arduino-IDE,» *International Journal of Morphology*, vol. 42, n° 5, pp. 1239-1247, 2024, <http://dx.doi.org/10.4067/S0717-95022024000501239>.
- [18] M. Prado, A. Severino, B. Gorotiza y D. Tenorio, «Robótica educativa aplicando el modelo instruccional ADDIE: estrategia didáctica para fortalecer la enseñanza- aprendizaje en la asignatura de Física,» *Revista Latinoamericana Ogmios*, vol. 4, n° 10, pp. 11-28, 2024, <https://doi.org/10.53595/rlo.v4.i10.100>.
- [19] R. Alvarez y C. Bazán, «Sistema de seguridad con reconocimiento facial en módulo ESP32,» *MARE NGENII. INGENIERÍAS*, vol. 4, n° 2, pp. 1-10, 2022, <http://dx.doi.org/10.52948/mare.v4i1.684>.
- [20] J. Basto, L. Núñez, j. Guarín y K. Guarín, «Aprendizaje Autónomo de la Astronomía con Python y TIC,» *Revista Docencia Universitaria*, pp. 8-10, 2024, <https://doi.org/10.18273/revdu.vesp1-2024003>.
- [21] J. Pinargote, V. Lino y B. Vera, «Python en la enseñanza de las Matemáticas para estudiantes de nivelación en Educación Superior,» *MQRInvestigar.*, vol. 8, n° 3, pp. 3966-3989, 2024, <https://doi.org/10.56048/MQR20225.8.3.2024.3966-3989>.
- [22] L. Landázuri, K. Gallo y D. Estrella, «Sistema de Monitoreo/Control de Consumo de energía eléctrica en el hogar mediante Raspberry Pi y Python,» *Revista Científica Multidisciplinar G-Nerando*, vol. 4, n° 2, pp. 1-16, 2023, <https://doi.org/10.60100/rcmg.v4i2.126>.
- [23] M. Zakaria, W. Jabbar y N. Sulaiman, «Development of a smart sensing unit for LoRaWAN-based IoT flood monitoring and warning system in catchment areas,» *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 249-261, 2023, <https://doi.org/10.1016/j.iotcps.2023.04.005>.
- [24] M. Pérez, D. Verde, H. Carvajal, E. Hernández, E. Villalvazo y J. Salome, «Diseño e implementación de una Red de Sensores gestionada por IoT para Aplicaciones de Domótica,» *INVURNUS*, vol. 18, n° 1, 2023, <https://doi.org/10.46588/invurnus.v18i1.60>.
- [25] J. Baquero, A. Yunda, S. Rico y J. Calderón, «Diseño e implementación IoT para temperatura en un proceso de aclimatación de semilla de palma,» *Revista Ingeniería, Matemáticas y*

Ciencias de la Información, vol. 12, n° 23, pp. 13--24, 2025,
<https://doi.org/10.21017/rimci.1103>.

- [26] Y. Córdova, J. Martínez y E. Córdova, «Propuesta de metodología para el diseño de dashboard,» *Revista Cubana De Transformación Digital*, vol. 2, n° 3, pp. 56-76, 2021, <https://doi.org/10.5281/zenodo.5545998>.
- [27] K. Bugwandeen y M. Ungerer, «Exploring the design performance dashboards in relation to achieving organisational strategic goals,» *South African Journal of Industrial Engineering*, vol. 30, n° 2, pp. 161-175, 2019, <http://dx.doi.org/10.7166/30-2-2021>.
- [28] B. Rajesh, M. Haraprasad, S. Indranil y R. Sandip, «Design of smart inventory management system for construction sector based on IoT and cloud computing,» *Elsevier*, vol. 2, p. julio, 2022, <https://doi.org/10.1016/j.prime.2022.100051>.
- [29] F. Ugbebor, M. Adeteye y J. Ugbebor, «Automated Inventory Management Systems with IoT Integration to Optimize Stock Levels and Reduce Carrying Costs for SMEs: A Comprehensive Review,» *Journal of Artificial Intelligence General Science (JAIGS)*, vol. 6, pp. 307-340, 2024. <http://dx.doi.org/10.60087/jaigs.v6i1.257>.
- [30] V. Aher, R. Pol, S. Gaikwad, D. Bhalke, A. Borkar y M. Kolte, «Sistema de inventario inteligente que utiliza IoT y tecnología en la nube,» *Revista Internacional de Sistemas Inteligentes y Aplicaciones en Ingeniería*, vol. 12, n° 4, pp. 187-192, 2023. doi: 10.12345/ijisae.2023.3766.
- [31] E. Setyawan, Y. Ajeng y S. Sekarjatiningrum, «Desarrollo de Sistema Automático de Monitoreo de Inventario en Tiempo Real utilizando Tecnología RFID en Almacén,» *JOIV*, vol. 6, n° 3, pp. 636-642, 2022. <https://dx.doi.org/10.30630/joiv.6.3.1231>.
- [32] R. Kamali, R. Ganesan, M. Dhanush, S. Jeisurya y N. Kamala, «Un sistema avanzado de gestión de inventarios impulsado por IoT y AI para el seguimiento y optimización en tiempo real,» *Revista Internacional para Tendencias e Innovación en Investigación*, vol. 10, n° 4, pp. 173-180, 2025. <https://doi.org/10.1234/ijrti.2025.2504022>.
- [33] W. Wu, C. Cheung, S. Lo, R. Zhong y G. Huang, «Un sistema logístico en tiempo real habilitado por IoT para una empresa de terceros: un estudio de caso,» *Procedia Manufacturing*, n° 49, pp. 16-23, 2020. <https://doi.org/10.1016/j.promfg.2020.06.005>.
- [34] V. Ayoola, G. Osam, C. Umeaku y B. Awotiwon, «Almacenes inteligentes impulsados por IoT con visión por computadora para mejorar la precisión del inventario y reducir discrepancias en sistemas automatizados,» *IRE Journals*, vol. 8, n° 5, pp. 176-193, 2024, <https://doi.org/10.12345/irejournals.2024.1706496>.
- [35] M. Armstrong, J. Fraser, A. Campbell y H. Castro, «IoT-Based Smart Inventory Management in Manufacturing Supply Chains,» *ResearchGate*, pp. 1-5, 2025, <https://doi.org/10.12345/researchgate.2025.391392579>.
- [36] L. Gultom y Y. Vina, «Sistema de seguimiento de inventario basado en IoT para PYMES,» *TEKNOSAINS: Jurnal Sains, Teknologi Dan Informatika*, vol. 11, n° 2, pp. 331-341, 2024, <https://doi.org/10.37373/tekno.v11i2.1092>.
- [37] T. Fernández, Ó. Blanco, I. Froiz y P. Fraga, «Hacia un almacén autónomo en la Industria 4.0: Un sistema basado en UAV y blockchain para aplicaciones de inventario y trazabilidad en la

gestión de la cadena de suministro impulsada por Big Data,» *Sensors*, vol. 19, n° 5, 2019, <https://doi.org/10.3390/sxx010005>.

- [38] H. J. S. Carranza, «Implementation of a Weather Station Using FPGA with Real Time Data Access and Analysis,» *FPGA, Raspberry Pi 4, LabVIEW, Estación meteorológica, Sensores*, vol. 8, n° 2, p. 1, 2025, <https://doi.org/10.29166/ingenio.v8i2.7449>.
- [39] T. Dos Santos, F. De Freitas, D. Carvalho y L. Fernández, «Diseño IoT y validación de sistema de medida para generación fotovoltaica,» *Ingenius Revista de Ciencia y Tecnología*, n° 8, pp. 44-52, 2022, <https://doi.org/10.17163/ings.n28.2022.04>.
- [40] L. García, J. Moreno, J. Pineda y D. Bueno, «Desarrollo De Redes De Sensores Inalámbricos En El Entorno De IoT Mediante Protocolos Wifi,» *IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, Colombia*, pp. 1-6, 2023, <http://dx.doi.org/10.1109/COLCOM59909.2023.10334252>.
- [41] O. Fuentes y M. Romero, «Diseño de un sistema de monitoreo de calidad de agua basado en IoT, aplicado a unidades de producción acuícola,» *Revista Facultad de Ciencias Básicas*, vol. 19, n° 1, pp. 47-58, 2023, <https://doi.org/10.18359/rfcb.7395>.

XVII. Anexos

Anexo 1

Materiales utilizados para el desarrollo del prototipo

En la figura 38 se observan las respectivas celdas de carga, cada una para capturar hasta 10 kg de peso.



Figura 38. Celdas de carga de 10 kg

Por consiguiente, en la figura 39 se muestran los módulos de control para las celdas de carga, los cuales son complementos para generar las mediciones de peso correspondientes.



Figura 39. Módulos de control para celdas de carga

En la figura 40 se observa la pantalla LCD, donde se reflejan los distintos pesos en 0.



Figura 40. Pantalla LCD

La figura 41 muestra los leds de señalización, los cuales permiten establecer la señal en las conexiones.



Figura 41. Leds de señalización

Luego está en la figura 42 el sensor Shieldv5.0, el cual permite conectar con cada uno de los sensores de peso.

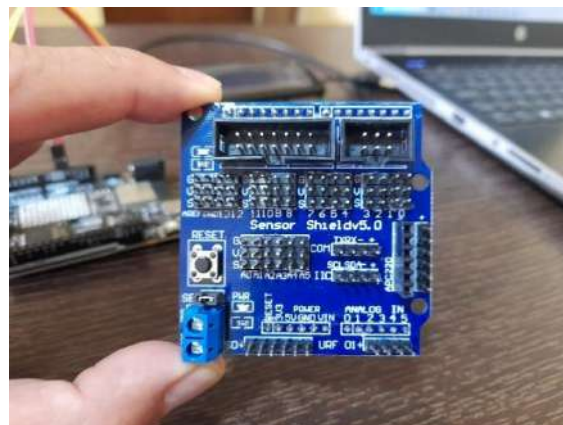


Figura 42. Sensor Shieldv5.0

En la figura 43 se tiene a los recipientes para colocar los productos, los cuales son plásticos para una mayor facilidad de colocación de los productos.



Figura 43. Recipientes para colocar los productos

Por consiguiente, la figura 44 muestra las etiquetas RFID, las cuales hacen posible la identificación de cada uno de los productos del inventario.



Figura 44. Etiquetas RFID

Anexo 2

Montaje del prototipo con cada uno de sus componentes

En la figura 45, se desglosa todo el proceso de montaje del prototipo



Figura 45. Montaje del prototipo

Anexo 3

Creación del dashboard en el entorno de desarrollo

En la figura 46 se observa el proceso de conexión del Raspberry Pi con el dashboard

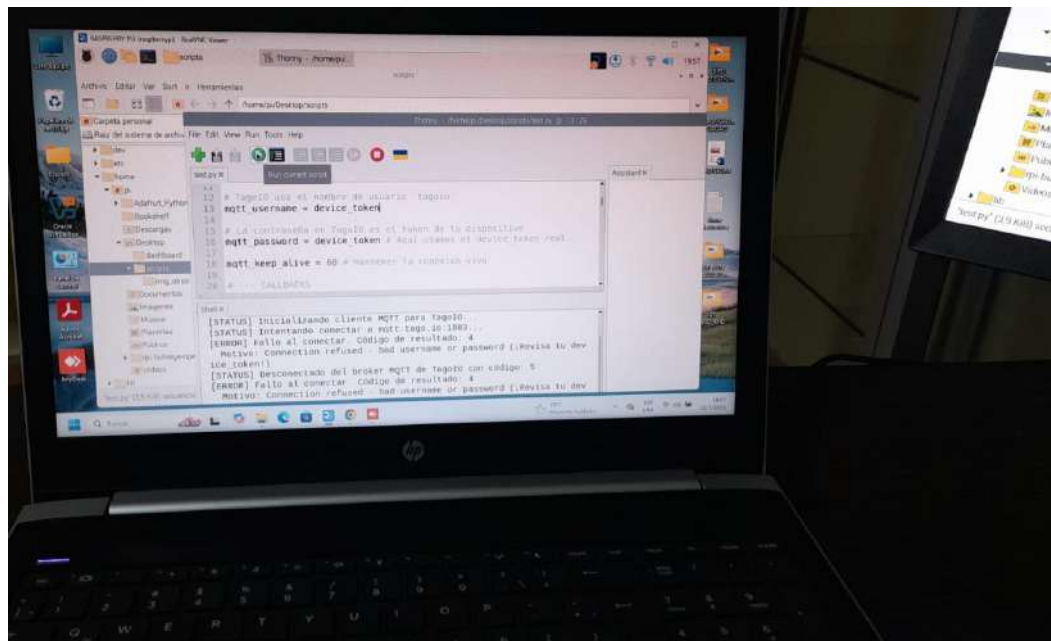


Figura 46. Conexión del Raspberry Pi con el Dashboard

En la figura 47, se observa que la conexión se lleva a cabo mediante un cable, que permite la transmisión de datos.



Figura 47. Se conecta a través del cable, lo que permite la transmisión de datos al monitor

En la figura 48 se expone la configuración del protocolo MQTT en la plataforma TagoIO, donde se establecieron cada uno de los parámetros para una correcta ejecución del sistema.

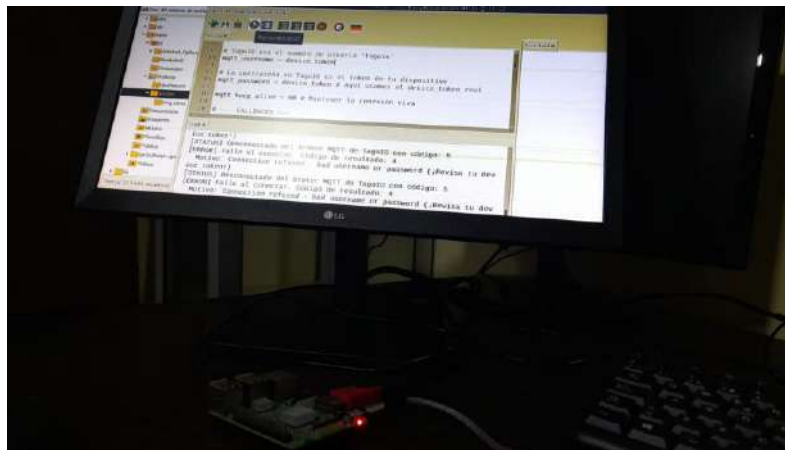


Figura 48. Configuración del protocolo MQTT en la plataforma TagoIO

Y, finalmente en la figura 49, se aprecia el resultado final del dashboard web, con sus respectivas gráficas y datos estadísticos de los pesos tomados.



Figura 49. Resultado final del dashboard web a través de la plataforma TagoIO